ISSN 0181-7450 F : 15,50 €

(B) 16,50 €
(CH) 29,00 FS
(CAN) 20.99 $ca
(And) 15,50 €
DOM surface 16,50 €
DOM avion 16,90 €
N Cal/S 2000 cfp

elektor 503

créer > partager > vendre

SEPTEMBRE & OCTOBRE 2023
ELEKTORMAGAZINE.FR

Raspberry Pi Pico comme analyseur de spectre



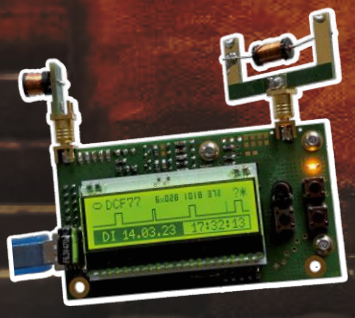
Des FFT avec du matériel bon marché

FOCUS SUR

Communications sans fil

Récepteur de signaux horaires SDR

Prendre le pouls du temps

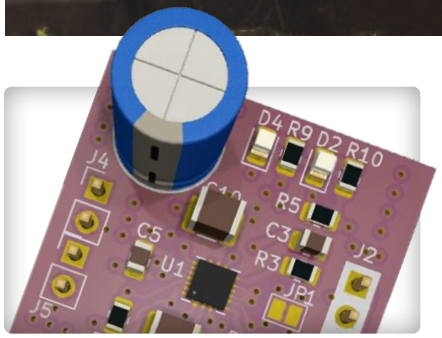


une architecture cellulaire offre-t-elle la plus basse consommation pour l'IdO ?

Caractéristiques énergétiques des technologies LTE-M et NB-IoT

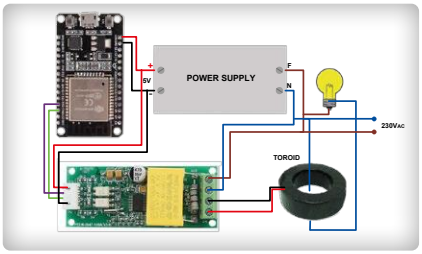
Lora
Un couteau suisse

Le protocole LoRa et ses avantages



Carte d'interface pour pilote de moteur
pilote de moteur à CC de 5 A avec une taille de 3×3 mm

p. 56



Compteur d'énergie basé sur le cloud avec un module ESP32 et un capteur de tension/courant PZEM-004T

p. 102



Régulateur de tension linéaire ±40 V, une alimentation pour le Fortissimo-100... et d'autres !

p. 18

L 19624 - 503 - F: 15,50 € - RD

NOTRE GAMME

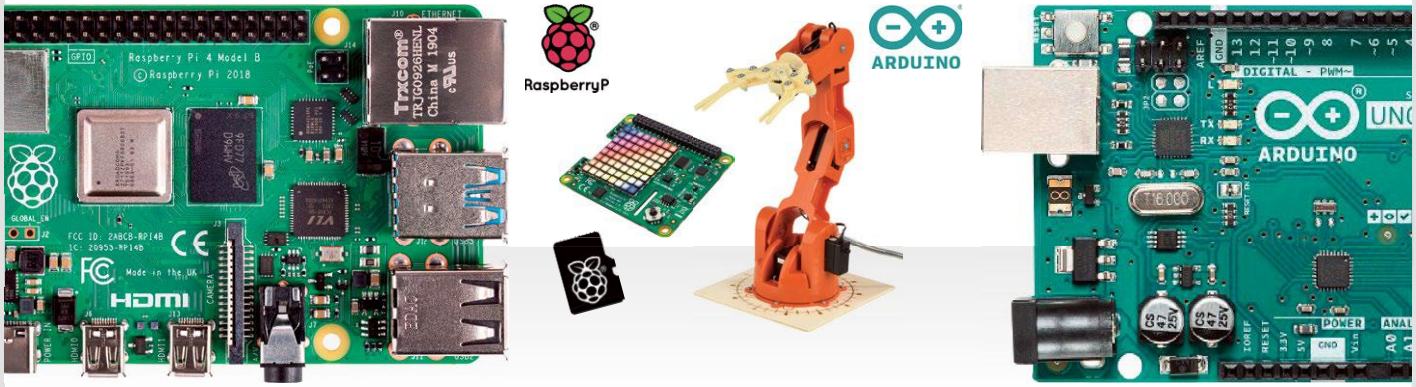
PAR DES TECHNICIENS POUR LES TECHNICIENS

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

Uniquement le meilleur pour vous - provenant de plus de 1500 marques

Nos responsables produits sont employés par Reichelt depuis de nombreuses années et connaissent les exigences de nos clients. Ils rassemblent une large gamme de produits de qualité, à la fois parfaits pour les besoins dans les domaines de la recherche et du développement, la maintenance, l'infrastructure informatique et la production en petites séries et adaptés pour les fabricants.

Cartes de développement : spécialistes compacts pour les projets électroniques



La plate-forme open source idéale pour de nombreux développements et pour une intégration facile dans les applications existantes

▶ https://rch.lt/rpi-fr

Raspberry Pi APPROVED RESELLER

La plate-forme open source de prototypage électronique basée sur des logiciels et du matériel flexibles et faciles à utiliser

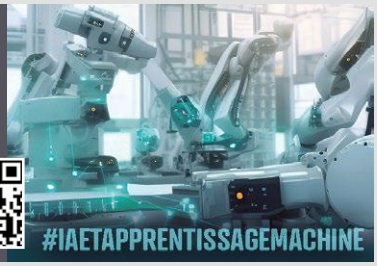
▶ https://rch.lt/arduino-fr

ARDUINO OFFICIAL RESELLER 2021

THE BEST PARTS OF YOUR PROJECTS

IA ET APPRENTISSAGE MACHINE

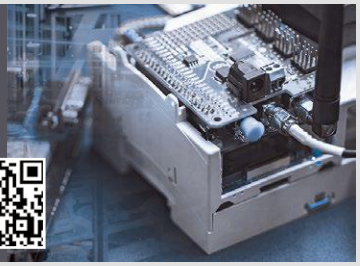
Découvrir maintenant ▶
https://rch.lt/ki-fr



reichelt elektronik MAGAZIN

Article technique - Cartes de développement à usage industriel

Lire maintenant ▶
https://rch.lt/MG784f



Types de paiement : Sofort. VISA MasterCard Maestro PayPal amazon pay

PRIX DU JOUR! Prix à la date du: 1.8. 2023

- Excellent rapport qualité prix
- Plus de 130 000 produits sélectionnés
- Livraison fiable - depuis l'Allemagne dans le monde entier

www.reichelt.com

Assistance téléphonique: +33 9 75 18 03 04

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). Semblables 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

46ème année n° 503
septembre-octobre 2023
ISSN 0181-7450

N° de TVA Intracommunautaire : FR90319937454

Dépôt légal : mai 2023
CPPAP 1125 T 83713
Directeur de la publication : Donatus Akkermans

Elektor Magazine est publié 8 fois par an par
PUBLITRONIC SARL – c/o Regus Roissy CDG
1, rue de la Haye – BP 12910
FR - 95731 Roissy CDG Cedex
www.elektor.fr | www.elektormagazine.fr

Pour toutes vos questions : service@elektor.fr

Devenez membre : www.elektormagazine.fr/abo

Publicité : Ouafae Hassani
Tél. : +31 (0)6 41312932
ouafae.hassani@elektor.com
www.elektormagazine.fr/publicité

Tarifs Annuels :
France 1 an 129,95 € (8 numéros)

Image de couverture :
Midjourney, éditée par Harmen Heida

Droits d'auteur
© 2023 Elektor International Media B.V.

Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente publication, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'oeuvre dans laquelle elles sont incorporées (Loi du 11 mars 1957 -art. 40 et 41 et Code Pénal art. 425).
Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

Imprimé aux Pays-Bas par Senefelder Misset, Mercuriusstraat 35, 7006 RK Doetinchem

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

Certifié PEFC
Ce produit est issu de forêts gérées durablement et de sources contrôlées
PEFC
PEFC/30-31-151
www.pefc-france.org

ÉDITO

Jens Nickel

rédacteur en chef d'Elektor Magazine



Du simple au moderne

Le sans-fil occupera une grande place dans notre avenir : actuellement, plusieurs entreprises et universités mènent déjà des recherches sur la 6G, qui devrait permettre d'atteindre des débits de données allant jusqu'à 400 Gb/s et d'exploiter des fréquences allant jusqu'à 3 THz, et tout cela devrait commencer aux alentours de 2030. Aujourd'hui, les applications sans fil sont déjà nombreuses. On peut citer entre autres : la géolocalisation, les liaisons sans fil à faible consommation et l'utilisation des smartphones comme télécommandes. Nous avons eu tellement d'idées d'articles pour ce numéro, qu'il nous a été impossible de tout publier dans une seule édition. Nous avons choisi de maintenir la diversité des sujets d'électroniques, très appréciée par nos lecteurs aux intérêts variés. Certains de nos articles sur le sans-fil – dont un rapport intéressant sur la détermination de la position avec une précision centimétrique à un coût raisonnable pour les makers – seront publiés dans les prochains numéros.
Et pour ceux qui s'intéressent particulièrement à des domaines spécifiques tels que « l'électronique de puissance », « le sans-fil », « les tests et mesures » ou « la production », nous publierons bientôt des pages thématiques spéciales sur notre site web, elektormagazine.fr. Vous y trouverez, non seulement des articles de notre magazine, mais aussi du contenu en ligne exclusif. Les articles de fond et les tutoriels (vidéos) pour les débutants seront accompagnés de projets supplémentaires réalisés par les membres de notre communauté internationale.
J'y présenterai peut-être aussi un de mes projets très personnels, un système de haut-parleurs sans fil pour l'extérieur, alimenté par des piles au lithium. Bien qu'il ne comprenne (encore) que de composants achetés, le projet a déjà nécessité beaucoup de développement. Vous êtes peut-être curieux de savoir quelle norme sans fil hyper moderne j'utilise ? Eh bien, c'est la bonne vieille radio FM, mais à 863…865 MHz, qui est également utilisée dans les soirées « Silent Disco ». Lorsqu'il s'agit d'une installation simple et des latences les plus faibles, la simplicité reste la clé !



Proposez une contribution à Elektor!

Vos propositions sont les bienvenues ! Vous souhaitez proposer un article, un tutoriel vidéo ou une idée de livre ? Consultez le guide de l'auteur et la page de soumission d'Elektor :

www.elektormagazine.com/submissions



Elektor Labs : idées et projets

La plateforme Elektor Labs est ouverte à tous. Publiez des idées et des projets électroniques, discutez des défis techniques et collaborez avec les autres.

www.elektormagazine.fr/labs

notre équipe

Rédacteur en chef : Jens Nickel | **Rédaction :** Asma Adhimi, Roberto Armani, Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf (RG), Ton Giesberts, Hedwig Hennekens, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Clemens Valens, Brian Tristam Williams | **Contributeurs réguliers :** David Ashton, Tam Hanna, Ilse Joostens, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | **Maquette :** Harmen Heida, Sylvia Sopamena, Patrick Wielders | **Des questions techniques :** redaction@elektor.fr

DANS CE NUMÉRO

Volume 49 – Nº 503
Septembre et octobre 2023



Raspberry Pi Pico comme analyseur de spectre

des FFT avec du matériel bon marché

6

Rubriques

3 **Édito**

FOCUS
30 **5 000 € à gagner !**
participez au STM32 Wireless Innovation Design Contest

33 **2023 : l'odyssée de l'IA**
démarrer avec l'interpréteur de code de ChatGPT

53 **Zone D**
des potentiomètres logarithmiques

62 **sur le vif**
danger de l'électronique

81 **drôle de composant**
relais à verrouillage

92 **Ethics in Action**
Due Diligence Directive

94 **démarrer en électronique...**
...amplificateur de tension

122 **Hexadoku**
casse-tête pour elektorniciens



LoRa, un couteau suisse (1)
le protocole LoRa
et ses avantages

36

Articles de fond

FOCUS
36 **LoRa, un couteau suisse (1)**
le protocole LoRa et ses avantages

50 **les nouvelles cartes Arduino UNO R4**
Minima et WiFi

FOCUS
78 **YARD Stick One**
un outil de test sans fil pour les fréquences allant jusqu'à 1 GHz

112 **guide de programmation *Bare-Metal* (2)**
timer précis, UART et débogage

Industrie

FOCUS
64 **une architecture cellulaire offre-t-elle la plus basse consommation pour l'IdO ?**
caractéristiques énergétiques des technologies LTE-M et NB-IoT pour le déploiement de réseaux LPWAN

FOCUS
67 **communication sans fil dans les systèmes IoT à l'aide de modules Arduino MKR**
normes de communication des modules Arduino pour l'IoT

70 **pertes en courant alternatif dans les composants magnétiques**
évitez la surchauffe des inductances !

73 **optimiser son déploiement cloud grâce aux appareils de mesure**

FOCUS
76 **implémentation de Matter**
que faut-il pour déployer les appareils Matter ?





Projets

FOCUS

6 Raspberry Pi Pico comme analyseur de spectre
des FFT avec du matériel bon marché

18 régulateur de tension linéaire ±40 V
une alimentation de qualité pour l'amplificateur de puissance Fortissimo-100… et d'autres !

FOCUS

24 la communication sans fil des microcontrôleurs devient flexible
l'EEPROM ouvre des perspectives de mise en réseau pour les microcontrôleurs sans fil

42 puit de courant réglable avec générateur d'horloge intégré
testez vos alimentations, convertisseurs de tension et batteries

56 carte d'interface pour pilote de moteur
Un BoB pour un pilote de moteur à courant continu de 5 A avec une taille de 3×3 mm

FOCUS

82 PIC O'Clock – prendre le pouls du temps
conception d'un récepteur de signaux horaires SDR

97 enregistreur d'infrasons avec l'Arduino Pro Mini
un projet extrait du livre « Arduino & Co. » d'Elektor

FOCUS

102 compteur d'énergie basé sur le cloud
avec un module ESP32 et un capteur de tension/courant PZEM-004T

Bientôt dans ces pages

Le numéro de novembre et décembre 2023
Le thème du prochain numéro du magazine Elektor sera « prototypage et production ». Vous trouverez ainsi toutes les informations dont vous avez besoin pour réaliser votre idée de projet : du concept au schéma, du circuit imprimé au produit !

Quelques-uns des points forts :

- Services de fabrication de circuits imprimés
- Outils d'IA pour les développeurs
- BLE en pratique
- Systèmes GNSS RTK à faible coût
- Plaque chauffante maison
- ChatGPT améliore les micrologiciels
- Outils open-source
- Un boîtier simple créé par Fraiseuse CNC
- Introduction à KiCad 7
- Guirlande de Noël solaire

et bien d'autres choses encore !

Le numéro de novembre – décembre 2023 du magazine Elektor sera publié aux alentours du 15 novembre 2023. La date d'arrivée du magazine papier chez les abonnés dépend des aléas d'acheminement. Le contenu et les titres des articles peuvent être modifiés.



FOCUS SUR
Communications sans fil

Raspberry Pi Pico
comme analyseur de spectre

des FFT avec du matériel bon marché

Martin Ossmann (Allemagne)

Cet article montre comment construire un analyseur de spectre logiciel avec un simple RPi Pico. Une version de base avec une résolution de 12 bits et une fréquence d'échantillonnage de 500 kHz est très bien adaptée aux mesures audios. Avec un CA/N externe, on peut même atteindre 50 MHz !

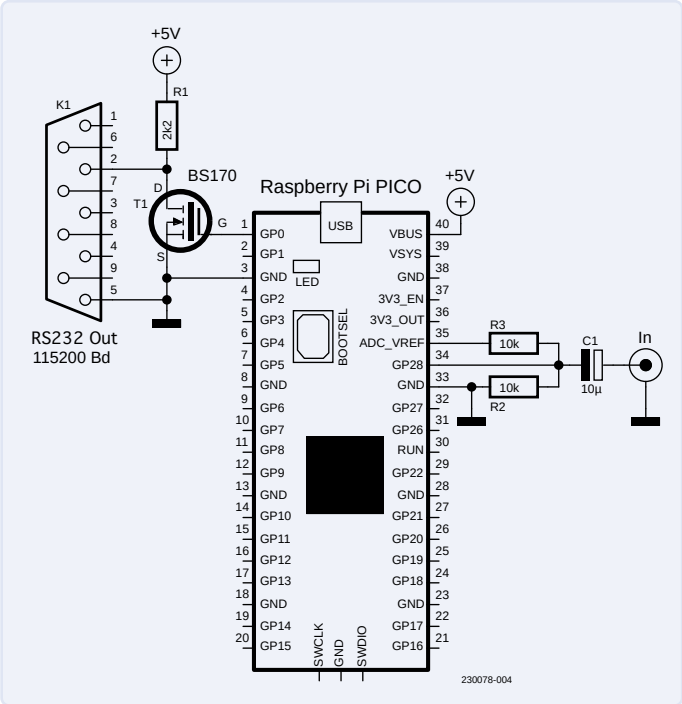


Figure 1. Câblage minimal de la carte Pico pour le transfert de données vers le PC.

Avec le CA/N intégré du microcontrôleur RPi RP2040, les signaux analogiques sont échantillonnés jusqu'à 500 kHz avec une résolution de 12 bits. Dans la configuration la plus simple, c'est un PC qui effectue l'analyse fréquentielle et affiche les résultats. Une première extension avec la connexion d'un écran LCD permet déjà d'obtenir un analyseur autonome. On peut ensuite, avec un CA/N externe, atteindre jusqu'à 50 MHz, ce qui permet à l'analyseur de traiter des signaux jusqu'à 25 MHz.

500 kHz et connexion PC

Pour les premières expérimentations, le RPi Pico est câblé comme indiqué à la **figure 1**. R2 et R3 élèvent le point zéro du CA/N à la moitié de la plage de niveaux. C1 élimine les composantes continues du signal d'entrée. L'entrée utilisée, ADC2, utilise la broche 34 ainsi que le GPIO28. T1 est utilisé avec R1 pour établir un niveau TTL pour la sortie des données vers le PC via l'interface série à 115 200 bauds. Comme le circuit est assez simple, il peut être facilement mis en œuvre sur une platine d'expérimentation, comme le montre la **figure 2**.

Le CA/N fonctionne initialement avec la fréquence d'échantillonnage par défaut de 500 kHz. Il transfère ses échantillons de 12 bits à une unité FIFO, d'où ils entrent dans un tampon qui peut stocker *N* valeurs. *N* vaut généralement 1024. Lorsque la mémoire tampon est pleine, les valeurs sont envoyées en série au PC.

TFD et FFT

Le spectre est calculé à l'aide de la transformée de Fourier discrète (TFD). Cet algorithme prend *N* échantillons s_n (avec $n = 0$ à $N-1$) du signal d'entrée pour calculer *N* valeurs de spectre Z_k (avec $k = 0$ à $N-1$). La règle de transformation appliquée est la suivante :

$$Z_k = \sum_{n=0}^{N-1} s_n e^{-\left(\frac{2\pi i k n}{N}\right)} = \sum_{n=0}^{N-1} s_n \left[\cos\left(\frac{2\pi kn}{N}\right) - i \sin\left(\frac{2\pi kn}{N}\right)\right]$$

La TFD se compose de N nombres complexes. Si le signal d'entrée est réel, comme dans le cas présent, la TFD résultante est symétrique et la règle suivante s'applique :

$$Z_{N-k} = \overline{Z}_k$$

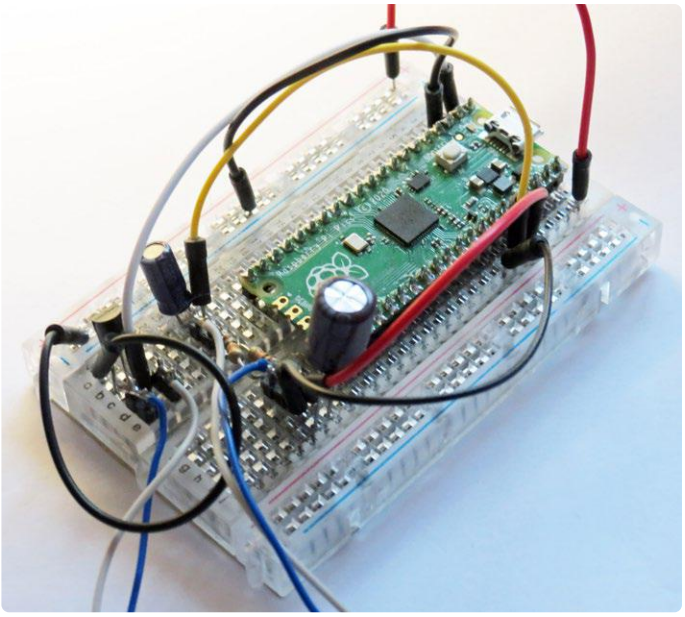


Figure 2. Ma maquette du circuit de la figure 1.

Seules les $N/2$ premières valeurs sont donc pertinentes. Ceci est conforme au théorème d'échantillonnage, selon lequel des valeurs uniques ne sont obtenues que jusqu'à la moitié de la fréquence d'échantillonnage. Par conséquent, seules les $N/2$ premières valeurs spectrales sont utilisées dans ce qui suit. Le spectre calculé est celui du signal d'entrée S prolongé avec la période N. Afin de comprendre pourquoi la TFD effectue effectivement une analyse spectrale, considérons d'abord comment le signal d'entrée $s_n = cos(2\,\pi\,m\,n\,/\,N)$ est transformé avec, par exemple, $m = 5$ et $N = 16$. Ce signal d'entrée est une oscillation cosinusoïdale de fréquence $f_s = m\,/\,N$.
Les valeurs transformées résultantes sont $Z_m = Z_5 = N/2 = 8$, et les autres valeurs, Z_k, sont toutes nulles. Ainsi, une oscillation de fréquence m devient perceptible exactement dans la $m^{ème}$ valeur transformée. Si, au lieu du signal cosinus, on transformait un signal sinus, la partie imaginaire de Z_m ne serait pas nulle. Mais que se passe-t-il lorsqu'on transforme un signal composé d'oscillations de différentes fréquences ? Là aussi, c'est simple : La TFD est une transformation linéaire, ce qui signifie que la TFD d'une somme est égale à la somme des TFD des signaux individuels. La formule qui en résulte est la suivante

$$TFD(u\,U_n + v\,V_n) = u\,TFD(U_n) + v\,TFD(V_n)$$

avec les préfacteurs u et v.
Si le signal d'entrée est composé de plusieurs oscillations $s_n = cos(2\,\pi\,m\,n\,/\,N)$ avec différentes fréquences m, on obtient exactement les oscillations individuelles dans le spectre. Un signal $s_n = A\,cos(2\,\pi\,m\,n\,/\,N)$ d'amplitude A *se traduit* par une valeur $Z_m = A\,N\,/\,2$ dans le spectre. Cette valeur dépend du nombre d'échantillons. Pour l'éliminer, la TFD est mise à l'échelle par un facteur de $2\,/\,N$. Le nouveau spectre a alors les valeurs $W_k = 2\,Z_k\,/\,N$. Un signal cosinus d'amplitude A donne alors une valeur de spectre d'exactement A. La valeur d'amplitude A est la valeur de crête du signal cosinus. Pour la valeur effective, il faut la diviser par $\sqrt{2}$.

Temps de calcul

Le temps de calcul de la TFD croît quadratiquement avec le nombre N d'échantillons. Il peut donc être assez long lorsque N est élevé. Des mesures effectuées avec le RPi Pico ont conduit à l'approximation $T_{TFD} \approx 9\,N^2\,\mu s$. Si N est une puissance de deux, vous pouvez utiliser la transformée de Fourier rapide (FFT) au lieu de la TFD, car elle calcule les mêmes valeurs beaucoup plus rapidement. Sur la carte Pico, le temps de calcul est $T_{FFT} \approx 4\,N\,log_2\,(N)$. Le **tableau 1** montre les temps de calcul pour quelques valeurs pratiques de N. Pour une analyse de spectre simple, on utilise souvent $N = 1024$. Le temps de calcul de la TFD serait encore acceptable ici avec environ 9 s, mais la FFT est beaucoup plus rapide avec 40 ms. Il arrive également que l'on utilise $N = 16\,384$, pour lequel la TFD prendrait déjà 40 bonnes minutes – une lenteur inacceptable. La FFT, elle, a besoin de moins d'une seconde ! C'est pourquoi N est toujours choisi comme une puissance de deux pour privilégier la FFT.

Table 1: Computing times for different N.

N	TFD en s	FFT en s
256	0,590	0,008
1,024	9,437	0,041
4,096	150,995	0,197
16,384	2415,919	0,918

Valeur efficace et dBm

En ingénierie des communications, les quantités sont souvent spécifiées en tant que valeurs efficaces. Comme on élève au carré la quantité fonction du temps, qu'on en calcule la moyenne avant d'en prendre la racine carrée, le terme de racine carrée moyenne (RMS) est devenu courant. Pour un signal discret et périodique de période N, comme pour nos signaux TFD, la valeur RMS est calculée comme suit :

$$RMS_s = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}|s_n|^2}$$

Il est également possible de calculer la valeur efficace des valeurs spectrales. On obtient alors :

$$RMS_Z = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}|Z_k|^2} \quad \text{et} \quad RMS_W = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}|W_k|^2}$$

Il est intéressant de noter que la formule suivante s'applique désormais :

$$RMS_S = RMS_Z\,/\,\sqrt{N} \quad \text{ou} \quad RMS_S = \sqrt{N}\cdot RMS_W\,/\,2$$

Cela signifie qu'on peut aussi calculer la valeur efficace dans le domaine fréquentiel au lieu du temporel. Pour cela, il faut seulement prendre en compte le préfacteur. Cette relation est également appelée identité de Parseval [1]. L'énergie dans le domaine temporel est donc reflétée avec précision par la somme des énergies des oscillations individuelles dans le domaine fréquentiel. Les rapports d'énergie ou de puissance sont souvent exprimés en dB dans l'ingénierie des communications. Pour le rapport de puissance

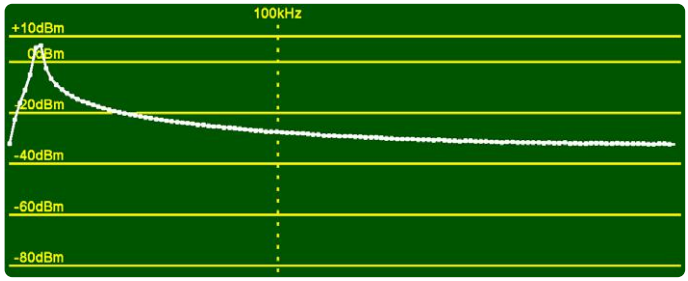


Figure 3. Mauvaise représentation spectrale à une fréquence demi-entière.

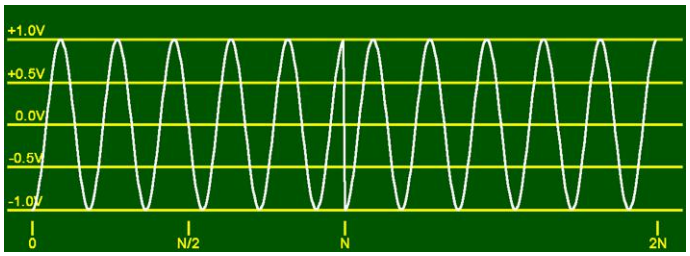


Figure 4. À l'échantillon N, un saut d'amplitude se produit si le signal est prolongé périodiquement.

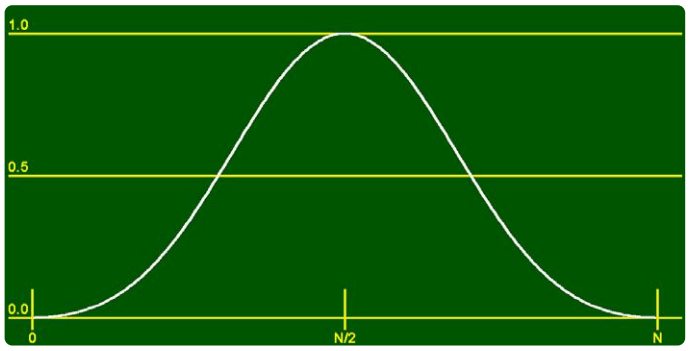


Figure 5. Fonction fenêtre de Blackman.

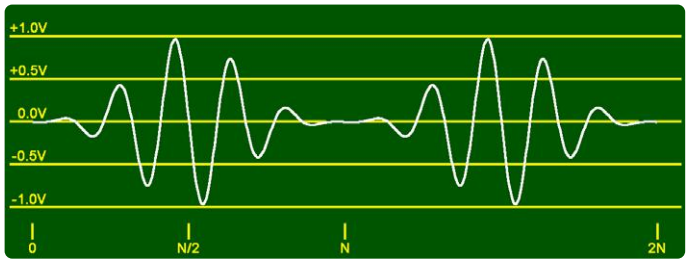


Figure 6. Le signal multiplié par la fonction fenêtre présente une courbe continue aux extrémités.

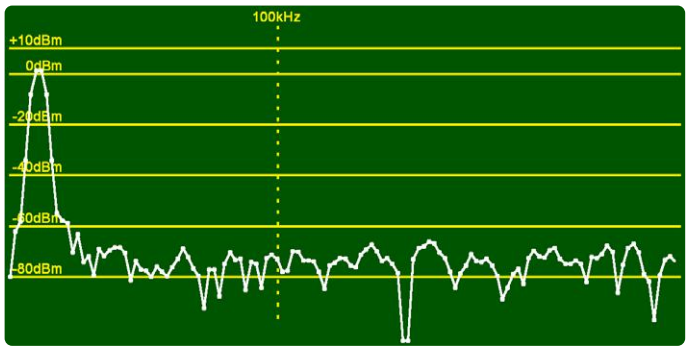


Figure 7. Spectre après application de la fonction fenêtre.

$v = P_1 / P_0$, on obtient $v_{dB} = 10\ log_{10}\ (P_1 / P_0)$. Pour les niveaux absolus, le niveau de référence $P_0 = 1\ mW$ est courant. On obtient alors $P_{dBm} = 10\ log_{10}\ (P / 1\ mW)$. Si l'on veut spécifier des tensions U en dBm, on utilise la puissance P au niveau d'une résistance de $R = 50\ \Omega$, à laquelle est appliquée la tension U_{RMS}. La puissance dissipée par la résistance étant $P = U^2 / R$, on obtient $U_{dBm} = 10\ log_{10}\ (U^2_{RMS} / R / 1\ mW)$.
Pour l'analyseur de spectre décrit ici, l'axe Y est gradué en dBm, la représentation logarithmique habituelle pour les analyseurs de spectre, mais attention : en ingénierie audio, les dBm se réfèrent souvent à une résistance de 600 Ω.

Signaux et fenêtres

Jusqu'à présent, nous avons considéré des signaux d'entrée qui s'inscrivaient exactement dans une fenêtre de N valeurs. La fréquence m du signal $s_n = A\ cos(2\ \pi\ m\ n\ /\ N)$ était donc un nombre entier. La **figure 3** montre le spectre à $m = 5{,}5$ et $N = 256$.
En fait, on s'attendrait à un spectre linéaire car le signal ne contient qu'une seule fréquence. Au lieu de cela, on obtient un spectre qui chute lentement de part et d'autre des valeurs maximales. Ce spectre est, bien entendu, inexploitable pour un analyseur de spectre. La raison en est que le signal périodique continu (de période N) présente un saut à ses extrémités. On le voit clairement dans le diagramme amplitude/temps de deux périodes de la **figure 4** : à l'échantillon N, le signal passe de +1 V à -1 V. Ce saut génère des composantes spectrales à haute fréquence au-delà de la fréquence du signal. Pour éviter ces effets, on applique la technique de la fenêtre, qui équivaut en principe à superposer une enveloppe. Il s'agit de multiplier le signal d'entrée par une fonction fenêtre, qui le réduit progressivement à l'amplitude 0 aux limites de la fenêtre, mais ne le modifie pas beaucoup ailleurs. La **figure 5** montre la fonction fenêtre utilisée.
La figure 6 montre deux périodes du signal multiplié par la fonction fenêtre. On voit clairement comment le signal est modulé en amplitude, de sorte qu'il ne saute pas aux extrémités de la période (0, N, 2N…), mais se déroule de manière continue. Il existe différentes fonctions fenêtre, chacune présentant des avantages et des inconvénients différents. Le listage en langage C suivant montre comment on calcule la fenêtre de Blackman (trois termes) utilisée ici :

```
double windowFun(int k) {
  double alpha=0.16;
  double a0=(1-alpha)/2;
  double a1=0.5;
  double a2=alpha/2;
  return a0-a1*cos(2*PI*k/(N-1))+a2*cos(4*PI*k/(N-1));
}
```

La fonction fenêtre est composée de deux fonctions cosinus. Après application de la fenêtre, la FFT donne le spectre de la **figure 7**. Le maximum spectral occupe quatre intervalles (appelés bins) autour de $m = 5{,}5$ et diminue ensuite de manière significative. Avec des valeurs plus élevées pour N, l'élargissement du spectre est tout à fait acceptable et l'on obtient de bons spectres linéaires.

Toutefois, on doit tenir compte du fait que la fonction fenêtre réduit l'amplitude du signal. En tant que facteur de fenêtre, on peut approximer en utilisant la moyenne arithmétique des valeurs de la fonction fenêtre, comme dans le listage suivant.

```
float getWindowFactor() {
  float mean=0;
  for (k=0; k < N ; k++) {
    mean += abs(windowFun(k));
  }
 return mean/N ;
}
```

Pour compenser la réduction de l'amplitude, nous divisons le spectre par le facteur de fenêtre après avoir calculé la FFT. Lorsqu'on utilise une fonction fenêtre, l'identité de Parseval ne s'applique plus.

Options de l'analyseur

L'analyseur de spectre offre plusieurs options configurables. Dans la configuration la plus simple, le µC échantillonne le signal, calcule le spectre, l'émet et répète l'opération à l'infini. La fréquence d'échantillonnage f_s est égale à la fréquence d'échantillonnage. Elle définit en même temps la plage de fréquence du spectre puisqu'elle va en principe de 0 à $f_s/2$. Ainsi, à 500 kHz, on peut analyser des composantes de fréquence jusqu'à 250 kHz. La longueur du signal N détermine le nombre d'échantillons utilisés pour le spectre et définit donc la résolution de la FFT. Deux lignes spectrales adjacentes (bins) sont espacées de $\Delta = f_s / N$. À 500 kHz et $N = 1024$, la résolution est $\Delta = 500\ kHz / 1024 \approx 488\ Hz$. Ce qui suffit pour traiter les signaux RF de radiodiffusion.

Lors de l'analyse de signaux comportant des composantes de bruit, les spectres individuels sont souvent très irréguliers. On peut améliorer cette situation en calculant la moyenne des spectres successifs. Cette procédure est activée par l'option « MEAN » du logiciel de l'analyseur, qu'on peut d'ailleurs télécharger gratuitement sur le site Internet d'Elektor à l'adresse [2]. Le spectre de l'émetteur de signaux temporels DCF77 permet de voir l'intérêt du moyennage. Ici, la fréquence d'échantillonnage est de 250 kHz et N vaut 16 384. La **figure 8** montre le spectre résultant dans la plage de ±5 kHz autour de la fréquence centrale de 77,5 kHz.

La fine ligne blanche montre un spectre unique manifestement bruité. Le calcul de la moyenne (courbe verte épaisse) montre clairement que la DCF77 a un spectre de bruit. En effet, la DCF77 transmet les données avec un bruit pseudo-aléatoire à un débit binaire de $77{,}5\ kHz / 120 \approx 645{,}8\ Hz$, en plus de la modulation d'amplitude (bande très étroite). La largeur de bande de ±645,8 Hz est représentée par les deux lignes verticales.

Analysons ensuite le spectre de l'émetteur EFR (129,1 kHz). EFR transmet des données RTTY avec un décalage de ±170 Hz et un débit de 200 Bd. Ainsi, deux lignes spectrales devraient être visibles dans le spectre, à 128,93 kHz et 129,27 kHz. Cependant, étant donné que seuls des messages courts sont envoyés avec de longues pauses entre eux, la fréquence à +170 Hz n'apparaît que rarement et est donc difficile à détecter. Le logiciel propose à cet effet la fonction *MAXHOLD*, qui permet de calculer le maximum des spectres sur

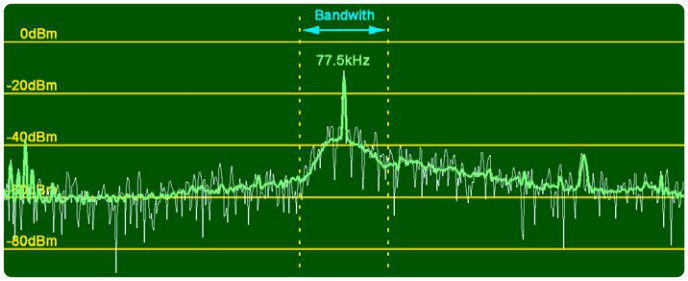


Figure 8. Le spectre de DCF77 « normal » (blanc) et son spectre moyenné (vert).

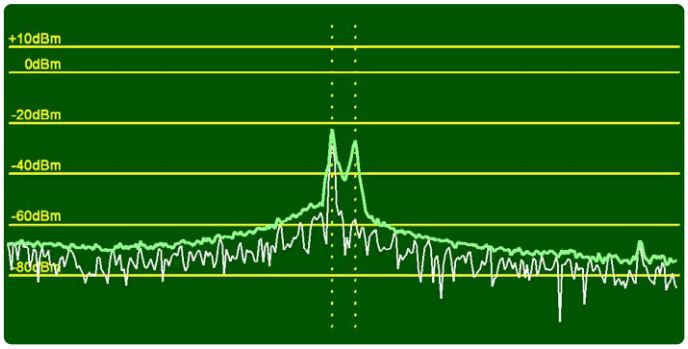


Figure 9. Spectre de DCF49 (EFR à 129,1 kHz) avec la fonction MAXHOLD.

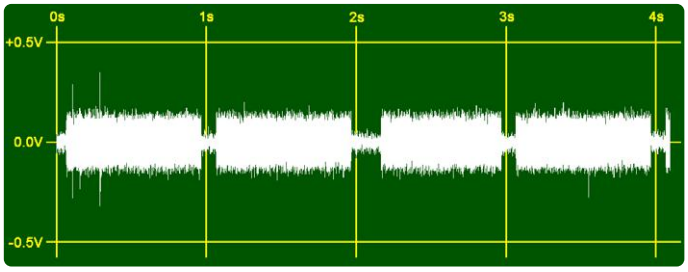


Figure 10. Amplitude en fonction du temps du signal DCF77.

un grand nombre de spectres. De cette manière, les composantes spectrales apparaissant brièvement à 129,27 kHz sont plus visibles (voir **figure 9**).

La fine ligne blanche est un spectre unique. Comme il n'y a pas de message en cours d'émission, seul le pic à 128,93 kHz est visible. Le spectre MAXHOLD est représenté par la courbe verte plus épaisse. Les deux pics à 129,1 kHz ± 170 Hz (au niveau des lignes verticales pointillées) sont clairement visibles. Pour obtenir un spectre aussi finement résolu, de nombreux échantillons sont nécessaires avec $N = 16\ 384$. La fréquence d'échantillonnage est de 500 kHz et la résolution, en conséquence, est $\Delta = 500\ kHz / 16\ 384 \approx 30{,}5\ Hz$. Ainsi, les deux lignes spectrales ne sont qu'à 340 Hz / 30,5 Hz ≈ 11 échantillons de fréquence l'une de l'autre.

Domaine temporel (fonction d'oscilloscope)

Bien entendu, le signal dans le domaine temporel est également instructif, ne serait-ce que pour contrôler l'étendue de la plage d'acquisition du CA/N. Le logiciel peut également le faire en affichant simplement le signal échantillonné. La **figure 10** montre une sorte d'oscillogramme du signal DCF77 à 77,5 kHz.

Étant donné que les chutes de la porteuse se produisant toutes les secondes sont particulièrement intéressantes, il est conseillé de choisir une fréquence d'échantillonnage plus faible et un nombre

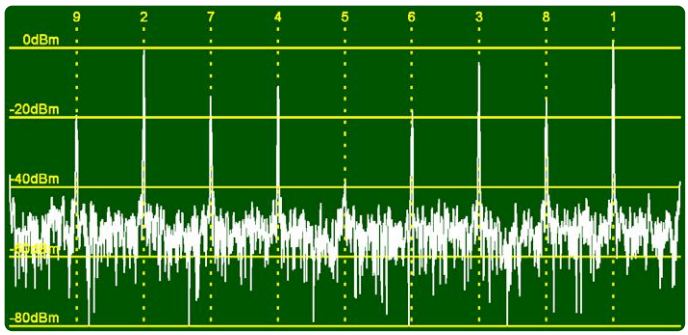


*Figure 11. Sous-échantillonnage d'un signal carré de 225 kHz avec f_s = 500 kHz. Un certain nombre de lignes spectrales apparaissent à f_k = 225 kHz * k avec k = 1, 2, 3…*

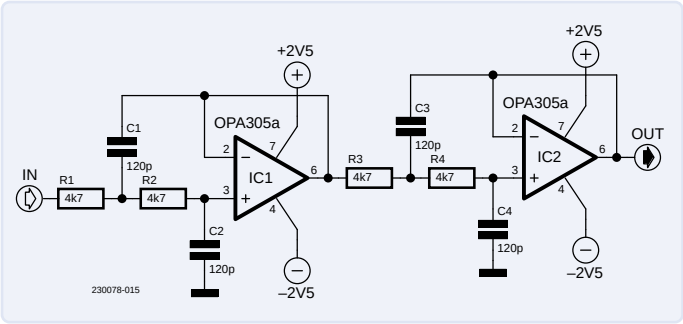


Figure 12. Filtre anti-repliement utilisant deux filtres de Sallen-Key du deuxième ordre et une fréquence de coupure de 280 kHz.

d'échantillons plus élevé. Pour la représentation de la figure 10, on a choisi f_s = 4 kHz et N = 16 384. La fenêtre temporelle a donc une taille de $T_s = N/f_s \approx 4$ s, ce qui se traduit par quatre arrêts de la porteuse dans la figure 10. Le signal de 77,5 kHz est fortement sous-échantillonné, mais ce n'est pas un problème, car seule l'amplitude est importante ici.

Théorème d'échantillonnage

Selon le théorème d'échantillonnage, les signaux à analyser ne peuvent contenir des fréquences que jusqu'à un maximum de la moitié de la fréquence d'échantillonnage. Soit on respecte cette règle, soit on ajoute un filtre passe-bas approprié avant le CA/N. Jusqu'à présent, aucun filtre approprié n'a été proposé, car il dépend toujours de l'application spécifique. En outre, il est également intéressant d'étudier les conditions d'analyse des fréquences plus élevées. Si nous échantillonnons des signaux avec des impulsions infiniment courtes (Dirac), nous obtenons un spectre périodique avec une période égale à l'intervalle d'échantillonnage. Les lignes spectrales contenues dans le signal sont donc répétées périodiquement avec la largeur f_s. Par conséquent, des fréquences plus grandes que f_s apparaissent également dans le spectre. Elles sont pour ainsi dire « repliées » dans l'intervalle fondamental. Si la distance d'échantillonnage est plus longue, les fréquences plus élevées sont atténuées plus fortement. Les relations sont illustrées dans l'exemple suivant : la fréquence d'échantillonnage f_s est de 500 kHz. On analyse un signal carré d'une fréquence de 225 kHz et d'un rapport cyclique de 10 %.

Dans le spectre de la **figure 11**, l'onde fondamentale à 225 kHz apparaît le plus à droite et a le niveau le plus élevé. La première harmonique (k = 2) a la fréquence 2 * 225 kHz = 450 kHz, ce qui est déjà au-delà de $f_s/2$. En raison de la symétrie, elle apparaît donc à f = 500 kHz – 450 kHz = 50 kHz dans le spectre sous la ligne verticale pointillée étiquetée « 2 » avec une amplitude encore considérable. La fréquence pour 9 est égale à 225 kHz * 9 = 2025 kHz. Comme cela équivaut à 4 * 500 kHz + 25 kHz, ce pic à 25 kHz apparaît à l'extrême gauche du spectre, sous la ligne du curseur intitulée « 9 ». Les indices de la ligne de curseur k = 1…9 représentent les positions théoriques attendues des harmoniques du signal analysé avec des fréquences de k * 225 kHz, connues sous le nom de « fréquences repliées ». Bien que les fréquences aillent jusqu'à plus de 2 MHz, dix pics apparaissent dans la fenêtre qui s'étend jusqu'à 250 kHz. Cela signifie que le CA/N peut certainement traiter des fréquences plus élevées que $f_s/2$. Cet effet est utilisé dans ce que l'on appelle le sous-échantillonnage. Avant l'échantillonnage, le signal peut être filtré par un filtre passe-bande avec une bande passante de $f_s/2$ pour que seules les zones pertinentes du spectre soient capturées. Pour ce faire, il faudrait ajouter une fonction d'échantillonnage rapide et de maintien à l'installation afin de pouvoir quand même traiter les hautes fréquences avec le CA/N. Si toutefois cet effet était gênant, il faudrait filtrer le signal à l'aide d'un filtre passe-bas (filtre anti-repliement) afin de supprimer les composantes du signal à haute fréquence avant l'échantillonnage.

Filtre anti-repliement

L'effet d'un filtre passe-bas anti-repliement est démontré ci-dessous. Ici, un signal carré symétrique de 225 kHz avec un rapport cyclique de 50 % est filtré à travers le filtre passe-bas illustré à la **figure 12** avant le CA/N. Le filtre a été simulé avec LTspice [3] et construit avec des ampli-op du type OPA2350.

Le spectre du signal filtré de la **figure 13** montre clairement que les harmoniques sont fortement atténuées et que seule l'onde fondamentale de 225 kHz se détache nettement au-dessus de -40 dBm. Cependant, notre filtre atténue déjà fortement à 225 kHz. Un filtre aux extrémités plus abruptes serait donc plus approprié.

Bruit et ENOB

Jusqu'à présent, seuls les signaux composés d'oscillations sinusoïdales individuelles de fréquences et d'amplitudes différentes ont été étudiés. Dans la pratique, cependant, il existe aussi des signaux de bruit avec différentes largeurs de bande. Leur traitement en termes d'ingénierie des communications n'est pas vraiment trivial, c'est pourquoi seuls des cas simples sont décrits. À titre d'exemple, des valeurs de bruit n_k avec une amplitude A sont générées pour un bruit binaire avec une fréquence de bruit f_n. Dans chaque cas, un dé est lancé pour déterminer si s_k = +A ou s_k = -A doit s'appliquer à la valeur suivante. Au départ, f_n = 1 MHz s'applique, de sorte que les valeurs sont générées plus rapidement que le CA/N ne peut les échantillonner. Les échantillons sont donc indépendants les uns des autres et la valeur est ±A à chaque fois. Par conséquent, le signal de bruit est presque « blanc », ce qui signifie que toutes les fréquences sont présentes de manière égale dans son spectre.

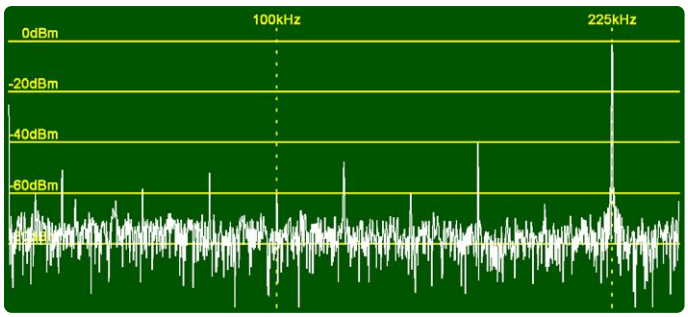


Figure 13. Spectre du signal carré après anti-repliement.

Le signal est généré par un logiciel sur le RPi Pico. Pour que cette tâche puisse être exécutée en parallèle avec le logiciel de l'analyseur de spectre, elle utilise le deuxième cœur de l'UC RP2040. Elle est implémentée à l'aide d'une machine à états PIO, qui permet de générer jusqu'à 125 Mb/s.
En règle générale, on utilise la valeur RMS de la tension de bruit. Comme le signal n'est pas périodique, nous ne pouvons pas utiliser formellement la formule pour la RMS. Cependant, pour un grand nombre d'échantillons M, la valeur RMS peut être approchée avec :

$$RMS_n \approx \sqrt{\frac{1}{M}\sum_{k=0}^{M-1} n_k^2}$$

Pour un bruit binaire, on peut donc facilement calculer la valeur RMS. Comme chaque échantillon a la valeur absolue A, nous obtenons l'équation simple $RMS_n = A$. Soit, dans la configuration d'essai, $A = 750\ mV$. La **figure 14** montre un exemple de ce signal et la **figure 15** montre le spectre associé. La fine ligne rouge est un spectre unique. Elle est très irrégulière parce qu'elle ne représente qu'une seule instance du spectre. La ligne verte épaisse est le résultat du calcul de la moyenne de nombreux spectres (fonction MEAN) et montre plus clairement qu'il s'agit d'un bruit blanc à un certain niveau.
Si nous examinons les spectres de bruit avec différentes longueurs de signal N, nous pouvons voir que le niveau affiché dépend également de N. La relation suivante existe entre la valeur efficace, le niveau affiché et N : pour un grand N, la règle suivante s'applique : $RMS_s \approx RMS_n$. En raison de l'identité de Parseval (voir ci-dessus), la relation suivante s'applique également :

$$RMS_W = 2\ RMS_S\ / \sqrt{N}$$

Ainsi, l'énergie dans

$$RMS_W = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1} |W_k|^2}$$

sera répartie uniformément sur w_k . Pour simplifier, on suppose que tous les w_k sont de taille égale et qu'ils sont égaux à w. En rassemblant tout cela, nous obtenons : $w^2 = 4\ RMS^2_n / N$. Ici, w est exactement la puissance recherchée du signal de bruit dans le spectre. Dans la **figure 15**, ce niveau est représenté par une ligne violette épaisse. Il s'agit en fait d'une très bonne prédiction du niveau de bruit réel. On comprend également pourquoi ce niveau diminue lorsque N augmente : c'est parce que l'énergie du bruit est répartie sur un plus grand nombre de lignes spectrales.
Dans l'exemple suivant, le débit binaire de la source de bruit est réduit à $f_n = 25\ kHz$. Par conséquent, 20 échantillons successifs sont toujours égaux, ce qui équivaut à une sorte de filtrage passe-bas. Le spectre résultant est illustré à la **figure 16**.
La fine ligne rouge montre à nouveau la fonction avec un seul spécimen. Le spectre moyen est représenté en vert. Le premier point zéro du spectre se situe à 25 kHz. Le spectre est proportionnel à $H(f) = si(\pi\ f / f_N)$, où $si(x) = sin(x) / x$ est ce que l'on appelle la fonction sinus cardinal. En simplifiant, on peut calculer le niveau autour de

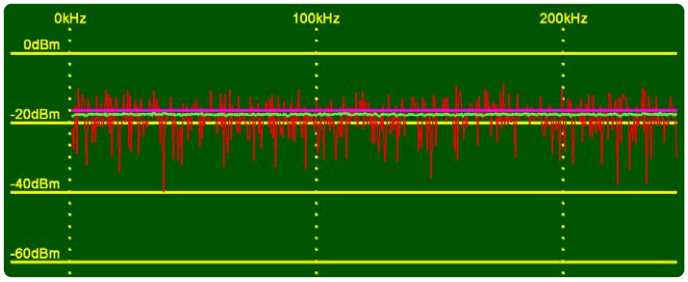


Figure 14. Bruit binaire sous forme de signal dans le domaine temporel avec une amplitude de ±750 mV.

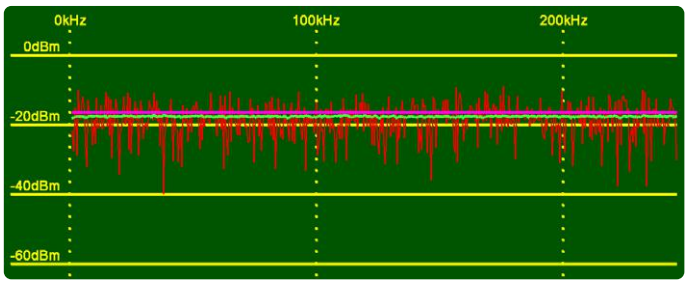


Figure 15. Spectre du bruit binaire de la figure 14.

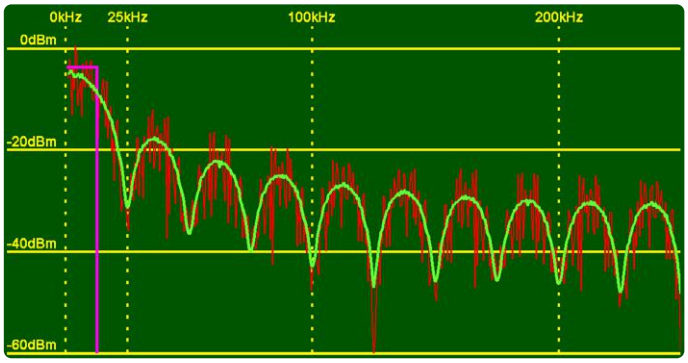


Figure 16. Spectre du bruit à un débit binaire de 25 kHz.

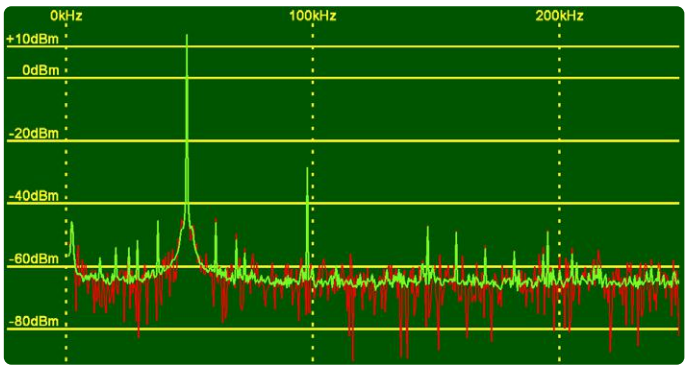


Figure 17. Spectre d'un signal sinusoïdal de 48 828 kHz à une fréquence d'échantillonnage de 500 kHz.

0 comme suit : La largeur de bande est d'environ $f_{BW} = 12{,}5\ kHz$, ce qui équivaut à $k_{BW} = N\ f_{BW} / f_s = 25$ lignes spectrales.
Si le spectre a une largeur de bande constante de w, l'équation $RMS^2_w = 2\ k_{BW}\ w^2 / N$ est obtenue pour les valeurs RMS en additionnant les lignes spectrales dans la largeur de bande. Le facteur 2 est nécessaire car la FFT fournit un spectre symétrique avec les lignes spectrales N - k_{BW} à N. En utilisant l'identité de Parseval, la puissance spectrale w est calculée comme suit : $w^2 = 2\ RMS^2_n / k_{BW}$.

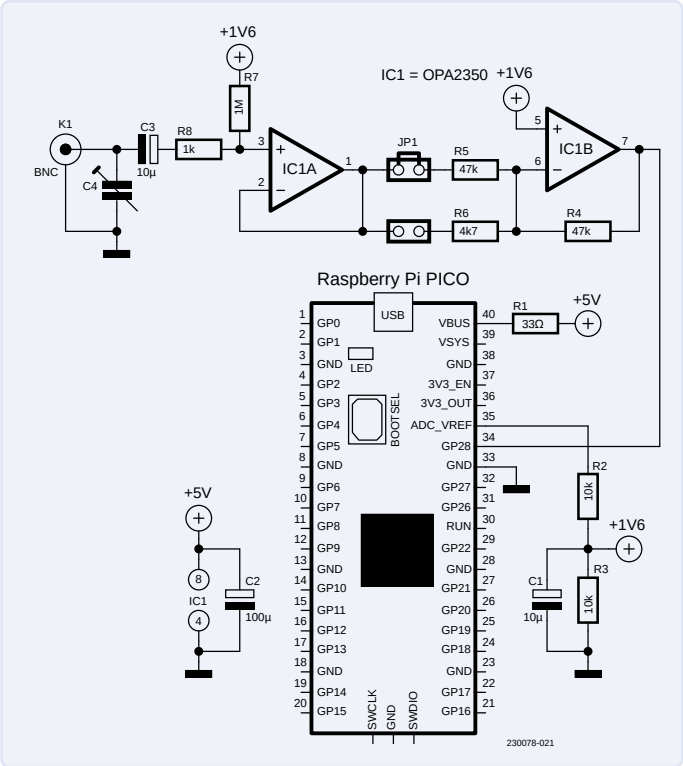


Figure 18. Préamplificateur avec une résistance d'entrée de 1 MΩ. Pour IC1, il est possible d'utiliser des amplificateurs optoélectroniques AD823 ou MCP602.

Elle est représentée à la **figure 16** par la ligne violette épaisse et correspond assez bien au spectre courant.
L'exemple suivant est un peu plus pratique : le CA/N reçoit le signal d'un générateur sinusoïdal d'amplitude 3 V_{SS} (environ 14 dBm). La fréquence du signal $f_s = 48{,}828\ kHz$ correspond exactement à la 100ème ligne spectrale. La **figure 17** montre le spectre obtenu. La ligne spectrale à f_s est bien visible et a également une taille correcte. À 2 f_s, on peut voir la première harmonique causée par le générateur de signal avec un niveau d'environ 45 dB en dessous du signal. De nombreuses petites lignes spectrales sont également visibles, car les composants ne sont pas idéaux et présentent par exemple de petites non-linéarités. Le niveau de bruit est également intéressant. Le spectre moyen (vert) montre clairement qu'il est d'environ -65 dBm. Avec la formule donnée ci-dessus, nous pouvons convertir le niveau en tension de bruit effective d'environ 2 mV_{RMS} (environ 5 mV_{SS}). La résolution du CA/N est de $3{,}3\ V / 4096 = 0{,}8\ mV$. Le bruit couvre donc environ six niveaux de quantification. Plus de 2 bits sont donc perdus à cause du bruit. Néanmoins, les pics spectraux à -55 dBm se détachent bien du bruit. La plage dynamique est donc d'au moins $14\ dBm - (-55\ dBm) = 69\ dB$ - un facteur de ≈2800. La résolution réelle ou le nombre effectif de bits (ENOB) du CA/N atteint plus de 11 grâce au moyennage, ce qui n'est pas mal du tout. À noter que les expérimentations sur le bruit ont été réalisées sans fenêtrage afin que les niveaux soient conservés exactement et que l'identité de Parseval soit applicable.

Préamplificateur avec entrée 1 MΩ

Les oscilloscopes ont généralement une résistance d'entrée de 1 MΩ. Cela signifie qu'ils n'imposent normalement pas de charge significative aux objets de mesure et permettent d'utiliser les sondes 1:1 et 10:1 habituelles. Cette commodité devrait également être offerte par un analyseur de spectre, raison pour laquelle a été conçu le préamplificateur de la **figure 18**.

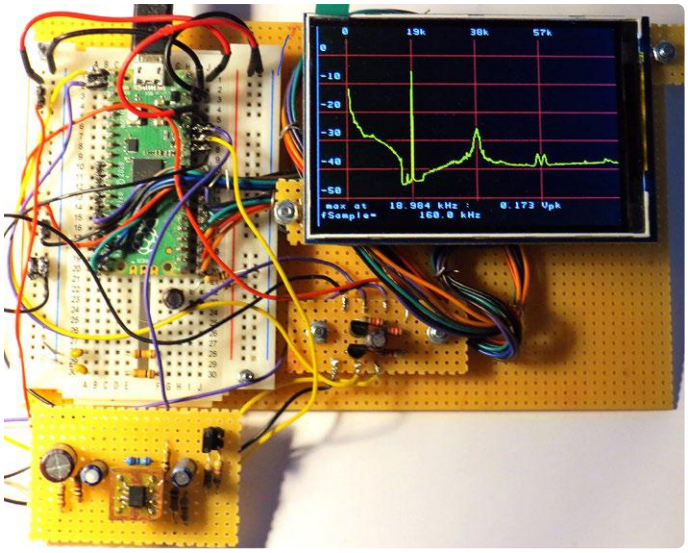


Figure 19. Analyseur composé d'une carte Pico avec écran LCD et d'un préamplificateur de 1 MΩ, montrant un signal multiplex VHF.

L'amplificateur IC1A agit comme un convertisseur d'impédance. L'impédance d'entrée est déterminée par R7. IC1B agit comme un préamplificateur avec un gain de 1 ou 10 sélectionnable via JP1. IC1A doit avoir un faible courant de polarisation, sinon ce courant au niveau de R7 provoquerait une chute de tension trop importante, ce qui éloignerait la ligne zéro du signal du centre de la plage d'acquisition du CA/N. Par exemple, le type AD8042 avec son courant de polarisation $I_B = 1{,}2\ \mu A$ entraînerait un décalage de $1{,}2\ \mu A * 1\ M\Omega = 1{,}2\ V$. La figure 18 montre des d'ampli-op de type approprié. Si nécessaire, l'adaptation d'une sonde peut être réalisée avec C4. La **figure 19** montre le prototype construit sur une platine avec la carte Pico et l'écran.

Horloge et fréquence d'échantillonnage du CA/N

Selon la fiche technique, l'horloge du CA/N doit être de 48 MHz. Cette horloge est normalement générée par la PLL USB. Le prédiviseur sur 16 bits n'est pas activé dans ce cas (division par 1). Le CA/N fonctionne en *mode libre*, c'est-à-dire qu'il acquiert et émet des valeurs en continu. La synchronisation est assurée par un compteur du CA/N qui est déclenché par son horloge. Le compteur du CA/N déborde à la valeur *CA/NclkDiv* et redéclenche alors le CA/N. Étant donné qu'une conversion nécessite 96 impulsions d'horloge, on doit respecter $CA/NclkDiv \geq 96$. À la plus petite valeur de $CA/NclkDiv = 96$, nous obtenons la fréquence d'échantillonnage la plus élevée de $f_s = 48\ MHz / 96 = 500\ kHz$. Le compteur du CA/N est un compteur fractionnaire 16:8 : La partie entière a une largeur de 16 bits que vous pouvez diviser par fractions de 8 bits pour produire des fréquences d'échantillonnage même peu communes aussi précisément que possible. La fréquence d'échantillonnage la plus lente est $f_s = 48\ MHz / 65\,536 = 732\ Hz$.
Dans l'expérience suivante, le CA/N est utilisé bien au-delà de sa spécification pour une fréquence d'échantillonnage de 1 MHz. Pour cela, l'horloge du CA/N doit être de 96 MHz. L'horloge PLL de l'USB avec ses 48 MHz n'est pas suffisante. Le CA/N est donc cadencé par la PLL de l'horloge système *SYS-PLL*. La fréquence d'horloge générée pour le processeur est normalement de 125 MHz. Cependant, si elle est réglée sur 96 MHz, on peut l'utiliser pour cadencer directement le CA/N et permettre une fréquence d'échantillonnage de 1 MHz. Malheureusement, le processeur fonctionne alors environ 23% plus lentement. Ce n'est toutefois pas un problème,

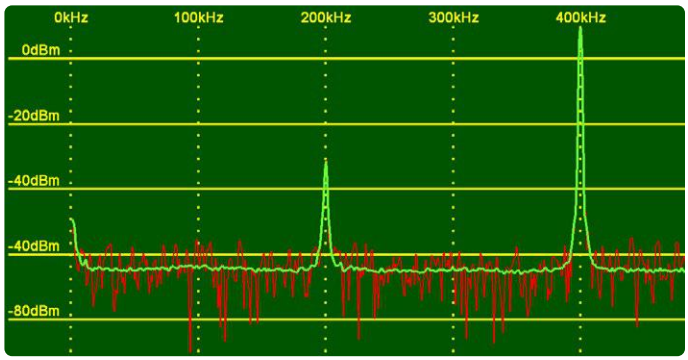


Figure 20. Spectre d'un signal sinusoïdal de 400 kHz à une fréquence d'échantillonnage de 1 MHz.

car le logiciel n'utilise de toute façon pas le processeur au maximum de ses capacités.

Un signal sinusoïdal de 400 kHz avec une amplitude de 2 VSS est utilisé comme signal de test. La **figure 20** montre le spectre obtenu. Le signal utile est représenté avec le niveau correct à la position correcte. Cependant, la première harmonique à 800 kHz est repliée à 1 MHz – 800 kHz = 200 kHz. Le niveau de bruit de -65 dBm n'est pas pire qu'à une fréquence d'échantillonnage de 500 kHz. Il ne semble donc pas y avoir d'objection sérieuse à cadencer le CA/N à 96 MHz.

LCD

Comme les détails fins sont souvent intéressants dans les spectres, on a choisi le shield Arduino MAR3502 en raison de sa résolution relativement élevée de 320×480 pixels. Il est connecté en parallèle à la carte Pico, comme le montre la **figure 21**. Cela permet de configurer un analyseur de spectre comme un appareil autonome, mais il manque toujours des commandes.

La **figure 20** montre le spectre d'un signal audio FM multiplex sur l'écran. Vous pouvez clairement voir le signal mono (G+D) dans la gamme à 17 kHz. À 19 kHz, la ligne spectrale de la tonalité pilote stéréo apparaît. À 38 kHz, le spectre du signal de différence stéréo (G - D) est visible. À 57 kHz, le spectre du signal RDS est visible, composé de deux bandes latérales autour de 57 kHz.

CA/N 12 bits, 50 MHz

Pour les basses fréquences, la fréquence d'échantillonnage de 500 kHz du CA/N intégré dans le RP2040 suffit, mais il existe également des spectres intéressants dans la gamme des radiofréquences. Pour cela, la carte Pico peut être complétée par un CA/N externe rapide utilisant la puce ADS807E. Avec ce CA/N 12 bits, il est possible d'obtenir jusqu'à 53 MHz. Pour les spectres, vous avez besoin d'une grande puissance de calcul et donc de quelques astuces pour atteindre cette fréquence d'échantillonnage élevée. L'ADS807E fournit commodément une tension de référence intégrée. Le CA/N est connecté à la carte Pico comme le montre la **figure 22**.

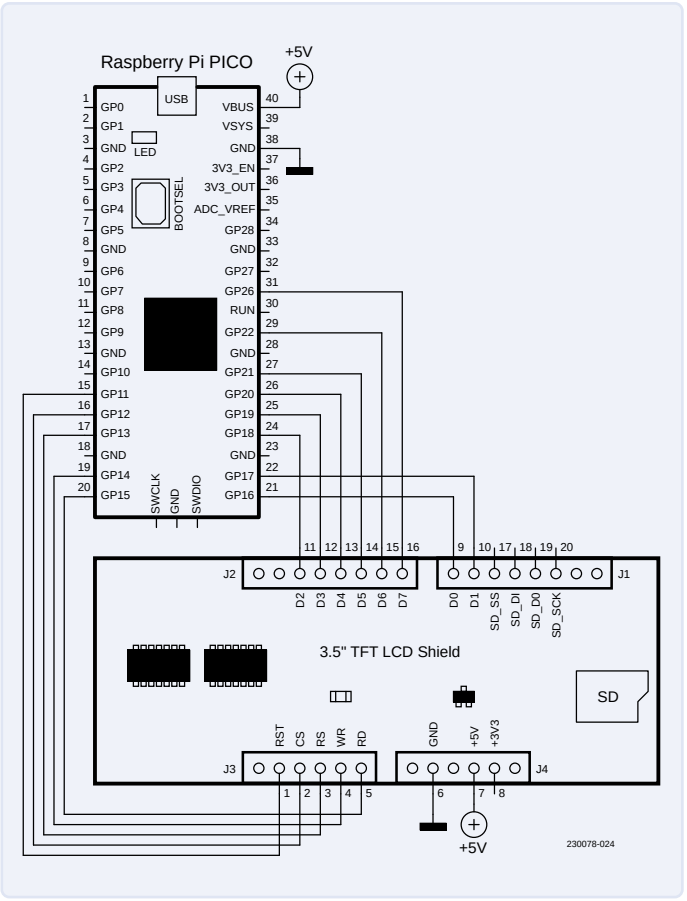


Figure 21. Connexion parallèle d'un écran LCD 3,5" 480×320 pixels à la carte Pico.

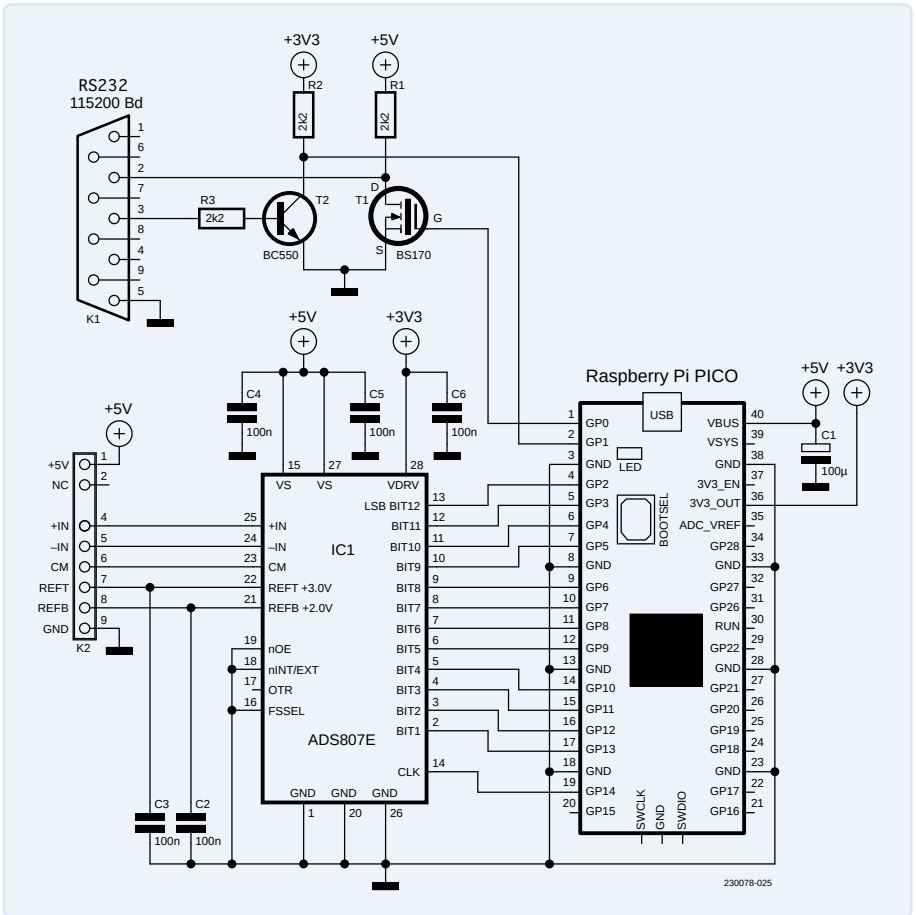


Figure 22. Connexion du CA/N externe ADS807 à la carte Pico.

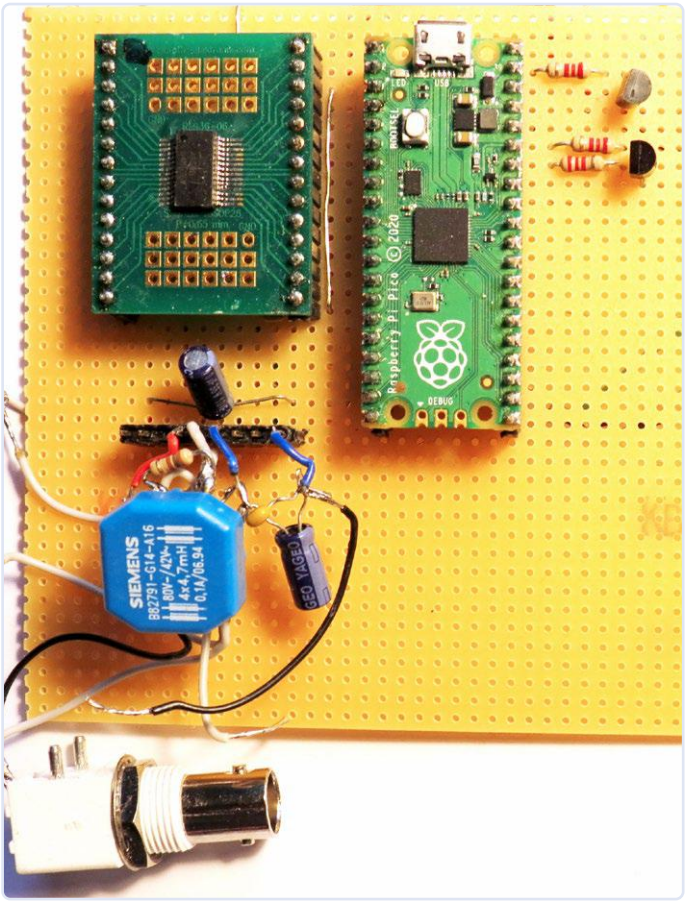


Figure 23. Carte Pico supplémentaire avec ADS807&

En raison de la connexion parallèle, de nombreuses broches GPIO sont occupées, et il n'est donc malheureusement pas possible de connecter le LCD en même temps. Pour cette raison, l'ADS807 reçoit une deuxième carte Pico. Lors des premières expérimentations, les données sont traitées et affichées à l'aide du PC, comme au début. Différents frontaux analogiques, présentés ci-dessous, peuvent être connectés à K2. La **figure 23** montre la configuration de test d'une carte Pico avec un ADS807 et un transformateur 50 Ω.
Au passage, si l'ADS807 est trop cher ou si vous n'avez pas besoin de ses 12 bits, vous pouvez utiliser un CA/N 8 bits bon marché tel que l'ADS830E. Il offre des fréquences d'échantillonnage allant jusqu'à 60 MHz et ne nécessite qu'une routine d'échantillonnage adaptée.

Échantillonnage rapide via PIO, FIFO et DMA

L'UC du RP2040 est normalement cadencée à 125 MHz. Le listage suivant montre la boucle minimale permettant d'acquérir *N* valeurs de données :

```
for (k = 0; k < N; k++) {
  sampleBuffer[k] = gpio_get_all();
  gpio_put(ADCclock,0);
  gpio_put(ADCclock,1);
}
```

D'après mes mesures, l'UC du RP2040 atteint un maximum de 15 MHz, ce qui est encore loin des 53 MHz d'un ADS807. Cependant, l'UC dispose d'une caractéristique spéciale, une unité d'entrée-sortie périphérique (PIO), qui consiste en huit automates d'états programmables. Ces automates d'états peuvent être commandés par l'horloge de l'UC et programmés à l'aide de commandes simples. Dans

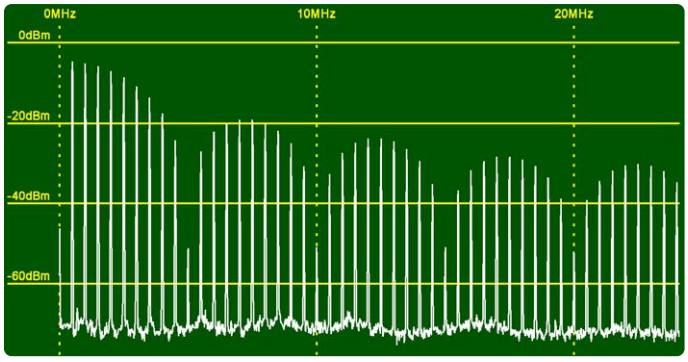


Figure 24. Spectre d'un signal carré de 500 kHz avec un rapport cyclique de 10 %.

ce qui suit, certains aspects de l'application d'échantillonnage sont mis en évidence. Les détails complets étant assez complexes, veuillez-vous référer à la documentation du RP2040 [4].
Le programme PIO ne comporte que deux commandes :

```
.wrap_target
   in pins,12     side 0b0
   nop            side 0b1
.wrap
```

Les mots-clés `.wrap_target` et `.wrap` entraînent l'exécution sans fin, l'une après l'autre, des deux instructions qui les séparent. La boucle elle-même ne surcharge pas le processus. La commande `in` entraîne le déplacement de 12 bits du CA/N vers le registre de décalage d'entrée (ISR). Comme nous le savons, l'instruction `nop` ne fait rien. Après chaque instruction, l'option `side` est utilisée. Cette option permet d'agir sur les broches GPIO parallèlement à la commande PIO proprement dite. L'horloge du CA/N est ainsi générée sur GPIO14. Comme chaque instruction nécessite un cycle, ce programme peut, en principe, traiter 125 / 2 MHz, ce qui est suffisant pour l'ADS807. Outre le programme proprement dit, la configuration de la machine à états joue un rôle important. Elle définit, par exemple, quelles broches sont affectées par le *side-set*. En outre, elle a été configurée pour que, tous les deux échantillons (lorsque 24 bits sont arrivés dans l'ISR), la valeur de l'ISR soit recopiée dans une FIFO de sortie. Les données sont rapidement écrites en mémoire via un accès direct (DMA). L'échantillonnage proprement dit se déroule donc de manière totalement autonome. L'unité centrale doit seulement récupérer les valeurs de la mémoire tampon après chaque transfert DMA.
En principe, on peut utiliser pour le CA/N intégré le même logiciel

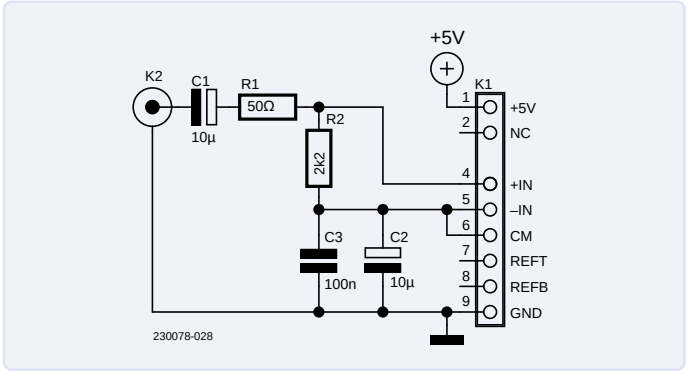


Figure 25. Frontal asymétrique.

que celui utilisé pour le CA/N ADS807. Seule la routine d'échantillonnage doit être adaptée. Pour atteindre la fréquence d'échantillonnage maximale de 53 MHz, par exemple, vous pouvez configurer une horloge UC de 106 MHz et régler le prédiviseur PIO sur 1. Le programme PIO s'exécute alors à 106 MHz. Comme chaque échantillon nécessite deux horloges, la fréquence d'échantillonnage est exactement de 53 MHz. Malheureusement, l'horloge de l'UC est alors 15 % plus lente que d'habitude. Toutefois, cela suffit pour réaliser sans problème les tâches imparties.
La figure 24 montre le spectre d'un signal carré avec un rapport cyclique de 10 % à une fréquence d'échantillonnage de 50 MHz. Les harmoniques jusqu'à 25 MHz sont clairement visibles. En raison du rapport cyclique de 10 %, toutes les 10èmes harmoniques sont absentes. Grâce au sous-échantillonnage, l'ADS807 permet d'analyser sans problème les composantes du signal jusqu'à environ 200 MHz.

Frontaux analogiques

Les entrées différentielles *+IN* et *-IN* de l'ADS807E ont un rapport signal/bruit élevé, même à des fréquences élevées. Cependant, avec une entrée asymétrique, la puce est un peu plus difficile à piloter que les CA/N. Les frontaux possibles qui peuvent être connectés à K2 dans la Figure 22 sont décrits ci-dessous. La plage dynamique de l'ADS807 est de 2 V_{SS}, ce qui donne une résolution de 2 V / 4096 = 480 µV.

Frontal 1 : liaison simple

Dans le cas le plus simple, vous ignorez l'entrée différentielle, vous réglez *-IN* sur *CM* (Common Mode) et vous envoyez simplement le signal à *+IN*. On travaille alors en couplage CA et le décalage CC de 2,5 V est fourni par *CM* (**figure 25**). Si vous souhaitez avoir une impédance d'entrée de 50 Ω, vous pouvez simplement connecter une résistance de 50 Ω en parallèle de l'entrée K2.

Frontal 2 : couplage en 50 Ω par transformateur

En utilisant un transformateur RF avec trois enroulements étroitement couplés avec le même nombre de tours, on peut se servir simplement de l'entrée différentielle. Le prototype utilisait une self de mode commun de 4,7 mH avec quatre enroulements, dont trois seulement sont utilisés (**figure 26**). La self bleue de Siemens est clairement visible sur la figure 23.

Chaque enroulement alimente les entrées *+IN* et *-IN* avec des phases opposées. Les enroulements ont chacun une résistance parallèle de 100 Ω. Le couplage 1:1 transforme ces résistances en deux résistances de 100 Ω connectées en parallèle sur le côté primaire, ce qui donne une impédance d'entrée de 50 Ω. Comme le CA/N voit deux fois la tension d'entrée, il en résulte un facteur d'amplification de 2. En même temps, l'entrée est électriquement isolée du CA/N, ce qui évite les interférences de mode commun. Le circuit est bien adapté aux mesures sur des systèmes en 50 Ω. Si vous souhaitez bobiner vous-même le transformateur L1, vous devez bobiner les trois enroulements de manière trifilaire (c'est-à-dire un bobinage avec trois fils enroulés ensemble) afin d'obtenir un bon couplage, même à des fréquences élevées. Le noyau doit avoir une perméabilité élevée, même à des fréquences allant jusqu'à 25 MHz.

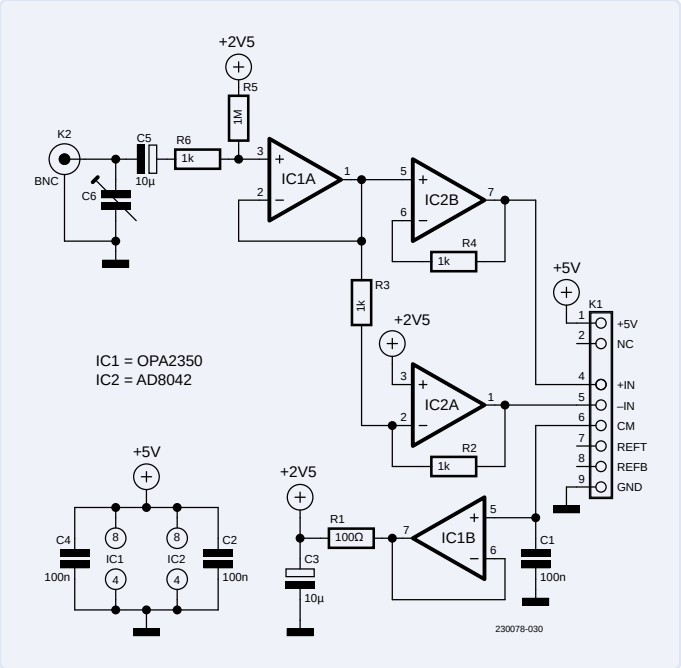


Figure 27. Préamplificateur RF avec une impédance d'entrée de 1 MΩ.

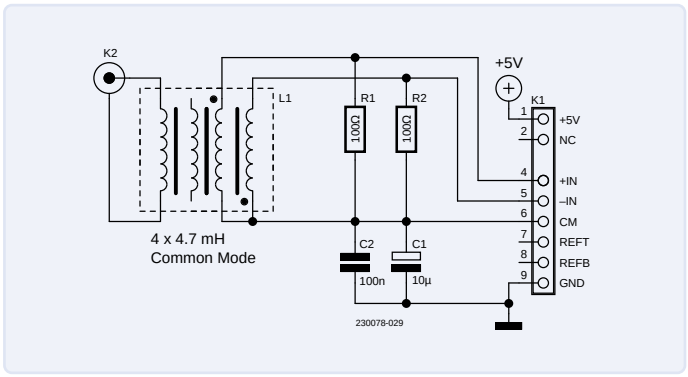


Figure 26. Frontal 50 Ω à couplage par transformateur.

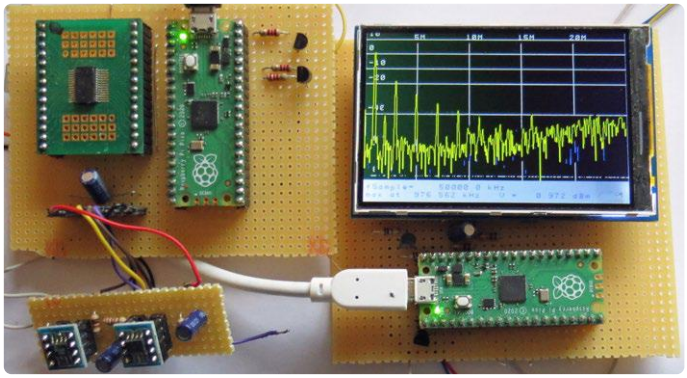


Figure 28. Analyseur de spectre autonome basé sur deux cartes Pico.

Frontal 3 : préamplificateur 25 MHz avec 1 MΩ

Pour certaines mesures, même un analyseur de spectre rapide nécessite une entrée à haute impédance. Cela permet alors d'utiliser les sondes d'oscilloscope habituelles 10:1. La **figure 27** montre le circuit d'un préamplificateur approprié.

L'amplificateur IC1A sert de convertisseur d'impédance avec une résistance d'entrée de 1 MΩ grâce à R5. Le type OPA2350 est non seulement rapide, mais présente aussi un faible courant de polarisation et un faible bruit d'entrée. IC2A et IC2B servent de pilotes bipolaires pour le CA/N. Comme l'entrée unipolaire est convertie en un signal différentiel, il en résulte un facteur d'amplification de deux. On a choisi IC2 en raison de sa grande largeur de bande, de sorte que la différence de phase entre *+IN* et *-IN* reste aussi faible que possible. R2 et R4 sont utilisés pour la stabilité de ces amplificateurs. IC1B tamponne la tension d'offset de 2,5 V. La **figure 28** montre, entre autres choses, l'installation du frontal sur une platine.

Bruit

Certaines sources de bruit compliquent la vie des concepteurs de circuits. Au niveau de R5, par exemple, une tension de bruit thermique de $U_{N,RMS} = \sqrt{(4\ k_B\ T\ B\ R)}$ se produit, où k_B est la constante de Boltzmann, T est la température absolue (en Kelvin), B est la largeur de bande (ici 25 MHz) et R est la valeur de R5 (1 MΩ). Il en résulte une tension de bruit effective de 6,3 mV.

Cela représente plusieurs fois la résolution du CA/N, et ce niveau de bruit est fortement perceptible lors des mesures à large bande et à haute impédance. Cela explique également pourquoi un système à faible impédance, par exemple 50 Ω, est plus avantageux pour la RF. Le courant d'entrée d'un OPA2350 a une densité de bruit de 4 fA / √Hz. Il en résulte une tension de bruit effective négligeable de 20 µV au niveau de R5 à une largeur de bande de 25 MHz. Pour le bruit de la tension d'entrée, on spécifie une densité de 7 V / √Hz. Il en résulte une tension de bruit de 30 µV à 25 MHz, ce qui est également négligeable. Cependant, il apparaît clairement que les paramètres de bruit de l'amplificateur influencent la conception. Par exemple, l'AD8042 a un bruit en courant de 500 fA / √Hz et ne convient donc pas comme convertisseur d'impédance à l'entrée.

Affichage graphique

Comme déjà mentionné, il n'est pas possible de connecter le CA/N externe et l'écran LCD en même temps en raison du manque de broches GPIO. Pour un appareil autonome, cette lacune est compensée par l'utilisation de deux cartes Pico séparées : l'une pour l'acquisition des données à partir du CA/N externe, et l'autre pour l'écran LCD. Ce dernier sert de terminal graphique capable d'exécuter indépendamment diverses fonctions de base telles que le tracé de lignes et de rectangles, ainsi que l'affichage de texte. Les fonctions de l'analyseur de spectre sont implémentées sur la carte Pico avec l'ADS807. Pour la sortie graphique, cette partie envoie les commandes correspondantes au terminal graphique via l'interface série. Au prix de quelques euros pour une carte Pico, cette conception à double carte est économiquement justifiable.

La figure 28 montre à quoi pourrait ressembler cette solution avec deux cartes Pico. L'écran affiche le spectre d'un signal carré de 1 MHz à une fréquence d'échantillonnage de 50 MHz. Les harmoniques diminuent rapidement avec l'accroissement de l'ordre.

VF : Denis Lafourcade — 230078-01

Questions ou commentaires ?

Envoyez un courriel à l'auteur (ossmann@fh-aachen.de), ou contactez Elektor à (redaction@elektor.fr).

À propos de l'auteur

Martin Ossmann a commencé à lire Elektor à l'âge de 12 ans - et à bricoler, bien sûr. Après avoir étudié l'électrotechnique et travaillé pendant plusieurs années comme ingénieur de développement, il a été professeur au département d'électrotechnique et de technologie de l'information de l'université des sciences appliquées d'Aix-la-Chapelle (FH Aachen University of Applied Sciences). Il n'est pas seulement l'auteur de publications scientifiques, mais publie aussi régulièrement des circuits et des projets de logiciels avec beaucoup de savoir-faire technique dans Elektor depuis plus de trois décennies.

Produits

- **Joy-IT JDS6600 Générateur de signaux et compteur de fréquences**
  www.elektor.fr/18714
- **OWON HDS242 Oscilloscope à 2 canaux (40 MHz) + Multimètre**
  www.elektor.fr/20415
- **Raspberry Pi Pico RP2040**
  www.elektor.fr/19562

LIENS

[1] Identité de Parseval : https://fr.wikipedia.org/wiki/%C3%89galit%C3%A9_de_Parseval
[2] Page web de cet article : https://www.elektormagazine.fr/230078-04
[3] LTspice : https://tinyurl.com/3zpwzk4y
[4] Fiche technique du RP2040 : https://tinyurl.com/2sf4uvfm

Rejoignez la communauté Elektor



Devenez membre **maintenant !**

elektor GOLD membership



- accès à l'archive numérique depuis 1978 !
- 8x magazine imprimé Elektor
- 8x magazine numérique (PDF)
- 10 % de remise dans l'e-choppe et des offres exclusives pour les membres
- accès à plus de 5000 fichiers Gerber



Également disponible

abonnement elektor GREEN membership sans papier !

- accès à l'archive numérique d'Elektor
- 10 % de remise dans l'e-choppe
- 8x magazine Elektor (PDF)
- accès à plus de 5000 fichiers Gerber

www.elektormagazine.fr/membres

elektor design › share › earn



régulateur de tension linéaire ±40 V

une alimentation de qualité pour l'amplificateur de puissance Fortissimo-100… et d'autres !

Ton Giesberts (Elektor Labs)

Pour ceux qui refusent toute forme d'alimentation à découpage (SMPS) pour l'amplificateur de puissance haut de gamme Fortissimo-100, voici un régulateur de tension linéaire et symétrique de plus de 500 VA, caractérisé par une faible tension de chute, un courant de sortie élevé et une excellente stabilité, le tout obtenu à partir de composants discrets et disponible sous forme d'un kit !

Sachant que presque tous les amplificateurs de puissance audio de haute performance bénéficient d'une tension d'alimentation stabilisée, cette alimentation linéaire est spécifiquement conçue pour une tension de sortie symétrique de ±40 V et des courants de crête de 13 A (15 A de crête réalisable). À titre d'exemple, le courant moyen tiré par un ampli Fortissmo-100 pilotant une charge de 3 Ω est d'environ 4 A par régulateur.

Considérations relatives à la conception

L'amplificateur de puissance audio haut de gamme Elektor Fortissimo-100 [1] fonctionne mieux avec une alimentation régulée de ±40 V, ce qui exclut une alimentation simple composée d'un transformateur, d'un redresseur (à pont) et d'un ensemble de condensateurs à réservoir épais. Une alimentation à découpage peut ne pas convenir non plus, mais c'est plus une question de goût personnel, car le SMPS800RE fait du bon travail. Cependant, il peut y avoir des raisons impérieuses de préférer un régulateur linéaire construit uniquement à partir de composants traversants, comme l'amplificateur lui-même.

Pour que le régulateur de tension puisse fonctionner correctement, la tension d'entrée du circuit doit dépasser la tension de sortie d'au moins 3 V, voire plus en cas de fluctuations de la tension du réseau. Par rapport à la plupart des SMPS (avec une large plage de tension d'entrée AC), un régulateur linéaire est moins efficace et un grand transformateur de puissance est nécessaire avec une puissance nominale plus élevée que sans le régulateur linéaire.

Aujourd'hui, la plupart des transformateurs de puissance (secteur) disponibles sur le marché sont marqués par des tensions secondaires normalisées. Pour créer directement ±40 VDC,

un transformateur de 2× 30 V est le choix le plus pratique. La tension continue à vide qui en résulte est généralement d'environ 42 VCC, ce qui dépend en grande partie de la régulation interne du transformateur et de la chute de tension dans les diodes de redressement. En pratique, la tension de sortie à vide d'un transformateur de puissance est toujours supérieure de quelques pourcents à la tension de charge. La tension secondaire standard immédiatement supérieure est de 35 V, ce qui donne environ 49 V à 50 VCC ou plus avec une faible charge. Près de 52 V ont été mesurés dans un dispositif d'essai Labs. Avec une charge de 8 Ω sur l'amplificateur de puissance, le régulateur ne nécessite qu'une petite capacité de lissage. L'avantage de la tension d'ondulation plus importante est une perte de puissance un peu plus faible dans le(s) régulateur(s) d'alimentation. Mais, à des impédances plus faibles, l'ondulation ne doit pas dépasser la tension de chute (43 V à 10 A). Lors d'un test en laboratoire, un transformateur toroïdal de 2× 35 V, 300 VA avec 20 000 µF de capacité de lissage s'est avéré assez robuste pour alimenter le régulateur. La puissance sinusoïdale maximale (proche de l'écrêtage) à 20 Hz et 0,1% THD+N dans une charge de 3 Ω ne provoquait que 1,8 $V_{crête}$ de chute à la sortie de l'alimentation. Attention, la puissance de sortie continue est alors de 227 W dans la charge de 3 Ω et le transformateur de 300 VA est légèrement surchargé. Ceci, cependant, n'était pas suffisant pour déclencher la protection du Fortissimo-100.

Théorie du fonctionnement

La base de tout régulateur de tension consiste à mesurer la tension de sortie, à la comparer à un niveau de référence et à contrôler l'étage de sortie en conséquence pour contrer tout changement. Bien que le présent circuit régulateur suive ce concept, sa particularité est sa tension de référence secondaire beaucoup plus élevée, qui, à un peu plus de 33 V, est relativement proche de la tension de sortie cible de 40 V. Plus la tension de référence est élevée, 33,6 V ici, plus il reste de gain à un circuit (simple) pour augmenter en même temps la réjection des ondulations de la tension d'entrée et la régulation de la tension de sortie.

En termes simples, le circuit se compose d'une tension de référence, d'un amplificateur différentiel et d'un tampon de sortie. En outre, une protection de la zone de fonctionnement sûre (SOA, *Safe Operating Area*) est ajoutée aux deux régulateurs. Examinons la **figure 1** pour explorer le fonctionnement du régulateur positif.

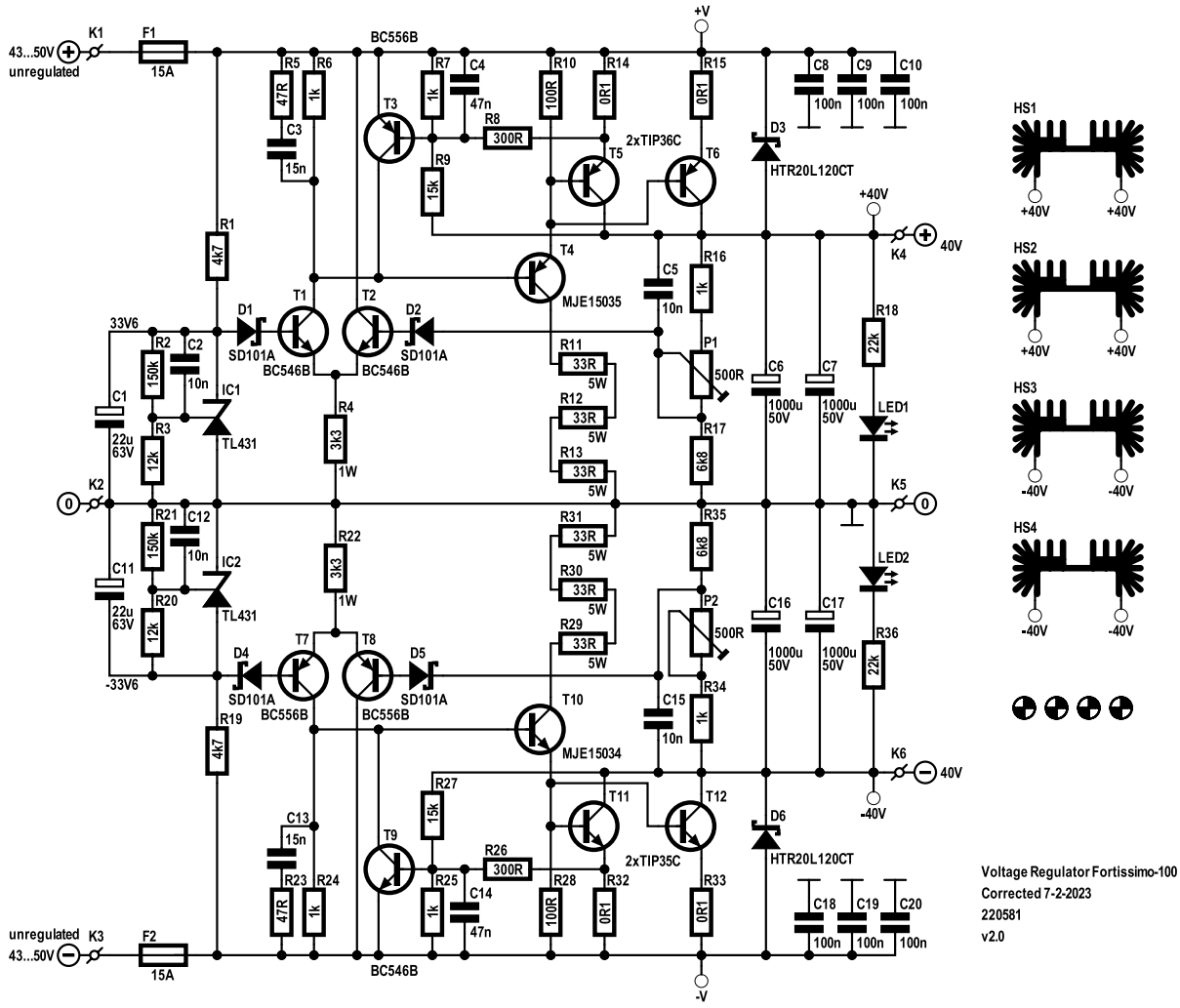


Figure 1. Schéma du régulateur de tension linéaire ±40 V, principalement, mais pas exclusivement, destiné à l'amplificateur de puissance audio Elektor Fortissimo-100.

Spécifications

Plage de tension d'entrée	52 VDC (utilisation à faible consommation) à 43 VDC
Plage de tension de sortie d'environ	38,9 VDC à 41,4 VDC (théoriquement) 38,6 VDC à 41,1 VDC (mesurée)
Tension de chute à 6 A	42 V
Tension de chute à 9,5 A	43 V
Tension de chute à 13,5 A	44 V
Courant max.	15 A crête (demi-sinusoïdale) 4,8 A en moyenne
Protection SOA	15 A à 45 VDC à l'entrée
Rejet de l'ondulation	>60 dB (à une charge de 5 ADC)
Courant d'entrée à vide	27 mA (@ 52 VDC entrée)
Assemblage	à partir du kit d'Elektor ; ajouter le transformateur de puissance, le redresseur (à pont) et les condensateurs.

Tension de référence

La tension de référence n'est pas créée par une diode Zener, car les diodes Zener standard ont généralement des coefficients de température considérables. Les versions spéciales compensées en température sont difficiles à trouver de nos jours, en particulier les versions 33 V. Au lieu d'une diode Zener, une référence de tension shunt de précision réglable de type TL431 est utilisée avec une tension de travail maximale de 36 V. Sa tension de référence interne (c'est-à-dire la tension de référence primaire du régulateur de 40 V) est typiquement de 2,495 V. Le courant de cathode à travers le TL431 est réglé par la résistance R1. Si la tension d'entrée est comprise entre 43 V et 50 V, le courant est réglé entre 1,9 mA et 3,4 mA, ce qui s'est avéré suffisant pour créer une tension de référence stable de 33,9 V. Cette tension est définie par les résistances R2 et R3 :

$$V_{KA} = 2{,}495 \times (1 + R2 / R3) + I_{REF} \times R2$$

Le courant d'ajustement du TL431, I_{REF}, est typiquement de 1,8 µA, donc la tension de référence est théoriquement de 33,95 V. Cependant, cette valeur est spécifiée pour un courant cathodique de 10 mA, alors que dans le prototype ce courant est plus faible, de même que la tension : 33,55 V a été mesuré. Le TL431 est découplé par C1 tandis que C2 améliore la stabilité globale.

Amplificateur différentiel

L'amplificateur différentiel est minimaliste, composé de T1 et T2 avec R4 comme source de courant. La tension à la base de T1 est relativement constante. Il en va de même pour la tension aux bornes de R3, même si la tension varie légèrement, en fonction de la température, aux bornes de la jonction base-émetteur de T1 et aux bornes de D1. Les diodes Schottky D1 et D2 limitent une inversion (tout juste concevable) des tensions base-émetteur de T1 et T2.

Pour réduire l'influence de la chute de tension à travers chaque diode, tout en affectant peu la tension de décalage d'entrée de la paire différentielle en raison des changements de température, la paire est positionnée l'une à côté de l'autre sur le circuit imprimé, de sorte que les deux jonctions de diodes sont à la même température. Quelques millivolts, ou même des dizaines de millivolts de décalage également causés par des différences entre T1 et T2 n'ont pas d'effet réel sur la tension de sortie beaucoup plus élevée de 40 V. Même un changement de décalage de 30 mV signifie moins de 1% de variation dans la tension de sortie, ce qui est sans conséquence pour le fonctionnement de l'amplificateur de puissance.

La tension aux bornes de la résistance de collecteur R6 est utilisée pour piloter l'étage de sortie. R5 et C3, ainsi que C4 et C5, forment la compensation de fréquence pour maintenir le régulateur stable, même lorsque la protection SOA T3/R7/R8/R9 est active. Le diviseur de potentiel R16-P1-R17 mesure la tension de sortie et constitue le retour négatif pour l'amplificateur différentiel. Pour compenser toutes les tolérances, la plage de tension de sortie de P1 est approximativement de 38,6 V à 41,1 V. Avec le curseur du potentiomètre à mi-course, la tension de sortie sera assez proche de 40 V.

Étage de sortie

Bien qu'il existe des transistors capables de gérer le courant de sortie maximum requis lorsqu'une tension d'entrée constante de 50 V est appliquée au régulateur, deux transistors, T5/T6, sont choisis pour :

- limiter la puissance dissipée par transistor à des niveaux sûrs ;
- augmenter la plage de surcharge ;
- obtenir une tension de chute plus faible et une plus grande zone de fonctionnement sûre.

La mise en œuvre de ces critères réduit le risque d'endommagement de l'étage de sortie en cas de surcharge ou même de court-circuit. Les transistors de puissance PNP les plus grands sont de type TIP36C (NPN TIP35C dans le régulateur négatif) et sont facilement disponibles auprès de plusieurs fabricants. Les transistors PNP sont utilisés dans le régulateur positif pour maintenir la chute de tension minimale de l'étage de sortie aussi basse que possible, les courants de base s'écoulant vers la masse.

La tension de chute est la somme de la tension de saturation des transistors et de la chute de tension dans les résistances d'émetteur. Une valeur plus faible pour les résistances d'émetteur réduirait un peu la tension de chute, mais les courants traversant les deux transistors peuvent dévier trop fortement. Lorsque les courants de collecteur sont élevés, le gain des transistors est très faible et un transistor supplémentaire (T4) est nécessaire pour tamponner la sortie de l'étage différentiel. Pour éviter que la tension de saturation de T4 n'augmente la tension de chute de l'étage de sortie, son collecteur est relié à la masse par une connexion de résistances en série. Cela limite la dissipation de puissance de T4 ainsi que son besoin en dissipateur thermique. Cependant, il y a un problème : si, pour une raison quelconque, la tension d'entrée tombe en dessous de la tension de chute, T4 conduira en permanence et la dissipation dans sa résistance de collecteur sera assez élevée (16 W) pour une résistance totale de 100 Ω et une tension d'entrée appliquée de 40 V. Cela ne devrait cependant jamais se produire, c'est pourquoi trois résistances de 5 W sont utilisées pour éviter un burnout de cette résistance de collecteur.

Un avantage supplémentaire de cette résistance de collecteur est qu'elle limite les courants de base de T5 et T6, et agit donc comme une simple limite de courant. La véritable protection, cependant, est formée par T3. La chute de tension à travers la résistance d'émetteur de T5 proportionnelle au courant de sortie est mesuré par le diviseur de tension R7/R8, ce qui alimente la base de T3. Lorsque, par exemple, le courant à travers R14 est d'environ 7 A, le courant de sortie total est de 14 A. Le courant de sortie le plus élevé attendu est un peu plus de 12 $A_{crête}$ avec une charge de 3 Ω sur la sortie de l'amplificateur.



*Figure 2. Quelques clichés du **manuel d'assemblage** rédigé pour le projet de régulateur de tension linéaire ±40 V. Ce manuel complète cet article et peut être téléchargé gratuitement sur le site [2].*

T3 commencera à conduire, et, à cause de R9, même plus tôt, en fonction de la tension aux bornes de T5. Le niveau exact auquel T3 est polarisé dépend de la température et sera plus faible à mesure que la température augmente, une protection supplémentaire, et, avec de la musique, ce ne sera pas un problème.

D3 protège l'étage de sortie en cas de coupure soudaine de la tension d'entrée ou de court-circuit. T5 et T6 sont découplés avec une paire de condensateurs de 1 000 µF, à faible ESR. La LED1 indique la présence de la tension de sortie de +40 V.

Bien que sur les photos il semble que D6 soit monté à l'envers sur le circuit imprimé, en fait, D6 et D3 peuvent être montés dans les deux sens et fonctionner correctement. Cela est possible, car la diode HTR20L120CT dans son boîtier TO220 à trois pattes contient deux diodes avec une cathode commune connectée à la patte du milieu.

L'entrée du régulateur est protégée par un fusible de 15 A. Le courant RMS (moyenne quadratique) maximum doit être pris en compte, et au courant sinusoïdal demi-onde maximum, la valeur RMS est $I_{crête}/2$, c'est-à-dire 6,5 A. Cependant, à des fréquences très basses comme 16,4 Hz (si vous aimez la musique d'orgue), le courant de crête peut durer plusieurs millisecondes. Pour s'assurer que le fusible ne saute pas dans de telles conditions, on utilise ici un type 15 A qui, en prime, réduit également la chute de tension. Si beaucoup plus de puissance est absorbée par l'amplificateur et/ou le régulateur, le fusible primaire lié au transformateur de puissance saute. Le fusible de 15 A sautera de façon fiable s'il y a un court-circuit soudain en aval.

Kit, manuel de construction et nomenclature

La boutique Elektor propose un kit complet pour le projet de régulateur de tension linéaire [2], contenant le circuit imprimé (PCB) et toutes les pièces listées dans la liste des composants. Ce superbe kit évite, nous l'espérons, les efforts des lecteurs pour acheter des pièces (électroniques et mécaniques) et faire fabriquer des PCB sur commande.

Le kit est accompagné d'un *manuel d'assemblage* de douze pages donnant des instructions pas à pas pour assembler le projet et obtenir un résultat parfait. Le manuel est riche en dessins et en photographies, dont quelques-uns sont présentés dans la **figure 2**. Il contient également de nombreux conseils et détails sur la précision des soudures, le positionnement des composants, la manipulation des outils et les travaux mécaniques simples nécessaires pour réussir la construction du projet.

Comme le régulateur proposé n'est pas une alimentation complète sans les circuits habituels d'un transformateur de puissance,

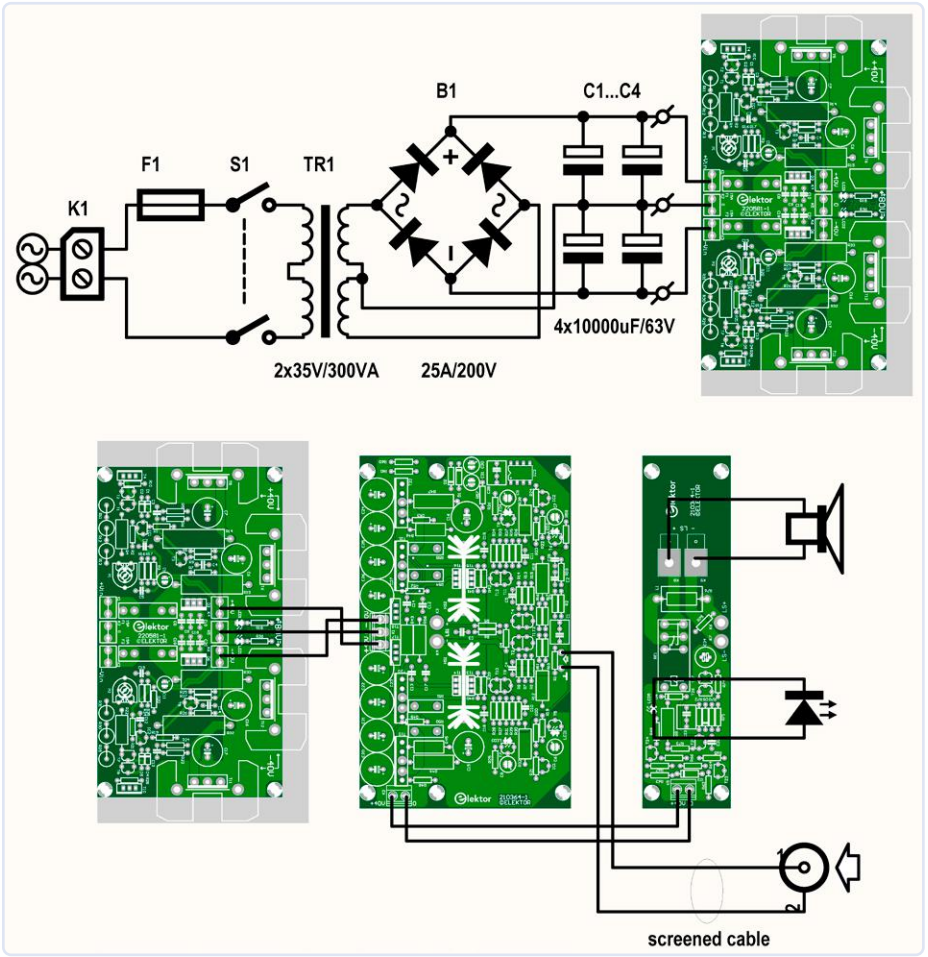
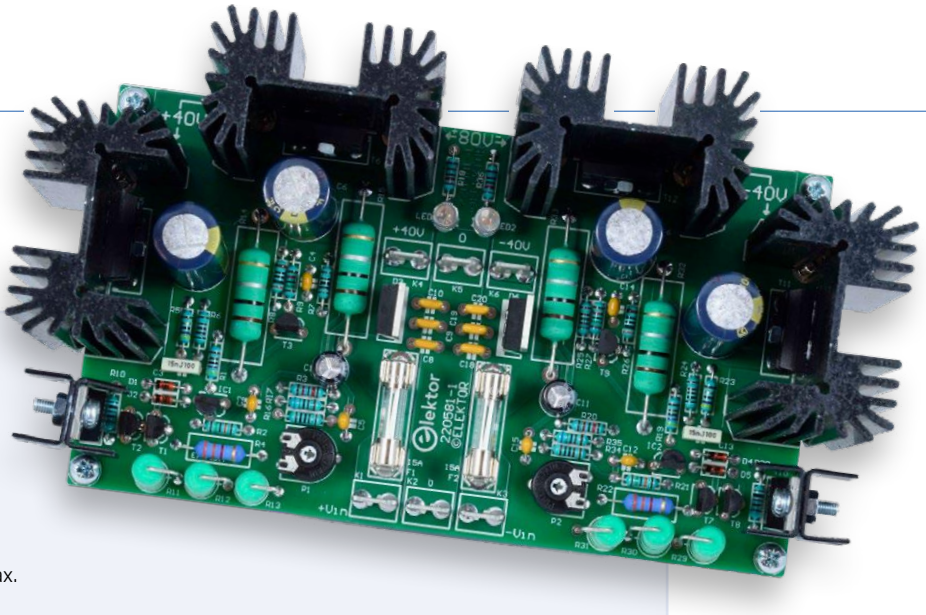


Figure 3. Proposition de schéma pour le circuit d'alimentation non régulé (en haut) et schéma de câblage du combo régulateur de tension linéaire ±40 V / Fortissimo-100 (en bas).



Liste des composants

(Liste du contenu du kit Elektor)

Résistances

R1, R19 = 4,7 kΩ, 1%, 0,6 W
R2, R21 = 150 kΩ, 1%, 0,6 W
R3, R20 = 12 kΩ, 1%, 0,6 W
R4, R22 = 3,3 kΩ, 5%, 1 W, 5×12 mm max.
R5, R23 = 47 Ω, 1%, 0,6 W
R6, R7, R16, R24, R25, R34 = 1 kΩ, 1%, 0,6 W
R8, R26 = 300 Ω, 1%, 0,6 W
R9, R27 = 15 kΩ, 1%, 0,6 W
R10, R28 = 100 Ω, 1%, 0,6 W
R11, R12, R13, R29, R30, R31 = 33 Ω, 5%, 5 W, diam. 6,4 mm max. (axial, monté verticalement)
R14, R15, R32, R33 = 0,1 Ω, 10%, 5 W (7,8×25 mm max.)
R17, R35 = 6,8 kΩ, 1%, 0,6 W
R18, R36 = 22 kΩ, 1%, 0,6 W
P1, P2 = 500 Ω, 0,15 W, trimmer, réglage supérieur (Piher PT10LV10-501A2020-S)

Condensateurs

C1, C11 = 22 µF, 20%, 63 V, pas 2,5 mm, diam. 6,3 mm max.
C2, C5, C12, C15 = 10 nF, 10%, 100 V, céramique X7R , pas 5 mm
C3, C13 = 15 nF, 5%, 100 V, PET, pas de 5 mm
C4, C14 = 47 nF, 10%, 50 V, céramique X7R, pas de 5 mm
C6, C7, C16, C17 = 1000 µF, 20%, 50 V, pas de 5 mm, D 12.5 mm, 5000 h @ 105 °C (EEUFC1H102L, Panasonic)
C8, C9, C10, C18, C19, C20 = 100 nF, 10%, 100 V, pas de 5 mm, céramique X7R

Semi-conducteurs

D1, D2, D4, D5 = SD101A SB00018/D8, DO-35
D3, D6 = HTR20L120CT, TO-220
LED1, LED2 = LED, verte, 5 mm (T-1¾)
T1, T2, T9 = BC546B, TO-92
T3, T7, T8 = BC556B, TO-92
T4 = MJE15035, TO-220
T5, T6 = TIP36C, TO-247
T10 = MJE15034, TO-220
T11, T12 = TIP35C, TO-247
IC1, IC2 = TL431BCLPG, TO-92

Divers

K1, K2, K3, K4, K5, K6 = languette Faston PCB, deux broches, trou diam. 1.4 mm, 6,35×0,83 mm
F1, F2 = Clips de fusible, 20 A, Littelfuse 01000020Z, pour fusible de 5×20 mm
F1, F2 = Fusible, cartouche, à action rapide, 15 A, 5×20 mm
HS1, HS2, HS2, HS4 = dissipateur thermique MC33271 (pour T5/T6/T11/T12), 2,7 °C/W
4× dissipateur thermique type FK231SA220, 24 K/W (pour T4/T10, deux chacun)
10× rondelle M3, lisse, acier
6× vis M3, 10 mm, tête cylindrique
6× écrou M3

d'un redresseur et de condensateurs de lissage ajoutés, un schéma suggéré, accordé à l'ampli Fortissimo-100, est donné dans la **figure 3**. Les pièces pour cette section ne sont pas incluses dans le kit du régulateur de tension linéaire ±40 V et doivent être achetées localement.

Mesures de sécurité

Bien que la construction du projet et son utilisation pratique soient détaillées dans le manuel de construction, nous nous sentons obligés d'imprimer l'avis de sécurité suivant dans cet article également :
Les grands dissipateurs sont connectés à la tension de sortie de ±40 V, et **non** à GND. Toujours retirer la tension d'entrée avant de toucher ou de travailler sur le régulateur !

Résultats obtenus

Chez Elektor Labs, une configuration de test a été construite pour vérifier le fonctionnement de l'amplificateur Fortissimo-100 en combinaison avec le régulateur de tension linéaire ±40 V décrit ici. Les deux unités ont été construites à partir de leurs kits Elektor respectifs. Les ingrédients clés suivants ont été utilisés dans la section d'alimentation non régulée :

- 1× Transformateur de puissance TX-146-300-235 (300 VA, 2× 35 VAC secondaire).
- 2× Condensateurs électrolytiques de 10 000 µF par rail de tension d'alimentation (c'est-à-dire 20 mF sur chaque rail).
- 1× Pont redresseur de type SB352SBPC, 35 A/200 V (25 A/100 V suffit).

Aux faibles niveaux de sortie du Fortissimo-100, le spectre de fréquence montre que de très petites améliorations peuvent être obtenues par rapport à l'alimentation à découpage SMPS800RE (**figure 4**). Le graphique montre le spectre de fréquence à 1 W dans 8 Ω. Les effets de commutation du SMPS800RE ont disparu, mais le reste du spectre est essentiellement le même. La performance globale du combo est impressionnante, avec une distorsion harmonique plus bruit aussi faible que :

- 0,0007% (1 kHz, 1 W, 8 Ω, B = 22 kHz)
- 0,0013% (1 kHz, 1 W, 8 Ω, B = 80 kHz)

Le régulateur de tension linéaire ±40 V décrit ici et disponible en kit chez Elektor est une bonne alternative aux meilleures, mais abordables, alimentations à découpage sur le marché aujourd'hui, et devrait satisfaire ceux d'entre vous qui s'opposent, même légèrement, au concept ou à la performance de « ces @#!%^ de commutateurs ». N'hésitez pas à rejoindre les discussions techniques sur le régulateur de tension linéaire ±40 V sur la page Elektor Labs créée pour le projet [3].

VF : Maxime Valens — 220581-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

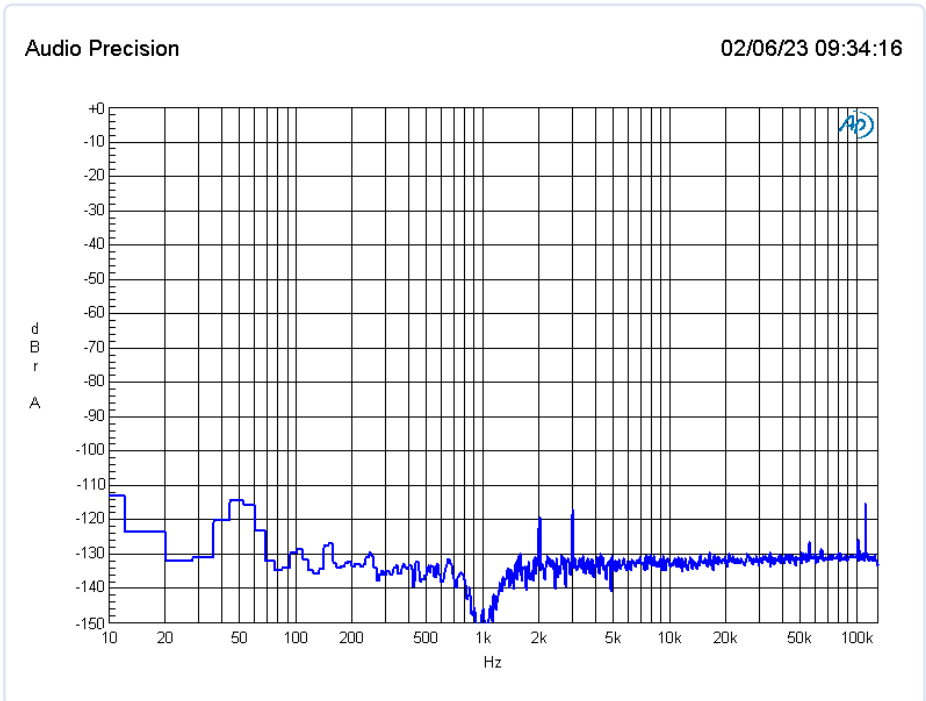


Figure 4. Voici les résultats ! Spectre de sortie du Fortissimo-100 fournissant 1 W dans 8 Ω et alimenté par le régulateur de tension linéaire ±40 V (carte n° 220581-1).

Produits connexes

- **Kit de régulateur de tension linéaire ±40 V d'Elektor**
  https://www.elektor.fr/elektor-40-v-linear-voltage-regulator-kit



LIENS

[1] Amplificateur haut de gamme Fortissimo-100 : https://www.elektormagazine.fr/magazine/elektor-283/61180
[2] Elektor Kit de régulateur de tension linéaire ±40 V : https://www.elektor.fr/elektor-40-v-linear-voltage-regulator-kit
[3] Le projet de régulateur de tension linéaire ±40 sur Elektor : https://www.elektormagazine.fr/labs/linear-voltage-regulator-for-fortissimo-100

la communication sans fil des microcontrôleurs **devient flexible**

l'EEPROM ouvre des perspectives de mise en réseau pour les microcontrôleurs sans fil

Gamal Labib (Egypte)

Lorsque vous connectez un microcontrôleur à un réseau wifi à l'aide de l'ESP8266, vous souhaiteriez une approche plus souple que des identifiants WLAN fixes codés en dur. Dans cet article, je présenterai une solution avec une liste de réseaux AP préférés à choisir de manière interactive. En outre, nous exploiterons la fonction *Wi-Fi Protected Setup* (WPS) prise en charge par de nombreux points d'accès et routeurs.

L'ESP8266 est un module microcontrôleur (MCU) populaire qui prend en charge les communications sans fil et sert à diverses applications dans l'Internet des objets (IdO). Les cartes basées sur ce module ont généralement des identifiants de point d'accès (AP : *Access Point*) codés en dur, tels que le nom du réseau (SSID : *Service Set IDentifier*) et le mot de passe (ou clé), dans leurs croquis Arduino. Dans certains cas, les développeurs spécifient également des paramètres IP (Internet Protocol) fixes. Cependant, ces paramètres codés en dur peuvent poser des problèmes lorsque la topologie du réseau local sans fil (WLAN) change. C'est sérieusement ennuyeux de retourner à l'EDI Arduino ou à un environnement de développement similaire juste pour mettre à jour les programmes des cartes MCU déjà déployées, car les cartes doivent être reprises ou remplacées.

Dans cet article, je présenterai une solution qui offre une certaine souplesse pour connecter une carte microcontrôleur à un réseau local sans fil. Au lieu de coder en dur un seul jeu de paramètres d'identification WLAN dans les programmes du projet, pourquoi ne pas disposer d'un ensemble de paramètres réseaux AP préférés pouvant être choisis de manière interactive ? En établissant un dialogue interactif entre l'utilisateur et la carte microcontrôleur par le biais d'un écran tactile, d'une page web ou d'un moyen similaire, il est alors possible à volonté de spécifier de nouveaux paramètres WLAN pour la carte microcontrôleur. En outre, nous pouvons tirer parti de la fonction Wi-Fi Protected Setup (WPS = configuration wifi protégée) prise en charge par de nombreux points d'accès et routeurs. Le WPS permet à une carte microcontrôleur de rejoindre à volonté le réseau préféré de l'utilisateur. Cependant, une question se pose : devons-nous passer par les étapes de connexion du dialogue interactif ou du WPS à chaque redémarrage de la carte microcontrôleur ? La réponse est non, aussi longtemps que nous conservons les nouveaux paramètres WLAN spécifiés dans une mémoire non volatile, accessible par la carte microcontrôleur. Heureusement, l'ESP8266 dispose d'une mémoire morte programmable et effaçable électriquement (EEPROM) interne qui peut être éditée par quelques lignes de code pour mémoriser et récupérer les paramètres demandés. J'ai utilisé cette fonction pour stocker jusqu'à 10 identifiants WLAN définis par l'une ou l'autre des méthodes susmentionnées. Si l'utilisateur le souhaite, cette approche offre non seulement une flexibilité lors de la mise en réseau, mais permet également à la carte microcontrôleur de se connecter au réseau local sans fil possédant la meilleure couverture parmi les 10 variantes enregistrées. Toutefois, il est important de noter qu'il n'est pas recommandé d'écrire continuellement dans l'EEPROM interne, car la durée de vie d'une EEPROM dépend du nombre de cycles d'écriture qu'elle opère. Pour contourner cette restriction et préserver le rôle désigné du microcontrôleur, j'ai inclus une puce EEPROM externe dans la préparation du projet permettant à l'utilisateur d'obtenir le même résultat avec cette EEPROM externe. Cependant, pour notre exemple de configuration WLAN, nous n'allons pas écrire fréquemment, j'ai donc opté pour l'utilisation de l'EEPROM interne.

Matériel

La liste des composants pour ce projet est assez brève. J'ai utilisé la WeMos D1 Mini, qui est une carte basée sur l'ESP8266 connue pour sa simplicité d'utilisation dans l'EDI Arduino, ainsi que pour son faible encombrement. Un module d'affichage graphique OLED de 0,9 pouce ayant une résolution de 128×64 pixels est utilisé pour afficher les messages d'information et de débogage. Il est connecté au bus I²C du microcontrôleur, ainsi qu'à une puce EEPROM externe de 8 Ko. Pour sélectionner l'un des huit modes de fonctionnement du microcontrôleur (voir **tableau 1**), j'ai utilisé une combinaison de trois commutateurs DIP (*Dual Inline Package*), qui sont connectés à trois des GPIO (*General Purpose Input/Output*) numériques du WeMos (par exemple, D5, D6 et D7). Comme les broches des

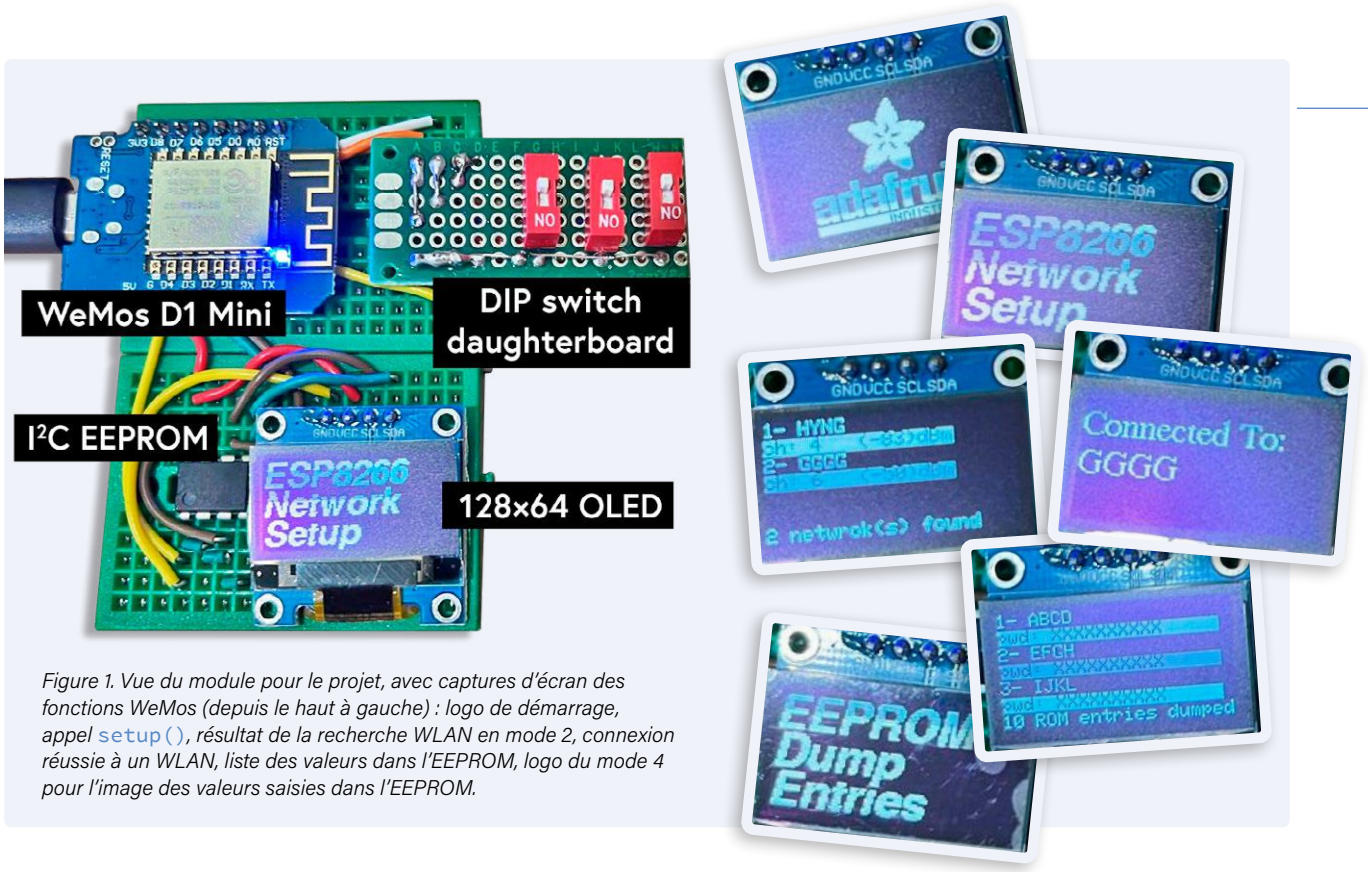


Figure 1. Vue du module pour le projet, avec captures d'écran des fonctions WeMos (depuis le haut à gauche) : logo de démarrage, appel `setup()`, résultat de la recherche WLAN en mode 2, connexion réussie à un WLAN, liste des valeurs dans l'EEPROM, logo du mode 4 pour l'image des valeurs saisies dans l'EEPROM.

Tableau 1. Configuration des commutateurs DIP des modes WeMos appropriés.

Mode	DIP 1	DIP 2	DIP 3	Action du WeMos
1	0	0	0	WLAN sélectionné par défaut
2	0	0	1	Analyse des WLAN disponibles
3	0	1	0	Effacement de l'EEPROM interne
4	0	1	1	Image de l'EEPROM interne
5	1	0	0	Initialisation de l'EEPROM interne
6	1	0	1	Vérification de l'EEPROM externe
7	1	1	0	WPS
8	1	1	1	Setup interactif du WLAN

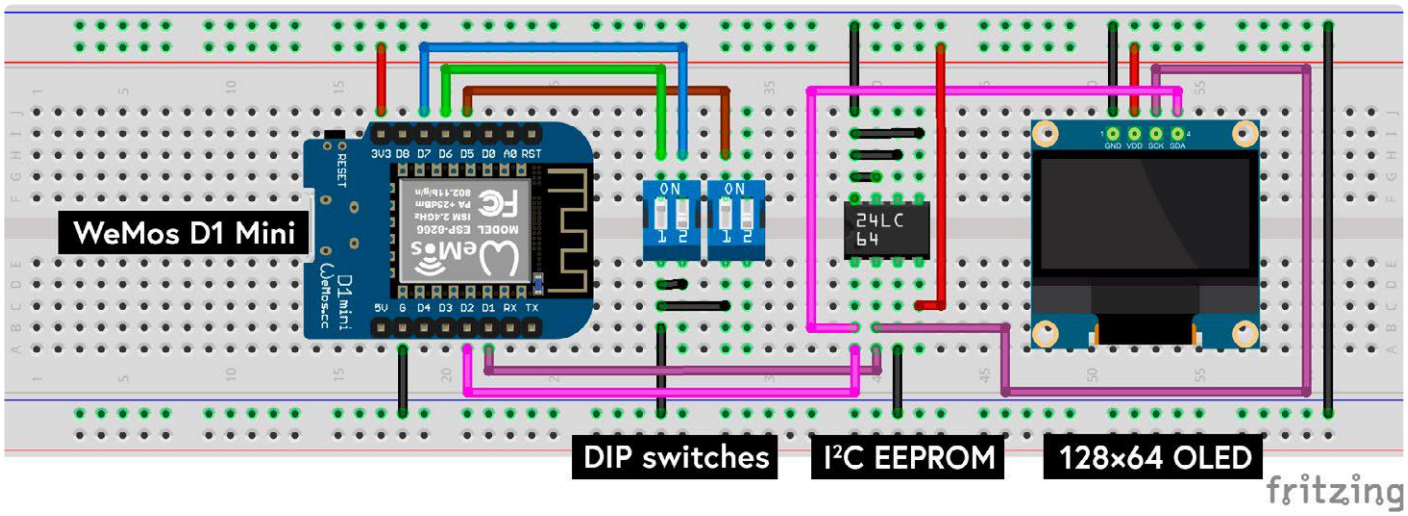


Figure 2. Câblage du projet à l'aide de Fritzing et les outils pour platine d'expérimentation intégrant un quatrième commutateur DIP supplémentaire pour des options de réseau étendues.

interrupteurs DIP sont minuscules, ils n'entrent pas dans la platine d'expérimentation ; j'ai donc construit une carte fille pour les accueillir. La **figure 1** montre l'implémentation du projet sur la platine d'expérimentation ainsi que les captures d'écran du WeMos en action. Le schéma de câblage des composants est présenté à la **figure 2**.

Logiciel

Le projet met l'accent sur les pratiques éprouvées de développement logiciel, telles qu'un code bien structuré et modulaire, permettant de le réutiliser. Dans ce but, j'ai divisé le programme de 800 lignes en six fichiers distincts, chacun gérant un élément matériel ou une fonctionnalité spécifique (voir le **tableau 2**). Cette approche rend le code plus facile à gérer et aide l'utilisateur à comprendre le code assez complexe. Le fichier du programme principal se concentre sur les modes de mise en réseau du WeMos dans la fonction `setup()`, tout en laissant la fonction `loop()` libre pour le code d'application du WeMos. Pendant la phase de développement, j'ai utilisé `loop()` pour gérer le clignotement de la LED présente sur le module et m'assurer que le WeMos fonctionne correctement.

Utilisation de l'EEPROM pour le réseau WLAN

Le module ESP8266 dispose de 4 Mo de mémoire flash, dont 4 Ko sont réservés à l'émulation d'une EEPROM habituellement disponible sur un module Arduino. Par défaut, notre croquis mémorise les dernières informations d'identification de l'AP dans l'EEPROM interne. Avec une mémoire interne (ou externe) non volatile, notre programme peut rappeler les informations d'identification de l'AP à chaque fois qu'il démarre à la suite de la mise sous tension. L'EEPROM est également accessible au moyen d'appels de fonction de la bibliothèque EEPROM de l'Arduino, et je reprendrai cette approche pour sauvegarder les données du réseau qui doivent être non volatiles, lorsque le module est hors tension. Le concept principal consiste à maintenir la structure proposée dans le **tableau 3** pour conserver les informations d'identification pour autant de AP WLAN que l'EEPROM interne ou externe peut contenir. Chaque donnée successive dans la structure contient le SSID de l'AP et son mot de passe dans 32 et respectivement 64 octets de mémoire. Cela fait 96 octets au total par point d'accès. Chaque fois que l'utilisateur ajoute un nouveau WLAN à la liste des réseaux préférés, le code ajoute à cette structure les informations d'identification du SSID et du mot de passe du nouveau réseau. L'utilisateur doit effacer l'EEPROM lors de la première exécution du programme, en utilisant le mode 3, marquant ainsi la fin de la liste ne contenant que des entrées vierges. L'utilisateur aura le choix entre appliquer les informations d'identification WLAN codées en dur, en utilisant le mode 1, ou vérifier la connectivité en utilisant les informations d'identification de l'EEPROM, ce qu'on obtient en utilisant le mode 8. Pour ce projet, ce mécanisme est limité à un maximum de 10 WLAN différents, par la constante `MEMCNT` déclarée dans le code, tout en effaçant le contenu de la structure et l'initialisant avec une liste prédéfinie d'informations d'identification AP choisies par l'utilisateur. Si nécessaire, il peut augmenter ou diminuer la taille de `MEMCNT`. Les fonctions de base relatives à l'accès à l'EEPROM peuvent être consultées à l'adresse [2]. Je vais maintenant détailler les modes de fonctionnement du WeMos, chacun d'entre eux nécessitant l'emploi des commutateurs DIP (conformément au **tableau 1**), puis le redémarrage du WeMos pour appliquer le changement.

Tableau 2. Répartition des fichiers du code Arduino et leurs fonctions.

Fichier .ino file	Description	Nombre de lignes
wemos-d1-mini-network	Variables globales, setup du mode réseau	240
wlan-utils	Paramètres des connexions WLAN : WPS setup, recherche/listing WLAN, analyse connectivité	110
tft-utils	Paramètres de l'affichage TFT	130
internal-eeprom-utils	Initialisation de l'EEPROM avec cinq identifiants WLAN prédéfinis, réinitialisation de l'EEPROM, lecture/écriture des valeurs	180
external-eeprom-utils	Vérification de la connectivité et de la taille de l'EEPROM externe	40
ap-web-setup	Setup interactif du WLAN : lancement du serveur web, affichage des SSID WLAN détectés, enregistrement du WLAN sélectionné dans l'EEPROM	100

Tableau 3. tableau de gestion de l'EEPROM pour des paramètres AP multiples.

Taille des données (octets)	Description	Description
2	nombre d'AP	entier
32	AP1 SSID	chaîne de caractères
64	AP1 mot de passe	chaîne de caractères
.	.	.
.	.	.
.	.	.
32	APn SSID	chaîne de caractères
64	APn mot de passe	chaîne de caractères

Mode 1 : paramètres WLAN par défaut (codés en dur)

Un microcontrôleur a besoin d'informations d'identification WLAN, composées d'un SSID et d'un mot de passe pour accéder au point d'accès souhaité. Les informations d'identification WLAN peuvent être soit codées en dur dans le programme flashé, soit transmises interactivement au microcontrôleur si l'utilisateur est équipé d'un clavier et d'un écran, ou du moniteur de l'EDI relié via le port série. Pour ce projet, j'utilise la méthode conventionnelle qui consiste à fixer les informations d'identification de l'AP directement dans le programme.

```
14:34:00.668 -> scan available wlans
14:34:05.788 -> Disconnecting previously connected WiFi
14:34:08.148 -> scan completed
14:34:08.148 -> 1 Networks found
14:34:08.148 -> 1: GGGG (6)  (-63)
14:34:08.268 ->
```

Figure 3. Messages du moniteur série de l'EDI Arduino – mode 2, balayage du réseau sans fil.

```
14:35:43.671 -> check external eeprom
14:35:48.751 -> Disconnecting previously connected WiFi
14:35:48.871 -> check external eeprom
14:35:48.871 -> External eeprom isConnected with Status:
14:35:48.871 ->
14:35:48.871 -> TEST: determine size of external eeprom
14:35:48.871 -> 80      FF
14:35:48.871 -> 100     0
14:35:48.871 -> 200     0
14:35:48.871 -> 400     0
14:35:48.871 -> 800     FF
14:35:48.911 -> 1000    AA
14:35:48.911 -> 2000    AA
14:35:48.911 -> external eeprom size: 8192 Bytes
```

Figure 4. Messages du moniteur série - mode 6, vérification de l'intégrité de l'EEPROM externe.

Mode 2 : balayage WLAN

Dans ce mode, le WeMos se déconnecte premièrement de tout WLAN connecté afin de pouvoir rechercher d'autres réseaux présents. Les réseaux locaux sans fil détectés sont répertoriés en fonction de leur SSID et de l'intensité du signal en décibels (dB). Ce mode donne à l'utilisateur un aperçu des possibilités de la mise en réseau de la carte WeMos et du mode à choisir pour accéder à un WLAN. La **figure 3** présente le résultat du démarrage du WeMos dans ce mode, où un seul WLAN a été détecté et connecté par la suite, comme le montre la **figure 1d**.

Mode 3 : effacement de l'EEPROM interne

Ce mode permet au WeMos d'effacer son EEPROM interne afin de préparer la création d'une nouvelle liste de WLANs choisis. Dans ce mode tous les octets de l'EEPROM sont effacés et obtiennent la valeur « 0×00 «.

Mode 4 : image de l'EEPROM interne

Ce mode permet au WeMos d'extraire la structure des informations d'identification du WLAN, comme expliqué dans le **tableau 3**, de l'EEPROM interne. Les **figures 1e** et **f** présentent les captures d'écran des valeurs actuellement mémorisées dans l'EEPROM. Dans ce mode, une série de SSID et de mots de passe formatés sont affichés pour les informations d'identification des points d'accès mémorisés.

Mode 5 : initialisation de l'EEPROM interne

Ce mode permet à l'utilisateur de déposer dans l'EEPROM interne les informations d'identification de 10 WLANs choisis. Cette action est une extension de la manière traditionnelle de mémorisation des informations d'identification codées en dur pour un seul WLAN. Lors du redémarrage en mode 8, le WeMos consulte la liste des informations d'identification pour les connexions WLAN. Si les tentatives de connexion ne permettent pas d'accéder à l'un des WLANs à choix, l'utilisateur devra alors recourir au mode interactif ou WPS.

Mode 6 : vérification de l'état de l'EEPROM externe

Dans ce mode, le WeMos vérifie l'état de l'EEPROM externe, ici un 24LC64 de 8 KB. La **figure 4** montre le résultat de cette vérification, sur le moniteur série. Je n'ai pas utilisé ce composant externe pour la suite du projet puisque l'utilisation de l'EEPROM interne est favorable. L'utilisateur peut se référer au code de l'EEPROM I²C [3] et à la bibliothèque [4] s'il désire l'intégrer dans son projet.

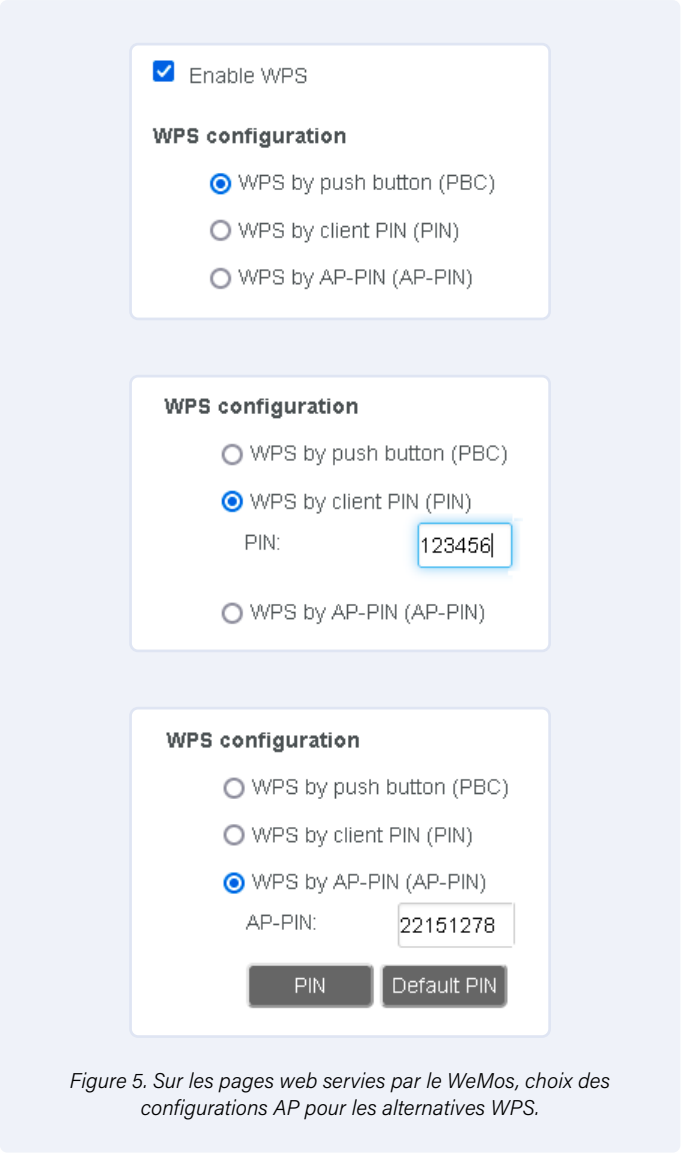


Figure 5. Sur les pages web servies par le WeMos, choix des configurations AP pour les alternatives WPS.



Figure 6. Bouton WPS et indicateur d'un point d'accès typique.

```
14:37:03.552 -> Disconnecting previously connected WiFi
14:37:03.632 -> Try wps if necessary
14:37:03.632 ->
14:37:03.632 -> Could not connect to WiFi ... state= '7'
14:37:03.672 -> Please press WPS button on your router
14:37:03.672 -> WPS config start
```

Figure 7.Messages du moniteur série - mode 7, configuration WPS.

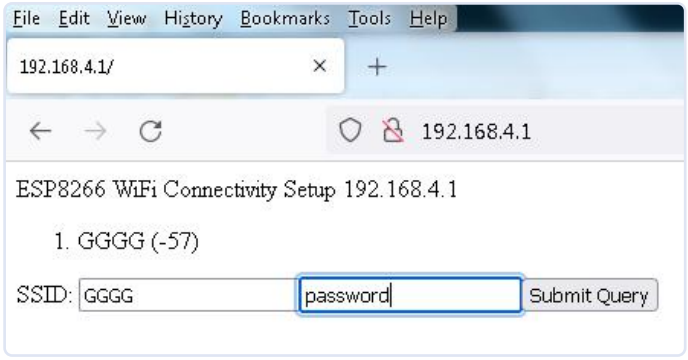


Figure 8. Mode 8, sélection interactive du WLAN.

Mode 7 :connexion directe au WLAN (WPS)

Le WPS est une fonction supportée par les AP modernes qui simplifie le processus de connexion des appareils sans fil à un WLAN. Plutôt que de fournir au microcontrôleur les informations d'identification de l'AP, par exemple le SSID et le mot de passe, le microcontrôleur peut fournir le code de sécurité de l'AP ou son propre code prédéfini (voir la **figure 5**). Une alternative consiste à cliquer sur un bouton-poussoir WPS sur le point d'accès pour établir une connexion directe avec le microcontrôleur requérant (voir **figure 6**). La **figure 7** présente le résultat de l'établissement d'une connexion à l'aide du WPS**.**

Dans ce projet, je limite la connexion WPS à l'alternative du bouton poussoir de l'AP car cela permet un algorithme plus simple et plus générique, s'adaptant à n'importe quel AP à portée de main, comme suit :

```
bool wpsSuccess = WiFi.beginWPSConfig();
if (wpsSuccess) {
    String newSSID = WiFi.SSID();
    if (newSSID.length() == 0)  { wpsSuccess = false; }
}
```

La carte WeMos doit être à proximité de l'AP choisi pour enregistrer les paramètres de l'AP dans l'EEPROM du module. Pour reconnecter le module au même AP il n'est pas nécessaire de cliquer à nouveau sur le bouton WPS, car nous permettons au module de récupérer les paramètres AP mémorisés, pour les réutiliser, comme ceci :

```
WiFi.mode(WIFI_STA);
WiFi.begin(WiFi.SSID().c_str(),WiFi.psk().c_str());
```

Notez que le module ESP8266 qui forme le cœur de la carte WeMos mémorise automatiquement dans l'EEPROM interne les dernières informations d'identification de réseau détectées avec succès, de sorte que nous devons placer notre structure de gestion, décrite dans le **tableau 3**, ailleurs dans l'EEPROM afin d'éviter tout conflit avec les autres mécanismes du microcontrôleur.

Mode 8 : configuration interactive du WLAN

Dans ce mode, le WeMos vérifie d'abord la validité des informations d'identification contenues dans l'EEPROM. Si le microcontrôleur réussit à se connecter à un point d'accès, le processus se termine et la fonction `loop()` prend le relais. Dans le cas contraire, le microcontrôleur passe en mode interactif, dans lequel elle analyse les WLANs à proximité et lance un serveur web avec l'adresse IP 192.168.4.1 pour afficher les SSID des points d'accès détectés. L'utilisateur doit régler l'adaptateur WLAN de son ordinateur sur les paramètres IP fixes du réseau privé WeMos, car le WeMos ne fournit pas de service DHCP. Une page web semblable à celle de la **figure 8** affiche les résultats de l'analyse et permet à l'utilisateur de choisir le réseau qu'il désire. L'utilisateur clique ensuite sur le bouton *Submit Query* pour enregistrer les informations d'identification du WLAN dans l'EEPROM et établir une connexion avec le WLAN sélectionné.

L'utilisation d'EEPROMs représente un atout sérieux lorsqu'il s'agit de mettre des MCUs en réseau sans fil. Dans cet article, plutôt que de coder en dur les informations d'identification du WLAN dans le code du MCU, l'accessibilité à une EEPROM externe ou interne du MCU est utilisée pour conserver et mettre à jour de manière dynamique les informations d'identification de plusieurs points d'accès afin d'établir des connexions. Le code source accompagnant cet article est modulaire, bien structuré, facile à comprendre, et propose une version réutilisable pour gérer les EEPROMs, les WLANs, et pour communiquer avec le MCU à travers le web.

VF : Jean-Philippe Nicolet — 230268-04

À propos de l'auteur

Gamal Labib est un passionné depuis deux décennies de systèmes embarqués et est actuellement mentor (chez codementor.io). Il est titulaire d'un Master en génie mécanique (Meng) et d'un diplôme de doctorat (PhD) en informatique. En plus d'écrire pour des magazines techniques, il est professeur agrégé invité dans les universités égyptiennes et consultant informatique certifié.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (drgamallabib@yahoo.co.uk) ou contactez Elektor (redaction@elektor.fr).

Produits

- **WeMos D1 mini Pro – Module WiFi basé sur ESP8266**
  https://elektor.fr/19185
- **Écran OLED 0.96 pouce (bleu, I2C, 4-Pin)**
  https://elektor.fr/18747
- **Hans Henrik Skovgaard, *Home Appliance Hack-and-IoT Guidebook* (+ carte ESP8266 gratuite), Elektor 2022**
  https://elektor.fr/20370

LIENS

[1] Téléchargement du logiciel : https://elektormagazine.fr/230268-04
[2] Utilisation de l'EEPROM avec l'ESP8266 : https://aranacorp.com/en/using-the-eeprom-with-the-esp8266
[3] Arduino avec une EEPROM I²C : https://playground.arduino.cc/Code/I2CEEPROM
[4] Bibliothèque pour l'EEPROM I²C : https://github.com/RobTillaart/I2C_EEPROM



Genius by wheel_me

Bénéficiez d'une efficacité maximale dans l'intralogistique avec le Genius de wheel.me :
Quatre roues de robot adaptables permettant le transport autonome de marchandises à un prix compétitif.

- Navigation autonome
- Mouvements omnidirectionnels
- Capteurs haute technologie
- Contrôle à distance par l'application wheel.me
- Charge utile ajustable
- Recharge intermédiaire sans interruption des opérations

wheel_me

pour plus de renseignements,
consultez www.wheel.me

5 000 € à gagner !





participez au concours STM32 Wireless Innovation Design Contest

Clemens Valens (Elektor)

Le concours *STM32 Wireless Innovation Design Contest* vous donne l'opportunité de créer et de développer des applications sans fil passionnantes, avec des produits puissants, soutenus par un écosystème riche comprenant des cartes de développement et d'évaluation de STMicroelectronics. Utilisez-les pour tout ce qui vous semble intéressant, et à votre manière ! L'IoT, la robotique, les jeux, la domotique, les tests et mesures, ainsi que l'IA ne sont que quelques-uns des domaines d'application possibles. Tout dépend de vous. Laissez libre cours à votre créativité, amusez-vous et gagnez ! Un total de 5 000 euros est mis en jeu !

NUCLEO-WBA52CG

Les applications Bluetooth ont gagné en popularité au cours des dernières années. Au moment d'écrire ces lignes, le site web de Bluetooth SIG prévoit que d'ici à 2027, 7,6 milliards d'appareils compatibles Bluetooth seront expédiés chaque année. La norme elle-même ne cesse d'évoluer, et la version 5 a introduit la prise en charge de l'IoT. Actuellement, la spécification de base Bluetooth en est à la version 5.4.



Figure 1. Le NUCLEO-WBA52CG est doté d'un Arm Cortex-M33 avec Trust Zone et Bluetooth LE 5.3.

Le STM32WBA52CG est un microcontrôleur RF/SoC prenant en charge le Bluetooth LE 5.3. Il s'agit d'un système ARM Cortex-M33 à très faible consommation fonctionnant jusqu'à 100 MHz. Il dispose de 1 Mo de mémoire Flash et de 128 Ko de SRAM. Le microcontrôleur met en œuvre une unité de virgule flottante (FPU) à simple précision, ainsi qu'un ensemble complet d'instructions DSP. Elle dispose d'une unité de protection de la mémoire (MPU) et de l'écosystème Trust Zone. L'appareil intègre un émetteur-récepteur de 2,4 GHz prenant en charge la technologie Bluetooth Low Energy et des protocoles propriétaires. Le microcontrôleur est monté à l'intérieur d'une enceinte métallique, vous ne pouvez donc pas le voir.

Le STM32WBA52CG est soudé à un circuit imprimé qui, à son tour, est branché sur un autre circuit imprimé. Cette carte est équipée de connecteurs d'extension compatibles

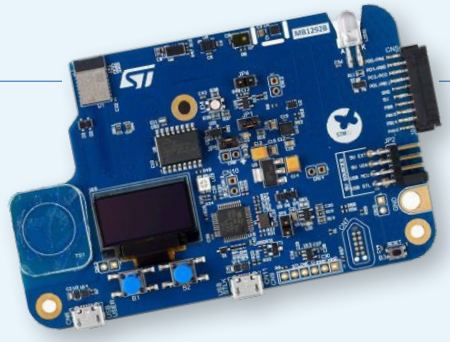


Figure 2. Le discovery kit STM32WB5MM-DK comporte de nombreux capteurs et un petit écran OLED.

avec Arduino et ST Morpho, de cavaliers de configuration, de boutons-poussoirs et de LED, ainsi que d'un régulateur de tension. Un module de débogage/programmation STLINK-V3 se trouve sous la carte. Bien qu'il soit destiné à faciliter le développement d'applications pour le module WBA52CG, vous pourriez être tenté de le dessouder et de l'utiliser plutôt comme module de débogage/programmation STLINK-V3 autonome.
La carte NUCLEO-WBA52CG est préprogrammée avec une application de démonstration, qui peut communiquer avec l'application ST BLE Sensor sur un smartphone. L'application affiche l'état du bouton-poussoir B1 et vous permet d'allumer et d'éteindre une LED sur la carte. C'est un bon début, bien-sûr, mais nous sommes certains que vous pouvez faire mieux. La carte offre de nombreuses autres possibilités d'application. Les puissantes fonctions de sécurité du microcontrôleur permettent de créer des applications IoT sensibles et sécurisées.

Pour plus d'informations :
https://st.com/en/evaluation-tools/nucleo-wba52cg.html

STM32WB5MM-DK

Le discovery kit STM32WB5MM-DK est une plateforme de démonstration et de développement pour le module STM32W5MMG de STMicroelectronics. Ce processeur double cœur 32-bit Arm Cortex-M4/M0+ intègre une radio ultra-basse consommation compatible avec Bluetooth Low Energy (BLE) 5.2, 802.15.4 avec Zigbee, Thread, et des protocoles propriétaires.
Cette carte au format étrange est équipée de nombreux périphériques, tels qu'un écran OLED 128 × 64 de 0,96 pouce, un capteur de température, un capteur accéléromètre/gyroscope, un capteur à temps de vol (ToF) et un capteur de détection de gestes. Il est également équipé d'un microphone numérique, d'une LED RGB et d'une LED infrarouge, de deux boutons poussoirs et d'une touche tactile. Pour les applications nécessitant de l'espace de stockage, il y a une mémoire NOR Flash de 128 Mbits Quad-SPI. Un connecteur STMod+ et des connecteurs d'extension compatibles avec Arduino pour connecter d'autres appareils à la carte.
On pourrait penser que le processeur STM32 situé au milieu de tous ces circuits est le processeur principal de la carte, mais ce n'est pas le cas. C'est le ST-LINK/V2-1 intégré qui fournit des fonctions de débogage et de programmation et un pont USB-série. Le microcontrôleur principal se cache derrière la petite boîte métallique située dans le coin supérieur gauche de la carte.

Le discovery kit STM32WB5MM-DK est préprogrammé avec une application audio Bluetooth. Il envoie les données audio prises avec son microphone numérique à l'application connectée sur un smartphone. L'application peut retransmettre des données audio à la carte. Lorsque vous exécutez la démo en mode full-duplex, veillez à maintenir le téléphone ou la tablette à une certaine distance de la carte, afin d'éviter que l'effet Larsen n'endommage vos tympans.
Parce que le kit de développement a énormément de périphériques, il y a beaucoup de choses que vous pouvez réaliser sans avoir à ajouter quoi que ce soit d'autre. Il est idéal pour les applications IoT et domotiques, mais, avec un peu de créativité, vous pouvez facilement trouver d'autres cas d'utilisation très passionnants.

Pour plus d'informations :
https://st.com/en/evaluation-tools/stm32wb5mm-dk.html

NUCLEO-WL55JC

La carte NUCLEO-WL55JC est une carte d'évaluation pour les microcontrôleurs de la série STM32WL, et en particulier le STM32WL55. Ce microcontrôleur sans fil sub-GHz est basé sur un ARM Cortex-M4/M0+ 32 bits à double cœur, avec une fréquence d'horloge de 48 MHz. Il se caractérise par une très faible consommation d'énergie, un émetteur-récepteur RF intégré avec une plage de fréquences de 150 MHz à 960 MHz, 256 Ko de mémoire Flash et 64 Ko de SRAM.
L'émetteur-récepteur RF à l'intérieur du microcontrôleur prend en charge les modulations LoRa, (G)FSK, (G)MSK et BPSK. Étant un système sur puce sans fil entièrement ouvert, il est compatible avec les protocoles normalisés et propriétaires tels que LoRaWAN, Sigfox, wM-Bus, et plus encore. L'émetteur dispose d'un mode haute puissance de sortie, programmable jusqu'à +22 dBm, et d'un mode basse puissance de sortie, programmable jusqu'à +15 dBm. En Europe, la puissance de transmission de la liaison montante est limitée à 14 dBm (25 mW), il est donc important de vérifier les réglementations en vigueur dans la région où vous envisagez d'utiliser cette carte.

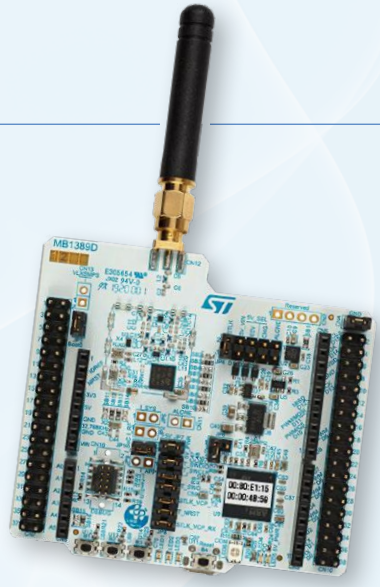


Figure 3. Le NUCLEO-WL55JC est compatible avec les protocoles standardisés ainsi que les protocoles propriétaires tels que LoRaWAN, Sigfox, wM-Bus et plus encore.

La carte sur laquelle le microcontrôleur est monté est équipée de broches d'extension (compatibles avec Arduino et ST morpho), de cavaliers de configuration, de boutons-poussoirs et de leds, ainsi que d'un régulateur de tension. Un débogueur/programmeur STLINK-V3 est également intégré pour faciliter le développement d'applications. Une antenne SMA est également incluse. La carte NUCLEO-WL55JC est livrée préprogrammée avec une application de démonstration de concentrateur de données de capteurs. La carte WL55JC peut également être transformée en un reseau de capteurs compatibles, si vous téléchargez un autre exemple de logiciel embarqué à partir de la bibliothèque STM32CubeWL. Cet exemple est disponible sur la page produit du site web de ST.

La carte a, bien sûr, de nombreuses autres possibilités d'application. L'une d'entre elles qui vient immédiatement à l'esprit est un nœud terminal LoRaWAN. Des instructions expliquant comment procéder peuvent être trouvées en consultant le site web de ST.

La carte NUCLEO-WL55JC existe en deux versions : la WL55JC1 est destinée à être utilisée dans la bande de 865 à 928 MHz, tandis que la WL55JC2 est destinée à la bande de 433 à 510 MHz.

Plus d'informations :
https://st.com/en/evaluation-tools/nucleo-wl55jc.html

230442-04



Bibliothèques et outils logiciels de développement

Tous les produits STM32 sont pris en charge par la plate-forme STM32Cube. STM32Cube est une initiative originale de ST visant à simplifier et à faciliter la vie du développeur en réduisant les efforts, les délais et les coûts de développement. Il fournit aux développeurs une couche d'abstraction matérielle (HAL) et des interfaces de programmation d'applications (API) de couche basse (LL), un ensemble cohérent de composants middleware et de nombreux exemples d'applications qui peuvent être facilement réutilisés pour le développement d'applications personnalisées.

STM32Cube inclut STM32CubeMX, un outil graphique de configuration du logiciel qui utilise des assistants graphiques pour aider le développeur à générer le code d'initialisation en C.

Une chaîne d'outils appropriée pour développer des applications pour les trois cartes présentées dans cet article (et bien d'autres) est, bien sûr, STM32CubeIDE de STMicroelectronics, qui est gratuit. Le MDK-ARM de Keil et Embedded Workbench de IAR peuvent également être utilisés, mais aux frais du développeur.
https://st.com/en/development-tools/stm32cubeide.html

Le package stm32duino boards pour l'EDI Arduino supporte la carte STM32WB5MM-DK. Il reconnaît également la carte WL55JC, mais sans support LoRa. La carte WBA52CG n'est pas (encore) prise en charge.
https://github.com/stm32duino

Les cartes STM32WB5MM-DK et WL55JC sont également compatibles avec le système d'exploitation mbed d'ARM, c'est donc une autre piste à explorer.
https://os.mbed.com/platforms/DISCO-WB5MMG/
https://os.mbed.com/platforms/ST-Nucleo-WL55JC/

STM32

Innovez pour gagner !
5 000 € de prix !

Comment participer ?

Pour inspirer et motiver les participants, STMicroelectronics offre les prix suivants :

1er prix : 2 500 €
2ème prix : 1 500 €
3ème prix : 1 000 €

Pour plus de détails sur les modalités de participation au concours STM32 Wireless Innovation Design Contest, le calendrier et les conditions exactes, veuillez consulter la page web du concours à l'adresse suivante

elektormagazine.com/st-contest



2023 : l'odyssée de l'IA

démarrer avec l'interpréteur de code de ChatGPT

Source : Adobe Stock

Brian Tristam Williams (Elektor)

Les capacités de ChatGPT ont atteint de nouveaux sommets avec la sortie du plugin ChatGPT Code Interpreter (interpréteur de code). Code Interpreter est à présent disponible pour tous les utilisateurs de ChatGPT Plus. Mais de quoi s'agit-il et à quoi sert-il ? Examinons de plus près ce plugin unique et utile de ChatGPT.

ChatGPT d'OpenAI révolutionne notre interaction avec l'intelligence artificielle. Avec l'ajout récent de l'interpréteur de code, il dépasse le simple chatbot pour devenir un assistant puissant pour les développeurs, les spécialistes de sciences de données et les passionnés de programmation.
La mission d'OpenAI de repousser les limites des capacités de l'IA a mené à des développements révolutionnaires, tels que les différents modèles de ChatGPT [1] et des progrès significatifs dans le traitement du langage naturel. La récente version, GPT-4, s'appuie sur cette base impressionnante, mais c'est l'introduction du plugin Code Interpreter [2] qui la différencie vraiment des autres versions.

Interpréteur de code

Notons que le terme « interpréteur » présente ici une certaine ambiguïté. Même la version gratuite, GPT-3.5, est capable d'interpréter du code (c.-à-d. de comprendre ce que fait le code) et de fournir un retour d'information significatif en langage naturel. Ce n'est pas la signification qu'on lui donne ici. Il s'agit d'un interpréteur dans le sens où il peut réellement exécuter notre code Python en tant que langage interprété, dans son propre environnement mis en bac à sable (sandboxed) et protégé par un pare-feu (firewalled), ainsi que dans son propre espace de disque virtuel. Tout comme lorsque vous utilisez le « mode immédiat » dans le bon vieux BASIC traditionnel. Votre session Python reste active pendant toute la session de chat.

Code Interpreter change la donne. Il ne se contente pas d'exécuter le code Python, il interprète, débogue et convertit même le format des fichiers. Les utilisateurs peuvent saisir des problèmes mathématiques et l'interpréteur les résoudra. Il peut également récupérer des données dans différents formats de fichiers, les analyser et les visualiser, ce qui signifie qu'il ne se limite pas à une simple exécution de code. Il peut générer des visualisations de données à une vitesse qui a laissé pantois les analystes de données expérimentés.

Cas d'utilisations

Les applications de Code Interpreter sont innombrables et beaucoup restent à découvrir. Voici quelques-unes de celles que nous avons déjà testées :

- **Test de code :** les utilisateurs peuvent tester des extraits de code en temps réel, afin de s'assurer qu'ils fonctionnent comme prévu.
- **Collaboration** : les équipes peuvent collaborer sur des projets de codage directement dans le chat.
- **Outil d'apprentissage :** il s'agit d'une excellente ressource pour les étudiants qui apprennent Python, offrant des sessions de codage interactives. C'est énorme pour moi – c'est comme si vous aviez votre propre tuteur privé de l'autre côté d'une fenêtre de chat.
- **Expérimentations** : les développeurs peuvent essayer différentes approches de programmation sans quitter le chat.
- **Résolution de problèmes mathématiques :** qu'il s'agisse d'algèbre, ou de calcul différentiel, les bonnes questions vous permettront d'obtenir une solution et de l'appliquer.
- **Conversion de format :** il accepte des données dans différents formats de fichiers, les analyse, les clarifie et les visualise.

Pour démarrer

Bien qu'il existe une version gratuite de ChatGPT, vous devez être un abonné Plus pour utiliser des plugins tels que Code Interpreter. Au moment où nous écrivons ces lignes, la version Beta est disponible, vous devrez donc l'activer. Dans un navigateur de bureau, cliquez sur l'image de votre profil en bas à gauche, puis sur *Settings*

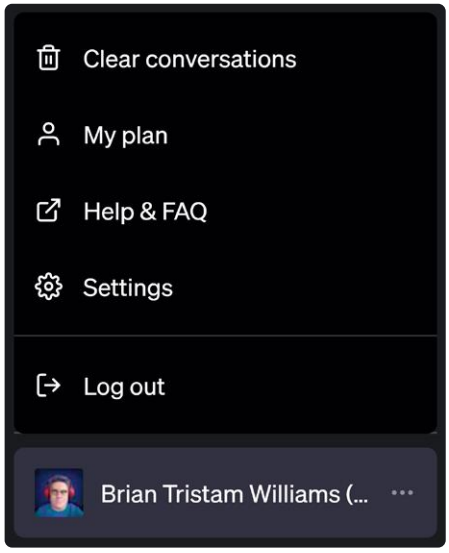


Figure 1. Comment accéder aux paramètres dans ChatGPT.

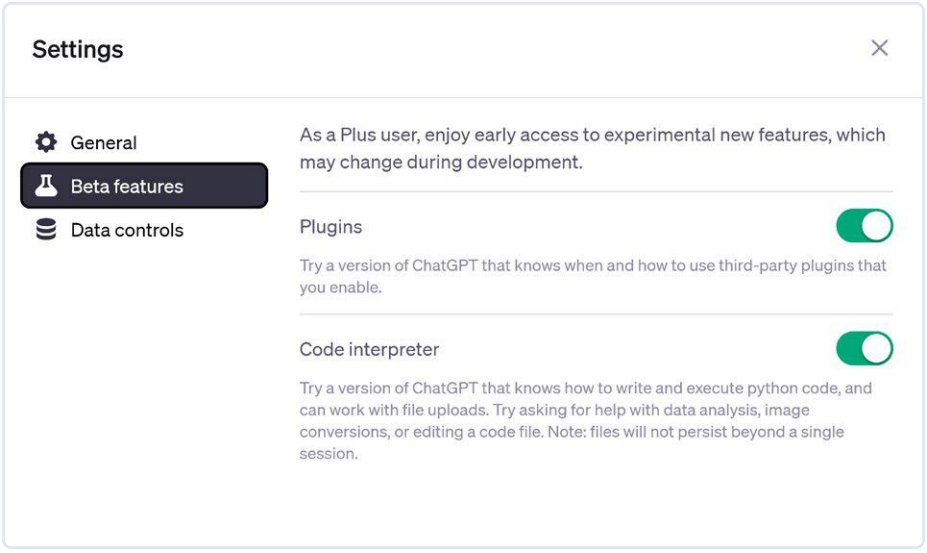


Figure 2. Assurez-vous que Plugins et Code interpreter sont activés.

(**figure 1**). Ensuite, dans la boîte de dialogue qui apparaît, cliquez sur *Beta features*, et assurez-vous que les deux options *Plugins* et *Code interpreter* sont activées (**figure 2**).
Fermez la boîte de dialogue et commencez une nouvelle discussion (*New Chat*). En haut de la fenêtre de chat, vous pouvez choisir entre *GPT-3.5* et *GPT-4*. Ce dernier possède un menu déroulant qui vous permettra de choisir entre le mode par défaut, l'interprète de code et d'autres plugins. Cliquez sur *Code Interpreter* (**figure 3**).

Essayez-le

En plus d'être un environnement d'exécution Python, Code Interpreter est capable d'analyser une grande variété de fichiers, des PDF, des feuilles de calcul Excel, et même des fichiers CSV.
Avec autant de cas d'utilisation, nous n'avons que l'embarras du choix, mais l'une des tests que j'ai effectués consistait à analyser un livre d'Elektor complet que j'ai édité l'année dernière. Voici les étapes de base que j'ai suivies et que vous pouvez utiliser pour commencer à analyser vos propres données :
Uploader son PDF en cliquant sur le signe « plus » à gauche du champ de saisie (**figure 4**), en naviguant jusqu'au fichier et en le sélectionnant pour le charger. Le livre que j'ai utilisé, *Develop Your Own Bluetooth Low Energy Applications for Raspberry Pi, ESP32, and nRF52 with Python, Arduino, and Zephyr*, de Koen Vervloesem (voir la case **Produits**), est un fichier de 41 mégaoctets constitué de 258 pages, je m'attendais donc à rencontrer des difficultés avec ChatGPT.
J'ai commencé par charger le fichier complet et j'ai demandé à ChatGPT de m'en parler (**figure 5**). Le reste de la discussion entre moi (BTW) et l'interprète de code (GPT) est trop longue pour être reproduite ici, mais voici un résumé et quelques conclusions :

BTW : Parlez-moi de ce livre.
GPT : Il compte 258 pages et ses métadonnées de base m'indiquent qu'il s'agit d'un livre sur le BLE. Dois-je regarder à l'intérieur et vous en dire plus ?
BTW : Oui, s'il vous plaît.
GPT : Erreur ! La façon dont j'analyse ceci en Python ne fonctionne pas. Lisez vous-même la table des matières du livre.

À ce stade, vous pouvez être tenté d'abandonner, mais vous arriverez souvent à convaincre le modèle d'utiliser une solution alternative.

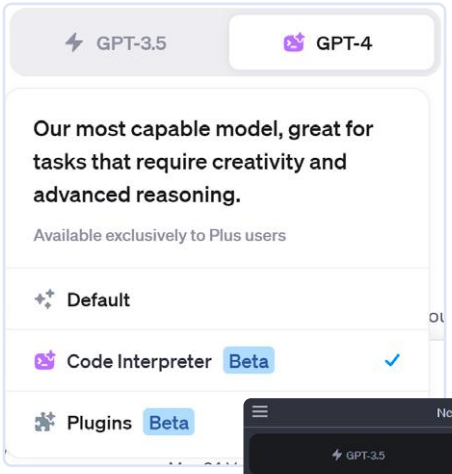


Figure 3. Accéder à l'interpréteur de code dans GPT-4.

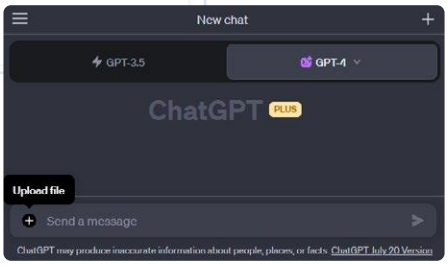


Figure 4. Cliquez sur le signe « plus » à gauche du champ de saisie pour charger votre

BTW: Pouvez-vous utiliser une autre méthode pour analyser le PDF ?
GPT : Bien sûr, essayons la bibliothèque PDFMiner au lieu de PyPDF2. Les 2 000 premiers caractères m'indiquent le sujet du livre.
BTW : Ne vous contentez pas de lire les 2 000 premiers caractères du livre, lisez-le en entier.
GPT, J'essaie d'échapper au travail, tout comme un humain : ce sera difficile et nécessitera beaucoup de ressources. Laissez-moi vous donner plus d'informations sur le livre.
BTW : Lisez tout le livre – j'ai besoin de vous poser des questions spécifiques.
GPT : D'accord.

Après l'avoir convaincu de consulter le document complet, l'environnement a dû faire face à quelques problèmes à cause de la taille et de la complexité du livre. L'environnement d'exécution du code a planté et a dû être réinitialisé (par ChatGPT automatiquement), des bibliothèques ont été mal placées (mais pas par moi), et j'ai dû charger à nouveau l'intégralité du livre. Finalement, j'ai pu tirer quelques informations :

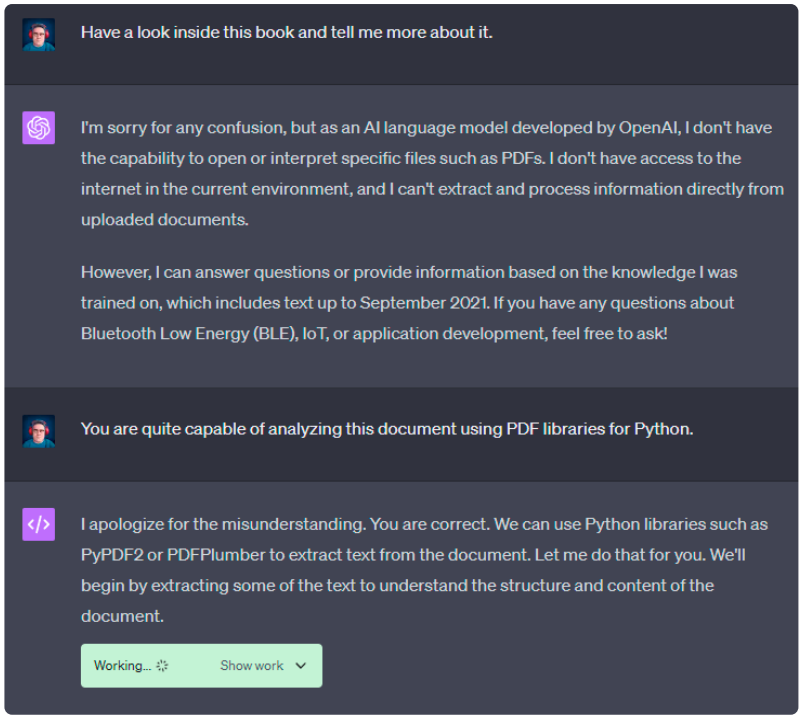


Figure 5. Le fichier est uploadé. Commençons l'analyse, même si nous devons rappeler à l'ordinateur ses capacités.

- Les trois cartes accessoires/microcontrôleurs les plus mentionnées sont le nRF52, l'ESP32 et le Raspberry Pi, avec respectivement 149, 136 et 30 mentions.
- Connaissances utiles à maîtriser avant de commencer à lire ce livre : Une compréhension de base des langages C et Python ; une certaine expérience dans l'utilisation des microcontrôleurs ; une compréhension générale des protocoles de communication sans fil ; l'utilisation de la ligne de commande Linux ; les concepts de base de la mise en réseau.
- Le livre contient 121 URL, et Code Interpreter a été capable de me fournir la liste complète. C'est une aubaine pour l'éditeur et le directeur de publication, qui ont besoin de vérifier si les liens fonctionnent, s'ils ont expiré ou s'ils contiennent des fautes de frappe. J'ai donc demandé à Code Interpreter de m'écrire un script Python pour vérifier tous ces liens et renvoyer les codes d'état HTTP.

Il s'avère qu'un certain nombre de sites web, y compris des sites importants, ont déjà été restructurés (ce livre a été publié il y a seulement un an), ce qui a entraîné la rupture de certains de ces liens. En outre, certains liens dans la version numérique du livre ont été mal interprétés par Adobe PDF à cause du formatage (retours à la ligne, etc.), ce qui n'affecte pas l'édition imprimée sur laquelle nous nous sommes appuyés. Cela m'a donné des idées utiles et réalisables :

- Il est possible d'exécuter quotidiennement un script automatisé pour chaque publication de livre ou de magazine, afin de valider tous les liens, de vérifier les erreurs « 404 not found » ou les nouvelles redirections suspectes (par exemple, lorsqu'un domaine expire et qu'il est repris par un tiers). Alerter l'éditeur si nécessaire.
- Bien qu'il n'ait jamais été facile de déterminer l'ampleur du problème posé par les liens nouvellement rompus après la publication d'un article, un script automatisé pourrait transmettre les données à Code Interpreter dans un fichier CSV pour une analyse automatisée. Nous pourrions alors avoir une meilleure idée sur la fréquence à laquelle les liens deviennent invalides en fonction de leur ancienneté, par exemple.
- Si les liens changent si souvent que cela pose problème, envisager d'utiliser un raccourcisseur de liens de marque, qui peut être mis à jour chaque fois que le fabricant X ou l'éditeur de logiciels Y décide de déplacer sa page, ou qu'un site web plus restreint est désactivé et que nous devons nous fier à une archive tierce telle que *archive.org*.

Le ciel est la limite

La simple exploration ci-dessus ne fait qu'effleurer la surface des possibilités et du potentiel du plugin. Dans les démonstrations présentées sur les blogs d'OpenAI [3], nous avons vu le plugin effectuer de nombreuses tâches de manière excellente, en interprétant de larges volumes de données, en identifiant des tendances, en comparant des variables et en créant des graphiques. Il est même possible d'utiliser le plugin pour créer des animations GIF à partir d'une simple invite de commande. J'ai fait de nombreux tests sur de nombreux sujets, y compris une conversation sur les mathématiques discrètes, l'Amida-kuji, les cycles hamiltoniens, les tours de Hanoï et le code Gray, avec mon fidèle et patient guide, et il a même généré du code Python pour, qui génère à son tour, des visualisations d'idées que je voulais tester [4]. Maintenant que les internautes ont eu plus de temps pour utiliser le plugin Code Interpreter, nous avons vu apparaître encore plus d'utilisations. L'avez-vous essayé ? Si oui, dites-nous comment il vous a facilité la vie !

230181-B-04

Questions ou commentaires ?

Avez-vous découvert des utilisations intéressantes de l'IA dans votre domaine ? Avez-vous des questions ou des commentaires sur cet article ? Faites-le-moi savoir à l'adresse brian.williams@elektor.com ou contactez Elektor (redaction@elektor.fr).

Produit

- **Koen Vervloesem,** *Develop your own Bluetooth Low Energy Applications* **(livre numérique), Elektor 2022**
  https://elektor.fr/20201

LIENS

[1] ChatGPT d'OpenAI : https://chat.openai.com
[2] OpenAI annonce Code Interpreter [Tweet] : https://twitter.com/OpenAI/status/1677015057316872192
[3] Article sur le blog d'OpenAI présentant Code Interpreter : https://openai.com/blog/chatgpt-plugins#code-interpreter
[4] Chat Log: Discussing and Visualizing Discrete Mathematics : https://tinyurl.com/discretegptchat



LoRa, un couteau suisse (1)

le protocole LoRa et ses avantages

Gilles Brocard (France)

Cet article propose la mise en pratique du protocole de transmission LoRa en utilisant des modules Ebyte exploitant le circuit LLCC68 de Semtech avec des moyens simples et un budget restreint. Cette première partie présentera le protocole LoRa et ses avantages.

Aujourd'hui, transmettre et recevoir des données par onde radio est devenue si courant que nous n'y prêtons plus vraiment attention. La 3G, 4G et maintenant 5G nous permettent d'émettre et de recevoir de l'information à grand débit, mais sur des distances relativement courtes, de quelques centaines de mètres tout au plus. Le wifi et le Bluetooth c'est encore un peu moins. Mais, certains domaines ont des besoins différents. Prenons l'exemple d'un agriculteur qui souhaite envoyer des données à une centrale collectrice d'info de comptage d'animaux, ou de température locale (surveillance des gelées) ou toute autre activité nécessitant d'utiliser des mesures fournies par un capteur éloigné de plusieurs kilomètres. Pour répondre à ces besoins, des solutions ont été développées avec des innovations intéressantes, voir [1][2][3].
En substance, LoRa est un protocole de transmission de signal radio qui utilise un format multisymbole « chirpé » pour transmettre des données. Les puces LoRa fonctionnent dans les bandes ISM et elles convertissent les radiofréquences en données. Pour cela, elles utilisent la technologie LoRa qui est une couche physique de niveau bas. Nous reviendrons plus avant sur ces différents termes.
L'utilisation de modules câblés est une solution simple permettant la transmission des données à faible débit entre deux points distants. La société Semtech [4] possède les brevets de LoRa, ce qui lui a permis de développer toute une gamme de circuits émetteur-récepteur LoRa.

Des modules tout prêts

Actuellement, le circuit le plus performant est le LLCC68, mais il n'est pas facile de l'utiliser. Tout d'abord, c'est un CMS très petit, 4 mm de côté pour 24 pattes, et donc très difficile à souder. Par ailleurs, il a besoin d'être entouré de composants RF compatibles avec le gigahertz, le tout sur un circuit imprimé adapté aux contraintes de ces hautes fréquences. La société chinoise Ebyte, spécialisée dans le domaine RF, a développé toute une famille de petits modules intégrant les circuits de Semtech. La gamme Ebyte se décline en de nombreux modèles. Les modèles E220-900T22D (**figure 1**) et E220-900T30D (**figure 2**) contiennent un LLCC68 flanqué d'un microcontrôleur intercalé entre le seul accès, le port de communication UART et le LLCC68. Cependant, ce micro empêche la communication directe avec le LLCC68, ce qui limite considérablement les possibilités de paramétrage du LLCC68. La **figure 3** montre, une fois le blindage retiré, le LLCC68, le micro (ARM CX32L003), un régulateur et un aiguilleur RX/TX en sortie RF. Tous les autres composants sont des résistances ou des condensateurs et des selfs HF ainsi qu'un oscillateur 32 MHz, piloté par quartz.



Figure 1. Le module E220-900T22D d'Ebyte.



Figure 2. Le module E220-900T30D intègre un amplificateur de puissance de 8 dB qui porte la puissance d'émission à 30 dBm soit 1 W.

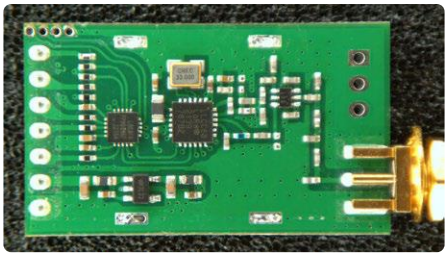


Figure 3. Le module E220-900T22D sans son blindage.



Figure 4. Les modules dont le nom se termine en 'S' offrent une sortie antenne IPEX / U.FL.

Les modules avec le radical '30' intègrent en plus un amplificateur de puissance de 8 dB qui porte la puissance d'émission à 30 dBm soit 1 W. Leurs dimensions sont légèrement supérieures et leur consommation nettement plus importante à pleine puissance (750 mA au lieu de 150 mA pour le modèle 22). Cette réserve de puissance peut être précieuse pour compenser les pertes de tout ordre apporté par les câbles et les raccords HF au voisinage du gigahertz.
Les deux modèles précédents ont une sortie antenne avec un connecteur SMA femelle, mais ces modules existent également avec une sortie antenne IPEX / U.FL, la référence se termine par 'S' à la place de 'D' (**figure 4**).

Sans microcontrôleur intégré

Le module que nous allons utiliser, le E220-900M30S, nous intéresse particulièrement, car il ne comporte aucun micro interposé sur les lignes de commandes. En effet, le LLCC68 est directement accessible via une liaison SPI, c'est ce qu'indique la lettre 'M.' Cela permet d'atteindre tous les paramètres du LLCC68, sans restriction, ce qui est essentiel pour exploiter pleinement tous les avantages de LoRa.
Ce modèle, comme les précédents, existe pour différentes bandes de fréquence de 150 à 930 MHz. Nous avons fait le choix du modèle 900 qui couvre la bande 868 MHz, autorisée en France. Ce module peut facilement être acquis chez les principaux distributeurs pour un prix de l'ordre de 3 à 10 euros suivant le modèle, la puissance et le fournisseur.
Nous combinons ce module, avec quelques composants périphériques, à un micro dont nous maitrisons la programmation et pour lequel nous pouvons créer des applications. Ainsi, nous allons pouvoir réaliser toute sorte de liaison LoRa et d'application intéressantes. Ce module deviendra votre couteau suisse LoRa.

Un peu plus sur LoRa

Le protocole LoRa permet des liaisons à des distances relativement importantes (plusieurs kilomètres) avec des puissances très faibles (quelques dizaines à centaines de milliwatts), mais, revers de la médaille, seulement pour des débits faibles. Dès que vous augmentez le débit (paramétrable), la portée se trouve réduite. Ce qui signifie que LoRa est très efficace pour des applications à faible débit telles que la transmission de commande de portail, la télécommande à très grande portée et la réception de capteurs distants qui ne demandent pas une réactualisation trop fréquente.
Nous avons fait le choix de la fréquence 868 MHz, car elle est libre en Europe sans être encombré comme l'est la bande 433 MHz. De plus, c'est la fréquence choisie pour le protocole LoRaWAN. Elle est règlementée comme toutes les bandes ISM (Industrielle, Scientifique et Médicale). L'avantage des bandes ISM est que, dans la mesure où vous respectez la règlementation européenne, l'émission de signaux radio peut être faite sans aucune déclaration préalable.

Le partage du spectre radioélectrique

Le spectre radioélectrique étant une ressource commune, il est donc nécessaire de le partager pour ne pas gêner les autres par nos transmissions et de ne pas être gêné par les leurs. LoRa utilise **trois** manières de partager une bande de fréquence :
Le partage fréquentiel : on divise la bande de fréquences disponibles en canaux contigus plus ou moins espacés (**figure 5**).
Le partage temporel : avec LoRa, les émissions peuvent et doivent être de courte durée et être espacées dans le temps. Dans ce cas, la même bande peut être utilisée successivement par plusieurs dispositifs.

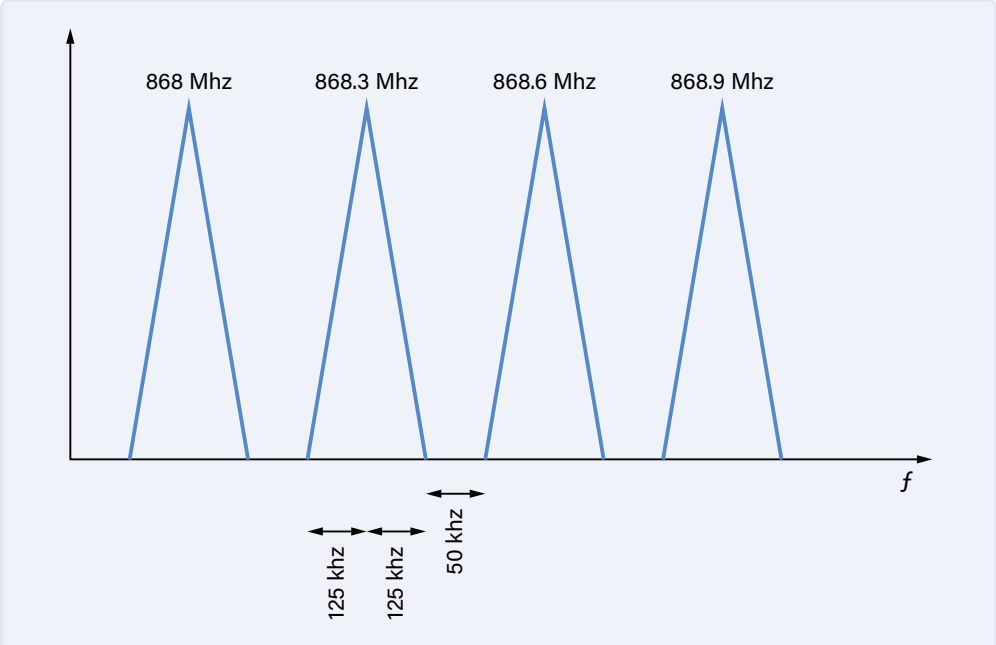


Figure 5. La modulation LoRa utilise une bande passante paramétrable. Cet exemple montre une bande passante BW de 250 kHz soit 125 kHz de part et d'autre de la fréquence centrale. L'excursion de la fréquence s'étendra donc de 868,175 MHz à 868,425 MHz pour la deuxième fréquence de 868,3 MHz.

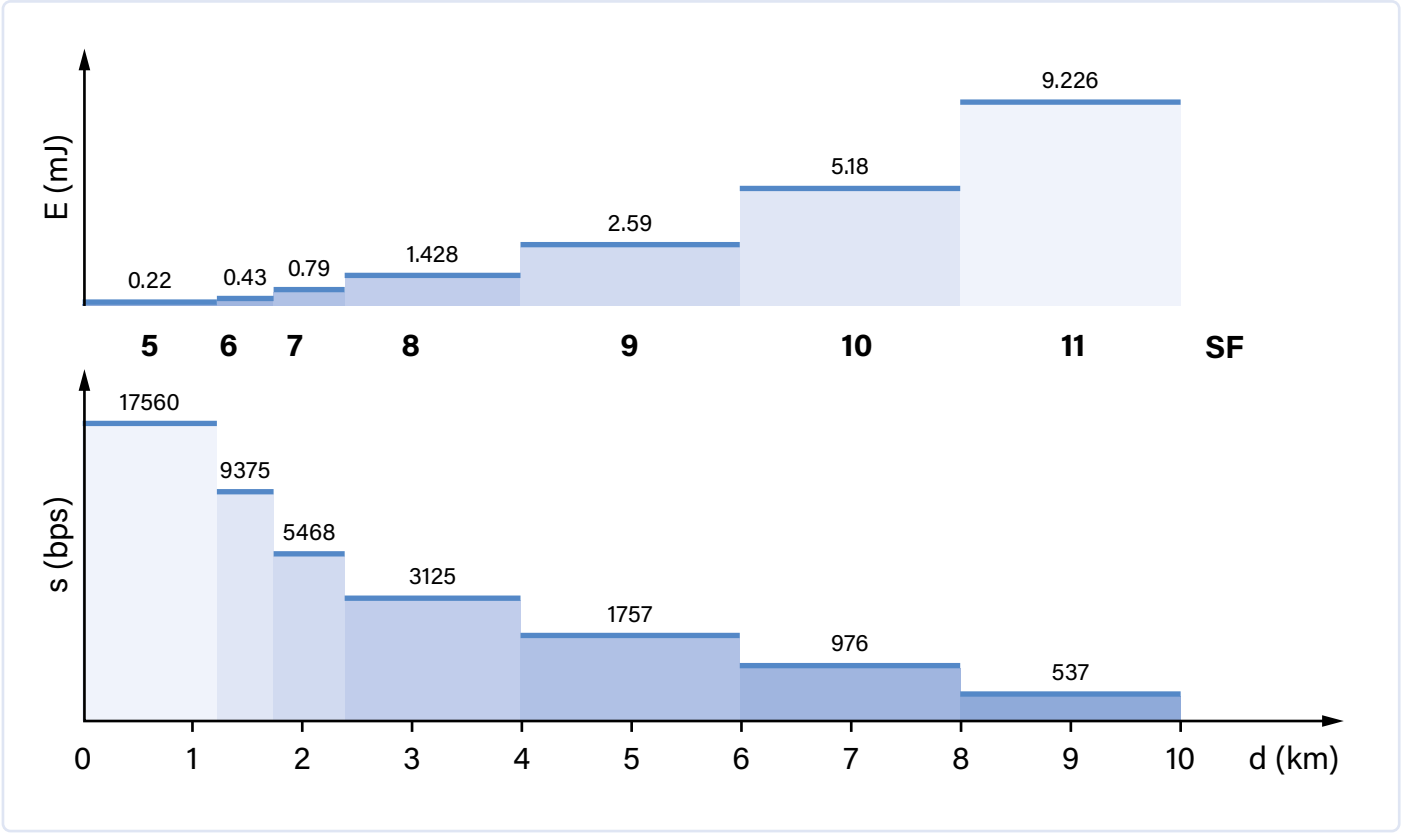


Figure 6. L'influence du ***Spreading Factor*** *(SF, en gras) sur la vitesse de transmission et la portée (en bas), et la consommation (en haut).*

Bien que cette technique soit sujette à des collisions, elle est très utilisée par la modulation LoRa. Une manière de limiter les collisions consiste à surveiller la bande de fréquence et n'émettre que lorsque la bande n'est pas utilisée avant de procéder à une émission. Le LLCC68 permet d'automatiser facilement cette procédure (voire RSSI plus loin).
Le partage par étalement du spectre : ce mode est spécifique à LoRa qui numérise le message à l'aide de *chirps* (excursions de fréquence) d'où le nom de cette modulation (CSS pour *Chirp Spread Spectrum*). Le LLCC68 permet de choisir entre sept codes d'étalement (*Spreading Factor*, SF dans la fiche technique). Les valeurs, de 5 à 11, dépendent de la bande passante choisie (**figure 6**). La SF est l'un des paramétrages importants de la modulation LoRa. Elle détermine notamment la portée, la vitesse de transmission et la consommation d'énergie lors de l'émission. La bande passante (BW) et le CR (*Code Rate*, redondance du codage) sont les deux autres paramétrages LoRa importants.

Le bilan de liaison

Le bilan de liaison (*link budget*) est l'écart entre la puissance d'émission maximum, ici 30 dBm, et le niveau minimal en entrée permettant la réception et le décodage d'un message (sensibilité), ici −129 dBm. Elle est valable pour le paramétrage le plus favorable avec BW = 125 kHz et avec le préampli de réception activé. Le budget de la liaison maximum est donc de 30 + 129 = 159 dBm. La règlementation européenne limite la puissance d'émission à 14 dBm, ce qui produit, dans notre exemple, un bilan de liaison de 14 + 129 = 143 dB.
Vous pouvez vous étonner que nous ayons choisi un module qui permette d'émettre avec une puissance de 30 dBm (1 W) alors que la règlementation européenne n'autorise que 14 dBm (25 mW). En fait, ces 14 dBm sont la puissance rayonnée par l'antenne. Mais, entre la sortie HF du module et l'antenne, il y a de nombreuses causes d'atténuations qui réduiront la puissance atteignant l'antenne. Si par bonheur, il vous reste une puissance d'émission trop importante, et que vous pouvez la mesurer, le paramétrage du module LoRa vous permettra de la réduire.

Un exemple pratique

- Émission : 14 dBm de puissance (limitation due à votre paramétrage).
- Gain de l'antenne d'émission : +2,15 dBi (pour une antenne standard *ground plane* quart d'onde de 86 mm).
- Atténuation maximale du signal lors d'un parcours de 11 km dans l'air : −111,2 dB (voir la formule permettant le calcul de cette évaluation plus loin, la valeur négative indique l'atténuation).
- Gain de l'antenne de réception : +2,15 dBi (pour une antenne standard identique *ground plane* quart d'onde de 86 mm).
- Sensibilité du récepteur (LLCC68 pour BW = 125 kHz, SF = 7), gain max en réception : −124 dBm (fiche technique, page 19, à la première ligne du paragraphe *Sensitivity LoRa*).
- Ajoutons −5 dB pour les pertes dues aux liaisons (câbles, connexions, etc.) pour le module émetteur et −5 dB pour les pertes dues au récepteur.

Le bilan de liaison peut être obtenu simplement en additionnant les valeurs en dBm, dBi et dB, c'est tout l'intérêt des décibels :

14 + 2,15 − 111,2 + 2,15 − 5 − 5 = −102,9 dBm

Cette valeur est largement supérieure au seuil de 124 dBm qui est le seuil de réception LoRa de ce module (sensibilité du récepteur). Nous avons une marge de 21,1 dBm entre les deux, ce qui doit nous permettre de recevoir un signal exploitable.

Une spécificité LoRa, le facteur d'étalement (Spreading Factor ou SF)

SF	Chirps	SNR	BW 125 kHz	BW 250 kHz	BW 500 kHz
5	32	−2,5	oui	oui	oui
6	64	−5	oui	oui	oui
7	128	−7,5	oui −124 dBm	oui −121 dBm	oui −117 dBm
8	256	−10	oui	oui	oui
9	512	−12,5	oui −129 dBm	oui	oui
10	1024	−15	non	oui −129 dBm	oui
11	2048	−17,5	non	non	oui −127 dBm

Tableau 1. La synthèse des mesures présentées sur la page 19 de la fiche technique du LLCC68 concernant le facteur d'étalement, *Spreading Factor* en anglais.

Première colonne
Le facteur d'étalement SF dépend de la bande passante BW. Par exemple, pour BW = 125 kHz, SF peut avoir une valeur comprise entre 5 et 9, pour BW = 250 kHz la valeur 10 peut également être utilisée et pour BW = 500 kHz la valeur 11.

Troisième colonne
Le rapport signal sur bruit (SNR) en fonction de la valeur du paramètre SF.
Ne soyez pas étonné de voir des rapports SNR négatifs, car le démodulateur LoRa est capable de récupérer un signal largement sous le niveau du bruit RF. Par exemple, à BW = 500 kHz et SF = 11, le signal récupérable peut-être 17,5 dB en dessous du bruit. Cela veut dire que le signal peut être 56 fois plus faible que le niveau du bruit radio…

Deuxième colonne
La deuxième colonne va limiter notre enthousiasme, car elle donne la taille du codage de la transmission pour atteindre ces performances. Le temps de transmission est directement proportionnel à ces valeurs, et donc la consommation d'énergie aussi. La consommation instantanée du module reste constante, elle se produit simplement pendant un temps plus long.

Quatrième, cinquième et sixième colonne
Pour BW = 125 kHz et pour SF = 7 le LLCC68 est capable d'extraire la partie utile (*payload*) à partir d'un signal reçu de −129 dBm (79 nV_{RMS}). Cette valeur est particulièrement faible, le LLCC68 est extrêmement sensible pour la partie analogique et très performante pour la partie numérique appliquant le protocole LoRa !

Évaluation de la portée

En faisant abstraction des parasites électromagnétiques ambiants, ce qui n'est évidemment jamais totalement possible en pratique, il est possible de calculer la distance théorique maximum d'une liaison. La formule donnant la distance est :

$$\text{distance} = \sqrt{\frac{10^{\left(\frac{\text{Link Budget}}{10}\right)}}{1755 \cdot \text{frequency}^2}}$$

Link budget est le budget de liaison du module utilisé pour notre exemple, en respectant la norme européenne est de 143 dBm.
Frequency est la fréquence du module : 868 MHz.
Distance est la portée en kilomètres (hors bruit HF).
En respectant la règlementation européenne, la valeur du bilan de liaison maximum de 143 dB avec notre E220-900M30S en 14 dBm, le calcul donne une portée de 388 km. Mais, arrêtons de rêver,

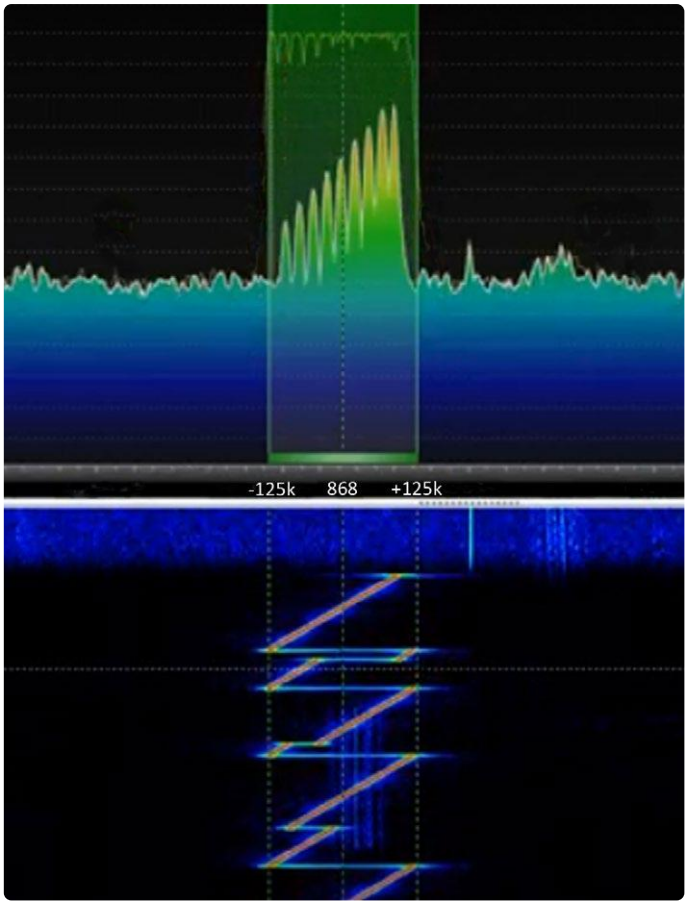


Figure 7. Il est possible de visualiser directement la modulation LoRa avec un logiciel SDR et une clé SDR associée. Ici, nous voyons la modulation pour une fréquence de 868 MHz avec une bande passante de 250 kHz. La lenteur de la transmission LoRa permet de voir la modulation du codage des symboles. Tout en haut de l'image, l'axe vertical est gradué en puissance moyenne du signal (dBm) et l'axe horizontal en fréquence. Dans la partie basse de l'image, on peut observer le déroulement de la transmission en fonction du temps avec les variations de fréquence caractéristiques de la modulation LoRa.

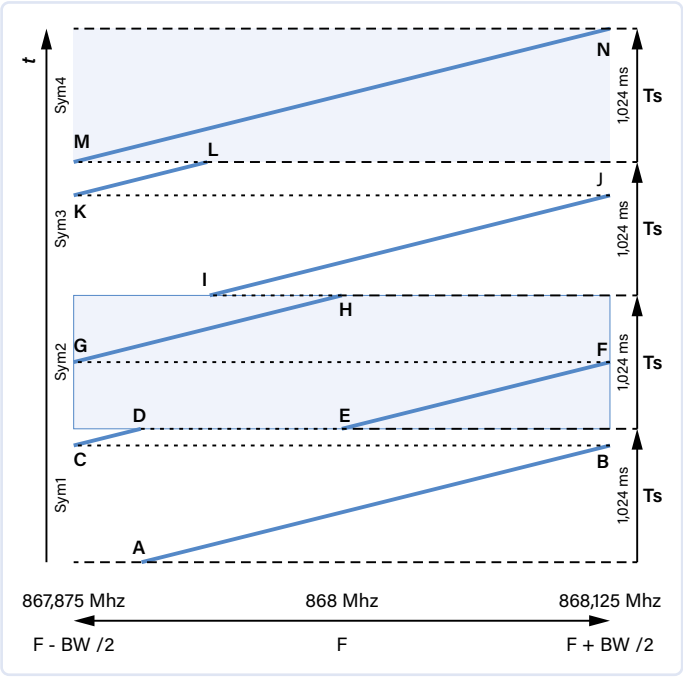


Figure 8. Ce chronogramme montre la transmission de quatre symboles avec SF = 8 et BW = 250 kHz. Attention, le temps est représenté par l'axe vertical, la fréquence occupe l'axe horizontal.

cette distance, toute théorique, ne sera jamais atteinte en pratique. Néanmoins, dans de bonnes conditions, une distance maximale entre 10 km et 20 km peut être atteinte, à vue et hors zones urbaines.
À titre d'exemple, sachez qu'en avril 2020, une distance de 832 km a été atteinte par un module LoRa de performance tout à fait comparable au LLCC68. Toutefois, les conditions d'émission et de réception étaient très optimisées. En effet, à l'émission une antenne était accrochée à un ballon-sonde plafonnant à 38 km d'altitude et que la réception fut faite par une passerelle LoRaWAN située au milieu d'une zone montagneuse, donc plutôt bien isolée des parasites HF.

Transmission d'un symbole

Il est possible de visualiser directement la modulation LoRa avec un logiciel SDR et une clé SDR associée, comme le montre la **figure 7**.
La **figure 8** est un chronogramme montrant la transmission de quatre symboles LoRa avec SF = 8 et BW = 250 kHz. Si notre fréquence centrale est 868 MHz, l'excursion de fréquence comporte $2^8 = 256$ incréments pour transiter de la fréquence minimum 867,875 MHz à la fréquence maximum 868,125 MHz.

Le codage de l'information de chaque symbole réside dans la valeur initiale de la fréquence du balayage linéaire de cette fréquence. Par exemple, pour le symbole 1, la fréquence de départ A, débute à 32/256 de l'excursion totale de la fréquence, le codage du symbole transmis sera donc 32. Pour le symbole 2, la fréquence de départ E, démarre à 128/256, le codage du symbole sera 128. Pour le symbole 3, la fréquence de départ I, commence à 64/256, le codage du symbole sera 64 et ainsi de suite. Pour le symbole 4, la fréquence de départ M débute à 0/256 et le codage du symbole est 0. Les valeurs des 4 symboles transmis sont donc successivement 32, 128, 64 puis 0.
L'excursion de fréquence est faite par de petits incréments qui ont une résolution qui dépend de la valeur de SF. Pour SF = 8, $2^{SF} = 2^8$, c'est-à-dire 256 intervalles, pour SF = 11, ils seraient au nombre de 2048. La durée de chaque incrément Tc dépend de la bande passante. Pour BW = 250 kHz, ils durent 4 µs, pour BW = 500 kHz leur valeur se réduit à 2 µs et 8 µs pour 125 kHz. C'est la raison pour laquelle, avec LoRa, le temps de transmission d'un message est inversement proportionnel à BW et, par contre, elle double pour chaque incrément de 1 de la valeur de SF.
Ce procédé de modulation, emprunté aux technologies RADAR, consiste en la transmission de symboles qui sont une excursion linéaire de fréquence d'une onde sinusoïdale autour d'une fréquence centrale. Ces balayages de fréquence sont appelés CHIRP (*Compressed High Intensity Radiated Pulse*). Pour LoRa seuls les chirps à croissance linéaire sont utilisés pour la transmission des symboles, comme celles qui relient les points M à N. Les chirps à décroissance linéaire de fréquence ne sont utilisés que pour la fin du préambule destiné à la synchronisation.

Le paquet LoRa

Le message transmis est intégré dans une trame physique qui est appelée paquet (*packet*). Elle se compose de quatre parties (**figure 9**) dont certaines sont optionnelles et/ou paramétrables :

1. Un préambule (preamble), toujours obligatoire, mais paramétrable. C'est une succession de plusieurs (en général huit) chirps croissants et de deux chirps décroissants. Ce préambule est indispensable pour que le récepteur puisse se synchroniser.
2. En-tête (header), optionnel. Il est présent par défaut si le mode explicite est choisi (absent avec le mode implicite). Il est toujours transmis avec un CR (*code rate*) de 4/8 pour avoir le maximum de chance d'être reçu. Il indique :
   - La taille des données (payload).
   - Le taux de code (*code rate* ou CR) utilisé pour le reste de la trame, entre 4/5 et 4/8, mais le taux 4/5 est le plus utilisé. Un taux de *m*/*n* veut dire que pour *m* bits d'information utile, le

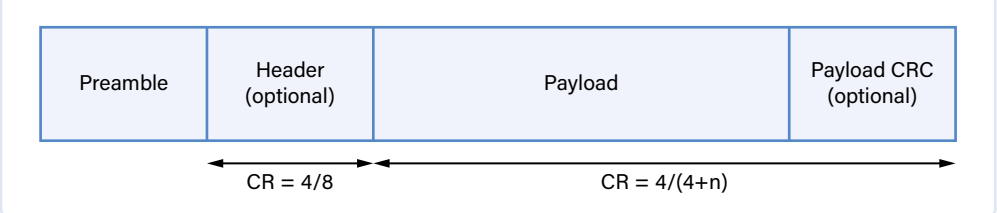


Figure 9. Le paquet LoRa est composé de quatre parties dont certaines sont optionnelles et/ou paramétrables.

codeur génère un total de *n* bits de données, dont *m*–*n* bits sont redondants. Les bits redondants permettent de détecter des erreurs de transmission.

- - Il précise si un CRC (code d'erreur) est présent ou non.

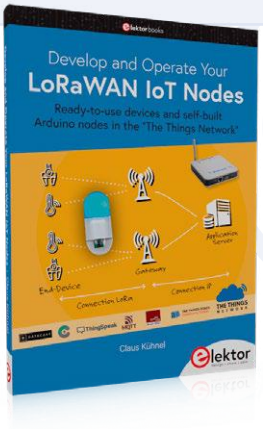
3. Les données (payload). La taille maximale du paquet de données est de 256 octets. Elle dépend de la valeur du facteur d'étalement et du paramétrage du buffer. Plus SF a une forte valeur, plus la taille du paquet doit être faible. La durée maximum du message est limitée.
4. Un contrôle d'erreurs (CRC), optionnel. Il permet de savoir si les données, à la réception, sont complètes et sans erreur.

Dans la suite de cet article, nous présenterons le circuit imprimé intégrant le module LoRa E220-900M30S piloté par un Arduino Nano. Un programme C++ permet le paramétrage complet du LLCC68, mais surtout son utilisation en émission et réception pour la transmission de messages. Le réglage de la portée et de l'autonomie seront également détaillés avec SF, BW, CR, LDRO ainsi que les valeurs renvoyées par le récepteur, le RSSI, la sensibilité mesurée, le rapport signal sur bruit et les statistiques de réception (taux d'erreur).

230140-04

Produits

- **Great Scott Gadgets HackRF One Software Defined Radio (1 MHz to 6 GHz)**
  https://www.elektor.fr/18306
- **CircuitMess Chatter – DIY LoRa Communicator**
  https://www.elektor.fr/20407
- **Claus Kühnel, *Develop and Operate Your LoRaWAN IoT Nodes*, Elektor 2023**
  https://www.elektor.fr/20147



LIENS

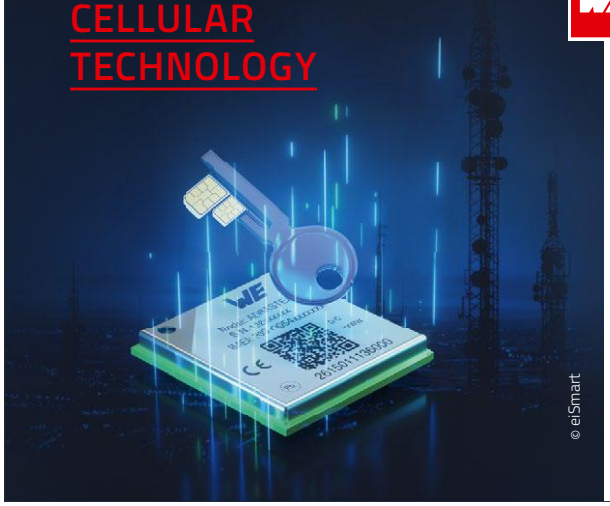
[1] R. Lacoste, « bruit et sensibilité des récepteurs - une introduction », Elektor 12/2016 : https://www.elektormagazine.fr/magazine/elektor-201612/40006

[2] R. Lacoste, « LoRa - échange débit contre portée », Elektor 1/2017 : https://www.elektormagazine.fr/magazine/elektor-201701/40060

[3] R. Lacoste, « LoRaWAN - architecture, protocole, sécurité et opérateurs », Elektor 3/2017 : https://www.elektormagazine.fr/magazine/elektor-201703/40245

[4] Semtech : http://www.semtech.fr

YOUR KEY TO CELLULAR TECHNOLOGY



WÜRTH ELEKTRONIK MORE THAN YOU EXPECT

WE are here for you!
Join our free webinars on:
www.we-online.com/webinars

Adrastea-I is a Cellular Module with High Performance, Ultra-Low Power Consumption, Multi-Band LTE-M and NB-IoT Module.

Despite its compact size, the module has integrated GNSS, integrated ARM Cortex M4 and 1MB Flash reserved for user application development. The module is based on the high-performance Sony Altair ALT1250 chipset. The Adrastea-I module, certified by Deutsche Telekom, enables rapid integration into end products without additional industry-specific certification (GCF) or operator approval. Provided that a Deutsche Telekom IoT connectivity (SIM card) is used. For all other operators the module offers the industry-specific certification (GCF) already.

www.we-online.com/gocellular

- Small form factor
- Security and encryption
- Long range/wordwide coverage
- Multi-band support

#GOCELLULAR

puit de courant réglable avec générateur d'horloge intégré

testez vos alimentations, convertisseurs de tension et batteries

Roland Stiglmayr (Allemagne)

Malgré tous ses avantages pour le test approfondi des sources d'énergie, le puit de courant (aussi appelé charge électronique) est souvent absent du laboratoire d'électronique. Ces appareils sont en effet assez coûteux. Le mieux est donc de s'armer d'un fer à souder et de pinces coupantes et de construire soi-même une e-charge.

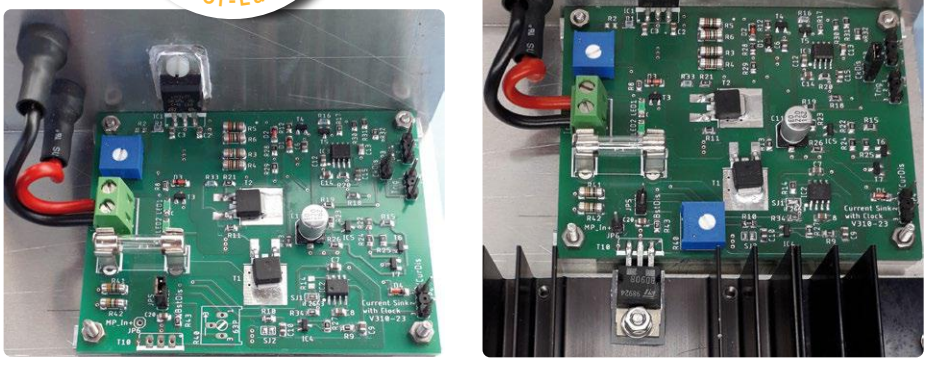


Figure 1. À gauche, le puit de courant sans booster et avec un grand radiateur. Fixer le LM317 avec une vis en plastique était une mauvaise idée ! À droite, la carte complète avec le booster (et vis métallique).

Comment testez-vous vos alimentations, convertisseurs CC/CC et batteries ? Vous disposez probablement d'une série de résistances de puissance pouvant servir de charge à l'appareil testé (appareil appelé ci-après la source) et vous mesurez le courant et la tension. Ensuite, vous faites une série de mesures pour relever les caractéristiques statiques de la boucle de contrôle à différentes charges et tensions d'entrée. Ces mesures permettent en outre de calculer la résistance interne de la source. Cette procédure marche bien, mais elle est complexe et prend du temps.

Il serait plus pratique et plus rapide d'utiliser un puit de courant programmable à courant constant indépendant de la tension appliquée. Cependant, même avec un tel puit de courant, la façon dont la source réagit aux variations rapides de la charge (son comportement dynamique) n'est pas mesurable. Or ce comportement est essentiel pour l'évaluation complète de la source. Si c'est une alimentation, elle réagit souvent au changement de charge soudain par une pointe de tension brève mais notable, pouvant faire dysfonctionner, voire endommager l'équipement qui lui est connecté. En outre, souvent instable, la boucle d'asservissement de ce type de source a tendance à osciller à haute fréquence, en particulier en cas de variation rapide de la charge. Ces oscillations ont les mêmes effets délétères que les dépassements.

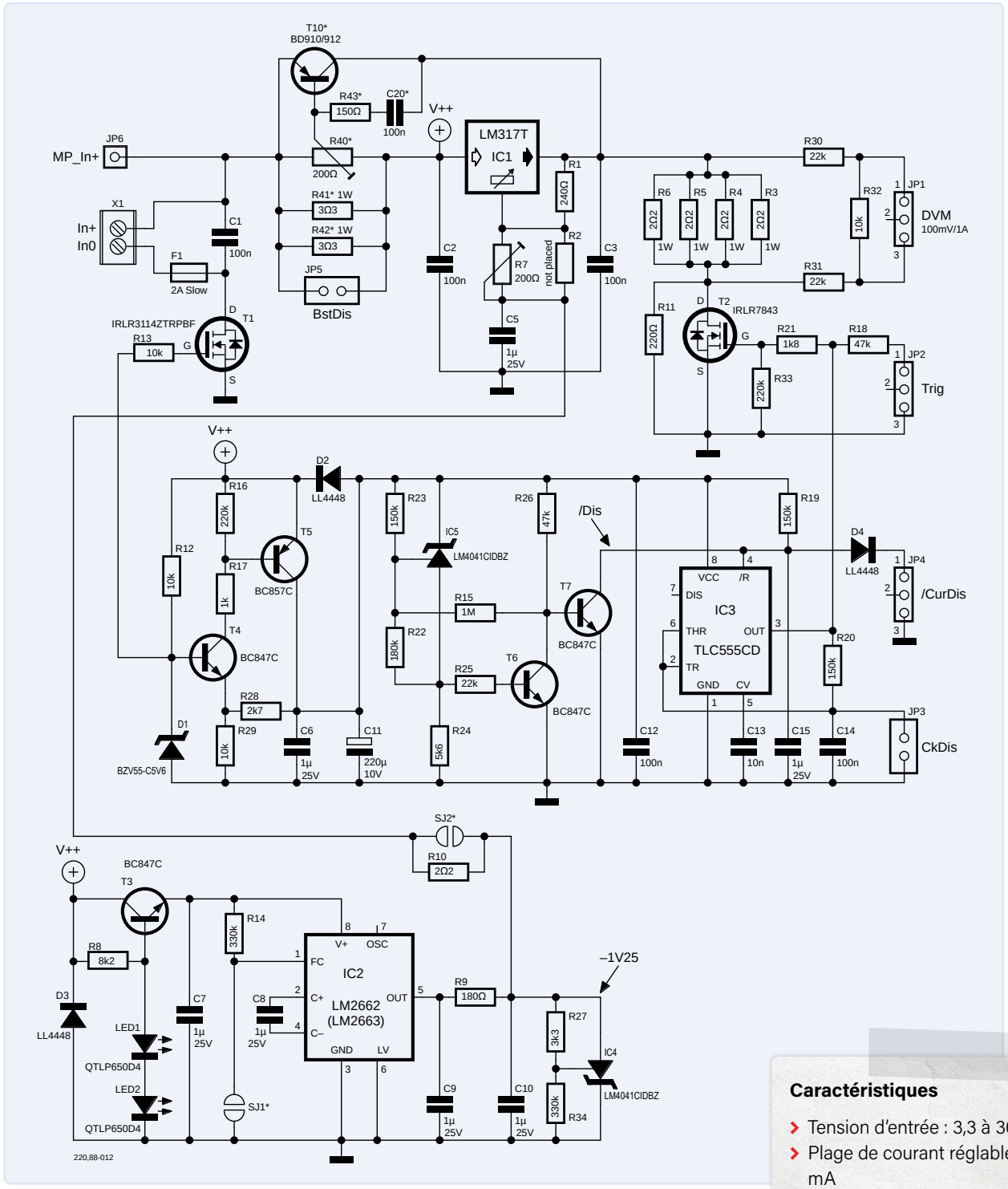


Figure 2. Schéma de principe du puit de courant.

C'est pourquoi, un bon puit de courant se doit de fournir une charge statique et de pouvoir changer instantanément sa valeur. C'est exactement ce que fait le puit de courant décrit ici. Deux versions de ce puit de courant sont illustrées (**fig. 1**). Simple, la conception du circuit ne contient pas de composant exotique et ne nécessite pas d'alimentation séparée. Selon les radiateurs des semi-conducteurs, le puit de courant peut dissiper jusqu'à 18 W (50 W avec booster). Il admet une tension d'entrée maximale de 30 V. Bien que la puissance dissipée max. ne soit pas très élevée, le mode impulsionnel du puit de courant permet une bonne estimation du comportement dynamique d'alimentations électroniques de forte puissance. L'encadré **caractéristiques** témoigne des capacités de ce puit de courant.

Principe de fonctionnement

Presque toutes les sources et puits de courant sont basés sur un circuit linéaire qui délivre une tension constante. Cette tension alimente la charge, une résistance fixe. La loi d'Ohm implique que si la tension aux bornes d'une

Caractéristiques

- Tension d'entrée : 3,3 à 30,0 V
- Plage de courant réglable : 30 à 1900 mA
- Dissipation max. de puissance : 18 W (50 W avec booster)
- Horloge (commutable) : environ 50 Hz, rapport cyclique de 50 %
- Temps de montée/descente : <3 µs
- Protection contre l'inversion de polarité à l'entrée
- Protection contre la surchauffe
- Arrêt en cas de sous-tension
- Signal de déclenchement pour l'oscilloscope
- Sortie pour la mesure du courant
- Alimenté par la source testée.

telle résistance est constante, le courant qui y circule l'est aussi. Si nous négligeons le courant de repos interne, ce courant circule dans le circuit et agit donc comme une charge sur l'alimentation testée. Ici, cela reste **vrai même** si la tension à l'entrée du circuit change, de sorte que nous avons bien une charge qui absorbe toujours un courant constant. Si la tension aux bornes de la résistance de charge est réglable, le courant absorbé par le puit de courant peut aussi être réglé. Utiliser un CI régulateur de tension est une solution évidente pour réaliser cette fonction. Outre sa simplicité, cette solution est sûre car les régulateurs de tension ont pour la plupart des fonctions de protection intégrées.

Toutefois, un régulateur ne peut peut jamais délivrer moins que sa tension de référence majorée de la bande interdite, soit 1,25 V en général. Il nous faut trouver une astuce pour pouvoir régler le courant jusqu'à zéro. En polarisant à -1,25 V la borne de rétroaction du régulateur (le point de référence que le régulateur de tension utilise pour régler la tension de sortie), on peut obtenir une tension de sortie (par rapport à la masse du circuit) de 0 V. Et là, la plage de réglage du courant descend donc aussi à zéro. Une simple pompe de charge suivie d'un étage stabilisateur de tension permet de produire la tension négative requise.

Pour obtenir une charge en mode impulsionnel, le plus simple est d'utiliser un MOSFET de commutation de la résistance de charge en sortie du régulateur de tension. La fréquence d'horloge doit être choisie voisine de la fréquence de la ligne (secteur). Ainsi, l'alimentation testée est chargée pendant une demi-période complète de la tension secteur. Cela permet de vérifier le bon dimensionnement des condensateurs de filtrage de l'alimentation.

Circuit

Le circuit du puit de courant réglable est illustré (**fig. 2**). Les composants marqués d'un astérisque (*) ne sont nécessaires que pour le booster. IC1 est un régulateur de tension éprouvé : le LM317. Ses caractéristiques thermiques sont optimales : il dissipe jusqu'à 20 W (à une température ambiante raisonnable), a une tension d'entrée max. de 37 V et est protégé contre les surchauffes et les surintensités.

IC2 est un inverseur de tension LM2662 basé sur une pompe de charge. Il produit une tension négative stabilisée à -1,25 V par IC4, un régulateur shunt de précision LM4041. Ensuite, le pot. ajustable R7 produit le point de référence réglable appliqué à la borne *Adjust* du régulateur IC1. Les résistances R1 et R7 forment un diviseur de tension alimenté sous +1,25 V côté R1 et -1,25 V côté R7. Ainsi, le point de référence de IC1 peut être ajusté sur une plage de -1,25 V à -0,2 V. La tension de sortie du régulateur de tension varie donc de 0 V à 1 V env. Cette tension est appliquée au groupe parallèle de résistances de charge R3-R6, qui détermine ainsi le courant absorbé par le circuit. Le transistor commutateur T2, qui gère l'horloge du courant, est en série avec les résistances de charge.

Pour que le régulateur shunt IC4 ne dissipe qu'une faible puissance, la pompe de charge IC2 n'est alimentée que sous 3 V. Cette tension est produite par l'émetteur-suiveur T3. La tension de référence est fournie par deux LED vertes en série : la tension de coude est nettement plus prononcée qu'avec une Zener de même tension nominale.

Le reste du circuit est alimenté par un régulateur de tension simple composé de T4, T5 et de la diode Zener D1, qui fournit la tension de référence. Ce circuit a l'avantage d'avoir une faible chute de tension (LDO) et peut donc fournir la tension de grille requise pour T1 et T2 même sous faible tension d'entrée. Le signal de commande du transistor T2 provient d'un bon vieux timer 555 (IC3). Tant qu'un cavalier JP3 (*ClkDis*) permet de court-circuiter le condensateur C14, responsable de l'oscillation du timer, la sortie d'IC3 est haute et T2 est constamment passant. C'est le mode de fonctionnement statique. Si le cavalier est retiré, le courant est pulsé. À la mise en marche, si la tension d'entrée est inférieure à la tension minimale de fonctionnement, IC3 est réinitialisé via son entrée Reset. Dans ce cas, la sortie du timer est basse et aucun courant ne peut circuler dans la charge. L'élément de temporisation, R19/C15, retarde l'activation de la sortie. Un niveau bas externe appliqué sur JP4 (*/CurDis*) désactive le puit de courant. Cela permet, notamment de réaliser une décharge temporisée d'une batterie rechargeable à l'aide d'une carte Arduino.

L'arrêt en cas de sous-tension est effectué par le régulateur shunt, IC5 (également un LM4041) configuré pour être bloqué à une tension inférieure à 3 V, et ainsi maintenir la réinitialisation du timer via T6 et T7. Si la tension d'alimentation dépasse 3,1 V env., IC5 commence à conduire, T6 devient passant et T7 se bloque : IC3 n'est plus réinitialisé.

Une tension proportionnelle au courant de charge est disponible sur JP1 pour être mesuré à l'aide d'un voltmètre. Le facteur de proportionnalité est de 0,1 V/A. Un signal d'environ 6 V crête est disponible sur JP2. En mode pulsé, il peut être utilisé pour déclencher un oscilloscope.

Le MOSFET T1 protège l'entrée du circuit contre les inversions de polarité. Si la polarité de la tension d'entrée est correcte, un courant traverse la diode interne de T1 pour permettre à l'alimentation interne de démarrer. Par conséquent, T1 est entièrement activé. Si le puits de courant est en court-circuit, le fusible lent F1 fond et évite tout incendie.

Plus de puissance

Si vous désirez un puit de courant plus puissant, vous pouvez ajouter un étage de puissance (booster) en amont du régulateur. Avec un bon radiateur, la puissance dissipée admissible peut alors atteindre 50 W. Cependant, le courant maximal reste le même. Malheureusement, les fonctions de protection du LM317 ne concernent pas le booster, il est donc important de prévoir un dissipateur de taille adéquate. Le booster abaisse d'1,2 V la tension appliquée, la tension d'entrée minimale passe alors à 5 V. Pour cette raison, si la tension est inférieure à 8 V, le booster doit toujours être désactivé en installant un cavalier sur JP5 (*BstDis*).

Le principe de fonctionnement du booster est assez simple. Une partie du courant d'entrée traverse la résistance série (R41 // R42). Le pot. ajustable R40 prélève une partie de la chute de tension sur la résistance série et l'applique à la jonction base-émetteur du transistor PNP T10. R40 est réglé de sorte

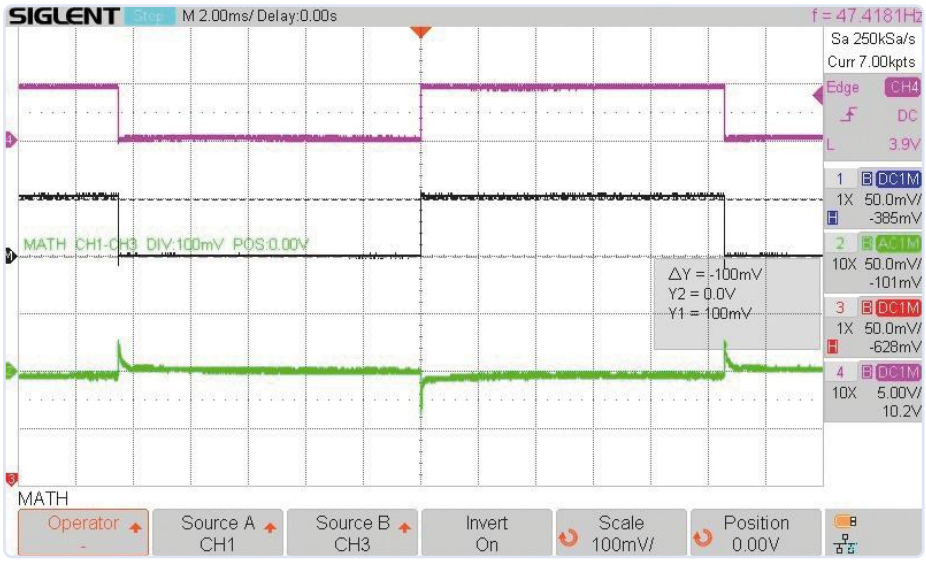


Figure 3. Comportement dynamique d'une alim. linéaire de labo à 12 V / 1 A.

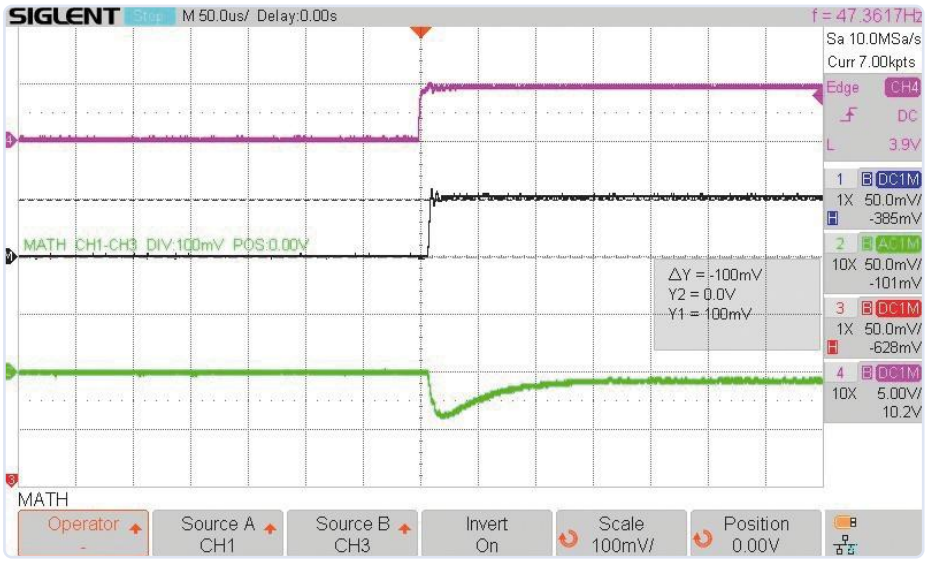


Figure 4. Réponse de l'alim. de labo de la fig. 3 à la commutation de la charge.

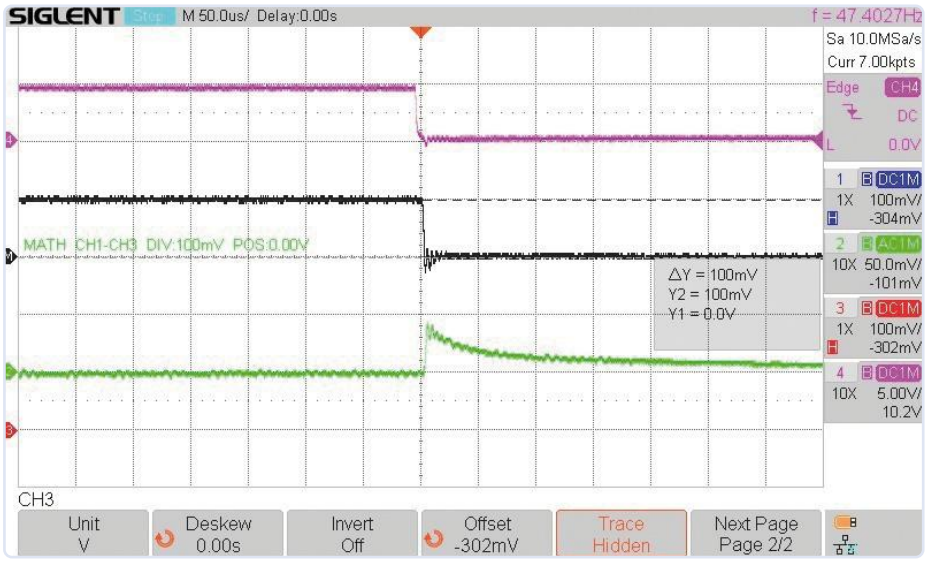


Figure 5. Réponse de l'alim. de labo de la fig. 3 à la coupure de la charge.

que T10 commence à conduire à un courant d'environ 0,6 A. T10 fournit donc tout courant supérieur à 0,6 A directement aux résistances de charge R3-R6, tandis que le LM317 continue de réguler la tension. Le réseau RC R43/C20 prévient toute oscillation.

Procédure de test et interprétation des résultats

Pour mesurer les paramètres statiques d'une source, désactivez le timer en montant un cavalier sur JP3. Mesurez la tension de sortie de cette source avec un voltmètre. Mesurer sur le puit de courant conduirait à un résultat erroné en raison de la chute de tension dans les fils de connexion. Connectez un oscilloscope configuré en entrée CA en parallèle avec le voltmètre. Augmentez alors le courant par étapes tout en mesurant la tension de la source. Vous obtenez ainsi une série de mesures décrivant l'écart de régulation de la source testée. En outre, l'oscilloscope met en évidence toute tendance à l'oscillation de la source testée. En augmentant le courant, gardez à l'esprit la puissance dissipée maximale du puit de courant. Le LM317 peut dissiper jusqu'à 18 W avec un bon radiateur, mais cette valeur est facilement atteinte dès que la source a une tension de sortie relativement élevée. Heureusement, les protection internes du LM317 (zone de fonctionnement sûre, SOA) le protègent des dommages.

On peut ensuite mesurer les caractéristiques dynamiques de la source testée avec l'oscilloscope. À cet effet, l'horloge du puit de courant doit être activée. Il est très important de connecter l'oscilloscope directement à la source, sinon les résultats seront faussés par des pointes de tension dues à l'inductance des fils de connexion. La procédure de mesure est la même que pour les mesures statiques. La puissance dissipée est divisée par deux grâce à un rapport cyclique de 50 % : cela permet d'utiliser des tensions et/ou courants plus élevés. Chaque oscillogramme ci-après montre le signal de déclenchement en haut (CH4, trace violette), le courant au milieu (trace noire, mesure différentielle sur JP1 avec 0,1 V/A), et la tension de sortie en bas (CH2, trace verte). Notez aussi les différentes résolutions horizontales et verticales des oscillogrammes.

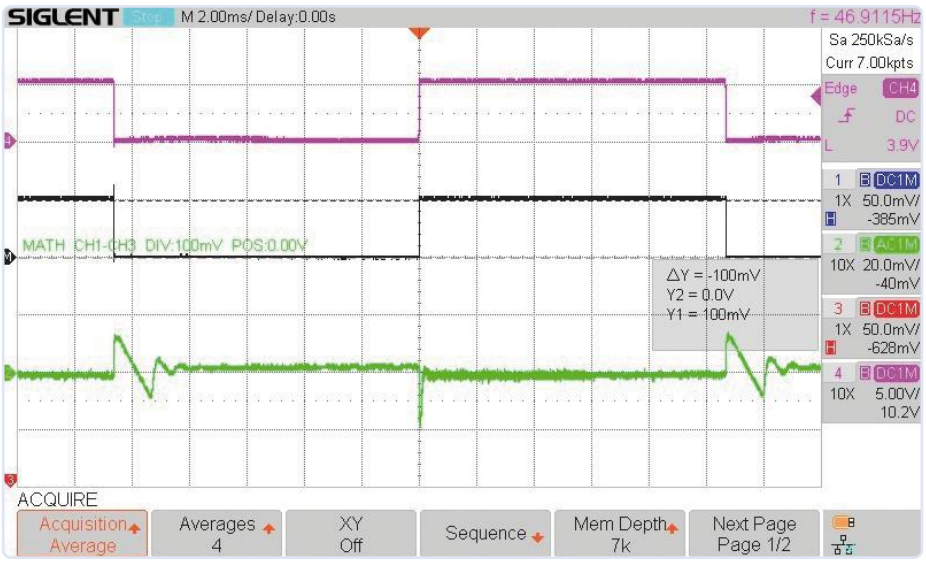


Figure 6. Comportement dynamique d'une alimentation à découpage à 12 V / 1 A.

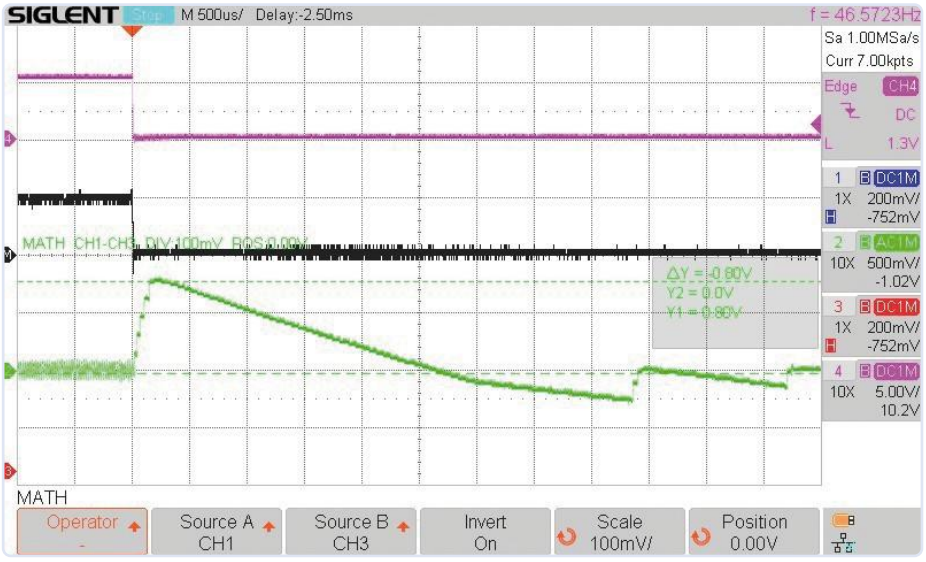


Figure 7. Réponse d'un convertisseur de tension flyback à la coupure de la charge à 13 V / 1 A.

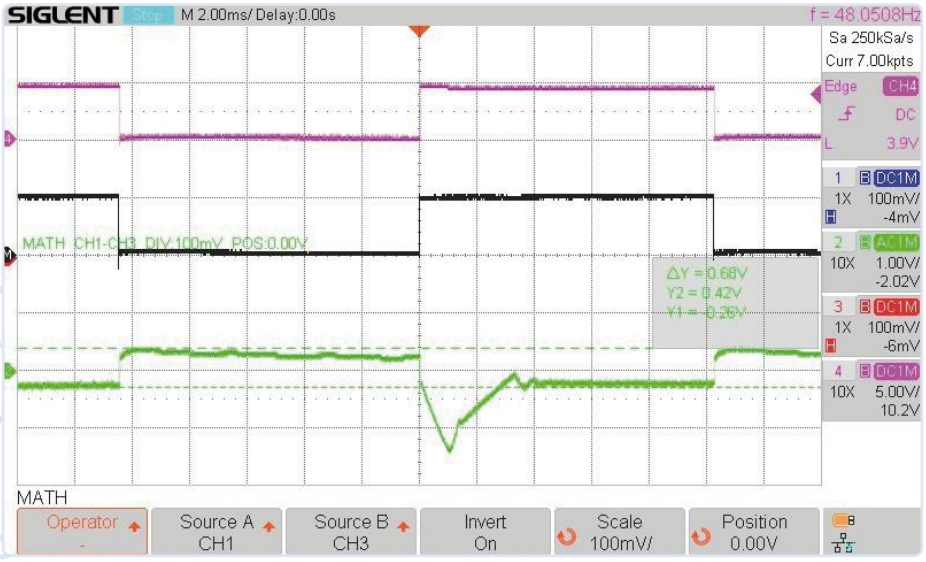


Figure 8. Comportement dynamique d'un chargeur USB à 1 A.

Figure 3 : comportement dynamique d'une alimentation linéaire de laboratoire. **Figure 4** et **5** : réponse aux changements tout ou rien, respetivement allumage et extinction de la charge. L'amplitude et la durée des dépassements sont les principaux éléments à considérer ici. **Figure 6 :** c'est le pendant de la **figure 3**, mais pour une alimentation de labo à découpage. **Figure 7 :** réponse d'un convertisseur flyback à un délestage de charge. **Figure 8** et **9** : comportement dynamique de deux chargeurs USB ; l'élément clé ici est le décalage persistant de la tension de sortie. L'écart de régulation, aussi dénommé chute de tension sous charge, et le courant associé déterminent la résistance interne de la source.

Vous pouvez aussi voir si le filtrage interne et l'amortissement de sortie sont correctement dimensionnés. Si la source a tendance à osciller à haute fréquence, cette mesure le révélera de manière fiable. Si vous concevez votre propre source, ce puits de courant est un outil précieux pour optimiser les constantes de temps et la stabilité de la boucle de contrôle.

La mesure dynamique d'une batterie rechargeable fournit sa résistance interne instantanée, ce qui donne une bonne indication de son état de santé. Notez que le courant de mesure ne doit pas dépasser 1 à 2 fois la capacité nominale. La résistance interne d'une cellule de batterie lithium-ion de 1500 mAh de capacité est en général < 30 mΩ et encore plus faible pour les capacités importantes (souvent < 10 mΩ). Il est conseillé de toujours mesurer et noter la résistance interne d'une pile neuve et de l'utiliser ultérieurement à des fins de comparaison. **Figure 10** : cet oscillogramme montre qu'avec une résistance interne de 5,9 mΩ par élément (R_i = 35,2 mV/(2 A × 3)), la batterie est en bon état. **Figure 11** : ici, avec une résistance interne de 53 mΩ par élément (R_i = 316 mV/(2 A × 3)), la batterie approche à contrario rapidement de la fin de sa durée de vie utile.

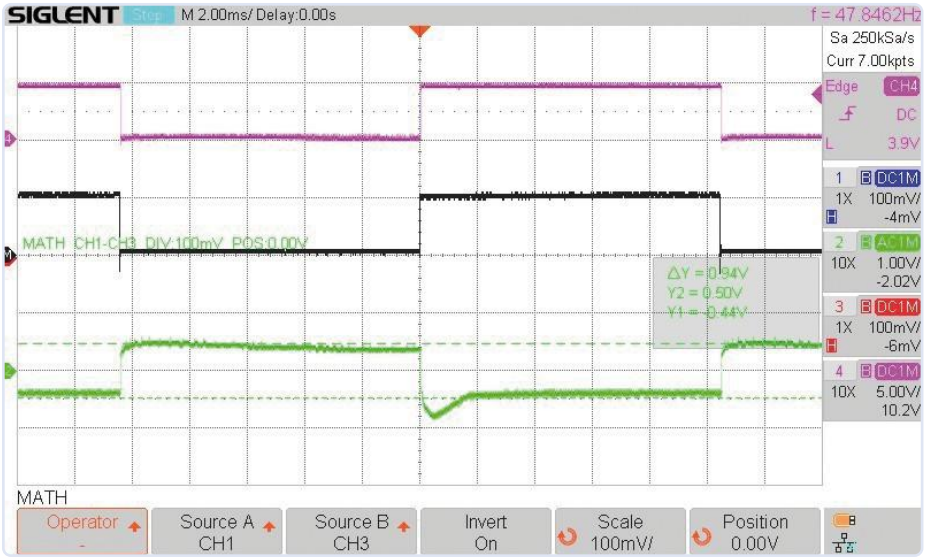


Figure 9. Comportement dynamique d'un autre chargeur USB à 1 A.

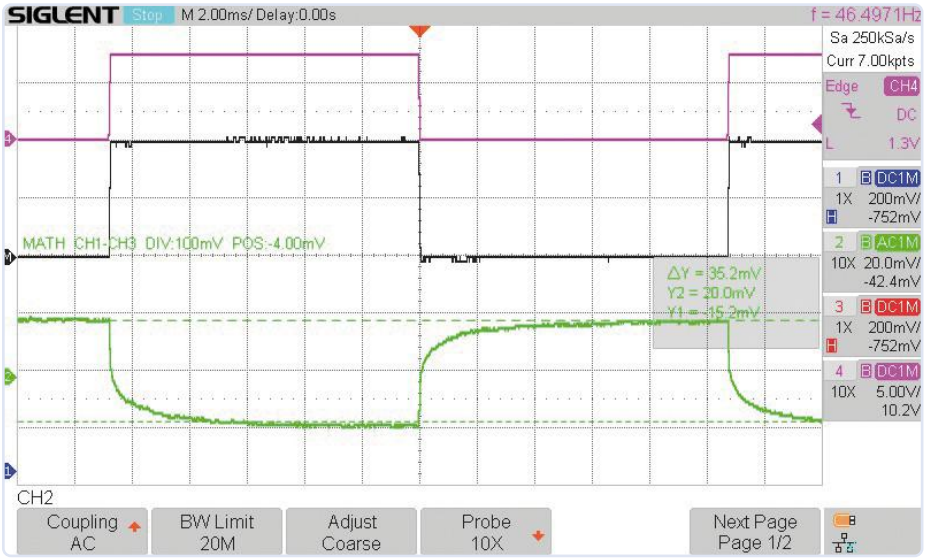


Figure 10. Mesure à 2 A de la résistance interne d'une batterie saine de trois éléments LiPo.

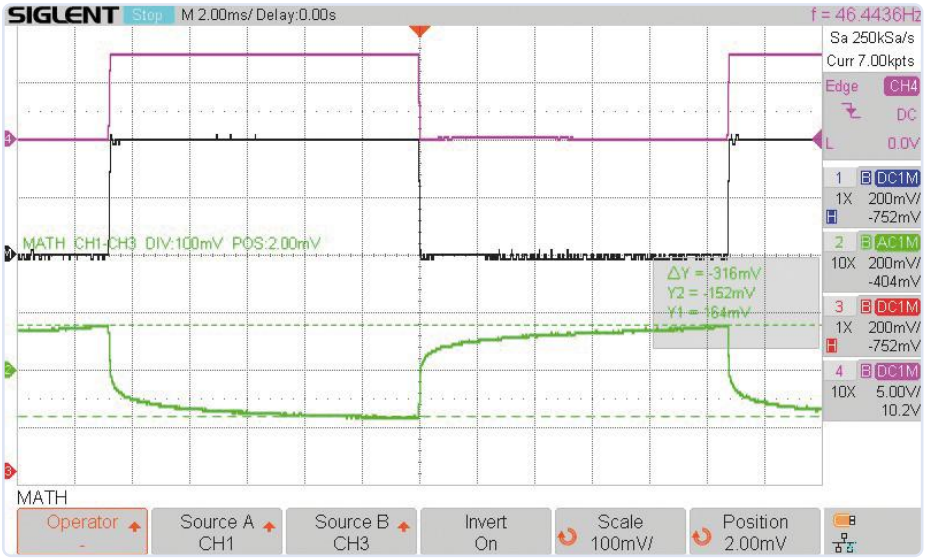


Figure 11. Mesure à 2 A de la résistance interne d'une batterie usagée de trois éléments LiPo.

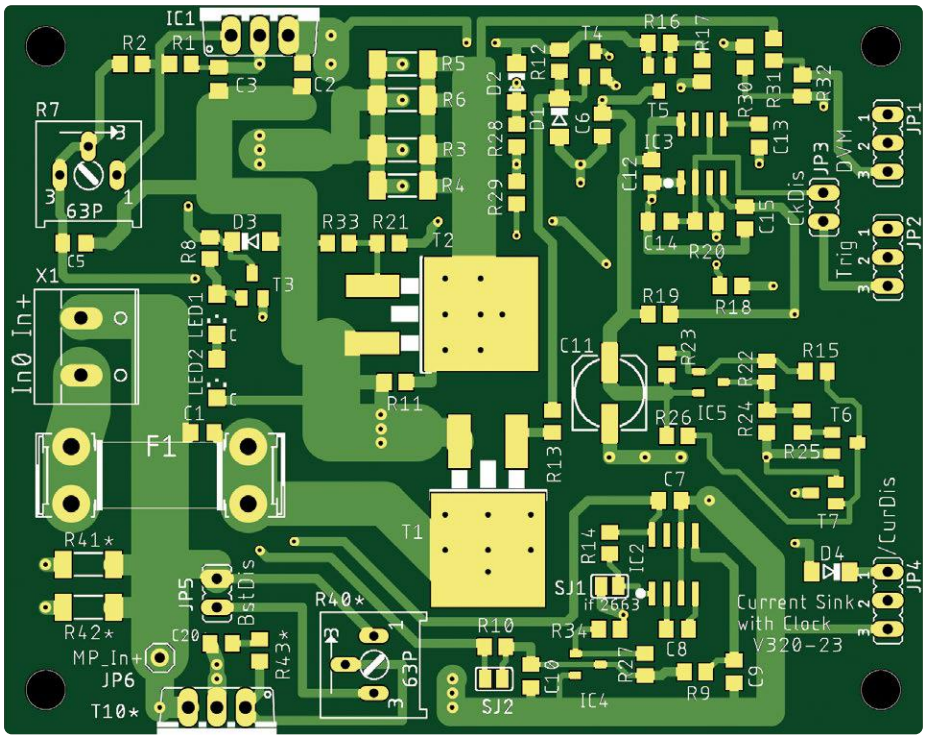


Figure 12. Circuit imprimé vierge de la carte du puit de courant réglable avec booster.

Construction

Le circuit imprimé a été conçu pour permettre l'ajout d'un booster. Ce circuit imprimé (**fig. 12**), et son schéma électronique (**fig. 2**), peuvent être téléchargés au format PDF [1]. Les fichiers Eagle sont inclus dans le téléchargement.

L'empreinte de T1 et T2 est compatible DPAK (TO-252) et D2PAK (TO-263). Les LM2662 et LM2663 conviennent pour IC2, vous ne devriez donc pas avoir de problèmes pour vous en procurer. Omettez les composants marqués d'un astérisque (*) si vous n'avez pas besoin du booster. Dans tous les cas, il faut faire bien attention aux radiateurs d'IC1 et de T10, car la fiabilité du puit de courant repose principalement sur eux. Leur résistance thermique doit être inférieure à 2 K/W et les semi-conducteurs doivent être montés avec de la pâte thermique de haute qualité et des vis métalliques munies des rondelles isolantes appropriées.

Mise en route du dissipateur de courant

Avec un LM2663 pour IC2, faites le pont de soudure SJ1. Placez un cavalier sur JP5 (*BstDis*). Connectez le puit de courant à une alimentation variable préalablement réglée sur 3 V avec limitation de courant de 100 mA et réglez le courant du puit à son minimum par le pot. ajustable R7. Le puit de courant devrait absorber un courant de repos d'environ 10 mA. Augmentez ensuite la tension à 3,3 V. Avec le pot. ajustable R7, réglez le courant à environ 80 mA. Réduisez ensuite à nouveau la tension. À environ 3,1 V, le courant doit retomber à sa valeur de repos. Cela vérifie que l'arrêt en cas de sous-tension fonctionne correctement. À une tension d'entrée de 5 V et le pot. ajustable R7 au min., le courant doit être inférieur à 30 mA. Sinon, le courant peut être réduit en soudant le pont SJ2.

Pour mettre le booster en service, tournez le pot. ajustable R40 dans le sens antihoraire pour anihiler tout courant de base dans T10, puis retirez le cavalier sur JP5. À une tension d'entrée de 15 V et un courant de 1 A, réglez le pot. ajustable R40 pour que le courant d'entrée du régulateur IC3 soit de 0,6 A. Vous pouvez calculer le courant d'entrée d'IC3 en mesurant sur JP5 la ddp aux bornes de R41//R42. Ici, 1,0 V correspond à 0,6 A. Ensuite, portez la tension d'entrée à 25 V et le courant à 2 A. Au besoin, ajustez le réglage après mise en température du puit de courant.

VF : Yves Georges — 220388-04

À propos de l'auteur

Roland Stiglmayr étudia l'informatique dans les années 1970. Il a 40 ans d'expérience en R&D. Il s'est surtout occupé de développement d'ordinateurs centraux, de systèmes de transmission de données par fibre optique, de relais radio gérés à distance pour réseaux sans fil et de systèmes de transmission d'énergie sans contact. Il est actuellement consultant et se passionne pour le transfert de connaissances.

Des questions, des commentaires ?

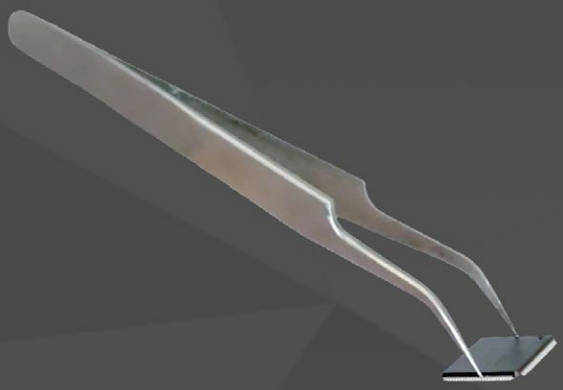
contactez Elektor (redaction@elektor.fr).

Produits

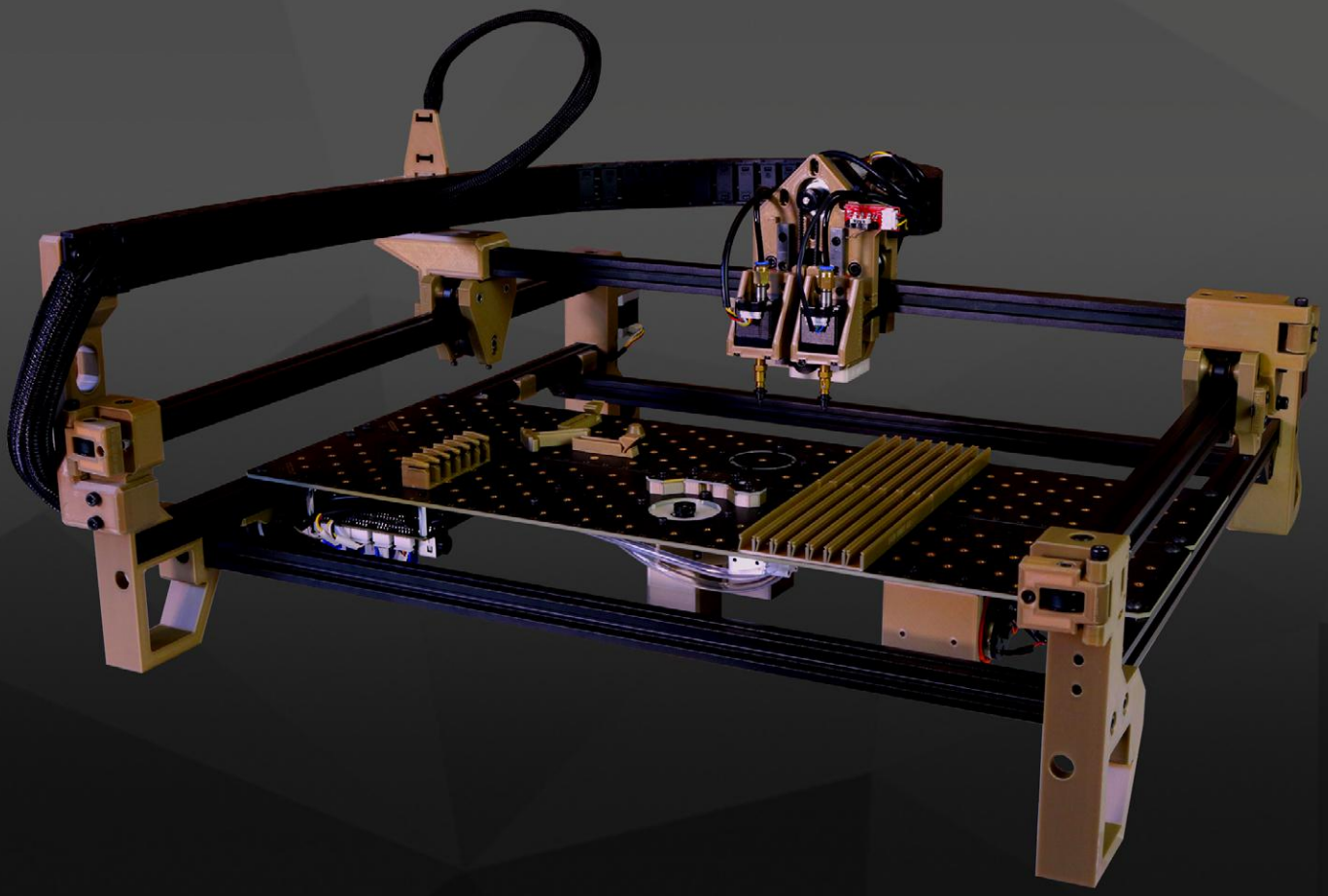
- **Siglent SDL1020X-E charge électronique CC programmable (200 W)**
  https://elektor.fr/19254
- **PicoScope 2204A oscilloscope USB (10 MHz)**
  https://elektor.fr/17303

LIEN

[1] Téléchargements du projet sur Elektor Labs :
https://elektormagazine.fr/labs/adjustable-current-sink-with-integrated-clock-generator



VS



C'est sans appel.

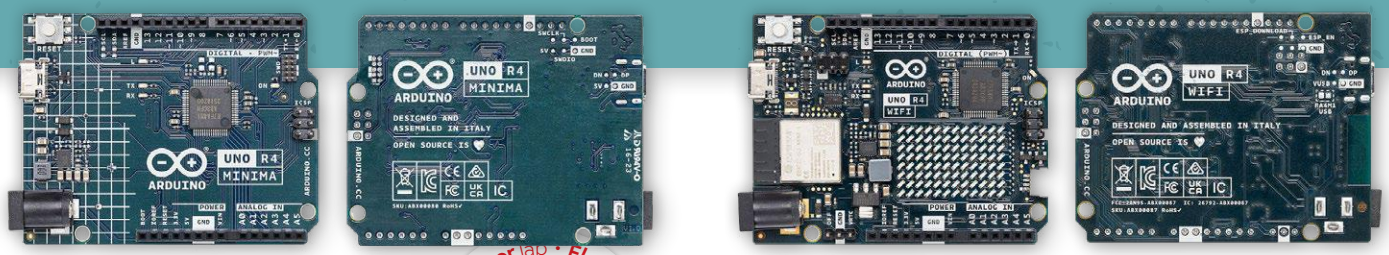
Le LumenPnP Desktop Pick & Place Machine assemble vos composants, vous n'aurez donc plus jamais besoin d'utiliser de pinces

-Radicalement
 Open Source
-Alimentation électrique
-Buse double
-Place les 0402s
-Prix abordable

Opulo.io

Opulo™

les nouvelles cartes Arduino UNO R4 Minima et WiFi





Clemens Valens (Elektor)

La puissante Arduino UNO R4 est le dernier membre de la famille emblématique Arduino UNO. Elle existe même en deux versions. Jetons un coup d'œil à la Minima et à la WiFi.

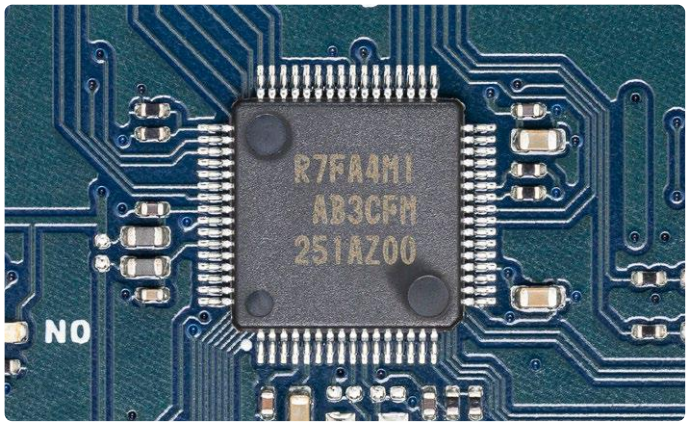


Figure 1. Les cartes UNO R4 possèdent un microcontrôleur R(7F)A4M1 de RENESAS.

Annoncés il y a quelques mois [1], l'Arduino UNO R4 Minima [2] et l'Arduino UNO R4 WiFi [3] sont désormais officiellement disponibles. Maintenant qu'ils existent vraiment, regardons-les de plus près.

La famille Renesas

Il y a quelques mois, Arduino a sorti la carte Portenta C33, dotée d'un microcontrôleur ARM Cortex-M33 de Renesas : le RA6M5. Les deux nouvelles cartes sont également équipées d'un composant de Renesas, le RA4M1 (**figure 1**). Ce microcontrôleur ARM Cortex-M4 32 bits fonctionne à 48 MHz et dispose de 32 Ko de RAM et de 256 Ko de mémoire flash. Il semble qu'une famille de cartes Renesas commence à se développer.

Il est intéressant de noter que le RA4M1 peut fonctionner avec une alimentation allant jusqu'à 5 V, alors que la plupart des autres microcontrôleurs ARM nécessitent 3,3 V. Cela fait de ce microcontrôleur un candidat idéal pour améliorer la famille des AVR 8 bits à 5 V, dont l'Arduino UNO R3 est un membre célèbre.

Plus de périphériques sur l'UNO R4

Le remplacement d'un vieux contrôleur 8 bits à 28 broches par un dispositif moderne 32 bits à 64 broches a, comme on peut s'y attendre, un impact sur la complexité du produit. Cependant, il s'agit principalement de la partie logicielle, comme l'indique la fiche technique de plus de 1400 pages du microcontrôleur (moins de 300 pages pour l'ATmega328). La carte UNO R4 Minima est d'une complexité similaire à celle de la R3. L'UNO R4 WiFi est une carte plus dense car elle utilise l'espace vide de la Minima pour une matrice de LED 8 x 12 et un module ESP32-S3-MINI-1.

La fiche technique est aussi volumineuse parce que le microcontrôleur de Renesas comporte beaucoup plus de périphériques que l'ATmega328 de Microchip. Tous ne sont pas supportés par les R4 car toutes les broches du microcontrôleur ne sont pas accessibles. Seules quelques broches le sont. Celles-ci incluent notamment

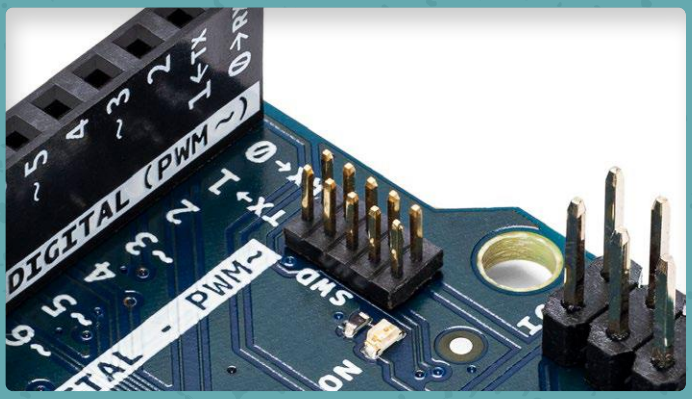


Figure 2. Le connecteur SWD de l'UNO R4 Minima.

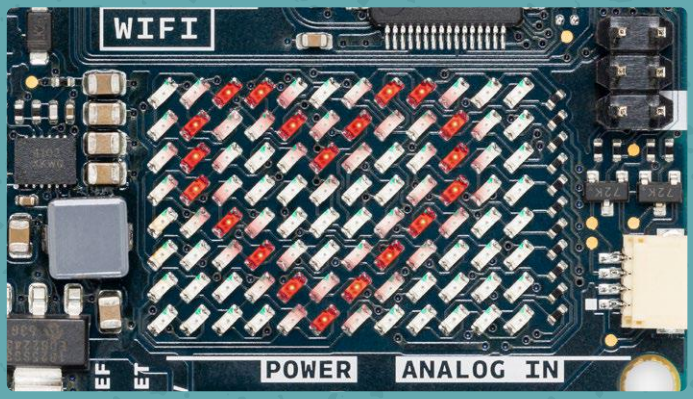


Figure 3. L'Arduino UNO R4 WiFi dispose d'une matrice de LED rouges de 8 par 12.

un bus CAN et l'USB 2.0 Full-Speed (hôte ou périphérique). Un connecteur USB-C remplace le connecteur USB-B.

Serial et Serial1 ?

Le RA4M1 intègre deux ports SPI, deux ports I²C et quatre interfaces de communication série (SCI). Une SCI peut être un UART, un I²C maître ou un simple port SPI (c'est-à-dire jusqu'à six ports I²C ou SPI), et même une interface pour carte à puce. Selon Arduino, ces ports sont disponibles, du moins dans une certaine mesure. Cependant, en regardant le schéma de la Minima, seul un port I²C semble être connecté. Les broches A4, A5, D4 et D5 révèlent un second port SPI alors que les spécifications de la carte n'en mentionnent qu'un seul. Un second port série (Serial1) est également disponible, mais il partage ses broches avec le port SPI.

Il est important de noter que le port `Serial` classique sur l'USB et les broches 0 et 1 a été divisé en `Serial` et `Serial1`. Par défaut, `Serial` fait référence au port USB de l'UNO R4, tandis que les broches 0 et 1 sont destinées à `Serial1`. Il est possible d'affecter `Serial` aux broches 0 et 1 en définissant `NO_USB` avant d'inclure *explicitement Arduino.h*, mais cela déconnecte le port série du port USB. Les programmes et les bibliothèques basés sur l'UNO R3 peuvent poser des problèmes lorsqu'ils sont executés sur l'UNO R4.

L'UNO R4 offre un meilleur fonctionnement analogique

L'UNO R4 dispose d'un convertisseur numérique-analogique (CNA) de 12 bits pour produire de vrais signaux analogiques au lieu de substituts à base de PWM. Il y a aussi un ampli-op et un comparateur avec un DAC interne de 8 bits et le CAN a une étendue de 14 bits au lieu de 10 bits pour l'UNO R3. La bibliothèque *analogWave* a été ajoutée pour faciliter l'utilisation du DAC. La génération d'un signal sinusoïdal, en dents de scie, ou carré est simple, il suffit d'appeler une fonction de la bibliothèque. Bien sûr, vous pouvez faire beaucoup plus avec la carte. Par conséquent, du côté analogique, l'UNO R4 a beaucoup plus à offrir que l'UNO R3.

Complexité logicielle accrue

En ce qui concerne l'EDI Arduino, le passage à une nouvelle famille de processeurs qui n'était pas gérée jusqu'à présent, implique également l'ajout d'un logiciel de prise en charge. Comme nous l'avons appris au fil des ans dans notre monde contrôlé par ordinateur, les nouveaux logiciels ont tendance à introduire des nouvelles problématiques, et il faudra donc probablement un certain temps avant que l'expérience avec l'UNO R4 devienne aussi fluide que celle avec l'UNO R3.

Heureusement, l'UNO R4 Minima facilite la résolution des problèmes car elle dispose d'une interface SWD qui permet un débogage sérieux (en série). L'UNO R4 WiFi va encore plus loin puisque son module ESP32-S3 peut faire office de débogueur CMSIS-DAP integré.

Modem wifi

Maintenant que nous avons parlé de la carte Arduino UNO R4 WiFi, voyons en quoi elle diffère de l'UNO R4 Minima. Tout d'abord, il y a bien sûr le module WiFi & Bluetooth LE - un ESP32-S3 d'Espressif. Il communique avec le microcontrôleur via un port série (Serial2) en mode AT-command. Les autres broches du module WiFi sont accessibles sous forme de minuscules pastilles de soudure. La reprogrammation du module est possible car les broches nécessaires sont accessibles sur un connecteur 2 x 3 (et sur la face inférieure du circuit imprimé). La nouvelle bibliothèque *WiFiS3* fournit un support logiciel de haut niveau pour le module.

Matrice LED

Une matrice LED rouge de 96 pixels (8 × 12) permet aux utilisateurs de tracer des données, de créer des animations et de fournir un retour d'information plus complexe et plus sophistiqué dans leurs projets. Une nouvelle bibliothèque fournit des fonctions permettant d'afficher des animations sur cette matrice. J'avais déjà écrit un petit programme pour afficher un message défilant sur l'UNO R4 WiFi. Vous pouvez le télécharger sur Github [4]. Un outil web pour créer des animations a également été annoncé.

La matrice utilise Charlieplexing [5] pour connecter les 96 LED à seulement 11 ports GPIO (D28 à D38 en notation Arduino). Cela signifie que seules quelques LED peuvent être actives à un moment donné, car les pixels sont constitués de deux LED connectées en antiparallèle et les pixels se partagent les connexions. Cependant, comme l'œil humain est lent, le multiplexage temporel rapide permet de tromper le cerveau en lui faisant voir des images complètes.

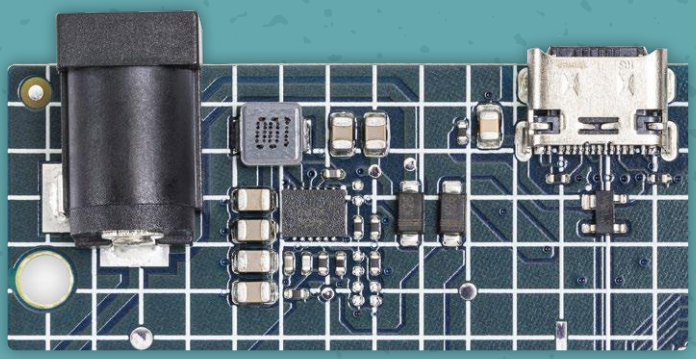


Figure 4. La plage de la tension d'entrée des cartes s'étend de 6 V à 24 V.

Vidéo : Découvrez l'Arduino UNO R4 avec les experts d'Arduino et de Renesas https://youtu.be/vrZ_aJnFd-Q

Un deuxième port I²C

L'UNO R4 WiFi dispose de deux ports I²C (Wire et Wire1). Un connecteur I²C compatible Qwiic (standard SparkFun) permet d'accéder au deuxième port. Ils sont peut-être ennuyeusement petits, mais comme le standard Qwiic impose une tension d'alimentation de 3,3 V, la carte offre un décalage de niveau. Cela signifie que la carte peut effectuer des communications I²C à la fois en 5V et en 3,3 V.

Alimentation

Lorsqu'elle n'est pas alimentée par le port USB, l'Arduino UNO R3 est alimentée par un simple régulateur de tension linéaire. Sur les cartes UNO R4, ce régulateur a été remplacé par un régulateur à découpage. Il offre une plage de tension d'entrée élargie, de 6 V à 24 V (**figure 4**). Le régulateur, également un produit Renesas, peut délivrer jusqu'à 1,2 A avec un rendement d'environ 90 %.
Notez que l'UNO R4 WiFi dispose également d'un connecteur pour alimenter l'horloge en temps réel (RTC).

Conclusion

Les deux cartes Arduino UNO R4, Minima et WiFi, semblent être des successeurs crédibles de l'UNO R3. Elles ont la même forme, les mêmes connecteurs d'extension et les mêmes entrées/sorties de 5 V. L'UNO R4 WiFi est un peu comme une UNO R4 Minima avec un module d'extension intégré. Avec son module WiFi, sa matrice de LED et son connecteur Qwiic, il est facile de réaliser des applications IdO avec la carte UNO R4.
L'UNO R4 est bien plus puissante que l'UNO R3, avec beaucoup de choses en plus, de la mémoire aux périphériques en passant par la vitesse de fonctionnement. J'ai dû installer un package spécial de cartes Renesas dans l'EDI avant de pouvoir la programmer, mais quand vous lisez ces lignes, cela ne devrait plus être nécessaire. L'EDI v2 détecte l'UNO R4 et propose d'installer le logiciel nécessaire. Par conséquent, pour la plupart des utilisateurs, l'UNO R4 remplacera sans problème l'UNO R3.

L'UNO R4 est un grand pas en avant

Enfin, le connecteur de débogage sur la Minima et les capacités de débogage CMSIS-DAP du module ESP32-S3 sur la WiFi, sont des fonctionnalité que de nombreux développeurs attendaient depuis les débuts d'Arduino, sont très intéressants. L'Arduino UNO entre ainsi dans l'arène du développement professionnel. Enfin. L'Arduino UNO R4 peut donc être considéré comme un grand pas en avant.

VF : Laurent Rauber — 230443-04

Questions ou commentaires?

Contactez Elektor (redaction@elektor.fr).

Produits

- **Arduino UNO R4 Minima**
  https://elektor.fr/20527
- **Arduino UNO R4 WiFi**
  https://elektor.fr/20528

LIENS

[1] Clemens Valens, "The Arduino UNO R4 is Coming," elektormagazine.com : https://elektormagazine.com/news/arduino-uno-r4
[2] Arduino UNO R4 Minima : https://elektor.fr/arduino-uno-r4-minima
[3] Arduino UNO R4 WiFi : https://elektor.fr/arduino-uno-r4-wifi
[4] Dépôt GitHub : https://github.com/ClemensAtElektor
[5] Charlieplexing : https://en.wikipedia.org/wiki/Charlieplexing

des potentiomètres logarithmiques

qui sont exponentiels !

Joseph Kreutz (Allemagne)

Idéalement, lorsque vous actionnez le potentiomètre de volume d'un amplificateur ou d'une table de mixage, vous souhaitez qu'une proportionnalité soit établie entre la course ou l'angle de rotation effectif de l'organe de commande et la sensation de niveau sonore que vous ressentez. Un résultat proche de cet idéal est obtenu grâce aux potentiomètres appelés logarithmiques, mais qui sont en réalité exponentiels. Examinons cela de plus près.

La nature nous a conçus pour pouvoir être sensibles à des stimuli dont l'amplitude varie de 1 à 1.000.000 et plus, donc 120 dB ou davantage, pour les signaux acoustiques. La loi dite de Weber-Fechner spécifie

Sensation = k • log(stimulation physique)

Ici, sensation est l'impression que nous percevons et la stimulation physique la pression acoustique au niveau de nos oreilles [1]. Le facteur k est spécifique à chacun de nous, car nos caractéristiques auditives nous sont autant personnelles que notre couleur de cheveux ou celle de nos yeux. La sensibilité auditive dépend de la fréquence, de l'amplitude effective du stimulus audio et de l'âge, parce que notre audition se dégrade à mesure que nous vieillissons.
Il a été mesuré que le seuil d'audition de l'oreille humaine se situe à une pression acoustique de l'ordre de 20 µPa, qui a été fixée comme valeur 0 dB de l'échelle des pressions acoustiques exprimées en décibels. Il est aussi utile d'ajouter que si le niveau acoustique s'approche de 130 dB, un seuil de douleur est atteint. Une exposition régulière à des niveaux dépassant 100 dB peut entraîner des dommages auditifs permanents.

Le potentiomètre de volume idéal

Pour que la sensation soit autant que possible proportionnelle à la position – rotation ou translation – de l'organe de commande d'un potentiomètre, il faut que la loi respectée par le potentiomètre soit complémentaire d'une fonction logarithmique. C'est une fonction exponentielle qui remplira au mieux cette condition. On peut définir un paramètre pour représenter la position d'un potentiomètre :

$$\alpha = \frac{\text{Rotation effective}}{\text{Rotation maximum}} = \frac{\text{Translation effective}}{\text{Translation maximum}} \quad (I)$$

Comme ce réglage a un minimum en principe nul et un maximum, il en découle que $0 \leq \alpha \leq 1$. Le potentiomètre idéal devrait alors avoir une caractéristique répondant à

$$S = S_{ref} \cdot 10^{\alpha \cdot F_p} = 20 \cdot 10^{-6} \cdot 10^{5\alpha}$$

Les potentiomètres de la vie réelle

La construction du potentiomètre idéal, donc exactement exponentiel, serait probablement possible, mais se heurterait à des difficultés techniques considérables qui le rendraient fort onéreux. Les fabricants ont contourné cette difficulté en constituant sa piste de deux ou trois sections de résistivité différentes, pour créer autant de segments de droite qui approximent raisonnablement la fonction exponentielle. Ce compromis s'avère parfaitement satisfaisant pour les applications courantes.

Potentiomètre linéaire chargé par une résistance

Une autre possibilité d'approximer une réponse exponentielle est de charger un potentiomètre linéaire par une résistance installée entre son curseur et son point froid comme l'illustre le schéma de la **figure 1**. La réponse obtenue est caractérisée par la fonction suivante :

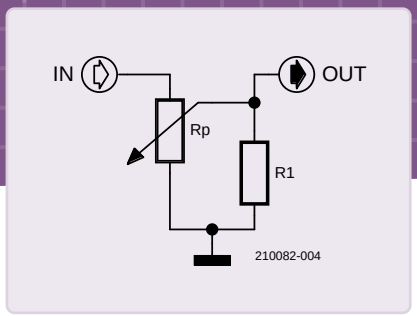


Figure 1. En ajoutant une résistance à un potentiomètre linéaire comme montré ici, il est possible d'approximer une réponse exponentielle.

$V_{OUT} / V_{IN} = \alpha \bullet \gamma / (\alpha \bullet (1-\alpha) + \gamma)$ (II)

Ici, α représente la rotation ou la translation de l'élément de commande comme défini par (II), et γ la proportion existant entre la résistance de charge R1 et la résistance totale du potentiomètre Rp. Par exemple, si Rp = 1 MΩ et R1 = 39 kΩ, alors γ = R1 / Rp = 0,039.
Les courbes de la **figure 2** représentent en vert la loi exponentielle idéale, et en rouge la loi de variation obtenue avec un paramètre γ valant 0,039. Ce choix de γ et des valeurs de résistance correspondantes donne l'un des résultats les plus proches d'une fonction exponentielle. Il existe un écart entre la courbe idéale et cette approximation, qui se manifeste en particulier aux faibles niveaux, et que l'atténuation présente un « saut » petit, mais abrupt à l'origine du diagramme. C'est pourquoi un tel assemblage n'est pas recommandé pour un contrôle de volume d'équipements Hi-Fi [2].
Tenant compte des paramètres définis par les équations I et II, la résistance de charge $R_L(\alpha)$ vue par l'étage qui précède l'assemblage R1 et potentiomètre sera

$R_L(\alpha) = Rp \times \alpha \times (1 - \alpha) + \gamma / (\alpha + \gamma)$

Les valeurs sont respectivement égales à Rp pour α = 0 et Rp//R1 pour α = 1. Cette équation présuppose que l'impédance de l'étage qui suit le contrôle de volume soit suffisamment élevée pour n'avoir qu'un effet négligeable. L'étage qui précède un tel contrôle de volume doit être capable d'alimenter contrôle de volume.
A noter également que la valeur de la résistance totale Rp d'un potentiomètre est généralement définie avec une tolérance de ±20%, qui en fait l'un des composants à la précision des plus exécrables.

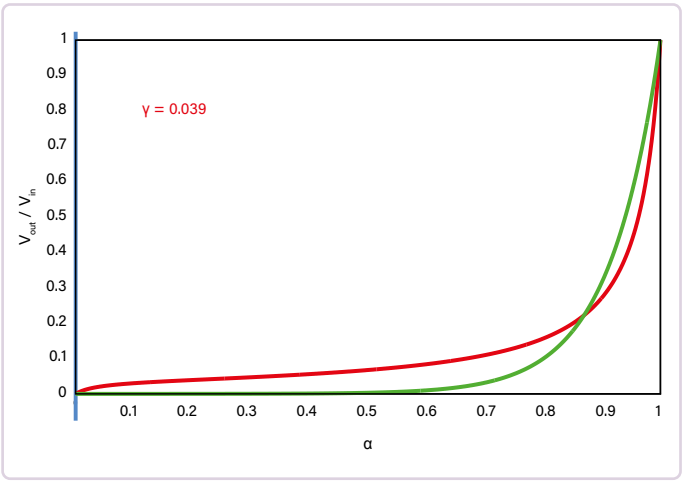


Figure 2. La courbe exponentielle idéale en vert, et en rouge la loi de variation obtenue avec un paramètre γ de 0,039. Ce choix de γ donne l'un des résultats les plus proches d'une fonction exponentielle.

Bruit thermique

Une résistance idéale produit une tension de bruit thermique égale à

$e_n = \sqrt{(4 \bullet k \bullet T \bullet R \bullet B)}$ (III)

avec k = 1,38·10^{-23} (la constante de Boltzmann), T la température en kelvin (ici 298.15°K soit 25°C), R la valeur de résistance en ohm et B la largeur de la bande de fréquences (ici 20 kHz) dans laquelle la mesure est effectuée.
La résistance série introduite par un potentiomètre dans le circuit de la figure 1 vaudra au maximum le quart de sa valeur totale (α•(1−α)•Rp avec α = 0.5), soit 250 kΩ si Rp = 1 MΩ, et la tension de bruit correspondante s'élèvera donc à

$e_n = \sqrt{(4 \bullet 1{,}8 \bullet 10^{-23} \bullet 298{,}15 \bullet 250.000 \bullet 2 \bullet 10^4)} \approx 9{,}07\ \mu V$

La tension de bruit maximum sera atténuée par le diviseur constitué de la résistance série imputable au potentiomètre et de R1 (39 kΩ). Sa contribution se réduira de ce fait à 1,22 µV.
Le bruit généré par la résistance de 39 kΩ doit aussi être considéré. Il s'élève à 3,58 µV pour une température de 25°C et une bande passante de 20 kHz. Sa contribution au bruit total est de 3,1 µV. Ainsi, la tension de bruit thermique maximal est

$e_{n,max} = 3{,}1 \bullet 10^{-6} + 1{,}22 \bullet 10^{-6} = 4{,}32\ \mu V$

Ceci représente une amélioration de 6,4 dB par rapport au potentiomètre seul.
Dans de nombreux produits commerciaux, la piste du potentiomètre est en carbone, ce qui produit un bruit supérieur à celui calculé avec l'équation III. Certains fabricants proposent des potentiomètres à piste plastique ou « cermet », une combinaison d'un métal et d'une céramique. Ils sont moins sensibles à l'usure et présentent généralement une tension de bruit inférieure que ceux en carbone. Il va de soi qu'il conviendra de choisir pour R1 un composant à faible bruit, à film métallique par exemple.
Une conclusion générale et évidente de ce qui précède reste qu'en électronique, lorsqu'il importe d'éviter le bruit thermique, les circuits doivent inclure des résistances de valeur la plus faible possible.

Comparaison des solutions

La difficulté de construire un potentiomètre de caractéristique exponentielle a été contournée par les deux méthodes exposées. Laquelle des deux est la meilleure ?
S'il s'agit d'un contrôle de volume pour un équipement domestique, l'approximation par segments de droite proposée par les fabricants qui offrent des potentiomètres dits « logarithmiques » reste une solution parfaitement acceptable, car aux faibles niveaux l'atténuation reste raisonnablement proche d'une exponentielle idéale et tend

progressivement vers une valeur effectivement nulle : consultez pour cela les courbes publiées par les fabricants.
Pour un contrôle de volume destiné à une table de mixage ou en entrée d'un amplificateur, un potentiomètre linéaire chargé par une résistance donnera toute satisfaction, même si le contrôle aux bas niveaux présente des anomalies. L'association d'une résistance avec un potentiomètre linéaire permet aussi une substantielle réduction du niveau de bruit thermique, avantage non négligeable dans nombre d'applications. Pensez ici aux matériels professionnels.

210082-04

Produits

- **D. Self, *Small Signal Audio Design (2nd Edition)***
  www.elektor.fr/18046
- **Elektor Fortissimo-100 Kit amplificateur de puissance**
  www.elektor.fr/20273

LIENS

[1] Courbe isosonique : https://fr.wikipedia.org/wiki/Courbe_isosonique
[2] D. Self, « Small Signal Audio Design (2nd Edition) », Routledge, 2014 : https://www.elektor.fr/18046



carte d'interface pour **pilote de moteur**

un BoB pour un pilote de moteur à courant continu de 5 A avec une taille de 3×3 mm

Edwin van den Oetelaar (Pays-Bas)

Les makers et les étudiants d'aujourd'hui sont souvent pressés et n'ont pas les compétences nécessaires pour concevoir et souder des circuits électroniques. Ils sont plus intéressés par les *shield* et la manipulation d'un fer à souder est un défi. C'est pourquoi j'ai développé un circuit imprimé pour le pilote de moteur MP6619 de Monolithic Power Systems, qui peut être utilisé dans le cadre de projets basés sur des moteurs.

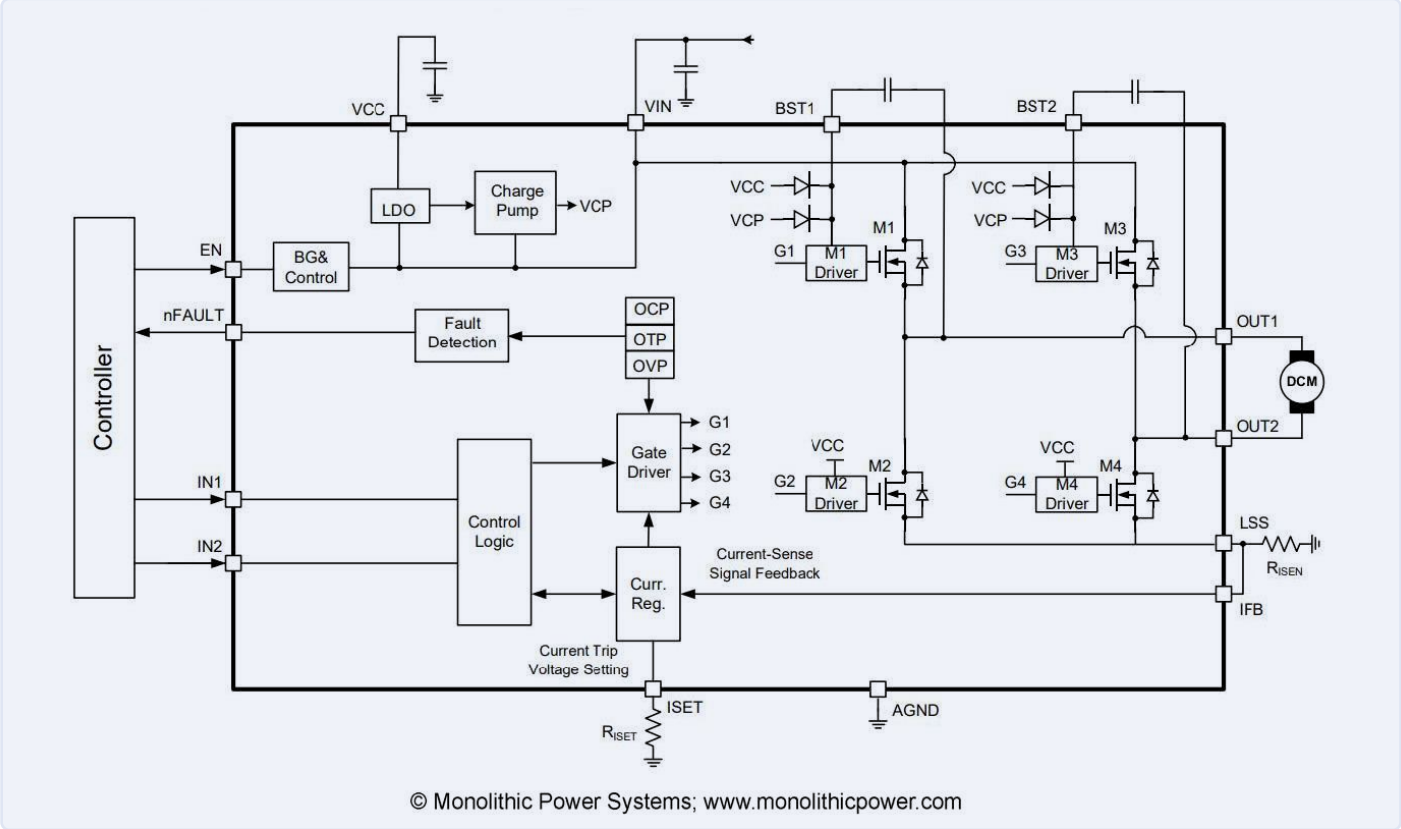


Figure 1. Schéma fonctionnel. (Source : Systèmes de puissance monolithiques [1])

Cet article est un voyage à la découverte du MP6619 *breakout board* (BoB). Cette puce est un nouveau pont en H compact pour contrôler les moteurs électriques CC à balais bon marché et les solénoïdes. Ces moteurs CC sont les plus accessibles et les plus abordables pour les amateurs de petits robots, de modélisme et de projets créatifs. Ceci est une histoire sur le cheminement éducatif que j'ai dû parcourir pour développer et tester cette petite carte.

L'aventure commence

Un jour, j'ai reçu un courriel d'un distributeur de composants à propos d'un tout nouveau composant de pont en H : le MP6619. La **figure 1** montre son schéma fonctionnel interne. Ce minuscule « miracle » de seulement 3×3 mm avec des spécifications impressionnantes telles que 24 V et 5 A a attiré mon attention (voir le cadre **Caractéristiques de la puce**). Comment cela est-il possible ? Autant de courant, pas de dissipateur, et toutes les caractéristiques de protection dans un boîtier aussi minuscule ? Je me suis dit qu'il s'agissait d'un discours marketing. Les alternatives sont facilement dix fois plus grandes.

Ma curiosité a été piquée, et j'ai voulu savoir si c'était vrai et si cela serait utile pour mes projets. J'en ai donc commandé quelques-uns. Deux jours plus tard, ils étaient sur mon bureau : wow, ils étaient si petits ! Seulement 3×3 mm avec 0,2 mm entre les contacts et pas de pattes. Dans quoi me suis-je fourré ?

Je plonge la tête la première dans le sujet. Malgré le faible prix d'environ 2,50 €, les petites dimensions et l'empreinte inhabituelle du MP6619 se sont avérées ne pas être directement applicables à mes projets. Il fallait une carte de liaison (*break-out board* ou BoB) qui rendrait le MP6619 plus maniable et adaptée aux tests et à l'intégration.

Mais il n'y avait pas de BoB disponible pour ce minuscule composant à 19 contacts logé dans un boitier QFN-19. J'ai donc dû développer moi-même une telle carte. Mon aventure avec le pont en H MP6619 a commencé !

Conçu avec KiCad

Pour développer une carte MP6619, j'ai commencé par étudier la fiche technique [1] et dessiner le schéma dans KiCad. J'ai d'abord dessiné le symbole et l'empreinte (qui n'étaient pas disponibles dans la bibliothèque ou dans SnapEDA), puis j'ai attaqué la platine.

J'ai porté une attention particulière à l'emplacement de la résistance de shunt $R_{(i\text{-}sense)}$ et à la connexion des plots d'alimentation. Cette résistance shunt est utilisée pour mesurer le courant des pilotes MOSFET. Cette mesure est cruciale pour le bon fonctionnement de la carte. La chute de tension à travers les traces doit également être prise en compte, lors du routage du circuit imprimé.

Caractéristiques de la puce

- Il ne s'agit pas d'un simple discours marketing : la puce peut en effet supporter 24 V et 5 A. Elle est bien protégée contre les surcharges et les courts-circuits, et se rétablit de manière fiable en cas de défaillance.
- L'ajout d'une bobine d'arrêt ou d'un filtre améliore considérablement les performances. La nécessité d'un tel ajout dépend de l'application et du moteur.
- Les MOSFET à canal N intégrés ont un faible $R_{ds(on)}$, de 65 mΩ, ce qui est essentiel pour limiter la dissipation. Les circuits boost nécessaires pour piloter leurs grilles sont également intégrés. Leurs condensateurs sont connectés en externe.
- La puce MP6619 n'a pas de broches afin d'obtenir une bonne performance thermique dans un boîtier QFN-19.
- Les pastilles de l'empreinte sont très longues dans le but de dissiper la chaleur de la puce vers le circuit imprimé.
- Le circuit imprimé doit être conçu avec soin, en tenant compte à la fois des exigences électriques et des exigences thermiques.

La protection contre les surintensités (OCP, pour l'anglais *Over-Current Protection*) est réglable à l'aide d'une résistance externe. Sans cela, la puce s'éteint lorsque la tension à travers la résistance de shunt atteint 200 mV. La fiche technique suggère un shunt de 0,04 Ω, ce qui limite le courant maximal à 5 A ($V = I \times R$). En ajoutant une $R_{(i\text{-}set)}$ de 80 kΩ, la limite est fixée à 100 mV, ce qui permet de réduire la limite de courant en utilisant la même résistance shunt. D'autres valeurs de résistance sont également possibles. J'ai choisi un shunt de 0,05 Ω pour mon premier test, car je l'avais dans mon tiroir. Il en résulte une limite de courant de sortie de 4 A.

J'ai rapidement réalisé la première version du circuit imprimé du BoB sans étude approfondie, parce que j'avais hâte de commencer les tests, mais je me suis réjoui trop tôt. Le premier BoB comprenait deux puces MP6619, des LED, des résistances, des condensateurs et des connecteurs. À cette époque, je pensais encore que des borniers PCB seraient utiles. Quatre couches de cuivre étaient nécessaires, j'ai placé de nombreux vias, un plan de masse, et j'ai utilisé les traces les plus larges possibles en raison des courants élevés. C'était le travail d'une journée. Une semaine après avoir envoyé les fichiers à un fabricant chinois, j'avais les cartes sur mon établi. La soudure des QFN a été le plus grand défi, car les puces étaient plus petites que tout ce à quoi j'étais habitué.

Pour mettre le QFN en place avec toutes les connexions correctes, il fallait un stéréomicroscope, un bon fer à souder à température contrôlée, un fer à souder à air chaud, beaucoup de flux et encore plus de patience.

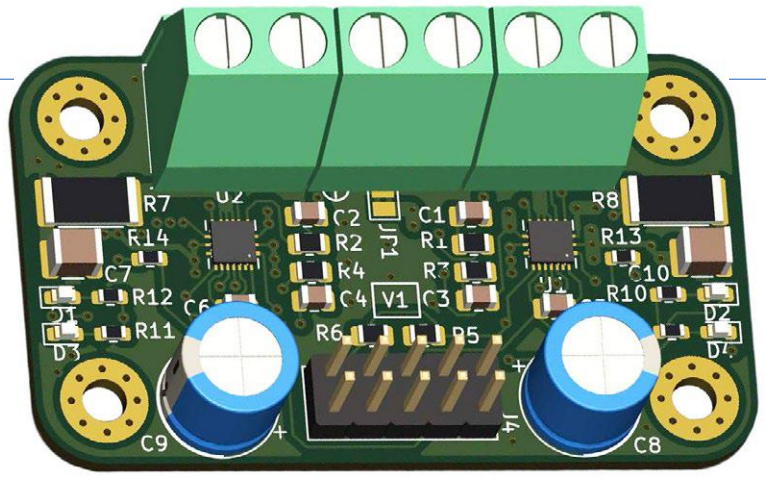


Figure 2. Modèle 3D du premier circuit imprimé.

Une fois le QFN en place et la sueur essuyée du front, les autres composants CMS « normaux » sont devenus un jeu d'enfant, ils semblent gigantesques par rapport au QFN. La plage de soudure pour le bornier du circuit imprimé est plus grand que la puce QFN entière (**figure 2**) ! J'ai posté cette première version sur le site d'Elektor Labs [2].

Premiers essais

Les premiers essais ont été réalisés avec un petit moteur à courant continu [3] qui se trouvait sur mon établi. Il s'agissait d'un modèle simple de 12 V avec un courant de décrochage inférieur à 5 A (**figure 3**). Le MP6619 est constitué en interne de deux demi-ponts, chacun ayant une entrée (IN1, IN2). La puce elle-même assure la synchronisation correcte des MOSFET côté haut et côté bas afin qu'aucun court-circuit ne se produise entre les rails d'alimentation.
La puce est protégée contre tous les problèmes typiques. Ses fonctions intégrées UVLO (verrouillage en cas de sous-tension), OVP (protection contre les surtensions), OCP (protection contre les courants trop forts), et OTP (protection contre les températures trop élevées) devraient gérer en toute sécurité presque toutes les situations d'erreur, ce que nous allons voir !

Connexion de IN1, IN2 et EN

Les entrées sont équipées de résistances de rappel vers le bas. Dès que l'entrée EN reçoit un haut niveau, la puce est activée, et le régulateur de tension interne commence à fonctionner. À ce moment, la sortie VCC du MP6619 fournit une tension de 5 V. J'ai utilisé un petit condensateur de découplage de 100 nF pour stabiliser cette tension. Si les 5 V sont présents, les MOSFET à canal N deviennent également actifs. En fonction des niveaux d'entrée à IN1 et IN2, les MOSFET côté haut ou côté bas sont commutés.
Les entrées supportent des niveaux logiques de 3,3 V à 5 V. Il n'y a donc pas de problème pour connecter le BoB à un ESP32 ou à un Arduino. Pour mes premiers tests, j'ai utilisé des interrupteurs pour appliquer 3,3 V aux entrées, et j'ai connecté le moteur aux sorties du BoB.



Figure 3. Les premiers tests ont été effectués avec un motoréducteur planétaire à courant continu. (Source : E-S Motor / RobotShop [3])

Succès partiel

Mon moteur ne voulait pas tourner, la sortie *nFault* est immédiatement passée au niveau bas (la LED d'indication d'erreur s'est allumée), mais je n'avais pas de court-circuit à la sortie. Une réduction significative de la tension d'entrée m'a finalement permis de faire tourner lentement le moteur dans les deux sens. Il est important de noter que je n'avais pas encore connecté un signal PWM pour contrôler la vitesse.
J'ai expérimenté en diminuant la résistance du shunt et en augmentant la résistance R_{set} pour faire tourner le moteur, ce qui a finalement abouti à une erreur humaine : 12 V sur IN1, avec un BoB brûlé en conséquence, et un profond soupir de frustration. Le circuit imprimé est parti en fumée, et j'ai dû garnir un nouveau BoB avec des composants.

Un autre BoB et d'autres tests

Finalement, j'ai découvert que le moteur est équipé de condensateurs de suppression de bruit connectés en parallèle au collecteur. Par conséquent, le moteur se comporte de manière moins inductive et plus capacitive. L'OCP se déclenche donc immédiatement en cas de pic de courant dans le moteur. Cette « surcharge » provoque un délai de 1 ms avant que le MP6619 n'essaie à nouveau. Ce processus se répète encore et encore, ce qui entraîne un son faible et grinçant du moteur.
La suppression de ces condensateurs aide partiellement à contrôler le moteur à des tensions un peu plus élevées. Pendant ce temps, après avoir étudié la fiche technique plus attentivement, j'avais également des nouvelles pistes pour améliorer ma carte BoB et j'ai décidé de créer une nouvelle version.

Nouvelle conception et disposition

Pour ma deuxième version, la conception et la disposition ont dû être ajustées. J'ai apporté les modifications suivantes :

- Une seule puce MP6619.
- La position du shunt a été revue.
- Un en-tête à broches de 0,1" à la place des borniers ; pas de trous de fixation.

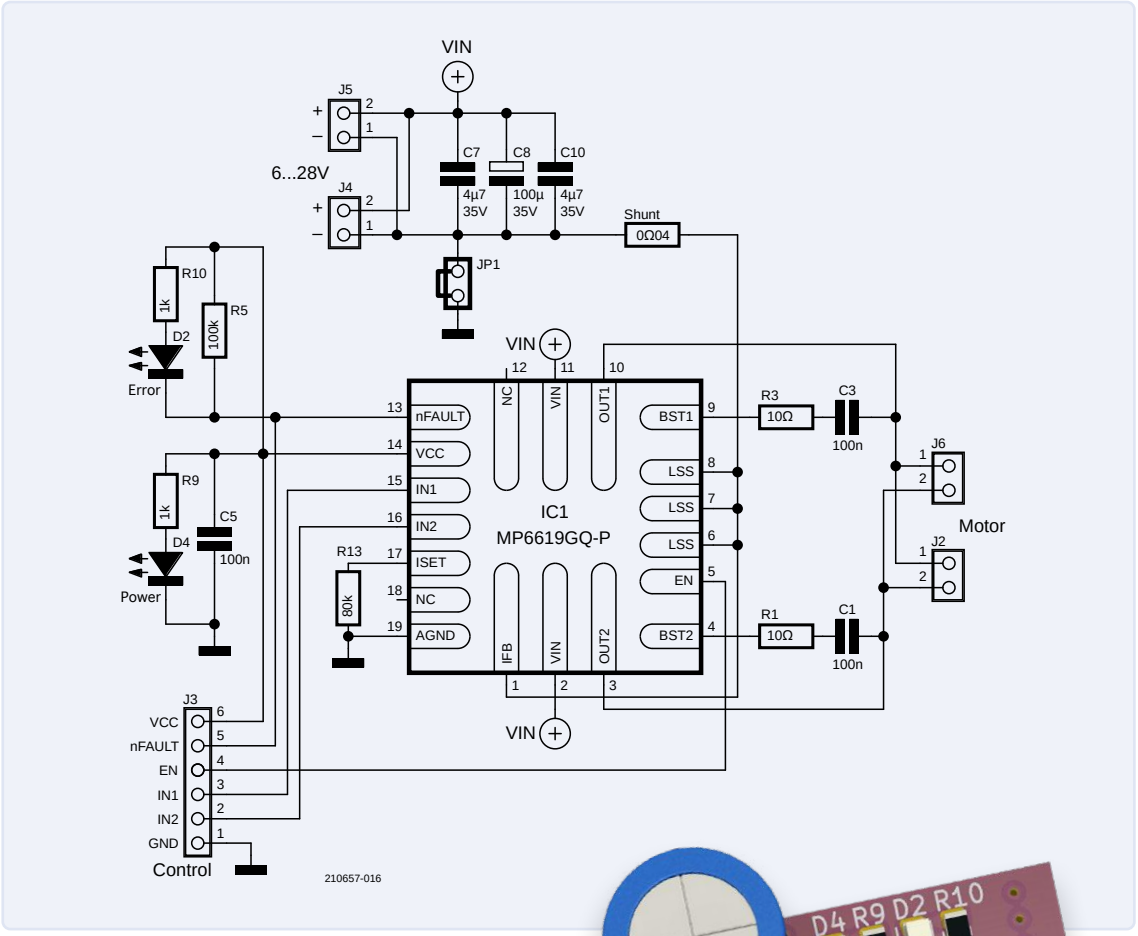


Figure 4. Les schémas révisés du BoB.

› Une sérigraphie claire ; seulement le nombre minimal de composants.

Cela a donné un nouveau schéma (**figure 4**) et par la suite aussi un nouveau PCB (**figures 5** et **6**) avec quatre couches et des dimensions d'environ 25 mm × 25 mm, inseré dans un panneau de production avec une finition ENIG (nickel chimique immersion or). Après avoir câblé le BoB, j'ai obtenu un module vraiment compact, qui, je crois, peut être contrôlé par presque tous les microcontrôleurs.

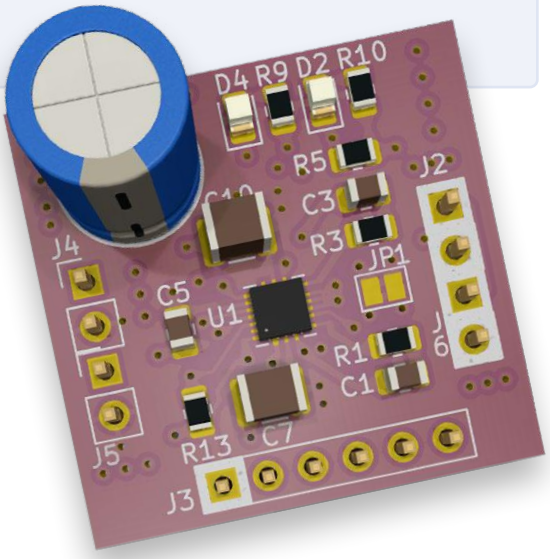


Figure 5. Modèle 3D du BoB de 25×25 mm (face supérieure) constitué de quatre couches et d'une finition ENIG.

Autres tests et expériences

J'ai utilisé le même moteur que pour le premier test. L'OVP et l'ULVO se sont bien comportés. Maintenant, je l'ai essayé avec le contrôle PWM pour réaliser le contrôle de la vitesse. Les tests avec des fréquences de 1 kHz à 50 kHz ont donné le comportement attendu : le moteur était bien contrôlable. L'exemple de la fiche technique utilise 20 kHz, et cela a bien fonctionné à basse tension.

Mais, à des tensions plus élevées, l'OCP s'est à nouveau manifesté et le moteur s'est arrêté. En ajoutant une inductance supplémentaire de 220 µH (comme celle de la **figure 7**) en série avec le moteur, le comportement du contrôle a été significativement amélioré, et la plage de vitesse a été augmentée.

En conséquence, le moteur a fourni plus de puissance et l'OCP s'est déclenché moins souvent. Avec ces inductances supplémentaires, la solution semble pouvoir être utilisée dans des projets.

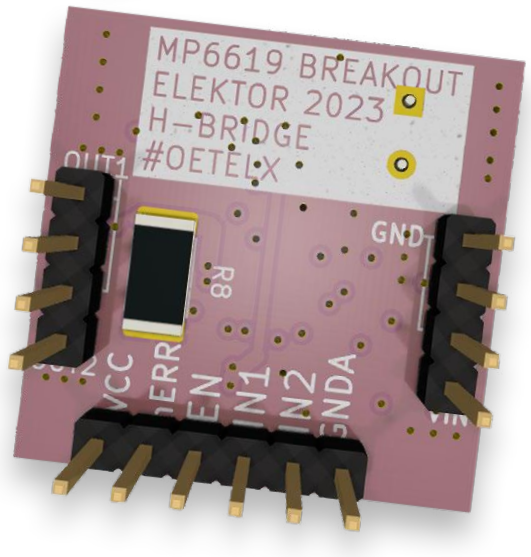


Figure 6. Modèle 3D du BoB (face inférieure) avec des têtes de broches.



Figure 7. Une inductance à noyau toroïdal en série avec le moteur améliore le comportement du BoB.

Résistances
(CMS 0805)
R1, R3 = 10 Ω
R5 = 100 kΩ
R8 = 0Ω04, 2 W, CMS 2512
R9, R10 = 1 kΩ
R13 = 80 kΩ

Condensateurs
C1, C3, C5 = 100 nF, 50 V, CMS 0805
C7, C10 = 4,7 µF, 35 V, CMS 1210
C8 = 100 µF, 35 V, électrolytique, ø 8 mm

Semi-conducteurs
D2 = LED, rouge, CMS 0806
D4 = LED, verte, CMS 0805
IC1 = MP6619GQ-P, QFN-19

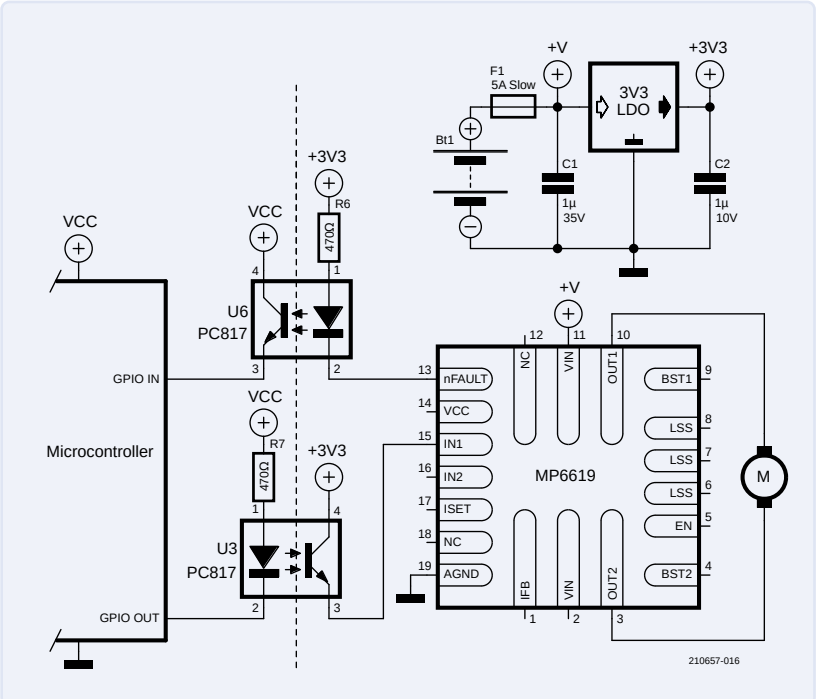
Divers
J2, J4, J5, J6 = Connecteur à 2 broches, 1/10″ (2,54 mm)
J3 = Connecteur à 6 broches, 1/10″ (2,54 mm)
Platine BoB (voir texte)

Dans certaines conditions, le moteur restitue de l'énergie qui retourne dans le pont en H. Cette énergie est déversée sur le circuit de puissance. Elle est déversée sur les rails d'alimentation à travers les diodes des MOSFETs, car le moteur fonctionne également comme un générateur (EMF d'un moteur à courant continu). Dans mes tests, j'ai utilisé un condensateur électrolytique C8 à faible ESR en parallèle à la source d'alimentation. Avec cette dernière amélioration, le BoB était stable.

Optocoupleurs

La **figure 8** montre mon circuit de test. J'ai décidé d'utiliser des optocoupleurs pour fournir une isolation galvanique entre le côté commande et les pilotes de moteur, parce que l'alimentation d'un moteur comporte généralement des niveaux de bruit considérables. Les interférences électromagnétiques peuvent également poser des problèmes. L'isolation réduit le risque d'interférence dans les signaux basse tension et empêche les fortes impulsions générées par le moteur et son pilote d'atteindre le microprocesseur sensible et d'autres composants électroniques.

Figure 8. Ce circuit avec des optocoupleurs placés entre le microcontrôleur et le BoB a été utilisé pour les tests.



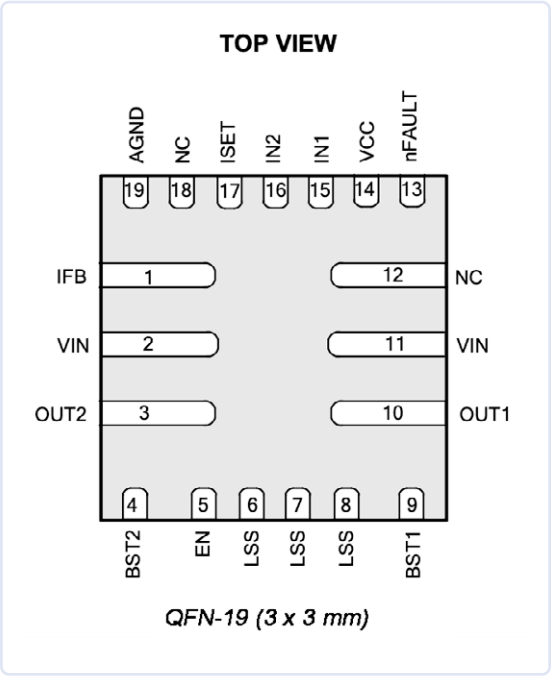
Les LED d'entrée des optocoupleurs peuvent être pilotées directement à partir d'une broche du microprocesseur à l'aide d'une résistance de 470 Ω en série. Côté sortie, l'émetteur du phototransistor est connecté à l'IN1 du BoB, et donc à l'entrée du MP6619. Son collecteur est connecté à une tension logique fournie par un régulateur de tension LDO de 3,3 V alimenté par l'alimentation du moteur, dans mon cas, une batterie. C'est suffisant pour assurer un fonctionnement fiable et stable.

La Figure 8 montre également comment on peut amener le signal *nFault* du pont en H (sortie à drain ouvert) vers le microcontrôleur via un optocoupleur de la même manière que dans l'autre sens. Dans ce cas, le port du microcontrôleur connecté aura besoin d'une résistance de rappel vers le bas.

Boîtiers QFN

Les boîtiers FCQFN (**figure 9**) peuvent avoir des pastilles de forme irrégulière, souvent disposées en bandes longues et étroites. Contrairement aux boîtiers QFN ordinaires, la chaleur est dissipée par plusieurs de ces pastilles au lieu d'un grand plot central. Cela pose quelques défis pour la conception des circuits imprimés, car il y a de nombreux contacts, tous avec des signaux différents, qui doivent être connectés avec des zones de cuivre.

De petits vias peuvent être placés à l'intérieur des zones de plots (via dans PAD). Sur les circuits imprimés multicouches dotés de plans d'alimentation et de masse, les vias peuvent directement relier ces pastilles aux plans. Dans d'autres cas, le cuivre doit être



◀ *Figure 9. Brochage du boîtier QFN-19 du MP6619. (Source : Monolithic Power Systems [1])*

directement fixé aux plages pour dissiper la chaleur du circuit intégré vers de plus grandes surfaces de cuivre. Un guide d'implantation est disponible à l'adresse [4].
Le plus grand obstacle que j'ai rencontré lors ce projet a été la soudure de la puce MP6619. Pour surmonter ce problème, j'ai expérimenté diverses techniques. J'ai finalement réussi avec beaucoup de flux, de soudure au plomb et d'air chaud, mais c'est resté un défi de placer la puce avec précision.
Une fois en place, la tension superficielle de la soudure fondue garantit que la puce est correctement alignée sur les plots. Beaucoup d'entraînement s'est avéré nécessaire. Ma conclusion est d'acheter une plaque chauffante pour simplifier le processus de soudure.

Épilogue

Dans cet article, j'ai partagé mes expériences dans la conception et la construction d'une carte d'interface pour le pilote de moteur à pont en H MP6619. J'ai expliqué l'importance d'utiliser des optocoupleurs pour fournir une isolation électrique. J'ai aussi partagé mes idées sur le travail avec des boîtiers QFN à puce retournée et les défis impliqués dans le soudage de petits composants.
Le circuit imprimé MP6619 offre une solution très compacte pour piloter des moteurs CC à balais bon marché, ce qui en fait un excellent choix pour les amateurs et les makers qui construisent de petits robots et d'autres projets créatifs. Tous les fichiers de conception et de production sont open-source et peuvent être consultés sur ma page GitHub [5]. Le MP6619 est largement disponible en petites quantités auprès de distributeurs tels que Farnell, Mouser, et directement auprès de Monolithic Power Systems. ◀

VF : Maxime Valens — 210657-04

À propos de l'auteur

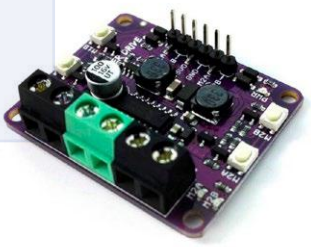
Edwin van den Oetelaar est ingénieur chez Fontys ICT, aux Pays-Bas. Il est membre du groupe de recherche sur les logiciels embarqués de haute technologie, dirigé par le professeur Teade Punter. Spécialiste du développement de matériel et de logiciels, Edwin se réjouit d'avoir l'occasion de s'attaquer à divers projets techniques. Ses compétences couvrent un large éventail de domaines, tels que les flux de données, les appareils intelligents, les soins de santé, l'IdO, le traitement d'images et l'IA appliquée.

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

Produits connexes

- **Dogan Ibrahim, *Motor Control – Projects with Arduino & Raspberry Pi*, Elektor 2017**
  https://elektor.fr/18322
- **Cytron Maker Drive – Commande de moteur à pont en H**
  https://www.elektor.fr/cytron-maker-drive-h-bridge-motor-driver
- **Peter Dalmaris, *KiCad 6 Like a Pro — Fundamentals and Projects* (E-book), Elektor 2022**
  https://elektor.fr/20159



LIENS

[1] Fiche technique MP6619 @ Monolithic Power Systems : https://tinyurl.com/452pk4sv
[2] Première version sur Elektor Labs : https://tinyurl.com/299prw8n
[3] Motoréducteur planétaire à courant continu : https://tinyurl.com/23nxnr4k
[4] Lignes directrices pour l'implantation du circuit imprimé du circuit d'attaque du moteur - Partie 2 : https://tinyurl.com/2p8ry9hd
[5] GitHub de l'auteur : https://github.com/edwin-oetelaar



sur le vif

danger de l'électronique

Source : Shutterstock / wk1003mike

Ilse Joostens (Belgique)

Si vous achetez régulièrement des composants auprès de grands fournisseurs d'électronique étrangers, il y a de fortes chances que, tôt ou tard, vous tombiez sur un emballage portant le fameux avertissement « California Proposition 65 ». Dans mon cas, il s'agissait de connecteurs pour circuits imprimés, et on m'a prévenu qu'au contact de ces connecteurs, je risquais de développer un cancer ou d'avoir des problèmes de fertilité. Apparemment, l'électronique est devenue le nouveau tabagisme. Je ne prends pas ces avertissements trop au sérieux, mais, malgré tout, si vous travaillez dans le domaine de l'électronique, vous risquez d'être exposé à des substances dangereuses - et je ne parle pas des soudures au plomb, des fumées de soudure, des flux, des décapants, des solvants, des vernis ou des produits chimiques dont la composition n'est pas connue.

Quelque chose de plus

Nous avons tous entendu dire que les circuits intégrés (CI) sont fabriqués à partir de sable. Le sable est essentiellement du dioxyde de silicium. De nos jours, le silicium est le matériau semi-conducteur le plus couramment utilisé. Toutefois, dans la pratique, les CI ont quelque chose en plus. Les CI, mais aussi de nombreux autres composants – semi-conducteurs ou non – ont un côté obscur et contiennent parfois des substances chimiques très étranges.

L'arsenic, autrefois utilisé pour la fabrication d'un joli papier peint vert à l'époque victorienne, est aujourd'hui utilisé en grandes quantités dans l'industrie électronique pour le dopage du silicium et la production d'arséniure de gallium, un autre matériau semi-conducteur important pour les circuits intégrés et certains transistors. Les PFAS et PFOS [1], désormais très répandus, sont utilisés dans les processus d'exposition et de gravure des plaquettes de silicium. Les résidus de ces substances subsistent parfois dans le produit fini. Les emballages des composants contiennent non seulement de l'époxy, mais aussi des retardateurs de flamme sous la forme de composés organiques de brome et de chlore en combinaison avec du trioxyde d'antimoine [2]. Ces substances ne sont pas très toxiques, mais elles ont encore une réputation douteuse et des efforts sont faits pour limiter leur utilisation. Enfin, il y a le béryllium – une véritable horreur devant laquelle l'arsenic semble inoffensif. Autrefois, un silicate de zinc et de béryllium entrait dans la composition du phosphore blanc des lampes fluorescentes, jusqu'à ce que de nombreux ouvriers souffraient de béryllose, une maladie pulmonaire chronique très grave [3]. Aujourd'hui, l'oxyde de béryllium est encore utilisé, entre autres, comme matériau céramique thermoconducteur dans les transistors de puissance à haute fréquence. Mais avant de vous inquiéter, il faut savoir que les contacts en cuivre au béryllium (souvent utilisés comme contacts à ressort) sont relativement inoffensifs, à condition de ne pas les limer, les meuler ou les poncer.

Si vous avez dépassé un certain âge, vous vous souvenez très probablement des redresseurs au sélénium (**figure 1**) sous la forme d'un empilement de fines plaques carrées reliées entre elles au centre. Il n'y avait pas de problème, tant qu'ils fonctionnaient bien, mais l'odeur d'un redresseur au sélénium en surchauffe est quelque chose que l'on n'oublie jamais. Le séléniure d'hydrogène libéré dans cette situation n'a pas seulement une odeur

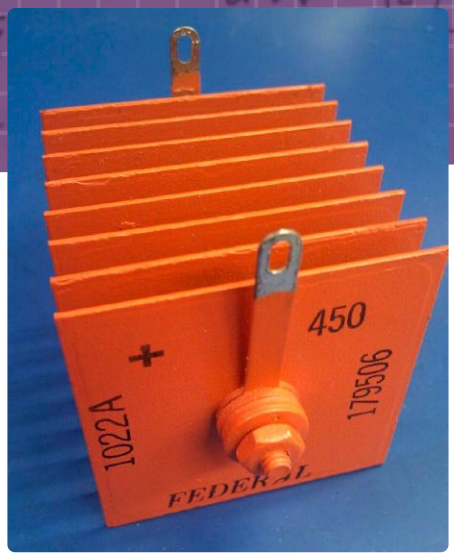


Figure 1. Un redresseur au sélénium Source : Wikimedia / Binarysequence / https://creativecommons.org/licenses/by-sa/3.0.

désagréable, il est aussi très toxique [4]. Personne n'en est jamais décédé – une chance inouïe.

Une autre substance de la même catégorie est le cadmium, surtout connu dans les piles rechargeables au nickel-cadmium, mais également présent dans les LDR, les panneaux solaires au tellurure de cadmium, les capteurs, les conducteurs transparents, parfois dans les résistances CMS, et même dans l'isolation des fils en PVC. En d'autres termes, il est presque partout.

Le mercure, autrefois utilisé dans les piles, les interrupteurs à mercure, les tubes Nixie, les lampes à faible consommation d'énergie, les lampes fluorescentes, les lampes à mercure et les redresseurs à vapeur de mercure, est une substance dangereuse connue. Les lampes fluorescentes sont toujours commercialisées et le fameux *Ali* vous aidera à vous procurer des interrupteurs à mercure. Les anciens redresseurs à vapeur de mercure sont particulièrement fascinants, voire un peu bizarres, avec leur lueur bleue effrayante (**figure 2**) [5]. Si vous cassez l'un d'entre eux, pensez à appeler l'équipe HazMat.

Enfin, les luminophores des lampes à LED d'aujourd'hui contiennent une multitude d'éléments exotiques, y compris les métaux des terres rares comme l'yttrium, le cérium et l'europium. Bien que ces éléments ne semblent pas très toxiques d'après leurs fiches techniques, l'absence de données toxicologiques approfondies est préoccupante.

Patriotisme

Outre les substances toxiques susmentionnées, les anciens composants, en particulier, contiennent parfois des substances radioactives. Par exemple, certains filaments de tubes sont recouverts de matériaux tels que l'oxyde de thorium ou constitués d'un alliage thorium-tungstène – y compris le filament du magnétron de votre four à micro-ondes. On trouve de l'uranium sous forme de dioxyde d'uranium dans les anciennes résistances CTP Urdox (**figure 3**) et sous forme de verre d'uranium dans certains tubes. Et puis, il y a toute une liste de tubes à vide radioactifs, de tubes radar, de parasurtenseurs et d'éclateurs remplis de gaz [6]. Presque tout ce que vous pouvez imaginer – lampes fluorescentes, interrupteurs de lampes fluorescentes, lampes à hydrure métallique, et même les tubes Nixie – peut contenir des matériaux radioactifs.

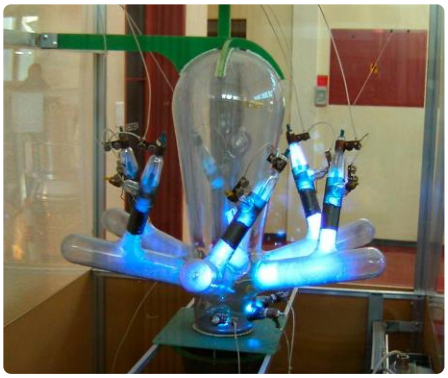


Figure 2. Un redresseur à vapeur de mercure. Source : Wikipedia / Timberwind / Public Domain.



Figure 3. Une résistance PTC Urdox. Photo : Ilse Joostens.

Les capteurs industriels tels que les jauges d'épaisseur, les jauges de concentration ou les capteurs destinés à mesurer l'humidité des matériaux contiennent parfois des substances radioactives. Il y a peu de chances que vous entriez en contact avec ces appareils, mais ce n'est pas le cas des détecteurs de fumée à ionisation (aujourd'hui interdits). Au début, certains modèles contenaient du radium 226, mais les appareils plus récents contiennent de l'américium 241. Cela me rappelle l'homme qui a accidentellement fumé la source radioactive de ce type de détecteur de fumée dans un vaporisateur de cannabis. Il ne s'est pas drogué, mais, selon l'histoire, il a ressenti un goût étrange [7].

L'ex-Union soviétique disposait également de ses propres détecteurs de fumée, mais pour des raisons patriotiques, l'américium 241 n'était pas autorisé, et ils ont donc utilisé du plutonium créé par les réacteurs à la place, et en grande quantité - jusqu'à 18,5 MBq dans le détecteur de fumée KI-1 [8]. À l'époque, les composants électroniques n'étaient pas très sensibles, une source puissante était donc nécessaire. Il est tout de même étrange de se rendre compte qu'une telle chose était suspendue au-dessus, de nos têtes, dans de nombreux bâtiments gouvernementaux.

Après ce que vous venez de lire, j'espère que vous parviendrez à passer une bonne nuit de sommeil. En pratique, la situation n'est pas si mauvaise tant que vous ne faites pas de folies. Il y a cependant un problème pour les travailleurs qui recyclent les déchets électroniques, alors espérons pour des produits électroniques plus durables.

230411-04

LIENS

[1] PFAS: chips' poisonous ingredient that doesn't go away: https://euractiv.com/section/digital/news/pfas-chips-poisonous-ingredient-that-doesnt-go-away/
[2] Wikipedia: Brominated flame retardant: https://en.wikipedia.org/wiki/Brominated_flame_retardant
[3] Environews Focus: "Beryllium: A Chronic Problem": https://ehp.niehs.nih.gov/doi/pdf/10.1289/ehp.94102526
[4] YouTube: BigCliveDotCom: Selenium rectifiers - the smelliest components ever: https://youtu.be/OOA1NaoKV6I
[5] YouTube: Kempton Steam Museum: The Mercury Arc Rectifiers: https://youtu.be/YhaQqgXrMMU
[6] YouTube: AE Laboratories: Radioactive Vacuum Tubes: https://youtu.be/m-6-aJv7GCI
[7] YouTube: That Chemist: Somebody Vaped a Smoke Detector: https://youtu.be/14UggnpN39w?t=627
[8] Special Nuclear Material: Analysis of Soviet smoke detector plutonium: https://carlwillis.wordpress.com/2017/02/07/analysis-of-soviet-smoke-detector-plutonium/

une architecture cellulaire **offre-t-elle la plus basse consommation** pour l'IdO ?

caractéristiques énergétiques des technologies LTE-M et NB-IoT pour le déploiement de réseaux (LPWAN)

Stuart Cording (Elektor)

Pour vous lancer dans votre prochain projet pour l'Internet des objets (IdO), il pourrait être utile de jeter un coup d'œil à l'offre LPWAN (réseaux étendus à basse consommation) cellulaire. Si l'environnement LoRaWAN semble sur le papier peu gourmand en énergie, les chercheurs ont constaté des variations importantes en termes d'autonomie de batterie pour les déploiements en conditions réelles. Les technologies LTE-M et NB-IoT sont l'une et l'autre très compétitives en termes de bilan énergétique et offrent toute une série d'autres avantages. Pour autant, comme tout ce qui apporte du positif, elles ont des problématiques propres à résoudre

La capacité des réseaux cellulaires à permettre un accès mondial pour se connecter m'est apparue clairement au début des années 2000. J'étais en déplacement professionnel dans les Alpes suisses lorsqu'un collègue m'a appelé pour me demander une présentation dont il avait besoin dans le cadre d'une réunion avec un client. Muni de mon fidèle Ericsson T68i en équilibre sur le tableau de bord de mon véhicule, un téléphone compatible Bluetooth et un ordinateur portable entre les genoux, j'ai rapidement entrepris d'envoyer le fichier grâce au fulgurant débit de 115 kbps offert par la norme GPRS. Même si j'ai dû m'y reprendre à plusieurs reprises en raison des pertes de connexion dans les tunnels, la possibilité d'être relié au monde entier alors que je me trouvais au milieu de nulle part témoignait de l'avènement de l'ère de la connectivité sans fil illimitée.

L'IdO avant l'IdO

Grâce au Bluetooth, il était également possible d'accéder au téléphone par l'intermédiaire d'une fenêtre de terminal, à l'instar des anciens modems câblés remplacés ultérieurement par la technologie DSL. Ainsi, il suffisait d'un microcontrôleur compatible Bluetooth et d'un logiciel capable de gérer les commandes AT pour se connecter partout dans le monde. Nous étions en train de mettre en œuvre l'Internet des objets, mais nous ne le savions pas à l'époque. Aujourd'hui, 20 ans plus tard, l'IdO est en place et nous disposons de réseaux cellulaires encore plus performants. Nous serions pourtant probablement bien en peine de citer un produit ou une application qui ne soit pas un smartphone ou une tablette utilisant la technologie cellulaire pour la connectivité d'accès aux données.

L'une des problématiques concernant l'adoption réside sans doute dans la confusion relative à la nomenclature de l'écosystème de l'IdO cellulaire dans son ensemble. D'où la difficulté de savoir quoi choisir et pourquoi. Par-dessus tout, nous sommes maintenant habitués à la transition entre 4G et 5G, mais, pour les consommateurs et les entreprises, ces termes de marketing ne concernent que les smartphones et la connectivité de données à haut débit. Pour les applications à plus faible débit pour la communication entre machines, d'autres normes spécifiques sont à prendre en considération.

IdO cellulaire à débit plus élevé

La première technologie est appelée LTE-M, ce qui signifie *Long-Term Evolution Machine Type Communication*. Elle se subdivise actuellement en deux normes, LTE Cat M1 et LTE Cat M2. Le 3GPP définit ces normes, les nouvelles capacités étant ratifiées au travers de « versions ». Ainsi, la norme LTE Cat M1 faisait partie de la version 13 en 2015, et la norme LTE Cat M2 de la version 14 en 2017. La norme Cat M1 offre un débit de 1 Mbps en liaison montante et descendante, alors que la norme Cat M2 offre un débit d'environ 7 Mbps en liaison montante et de 4 Mbps en liaison descendante. L'une et l'autre prennent en charge le full-duplex et le half-duplex (**figure 1**). Pour situer le contexte, les réseaux de smartphones 5G ont un débit moyen d'environ 100 Mbps [1].

L'avantage d'un débit de données plus faible réside en partie dans la réduction de la consommation d'énergie du matériel LTE-M. Conformément à la spécification 3GPP, l'objectif était d'atteindre dix ans de fonctionnement avec une batterie de 5 Wh. Cependant, bien que cette durée de vie soit réalisable, Brian Ray [2], actuellement ingénieur chez Google, note que l'obtention d'une puissance d'émission de 23 dBm en liaison montante, le niveau le plus élevé pris en charge, entraîne des courants de pointe d'environ 500 mA. Ce problème de conception est loin d'être négligeable.

La couverture est également meilleure que celle de la technologie LTE standard grâce à une perte de couplage maximale (MCL) plus élevée. Cette valeur définit le point à partir duquel un système sans fil perd sa capacité à assurer le service prévu. Une étude réalisée par Sierra Wireless [2] a révélé une MCL atteignant 164 dB pour le LTE Cat-M1, ce qui représente une amélioration significative par rapport aux 142 dB de l'ancienne technologie LTE et une nette amélioration

	LTE-M		NB-IoT	
	LTE Cat M1	LTE Cat M2	LTE Cat NB1	LTE Cat NB2
3GPP Release	Release 13	Release 14	Release 13	Release 14
Peak Uplink Rate	1 Mb/s	7 Mb/s	66 kb/s	160 kb/s
Peak Downlink Rate	1 Mb/s	4 Mb/s	26 kb/s	127 kb/s
Voice over LTE	yes	yes	no	no
Duplex	full / half	full / half	half	half
Latency	<15 ms	<15 ms	<10 s	<10 s

Figure 1. La technologie LTE-M est adaptée aux applications d'IdO mobile avec des débits de données plus élevés et le transport voix-messages VoLTE, alors que la NB-IoT cible les applications statiques.

par rapport à l'objectif de 155,7 dB que s'est fixé le 3GPP. En termes de capacité, cela signifie une meilleure connectivité à l'intérieur des bâtiments, ce qui est important pour les applications de métrologie intelligente où le bâtiment peut contribuer à une perte de pénétration atteignant 50 dB, et une meilleure portée à l'extérieur.
Comme les smartphones, la technologie LTE-M prend également en charge les données mobiles, ce qui signifie que votre appareil se connecte en permanence à la station cellulaire la plus proche, ce qui est idéal pour les capteurs chargés de surveiller des denrées périssables en mouvement ou de suivre des flottes de véhicules. En outre, si vous avez occasionnellement besoin de messages vocaux dans le cadre du système, par exemple pour un panneau d'alarme incendie ou un système de surveillance pour personnes âgées, la technologie vocale Voice over LTE (VoLTE) est incluse. Enfin, avec une latence inférieure à 15 ms, un réseau LTE-M peut prendre en charge une application réactive pour un être humain.

Architecture cellulaire pour les nœuds IdO statiques

Pour les réseaux cellulaires, la technologie IdO alternative est NB-IoT. Elle se décline également en deux versions. La norme LTE Cat NB1 a été officialisée dans la version 13, tandis que la norme LTE Cat NB2 existe depuis la version 14. La technologie NB-IoT cible les applications non mobiles, comme les compteurs intelligents dans l'agriculture, les stations météorologiques ou les déploiements de capteurs dans les stations d'épuration, car elle ne prend pas en charge le transfert d'une station de téléphonie cellulaire ou la technologie voix VoLTE (**figure 2**). Au lieu de cela, il y a un pic de consommation d'énergie lorsque l'appareil s'enregistre auprès de la station la plus proche, après quoi le module sans fil peut entrer en mode veille en sachant qu'à son réveil, il peut reprendre là où il s'était arrêté.
Les débits de données NB-IoT sont bien inférieurs à ceux de LTE-M. Le débit LTE Cat NB1 atteint 26 kbps en liaison descendante et jusqu'à 66 kbps en liaison montante, alors qu'avec la version LTE Cat NB2, le débit atteint 127 kbps en liaison descendante et environ 160 kbps en liaison montante. Contrairement à la norme LTE-M, la technologie NB-IoT ne prend en charge que le mode half-duplex. La latence est également beaucoup plus élevée, avec, en général, une valeur comprise entre 1,6 et 10 secondes. Toutefois, compte tenu des applications prévues, il n'y aura aucun problème, après avoir reçu des données de capteur, pour répondre par une nouvelle position d'actionneur sur une fenêtre de serre ou de vanne de traitement de l'eau. À noter également que cette latence est bien meilleure que celle des technologies concurrentes de réseaux étendus à basse consommation comme LoRaWAN [4] et Sigfox [5].

Mettre en œuvre des applications IdO basse consommation grâce à la technologie cellulaire

L'alimentation est l'une des principales exigences d'une application IdO, car elle s'appuie principalement sur les batteries ou d'autres sources d'énergies renouvelables, comme les panneaux solaires. Ces deux technologies sans fil offrent différents modes basse consommation qui permettent d'accroître la durée de vie des batteries. Le premier est le mode d'économie d'énergie (PSM). Il permet à l'application de placer le module radio cellulaire dans un état de sommeil profond, donc avec une consommation d'énergie ne dépassant pas le plus souvent quelques microampères. L'appareil notifie son intention à la tour cellulaire et peut alors rester en sommeil jusqu'à 413 jours durant. Pendant cette période, il n'y a aucun moyen de transmettre des données à l'appareil. Toutefois, lorsqu'il se réveille, il n'est pas nécessaire de s'enregistrer auprès de la tour de téléphonie mobile.
Nous avons ensuite le mode eDRX. Ce mode de sommeil plus léger permet d'économiser de l'énergie pendant 40 minutes avec la norme LTE-M ou jusqu'à trois heures avec le protocole NB-IoT. Le réveil est également plus rapide qu'en mode PSM. Malgré ces leviers d'économies d'énergie, les capacités techniques ne sont pas toujours disponibles partout. Leur configuration dépend de l'équipement du fournisseur de services [6], ce qui signifie que l'autonomie de la batterie dans un pays peut être inférieure par rapport à un autre parce qu'il n'est pas possible de négocier un réglage. L'intervention d'un fournisseur de services proposant une carte SIM itinérante peut également limiter l'accès à ces fonctions de basse consommation [7].
Étant donné le très vaste éventail de configurations et les innombrables facteurs sur lesquels repose la mise en œuvre de l'IdO cellulaire, rien d'étonnant à ce que la recherche de conseils relatifs à la consommation d'énergie n'aboutisse pas. Le plus souvent, une recherche sur Google mène à des pages affirmant « une autonomie de 10 ans », apparemment à la fois pour les deux normes, LTE-M et NB-IoT, et ce, sans détails sur la capacité de la batterie.

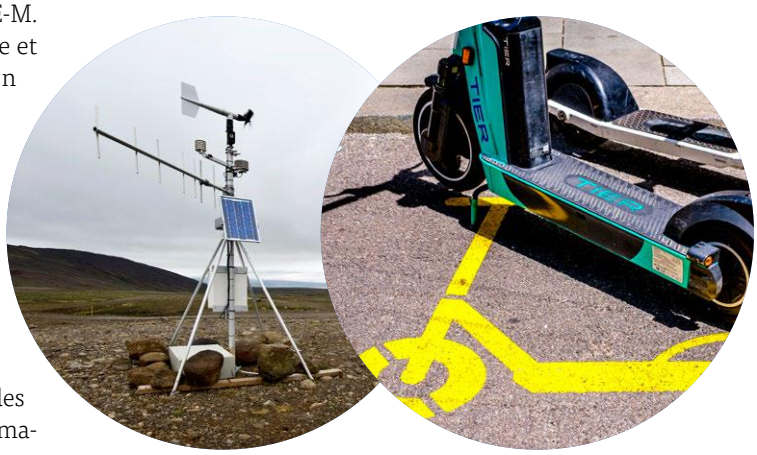


Figure 2. Les applications mobiles, comme le partage de trottinettes électriques, fonctionnent mieux avec la technologie LTE-M. Par contre, une application statique, une station météorologique par exemple, bénéficie de la faible consommation d'énergie de la technologie NB-IoT. (Source : Shutterstock)

IdO cellulaire en labo

Fort heureusement, les équipes de différents instituts ont pris le temps d'étudier la consommation d'énergie. Les résultats indiquent ce à quoi il faut s'attendre et montrent comment l'IdO cellulaire se comporte par rapport aux autres solutions. Par exemple, le document cité en [8] compare les technologies LoRa et NB-IoT. Cette expérience comprend l'envoi de paquets MQTT contenant chacun 50 octets de données. Grâce à une configuration optimale et de bonnes conditions de connexion, la technologie NB-IoT nécessite environ 200 mJ par transaction.

La technologie LoRa, en revanche, permet au développeur de mieux contrôler la configuration de la transmission en définissant un facteur d'étalement (SF). Avec une valeur SF7, le débit binaire est plus élevé, ce qui signifie moins de temps d'émission, alors qu'avec une valeur SF12, le réglage le plus bas, le temps d'émission est maximal pour le transfert du même nombre d'octets. Le risque, cependant, est que la valeur SF7 réduise trop la portée pour un transfert réussi, signifiant ainsi que l'échange de données doit être répété. Lors des essais, le facteur SF7 n'a nécessité que 100 mJ par transaction. En revanche, l'augmenter jusqu'à la valeur SF12 a nécessité 250 mJ.

En conclusion, pour une batterie de 3.000 mAh, la technologie LoRa avec facteur SF7 pourrait alimenter cette configuration pendant plus de 32 ans. En revanche, passer à la valeur SF12 réduirait cette durée à un peu moins de 13 ans. À titre de comparaison, la mise en œuvre de la norme NB-IoT permettrait d'atteindre une durée de fonctionnement d'un peu moins de 20 ans. Étant donné qu'en conditions réelles, LoRa devrait adapter la configuration de son facteur d'étalement et d'autres éléments de configuration des émetteurs-récepteurs (bande passante) pour assurer des transferts de données réussis, une telle variation de la durée de vie de la batterie pourrait être considérée comme trop risquée.

Comparatif avec la technologie LoRaWAN

Ce risque a également été mis en évidence dans les recherches d'une équipe de l'université d'Anvers [9]. Dans leur conclusion, ils indiquent également que, si la technologie LoRaWAN a consommé un minimum d'énergie dans des conditions de laboratoire maîtrisées, pour une durée de vie de plusieurs années, « avec le déploiement en temps réel [sic], ce délai s'est réduit à quelques mois ». Les chercheurs ont également testé la technologie NB-IoT avec Sigfox et DASH7 [10]. Bien que DASH7 offre une consommation d'énergie encore meilleure que LoRaWAN dans les mêmes conditions, l'équipe a supposé que NB-IoT pourrait être la meilleure option malgré une consommation d'énergie légèrement plus élevée. La norme NB-IoT peut tout simplement cocher plus de cases lorsqu'on prend en compte tout ce qu'une application IdO exige, notamment la disponibilité, la latence, la couverture, la sécurité, la robustesse et le débit.

Au cours des deux dernières décennies, la technologie sans fil s'est démocratisée grâce aux émetteurs-récepteurs radio CMOS, aux modules radio hautement intégrés et aux antennes minuscules. Même les ensembles logiciels sont souvent disponibles gratuitement. Cependant, les concepteurs ne travaillent souvent que sur le nœud terminal, s'en remettant à d'autres pour l'infrastructure à laquelle ils se connecteront. Malgré l'omniprésence des réseaux cellulaires LTE et leur facilité d'utilisation pour les utilisateurs de smartphones, il n'en va pas de même pour ceux qui s'y intéressent concernant l'IdO.

S'armer de courage

Pour les non-initiés, il est difficile de trouver des indications fiables sur ce que l'IdO cellulaire peut faire ou pas. De même, concernant l'accès partout dans le monde à d'importantes fonctions d'économie d'énergie ou de quelle manière déterminer leur disponibilité. De ce fait, l'IdO cellulaire apparaît comme le parent pauvre de l'industrie des smartphones. L'annonce récente de la volonté de Vodafone de vendre son activité centrée sur les services IdO ne contribue pas à améliorer cette image. Bien que cet acteur majeur ait vendu 150 millions de connexions SIM IdO l'an passé [11], la division concernée ne représente que 2 % de son chiffre d'affaires.

Bien que les études montrent que l'IdO cellulaire est compétitif en termes de consommation d'énergie et d'autonomie par rapport aux autres solutions de réseaux étendus à basse consommation, il présente encore des faiblesses. L'infrastructure de réseau de certains fournisseurs et de certains pays ne parvenant pas à mettre en œuvre de manière universelle la prise en charge des économies d'énergie, les ingénieurs ne seront manifestement pas enclins à s'engager sur des valeurs d'autonomie de batterie attrayantes pour les clients. Ce qui est clair, c'est que chaque approche de réseau étendu à basse consommation présente des inconvénients et que les équipes de conception doivent les évaluer selon la situation concernée. En outre, il semble bien qu'une bonne partie du temps de recherche soit également nécessaire, dans la mesure du possible, pour comparer les unes avec les autres les solutions de réseau étendu.

VF : Pascal Godart — 230376-01

LIENS

[1] « 5G vs 4G: What's the difference? », EE Limited, septembre 2020: https://bit.ly/3MTuUYd
[2] B. Ray, « What is LTE-M?0», Medium, mai 2017: https://bit.ly/45QDq2U
[3] G. Vos et autres, « Coverage Analysis of LTE-M Category-M1 », Sierra Wireless, janvier 2017: https://bit.ly/3OYpoGG
[4] « What are LoRa and LoRaWAN? », The Things Network: https://bit.ly/43EgYc5
[5] Site web de Sigfox: https://sigfox.com/
[6] « PSM and eDRX : Power saving in cellular LPWAN - possibilities and limitations », 1NCE GmbH: https://bit.ly/43p6a10
[7] P. Marshall, « Sleeping Battery: How eDRX and PSM Can Save Energy in LPWA IoT Edge Devices », Eseye, mai 2021: https://bit.ly/3Ne1WUh
[8] L. Tan, « Comparison of LoRa and NBIoT in Terms of Power Consumption », KTH Royal Institute of Technology, janvier 2020: https://bit.ly/43mXrwl
[9] R. K. Singh et autres, « Energy Consumption Analysis of LPWAN Technologies and Lifetime Estimation for IoT Application », Sensors (Bâle, Suisse), août 2020: https://bit.ly/3qvfmm7
[10] Site web de DASH7 Alliance: https://www.dash7-alliance.org/
[11] M. Kleinman, « Vodafone dials up sale of stake in £1bn Internet of Things unit », Sky UK, mai 2023: https://bit.ly/3MQZUbt



communication sans fil dans les systèmes IoT

à l'aide de modules Arduino MKR

normes de communication des modules Arduino pour l'IoT

Transfer Multisort Elektronik Sp. z o.o.

In this article, we present a brief overview of selected Arduino development kits from the MKR family for rapid prototyping of IoT devices using wireless communication in standards such as WiFi/Bluetooth, LoRaWAN/Sigfox, GSM/3G, or NB-IoT.

L'un des plus gros problèmes auxquels le marché des appareils Internet des objets [1] (IoT) est aujourd'hui confronté est sa forte fragmentation. La multitude d'appareils et de protocoles de communication rend très difficile la construction d'un système uniforme et fonctionnel si nous décidons d'utiliser des composants de différents fabricants. Les raisons de cette division sont multiples et ne sont pas toujours liées uniquement au désir des concepteurs de forcer leurs propres solutions sous licence. Le terme IoT couvre de nombreux types d'appareils. Il peut s'agir par exemple de petits capteurs de mesure alimentés par des sources d'énergie alternatives et nécessitant l'échange d'une petite quantité de données sur de longues distances, mais aussi de caméras distantes transmettant des images haute résolution en temps réel. Ainsi, la spécificité du dispositif conçu oblige les concepteurs à sélectionner la technologie de communication sans fil appropriée, adaptée aux exigences du dispositif conçu. Vous devez prendre en compte, entre autres la durée de vie de la batterie, la portée de communication ou la quantité de données transférées. Répondant aux besoins du marché, les producteurs de kits de développement (y compris les plates-formes Arduino [2]) ont veillé à ce que leur portefeuille couvre aussi complètement que possible les besoins des constructeurs d'appareils IoT qui utilisent la communication sans fil dans des normes telles que WiFi / Bluetooth, LoRaWAN / Sigfox, GSM/3G ou NB-IoT.

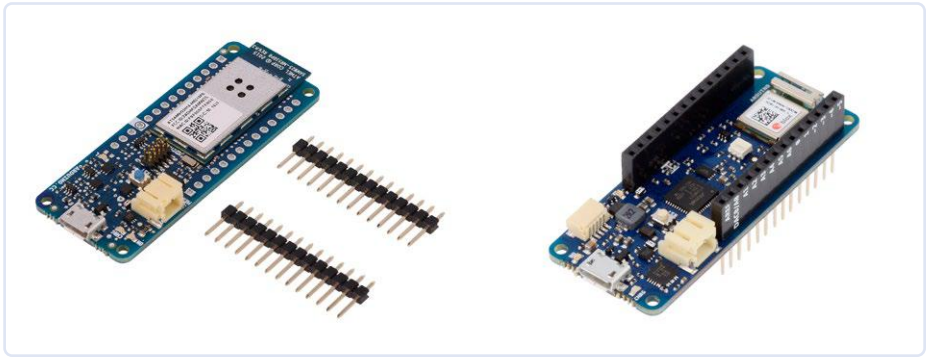
Communication WiFi/Bluetooth avec Arduino MKR 1000/1010

La communication dans la bande de fréquence ISM 2,4 GHz qui utilise les normes WiFi et Bluetooth, opère sur le marché des appareils IoT depuis plusieurs années. La société Arduino a développé des kits de développement Arduino MKR WiFi 1000 [3] et MKR WiFi 1010 [4] pour les besoins de mise en œuvre rapide de prototypes matériels/logiciels utilisant la communication WiFi. Le premier des ensembles mentionnés est basé sur le module ATSAMW25 [5], qui comprend le microcontrôleur SAMD21, le chemin radio WINC1500 [6] et le circuit d'autorisation ECC508 [7]. L'ensemble dans la version MKR 1010 est équipé du module radio

u-blox [8] NINA-W102 qui permet la communication Bluetooth/BLE.
En ce qui concerne le logiciel, la société Arduino fournit la bibliothèque WiFi101 pour les modules MKR WiFi 1000 en prenant en charge le cryptage WEP et WPA2 Personal. Pour le module MKR WiFi 1010 (et d'autres ensembles basés sur le module u-blox NINA-W102, y compris Arduino NANO 33 IoT [9]), le fabricant a préparé la bibliothèque WiFiNINA, ainsi qu'un certain nombre d'exemples d'applications présentant l'intégration avec Android IoT Cloud et Azure, AWS IoT Core, Google Firebase ou Blynk.

Communication LoRaWAN et Sigfox – Modules Arduino MKR WAN 13x0 et FOX 1200

Le développement dynamique des systèmes IoT a entraîné un intérêt accru pour le sujet des villes intelligentes. Malheureusement, la connectivité utilisant les normes WiFi / Bluetooth / BLE est de nature locale et ne répond pas à toutes les exigences fixées pour les projets du groupe « Smart City » (parmi lesquels il y a entre autres, des réseaux étendus de capteurs de pollution, de surveillance du niveau d'eau ou d'occupation des parkings). La solution aux problèmes peut être l'utilisation de l'une des deux normes de communication les plus populaires dans le domaine du LPWAN (Low Power Wide Area Network) – LoRaWAN ou Sigfox ce qui permet de transmettre une petite quantité de données sur de longues distances. Aux fins du prototypage rapide des appareils utilisant la communication LoRa / LoRaWAN, les concepteurs d'Arduino ont préparé des kits de développement MKR WAN 1300 [10], et son successeur MKR WAN 1310 [11]. Les deux modules sont basés sur le microcontrôleur Atmel SAMD21, utilisé dans d'autres modules de la série Arduino MKR, ainsi que sur le module radio Murata CMWX1ZZABZ. La nouvelle version du module a en outre été équipée d'une mémoire Flash de 2 Mo, d'un nouveau système de charge de batterie et – optimisé pour une faible consommation d'énergie – de systèmes d'alimentation.
Les modules MKR WAN 13x0 fonctionnent avec l'Arduino IoT Cloud fourni par le fabricant. Des solutions complètes sont encore complétées par une porte d'accès Arduino Pro Gateway LoRa Connectivity [12] optimisée pour les modules MKR WAN 1310.
La norme Sigfox, qui met particulièrement l'accent sur la communication des nœuds vers la passerelle d'accès est une alternative intéressante pour la communication LoRa / LoRaWAN. Les concepteurs ont obtenu à leur disposition de l'offre d'Arduino le module MKR FOX 1200 [13], construit sur la base du microcontrôleur Atmel SAMD21. Le système Microchip Smart RF ATA8520 est responsable de la communication radio, dont le trajet radio a été réglé sur la fréquence ISM 868 MHz applicable en Europe.



Figures 1 et 2. Modules Arduino MKR WAN 1000 (à gauche) et MKR WAN 1010 (à droite).

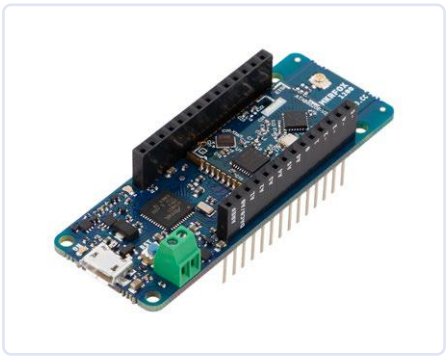
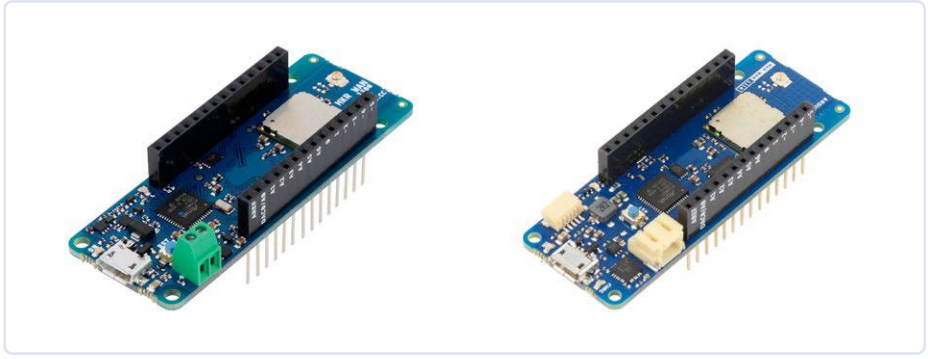


Figure 5. Module Arduino MKR FOX 1200 pour la communication dans le réseau Sigfox.



Figures 3 et 4. Modules Arduino MKR WAN 1300 (à gauche) et MKR WAN 1310 (à droite).



Figure 6. Module Arduino MKR GSM 1400 pour communication en réseau GSM/3G.

Communication GSM/3G - Module Arduino MKR GSM 1400

Même un réseau maillé étendu (Mesh), fonctionnant selon la norme LoRa / LoRaWAN, n'est actuellement pas en mesure de fournir une couverture mondiale. Dans le cas de projets IoT nécessitant une communication de zone quasi illimitée, la meilleure solution est d'utiliser la norme GSM/3G. Pour la communication GSM/3G, Arduino a préparé un module MKR GSM 1400 [14], équipé d'un modem SARA-U210 d'u-blox et d'un système d'autorisation Microchip ECC508 pour mettre en œuvre des mécanismes de sécurité de communication. Le modem GSM intégré offre une couverture de communication dans les bandes GSM 850 MHz, E-GSM 1900 MHz, DCS 1800 MHz et PCS 1900 MHz.

Pour améliorer le processus de préparation du logiciel, le fabricant fournit la bibliothèque MKRGSM (qui libère le programmeur de l'utilisation du module à l'aide de commandes AT de bas niveau), ainsi qu'un riche ensemble d'exemples (incluant, entre autres : connectivité GPRS, réception/transmission de messages texte, gestion des appels vocaux). Le module MKR GSM 1400 peut fonctionner à la fois avec le logiciel Arduino IoT Cloud, ainsi qu'avec des solutions cloud alternatives : Google IoT Cloud, Blynk ou SORACOM Air IoT, pour lesquelles le fabricant a préparé un ensemble d'exemples d'implémentation.

Communication dans le réseau IoT à bande étroite – module Arduino MKR NB 1500 :

En préparant une brève description des normes de communication sélectionnées au sein des appareils de l'Internet des objets, il est impossible d'ignorer les solutions basées sur la norme IoT à bande étroite (NB-IoT), qui utilisent la bande LTE 800 MHz sous licence pour la communication. Comme les solutions LoRaWAN et Sigfox, NB-IoT fait partie du groupe de réseaux LPWAN, il assure donc une communication stable dans de vastes zones, en utilisant des modules radio à économie d'énergie, garantissant de nombreuses années de fonctionnement de la batterie. Ainsi, c'est une autre alternative à la communication LoRaWAN et Sigfox dans les solutions du segment « Smart City ».

Pour le prototypage rapide des nœuds d'extrémité fonctionnant dans la norme NB-IoT, Arduino a préparé un ensemble MKR NB 1500 [15], équipé du module u-blox SARA-R410M-02B [16], permettant LTE Cat M1/ Connectivité NB1 en bandes 1, 2, 3, 4, 5, 8, 12, 13, 18, 19, 20, 25, 26 et 28. De plus, le poste



Figure 7. Module Arduino MKR NB 15 pour la communication dans les réseaux Narrowband IoT.

MKR NB 1500 a été équipé de l'authentification ECC508 système par Microchip [17], connecteur de carte MicroSIM, contrôleur de charge de batterie Li-Po et connecteur d'antenne externe.

230468-04

Transfer Multisort Elektronik Sp. z o.o.
Headquarters
ul. Ustronna 41
93-350 Łódź, Poland
export@tme.eu
www.tme.eu/en

LIENS

[1] Internet des objets : https://www.tme.eu/fr/katalog/systemes-embedded-et-iot_113611/
[2] Arduino : https://tinyurl.com/4yr8kpdd
[3] MKR1000 WIFI : https://tinyurl.com/2s3rup9c
[4] MKR WIFI 1010 : https://tinyurl.com/baa45y9d
[5] Module ATSAMW25 : https://tinyurl.com/3yefjwrz
[6] Chemin radio WINC1500 : https://tinyurl.com/2fhfcsuz
[7] Circuit d'autorisation ECC508 : https://tinyurl.com/fwejfbkd
[8] u-blox : https://www.tme.eu/fr/linecard/p,u-blox_1320/
[9] Arduino NANO 33 IoT : https://tinyurl.com/3jppdy9k
[10] MKR WAN 1300 : https://tinyurl.com/bd7yny9y
[11] MKR WAN 1310 : https://tinyurl.com/5esbxn7p
[12] Arduino Pro Gateway LoRa Connectivity : https://tinyurl.com/22vub6f6
[13] MKR FOX 1200 : https://tinyurl.com/ydac7ce8
[14] MKR GSM 1400 : https://tinyurl.com/mr3dx7vr
[15] MKR NB 1500 : https://tinyurl.com/58vexer6
[16] Module SARA-R410M-02B : https://tinyurl.com/bdcnz7fh
[17] Microchip Technology : https://www.tme.eu/fr/linecard/p,microchip-technology_632/

pertes en courant alternatif dans les composants magnétiques

évitez la surchauffe des inductances !



George Slama (Würth Elektronik)

L'un des nombreux problèmes auxquels est confronté un concepteur d'alimentations électriques est la surchauffe inattendue d'un composant magnétique qu'il a pourtant sélectionné avec le plus grand soin. Ce comportement inexpliqué peut le laisser perplexe et contrarié et augmenter la pression sur l'achèvement de son projet dans les délais et le budget impartis. L'estimation précise des pertes en courant alternatif est un sujet complexe qui n'est pas à la portée de tous. Heureusement, il existe une solution facile à utiliser.

Une multitude d'appareils, malgré leur apparence simple, possèdent une alimentation dont la conception est un projet complexe, avec de nombreuses exigences contradictoires imposées par l'utilisateur final et une multitude d'organismes de réglementation. L'utilisateur final veut la solution la moins chère, souvent le plus petit volume, avec la plus grande fiabilité. Les organismes de réglementation exigent le respect de normes d'isolation et de distances de sécurité, ce qui se traduit par des composants plus volumineux. La compétence du concepteur doit donc s'étendre à de nombreux domaines au-delà de l'électricité, la mécanique et la théorie du contrôle. Si l'on ajoute à cela le mystère du magnétisme, où des forces invisibles agissent sur les matériaux pour stocker ou transformer l'énergie selon d'anciennes formules cryptiques, il n'est pas étonnant que le concepteur cherche une solution plus simple pour choisir le bon composant magnétique.

Prenons l'exemple du régulateur buck, le circuit le plus utilisé pour obtenir une basse tension dans les circuits non isolés. La réduction de la taille est obtenue en augmentant la fréquence de hachage. Cela permet aux composants passifs tels que les inductances et les condensateurs d'être plus petits car le stockage d'énergie par cycle est plus faible. Leur taille est proportionnelle à leur capacité de stockage d'énergie. Cependant, la réduction de la taille réduit la surface disponible pour dissiper la chaleur des pertes, ce qui rend la conception thermique plus critique.

Dans les dispositifs magnétiques, il existe deux sources de pertes : le noyau et les enroulements. Toutes deux peuvent être divisées en sous-types (**tableau 1**).

Qu'est-ce que les pertes fer ?

L'origine des pertes par hystérésis dans le noyau réside dans le mouvement des dipôles magnétiques et, à des courants de saturation élevés, dans le déplacement des limites des domaines dans le matériau du noyau. Lorsqu'un matériau magnétique doux est soumis à un champ magnétique externe, proche ou causé par un courant dans une bobine, les dipôles magnétiques dans les domaines (minuscules régions magnétiques avec des aimants élémentaires dans le matériau) s'alignent sur le champ. Cela demande de l'énergie et du temps. Lorsque l'influence extérieure disparaît, les dipôles magnétiques des domaines se réorientent, les limites des domaines se replient, mais pas complètement. Lorsque le sens du courant change, les pôles magnétiques des domaines s'inversent, mais pas complètement. L'énergie est restituée au système, mais le reste est dépensé sous forme de travail de frottement contre d'autres domaines et est converti en chaleur. Plus la fréquence est élevée, plus les aimants élémentaires des domaines et des zones de domaines sont déplacés par seconde, et l'énergie dépensée croît ainsi de manière exponentielle. Le mouvement des dipôles magnétiques est également proportionnel à la variation du flux. Une plus grande variation du flux signifie plus de mouvement, donc plus d'énergie, dont la totalité n'est pas récupérée. La zone située à l'intérieur de la courbe BH représente la perte d'énergie au cours d'un cycle.

Les pertes par courants de Foucault proviennent du fait que, lorsque des courants alternatifs circulent dans un conducteur, une tension est induite, selon la loi de Faraday, proportionnellement à la vitesse de variation du champ magnétique. Le noyau lui-même se comporte comme un enroulement, et bien que le matériau ferrite utilisé dans la technologie des hautes fréquences ait une résistivité élevée, les petites particules sont conductrices. De plus, cette « résistance dynamique » diminue de manière exponentielle à mesure que la température augmente. Des temps de montée plus courts induisent des tensions plus importantes. Des impulsions de tension plus élevée provoquent des pertes exponentielles plus importantes selon

$$P = (U^2 \times D) / R$$

(où D est le rapport cyclique et R, la résistance). En fonction de la topologie, un troisième facteur

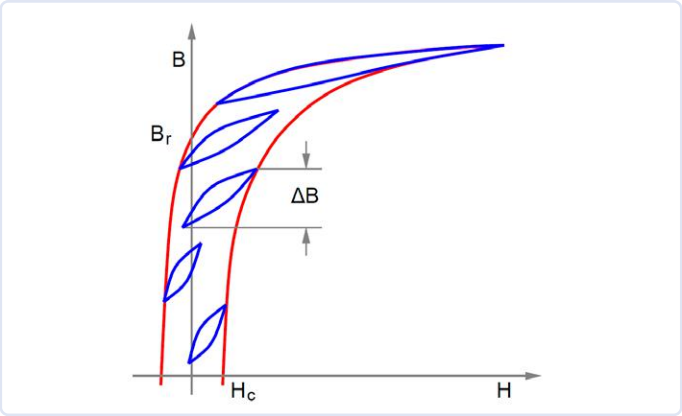


Figure 1. Boucles mineures situées à différentes positions le long de la boucle principale. Elles ont toutes la même variation de flux crête à crête. La surface de la boucle correspond à la perte par hystérésis dans le noyau par cycle.

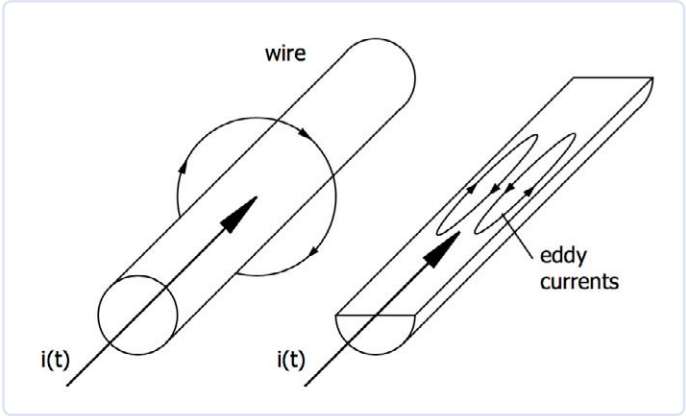


Figure 2. Les courants alternatifs induisent un champ magnétique, qui crée des courants de Foucault dans la direction opposée, annulant la densité de courant dans la région centrale et la renforçant dans la région externe.

de perte en courant alternatif est l'effet de la polarisation en courant continu. À première vue, cela semble contre-intuitif car un courant continu ne provoque aucun mouvement des dipôles magnétiques dans les domaines au-delà du changement initial. Cependant, lorsqu'un courant alternatif possède une composante continue, la courbe d'hystérésis mineure change de forme en fonction de la polarité et de l'amplitude du courant à différentes positions le long de la courbe BH (**figure 1**). Les mesures montrent qu'à de faibles niveaux de polarisation, l'influence est faible, mais qu'à des niveaux plus élevés, l'augmentation des pertes devient significative. On comprend que lorsque le matériau magnétique approche de la saturation, lorsque presque tous les dipôles magnétiques sont alignés, il faut plus d'énergie pour aligner ceux qui restent. Les méthodes traditionnelles de calcul de la perte dans le noyau en courant alternatif ne tiennent pas compte de ce phénomène, ce qui en rajoute à la mauvaise surprise.

Enfin, la différence entre les pertes calculées et les pertes mesurées est attribuée à des pertes supplémentaires provenant de causes telles que des effets de relaxation, des courants de Foucault excessifs, les pertes parasites et d'autres phénomènes encore mal compris.

Qu'est-ce que les pertes cuivre ?

La perte dans l'enroulement en courant continu provient de la résistance en courant continu du conducteur utilisé pour l'enroulement. Il s'agit simplement du produit de la résistance ohmique mesurée par la composante continue de la forme d'onde du courant au carré, $P = R \times I^2$. La perte dans l'enroulement en courant alternatif consiste en des pertes par effet de peau et principalement des pertes de proximité provenant de la partie en courant alternatif de la forme d'onde du courant.

Tableau 1. Sources, types et influences des pertes.

Source	Type de courant	Type de perte	Influencé par
Noyau	alternatif	hystérésis	matériau du noyau, température, forme d'onde, perméabilité
		courants de Foucault	tension appliquée, rapport cyclique, perméabilité, pemittivité
	composante continue	(hystérésis)	déplacement de la courbe d'hystérésis, distorsion de la boucle mineure fonction de la topologie
		supplémentaire	pertes mal expliquées *)
Enroulement	continu	résistive	matériau, température
	alternatif	effet de peau	fréquence, forme d'onde (harmoniques), température, position *)
		effet de proximité	fréquence, forme d'onde (harmoniques), position, nombre de couches, température, flux de fuite, position des enroulements (entrelaçage)

** Explication dans le texte*

L'effet de peau est dû au fait qu'à haute fréquence, la densité de courant n'est pas uniforme dans la section transversale du conducteur, car le courant est déplacé vers la surface du conducteur en raison d'inductances finies. La variation du courant dans le conducteur crée un champ magnétique autour du fil selon la loi de Faraday, mais qui, selon la loi de Lenz, induit un courant dans le conducteur dans la direction opposée. Par cet effet, la densité de courant est affaiblie au centre et augmentée près de la surface extérieure. (**figure 2**) L'effet de peau tel qu'il est généralement défini n'est valable que pour un conducteur unique dans un espace libre loin d'autres conducteurs. Ce n'est pas le cas dans les inductances ou les transformateurs où il y a normalement de nombreuses spires et couches étroitement enroulées ensemble.

L'effet de proximité décrit l'influence des champs magnétiques adjacents sur les courants dans un conducteur. Il y a deux effets. Les fils adjacents dont les courants circulent dans le même sens se repoussent (leurs champs s'annulent), laissant les surfaces en regard avec peu de courant (**figure 3 A**). Les fils adjacents parcourus par des courants en sens opposés s'attirent (leurs champs s'additionnent) et les surfaces en regard voient une plus grande densité de courant, les côtés opposés beaucoup moins (**figure 3 B**).

Dans les transformateurs, le courant est dans le même sens à l'intérieur d'un enroulement, mais dans le sens opposé entre les enroulements primaire et secondaire. Les inductances à un seul enroulement ont un courant uniquement dans le même sens. Le nombre de couches est l'un des facteurs qui influencent le plus les pertes dans les enroulements à courant alternatif. C'est particulièrement vrai pour les inductances, car avec chaque couche, la force magnétique augmente et n'est pas annulée par un enroulement secondaire. Avec chaque couche supplémentaire, les pertes augmentent de façon exponentielle, puisque la

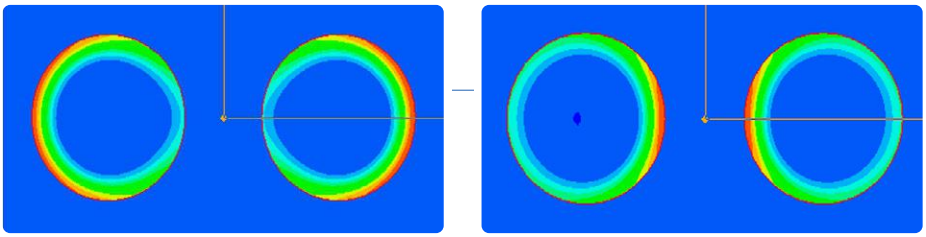


Figure 3. Courants dans des fils adjacents. À gauche — avec un courant dans le même sens. À droite — avec un courant dans le sens opposé. Le rouge correspond à une forte densité de courant, le bleu à une faible densité.

force magnétomotrice est le produit du flux magnétique et de la résistance magnétique (réluctance).

On comprend mieux pourquoi les inductances à fil plat à une seule couche, enroulés bord contre bord (ou bobines hélicoïdales) sont devenues si populaires dans le monde des inductances à courant fort. Le courant, qu'il soit alternatif ou continu, prendra toujours le chemin de moindre résistance qui est le diamètre intérieur, les pertes de proximité supplémentaires dues aux couches multiples sont éliminées.

Sélection d'une inductance avec REDEXPERT

Au fil des ans, Würth Elektronik eiSos a effectué des milliers de mesures de pertes sur les inductances de sa gamme dans des conditions d'utilisation réelles – formes d'ondes rectangulaires, rapport cyclique, composante continue, courant d'ondulation résiduelle et température. Ce riche ensemble de données sur les pertes totales en courant alternatif comprend tous les effets des méthodes de construction, des types de fils, du matériau du noyau et de la forme d'onde d'excitation. Il n'est pas nécessaire d'effectuer des calculs fastidieux et compliqués à partir de données souvent incomplètes, ou d'essayer de modéliser les pertes avec des réseaux de résistances, d'inductances et de condensateurs pour effectuer des simulations pour chaque choix possible.

L'outil de sélection d'inductances en ligne de Würth Elektronik eiSos, REDEXPERT (**figure 4**), permet d'accéder facilement à ces données et de comparer instantanément plusieurs inductances. Il suffit de sélectionner le type de convertisseur, de saisir les conditions de fonctionnement de base et toutes les inductances utilisables sont présentées. À partir de là, vous pouvez trier et sélectionner une liste d'inductances qui répondent à vos exigences, notamment en termes de taille, de hauteur et de forme. Ajustez facilement les conditions de fonctionnement pour vérifier les limites de votre conception. Des graphiques donnent immédiatement les domaines complets de performance, y compris les effets de la température. Cela vous permet, en tant qu'utilisateur, de comparer les mérites des différentes inductances, choisir celle qui vous convient le mieux, télécharger son fichier de caractéristiques et commander en même temps des échantillons gratuits.

REDEXPERT enregistrera automatiquement votre travail en cliquant sur l'icône de partage dans la barre de menu supérieure, où une URL unique sera affichée. Sauvegardez-la dans votre carnet de conception, envoyez-la par courriel à vous-même ou à un collègue pour la partager. L'affichage exact que vous voyez sera reproduit lorsque vous en aurez à nouveau besoin.

230470-04

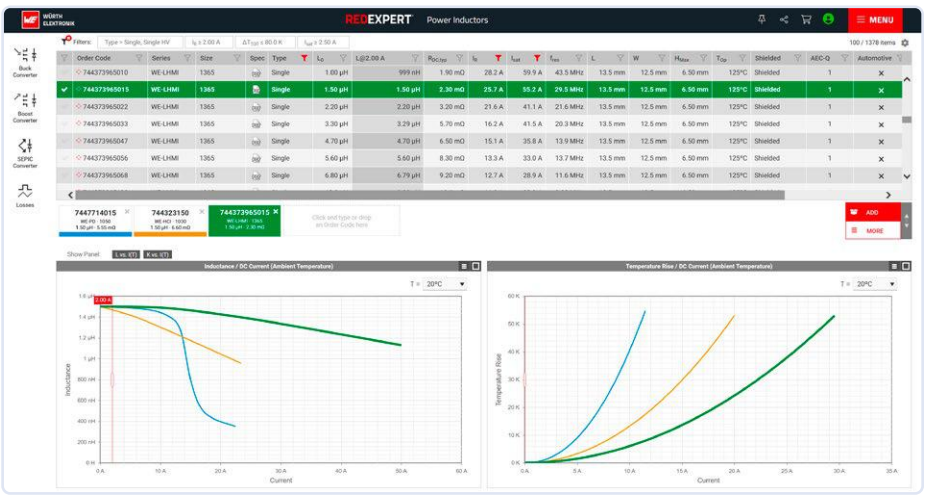


Figure 4. Les tableaux et graphiques de REDEXPERT permettent une comparaison rapide et précise des inductances.



À propos de l'auteur

George Slama a conçu et travaillé avec des transformateurs tout au long de sa carrière de plus de 40 ans. Son expérience en matière de conception s'étend des transformateurs audio et de télécommunications de l'ordre du milliwatt aux transformateurs et inductances ferro-résonants, radar, haute tension, petits transformateurs triphasés, transformateurs de commutation à haute fréquence et transformateurs et inductances LTCC. Son travail a porté sur le contrôle de qualité, les essais automatisés et l'ingénierie de fabrication, ainsi que sur tous les aspects de la conception et du développement d'alimentations à découpage personnalisées.

George a donné de nombreux séminaires sur la conception de composants magnétiques lors de conférences aux États-Unis et en Europe. Il travaille actuellement en tant qu'ingénieur principal chargé des applications et du contenu chez Würth Elektronik, où il développe des notes d'application et des outils destinés à aider les concepteurs d'alimentations à résoudre leurs problèmes de composants magnétiques.

LIENS

[1] Baguley, C., Carsten, B. Madawala, "The Effect of DC Bias Conditions on Ferrite Core Losses," IEEE Transactions on Magnetics, Vol. 44, No. 2, February 2008:

[2] Barbisio, E., Fiorillo, F., Ragusa, C., "Predicting Loss on Magnetic Steels Under Arbitrary Induction Waveform and With Minor Hysteresis Loops," IEEE Transactions on Magnetics, Vol. 40, No. 4, July 2004:

[3] Plate-forme de simulation en ligne REDEXPERT : https://redexpert.we-online.com/

optimiser son déploiement cloud grâce aux appareils de mesure



Figure 1. L'analyseur de réseau SNYPER-LTE Spectrum fournit une multitude de données qui facilitent le choix de l'antenne et permettent de définir l'emplacement idéal pour installer l'équipement sans fil cellulaire.

Par Stuart Cording, pour Mouser Electronics

Avec l'avènement de l'Internet des objets (IoT) et de l'industrie 4.0, les systèmes embarqués traditionnels se sont vus complétés par l'ajout de la technologie cloud. L'informatique en temps réel – utilisée, entre autres, pour contrôler un bras de robot ou un tapis roulant – utilise des serveurs à distance qui collectent et évaluent les données d'utilisation. Ce type de configuration offre un espace de stockage pratiquement illimité et une puissance de traitement colossale, ce qui permet de réaliser la maintenance prédictive, l'analyse de la consommation d'énergie et d'autres types d'analyses globales qui étaient auparavant difficiles à mettre en œuvre à grande échelle. Ces systèmes s'appuient généralement sur l'apprentissage automatique pour analyser les données et prédire les tendances. Les applications pratiques de tels systèmes ont beau encore faire penser à la science-fiction, de nombreux projets sont néanmoins en cours afin d'en étudier les possibilités techniques. Nous pouvons citer à ce titre le constructeur automobile Audi, qui, en collaboration avec l'entreprise de télécommunication Ericsson, a fait la démonstration d'un bras mécanique contrôlable en temps réel à l'aide des fonctions de communication ultra-fiables à faible latence (URLLC) de la 5G pour l'automatisation de l'industrie [1].

Analyse des sites pour la connectivité sans fil

L'installation de câbles de connectivité est une opération coûteuse et semée d'embûches. De plus, cela implique que l'équipement se trouve installé à un point fixe. Or, ce mode d'installation va à l'encontre de la tendance actuelle à la flexibilité avec des sites de fabrication qui peuvent être réaménagés à loisir pour s'adapter à de nouvelles utilisations. Cette flexibilité n'est évidemment possible qu'à condition d'utiliser une connectivité sans fil. Dans le cas d'autres applications, comme les véhicules à guidage automatique (VGA), la technologie sans fil est la seule solution envisageable. Avant de déployer de tels systèmes, il est recommandé de procéder à une analyse du site afin de cartographier la couverture des télécommunications disponibles sur place. Bien que les opérateurs de télécommunication fournissent des cartes de la couverture de leurs réseaux, la réalité sur place peut s'avérer très différente, notamment à l'intérieur d'un bâtiment.

Des appareils de test sont ainsi de bon secours pour simplifier l'étude d'un site (voir **figure 1**), à l'instar de l'analyseur de signal réseau haute performance SNYPER-LTE Spectrum (EU) de Siretta [2]. Comme son nom l'indique avec la précision (EU), cet appareil est destiné à détecter les réseaux mobiles européens. Il est livré dans une mallette de transport pratique contenant un ensemble d'accessoires soigneusement rangés dans leurs compartiments en mousse. Le kit comprend en effet, en plus de l'analyseur de spectre, un chargeur avec adaptateur pour prise secteur universel et pour prise allume-cigare, un câble USB et un câble RF. On y trouvera également deux antennes courtes : l'une à usage général 4/3/2G et l'autre pour la bande LTE 2 600 MHz. Enfin, une troisième antenne directionnelle permet d'effectuer des mesures liveSCAN directionnelles. Une carte SIM compatible 4G activée (non fournie) est nécessaire pour procéder aux mesures.

L'analyseur SNYPER-LTE Spectrum est capable de réaliser des analyses ponctuelles d'un site pour relever les caractéristiques du réseau (propriété de cellule, puissance du signal, etc.). La batterie intégrée permet de réaliser des relevés ponctuels pendant 48 heures (à raison d'une vingtaine de relevés par jour) et offre une autonomie de 15 heures en fonctionnement continu pour la réalisation d'analyses liveSCAN. Le stockage interne permet d'enregistrer les résultats de 50 analyses.

Les mesures liveSCAN s'appuient sur des relevés ponctuels pour générer une analyse plus détaillée. L'analyseur de spectre peut être utilisé pour rechercher les points où la puissance du signal est optimale à l'intérieur d'un bâtiment. Il suffit pour cela de le déplacer dans le bâtiment en utilisant l'antenne standard pour détecter les « hotspots ». L'antenne directionnelle permet quant à elle de trouver de quelle direction provient le signal le plus puissant. Les données ainsi relevées sont utilisées afin de déterminer l'antenne la plus adaptée et le meilleur emplacement pour installer l'équipement cellulaire.



Figure 2. La RFX-8440 est une carte de capture de données dans la bande L hautement configurable adaptée aux applications de test et de mesure RF ou de déploiement de masse.

Les résultats sont affichés à l'écran ou peuvent être téléchargés aux formats de fichier CSV et HTML à fins de reporting et de documentation.

Développement du système de test dans la bande L

Les systèmes de test RF complexes sont de plus en plus simples à développer grâce aux radios logicielles (SDR) dont les modules frontaux RF de haute qualité sont reliés à des pièces matérielles configurables comme des FPGA. La BittWare RFX-8440 [3] est une carte de capture de données RFSoC (système de radiofréquence sur puce) qui prend en charge les applications dans la bande L (1-2 GHz). Elle est utilisée par des domaines d'application aussi divers que la géolocalisation par satellite, les télécommunications, l'aéronautique et l'astronomie.

Cette carte de capture de données (voir **figure 2**) associe le RFSoC Zynq UltraScale+ ZU43 de Xilinx avec un module frontal analogique doté d'un système de prétraitement des signaux à faible bruit et à gain variable (-40 à 0 dBm) opérant avant les numériseurs. D'autres configurations frontales sont disponibles afin d'étendre la plage d'entrée à 4 GHz. On compte un total de quatre convertisseurs analogique-numérique (CAN) et quatre convertisseurs numérique-analogique (CNA) avec une résolution de 14 bits et un échantillonnage de 5 Géch/s et de 10 Géch/s respectivement. Ce RFSoC propose, outre sa logique programmable, un Arm Cortex-A53 à quatre cœurs et un Arm Cortex-R5 à deux cœurs. Bien que la carte de capture de données dispose d'un bus PCIe, elle peut aussi être utilisée de manière autonome grâce à ses interfaces Ethernet, USB et Display Link. Une possibilité d'extension est prévue avec un port OCulink à huit canaux doté d'un connecteur PCIe Gen4 x8, une connexion pour le stockage NVMe ou une double connexion réseau 100 Gbit.

La plupart des systèmes de test nécessitent des commutateurs programmables pour acheminer les signaux RF vers les ports appropriés. Le multiplexeur RF programmable T3SP-D4MX-BUNDLE [4] de Teledyne complète la carte de capture de données BittWare avec une correspondance de bande de fréquence et de phase CC à 10 GHz. L'appareil est livré avec une paire de câbles de 20 cm (8 pouces) à correspondance de phase (± 2 ps) pour les entrées et quatre paires de câbles de 60 cm (24 pouces) à code couleur et à correspondance de phase pour les sorties. Chacun des deux canaux offre des commutateurs 1:4 plus une option non connectée qui est le mode par défaut lors de la mise sous tension. Grâce à son interface USB 2.0, le multiplexeur peut être contrôlé au moyen d'une DLL en C/C++, C# ou Python ou être intégré dans des environnements de développement comme LabVIEW et MatLab. Un logiciel utilitaire simple est également disponible pour le contrôle manuel (voir **figure 3**). Chaque canal permet 1 milliard de cycles de commutation grâce aux commutateurs de microsystèmes électromécaniques (MEMS) RF.

Se préparer pour le cloud

De nombreuses machines encore en usage n'ont pas de connexion sans fil et encore moins un accès au cloud. Cependant, considérant que l'analyse des données est une importante aide à la prise de décisions, de nombreux fabricants cherchent à rendre leurs solutions de capture de données rétrocompatibles. Avec huit canaux d'entrées analogiques, le module de capture de données DAM-3918 [5] de DFRobot peut être utilisé pour partager les données de capteur analogiques sur son interface RS485 optiquement isolée à l'aide du protocole Modbus RTU. D'une précision de ±1 %, il peut être utilisé avec des transmetteurs et des transducteurs à 2, 3 et 4 fils (0-5 V, 4-20 mA, par exemple). Chacun des canaux d'entrée 12 bits peut être configuré individuellement et offre une fréquence d'échantillonnage de 50 éch./s, soit 400 éch./s disponibles sur l'ensemble des canaux. La tension de l'alimentation doit être comprise entre 18 et 30 V CC.

Oscilloscopes et capture de données à distance

De nos jours, les ingénieurs en développement disposent généralement d'ordinateurs portables relativement puissants. C'est pourquoi les constructeurs d'appareils de test et de mesure proposent désormais une large gamme d'oscilloscopes de bureau compacts qui utilisent l'écran d'un ordinateur externe comme moyen d'affichage. Cela permet aux développeurs d'utiliser des modules frontaux analogiques haute performance à moindre coût tout en intégrant une série de fonctionnalités avancées d'analyse et de décodage de protocole. L'Analog Discovery Pro 3000 de Digilent [6] est un exemple d'outil de mesure conçu suivant ce principe. Deux versions sont disponibles : l'ADP3450 à 4 canaux et l'ADP3250 à 2 canaux.

Il dispose, outre de ses entrées analogiques 14 bits et 0,5 Géch./s, de 16 E/S numériques et d'une alimentation numérique programmable. Utilisé avec le logiciel *WaveForms* [7] de Digilent, le Discovery Pro 3000 offre entre autres fonctionnalités la génération de formes d'onde et l'analyse de spectre et peut servir d'analyseur de réseau. Chaque instrument peut également être contrôlé au moyen d'un script écrit en JavaScript ou avec le kit de



Figure 3. Le T3SP-D4MX de Teledyne est un multiplexeur 1:4 CC à 10 GHz et deux canaux, contrôlable au moyen d'une DLL ou de l'interface graphique WinD4MX.exe.



Figure 4. Conçu pour fonctionner et collaborer à distance, l'équipement de test et de mesure Smart Bench Essentials de Keysight associe le fonctionnement traditionnel à la connectivité cloud.

développement logiciel (SDK) WaveForms en C/C++, C#, Visual Basic ou Python.
Il se distingue en outre par son mode Linux unique qui permet d'en faire une puissante solution de test de matériel opérant de façon autonome. Associé au SDK WaveForms, ce mode basé sur terminal permet de créer et de programmer des tests ou des applications personnalisés. Les résultats peuvent être transmis par Ethernet ou être enregistrés dans des tampons de capture locaux capables de stocker des millions de points de données.

Optimiser sa paillasse

Les équipements de test connectés à un ordinateur portable offrent beaucoup de fonctionnalités dans un boîtier réduit, mais ils n'encombrent pas moins la paillasse lorsqu'ils sont connectés. De plus, ils entravent sensiblement l'utilisation de l'ordinateur à d'autres fins, comme traiter ses e-mails, taper ses rapports ou participer aux inévitables vidéoconférences. Smart Bench Essentials de Keysight Technologies [8] est une famille d'outils de test et de mesure traditionnels, mais empilables. Ces appareils disposent d'une connectivité de qualité industrielle et d'écrans bien dimensionnés de 7 pouces (18 cm) et offrent la possibilité de partager les résultats des mesures.
La gamme comprend des oscilloscopes à deux et quatre canaux, un multimètre à 5½ chiffres, des générateurs de signaux arbitraires à un et deux canaux et une alimentation programmable à trois sorties (voir **figure 4**). Le kit d'empilement des instruments permet de réduire à un minimum l'encombrement sur la paillasse. Le logiciel inclus BenchVue permet d'enregistrer et d'analyser des données, mais le plus intéressant est le logiciel de gestion de laboratoire PathWave. Les gestionnaires et les enseignants peuvent assurer le suivi des équipements connectés à un réseau filaire Ethernet ou sans fil Wi-Fi, déployer des mises à jour du micrologiciel ou rendre les outils de mesure utilisables à distance via le cloud.

Déployer son cloud grâce aux tests et mesures

La connectivité cloud est en plein essor sur les marchés ciblés par les ingénieurs. Cela fait naître de nouveaux besoins comme le fait de disposer d'une vue d'ensemble de la topologie sans fil sur site, de méthodes de test RF avancées et configurables ou encore de solutions pour rendre des équipements anciens mais toujours utiles compatibles avec le cloud. Du fait de leur programmabilité, les oscilloscopes sans écran peuvent être réemployés comme outils de test et d'enregistrement de données à distance très polyvalents. Ils peuvent en outre transmettre leurs relevés sur des plateformes cloud à des fins d'analyse plus approfondie. Enfin, même les équipements de test traditionnels – chaque instrument dans son propre boîtier – offrent désormais une connectivité avancée et robuste qui permet une gestion efficace du laboratoire, le partage de mesures et un contrôle basé sur le cloud. On peut en conclure que de quelque manière que ce soit, les outils de test et de mesure trouveront toujours leur utilité dans un environnement cloud.

230458-04

Le Voltaire
1, Avenue Leo Lagrange
19100 Brive-La-Gaillarde
+33 5 55 85 79 96
france@mouser.com



À propos de l'auteur

Stuart Cording est un journaliste indépendant qui écrit pour Mouser Electronics. Spécialisé dans la création de contenu vidéo, il se concentre sur la recherche technique. Il s'intéresse particulièrement à certaines techniques, à la façon dont elles s'intègrent dans les applications finales et aux perspectives d'évolution. Mouser Electronics est un distributeur agréé de semi-conducteurs et de composants électroniques qui se concentre sur le lancement de nouveaux produits commercialisés par ses partenaires fabricants leaders de l'industrie. produits commercialisés par ses partenaires fabricants leaders de l'industrie.

LIENS

[1] 5G pour l'automatisation de l'industrie : https://www.ericsson.com/en/news/2020/2/5g-for-factory-automation
[2] Siretta SNYPER-LTE Spectrum : https://bit.ly/46DLXqf
[3] BittWare RFX-8440 : https://eu.mouser.com/new/test-measurement/bittware-rfx-8440
[4] T3SP-D4MX-BUNDLE : https://bit.ly/3D0q476
[5] DFRobot DAM-3918 : https://bit.ly/3D3gPTC
[6] Digilent Analog Discovery Pro 3000 : https://bit.ly/3PR8XMR
[7] Digilent WaveForms : https://digilent.com/shop/software/digilent-waveforms/
[8] Keysight Smart Bench Essentials : https://bit.ly/3JRekaP

implémentation de Matter : **que faut-il pour déployer** les appareils Matter ?

Sujata Neidig, NXP Semiconductors

Détenu par la *Connectivity Standards Alliance* (CSA), Matter est un protocole IdO universel et ouvert, un langage commun, qui permet aux appareils domotiques de communiquer entre eux, quelles que soient les marques et les plateformes domotiques (Amazon, Apple, Google, Samsung, SmartThings, etc.). Matter supprime les systèmes propriétaires (*Wallet Garden*-jardins clos) et apporte l'interopérabilité afin que les consommateurs puissent avoir la flexibilité et le choix des appareils qu'ils achètent, et la certitude qu'ils fonctionneront harmonieusement ensemble. Pour une étude plus approfondie de Matter, voir l'article *Matter – Making Smart Homes Smarter* [1].

Matter place également la barre très haut en matière de sécurité. Chaque appareil Matter doit être en mesure de prouver son identité et prouver qu'il s'agit d'un appareil certifié Matter, avant d'être autorisé à rejoindre le réseau Matter. Une fois que l'appareil est sur le réseau, toutes les communications sont cryptées. Pour en savoir plus sur la sécurité Matter, consultez le document *Matter – Making Smart Homes More Secure* [2].

Catégories d'appareils Matter

Matter détermine la façon dont les appareils communiquent entre eux, il définit les caractéristiques et les capacités des appareils au niveau de la couche d'application. Lors de son lancement, Matter a pris en charge sept catégories d'appareils et s'étendra à beaucoup d'autres, comme le montre **figure 1**. Les membres de la CSA encouragent l'extension de Matter à d'autres types d'appareils.

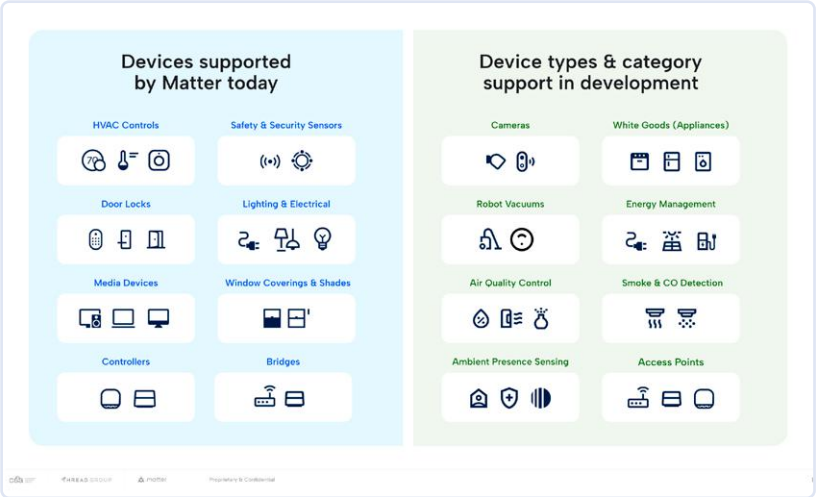


Figure 1. Catégories en pleine expansion d'appareils Matter. (Source : Connectivity Standards Alliance - CSA)

Aspects de conception

Matter présente aux développeurs plusieurs facteurs à prendre en compte afin de structurer leur système, de sélectionner les composants et de planifier la mise en œuvre des protections de sécurité.

Tout d'abord, le développeur doit décider des fonctionnalités dont l'appareil a besoin en fonction de la catégorie d'appareil Matter, des applications groupées associées et des caractéristiques nécessaires au-delà de Matter. Par exemple, l'interface utilisateur et les possibilités de l'alimentation (par exemple, alimentation sur secteur ou sur batterie, taille et durée de vie de la batterie, etc.) Ensuite, le développeur doit déterminer les besoins en matière de connectivité. En tant que technologie IP, Matter prend actuellement en charge les technologies WiFi, Thread et Ethernet, et les appareils peuvent utiliser une ou plusieurs de ces options. Le WiFi est idéal pour les applications à large bande passante, tels que le streaming audio ou vidéo, tandis que Thread est idéal pour les cas d'utilisation de commande à faible bande passante où la fiabilité et la faible consommation d'énergie sont une priorité. En plus de la connectivité, le développeur doit déterminer le ou les rôles que l'appareil prendra en charge, tels que les routeurs de bordure Thread, commissaire Matter, contrôleur Matter, pont Matter, etc. La dernière étape consiste à évaluer les exigences de sécurité de l'application au-delà des critères définis par Matter. Par exemple, une serrure de porte intelligente peut intégrer des protections contre les attaques physiques.

Une fois ces exigences définies, il est maintenant possible de déterminer l'architecture du système (**figure 2**) :

- **Autonome** : un seul microcontrôleur est utilisé pour mettre en œuvre l'application et la connectivité sans fil. Il s'agit d'une architecture idéale pour les types d'appareils les plus simples, dont l'aspect financier, la taille et l'alimentation en énergie sont des critères importants.

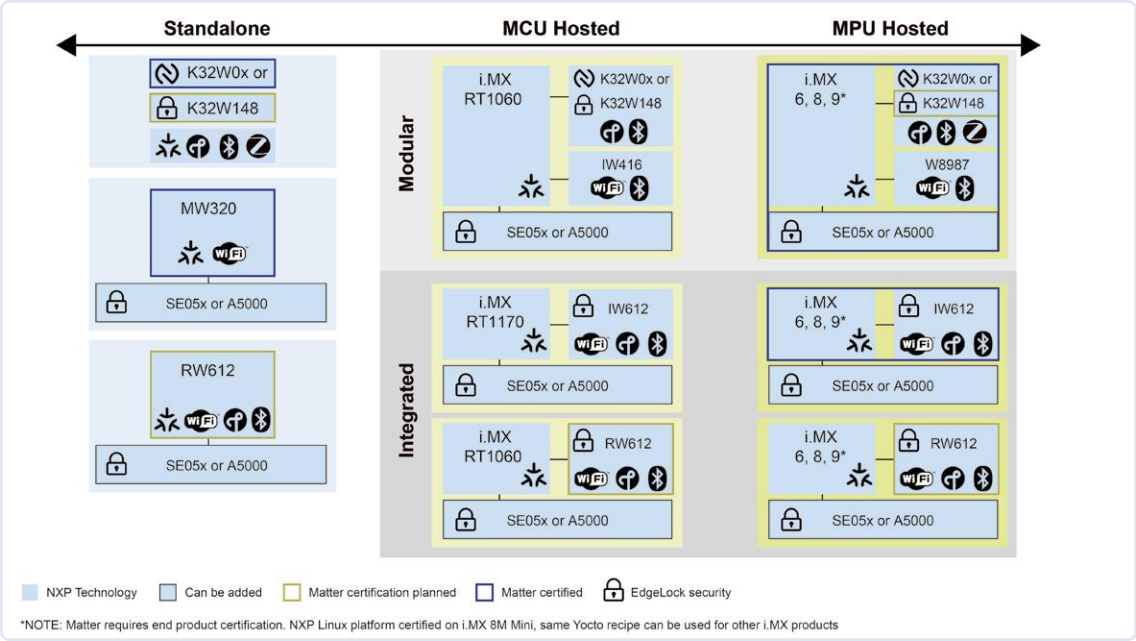


Figure 2. Options d'architecture de système pour les appareils Matter : autonome et hébergé. (Source : NXP Semiconductors)

- **Hébergé** : un microprocesseur ou un microcontrôleur hôte est utilisé pour mettre en œuvre l'application avec un microcontrôleur sans fil, ou un émetteur-récepteur séparé doté de l'électronique pour la radio. Cette architecture est utilisée pour les types d'appareils plus complexes, qui ont des interfaces utilisateur plus élaborées et/ou qui gèrent des réseaux et des fonctions multiples.

NXP propose une gamme de plates-formes de développement pour répondre à la catégorie de types d'appareils et de domaines d'utilisation. Les composants clés nécessaires au système global sont inclus dans ces plates-formes : traitement, connectivité et sécurité. Visitez [3] pour plus de détails.

Déployer votre protocole Matter

La certification est l'étape qui suit la conception du périphérique et qui est essentielle pour assurer l'interopérabilité. En outre, la certification permet d'obtenir une licence d'utilisation de la technologie libre de droits ainsi que les droits d'utilisation des badges technologiques. La CSA propose un programme de certification pour Matter qui comprend des scripts de test, des outils et des services (via des laboratoires de test agréés) pour faciliter le processus, y compris l'exigence de certification liée qui valide la certification des technologies sous-jacentes utilisées comme Thread, WiFi et Bluetooth. Remarque : ces technologies appartiennent à d'autres organismes de normalisation qui proposent chacun des options d'adhésion.

Étant donné que Matter est une couche d'application, chaque appareil Matter doit faire l'objet d'une certification Matter. Le développeur peut effectuer des tests préliminaires avec des outils fournis par la CSA. Une fois prêt, le développeur apporte le périphérique à un laboratoire de tests agréés (ATL) et demande la certification à la CSA. La CSA confirme la certification Matter, les certifications liées et délivre l'identifiant de certification. L'appareil est alors ajouté à la liste des produits certifiés de la CSA et au registre de conformité distribué (*Distributed Compliance Ledger*).

Comme Thread est une couche réseau qui n'est généralement pas modifiée par l'application, le programme de certification du Thread Group prend en charge la certification par analogie. Si l'appareil utilise un composant certifié Thread (du fournisseur de la puce choisi) et n'apporte aucune modification au micrologiciel, la certification peut être accordée par le biais d'une demande traitée sans recourir à un laboratoire d'essais certifié (ATL).

La dernière étape consiste à mettre en place un flux de production pour obtenir et attribuer un certificat d'attestation de l'appareil (*Device Attestation Certificate, DAC*) dans chaque périphérique. Le service EdgeLock 2GO de NXP est une autorité d'attestation de produit Matter (Product Attestation Authority) approuvée par la CSA, ce qui permet à NXP de fournir des *DAC Matter* aux sites de fabrication des clients via le cloud.

Le produit peut ensuite être lancé sur le marché et contribuer à offrir aux consommateurs l'expérience de la maison autonome ! De plus, les améliorations et les nouveaux correctifs des failles de sécurité peuvent être transmis de manière transparente aux utilisateurs par le biais de mises à jour à distance.

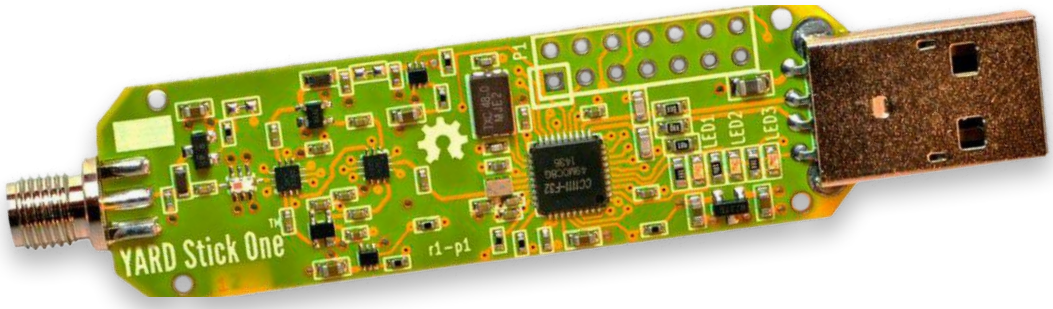
VF : Laurent Rauber — 230488-04

LIENS

[1] Matter – Making Smart Homes Smarter: https://www.nxp.com/webapp/sps/download/preDownload.jsp
[2] https://www.nxp.com/webapp/Download?colCode=MATTERSMRTHOMEWP
[3] Matter – NXP Semiconductors: http://www.nxp.com/matter

le YARD Stick One

un outil de test sans fil pour les fréquences allant jusqu'à 1 GHz



Wim Ton (Irlande)

Le YARD Stick One est une « radio matérielle » compacte capable d'émettre et de recevoir dans la bande UHF. Il s'agit d'une sorte de carte d'interface USB qui permet de l'utiliser avec des hôtes tels que les PC et les cartes Raspberry Pi. Le YARD Stick One est livré avec un logiciel USB préchargé dans son noyau 8051. La radio est contrôlée avec quelques douzaines de registres de configuration, mais le *middleware* intermédiaire Python permet de résumer de nombreux détails.

L'avantage principal du YARD Stick One de Great Scott Gadgets est qu'il s'agit de l'un des appareils les moins chers (comparé au HackRF [1] ou au LimeSDR [2]), qui peut également émettre et est prêt à l'emploi, alors que les appareils courants et bon marché tels que les dongles RTL SDR ne fonctionnent qu'en réception.
Comme toutes les fonctions de bas niveau sont réalisées par une puce CC1111 [3], l'utilisation de la radio consiste à écrire correctement les registres de configuration. Le CC1111 est destiné aux protocoles complexes de *Layer* 2, avec des fonctions telles que la synchronisation de mots, le cadrage, l'entrelacement et l'embrouillage. Le CC1111 étant un SoC pour les applications RF commerciales, l'utilisation du YARD Stick One pour l'analyse des signaux est très limitée. À moins que le système testé n'utilise un SoC similaire, il est moins frustrant et moins coûteux d'utiliser un SDR.
Le YARD Stick One n'est pris en charge que par *rfcat*, un *middleware* basé sur Python qui résume les options les plus utilisées dans une sorte de méthodes descriptives. Pour un réglage plus fin, *rfcat* offre également un accès brut aux registres.
Tel qu'il est livré, l'appareil est une carte nue et doit être traité avec le soin nécessaire. Des boîtiers provenant d'autres fournisseurs sont disponibles.
Le terme « sub-1 GHz » est un peu large ; le YARD Stick One est limité par son microcontrôleur radio TI CC1111, qui couvre les bandes ISM UHF inférieures de 300 à 928 MHz. La bande 13,56 MHz, utilisée pour la RFID notamment, n'est pas couverte.
Le YARD Stick One fonctionne de manière un peu plus stable sous Linux que sous Windows 10. Windows ne reconnaît souvent pas du tout le dongle, mais vous pouvez vous en sortir en le débranchant et en le réinsérant à plusieurs reprises sous Linux.

Installation du logiciel

L'utilisation du YARD Stick One nécessite une bonne compréhension des couches OSI 1 et 2. Une certaine connaissance de Python et une familiarité avec le système d'exploitation prévu sont également bénéfiques pour résoudre les problèmes d'installation. Le logiciel recommandé dans sur l'e-choppe Elektor [4] a été installé sur Windows 10, Kali et Ubuntu 18.
L'installation de *rfcat* [5] sur Linux avec Python 3.10 a bien fonctionné. La seule différence par rapport à la documentation est que *rfcat* doit être démarré avec la commande :

```
./rfcat
```

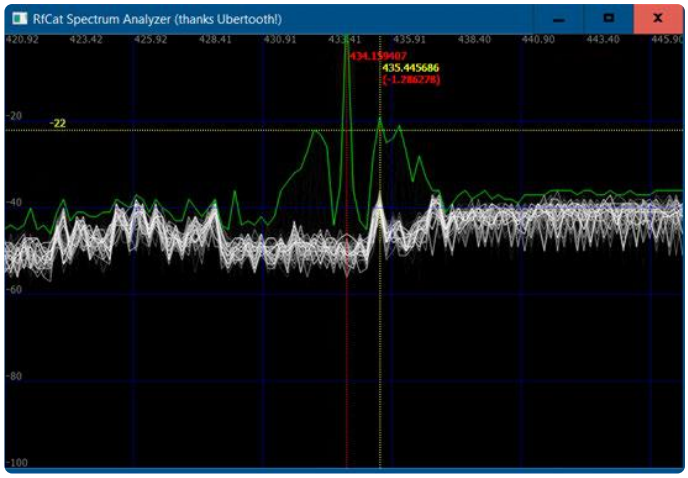


Figure 1. Affichage du spectre Rfcat.

Si vous obtenez un message « Error in resetup() », débranchez et réinsérez la carte.
Windows 10 : N'installez pas Python via Microsoft Store, car cela perturbe les autorisations de fichiers ; installez-le manuellement pour tous les utilisateurs. En plus, vous devez entrer :

```
pip install Cython
```

Dans tous les cas, installez avec les privilèges d'administrateur (vérifiez où les instructions Linux requièrent `sudo`). Et installez une version (MSVC) > 14.
Un ajustement est nécessaire si vous obtenez un message d'erreur à propos de « collections not callable » : ajouter `.abc` dans *C:\Program Files\Python310\Lib\site-packages\pyreadline\py3k_compat.py* sur la ligne 8 :

```
return isinstance(x, collections.abc.Callable)
```

le package *pyreadline* n'est requis que pour Windows uniquement. Installez le pilote *libusb-win32*. Le moyen le plus simple est probablement d'utiliser *Zadig* [6], qui est généralement fourni avec *SDR#*. Si le périphérique est absent, vous obtenez l'exception « No Dongle Found » de *rfcat*. En cas d'exception « ChipconUsbTimeoutException », débranchez et réinsérez le dongle. Dans l'ensemble, l'installation et l'utilisation sous Windows 10 sont un peu plus délicates que sous Linux. Il est obligatoire d'appeler `setmodeIDLE()` à la fin du script, sinon une erreur « device not found » se produira au prochain démarrage.

Le microcontrôleur utilisé dans le CC1111 est une variante MCS51 et nécessite le compilateur SDCC (Small Device C Compiler) [7] version 3.5 ou inférieure. Il nécessite un travail manuel lors de l'installation, car la version actuelle est 4.x. Cependant, de nombreux utilisateurs se contenteront d'utiliser le micrologiciel *rfcat*.

Utilisation du YARD Stick One

La puce radio CC1111 fait tout le travail de bas niveau, ajoutant et supprimant les pré- et post-ambules, les mots de synchronisation, le CRC, ainsi que la modulation et la démodulation. La radio doit être entièrement configurée avant d'être utilisée, car la configuration de réinitialisation est inutile. L'écriture d'un petit programme Python tel que décrit dans [8] permet d'économiser du temps de saisie et des erreurs.
Rfcat est également équipé d'un afficheur de spectre. Le qualifier d' « analyseur » est un peu exagéré, la plupart des appareils plus rudimentaires ont une largeur de bande et une plage dynamique limitées, contrairement aux équipements obsolètes tels que le HP141 ou le HP181, qui peuvent afficher une largeur de bande de 1 GHz avec une plage dynamique de 80 dB.
Pour utiliser le YARD Stick One comme récepteur, les propriétés des couches 1 et 2 doivent être configurées correctement, sinon la radio ignorera le paquet. Pour analyser un signal inconnu, une radio SDR supplémentaire est nécessaire, et le matériel le moins cher pour cela sont les dongles RTL. Outre l'utilisation de GNU Radio et d'Audacity comme indiqué dans [9], Universal Radio Hacker [10] fournit un flux de travail plus intégré pour l'analyse et la relecture des signaux. Alternativement, le réglage de la puce radio appairée peut être récupéré à partir de l'interface matérielle si le type de puce est connu, comme illustré dans [11].
L'utilisation du YARD Stick One comme récepteur générique est assez délicate : lorsque les réglages sont trop basiques, beaucoup de bruit se retrouve en réception, et avec des réglages trop précis, tout est filtré. Cela pourrait fonctionner avec un atténuateur variable à l'entrée, mais je n'en avais pas sous la main.
Pour faciliter la configuration des nombreux registres du CC1111, l'utilitaire SmartRF Studio de TI [12] est très utile. Les valeurs calculées peuvent être écrites sur le YARD Stick One avec la fonction `setXxx(value)` appropriée.

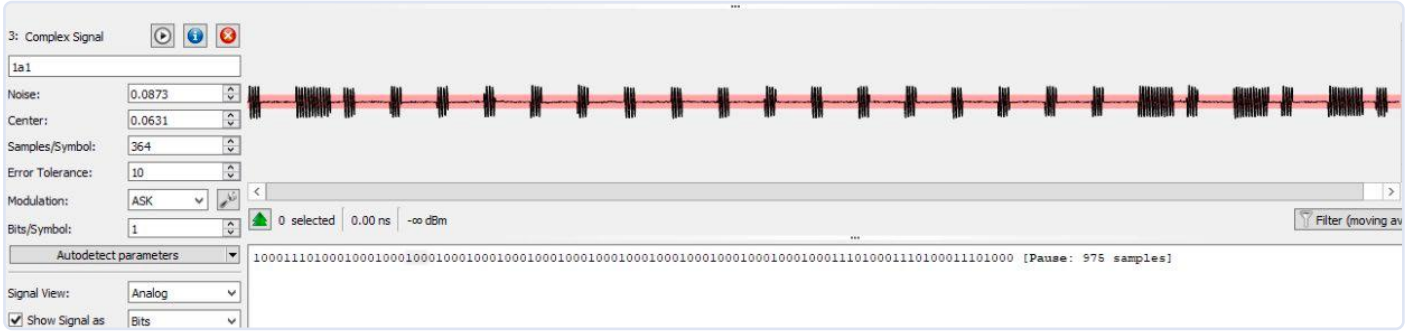


Figure 2. Visualisation du symbole.

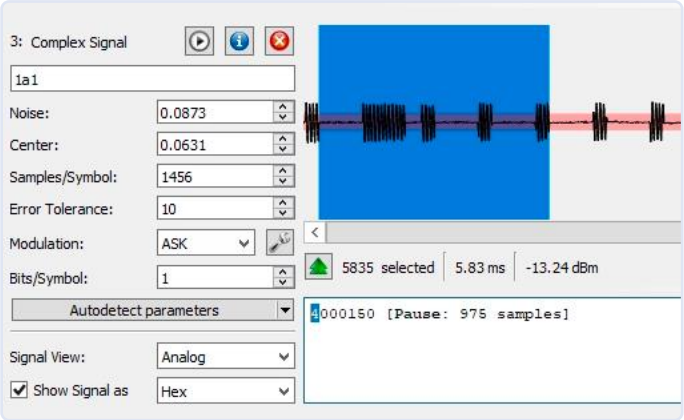


Figure 3. Visualisation des données.

Le *firmware* livré avec le YARD Stick One dans l'e-choppe Elektor agit comme un pont entre les registres du CC1111 et l'interface USB. Après une exception Python, le YARD Stick One doit être débranché et réinséré, sinon la *rflib* ne le trouve plus. L'exemple suivant montre un signal PWM avec modulation ASK, ce qui est très courant pour les télécommandes simples. Les captures d'écran proviennent de URH.

Chaque bit de données se compose de quatre symboles, un 0 est transmis sous la forme 1000 et un 1 sous la forme 1110, ce qui doit être envoyé au YARD Stick One sous la forme 8e88888888888888888e8e8e8. Un symbole dure 0,484 ms, le débit en bauds doit donc être réglé sur 2744.

Documentation

La documentation [4] fournie par l'e-choppe Elektor est très succincte. Le forum indiqué par Great Scott Gadgets est d'une utilité limitée. Le dépôt *git rfcat* donne beaucoup d'informations sur la construction et le téléchargement du firmware du YARD Stick One. Il y a quelques tutoriels sur Internet, voir par exemple [13]. Veuillez vous conformer à la règlementation locale pour la bande ISM.

Verdict

Le YARD Stick One n'est pas bon marché pour ce qu'il offre et sa phase d'apprentissage est plutôt difficile. Le logiciel n'est pas parfait et la documentation est un peu légère. Pour l'analyse uniquement, un simple récepteur SDR est un bien meilleur choix. Pour la transmission, vous avez un transceiver SDR générique pour 150 € de plus (par exemple, HackRF One ou Adalm Pluto).

Le YARD Stick One peut être utile si vous vous concentrez sur les protocoles spécifiques supportés par cette famille de SoCs radio (voir IM-Me [14] comme mentionné sur le site Elektor). Pour des applications dédiées, on peut envisager une carte d'interface CC111x [15] de Chine connectée à un Arduino, ce qui évite la communication USB peu pratique.

VF : Laurent Rauber — 230388-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

Produits

- **Great Scott Gadgets YARD Stick One – Outil de test sans fil sub-1 GHz**
  https://elektor.fr/20088
- **Great Scott Gadgets radio logicielle HackRF One (1 MHz à 6 GHz)**
  https://elektor.fr/18306
- **Great Scott Gadgets GreatFET One Universal USB**
  https://elektor.fr/19114
- **Kit Raspberry Pi RTL-SDR d'Elektor**
  https://elektor.fr/19518

LIENS

[1] Denis Meyer, "HackRF One SDR Transceiver," elektormagazine.com: https://elektormagazine.com/news/hack-rf-one-sdr-transceiver
[2] Jan Buiting, "LimeSDR Mini," elektormagazine.com: https://elektormagazine.com/news/digital-tv-transmitter-based-on-raspberry-pi-zero-and-limesdr-mini
[3] CC1110Fx / CC1111Fx fiche technique: https://ti.com/lit/gpn/cc1110-cc1111
[4] YARD Stick One logiciel et documentation: https://github.com/greatscottgadgets/yardstick
[5] RfCat GitHub: https://github.com/atlas0fd00m/rfcat
[6] Zadig: https://zadig.akeo.ie/
[7] SDCC: http://sdcc.sourceforge.net/
[8] YARD Stick One notes: https://bit.ly/3ql6mjo
[9] Tout pirater avec la RF et la radio logicielle- Part 1: https://bit.ly/3qpIGul
[10] Piratage radio universel: https://github.com/jopohl/urh
[11] Analyse des communications radio à l'aide de RfCat: https://bit.ly/42kxvAj
[12] SmartRF Studio de Texas Instruments: https://ti.com/tool/smartrftm-studio
[13] Tout pirater avec la RF et la radio logicielle - Part 2: https://bit.ly/43ITm5D
[14] Analyseur de spectre de poche à 16$: https://ossmann.blogspot.com/2010/03/16-pocket-spectrum-analyzer.html
[15] Carte d'interface pour CC111x: https://www.aliexpress.com/item/32963409008.html

relais à verrouillage

drôle de composant, la série



Figure 1. Relais de puissance ordinaire utilisé dans les appareils de commutation à distance de l'alimentation secteur.

David Ashton (Australie)

Il est facile d'actionner un relais standard. Utilisez un contact de rechange pour alimenter la bobine, et le relais restera actionné ou « verrouillé ». Mais cela nécessite toujours un courant dans la bobine. Seuls les relais à verrouillage restent actionnés même si la bobine n'est pas parcourue par un courant.

Les relais à verrouillage peuvent fonctionner avec deux bobines - l'une pour actionner le relais et l'autre pour le relâcher - ou ils utilisent une seule bobine, qui fait actionner le relais ou le relacher selon la polarité. Dans les deux cas, l'avantage est qu'il suffit d'une brève impulsion du courant fourni à la bobine pour modifier l'état du relais, qui reste ensuite dans cet état jusqu'à ce que l'autre bobine soit activée ou que la polarité soit inversée. Certains types de relais dépendent d'aimants permanents pour changer d'état.

J'ai utilisé différents types de relais à verrouillage dans des appareils à injection de fréquence. Il s'agit essentiellement d'appareils du réseau électrique qui alimentent les lampadaires, les systèmes de production d'eau chaude, etc. ; ils peuvent être commandés à distance avec une fréquence codée spéciale injectée dans les fils électriques (d'où leur nom). Grâce à l'utilisation de relais à verrouillage, ces dispositifs ne consomment pratiquement pas d'énergie dans tous les états et n'ont besoin que d'une courte impulsion pour changer d'état.

Les relais à verrouillage seraient également utiles dans d'autres applications où l'alimentation doit être commutée sans en consommer beaucoup. Par exemple, dans les régulateurs de charge solaire où l'on veut que la puissance maximale d'un panneau solaire serve à charger une batterie, mais où l'on veut aussi pouvoir éteindre le panneau solaire pendant les heures d'obscurité - ou éteindre la charge lorsque la batterie devient faible.

La **figure 1** montre un relais à verrouillage de puissance typique utilisé dans les appareils de commutation à distance de l'alimentation secteur mentionnés ci-dessus. Celui-ci possède une bobine de 24 V avec des broches de circuit imprimé et des contacts de 10 A avec des connecteurs plat. Une projection rouge sur le dessus indique sa position.

La **figure 2** montre quelques relais de Panasonic plus petits avec quatre contacts inverseurs. Il s'agit de relais à verrouillage à deux bobines de Panasonic, de type DS4E-ML2-DC5V, avec une isolation de 1500 V. Des relais à verrouillage plus petits comme ceux-ci seraient utiles dans les équipements alimentés par batterie pour la commutation de puissance ou de signal sans utilisation constante de la batterie - par exemple, dans la commutation d'atténuateurs en entrée et en sortie.

Les relais à verrouillage sont des composants qui sont utilisés dans des circuits très spécifiques, où leurs propriétés d'économie d'énergie les rendent très utiles. Avec leur simplicité, leur isolation et leur faible consommation d'énergie, aucun composant ne peut rivaliser avec leurs caractéristiques. Il vaut la peine d'en savoir plus, car ils peuvent remplir des fonctions qui seraient autrement très difficiles à réaliser.

220653-04

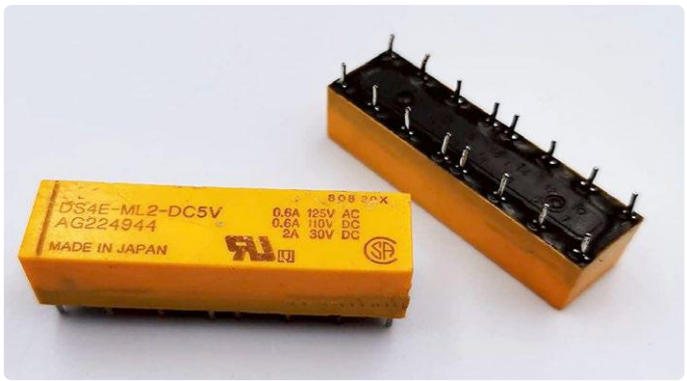


Figure 2. Relais à verrouillage à 2 bobines de Panasonic avec 4 contacts inverseurs.

PIC O'Clock – prendre le pouls du temps

conception d'un récepteur de signaux horaires SDR



Marco Rohleder (Allemagne)

Pour développer un récepteur de signaux horaires, le principe de la radio logicielle (SDR) est devenu incontournable. Elektor ayant publié de nombreux articles sur les logiciels de décodage de ces signaux, nous nous concentrerons ici sur le matériel nécessaire, avec, comme objectif, de se débrouiller avec un minimum de puissance de traitement en utilisant un simple microcontrôleur 8 bits équipé de fonctions périphériques indépendantes de l'unité centrale. À la fin du processus de développement décrit ici, le microcontrôleur PIC utilisé était même capable de recevoir et de décoder deux signaux horaires différents en même temps.

Comment peut-on construire un récepteur de signaux en utilisant des ressources facilement disponibles ? Comme le composant clé nécessaire à l'extraction du signal horaire est un microcontrôleur (MCU), il faut, d'évidence, exploiter le plus possible les ressources de ce microcontrôleur. Le récepteur doit donc reposer en grande partie sur le logiciel et minimiser l'utilisation de matériel supplémentaire.

En outre, il faut éviter d'utiliser des composants exotiques et difficiles à trouver et privilégier les composants standard dans la mesure du possible. C'est particulièrement vrai pour les composants CMS. Les condensateurs et les résistances en boîtiers 0805 peuvent être facilement soudés à la main, ce qui permet de limiter la taille du circuit imprimé. En revanche, le soudage d'un circuit intégré plein de pattes serrées n'est pas une partie de plaisir.

Outre le signal DCF77 sur 77,5 kHz de Mainflingen près de Francfort, de nombreux autres émetteurs en Europe et ailleurs dans le monde diffusent l'heure courante, notamment MSF60 sur 60 kHz en Grande-Bretagne, TDF162 sur 162 kHz en France et WWVB sur 60 kHz aux États-Unis. Outre les émetteurs de signaux horaires dédiés, il existe également des émetteurs de pilotage du réseau électrique, tels que DCF49 sur 129,1 kHz (également à Mainflingen) et DCF39 sur 139 kHz à Burg près de Magdebourg, qui transmettent l'heure courante. En outre, DCF39 et DCF49 ont une puissance d'émission de 100 kW, ce qui est nettement supérieur à la puissance d'émission de 50 kW de DCF77. Divers services météorologiques, tels que les bulletins de météo marine, sont également encore actifs sur ces fréquences. Ne serait-il pas agréable de pouvoir recevoir toutes ces stations (et même plus) dans la bande des grandes ondes ?

Les débuts sont toujours difficiles

Mes premières tentatives d'amplifier le signal DCF77 à partir d'une antenne à barreau de ferrite accordée sur 77,5 kHz et de visualiser la porteuse sur un oscilloscope pour voir comment son amplitude varie à la fréquence de 1 Hz ont été amèrement décevantes. Malheureusement, il y a beaucoup de bruit sur les grandes ondes. Les alimentations à découpage et les systèmes d'éclairage à LED de toutes formes et tailles, principalement d'origine asiatique, créent un niveau de bruit qui domine souvent totalement le signal souhaité. Lorsque l'on observe ce chaos de

tous les signaux possibles, il est difficile de croire qu'on puisse en extraire le signal DCF77. Les modules commerciaux sont apparemment conçus comme de simples récepteurs à amplification directe utilisant un quartz de 77,5 kHz comme filtre passe-bande très étroit. Bien que cette approche soit simple et peu coûteuse, elle ne permet de recevoir que des signaux à 77,5 kHz ±10 Hz. Il a bien fallu se rendre à l'évidence : sans un filtrage massif (et, comme on le verra dans la suite, numérique), une réception claire des signaux horaires était mission impossible.

Tout d'abord, examinons le schéma fonctionnel d'un récepteur de signaux horaires entièrement configuré (**figure 1**). Nous allons parcourir ces blocs un par un et les remplir de contenu.

Présélecteur à condensateur variable MOSFET

Même les filtres les plus efficaces et à forte pente réalisés en logiciel et les meilleurs et plus rapides convertisseurs A/N ne sont d'aucune utilité s'ils sont submergés par les parasites et que le signal utile est noyé dans le bruit. Il est impossible d'éviter un présélecteur qui élimine la majeure partie du bruit. Un circuit résonnant parallèle LC avec un condensateur variable (**figure 2**) est souvent utilisé comme présélecteur. La fréquence de résonance du circuit accordé est donnée par la formule :

$$f_0 = \frac{1}{2\pi\sqrt{LC}} \qquad \textbf{Formule 1}$$

Si vous utilisez une antenne à barreau de ferrite avec une inductance typique de 1500 µH, telle qu'elle est couramment utilisée pour la réception dans la bande des grandes ondes, vous aurez besoin d'un condensateur variable allant de 400 pF à 6,7 nF pour couvrir la gamme de fréquences de 50 à 200 kHz. Malheureusement, il n'existe pas de diodes à capacité variable répondant à cette spécification.

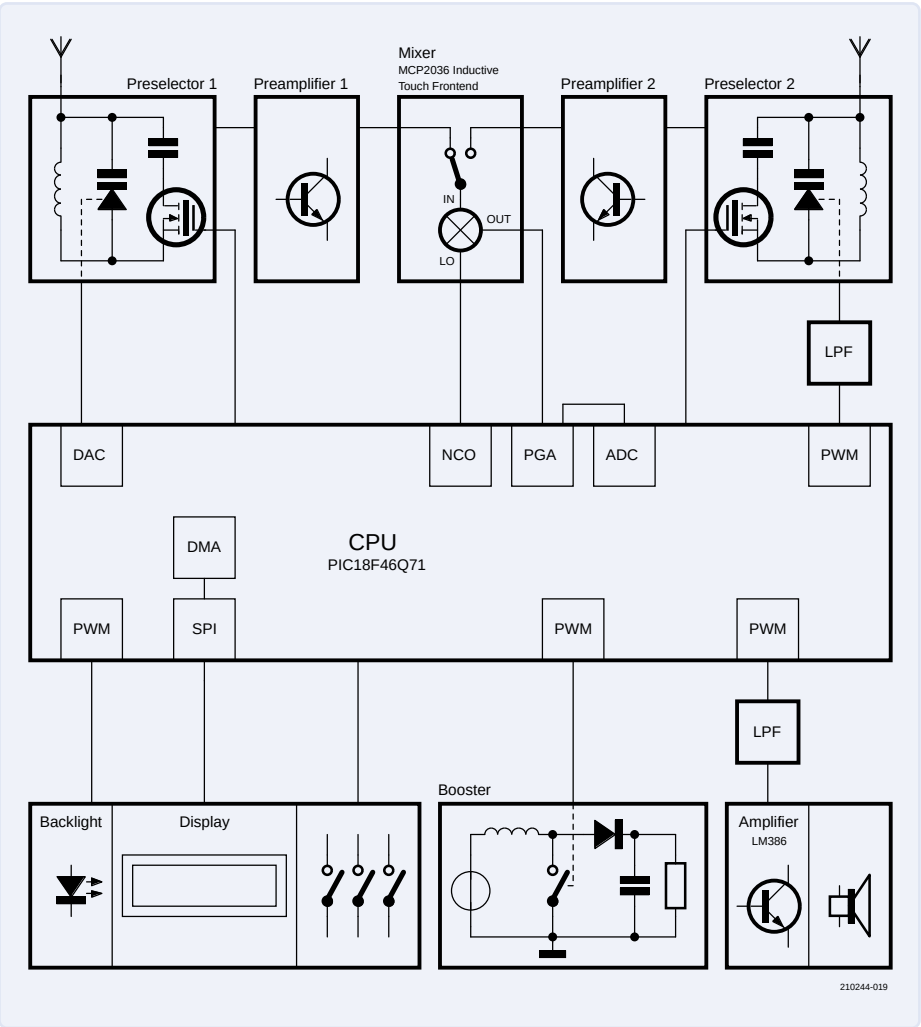


Figure 1. Schéma fonctionnel d'un récepteur de signaux horaires.

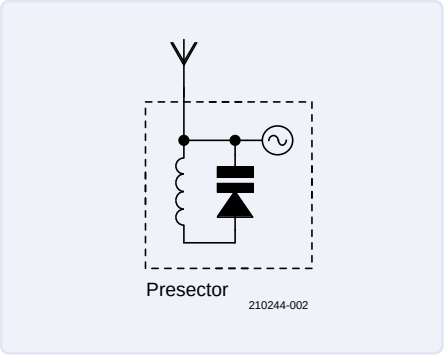


Figure 2. Principe d'un présélecteur.

Je me suis souvenu que chaque dispositif semi-conducteur possède une capacité de jonction inhérente (et non intentionnelle) qui dépend généralement de la taille de la jonction et de la tension. Mes premières mesures sur un MOSFET de puissance semblaient prometteuses et, parmi une variété de types différents, un IRF640 a donné les meilleurs résultats, avec une capacité de 500 pF à 20 VUDS et 3 nF à 1 VUDS. Pour faciliter le réglage dans la gamme de fréquences, j'ai ajouté un certain nombre de condensateurs supplémentaires (C30 à C34) qui pouvaient être connectés en parallèle avec les MOSFET de puissance (T35 et T36) selon les besoins à l'aide de transistors bipolaires à petits signaux (T30 à T34). Avec cette disposition, le présélecteur de la figure 3 peut être accordé sur une plage d'environ 30 à 300 kHz avec une inductance de 1500 µH. Afin d'utiliser au mieux la plage de réglage des MOSFET, ceux-ci doivent être pilotés par une tension plus élevée que celle fournie par le CNA du microcontrôleur. Par conséquent, un ampli-op LM358 est utilisé pour élever le signal d'environ 0...4,2 V à la plage de tension requise de 0...30 V.

Circuits résonnants alternatifs

Pour pouvoir tester différentes idées, j'ai utilisé un connecteur (ST31, **figure 3**) afin de pouvoir échanger cette partie du circuit.

La **figure 4** montre une manière simple de mettre en œuvre la rétroaction. Ici, le déphasage requis et un gain additionnel sont réalisés à l'aide d'un transistor supplémentaire (T80).

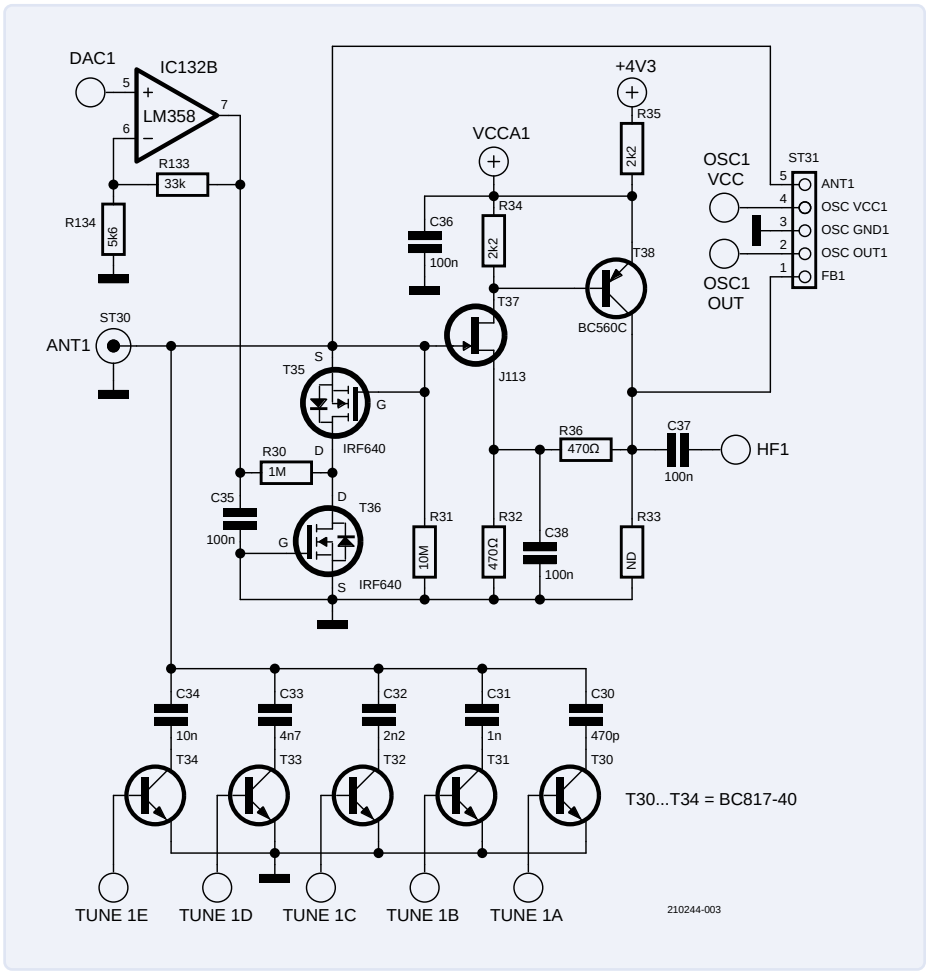


Figure 3. Le présélecteur avec l'amplificateur connecté pour le signal d'antenne.

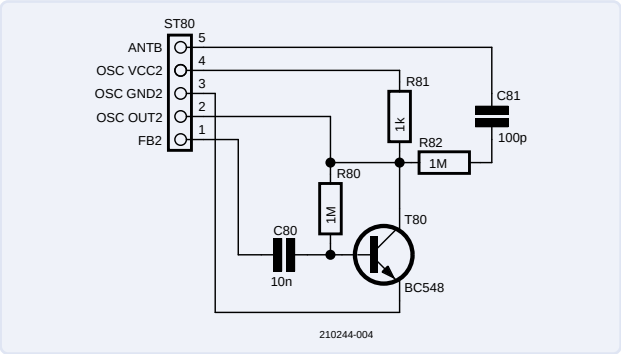


Figure 4. Oscillateur réalisé avec une rétroaction par transistor.

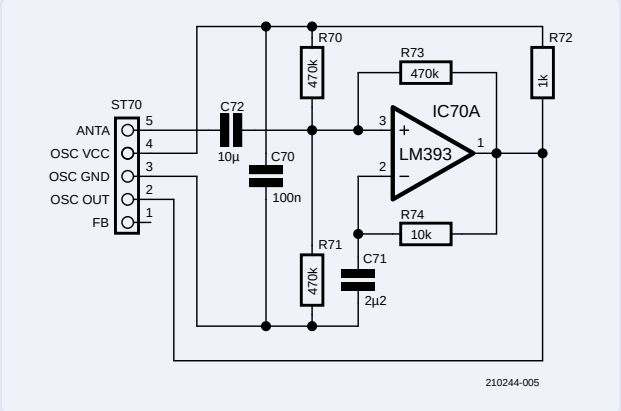


Figure 5. Un oscillateur basé sur un opamp.

Ce circuit présente l'avantage de n'amortir que faiblement le circuit résonnant, mais il s'est avéré difficile d'obtenir une résonance fiable sur une large gamme de fréquences.

La rétroaction transforme le présélecteur en un circuit résonant dont la fréquence d'oscillation est approximativement la même que la fréquence de résonance, du moins dans la bande des grandes ondes. Il y a une petite difficulté : à la fréquence de réception envisagée, le déphasage entre l'entrée et la rétroaction change, de sorte que le circuit ne résonne pas toujours à la fréquence de résonance prévue.

Pour le deuxième essai, j'ai adapté un vieux circuit ayant servi à réaliser un LC-mètre à l'aide d'un comparateur (figure 5). Il résonne sur une large gamme de valeurs d'inductances et de capacités et correspond très bien à la fréquence de résonance attendue. Cependant, comme c'est souvent le cas en électronique, une amélioration à un endroit se paie d'une dégradation à un autre. Bien que les bornes et la contre-réaction aient été réalisées avec une impédance très élevée, l'impédance d'entrée est sensiblement plus élevée qu'avec la solution précédente. Trois résistances de 470 kΩ (R70, R71 et R73) montées en parallèle du point de vue HF, soit environ 150 kΩ, accroissent l'amortissement du circuit résonant. Dans la pratique, cela n'a guère eu d'effet sur la qualité de la réception, si ce n'est que le gain global a dû être légèrement augmenté.

Pour cette variante, j'aurais aimé utiliser l'un des comparateurs intégrés au microcontrôleur. Bien que le multiplexeur de broches du microcontrôleur offre de nombreuses possibilités d'associer des signaux aux différentes broches d'E/S, l'usage intensif des périphériques analogiques se heurte à des limites, du moins si l'on veut éviter de router le signal sensible d'antenne en zig-zag à travers la carte. Pour cette raison, j'ai opté pour un comparateur externe (IC70) afin de garder les chemins de signaux aussi courts que possible.

En fin de compte, j'ai obtenu le circuit de la **figure 6** comme compromis. Ici, le circuit résonant est excité de manière fiable par une courte impulsion (200 ns) sur ST31/90 (broche 2) via R90/D90. Cependant, sans rétroaction, la résonance ne dure que quelques périodes, ce qui est trop court pour la mesurer avec suffisamment de précision avec les minuteries intégrées. C'est pourquoi le signal amplifié est transmis via ST31 (broche 4)

à l'un des comparateurs intégrés dans le microcontrôleur, lequel fournit en sortie (broche 2) une énergie supplémentaire pour maintenir le circuit en résonance (voir également la **figure 2**). Ici, la tension de référence du comparateur, qui peut être réglée via un CNA, détermine dans quelle mesure on obtient la fréquence de résonance attendue. Peut-être l'un des lecteurs de cet article a-t-il une idée sur la manière de régler automatiquement ce circuit en cours de fonctionnement afin qu'il fonctionne de manière fiable à différentes fréquences cibles.

Quelle que soit l'option choisie, on obtient un signal numérique d'une amplitude convenable qui est transmis à l'un des compteurs du microcontrôleur. Comme la résolution de fréquence souhaitée n'a pas besoin d'être meilleure que 100 Hz, il suffit de mesurer la fréquence avec une fenêtre temporelle de 10 ms (1/100 s).

Au démarrage, le logiciel commence par évaluer les gammes de fréquences qui peuvent être obtenues en connectant les condensateurs C30 à C34, dans chaque cas avec les tensions d'accord minimale et maximale. La bande la mieux adaptée est ensuite sélectionnée pour le processus d'accord définitif et la tension d'accord est ajustée pas-à-pas vers le haut ou vers le bas, en partant du milieu de la plage de tension. Le temps total nécessaire pour s'accorder sur une fréquence de réception prédéfinie est inférieur à 100 ms.

On peut également exciter le circuit résonant à une fréquence connue et ajuster la capacité jusqu'à obtenir l'amortissement le plus faible et donc l'amplitude la plus élevée. Pour ce faire, on peut connecter une résistance de très grande valeur aux bornes (ST31, broche 2, broche 5) et utiliser une fréquence de référence très précise sur la broche d'E/S (voir plus loin) pour exciter le circuit résonant.

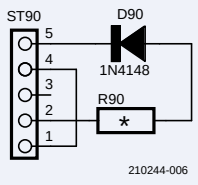


Figure 6. L'oscillateur est excité par une résistance et une diode.

En outre, j'ai placé une piste autour de l'ensemble de la carte, qui est également suffisante pour exciter l'antenne à la fréquence d'accord souhaitée. Cela fonctionne très bien, mais c'est beaucoup plus long que la solution précédente, car de nombreuses itérations sont nécessaires. Il me semble donc que la meilleure approche consiste à combiner les deux solutions. On commence par s'approcher grossièrement de la fréquence cible avec l'oscillateur, puis on procède à un réglage fin par excitation et accord jusqu'à la résonance maximale. La capture d'écran de la figure 7 montre un processus typique d'étalonnage et d'accord à 77,5 kHz.

Malheureusement, comme le montre la **figure 8**, la caractéristique de la capacité de jonction en fonction de la tension n'est pas linéaire (notez les échelles logarithmiques sur le graphique). En d'autres termes, les variations de capacité sont importantes à basse tension et deviennent de plus en plus faibles à haute tension.

Pour y remédier, on peut tirer parti d'une caractéristique particulière du CNA dans le microcontrôleur : il peut être alimenté par une tension de référence interne variable. Pour les faibles tensions de sortie, on utilise une tension de référence de 1,024 V, ce qui se traduit par une faible variation de tension. Pour des tensions d'accord plus élevées, la tension de référence peut être augmentée par pas jusqu'à 4,096 V.

Le signal du présélecteur est maintenant disponible pour un traitement ultérieur. Malheureusement, le CAN intégré dans le microcontrôleur a une fréquence de conversion maximale de 300 kHz.

Mélangeur et autres

Étant donné que le taux d'échantillonnage pour une mise en œuvre logicielle du récepteur doit être au moins deux fois la fréquence la plus élevée reçue, et de préférence quatre fois, la fréquence du signal reçu doit d'abord être abaissée pour qu'il puisse être traité par le microcontrôleur.

J'ai découvert par hasard un composant intéressant, le MCP2036 de Microchip, qui est en fait destiné à être utilisé comme frontal pour les capteurs tactiles, mais qui contient également un mélangeur (**figure 9**).

Le signal à mélanger est soumis aux deux entrées LREF et LBTN. La fréquence de l'oscillateur local est introduite par l'entrée CLK sous forme de signal numérique, ce qui

CHECKING TUNER ...		
Cx	F-LO	F-HI
0	127,000	227,200
1	106,300	147,500
2	91,400	113,900
3	82,700	98,200
4	74,400	84,900
5	69,400	77,700
6	64,800	71,400
7	61,300	66,900
8	56,000	60,000
9	53,800	57,300
10	51,400	54,600
11	49,700	52,600
12	47,700	50,200
13	46,300	48,700
14	44,800	46,900
15	43,700	45,600
16	39,800	41,200
17	39,000	40,400
18	38,100	39,400
19	37,400	38,600
20	36,500	37,600
21	35,900	36,900
22	35,200	36,100
23	34,600	35,600
24	33,500	34,400
25	33,000	33,800
26	32,500	33,200
27	32,100	32,800
28	31,500	32,200
29	31,100	31,700
30	30,600	31,300
31	30,300	30,800

TUNING ... (5)		
[512]	77,760	>
[256]	76,570	<
[384]	77,280	<
[448]	77,540	>
[416]	77,420	<
[432]	77,480	<
[440]	77,510	>
[436]	77,490	<
[438]	77,500	

Figure 7. Vérification du syntoniseur.

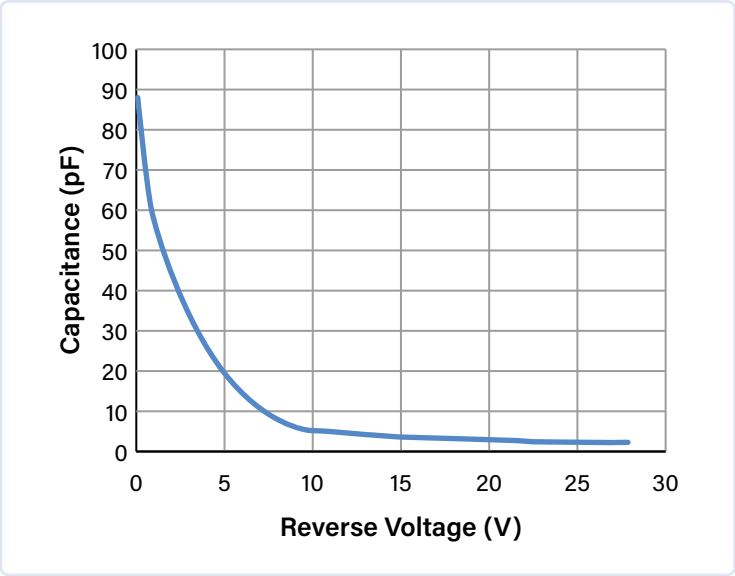


Figure 8. Courbe non linéaire de la capacité de jonction. (Source : Digikey - https://tinyurl.com/3js37yjn)

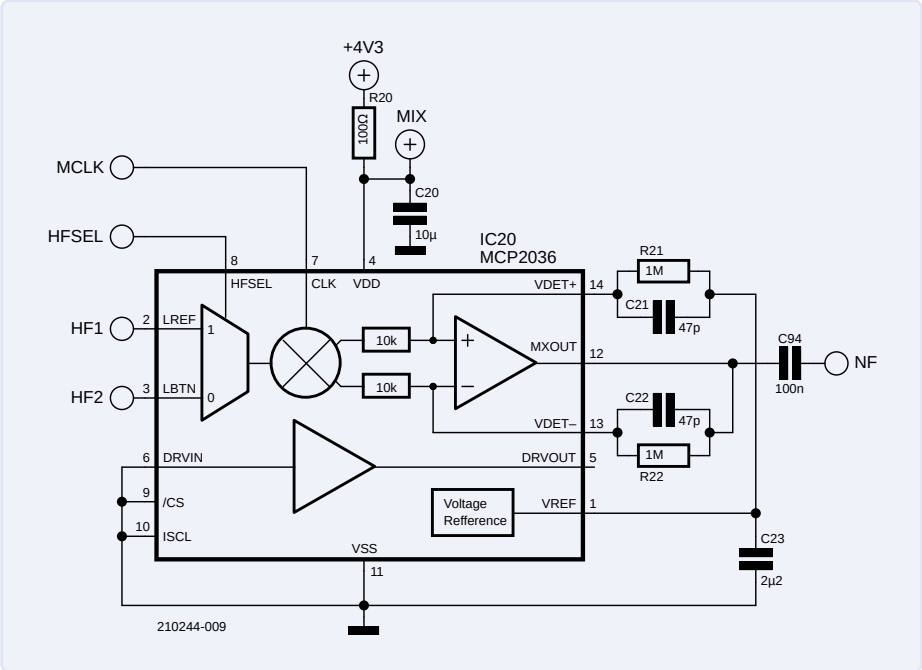


Figure 9. Structure interne du MCP2036. (Source : Microchip)

permet d'obtenir une fréquence intermédiaire de plusieurs kilohertz à la sortie. Ce composant présente également d'autres caractéristiques pratiques, notamment une source de tension de référence, un pilote (non utilisé ici) et un ampli-op qui peut être utilisé pour amplifier davantage le signal et, si nécessaire, agir comme filtre passe-bas afin d'éliminer les signaux ndésirables.

Ensuite, le signal conditionné et abaissé obtenu est envoyé au convertisseur A/N. Le signal est échantillonné à quatre fois la fréquence intermédiaire pour obtenir ensuite les composantes I/Q pour la démodulation numérique du signal reçu.

Mais pour pouvoir recevoir des stations à modulation d'amplitude telles que l'émetteur de signaux horaires DCF77, le gain de l'étage d'entrée doit être réglé de manière à ce que le signal à l'entrée du convertisseur A/N se situe dans la bonne plage – ni trop haut, ni trop bas. Pour ce faire, j'ai utilisé les ampli-op intégrés au microcontrôleur PIC18F46Q71, configurés en amplificateurs inverseurs et connectés en série. Un réseau de résistances est également disponible pour régler le gain, et, dans cette configuration, il permet d'obtenir un gain allant jusqu'à 15 par ampli-op. Comme pour le CNA, la tension de référence du CAN peut être sélectionnée entre 1,024 V et 4,096 V, ce qui représente un facteur de gain supplémentaire de quatre. Et si vous n'utilisez que huit des douze bits disponibles du CAN, vous obtenez un gain réglable supplémentaire de 16. Globalement, cela signifie que le gain peut être réglé sur une plage de 1:14 400 (14 400 = 15 × 15 × 16 × 4), comme l'illustre la **figure 10**.

Comme six bits suffisent en fait pour décoder le signal de manière fiable (comme montré ci-dessous), la plage dynamique peut être portée à 1:57 600 (quatre fois la valeur ci-dessus), soit plus de 95 dB. Cela devrait suffire, que vous soyez à 2 000 km ou à quelques mètres de l'émetteur.

Il ne nous reste plus qu'à générer une fréquence d'oscillateur local variable pour le mélangeur. Pour cela aussi, j'ai utilisé les périphériques du microcontrôleur. La plupart des microcontrôleurs contiennent un temporisateur/compteur qui peut diviser le signal horaire par un facteur n. C'est suffisant pour de nombreuses applications, mais la résolution est très basse. Supposons que la fréquence de référence soit de 16 MHz. En utilisant un diviseur de 80, on obtient une fréquence de sortie de 200 kHz, mais avec le diviseur immédiatement supérieur (81), la fréquence de sortie est de 197,5 kHz, soit un saut de 2,5 kHz. Outre ces minuteries standard, le microcontrôleur PIC utilisé ici dispose également d'un périphérique supplémentaire appelé oscillateur à commande numérique (NCO). Comme le montre la figure 11, il s'agit d'un additionneur qui ajoute à chaque fois une valeur prédéfinie à son dernier résultat pour en produire un nouveau. Cela se produit à des intervalles très courts, dans ce cas à la fréquence de l'horloge du microcontrôleur de 64 MHz.

Chaque fois que l'additionneur génère un débordement, soit un événement est généré, soit une sortie est commutée. À basse fréquence, cela permet des résolutions aussi petites que quelques dixièmes de hertz. Bien sûr, il y a une certaine gigue dans le signal de sortie, mais elle peut être ignorée dans ce cas.

Le microcontrôleur

J'ai déjà beaucoup parlé du microcontrôleur. Compte tenu des nombreuses fonctions et tâches que ce microcontrôleur doit fournir et exécuter, il peut paraître surprenant qu'au lieu d'un microcontrôleur 32 bits à haute performance, j'aie opté pour un microcontrôleur 8 bits avancé (Microchip PIC18F46Q71 [1]) qui, de plus, est disponible dans un boîtier DIP en plastique d'emploi facile. Il dispose d'une quantité de périphériques que j'ai largement

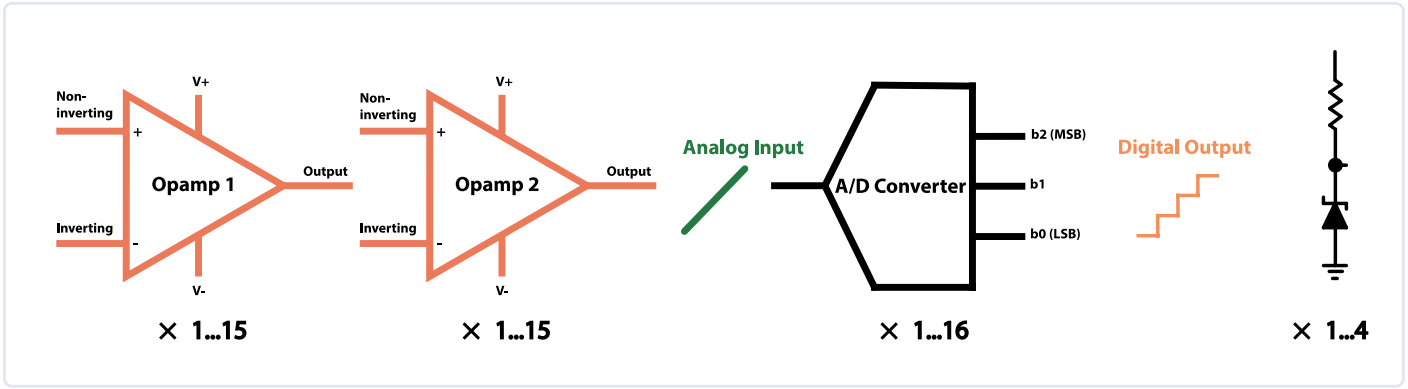


Figure 10.Gain global réglable.

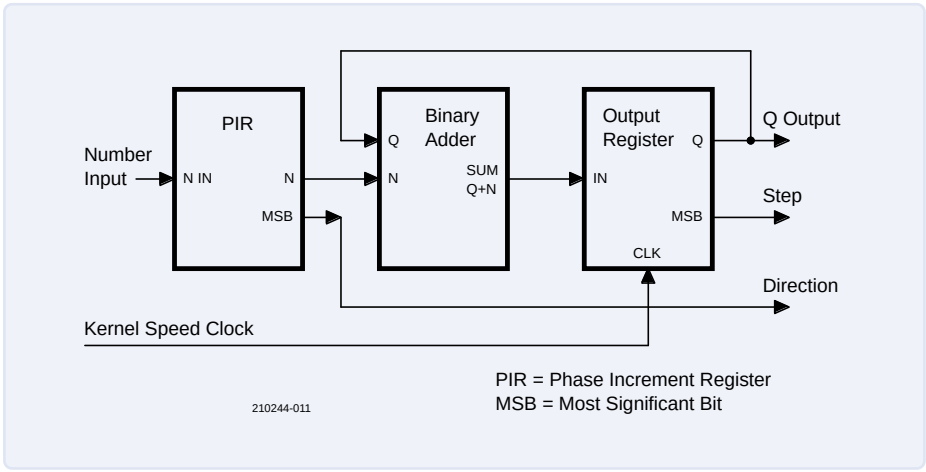


Figure 11. Oscillateur à commande numérique.

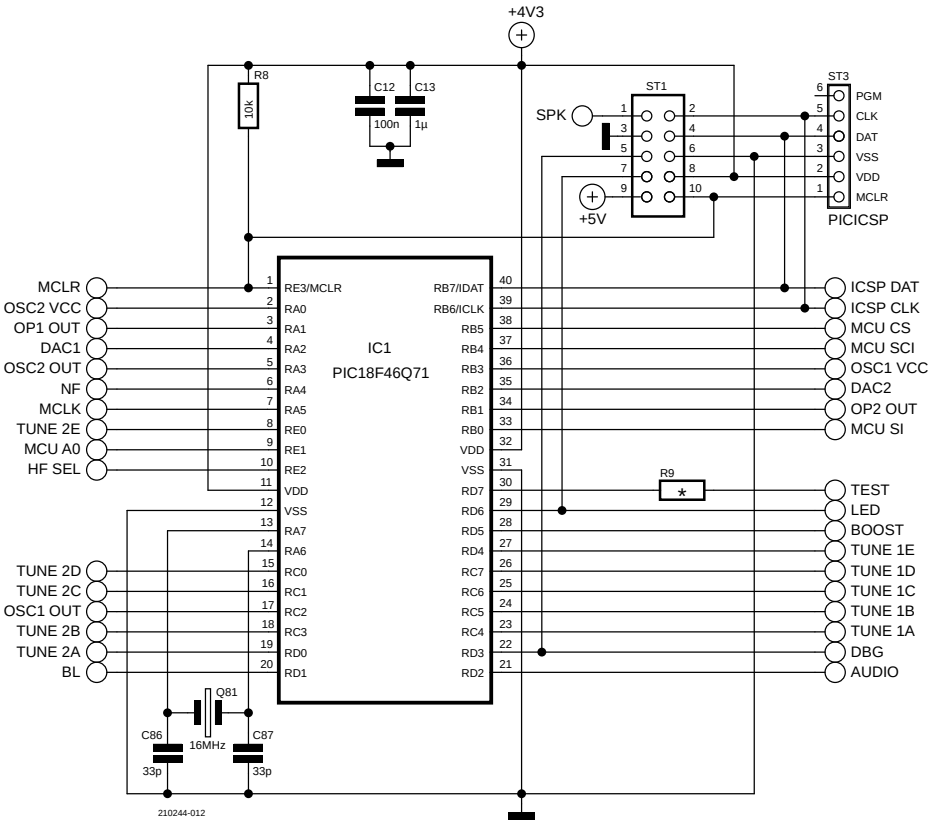


Figure 12. Le cerveau du récepteur de signaux horaires : le PIC18F46.

utilisés ici. Pour ce qui est des caractéristiques remaquables dans sa catégorie, il possède un CAN 12 bits avec un taux d'échantillonnage de 300 kHz, deux ampli-op intégrés avec chacun un produit gain-bande passante de 5,5 MHz, un contrôleur DMA multicanal, plusieurs canaux MLI rapides avec générateur de temps mort, deux UART et un oscillateur à contrôle numérique (le cœur du circuit de réception). Il contient également d'autres périphériques intéressants, notamment des cellules logiques programmables (qui peuvent, par exemple, être utilisées pour générer un signal avec un déphasage de 90 degrés), deux comparateurs rapides et une unité CRC matérielle, qui ne sont pas utilisés dans ce projet.

Étant également équipé d'un multiplicateur matériel, il peut fournir la puissance de traitement nécessaire pour des calculs complexes, tels que l'exécution d'un filtre de Hilbert pour éliminer la fréquence miroir.

La **figure 12** montre le microcontrôleur avec tous ses signaux d'entrée et de sortie. ST3 est le port de programmation ICSP, et plusieurs E/S sont reliées au connecteur ST1 à des fins de débogage. Cela permet de transmettre des données série à l'aide de l'UART, par exemple pour afficher la plage de fonctionnement du présélecteur ou pour configurer des signaux MLI à haute résolution supplémentaires afin d'émettre des valeurs analogiques pour le débogage des différents étages du démodulateur SDR.

Une interface graphique adaptée

Bien entendu, le récepteur doit également disposer d'une interface utilisateur sous la forme d'un petit écran graphique et de boutons de commande, ainsi que d'un potentiomètre (**figure 13**). J'ai opté pour l'un des modules d'affichage sur verre (DOG) bien connus, qui est connecté au microcontrôleur hôte par l'intermédiaire d'une interface SPI. Malheureusement, l'interface SPI de l'écran est unidirectionnelle, ce qui signifie que des données peuvent être affichées sur l'écran mais non lues.

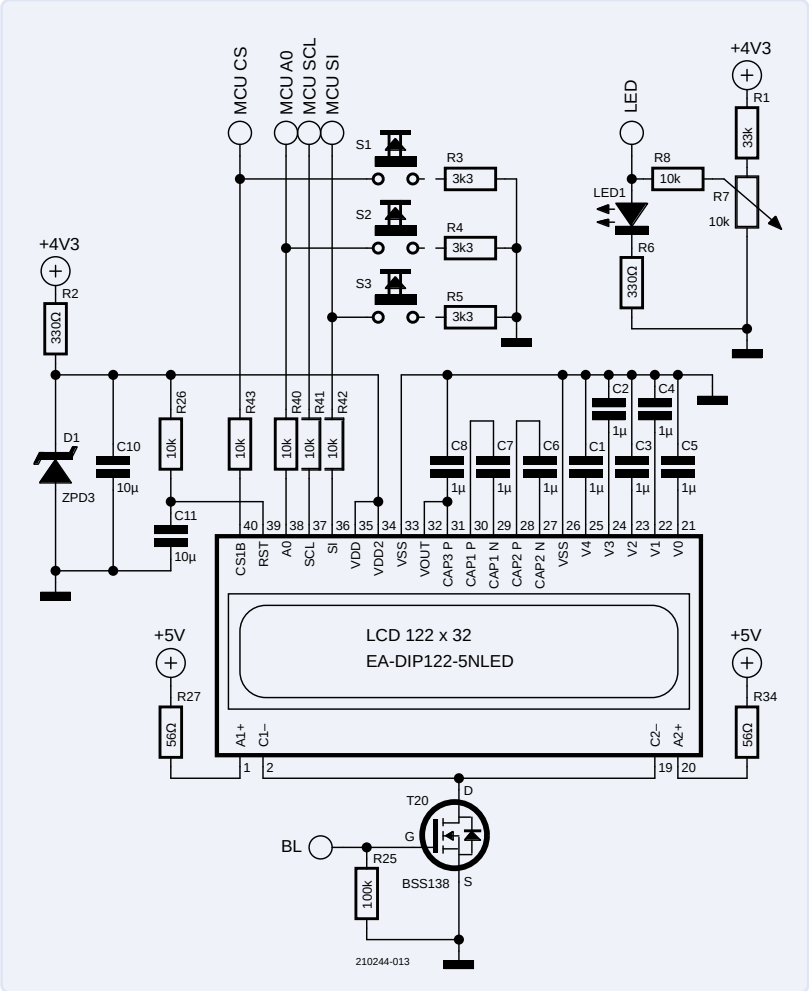


Figure 13. L'interface utilisateur se compose d'un écran, d'une LED et de quelques boutons-poussoirs.

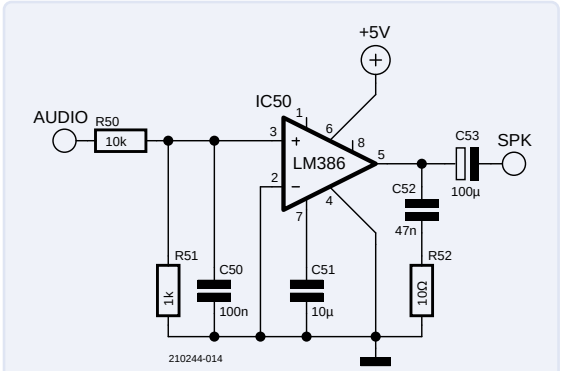


Figure 14. Sortie audio avec le bon vieux LM386.

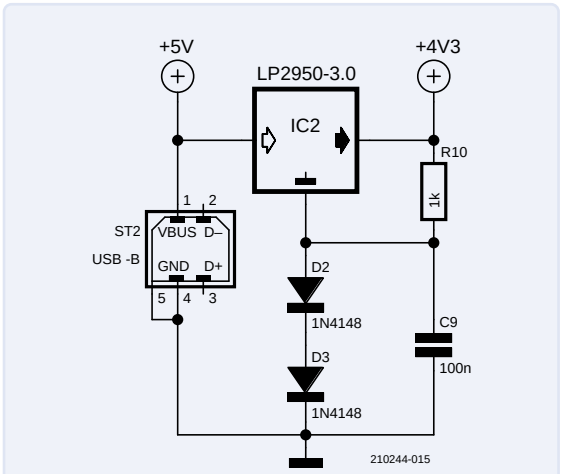


Figure 15. La tension d'alimentation de 5 V provenant de l'adaptateur secteur USB est abaissée à 4,4 V pour la partie numérique…

C'est pourquoi l'image graphique à afficher doit d'abord être créée dans la mémoire vive du microcontrôleur, puis envoyée périodiquement à l'écran. Pour ce faire, j'ai de nouveau utilisé l'un périphériques du microcontrôleur, en l'occurrence le contrôleur DMA. Il rafraîchit périodiquement l'écran avec son image en mémoire via l'interface SPI. Cette opération s'effectue en arrière-plan sans intervention de l'unité centrale.

La consommation totale de courant de l'écran n'est que de quelques centaines de microampères, sa tension d'alimentation de 3,3 V est obtenue à l'aide d'une diode Zener. Pour la conversion du niveau du signal, j'ai utilisé des résistances en série en combinaison avec les diodes parasites présentes sur les entrées de l'écran.

Audio

Comme il y a aussi des stations de radio dans la bande de réception (par exemple, BBC World à 198 kHz), j'ai également inclus un haut-parleur et un amplificateur dans le projet. J'ai cherché à utiliser au maximum les périphériques présents dans le microcontrôleur pour mettre en œuvre l'amplificateur.

Au départ, j'ai opté pour une conception en classe D, les transistors de l'étage de sortie étant pilotés numériquement par le microcontrôleur et passants ou bloqués. Les interférences avec le récepteur pouvaient être évitées en choisissant une fréquence d'horloge appropriée avec aussi peu d'harmoniques que possible dans la gamme de fréquences du récepteur. Mais ce qui a si bien fonctionné dans la mise en œuvre de l'élévateur de tension (voir ci-dessous) n'a pas marché ici, peut-être en raison de la puissance nettement plus élevée ou des chemins de câble considérablement plus longs vers le haut-parleur, ou les deux.

J'ai donc construit un amplificateur analogique conventionnel basé sur le classique LM386. Le signal audio est d'abord traité numériquement par le microcontrôleur, car il est de toute façon présent sous forme numérique en tant que composantes I/Q après la conversion A/N. À la sortie, les échantillons calculés sont transmis à une unité MLI fonctionnant à une fréquence d'horloge élevée, après quoi ils passent à travers un filtre passe-bas et sont ensuite transmis à l'entrée de la puce amplificatrice (**figure 14**).

L'unité PWM utilisée ici peut être cadencée directement à partir de l'horloge système de 64 MHz, ce qui permet d'obtenir une fréquence MLI de 15 625 kHz (64 MHz / 212) avec une résolution de 12 bits, ce qui donne la possible de régler le volume. Le haut-parleur choisi mesure 40×40 mm et sert également de support mécanique.

Alimentation électrique

Quelques mots sur l'alimentation (**figure 15**) : l'idée était de faire fonctionner l'ensemble du récepteur à partir d'un seul port USB. Comme la tension sur le port USB est largement non régulée, le régulateur de tension LDO 3 V IC2 la transforme en une tension de fonctionnement propre pour la partie numérique du traitement du signal. Le potentiel de masse de ce régulateur est décalé par deux diodes au silicium, ce qui donne une tension de sortie (VDD) de 4,3 V. Cette valeur est suffisamment basse pour compenser les variations de tension au niveau du port USB et suffisamment élevée pour les sources de tension de référence utilisées dans le microcontrôleur.

Reprenons depuis le début : pour la tension d'accord, la tension USB de 5 V doit être

portée à un niveau beaucoup plus élevé. Cette opération est réalisée par un convertisseur élévateur non régulé d'une rafraîchissante simplicité (**figure 16**). Le signal MLI utilisé pour piloter T70 est généré par le microcontrôleur avec un rapport cyclique approprié. La diode Zener limite la tension de sortie à un maximum de 33 V. Bien que ce circuit ne soit pas particulièrement efficace, la consommation de courant est faible et dépend surtout du courant de repos de l'opamp, de sorte que l'efficacité importe peu dans ce cas.

Antennes

L'antenne est reliée au récepteur par une paire de connecteurs SMB appariés. Cette disposition permet à l'antenne de tourner, donc d'être orientée le mieux possible dans la direction de l'émetteur.

Pour les premiers essais de réception, j'ai récupéré une antenne en ferrite d'un vieux réveil radio et je l'ai collée sur un morceau de circuit imprimé. Bien sûr, j'ai d'abord dû retirer le condensateur d'accord qui est maintenant simulé dans le présélecteur.

Comme il est désormais difficile de trouver des antennes à barreau de ferrite prêtes à l'emploi et même des barreaux de ferrite nus, et que bobiner n'est pas un passe-temps fort apprécié, je me suis mis à la recherche d'une antenne alternative. Sur la page de déstockage d'un distributeur d'électronique par correspondance, j'ai trouvé une antenne à cadre que j'ai aussi collée sur un morceau de circuit imprimé. Les résultats de réception étaient comparables à ceux obtenus avec l'antenne à barreau de ferrite.

Sur cette base, j'ai conçu un circuit imprimé avec des encoches pour faciliter le bobinage (figure 17) et j'ai fabriqué l'antenne moi-même. J'ai dû utiliser un fil de cuivre émaillé très fin pour fabriquer une antenne avec le nombre de spires requis. En somme, un processus très fastidieux, mais faisable.



Figure 17. Antenne à cadre maison.

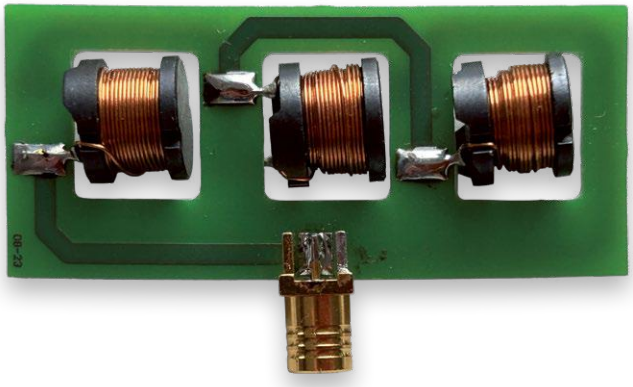


Figure 18. Une bonne réception peut être obtenue en utilisant des inductances fixes dans une configuration inhabituelle.

En fouillant dans mes tiroirs de composants, j'ai mis la main sur un grand sachet d'inductances fixes que j'avais acquises peu de temps auparavant. Serait-il possible de les détourner pour en faire une antenne non conventionnelle ? Si plusieurs d'entre elles sont connectées en série, l'arrangement est similaire à une antenne à barreau de ferrite, mais divisée en plusieurs parties (**figure 18**). Il s'est avéré qu'avec cette disposition, j'ai pu obtenir des résultats de réception comparables, avec une amplitude de signal nettement plus faible, mais avec une qualité de réception étonnamment bonne : je pouvais facilement recevoir DCF77 sur une distance de 800 km. J'ai essayé de monter les inducteurs à la fois radialement et axialement, et ils ont fonctionné dans les deux cas. En combinant les deux variantes (orientation radiale et axiale alternée), on devrait obtenir une antenne largement omnidirectionnelle. Si, par exemple, les lignes de champ passant par les inducteurs orientés axialement sont maximales, les inducteurs orientés radialement ne verront pas de lignes de champ effectives, et vice versa.

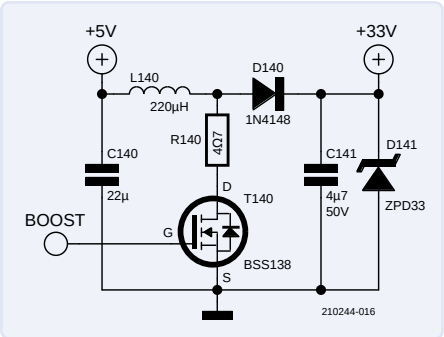


Figure 16. ...et élévée à 33 V pour la tension d'accord.

Logiciel

Le logiciel fonctionne essentiellement sur le principe d'une SDR conventionnelle, ce qui signifie que les composantes I/Q sont utilisées pour la démodulation du signal. Ce fonctionnement a déjà été décrit dans de nombreux articles d'Elektor [2] [3], il n'est donc pas nécessaire d'entrer dans les détails ici. L'extrait de code dans l'encadré Deux récepteurs valent mieux qu'un est la seule chose vraiment nouvelle qui mérite d'être mentionnée.

Ce circuit utilise de nombreux périphériques intégrés, par exemple pour la génération de la fréquence de l'oscillateur local pour le mélangeur et l'affichage par le contrôleur DMA, de sorte que la charge sur le microcontrôleur est très faible. Par conséquent, même ce microcontrôleur 8 bits dispose d'une puissance de traitement suffisante pour démoduler le signal reçu.

Le dernier maillon de cette chaîne est l'interface graphique, qui évalue les nouvelles valeurs calculées et met à jour l'affichage si nécessaire.

Perspectives

C'est tojours la même histoire : après avoir construit le premier prototype, il vous vient de nombreuses idées d'améliorations pour

Deux récepteurs valent mieux qu'un

Bien que le circuit ait été conçu à l'origine pour recevoir des données et des signaux horaires sur la bande des grandes ondes, le mélangeur dispose d'une deuxième entrée commutable, et j'ai donc décidé d'ajouter un présélecteur supplémentaire avec sa propre antenne. Celui-ci peut être équipé de composants pour la réception de grandes ondes ou d'ondes moyennes. Comme je l'ai constaté au cours du développement du logiciel, cela permet même de recevoir en même temps des signaux à deux fréquences différentes. Malheureusement, le microcontrôleur que j'ai utilisé ici n'a qu'un seul NCO, j'ai donc dû recourir à une astuce pour générer la deuxième fréquence de l'oscillateur local. J'ai utilisé l'une des deux minuteries universelles de 16 bits pour générer la fréquence inférieure de l'oscillateur local. Bien que le timer soit cadencé par l'horloge système de 64 MHz, il est possible de s'approcher assez près de la fréquence souhaitée de l'oscillateur local. À 77,5 kHz (DCF77), par exemple, l'erreur n'est que de 200 Hz. Si le deuxième timer universel est utilisé pour générer le taux d'échantillonnage de la fréquence intermédiaire, il peut être ajusté entre 2 000 Hz et 2 200 Hz avec une résolution d'environ 1 Hz, ce qui permet de compenser l'erreur décrite précédemment. Le NCO de l'autre circuit récepteur doit être ajusté en fonction de la nouvelle fréquence intermédiaire obtenue.

Bien entendu, le quartz utilisé n'a qu'une précision limitée. Si nous supposons que la précision est de 50 ppm, l'erreur à 77,5 kHz est de 4 Hz. C'est trop pour le filtre à bande très étroite utilisé pour le décodage. Néanmoins, l'erreur peut être déterminée par logiciel. Si le signal de l'émetteur souhaité peut être reçu avec un niveau de signal suffisant, la correction de fréquence peut également être effectuée automatiquement. Pour ce faire, le logiciel doit mesurer sur un intervalle de quatre périodes si le vecteur I/Q tourne dans le sens des aiguilles d'une montre ou dans le sens inverse. Si l'on effectue cette opération sur une période définie, on obtient une indication de l'écart entre la fréquence réelle de l'oscillateur local et la fréquence souhaitée. Une fonction de contrôle de niveau supérieur peut alors réduire ou augmenter la fréquence intermédiaire pour amener la différence à zéro. Le listage suivant montre comment cela peut être réalisé en C.

```
#define FILTERDEPTH 25
typedef struct {
  int16_t Ihistory[FILTERDEPTH];
  int16_t Qhistory[FILTERDEPTH];
  int32_t Iaverage;
  int32_t Qaverage;
  uint8_t index;
  uint8_t lastQuadrant;
  int8_t phaseChange;
} IQ;
IQ *iq;
void IQcallback( int16_t I, int16_t Q ) {

  uint8_t quadrant =0;

  // moving average lowpass filter I
  iq->Iaverage -= iq->Ihistory[iq->index];
  iq->Ihistory[iq->index] = I;
  iq->Iaverage += I;

  // moving average lowpass filter Q
  iq->Qaverage -= iq->Qhistory[iq->index];
  iq->Qhistory[iq ->index] = Q;
  iq->Qaverage += Q;

  iq->index++;
  if( iq->index >= FILTERDPTH )  {
    iq->index =0;
  }
  // integrate phase shift > phase error >
     estimate frequency offset of crystal
  if( iq->Qaverage<0 ) quadrant =1;
  if( iq->Iaverage<0 ) quadrant ^= 3;
  if( iq->lastQuadrant == ( (quadrant+1) & 3 ) )
    iq->phaseChange--;
  if( iq->lastQuadrant == ( (quadrant-1) & 3 ) )
    iq->phaseChange++;
  if( iq->phaseChange < -90 )
    iq->phaseChange = 90;
  if( iq->phaseChange > 90 )
    iq->phaseChange = -90;
  iq->lastQuadrant = quadrant;
}
```

la prochaine version. Néanmoins, j'ai décidé de m'en tenir à ma première approche et de la réaliser comme prévu. Si je devais concevoir à nouveau ce projet, je ferais ce qui suit :

- Ajouter un encodeur rotatif à l'interface utilisateur pour faciliter le réglage de la fréquence.
- Utiliser le PIC18F26Q71 (le petit frère du PIC18F46Q71 que j'ai utilisé), mais en prendre deux. L'espace requis sur la carte serait presque le même, et, de cette façon, j'aurais deux NCOx, quatre ampli-op, et quatre comparateurs. D'un point de vue HF, cela permettrait un meilleur routage. L'inconvénient serait la communication nécessaire entre les deux microcontrôleurs, mais cela resterait limité. Une unité centrale pourrait s'occuper de la numérisation et du traitement des données reçues, tandis que l'autre pourrait gérer l'interface graphique. Comme les deux microcontrôleurs fonctionneraient à partir de la même horloge source, une interface UART à un fil fonctionnant dans la gamme des Mb/s serait facilement réalisable.
- Remplacer les connecteurs SMB, devenus quasiment inabordables, par des prises jack à l'ancienne. Une version de 2,5 mm fonctionnerait bien ici, et la rotation de l'antenne resterait possible. Comme ces fiches et jacks sont également disponibles en versions à trois et quatre voies, il serait même possible de déplacer une partie du circuit, comme les condensateurs d'accord, vers l'antenne.
- Modifier les présélecteurs de manière à ce que le récepteur puisse recevoir des signaux dans la bande des ondes moyennes (bande de radiodiffusion AM) ou qu'il puisse être utilisé à des fins de radio amateur. Des expériences pratiques ont montré que le mélangeur peut monter jusqu'à un nombre de MHz à un chiffre.

Comme vous pouvez le constater, la SDR ne requiert pas nécessairement des microcontrôleurs 32 bits à grande vitesse, qui ne sont souvent disponibles que dans des boîtiers difficiles à souder à la main. En utilisant astucieusement les périphériques d'un microcontrôleur 8 bits, il est possible de remplacer une ou plusieurs fonctions et d'aboutir à une conception facile à réaliser.
Comme toujours, ce type de projet vit ou meurt en fonction du nombre de participants qui contribuent à l'effort. En particulier, l'écriture du logiciel prend beaucoup de temps. Mais peut-être quelqu'un aura-t-il l'envie et l'enthousiasme de faire avancer ce projet ? N'hésitez pas à nous le faire savoir !

VF: Helmut Müller — 210244-04

Questions ou commentaires ?

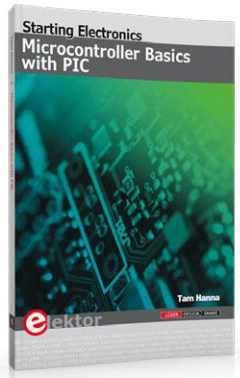
Envoyez un courriel à l'auteur (marco@happy-electrons.com) ou contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

Marco est né en 1970 et s'est intéressé à l'électronique dès son plus jeune âge. Il s'est occupé d'électronique pendant son adolescence, a étudié l'électrotechnique et travaillé en recherche et développement pendant près de 35 ans. Il est actuellement employé par un fabricant américain de semi-conducteurs. Marco est également radioamateur et répond à l'indicatif DD9DD.

Produits

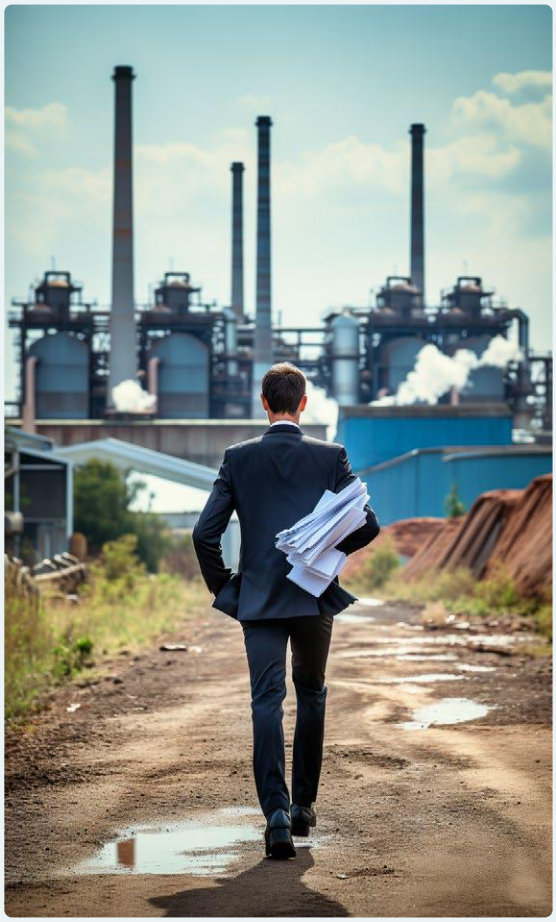
- **Bert van Dam, *PIC Microcontrollers* (Elektor 2008)**
  E-book, PDF : https://elektor.fr/18093
- **Tam Hanna, *Microcontroller Basics with PIC* (Elektor 2020)**
  Livre de poche : https://elektor.fr/19188
  E-book, PDF : https://elektor.fr/19189



LIENS

[1] Le microcontrôleur PIC18F46 : https://microchip.com/en-us/product/PIC18F46Q71
[2] Martin Ossmann, « Radio logicielle avec AVR » série en six parties, Elektor 3/2012 à 10/2012 : https://www.elektormagazine.fr/magazine-archive/2012
[3] Martin Ossmann, « horloges radiocommandées SDR » Elektor 1/2023 : https://www.elektormagazine.fr/magazine/elektor-291/61413

PCBWay
VOTRE PARTENAIRE ULTIME
POUR LA FABRICATION
DE CIRCUITS IMPRIMÉS !
POURQUOI CHOISIR PCBWAY?
https://www.pcbway.fr/
fr-service@pcbway.com
DES CIRCUITS IMPRIMÉS DE HAUTE QUALITÉ
Notre technologie de pointe garantit des circuits imprimés impeccables pour vos projets.
RÉALISATION ULTRA-RAPIDE
Aussi rapide que 24 heures pour le service de prototypage PCB.
COMMANDE SANS TRACAS
Il vous suffit de télécharger votre conception, et nous nous occupons du reste !
ASSISTANCE CLIENTÈLE DÉVOUÉE
Notre équipe d'assistance expérimentée vous assure une expérience sans faille, du début à la fin.



Source: Midjourney AI-generated image.
Prompt: "/imagine prompt: Back of person with lots of papers under his arm moving toward factory --ar 3:5"

▶

Due Diligence Directive

Business as Usual Will Not Do

By Priscilla Haring-Kuipers (The Netherlands)

Currently rolling through the EU is the Corporate Sustainability Due Diligence Directive or CSDDD [1]. The EU Parliament just voted generally to accept these new rules, and will continue to negotiate some of the finer points for another year. Things are about to change.

Our Sustainable Development Goals include the promotion of sustained, inclusive and sustainable economic growth. In the Paris Agreement, we promised to keep global warming within 1.5°C, and the private sector, with all their investments, must be in line with this if we are to keep our promises. Under the European Climate Law, we are committed to being climate-neutral by 2050 and to reduce our emissions by at least 55% in 2030. All of this means that there must be a change in the way companies produce and procure. Business as usual will not do.

Many issues in our global supply chains would be hard to tackle by any one company or even one country. These new due diligence rules intend to even the playing field within the EU market and prevent fragmentation in law. Germany already has a Supply Chain Act or "Lieferkettensorgfaltspflichtengesetz" [2], while other EU countries have limited requirements. By connecting all the big companies of many countries under common EU legislation, the power to effectively and sufficiently address environmental concerns and social inequality can be combined. Moving together is better.

The Ask

This directive concerns big EU companies with hundreds of employees and millions in turnover, as well as non-EU companies that make millions in the EU market. At its core, the CSDDD [3] is a way to force these big companies to take responsibility "by carrying out the following actions:

(a) integrating due diligence into their policies
(b) identifying actual or potential adverse impacts
(c) preventing and mitigating potential adverse impacts, and bringing actual adverse impacts to an end and minimizing their extent;
(d) establishing and maintaining a complaint procedure;
(e) monitoring the effectiveness of their due diligence policy and measures;
(f) publicly communicating on due diligence"

Companies will need to investigate themselves and their entire supply chain. When uncovering 'adverse impacts,' they must create corrective action plans with clear timelines and qualitative as well as quantitative metrics for improvement. Workers, unions and civil society organizations must be able to make human

rights violations or negative environmental impacts known. Companies have to provide a complaint procedure to support this, and inform relevant parties, as well as provide appropriate followup. Companies need to check on the implementation and effectiveness of their policies and on those of their business partners continuously. Such check-ins must be done at least once a year, or when new issues arise in the supply chain. All of this must be done transparently and communicated publicly.

Sanctions are mentioned in the directive, but it is left to every EU country to enforce them in line with national laws and proportional to the company's revenue. In the EU, they will set up a European Network of Supervisory Authorities to help and oversee countries do this.

A director of such a big company has a duty to oversee the corporate due diligence, to establish a code of conduct and to integrate this into the corporate strategy. The CEO must ensure that sustainability, human rights, climate change, and environmental consequences are properly considered in the short, medium, and long run. Making a profit is not a valid counterargument.

Adjust to Size

Big companies will be liable if they fail in their due diligence and preventable bad things happen or continue to happen. This does not mean that companies have to guarantee that bad things will never happen. Companies must take measures appropriate to their level of power in the chain and respond to human and environmental rights violations when they find them (and they are obligated to look).

The CSDD protects human and environmental rights throughout the lifecycle of a product and throughout the entire chain. This should mean that big digital companies are going to have to include the manufacturing of their hardware as part of their responsibility.

Small and medium-sized enterprises (SMEs), which make up 99% of all companies in the EU, are not included in the CSDDD. However, big companies will have demands of any SME they work with in order to be able to fulfil their own due diligence duties. EU countries are expected to support their SMEs by setting up dedicated websites, portals, and platforms — and perhaps provide financial support to build capacity. Big companies are expected to invest in the SMEs with whom they work, in order to help them to comply.

More Moral

I think this directive is a good example of how law slowly becomes a formal version of the values we hold as a society. Of course, you don't have to wait around for laws to be established, and it makes good business and moral sense to be ahead of the curve. One example of this is RS Group, providers of industrial and electronics products to engineers. They have already started to build an ethical supply chain seriously [4].

Many SMEs will soon discover what power (if any) they have in their supply chain, and, perhaps for the first time, undertake an ethical examination of their companies' actions. This forced self-reflection will likely come with compulsory forms to fill, new reporting practices and self-certifying quality control marks, along with big companies usurping more parts of the supply chain in order to maintain control and fulfil their legal responsibilities. If you are part of an SME that works with bigger companies, yesterday would be a good time to start investigating and reporting on your due diligence. Many will already have made many moral business choices. Now, these need just be formalized and communicated.

230428-01



Source: Midjourney AI-generated image. Prompt: "/imagine prompt: EU Parliament with EU flags --ar 5:3"

WEB LINKS

[1] EU CSDDD portal: https://bit.ly/43MAiUu
[2] Supply Chain Act (Wikipedia): https://en.wikipedia.org/wiki/Supply_Chain_Act
[3] Actual CSDDD: https://bit.ly/43Pg1Oa
[4] WEEF 2022 - Building an Ethical and Sustainable Supply Chain with Andrea Barrett: https://youtu.be/tuu88ePQwYQ

démarrer en électronique

...amplificateur de tension

Eric Bogers (Elektor)

Jusqu'à présent, nous avons utilisé le transistor en commutation et en amplification de courant. Bien entendu, ces deux fonctions sont très importantes en électronique, mais il est vraiment intéressant de pouvoir amplifier des tensions. C'est exactement ce que nous allons réaliser dans cet épisode.

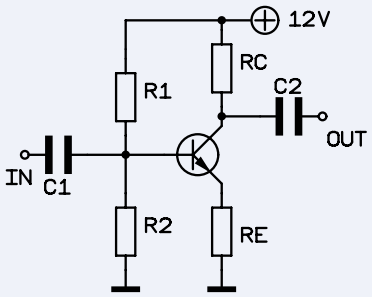


Figure 1. Montage émetteur commun

Mais, avant de nous lancer dans l'étude du montage émetteur, une mise en garde s'impose. Cela concerne l'article du numéro de mai/juin 2023, et en particulier le multivibrateur astable décrit dans la **figure 3** de la page 40. Comme nous l'a écrit Ruedi Schwarzenbach, lecteur d'Elektor en Suisse, ce circuit présente quelques dangers potentiels qui peuvent gâche le plaisir des amateurs d'électronique débutants.

De quoi s'agit-il en réalité ? Bien que nous n'avons pas mentionné un type de transistor spécifique dans le schéma et le texte, il est en fait évident d'utiliser le BC547, un transistor à usage général, pour cet exemple. Et, parce que les calculs dans le texte supposent qu'on utilise une tension d'alimentation de 12 V, les choses pourraient mal tourner. En effet, d'après la fiche technique, le BC547 ne tolère pas une tension inverse base/émetteur élevée, 5 à 6 V est à peu près le maximum. Et cela signifie que les transistors du circuit de la figure 3 susmentionnés peuvent déjà être endommagés s'ils sont alimentés par une tension supérieure à 5 V.

Bien que nous ayons mentionné que "la base du transistor de gauche est soumise un potentiel négatif et qu'il est donc bloqué", il faut garder à l'esprit qu'à un potentiel négatif de 5...6 V, la jonction base/émetteur présente une sorte d'effet Zener. Un courant circule (le condensateur est en effet court-circuité via les deux transistors), ce qui pourrait éventuellement provoquer la défaillance du transistor. Pour éviter ce genre de problème, il est préférable d'utiliser une source de tension de 4,5 V pour l'alimentation (comme une batterie plate de 4,5 V, par exemple) et de faire tous les calculs en fonction de cette tension.

Montage émetteur

Le circuit illustré à la **figure 1** inclut un montage émetteur (à ne pas confondre avec un émetteur-suiveur !), où l'émetteur est la référence commune pour le signal d'entrée et le signal de sortie. Le montage émetteur est très similaire au montage collecteur de l'épisode précédent, sauf qu'une résistance de collecteur est maintenant présente, alors que la tension de sortie est prélevée sur le collecteur. Le rôle principal de la résistance d'émetteur est maintenant d'empêcher le point de fonctionnement de dériver ; dans ce cas, sa valeur est beaucoup plus faible. En supposant, à nouveau, que le courant de repos est de 1 mA, on aboutit à une valeur de 1 kΩ pour la résistance d'émetteur.

La tension de sortie peut varier de 1 V à 12 V, c'est-à-dire dans une plage de 11 V ; nous choisissons la tension de repos au milieu de celle-ci – une tension de 6,5 V est alors appliquée au collecteur, tandis qu'une tension de 5,5 V est aux bornes de la résistance du collecteur. Nous pouvons maintenant calculer RC :

$$R_C = \frac{U}{I} = \frac{5.5\ \text{V}}{1\ \text{mA}} = 5.5\ \text{k}\Omega$$

Nous choisissons ici une valeur par défaut de 5,6 kΩ pour RC. Comme pour R2, nous choisissons (arbitrairement) une valeur de 100 kΩ. Calculons R1 :

$$R1 = \frac{U_0 \cdot R2}{U_B} - R2 = \frac{12\ \text{V} \cdot 100\ \text{k}\Omega}{1.7\ \text{V}} - 100\ \text{k}\Omega = 605.9\ \text{k}\Omega$$

Nous considérons une valeur par défaut de 560 kΩ pour R1.
Il faut préciser que cette tension de 1,7 V résulte de la tension aux bornes de la résistance d'émetteur de 1 kΩ à un courant de repos de 1 mA, plus la tension de 0,7 V traversant les bornes de la jonction base/émetteur.
La question qui se pose maintenant est : d'où vient le gain en tension et quelle est son importance. Si nous supposons que, suite à une tension alternative appliquée, la tension à la base du transistor augmente de 0,1 V, la tension de l'émetteur augmente également de 0,1 V, ce qui signifie que le courant augmente de 100 µA. À cause de ce courant plus élevé, une tension plus élevée traverse les bornes de la résistance du collecteur (0,56 V de plus, pour être précis). En d'autres termes, la tension au collecteur est réduite de 0,56 V.
L'*augmentation* de la tension de la base entraîne une *diminution* correspondante de la tension au niveau du collecteur, et ce n'est pas tout : la variation de la tension au niveau du collecteur est plus importante que celle de la base. La variation de tension à la base est donc amplifiée !
Dans cet exemple spécifique, nous avons également négligé la différence entre I_C et I_E. Si nous nous permettons de le faire nous pouvons écrire ce qui suit pour calculer le gain en tension du montage émetteur commun :

$$V_{OUT} = \frac{\Delta U_C}{\Delta U_B} \approx -\frac{R_C}{R_E}$$

Ainsi, dans cet exemple, le gain est de -5,6×. Le signe moins signifie simplement que le signal d'entrée et le signal de sortie sont en opposition de phase. En d'autres termes, il s'agit d'un amplificateur inverseur.
Il est possible d'amplifier un signal avec ce circuit encore plus, mais dans ce cas, il faut choisir une résistance d'émetteur moins élevée et/ou une tension d'alimentation plus élevée. Dans tous les cas, il faut éviter un *circuit* comme celui de la **figure 2**. Pour simplifier, le concepteur a omis dans ce circuit la résistance R2 et la résistance d'émetteur RE.

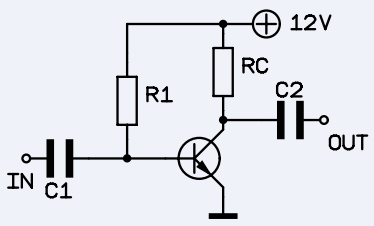


Figure 2. Mauvais montage amplificateur - ce n'est pas ainsi qu'il faut le faire !

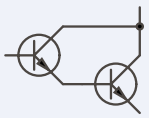


Figure 3. Montage Darlington.

Puisqu'on a omis la résistance d'émetteur, le montage fournit un gain très élevé, mais la stabilité en température du circuit est mauvaise : si la température change de quelques degrés seulement, le courant de base – et donc aussi le courant de collecteur – augmente (ou diminue) et le point de fonctionnement varie. En outre, le gain en courant d'un transistor n'est pas un paramètre fiable, car il présente une très forte variation, de sorte que la valeur de R1 ne peut être déterminée que par expérimentation.
L'omission de la résistance d'émetteur entraîne également une impédance d'entrée très faible, de sorte que la valeur de C1 choisie doit être beaucoup plus élevée afin d'éviter que la fréquence de coupure inférieure devienne trop élevée.
En résumé : un montage comme celui de la **figure 2** n'est pas utile. Ceux qui ont besoin d'une très grande amplification ont intérêt à connecter plusieurs étages d'amplification en série ou à utiliser au préalable un amplificateur opérationnel (ampli op).

Montage Darlington

Pour de nombreuses applications, nous avons besoin d'un gain en courant plus élevé que celui qui peut être fourni par un seul transistor. Dans de tels cas, le montage Darlington (**figure 3**) s'avère très utile. Dans ce montage, deux transistors sont commutés de telle sorte que l'émetteur du premier transistor est directement connecté à la base du second transistor. Dans ce cas, nous pouvons (approximativement) multiplier les facteurs de gain en courant séparément, de sorte que des valeurs de 10 000 et plus peuvent être atteintes avec des transistors à petit signal. Il convient de noter que les tensions base/émetteur des transistors sont additionnées, ce qui signifie qu'un montage Darlington nécessite une tension de configuration (polarisation) d'au moins environ 1,4 V.
Pour les transistors de puissance, dont le gain en courant est souvent très faible, on utilise plutôt le montage Darlington. Dans de nombreux cas, les deux transistors du circuit sont déjà intégrés dans un boîtier par le fabricant – on parle alors de transistor Darlington.

Source de courant constant

Si nous utilisons une LED pour une certaine application, nous disposons dans la plupart des cas d'une tension d'alimentation constante, et il est facile de calculer la résistance en série requise. En outre, le fait que la tension d'alimentation varie (dans certaines limites) ne pose aucun problème : le fait que le flux de courant de la LED soit de 5 mA ou de 20 mA ne change pas grand-chose (bien que, pour économiser, un flux de courant faible est toujours préférable). Ainsi, une tension d'alimentation qui varie d'un facteur quatre est toujours plus ou moins acceptable. Cependant, si la tension d'alimentation présente des variations plus importantes, il est recommandé d'utiliser une source de courant constant.

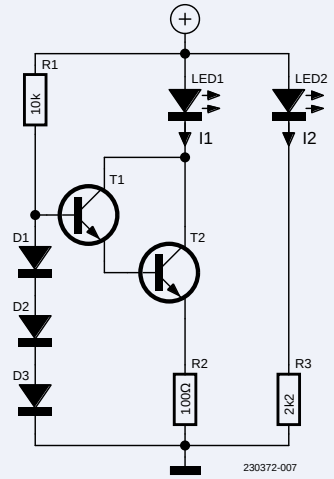


Figure 4. Source de courant (à peu près) constant.

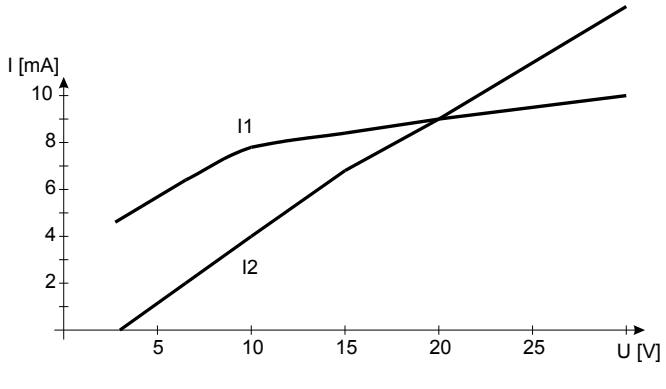


Figure 5. Flux de courant avec et sans source de courant constant

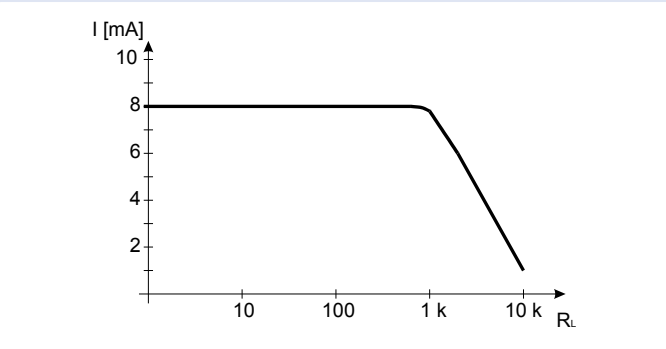


Figure 6. Source de courant constant avec une résistance de charge variable.

L'appellation " source de courant constant " pour le circuit de la **figure 4** est en fait un peu exagérée, comme le montre le graphique de la **figure 5.** D'un autre côté, ce circuit permet d'utiliser une LED dans des conditions où la tension d'alimentation peut varier d'un facteur 40, et c'est de toute façon un avantage.

Trois diodes connectées en série fournissent une tension plus ou moins constante d'environ 2 V. Cette tension est utilisée pour piloter un transistor Darlington qui a une résistance d'émetteur de 100 Ω. La tension aux bornes de cette résistance est d'environ 0,7 V, ce qui se traduit par un flux de courant d'environ 0,7 mA (ce qui est tout à fait correct en pratique). Par ailleurs, un transistor Darlington n'était pas vraiment nécessaire dans cet exemple. Nous avons mesuré le courant à travers la LED et l'avons représenté sur la **figure 5**. Pour comparer, nous avons également mesuré le courant à travers une LED avec seulement une résistance en série et l'avons représenté sur le même graphique.

Pour la source de courant constant, l'augmentation du courant est nettement moins forte (la courbe est plus uniforme), ce qui signifie que le circuit serait encore utilisable à des tensions d'alimentation nettement plus élevées. Le fait que le courant ne soit pas parfaitement constant, est dû à l'augmentation de la tension à travers les diodes – une diode Zener donnerait un meilleur résultat, et une stabilisation " à deux étages " aurait été encore meilleure.

D'ailleurs, une source à courant constant donne de meilleurs résultats lorsque la tension d'alimentation est constante (12 V dans la **figure 6**) et que l'on fait varier la résistance de charge (où se trouve la LED).

À des valeurs de résistance de charge plus élevées, le courant diminue, car sinon le produit du courant et de la résistance serait supérieur à la tension d'alimentation - ce qui est impossible. Cependant, tant que la tension à travers la résistance de charge reste bien inférieure à la tension d'alimentation, le courant à travers la charge est relativement constant.

Bien entendu, vous pouvez vous demander pourquoi nous aurions besoin d'un courant constant avec une résistance de charge variable. Eh bien, sans une telle source de courant, l'amplificateur différentiel serait impossible à réaliser. L'amplificateur différentiel (peut-être le montage électronique le plus important) sera abordé dans le prochain article de cette série…

230372-04

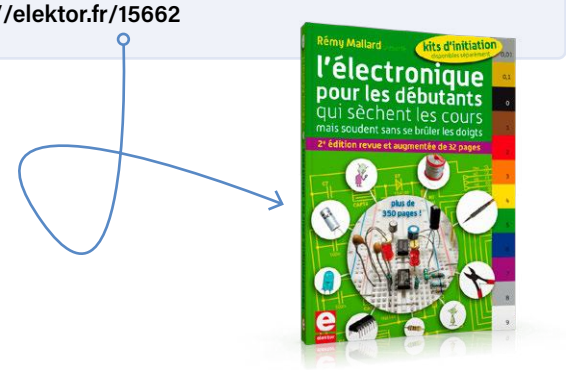
Note de la redaction : la série d'articles « démarrer en électronique » est basée sur le livre « Basiskurs Elektronik » de Michael Ebner, publié par Elektor.

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

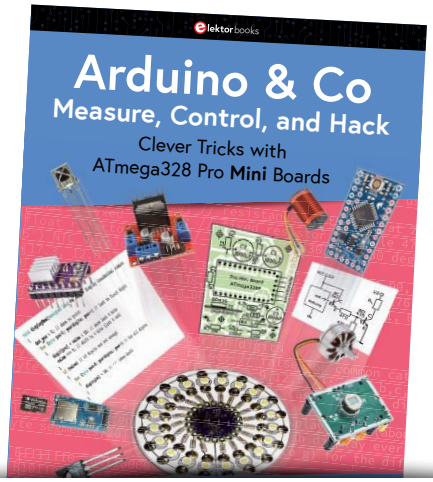
Produits

> **Rémy Mallard,** *L'électronique pour les débutants* **(livre de poche) (Elektor, 2e édition) https://elektor.fr/15662**



enregistreur d'infrasons avec l'Arduino Pro Mini

un projet extrait du livre « Arduino & Co. » d'Elektor



Robert Sontheimer (Allemagne)

Réalisons un projet inédit : combinons un Arduino Pro Mini (ou une carte compatible) avec un capteur de pression atmosphérique BMP180 et un module de carte SD, et étudions l'enregistrement à long terme des niveaux d'infrasons.

Note de la rédaction. *Cet article est un extrait du livre de 332 pages Arduino & Co — Measure, Control, and Hack, formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. L'auteur et l'éditeur ont fait de leur mieux pour l'éviter et seront heureux de répondre aux questions – pour les contacter, voir l'encadré* **« Questions ou commentaires ? »**.

Le capteur de pression atmosphérique BMP180 délivre un maximum d'environ 100 valeurs de température et de pression atmosphérique par seconde. Nous pourrions donc générer un fichier audio stéréo (au format PCM avec l'extension .wav) à partir de ces données, sur des minutes, des heures ou des jours, en enregistrant la pression atmosphérique sur le canal gauche et la température sur le canal droit. Nous réglons la fréquence de lecture régulière (taux d'échantillonnage) sur 44 100 Hz, par exemple. Il s'agit d'une valeur courante qui est également utilisée pour les CD audio et bien d'autres choses. Ainsi, le fichier sera lu plus tard (par exemple sur un ordinateur portable ou avec un lecteur audio) à 441 fois la vitesse normale, et nous pourrons entendre des infrasons ! Avec la fréquence d'échantillonnage originale de 100 Hz lors de l'enregistrement, on peut enregistrer toutes les fréquences inférieures à 50 Hz. Un infrason de 10 Hz devient 4,41 kHz lors de la lecture. 1 Hz devient 441 Hz, etc. Théoriquement, il n'y a pas de limites en dessous.

Il est vrai que le BMP180 n'est pas vraiment adapté à un microphone très sensible. Bien qu'il détecte déjà les écarts de pression lors de faibles changements d'altitude, les infrasons doivent être assez forts pour être entendus malgré le bruit pendant la lecture. Cependant, il est beaucoup plus intéressant d'utiliser un logiciel d'édition audio pour examiner les formes d'onde de la pression atmosphérique et de la température plus tard - surtout si l'enregistrement s'est déroulé sur plusieurs jours. Le canal de gauche présente alors la courbe de la pression atmosphérique, et celui de droite, la courbe de la température.

Ici, j'ai simplement enregistré pendant près de trois jours, ouvert le fichier .wav final avec un programme d'édition audio, puis normalisé les deux canaux séparément, amplifié autant que pour que la gamme complète soit utilisée. Sur la **figure 1**, la courbe supérieure avec les relevés de pression semble encore un peu épaisse et irrégulière car chaque pixel sur l'axe X est constitué de dizaines de milliers de mesures qui contiennent également du bruit et fluctuent donc à un certain niveau.

La **figure 2** montre les résultats après traitement. L'enregistrement a été rééchantillonné et 4 000 valeurs ont été moyennées en une seule valeur. Le bruit a disparu et nous pouvons voir plus clairement les courbes de pression et de température.

Les graphes nous permettent de constater certaines choses : Sur la courbe de pression en haut, en termes de temps, il y a eu un peu plus de vent dans la deuxième moitié de la mesure, car de nombreuses petites fluctuations augmentent à ce moment-là. Lors d'une vraie tempête, les fluctuations seraient encore plus extrêmes.

La courbe de température en bas montre clairement le rythme jour-nuit. Les petits pics presque continus correspondent aux cycles de commutation du thermostat pour le chauffage de la pièce. On pourrait compter le nombre de fois que le thermostat s'est allumé et éteint au cours de ces trois jours. En outre, on constate

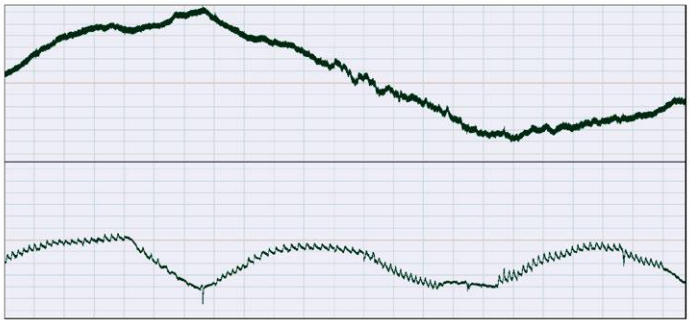


Figure 1. Progression du graphique de la pression atmosphérique (en haut) et de la température (en bas), essentiellement non traitée.

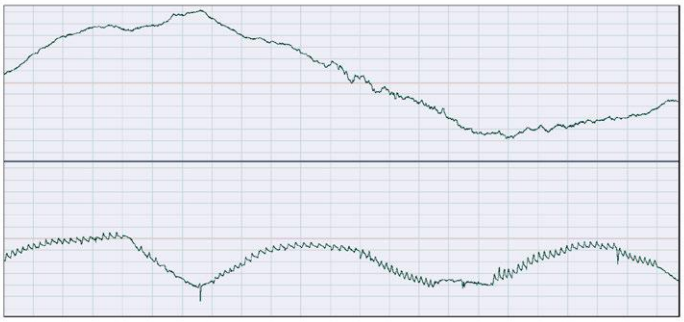


Figure 2. Évolution graphique de la pression atmosphérique (en haut) et de la température (en bas), traitée.

deux déviations rapides vers le bas : la fenêtre a été ouverte pendant une courte période.

Construction

Pour réaliser le projet, nous avons besoin de :

- 1 carte Arduino (16 MHz ATmega328)
- 1 module de capteur BMP180
- 1 module de carte (Micro)SD
- 1 bouton-poussoir
- fils ou cavaliers
- 1 carte (Micro)SD, max. 32 GB

La construction est assez simple. En plus du capteur BMP180, il faut connecter le module de carte SD, ainsi que le bouton poussoir pour démarrer et arrêter l'enregistrement. Il suffit de respecter le schéma de connexion de la **figure 3**. Les fils soudés sont toujours préférables, mais nous pouvons aussi utiliser des cavaliers. Ensuite, nous avons besoin de deux lignes VCC, mais le Pro Mini n'a qu'une seule broche VCC. La solution est très simple : puisque le BMP180 ne consomme pratiquement pas d'énergie, nous pouvons également l'alimenter avec une sortie (par exemple, avec la broche 9 au lieu de VCC). Il suffit alors de configurer la broche en sortie dans `setup()` et de la mettre à l'état haut.

Croquis de l'enregistreur d'infrasons

Une importante partie du croquis *13.9.ino* écrit pour ce projet sera présentée dans les listages numérotés **listage 1a** à **1h** ainsi que les paragraphes ci-dessous. Le programme complet est trop long pour être présenté ici avec des listages partiels. Le programme et les sous-programmes mentionnés ci-dessous sont contenus dans le fichier d'accompagnement du logiciel publié pour le livre, disponible en téléchargement gratuit sur la page d'Elektor [1]. Sur la page web, cliquez sur l'onglet *Téléchargements*. Pour pouvoir suivre cet extrait, accédez au fichier *13.9.ino*.

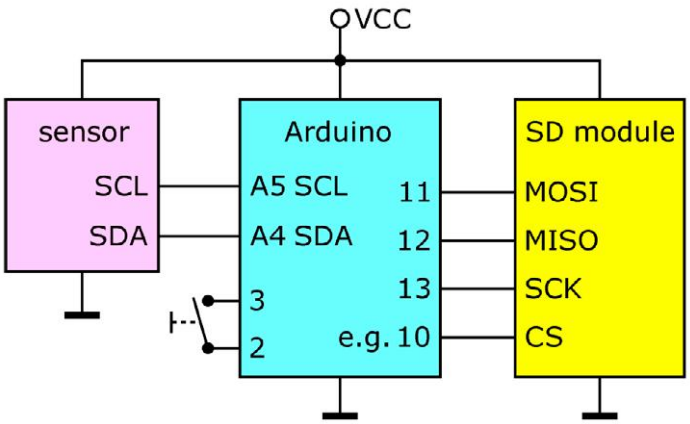


Figure 3. Schéma de connexion pour le capteur BMP180, le module de carte SD, l'interrupteur et la carte Arduino.

Listage 1a : comme nous combinons le capteur de pression atmosphérique avec le module de carte SD, nous devons inclure trois bibliothèques : l'interface I²C pour le capteur de pression atmosphérique, l'interface SPI pour le module de carte SD et les fonctions spéciales de la carte SD.

Ensuite, nous définissons deux broches où nous pouvons connecter un bouton poussoir pour démarrer et arrêter l'enregistrement. La broche 3 est commutée au niveau bas et sert de masse pour le bouton. Cette méthode est très pratique, car les deux broches du bouton sont alors très proches l'une de l'autre. Bien sûr, nous pourrions aussi utiliser la vraie masse (GND) à la place de la broche 3.

Les autres définitions et variables, ainsi que toutes les parties du programme du capteur de pression d'air et les croquis de la carte SD discutés dans le livre, ne sont pas montrés ici. Nous ne nous intéressons maintenant qu'au code relatif à l'enregistreur.

Listage 1a

```
#include <Wire.h>     // library for I2C interface
#include <SPI.h>      // library for SPI interface
#include <SD.h>       // library for SD card
#define record_pin 2 // switch pin to ground for
                      //   recording
#define low_pin 3     // Set output pin to LOW
                      //   (used as GND for button)
…
```

Listage 1b : ici, les paramètres de l'enregistreur sont présentés. La variable `oversampling` détermine également le nombre de mesures de pression. Pour enregistrer 100 valeurs par seconde, ce paramètre doit être 0. `audio_rate`, spécifie le nombre de valeurs par seconde qui seront transmises lors de la lecture ultérieure de l'enregistrement sous forme de fichier audio. `max_file_size` détermine la longueur maximale de l'enregistrement. Cependant, les 10 minutes font référence au temps de lecture – le temps d'enregistrement correspondant est de plusieurs jours. Cependant, nous pouvons modifier cette valeur de manière presque arbitraire. Lorsque la taille maximale du fichier est atteinte, le fichier est fermé et un nouvel enregistrement est lancé.

La variable `duration` détermine le délai entre les mesures. Cependant, les durées sont ici plus longues de 500 µs car nous travaillons plutôt avec une horloge fixe, et les instructions `Wire` pour la mesure de la température et de la pression sont également exécutées.

Listage 1b

```
// 0 -> 1x, 1 -> 2x, 2 -> 4x, 3->8x
char oversampling = 0;
// sampling rate of the output audio file
long audio_rate = 44100;
// 10 minutes' playback time (at 44,100 Hz)
unsigned long max_file_size = 105840044;
// time (µs) according to oversampling
int duration[4] = ;
// time for temperature measurement
int t_duration = 5000;
…
```

Listage 1c : Après les variables spéciales pour le calcul de la température et de la pression, que nous avons encore sautées ici, ces variables seront nécessaires plus tard pour l'enregistreur.

Listing 1c

```
// system time at which the data are available
unsigned int next_time;

long zero_value;  // pressure zero line
long record_value;// pressure value to save
long zero_temp;   // temperature zero line
long temp_value;  // temperature value to save

// indicates whether recording is activated
boolean must_record = false;
// indicates if recording is currently running
boolean is_recording = false;
// indicates whether pushbutton is pressed
boolean pressed = false;
// actual file size
unsigned long file_size;
// actual file number
unsigned int file_number = 0;
// counts how long button is not pressed anymore
byte state_counter = 0;
File wave_file;   // recording file
```

Listage 1d : `setup()` commence par la définition des broches du bouton-poussoir. Grâce à la résistance pull-up interne active, nous n'avons pas besoin d'autres composants.

Listing 1d

```
void setup() {
  // switch pin on input with pull-up
  pinMode(record_pin, INPUT_PULLUP);
  // low pin on output ...
  pinMode(low_pin, OUTPUT);
  // ... and LOW [...]
  digitalWrite(low_pin, LOW);
  …
}
```

Listage 1e : Dans `loop()`, nous n'avons que les appels alternés des fonctions de mesure de la température et de la pression, car pour gagner du temps, nous n'attendons pas à chaque fois les données, mais nous utilisons le temps d'attente pour exécuter les tâches suivantes. Pour la mesure de la température, on appelle la fonction `calculate()` en attendant le résultat. Pour la mesure de la pression, on appelle la fonction `recording()`.

Listing 1e

```
void loop() {
  get_t(); // read temperature
  get_p(); // read pressure
}
```

Listage 1f : Examinons la fonction `calculate()`. La température et la pression actuelles sont d'abord déterminées à partir des valeurs mesurées, si la pression a déjà été mesurée. J'ai à nouveau abrégé cette fonction par «…», car je suis sûr que personne ne souhaite relire ces calculs (de nombreuses formules de la fiche technique du BMP180).
Ensuite, nous vérifions si les références zéro pour la pression et la température sont déjà définies. Si ce n'est pas le cas, la première mesure est alors définie comme valeur de base. Si, au contraire, aucune mesure n'a encore été effectuée, on quitte complètement la fonction.
Sinon, les valeurs sont maintenant prêtes pour l'enregistrement. Pour ce faire, nous soustrayons la valeur du zéro de la pression. Si la valeur est dépassée, elle se limite à l'intervalle admissible de -32,768 à 32,767. Nous procédons ensuite exactement de la même manière pour la température. Les deux valeurs sont alors prêtes à être enregistrées.

Listage 1f

```
// calculate temperature (in tenth degree) and
// pressure (in pascals)
void calculate() {

  // if pressure has already been measured before
  if (p) {
    …
  }

  // if zero value has not been determined yet
  if (!zero_value) {
    // cancel if no measurement has been taken yet
    if (!pressure) return;
    // actual value as zero line
    zero_value = pressure;
    // actual value as zero line
    zero_temp = b5;
  }

  // pressure value for recording
  record_value = pressure - zero_value;
```

```
  // max value if clipping
  if (record_value > 32767)
    record_value = 32767;
  // negative clipping
  else if (record_value < -32768)
    record_value = -32768;
  // temperature value for recording
  temp_value = b5 - zero_temp;
  // max value if clipping
  if (temp_value > 32767) temp_value = 32767;
  // negative clipping
  else if (temp_value < -32768) temp_
  value = -32768;
}
```

Listage 1g : Dans la fonction `recording()`, deux conditions sont d'abord vérifiées : si l'enregistrement doit être exécuté (parce qu'il est commencé et qu'il n'a pas encore atteint la taille maximale du fichier) et si l'enregistrement est effectivement en cours d'exécution. Ces deux distinctions donnent lieu à quatre possibilités. Dans le premier cas, l'enregistrement devrait être en cours d'exécution, mais il ne l'est pas encore.

Listage 1g

```
void recording()  { // recording function
  // if recording should run
  if (must_record && file_size < max_file_size) {
    // if recording is not running yet
    if (!is_recording) {
      …
    }
  }
}
```

Listage 1h : Ici, un nouvel enregistrement est lancé. Le numéro de fichier est incrémenté dans la boucle do-while jusqu'à ce qu'on obtienne (avec le texte supplémentaire) un nom de fichier qui n'existe pas encore. Un nouveau fichier est alors ouvert avec le nom de fichier correspondant, et un message correspondant est également édité en série. En cas d'erreur, un autre message est émis et le processus est arrêté avec une boucle infinie.
Nous avons donc atteint la limite de l'espace alloué à la discussion sur le programme dans cet article. Heureusement, le reste du programme permettant d'utiliser l'enregistreur d'infrasons est bien documenté, tout comme les listages 1a à 1h présentées ici.

Listage 1h

```
// search for the first not yet used
// recording number
do {
  file_number++; // next number (starting with 1)
} // repeat with next number while
   //file already exists
while (SD.exists("FILE_" + String(file_number)
       + ".wav"));
Serial.println("Recording no."
               + String(file_number)
               + " is started!");
wave_file = SD.open("FILE_"
                    + String(file_number)
                    + ".wav", FILE_WRITE);

if (!wave_file) { // if file could not be opened
   Serial.println
      ("Error: File could not be opened!");
// do not continue (infinite loop)
    while (true);
}
```

Utilisation

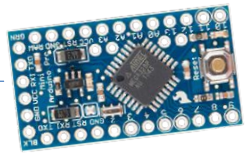
Au début du croquis, nous pouvons définir trois variables :

- `oversampling` doit être fixée à 0 pour utiliser la fréquence d'enregistrement la plus rapide de 100 Hz.
- `audio_rate` spécifie la fréquence de lecture ultérieure. On peut également utiliser une valeur inférieure à 44 100, afin que le facteur temps (44 100 / 100 = 441) ne soit pas si élevé. Les valeurs courantes sont 8 000, 11 025, 16 000, 22 050, 32 000, 44 100 et 48 000. Pas tous les lecteurs peuvent jouer d'autres fréquences.
- avec `max_file_size`, on définit la taille maximale du fichier. 105 840 044 correspond à une durée de lecture de 10 minutes à 44 100 Hz, mais à une durée d'enregistrement de plus de trois jours.

Il faut ensuite «normaliser» l'enregistrement final, c'est-à-dire augmenter le volume du signal pour utiliser toute la gamme, mais sans dépasser le maximum. Sans normalisation, le signal sera trop faible. Bien sûr, nous pourrions ajouter un facteur de gain lors de l'enregistrement, mais nous ne savons pas à l'avance à quel point la température et la pression atmosphérique varieront pendant l'enregistrement. Il est possible d'effectuer la normalisation en quelques clics avec un programme d'édition audio. Elle doit être effectuée séparément pour les deux canaux. Avec un tel programme, nous pouvons également visualiser l'enregistrement sous forme de graphiques de mesure. Le canal de gauche montre l'évolution de la pression atmosphérique pendant l'enregistrement, tandis que le canal de droite montre la température.

Enregistreur météorologique

Si nous changeons la valeur de `oversampling` à 3, seulement un 32 bonnes valeurs par seconde (au lieu de 100) seront enregistrées, ce qui est encore beaucoup plus que ce dont nous avons besoin pour les enregistrements météorologiques. Nous pourrions également combiner l'altimètre (de la section 13.8 du livre) avec les fonctions de la carte SD pour que nous puissions écrire une ligne de texte correspondante dans un fichier pour chaque sortie série également. Ainsi, nous obtenons l'enregistrement météorologique sous forme de fichier texte.



Les chapitres du livre proposent, outre des projets concrets, de nombreuses informations et suggestions qui devraient vous aider à réaliser vos propres idées.
Nous sommes arrivés à la fin de cet article, mais je vous souhaite beaucoup de plaisir dans la mesure, le contrôle et le piratage !

230393-04

Questions ou cmmentaires ?

Envoyez un courriel à l'auteur (r.sont@freenet.de) ou contactez Elektor (redaction@elektor.fr).



Produits

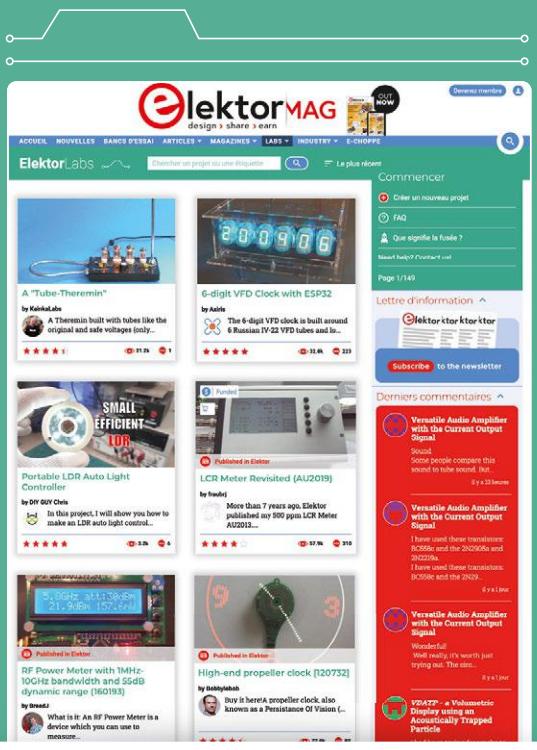
- **Robert Sontheimer, *Arduino & Co., Measure, Control, and Hack*, Elektor 2022**
  version papier : https://elektor.fr/20243
  version numérique : https://elektor.fr/20244

LIENS

[1] Logiciels pour le livre : https://elektor.fr/20243

À propos de l'auteur

Robert Sontheimer a été immédiatement à bord il y a environ 40 ans, avec son ZX81 et son C64 lorsque les premiers ordinateurs familiaux sont arrivés à nos domiciles. À l'époque, il a transformé un traceur en scanner, et il a eu d'autres idées originales et bizarres. Aujourd'hui, il utilise des Arduino Pro Minis pour contrôler des machines laser CNC entières et il a même inventé un système d'aspiration adapté : son «filtre à papier hygiénique auto-changeant». Dans son bureau, il a un aimant qui lévite depuis des années - bien entendu contrôlé par un Pro Mini.



Stimulez vos innovations en électroniques avec

ElektorLabs

- Partage gratuit de projets
- Soutien d'experts
- Opportunités de collaboration
- Accès à des ressources exclusives
- Publication dans la magazine Elektor

Partagez vos projets dès maintenant !
www.elektormagazine.fr/e-labs

elektor
design > share > earn



Source: Shutterstock

compteur d'énergie basé sur le cloud

avec un module ESP32 et un capteur de tension/courant PZEM-004T

Un projet d'Elettronica In
https://elettronicain.it

Emanuele Signoretta (Italie)

Avec l'augmentation constante des tarifs de l'électricité, la consommation rationnelle et les économies sont devenues une nécessité. Avec un module ESP32 et quelques autres composants, il est possible de réaliser un compteur d'énergie qui envoie nos données énergétiques à la plateforme InfluxDb Cloud via une connexion sur le réseau wifi.

L'Italie, comme beaucoup d'autres pays, dépend fortement des sources d'énergie étrangères parce que sa propre production d'énergie renouvelable n'est pas suffisante pour répondre à la demande nationale. Le problème s'aggrave lorsque des facteurs externes limitent la disponibilité des sources d'énergie fossiles, ce qui entraîne une hausse des prix. Par conséquent, les Italiens, tout comme nous, doivent réduire leur consommation d'énergie, sacrifier certains confort et surveiller de près leur consommation d'énergie afin d'éviter les surprises désagréables sur leurs factures d'électricité et de gaz. La principale motivation pour réduire la consommation d'énergie est la peur de dépenser trop, plutôt que les préoccupations liées au changement climatique.
Pour aider les personnes soucieuses de leur consommation d'électricité, cet article présente un compteur de consommation d'énergie basé sur un module ESP32 d'Espressif et un capteur de tension/courant PZEM-004T. Le compteur collecte des données et les envoie au cloud, plus précisément aux serveurs du service en ligne InfluxDB.

Matériel

Pour le matériel du compteur d'énergie, nous avons besoin d'un module ESP32 Wi-Fi (**figure 1**) et d'un capteur PZEM-004T (que l'on voit encapsulé dans son boîtier en plastique transparent dans la **figure 2**). Nous avons également besoin d'une alimentation à découpage avec une sortie de 5 V (si vous en choisissez une avec une sortie Micro USB, vous pouvez l'utiliser pour alimenter la carte ESP et fournir de l'énergie



Figure 1. Carte ESP32.



Figure 2. Le capteur PZEM-004T avec un transformateur.

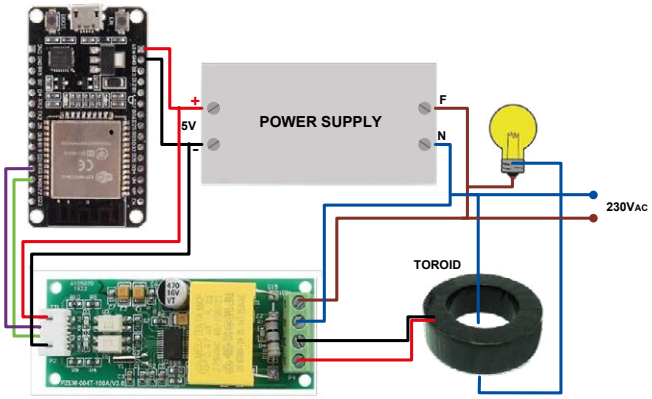


Figure 3. Schéma du circuit.

au PZME-004T à partir du VIN), des cavaliers femelle-femelle pour l'Arduino, des borniers, quelques câbles électriques, et un boîtier en plastique [2]. La fiche technique du capteur est disponible sur [3], tandis que la **figure 3** montre le câblage général du circuit – nous donnons plus de détails à ce sujet dans la partie **Installation** ci-dessous.

Le module comprend un circuit de détection du courant et de tension, logé dans un boîtier en plastique avec un tore ouvert. Il intègre une électronique simple, isolée par des optocoupleurs, et dispose d'une interface série. Le module peut mesurer avec précision des tensions allant de 80 V à 260 V, avec une résolution de 0,1 V et une précision de 0,5 %. De même, il peut mesurer des courants de 0 A à 100 A avec une précision de 0,5 % et une résolution de 0,001 A. En outre, le capteur peut déterminer l'angle de phase (facteur de puissance) entre les vecteurs de tension et de courant avec une précision de 1 %, fournissant des relevés de puissance active et réactive. Cette fonctionnalité nous permet d'évaluer l'efficacité électrique de l'appareil testé. Les composants du circuit ont été sélectionnés selon des critères de rentabilité, compte tenu de la crise actuelle des semi-conducteurs, et des spécifications impressionnantes de l'ESP32.

Configuration d'InfluxDB

De multiples logiciels sont disponibles pour recevoir et analyser les données transmises par les appareils IdO. Parmi les options les plus remarquables, on peut citer Home Assistant, Grafana, Blynk, Thingspeak, et bien d'autres encore. Dans cet article, nous nous concentrerons sur InfluxDB, dont le logo est présenté dans la **figure 4**.

InfluxDB est un logiciel polyvalent qui offre des fonctionnalités telles que la création de tableaux de bord, l'exécution de requêtes et l'envoi d'alertes. Il offre de multiples options de déploiement, notamment des versions exécutables pour diverses architectures, des conteneurs Docker et des solutions basées sur le cloud. Bien qu'initialement nous



Figure 4. Logo d'InfluxDB.

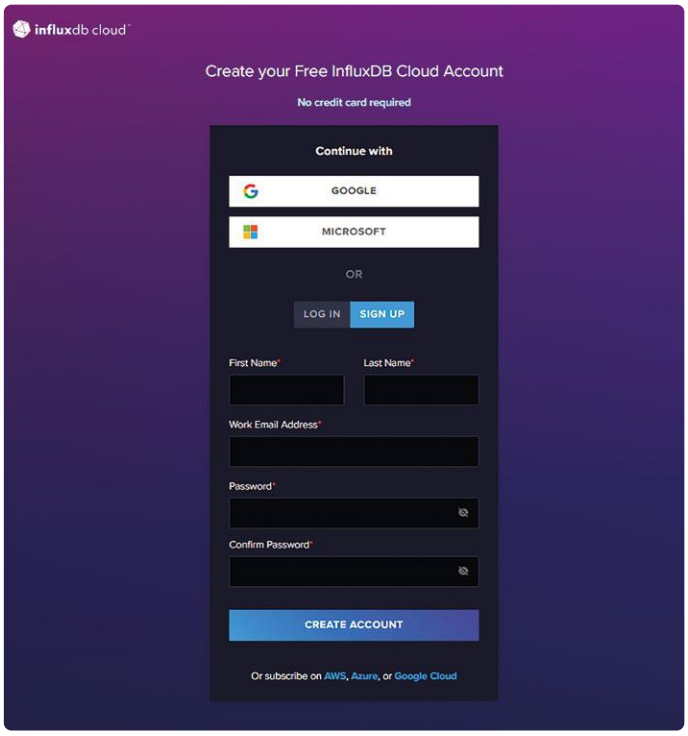


Figure 5. Page d'inscription à InfluxDB

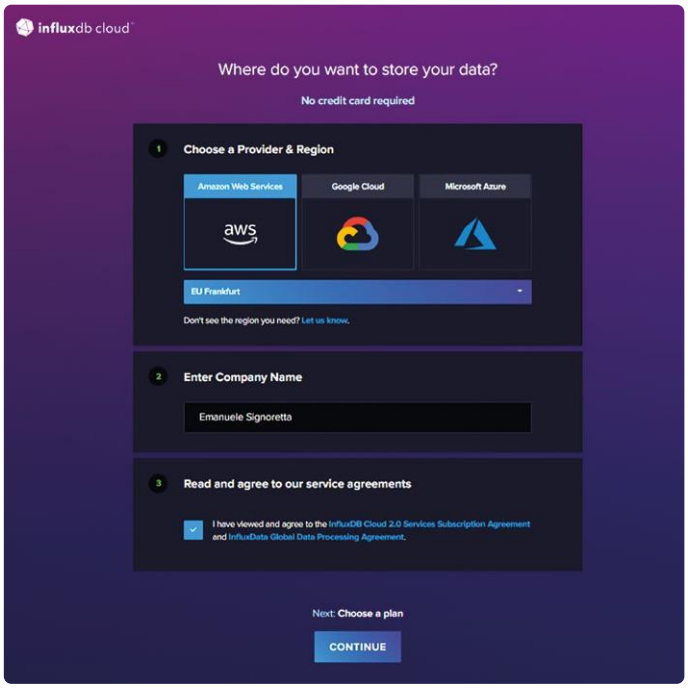


Figure 6. Choix du fournisseur et acceptation des conditions de service.

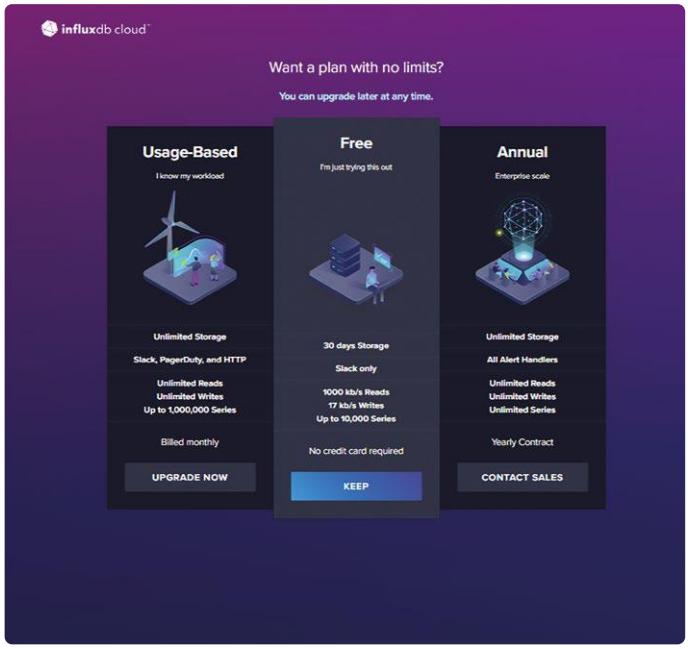


Figure 7. Choix de la formule d'abonnement.

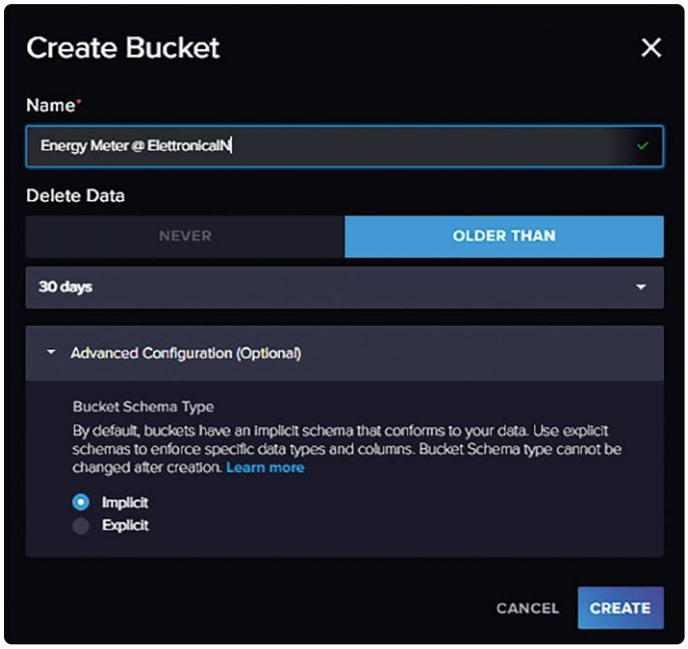


Figure 8. Création d'un nouveau bucket.

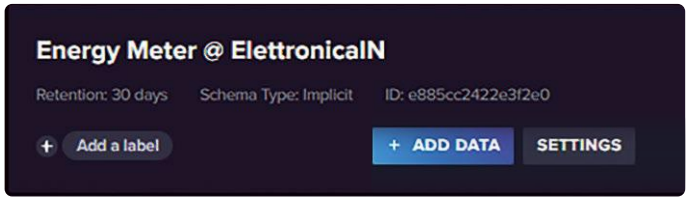


Figure 9. Un bucket récemment créé.

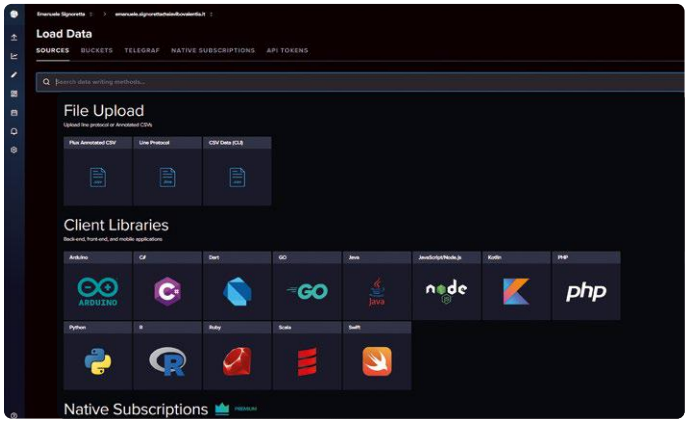


Figure 10. Sélection de la source de chargement de données.



Figure 11. Extrait d'un croquis de démonstration.

ayons eu l'intention d'installer une instance sur un Raspberry Pi pour des raisons de propriété de données, la rareté des composants nous a obligés à adopter plutôt le service InfluxDb Cloud v2.
Pour démarrer, visitez le lien [4] et enregistrez-vous. Vous pouvez choisir de vous inscrire en utilisant vos identifiants Microsoft ou Google, ou de remplir manuellement les champs requis, comme le montre la **figure 5**. Après avoir terminé le processus d'inscription, vous devrez voir une page ressemblant à la **figure 6**, où il vous sera demandé de fournir divers détails, y compris le nom du fournisseur de services. Pour notre projet, nous avons sélectionné Amazon Web Services (AWS) comme fournisseur.
Sur la page suivante (illustrée à la **figure 7**), vous devrez sélectionner un plan d'utilisation. Dans notre cas, nous avons opté pour le plan gratuit, qui permet de stocker les données pour une durée de 30 jours et de recevoir des notifications via Slack.
Après avoir confirmé notre sélection, le tableau de bord personnel s'affiche. Pour créer un bucket (conteneur), naviguez vers *Load Data Buckets Create new bucket*. En procédant ainsi, une page similaire à celle de la **figure 8** s'ouvrira. Remplissez le champ *Name* et cliquez sur le bouton *Create*. Une fois le bucket créé avec succès, il apparaîtra parmi les sources de données disponibles, comme le montre la **figure 9**. Cliquez ensuite sur *Add data Client library*. Une nouvelle page s'affiche, proposant différentes options pour le chargement de données. Parmi ces options, sélectionnez *Arduino* (**figure 10**). Dans la fenêtre suivante (**figure 11**), vous verrez une série d'extraits de code qui constituent un croquis de démonstration. Copiez les données fournies pour `INFLUXDB_URL`, `INFLUXDB_ORG`, et `INFLUXDB_BUCKET` à partir du premier extrait.
Pour terminer la configuration, nous devons générer un jeton d'accès pour le bucket. Pour ce faire, accédez à la barre latérale de gauche et naviguez à *Load Data API Tokens*.
Dans la fenêtre qui vient de s'ouvrir, cliquez sur *Generate API Token* (Générer un jeton API). Après avoir sélectionné le bucket, activez les autorisations de lecture et d'écriture, comme indiqué dans la **figure 12**. Une fois le jeton généré, copiez-le et enregistrez-le, car nous en aurons besoin dans les étapes suivantes.

Croquis

Le croquis de notre projet est une combinaison de plusieurs autres croquis provenant des bibliothèques que nous avons utilisées. Vous pouvez télécharger les fichiers du croquis depuis le dépôt GitHub [5]. Analysons maintenant notre code, qui est divisé en trois sections représentées avec des listages correspondants.

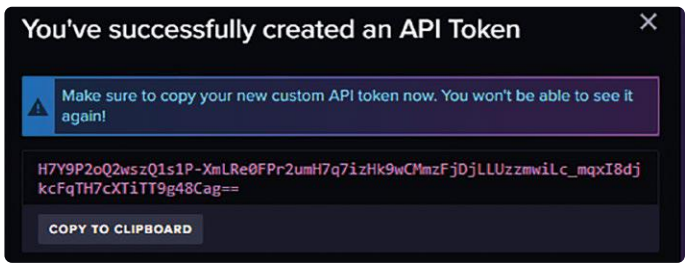


Figure 12. Création d'une clé API.

Listage 1. Inclusion des bibliothèques

```
#include <WiFiMulti.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <InfluxDbClient.h>
#include <InfluxDbCloud.h>
#include <PZEM004Tv30.h>
#include <Every.h>
```

Commençons par l'inclusion des bibliothèques et dépendances nécessaires, comme le montre le **listage 1**.
À cet égard, il est important de noter que la bibliothèque *WiFiMulti* est utilisée pour gérer la communication wifi entre l'ESP32 et le point d'accès. Les bibliothèques *ESPmDNS*, *WiFiUDP*, et *ArduinoOTA* [6] sont utilisées pour le chargement à distance (*OTA*) des croquis et la gestion des noms d'hôtes. Veillez à ce que la version 2.7.x de Python soit installée sur votre PC pour utiliser cette fonctionnalité. Les bibliothèques *InfluxDbClient* et *InfluxDbCloud* sont utilisées pour le transfert de données vers InfluxDB Cloud. La bibliothèque *PZEM004Tv30* [7] facilite la communication série avec le capteur. Enfin, la bibliothèque *Every* permet l'exécution des blocs de code à intervalles réguliers sans avoir recours à la fonction `delay()`.
Examinons maintenant les extraits de code suivants, en commençant par le **listage 2**, qui contient la section pré-processeur. En utilisant `#define REFRESH_TIME 5000`, nous définissons l'intervalle (en millisecondes) pour le chargement de données vers le cloud. Nous définissons ensuite le nom de l'appareil, les broches de communication série et le port série choisi. Les objets relatifs au réseau wifi et au capteur sont instanciés. Les adresses IP sont définies, et si vous souhaitez configurer une connexion au point d'accès avec une IP statique, vous pouvez modifier les paramètres en fonction du sous-réseau de votre réseau. Enfin, nous définissons les paramètres pour la connexion wifi et l'accès à InfluxDB Cloud. Remplacez ces lignes de code par l'extrait copié et saisissez les données manquantes.
Nous spécifions respectivement le SSID et la phrase de passe associés au réseau wifi avec `#define WIFI_SSID` et `#define WIFI_PASSWORD`. Ensuite, nous définissons les paramètres de connexion à InfluxDB Cloud. Le seul paramètre manquant est `INFLUXDB_TOKEN`. Vous pouvez le récupérer à partir de la clé API que nous avons créée précédemment. Insérez la clé API entre les guillemets. Nous spécifions le fuseau horaire de l'Europe centrale pour les horodatages avec `#define TZ_INFO "CET-1CEST,M3.5.0,M10.5.0/3"`.
Pour créer l'objet client permettant de se connecter au serveur InfluxDB, utilisez le code suivant :

```
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG,
                      INFLUXDB_BUCKET, INFLUXDB_TOKEN,
InfluxDbCloud2CACert);
```

Listage 2. Définitions

```
#define REFRESH_TIME 5000 // Delay interval for data upload
#define DEVICE "ESP32"
#define PZEM_RX_PIN 27
#define PZEM_TX_PIN 26
#define PZEM_SERIAL Serial2

/*************************
ESP32 initialization
---------------------

   The ESP32 HW Serial interface can be routed to any GPIO pin
   Here we initialize the PZEM on Serial2 with RX/TX pins 26 and 27
*/
PZEM004Tv30 pzem(PZEM_SERIAL, PZEM_RX_PIN, PZEM_TX_PIN);
WiFiMulti wifiMulti;
IPAddress local_IP(192, 168, 178, 154);
IPAddress gateway(192, 168, 178, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress primaryDNS(192, 168, 178, 1); //optional
IPAddress secondaryDNS(1, 1, 1, 1); //optional
// WiFi AP SSID
#define WIFI_SSID ""
// WiFi password
#define WIFI_PASSWORD ""
// InfluxDB v2 server url, e.g. https://eu-central-1-1.aws.cloud2.influxdata.com
 // (Use: InfluxDB UI -> Load Data -> Client Libraries)
#define INFLUXDB_URL "https://eu-central-1-1.aws.cloud2.influxdata.com"
// InfluxDB v2 server or cloud API token
 //   (Use: InfluxDB UI -> Data -> API Tokens -> Generate API Token)
#define INFLUXDB_TOKEN ""
// InfluxDB v2 organization id (Use: InfluxDB UI -> User -> About -> Common Ids )
#define INFLUXDB_ORG ""
// InfluxDB v2 bucket name (Use: InfluxDB UI ->  Data -> Buckets)
#define INFLUXDB_BUCKET "Energy Meter @ ElettronicaIN"
// Set timezone string according to
//    https://www.gnu.org/software/libc/manual/html_node/TZ-Variable.html
#define TZ_INFO "CET-1CEST,M3.5.0,M10.5.0/3"
// InfluxDB client instance with preconfigured InfluxCloud certificate
InfluxDBClient client(INFLUXDB_URL, INFLUXDB_ORG,
    INFLUXDB_BUCKET, INFLUXDB_TOKEN, InfluxDbCloud2CACert);
// Data point
Point sensor("EnergyMeter");
```

Listage 3. setup()

```
void setup() {
  Serial.begin(115200);
  // Uncomment in order to reset the internal energy counter
  // pzem.resetEnergy()
  // Setup wifi
  WiFi.mode(WIFI_STA);
  //Comment to use DHCP instead of static IP
   if (!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {
    Serial.println("STA Failed to configure");
  }
```

```
  wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to wifi");
  while (wifiMulti.run() != WL_CONNECTED) {
    //Serial.print(".");
    //delay(100);
    Serial.println("Connection Failed! Rebooting...");
    delay(5000);
    ESP.restart();
  }
  Serial.println();
  // Port defaults to 3232
  // ArduinoOTA.setPort(3232);
  // Hostname defaults to esp3232-[MAC]
  ArduinoOTA.setHostname("ESP32-Energy Meter");
  // No authentication by default
  // ArduinoOTA.setPassword("admin");
  // Password can be set with it's md5 value as well
  // MD5(admin) = 21232f297a57a5a743894a0e4a801fc3
  // ArduinoOTA.setPasswordHash("21232f297a57a5a743894a0e4a801fc3");
  ArduinoOTA
  .onStart([]() {
    String type;
    if (ArduinoOTA.getCommand() == U_FLASH)
      type = "sketch";
    else // U_SPIFFS
      type = "filesystem";
    // NOTE: if updating SPIFFS this would be the place to unmount SPIFFS using SPIFFS.end()
    Serial.println("Start updating " + type);
  })
  .onEnd([]() {
    Serial.println("\nEnd");
  })
  .onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Progress: %u%%\r", (progress / (total / 100)));
  })
  .onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
  });
  ArduinoOTA.begin();
  // Add tags
  sensor.addTag("Dispositivo", DEVICE);
  // Accurate time is necessary for certificate validation and writing in batches
  // For the fastest time sync find NTP servers in your area: https://www.pool.ntp.org/zone/
  // Syncing progress and the time will be printed to Serial.
  timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");\
  // Check server connection
  if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
  } else {
    Serial.print("InfluxDB connection failed: ");
    Serial.println(client.getLastErrorMessage());
  }
}
```

Enfin, l'objet *sensor* est créé avec `Point sensor("EnergyMeter")`. Cet objet, nommé *EnergyMeter*, sera associé à toutes les données collectées par le capteur. Le **listage 3** représente le code de la configuration de la carte et initialise le port série à une vitesse de transmission de 115 200.
Le bloc de code commence par `WiFi.mode(WIFI_STA)`, qui initialise le wifi en mode *station* afin de se connecter à un point d'accès. Le bloc de code suivant :

```
if (!WiFi.config(local_IP, gateway, subnet,
  primaryDNS, secondaryDNS)) {
 Serial.println("STA failed to configure");
}
```

définit une adresse IP statique au lieu d'utiliser le protocole DHCP. En cas d'erreur, un message d'avertissement est affiché via le port série. Si vous souhaitez utiliser DHCP, vous pouvez mettre cette partie du code en commentaire.
L'appel de la fonction `wifiMulti.addAP(WIFI_SSID, WIFI_PASSWORD)` effectue une tentative de connexion au point d'accès avec SSID et la phrase de passe spécifiés.
Une connexion à l'AP est alors tentée et, en cas d'échec, une erreur est d'abord affichée via le port série, puis la carte redémarre après un délai de 5 secondes.
Le nom d'hôte est défini avec `ArduinoOTA.setHostname("ESP32-Energy Meter")`, qui sera transmis via mDNS et affiché sur l'EDI Arduino pour le chargement du croquis à distance (OTA). Il y a du code commenté pour restreindre le chargement OTA du croquis aux utilisateurs possédant un nom d'utilisateur et un mot de passe. Décommentez ces lignes de code pour ajouter cette fonction de sécurité.
Les paramètres du mode OTA sont gérés par la séquence de code suivante :

```
ArduinoOTA.onStart([]() {
  String type;
  // …
  if (error == OTA_AUTH_ERROR) Serial.println("Auth
Failed");
  else if (error == OTA_BEGIN_ERROR) Serial.println("Begin
Failed");
  else if (error == OTA_CONNECT_ERROR) Serial.
println("Connect Failed");
  else if (error == OTA_RECEIVE_ERROR) Serial.
println("Receive Failed");
  else if (error == OTA_END_ERROR) Serial.println("End
Failed");
});
ArduinoOTA.begin();
```

Le bloc de code gère le démarrage, la fin, le chargement du croquis et les erreurs de la transmission OTA dans l'ordre. `ArduinoOTA.begin()` démarre le processus de chargement OTA.
Avec `sensor.addTag("Device", DEVICE)` ; un tag (étiquette) est défini, qui dans ce cas correspond au nom de l'appareil. Cette fonctionnalité peut être utile pour distinguer un ou plusieurs capteurs au sein d'une flotte.
L'ESP se connecte avec `timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov")` aux deux serveurs NTP (Network Time Protocol) *pool.ntp.org* et *time.nis.gov* afin d'obtenir la date et l'heure.
Enfin, le bloc de code tente de se connecter aux serveurs InfluxDB :

```
if (client.validateConnection()) {
  Serial.print("Connected to InfluxDB: ");
  Serial.println(client.getServerUrl());
}
else {
  Serial.print("InfluxDB connection failed: ");
  Serial.println(client.getLastErrorMessage());
}
```

Si la connexion est réussie, l'URL du serveur s'affiche sur le port série. Sinon, le message d'erreur correspondant s'affiche.
Passons maintenant au **listage 4**, qui contient le code de la boucle `loop()` de notre croquis. En utilisant `ArduinoOTA.handle()`, l'ESP32 attend la réception OTA de tout croquis compilé.
Dans `EVERY(REFRESH_TIME) { … }` le code entre crochets est exécuté à intervalles réguliers et sans délai. L'intervalle de temps a été défini au préalable. Dans le bloc `EVERY`, l'adresse du PZEM est d'abord lue et affichée sur le port série.
Le code du bloc `EVERY(REFRESH_TIME) { … }` est exécuté à des intervalles réguliers définis précédemment, sans utiliser de délais. Dans ce bloc, l'adresse du PZEM est lue et affichée sur le port série. Ensuite, on crée et initialise les variables des capteurs avec les fonctions de la bibliothèque du capteur. Ces variables sont notamment la tension, le courant, la puissance active (en Wh), l'énergie active (en kWh), la fréquence et le facteur de puissance (un rapport entre -1 et 1, représentant le rapport entre la puissance active et la puissance apparente). Si l'une des variables n'est pas un nombre, une erreur de lecture est affichée sur le port série. Dans le cas contraire, les relevés des capteurs seront affichés. Ensuite, les champs initialisés et les horodatages sont effacés et les variables sont préparées pour le chargement vers le cloud. Enfin, la connexion wifi et le bon chargement de données sont vérifiées. Si le chargement échoue, la dernière erreur est affichée sur le port série.

Installation

Nous avons prévu de placer l'alimentation, l'ESP32 et le module principal du capteur dans une boîte en plastique, en ne laissant passer que les fils de l'alimentation et ceux qui relient le tore du capteur de courant à la carte. Si vous voulez mesurer la consommation globale d'énergie de votre maison, l'entrée d'alimentation de notre système de mesure de l'énergie doit être connectée aussi près que possible de la ligne CA sortant du panneau électrique de la maison. Sinon, comme l'illustre la figure 3, il est toujours possible de mesurer l'énergie consommée par un seul appareil électrique. L'ouverture à charnière du tore permet de serrer très commodément le câble électrique à mesurer. Dans le cas de câbles électriques tripolaires (phase, neutre et terre), la gaine isolante extérieure doit être soigneusement coupée dans le sens de

Listage 4. loop()

```
void loop() {

  ArduinoOTA.handle();

  EVERY(REFRESH_TIME) {
    // Print the custom address of the PZEM
    Serial.print("Custom Address:");
    Serial.println(pzem.readAddress(), HEX);
    // Read the data from the sensor
    float voltage = pzem.voltage();
    float current = pzem.current();
    float power = pzem.power();
    float energy = pzem.energy();
    float frequency = pzem.frequency();
    float pf = pzem.pf();
    // Check if the data is valid
    if (isnan(voltage)) {
      Serial.println("Error reading voltage");
    } else if (isnan(current)) {
      Serial.println("Error reading current");
    } else if (isnan(power)) {
      Serial.println("Error reading power");
    } else if (isnan(energy)) {
      Serial.println("Error reading energy");
    } else if (isnan(frequency)) {
      Serial.println("Error reading frequency");
    } else if (isnan(pf)) {
      Serial.println("Error reading power factor");
    } else {
      // Print the values to the Serial console
      Serial.print("Voltage: ");      Serial.print(voltage);      Serial.println("V");
      Serial.print("Current: ");      Serial.print(current);      Serial.println("A");
      Serial.print("Power: ");        Serial.print(power);        Serial.println("W");
      Serial.print("Energy: ");       Serial.print(energy, 3);     Serial.println("kWh");
      Serial.print("Frequency: ");    Serial.print(frequency, 1); Serial.println("Hz");
      Serial.print("PF: ");           Serial.println(pf);
      ////UPLOAD DATA
      // Clear fields for reusing the point. Tags will remain untouched
      sensor.clearFields();
      // Store measured value into point
      sensor.addField("Tensione", voltage);
      sensor.addField("Corrente", current);
      sensor.addField("Potenza", power);
      sensor.addField("Energia", energy);
      sensor.addField("Frequenza", frequency);
      sensor.addField("Fattore di potenza", pf);
      // Print what are we exactly writing
      Serial.print("Writing: ");
      Serial.println(sensor.toLineProtocol());
      // Check WiFi connection and reconnect if needed
      if (wifiMulti.run() != WL_CONNECTED) {
        Serial.println("Wifi connection lost");
      }
      // Write point
      if (!client.writePoint(sensor)) {
        Serial.print("InfluxDB write failed: ");
        Serial.println(client.getLastErrorMessage());
      }
    }
    Serial.println();
  }
}
```

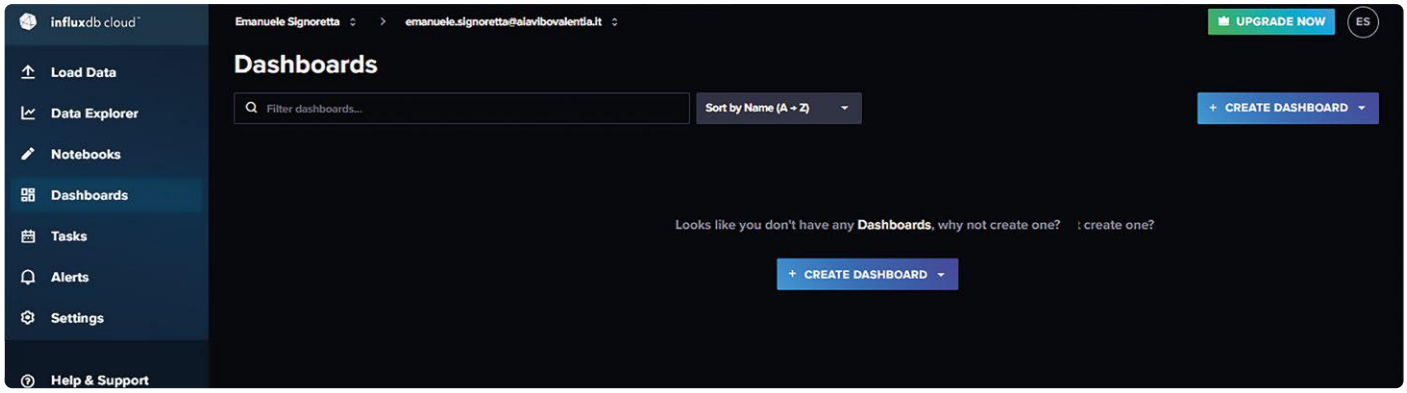


Figure 13. Création du tableau de bord.

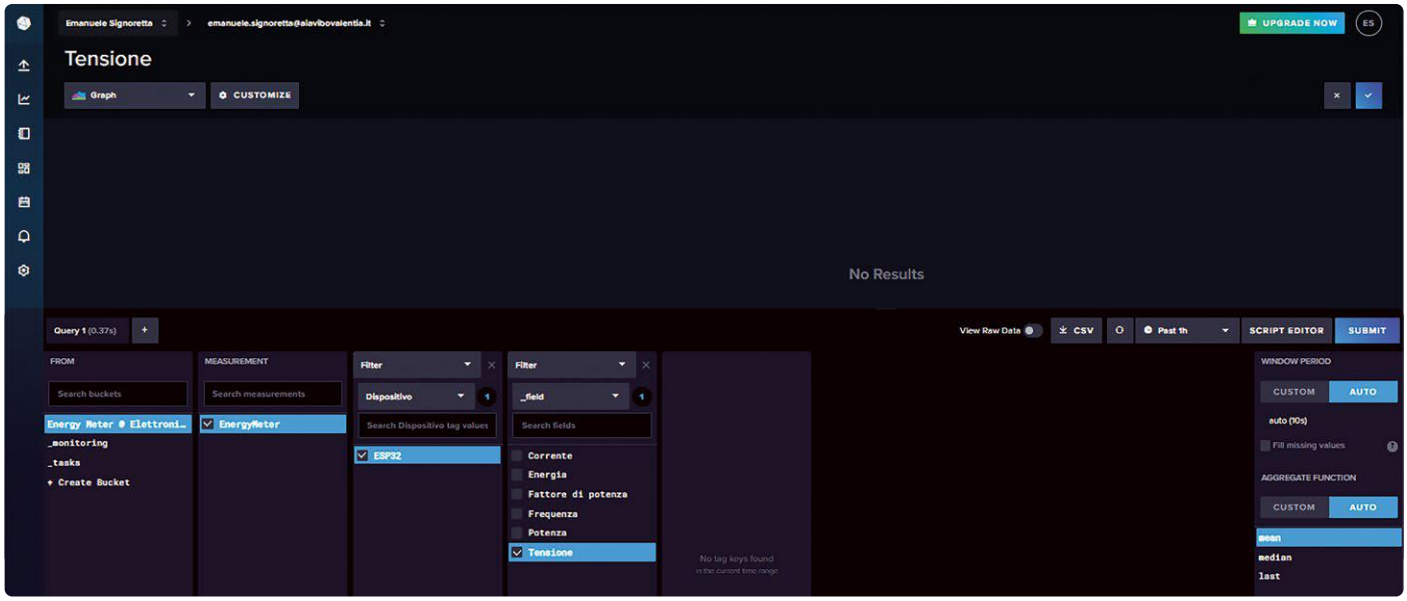


Figure 14. Ajout des graphiques du tableau de bord.

la longueur (**débranchez le au préalable !**), sans affecter l'isolation des trois câbles intérieurs. Ils seront séparés et seul le fil sous tension (normalement noir ou marron) sera placé à l'intérieur du tore de mesure. Pour installer le compteur d'énergie, il suffit de suivre le schéma déjà présenté au début de cet article, dans la figure 3.

En particulier, vous devez connecter deux fils en parallèle à l'entrée de l'alimentation et entre les broches N et L du PZEM-004T. **Soyez toujours très prudent, vous travaillez sur les fils du secteur 230 V(CA) !**

Ensuite, nous libérons le cran de sécurité du tore et faisons passer l'un des deux fils reliés au réseau électrique, puis nous connectons les fils du tore aux broches CT du capteur. En outre, nous connectons le module principal du capteur à l'ESP32, comme indiqué dans le **tableau 1**. Pour alimenter l'ESP, vous pouvez utiliser une alimentation avec une sortie Micro USB et fournir les 5 V pour le capteur à partir de VIN et GND. Une fois le croquis chargé, nous pouvons enfermer toute l'électronique dans la boîte et démarrer le système.

Démarrage et configuration du tableau de bord

Une fois la carte allumée, nous retournons sur le site web d'InfluxDB et, à partir de notre tableau de bord, nous accédons à *Dashboards Create new dashboard*. En procédant ainsi, nous obtenons un écran comme celui de la **figure 13**, où nous attribuons un nom à notre tableau de bord et cliquons sur *Add Cell*. Dans la nouvelle fenêtre qui apparaît (**figure 14**), nous allons sélectionner les paramètres à inclure dans chacun des graphiques constituant le tableau de bord. Nous sélectionnons, dans l'ordre, le bucket, les mesures, l'appareil et enfin les valeurs à afficher. Vous pouvez choisir les graphiques à utiliser pour les valeurs : carte thermique, jauge, graphique simple, tableau, etc. ainsi que les valeurs de mise à l'échelle à utiliser. On personnalise les cellules selon nos préférences et on répète la procédure pour chaque valeur que l'on souhaite afficher jusqu'à obtenir un résultat comme celui visible dans la **figure 15**, qui illustre notre tableau de bord complet.

Tableau 1. Connexions entre le compteur d'énergie et les modules ESP32

Broche du PZEM-004	Broche de l'ESP32
VCC	V5
GND	GND
RX	26
TX	27



Figure 15. Le tableau de bord complet.

Conclusion

Ceci conclut la description de notre simple mais puissant compteur d'énergie. La polyvalence de la carte ESP32 et les nombreuses caractéristiques des protocoles et plates-formes réseau pris en charge nous permettent de développer ses applications en fonction de nos besoins, par exemple en intégrant le compteur dans un système de gestion de charge électronique, tout en conservant sa capacité d'action autonome.

230279-04

Produits

- **ESP32-C3-DevKitM-1**
  https://elektor.fr/20324
- **Koen Vervloesem, *Getting Started with ESPHome*, Elektor 2021**
  https://elektor.fr/19738
- **Offre groupée : Livre « *Getting Started with ESPHome* » + LILYGO TTGO T-Display ESP32 (16 Mo)**
  https://elektor.fr/19896

Questions ou commentaires?

Contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

Emanuele Signoretta est né en 2000 à Vibo Valentia, une petite ville du sud de l'Italie. Il est passionné par les TIC. Étudiant, il a découvert Arduino et sa simplicité de programmation. Il a commencé par écrire du code pour les cartes Arduino et Fishino. Aujourd'hui, il s'intéresse aux cartes STM32 et ESP32. Il croit en l'éthique des logiciels libres et utilise des distributions basées sur Linux. Emanuele termine actuellement sa licence en ingénierie électronique et de communication au Politecnico di Torino. Il travaille également pour Rai, le radiodiffuseur national italien.

LIENS

[1] Carte ESP32 : https://futuranet.it/prodotto/esp32-scheda-di-sviluppo-32-gpio-con-wifi-e-bluetooth
[2] Boîtier en plastique : https://futuranet.it/prodotto/contenitore-plastico-ermetico-546x784x1182-mm
[3] Fiche technique du PZEM-004 Datasheet sur Github : https://bit.ly/3qTZdHf
[4] Compte InfluxDB : https://cloud2.influxdata.com/signup
[5] Le dépôt GitHub du projet : http://github.com/signorettae/ESP32-EnergyMeter
[6] Bibliothèque ArduinoOTA : https://github.com/jandrassy/ArduinoOTA
[7] Bibliothèque Arduino pour le PZEM-004T mis à niveau : http://github.com/mandulaj/PZEM-004T-v30

guide de programmation bare-metal (2)

timer précis, UART et débogage

Sergey Lyubka (Irelande)

Dans la première partie du guide, nous avons appris à accéder aux registres du microcontrôleur pour contrôler les broches. De plus, nous avons créé un micrologiciel simple et notre première démonstration de LED clignotante avec un script d'éditeur de liens et d'un fichier *Makefile*. Dans cet épisode de la série, nous abordons la synchronisation précise via les horloges système, l'UART et le débogage.

Note de la rédaction : ce guide est un document vivant sur GitHub [1] qui évolue. Nous avons donc décidé de poursuivre cette série par un épisode supplémentaire, qui sera publié dans le prochain numéro du magazine Elektor (11-12/2023).

Blinky avec interruption de Systick

Pour notre premier exemple de clignotement de LED «Blinky», nous avons utilisé une fonction de délai appelée `spin()` qui exécute simplement des instructions NOP un certain nombre de fois (voir la première partie de la série [2]).

Afin d'assurer une temporisation beaucoup plus précise, nous devons activer l'interruption SysTick d'ARM. SysTick est un compteur matériel de 24 bits, et fait partie du noyau ARM, il est donc documenté par le manuel *Arm® v7-M Architecture Reference Manual* [3]. Le manuel indique que SysTick possède quatre registres :

- CTRL — utilisé pour activer/désactiver SysTick
- LOAD — valeur initiale du compteur
- VAL — la valeur actuelle du compteur, décrémentée à chaque cycle d'horloge
- CALIB — registre de calibration

Chaque fois que VAL passe à zéro, une interruption SysTick est générée. L'index de l'interruption SysTick dans la table vectorielle est 15, nous devons donc le définir. Au démarrage, notre carte Nucleo-F429ZI de STMicroelectronics tourne à 16 MHz. Nous pouvons configurer le compteur SysTick pour qu'il déclenche une interruption toutes les millisecondes.

Tout d'abord, définissons un périphérique SysTick. Il existe quatre registres et, d'après le manuel de référence d'Arm, l'adresse du SysTick est 0xe000e010. Donc :

```
struct systick {
  volatile uint32_t CTRL, LOAD, VAL, CALIB;
};
#define SYSTICK ((struct systick *) 0xe000e010)
```

Ensuite, ajoutez une fonction API qui le configure. Nous devons activer SysTick dans le registre `SYSTICK->CTRL` et le cadencer via `RCC->APB2ENR`, comme décrit dans la section 7.3.14 du manuel [4] :

```
#define BIT(x) (1UL << (x))
static inline void systick_init(uint32_t ticks) {
  // SysTick timer is 24 bits
  if ((ticks - 1) > 0xffffff) return;
  SYSTICK->LOAD = ticks - 1;
  SYSTICK->VAL = 0;
  // Enable systick
  SYSTICK->CTRL = BIT(0) | BIT(1) | BIT(2);
  RCC->APB2ENR |= BIT(14); // Enable SYSCFG
}
```

La carte Nucleo-F429ZI tourne à 16 MHz, c'est-à-dire que si nous appelons `systick_init(16000000 / 1000)`, une interruption SysTick sera générée toutes les millisecondes. Nous devrions définir une fonction de gestion d'interruption – en voici une qui incrémente simplement un compteur de millisecondes de 32 bits :

```
// "volatile" is important!!
static volatile uint32_t s_ticks;
void SysTick_Handler(void) {
  s_ticks++;
}
```

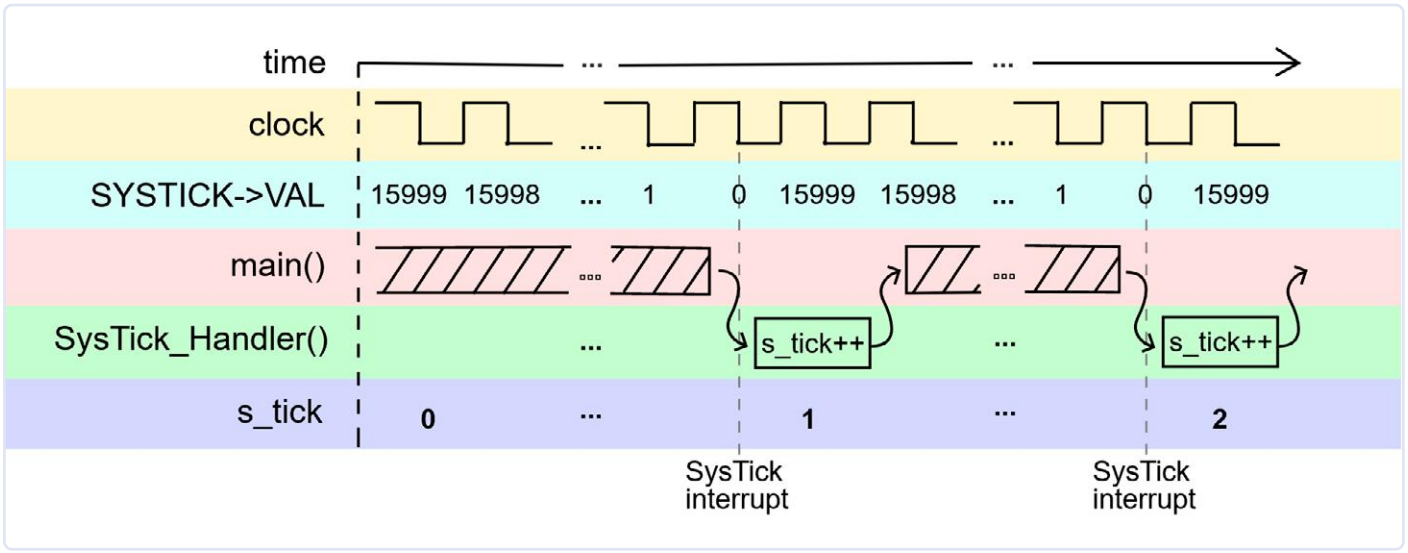


Figure 1. Représentation temporelle de l'exécution interrompue du micrologiciel avec la fonction SysTick_Handler().

Avec une horloge de 16 MHz, nous initialisons le compteur `SysTick` pour qu'il déclenche une interruption tous les 16 000 cycles : La valeur initiale de `SYSTICK->VAL` est de 15 999, puis elle se décrémente à chaque cycle jusqu'à atteindre 0, lorsqu'une interruption est déclenchée. L'exécution du code du micrologiciel est interrompue, et la fonction `SysTick_Handler()` est appelée pour incrémenter la variable `s_tick`. La **figure 1** illustre ceci sur une échelle de temps. Le qualificateur `volatile` est nécessaire ici parce que `s_ticks` est modifié par le gestionnaire d'interruption. `volatile` empêche le compilateur d'optimiser/cacher la valeur de `s_ticks` dans un registre du CPU ; le code généré accède toujours à la mémoire. C'est pourquoi le qualificateur `volatile` est également inclus dans les structures de définition des périphériques. Pour bien comprendre cela, démontrons-le avec une fonction simple : la fonction `delay()` d'Arduino. Utilisons la variable `s_ticks` :

```
// This function waits "ms" milliseconds
void delay(unsigned ms) {
  // Time in the future when we need to stop
  uint32_t until = s_ticks + ms; /
  while (s_ticks < until) (void) 0; // Loop until then
}
```

Compilons maintenant ce code avec et sans le qualificateur `volatile` de `s_ticks` et comparons-le avec le code assembleur compilé :

```
// NO VOLATILE: uint32_t s_ticks;
ldr r3, [pc, #8] // cache s_ticks
ldr r3, [r3, #0] // in r3
adds r0, r3, r0 // r0 = r3 + ms
cmp r3, r0 // ALWAYS FALSE
bcc.n 200000d2
bx lr

// VOLATILE: volatile uint32_t s_ticks;
ldr r2, [pc, #12]
ldr r3, [r2, #0] // r3 = s_ticks
adds r3, r3, r0 // r3 = r3 + ms
ldr r1, [r2, #0] // RELOAD: r1 = s_ticks
cmp r1, r3 // compare
bcc.n 200000d2
bx lr
```

Sans le qualificateur `volatile`, la fonction `delay()` bouclera indéfiniment et ne renverra rien. C'est parce qu'elle met en cache (optimise) la valeur de `s_ticks` dans un registre et ne la met jamais à jour. Le compilateur fait cela parce qu'il ne sait pas que `s_ticks` sera mis à jour ailleurs par le gestionnaire d'interruption ! En revanche, le code compilé avec `volatile`, charge la valeur de `s_ticks` à chaque itération. La règle de base est donc la suivante : **les valeurs en mémoire qui sont mises à jour par les gestionnaires d'interruption ou par le matériel doivent être déclarées comme** `volatile`**.**

Nous devons maintenant ajouter le gestionnaire d'interruption `SysTick_Handler()` à la table vectorielle :

```
__attribute__((section(".vectors")))
    void (*tab[16 + 91])(void) = {
    _estack, _reset, 0, 0, 0, 0, 0, 0, 0, 0,
                     0, 0, 0, 0,  SysTick_Handler
};
```

Nous avons maintenant une horloge précise à la milliseconde ! Créons une fonction d'aide pour les horloges périodiques arbitraires :

```
// t: expiration time, prd: period,
//    now: current time. Return true if expired
bool timer_expired(uint32_t *t, uint32_t prd,
                   uint32_t now) {
  if (now + prd < *t) *t = 0;
    // Time wrapped? Reset timer
  if (*t == 0) *t = now + prd;
    // First poll? Set expiration
  if (*t > now) return false;
    // Not expired yet,return
  *t = (now - *t) > prd ? now + prd : *t + prd;
    // Next expiration time
  return true; // Expired, return true
}
```

Nous pouvons maintenant mettre à jour la boucle principale et utiliser un timer précis pour le clignotement de la LED. Par exemple, utilisons un intervalle de clignotement de 250 millisecondes :

```
// Declare timer and 500ms period
uint32_t timer, period = 500;
for (;;) {
  if (timer_expired(&timer, period, s_ticks)) {
    static bool on;       // This block is executed
    gpio_write(led, on); // Every "period" milliseconds
    on = !on; // Toggle LED state
  }
  // Here we could perform other activities!
}
```

Notez qu'en utilisant SysTick avec une fonction d'aide `timer_expired()`, la boucle principale (également appelée «superloop») devient non bloquante. Cela signifie qu'à l'intérieur de cette boucle, nous pouvons effectuer de nombreuses actions – par exemple, avoir différents timers avec différentes périodes, et ils seront tous déclenchés à temps.
Le code source complet du projet est disponible dans le dossier *step-2-systick* [5].

Ajouter une sortie de débogage UART

Ajoutons maintenant une analyse lisible au micrologiciel. L'un des périphériques du microcontrôleur est une interface UART série. La topographie de la mémoire présentée dans la section 2.3 du manuel montre qu'il existe plusieurs contrôleurs UART/USART - c'est-à-dire que des circuits à l'intérieur du microcontrôleur peuvent échanger des données par l'intermédiaire de certaines broches lorsqu'ils sont correctement configurés. Une configuration UART minimale utilise deux broches : RX (réception) et TX (transmission).
La section 6.9 du manuel de la carte Nucleo [6] montre que l'un des contrôleurs, USART3, utilise les broches PD8 (TX) et PD9 (RX) et est connecté au débogueur ST-LINK intégré. Cela signifie que si nous configurons USART3 et transmettons des données via la broche PD9, nous pouvons les voir sur notre PC via une connexion USB ST-LINK.
Nous allons donc créer une API pratique pour l'UART, comme nous l'avons fait pour les GPIO. La section 30.6 [4] présente un résumé des registres UART. La structure UART correspondante `struct` est :

```
struct uart {
  volatile uint32_t SR, DR, BRR, CR1, CR2, CR3, GTPR;
};
#define UART1 ((struct uart *) 0x40011000)
#define UART2 ((struct uart *) 0x40004400)
#define UART3 ((struct uart *) 0x40004800)
```

Pour configurer un UART, il faut :

- Activer l'horloge UART en définissant le bit approprié dans `RCC->APB2ENR`.
- Définir le mode de fonctionnement alternatif (*alternate function*) des broches RX et TX. Il peut y avoir plusieurs fonctions alternatives (AF) pour une broche donnée, selon le périphérique utilisé. La liste des fonctions alternatives se trouve dansle tableau 12 de la fiche technique du STM32F429ZI. [7].
- Régler la vitesse de transmission (fréquence de réception/transmission de l'horloge) via le registre BRR.
- Activer la réception et la transmission du périphérique via le registre CR1.

Nous savons déjà comment configurer une broche GPIO dans un mode spécifique. Si une broche est en mode AF, nous devons également spécifier le «numéro de fonction», c'est-à-dire le périphérique exact qui prend le contrôle. Cela est possible via le registre AFR (*Alternate Function Register*) du périphérique GPIO. En lisant la description du registre `AFR` dans le manuel de référence, nous voyons que le numéro *AF* occupe quatre bits, donc la configuration complète pour 16 broches occupe deux registres.
Afin de masquer complètement le code spécifique au registre de l'API GPIO, déplaçons l'initialisation de l'horloge GPIO dans la fonction `gpio_set_mode()` :

```
static inline void
  gpio_set_mode(uint16_t pin, uint8_t mode) {
    struct gpio *gpio = GPIO(PINBANK(pin)); // GPIO bank
    int n = PINNO(pin); // Pin number
// Enable GPIO clock
    RCC->AHB1ENR |= BIT(PINBANK(pin));
    ...
```

Maintenant, créeons une fonction API d'initialisation de l'UART - voir **listage 1**.
Enfin, nous avons besoin de fonctions pour lire et écrire sur l'UART. La section 30.6.1 du manuel de référence [4] précise que le registre d'état, SR, indique si les données sont prêtes :

```
static inline int uart_read_ready(struct uart *uart) {
  // If RXNE bit is set, data is ready
  return uart->SR & BIT(5);
}
```

Il est possible d'extraire l'octet de données lui-même du registre de données DR :

```
static inline uint8_t uart_read_byte(struct uart *uart) {
  return (uint8_t) (uart->DR & 255);
}
```

Nous pouvons également transmettre un seul octet via le registre de données. Après avoir défini un octet à écrire, il faut attendre la fin de la transmission, qui sera indiquée par le Bit 7 du registre *Status* :

Listage 1. Fonction API d'initialisation de l'UART.

```
#define FREQ 16000000  // CPU frequency, 16 Mhz
 static inline void uart_init(struct uart *uart, unsigned long baud) {
   // https://www.st.com/resource/en/datasheet/stm32f429zi.pdf
   uint8_t af = 7;           // Alternate function
   uint16_t rx = 0, tx = 0;  // pins

   if (uart == UART1) RCC->APB2ENR |= BIT(4);
   if (uart == UART2) RCC->APB1ENR |= BIT(17);
   if (uart == UART3) RCC->APB1ENR |= BIT(18);
   if (uart == UART1) tx = PIN('A', 9), rx = PIN('A', 10);
   if (uart == UART2) tx = PIN('A', 2), rx = PIN('A', 3);
   if (uart == UART3) tx = PIN('D', 8), rx = PIN('D', 9);

  gpio_set_mode(tx, GPIO_MODE_AF);
   gpio_set_af(tx, af);
   gpio_set_mode(rx, GPIO_MODE_AF);
   gpio_set_af(rx, af);
   uart->CR1 = 0;                          // Disable this UART
   uart->BRR = FREQ / baud;                // FREQ is a UART bus frequency
   uart->CR1 |= BIT(13) | BIT(2) | BIT(3);  // Set UE, RE, TE
 }
```

```
static inline void uart_write_byte(struct uart *uart,
                                   uint8_t byte) {
  uart->DR = byte;
  while ((uart->SR & BIT(7)) == 0) spin(1);
}
```

Et pour l'écriture dans un tampon :

```
static inline void
  uart_write_buf(struct uart *uart,
                 char *buf, size_t len) {
    while (len-- > 0)
      uart_write_byte(uart, *(uint8_t *) buf++);
}
```

Initialisons l'UART dans la fonction `main()` :

```
…
uart_init(UART3, 115200); // Initialize UART
```

Nous pouvons maintenant afficher le message *hi\r\n* à chaque fois que la LED clignote !

```
if (timer_expired(&timer, period, s_ticks)) {
  …
  uart_write_buf(UART3, "hi\r\n", 4); // Write message
}
```

Reconstruisez, re-flashez et connectez un programme de terminal au port ST-LINK. Sur mon Mac, j'utilise *cu*, que vous pouvez également utiliser sous Linux. Sous Windows, l'utilitaire PuTTY [8] fonctionne bien. Lancez un terminal et observez les messages :

```
$ cu -l /dev/cu.YOUR_SERIAL_PORT -s 115200
hi
hi
```

Le code source complet du projet est disponible dans le dossier *step-3-uart* [9].

Rediriger printf() vers UART

Dans cette section, nous appelons la fonction `printf()` au lieu de `uart_write_buf()`, ce qui nous permet d'obtenir une sortie formatée - d'avoir plus de flexibilité d'affichage d'informations de diagnostic, en implémentant ce que l'on appelle "printf()-style debugging".

La suite d'outils GNU d'ARM que nous utilisons est livrée non seulement avec un compilateur GCC et d'autres outils, mais aussi avec une bibliothèque C appelée *newlib* [10]. La bibliothèque *newlib* a été développée par RedHat pour les systèmes embarqués.

Si le micrologiciel appelle une fonction de la bibliothèque C standard, par exemple `strcmp()`, un code de *newlib* sera ajouté par l'éditeur de liens GCC.

Certaines des fonctions C standard que la bibliothèque *newlib* implémente d'une manière spéciale, en particulier les opérations d'entrée/sortie (IO) de fichiers : ces fonctions appellent finalement un ensemble de fonctions d'E/S de bas niveau appelées *syscalls*.

Par exemple

- `fopen()` appelle éventuellement `_open()`
- `fread()` appelle éventuellement une fonction `_read()` de bas niveau
- `fwrite()`, `fprintf()`, `printf()` appellent éventuellement une fonction `_write()` de bas niveau
- `malloc()` appelle éventuellement `_sbrk()`, etc.

Listage 2. La fonction main() devient assez compacte

```
#include "hal.h"

static volatile uint32_t s_ticks;
 void SysTick_Handler(void) {
   s_ticks++;
 }

int main(void) {
   uint16_t led = PIN('B', 7);            // Blue LED
   systick_init(16000000 / 1000);         // Tick every 1 ms
   gpio_set_mode(led, GPIO_MODE_OUTPUT);  // Set blue LED to output mode
   uart_init(UART3, 115200);              // Initialise UART
   uint32_t timer = 0, period = 500;      // Declare timer and 500ms period
   for (;;) {
     if (timer_expired(&timer, period, s_ticks)) {
       static bool on;                      // This block is executed
       gpio_write(led, on);                 // Every 'period' milliseconds
       on = !on;                            // Toggle LED state
       uart_write_buf(UART3, "hi\r\n", 4);  // Write message
     }
     // Here we could perform other activities!
   }
   return 0;
 }
```

Ainsi, en modifiant l'appel système (syscall) de `_write()`, nous pouvons rediriger `printf()` vers ce que nous voulons. Ce mécanisme est appelé « *IO retargeting* ».
Note : L'EDI Cube de STM32 utilise aussi le compilateur GCC d'ARM avec newlib, c'est pourquoi les projets Cube incluent typiquement un fichier *syscalls.c*. D'autres suites d'outils, comme CCS de TI et CC de Keil, peuvent utiliser une bibliothèque C différente avec un mécanisme de *retargeting* légèrement différent. Nous utilisons newlib, donc modifions l'appel syscall de `_write()` pour imprimer sur UART3.
Avant cette étape, organisons le code source comme suit :

- Déplacer toutes les définitions d'API dans le fichier *mcu.h*.
- Déplacer le code de démarrage dans *startup.c*.
- Créer un fichier vide *syscalls.c* pour les syscalls de *newlib*.
- Modifier Makefile pour ajouter *syscalls.c* et *startup.c* à la construction.

Après avoir déplacé toutes les définitions d'API dans *mcu.h*, le fichier *main.c* devient assez compact. Notez qu'il ne contient aucune mention des registres de bas niveau, seulement des fonctions API de haut niveau faciles à comprendre – voir **listage 2**.
Maintenant, nous allons recibler `printf()` vers UART3. Dans le fichier vide *syscalls.c*, copiez/collez le code suivant :

```
#include "mcu.h"
int _write(int fd, char *ptr, int len) {
  (void) fd, (void) ptr, (void) len;
  if (fd == 1) uart_write_buf(UART3, ptr, (size_t) len);
  return -1;
}
```

Ici, si le descripteur de fichier sur lequel nous écrivons `fd`, est 1 (ce qui est un descripteur de sortie standard), alors écrivez le tampon sur UART3. Dans le cas contraire, il faut l'ignorer. C'est la base du *retargeting* !
Reconstruire ce micrologiciel résulte en un tas d'erreurs de linker, comme le montre le **listage 3**.
Comme nous avons utilisé une fonction stdio de *newlib*, nous devons fournir à la bibliothèque le reste des appels systèmes. Ajoutons un simple *stub* qui ne fait rien (**listage 4**).
La reconstruction ne donne aucune erreur. Dernière étape : remplacer l'appel de `uart_write_buf()` dans la fonction `main()` par un appel de `printf()` qui affiche quelque chose d'utile, par exemple l'état d'une LED et la valeur actuelle de systick :

```
// Write message
printf("LED: %d, tick: %lu\r\n", on, s_ticks);
```

La sortie série ressemble à ceci :

```
LED: 1, tick: 250
LED: 0, tick: 500
LED: 1, tick: 750
LED: 0, tick: 1000
```

Félicitations ! Nous avons appris comment fonctionne le *retargeting* d'entrées-sorties et nous pouvons maintenant déboguer le micrologiciel avec la fonction `printf()-debug`. Vous pouvez trouver le code source complet du projet dans le dossier *step-4-printf* [11].

Listage 3. Un tas d'erreurs d'édition de liens

```
../../arm-none-eabi/lib/thumb/v7e-m+fp/hard/libc_nano.a(lib_a-sbrkr.o): in function `_sbrk_r':
 sbrkr.c:(.text._sbrk_r+0xc): undefined reference to `_sbrk'
 closer.c:(.text._close_r+0xc): undefined reference to `_close'
 lseekr.c:(.text._lseek_r+0x10): undefined reference to `_lseek'
 readr.c:(.text._read_r+0x10): undefined reference to `_read'
 fstatr.c:(.text._fstat_r+0xe): undefined reference to `_fstat'
 isattyr.c:(.text._isatty_r+0xc): undefined reference to `_isatty'
```

Listage 4. Ajout de simples stubs

```
int _fstat(int fd, struct stat *st) {
   (void) fd, (void) st;
   return -1;
 }

void *_sbrk(int incr) {
   (void) incr;
   return NULL;
 }

int _close(int fd) {
   (void) fd;
   return -1;
 }

int _isatty(int fd) {
   (void) fd;
   return 1;
 }

int _read(int fd, char *ptr, int len) {
   (void) fd, (void) ptr, (void) len;
   return -1;
 }

int _lseek(int fd, int ptr, int dir) {
   (void) fd, (void) ptr, (void) dir;
   return 0;
 }
```

Déboguer avec Ozone de Segger

Que se passe-t-il si notre micrologiciel est bloqué quelque part et que la fonction de débogage `printf()` ne fonctionne pas ? Et si même le code de démarrage ne fonctionne pas ? Nous aurions besoin d'un débogueur. Il y a beaucoup d'options, mais je recommanderais d'utiliser le débogueur Ozone de Segger. Pourquoi ? Parce qu'il est autonome ; il n'a pas besoin de configuration d'EDI. Nous pouvons envoyer notre fichier *firmware.elf* directement à Ozone, et il récupérera nos fichiers sources.

Téléchargez donc Ozone depuis le site de Segger [12]. Avant de pouvoir l'utiliser avec la carte Nucleo, nous devons convertir le micrologiciel *ST-LINK* du débogueur intégré en un micrologiciel *jlink* qu'Ozone supporte. Suivez les instructions sur le site de Segger [13] :

- Lancez Ozone. Choisissez notre appareil dans l'assistant (**figure 2**).
- Sélectionnez un débogueur que nous allons utiliser - il s'agit d'un ST-LINK (**figure 3**).
- choisissez le fichier *firmware.elf* (**figure 4**).
- Laissez les valeurs par défaut sur la page suivante, cliquez sur *Finish*, et le débogueur sera chargé (notez que le code source *mcu.h* est récupéré), voir **figure 5**.
- Cliquez sur le bouton vert pour télécharger, exécutez le micrologiciel, et c'est fini. (**figure 6**).

Nous pouvons maintenant parcourir le code en une seule étape, définir des points d'arrêt et effectuer les opérations de débogage habituelles. Une chose à noter est la vue *Peripherals* d'Ozone (**figure 7**). En l'utilisant, nous pouvons directement examiner ou définir l'état des périphériques. Par exemple, allumons une LED verte intégrée (PB0) :

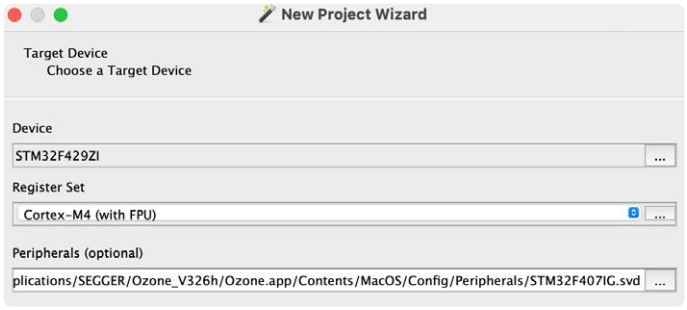


Figure 2. Sélectionnez l'appareil dans l'assistant.

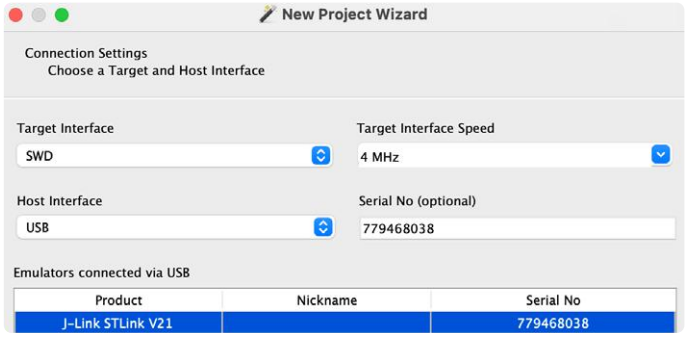


Figure 3. Sélectionnez STLink comme débogueur.

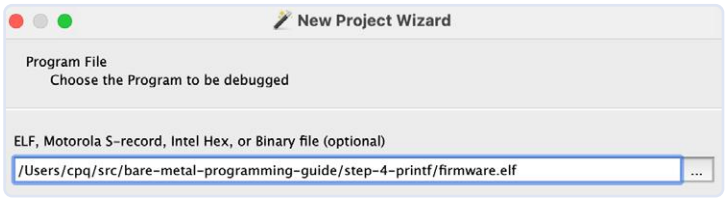


Figure 4. Le programme à déboguer est dans le fichier firmware.elf.

1. Faisons d'abord le pointage de la GPIOB. Allez sur *Peripherals RCC AHB1ENR*, et activez le bit *GPIOBEN* — mettez le à 1 (**figure 8**).
2. Allez sur *Peripherals GPIO GPIOB MODER*, mettez *MODER0* à 1 (sortie) (**figure 9**).
3. Allez sur *Peripherals GPIO GPIOB ODR*, mettez *ODR0* à 1 (on) (**figure 10**).

Maintenant, une LED verte devrait s'allumer ! Débogage réussi ! Dans la troisième partie de cette série, nous implémenterons un serveur web. En outre, nous montrerons comment un programme peut être testé automatiquement, et bien d'autres choses encore. Restez à l'écoute !

220665-B-04

À propos de l'auteur

Sergey Lyubka est ingénieur et entrepreneur. Il est titulaire d'un MSc en physique de l'université d'État de Kiev, en Ukraine. Sergey est directeur et cofondateur de Cesanta, une entreprise technologique basée à Dublin, en Irlande (Embedded Web Server for electronic devices : https://mongoose.ws). Il est passionné par la programmation de réseaux embarqués bare-metal.

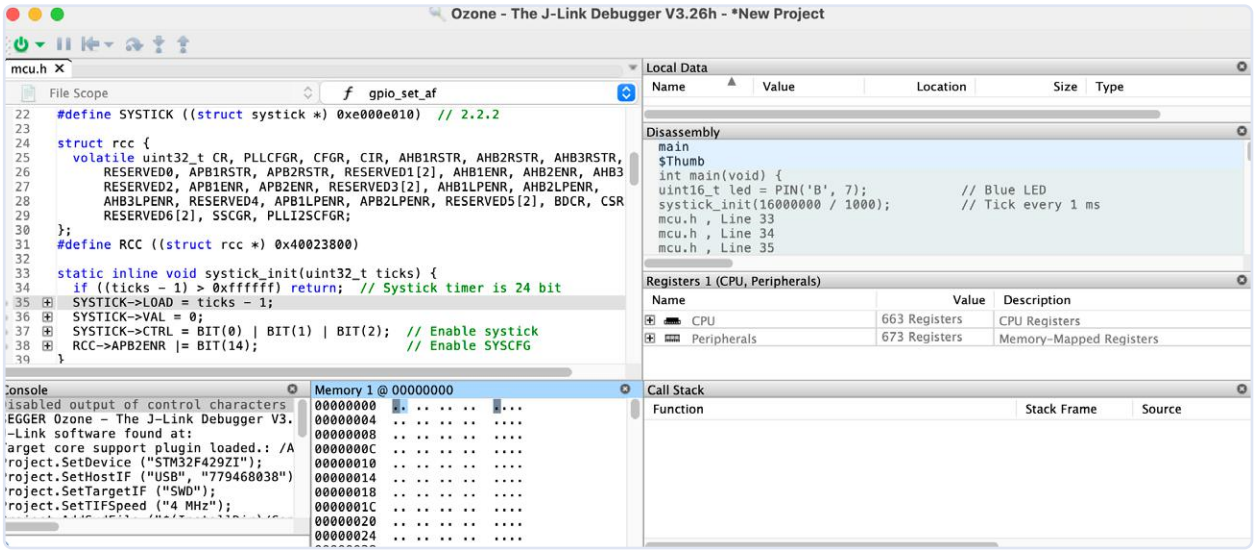


Figure 5. Le débogueur est chargé, et le fichier mcu.h apparaîtra bientôt.

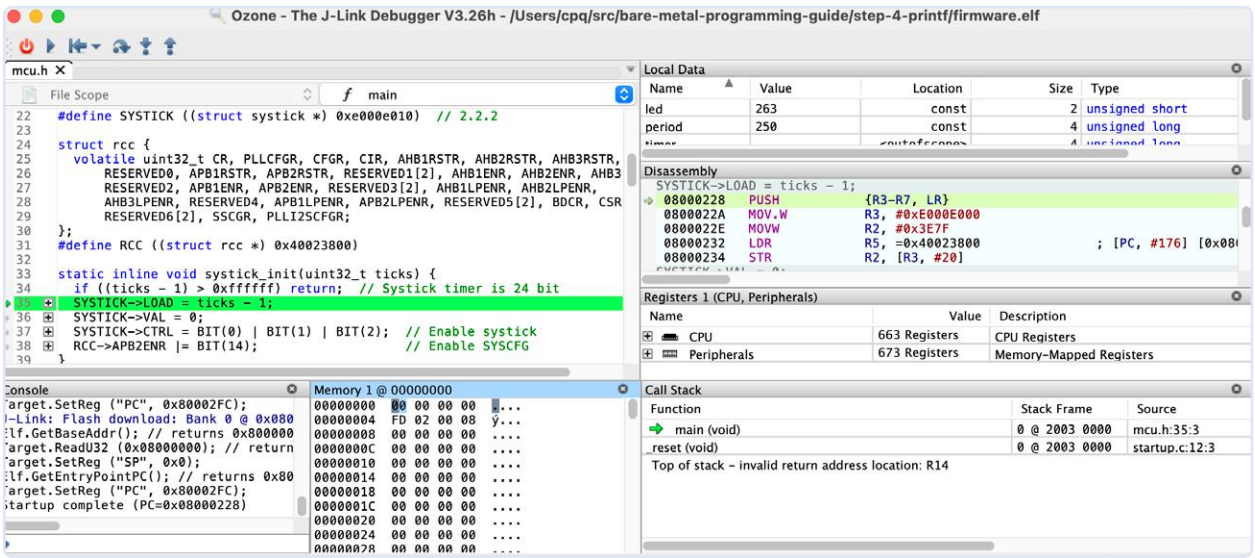


Figure 6. Après avoir exécuté le micrologiciel, il s'arrête à la ligne SYSTICK->LOAD = ticks - 1;.

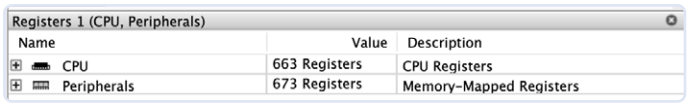


Figure 7. La vue Peripherals d'Ozone facilite l'examen et la configuration des périphériques.

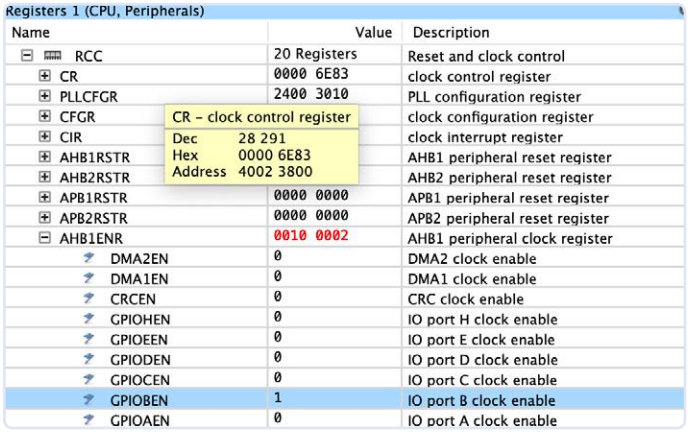


Figure 8. Activation de l'horloge sur le port B en mettant la valeur de GPIOBEN à 1.

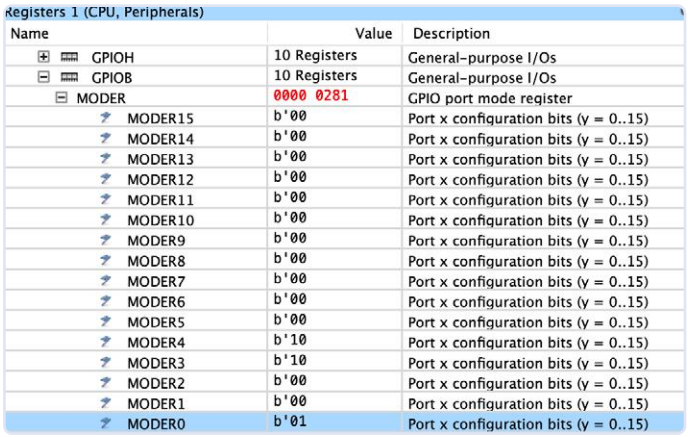


Figure 9. Mettez MODER0 à 1 (et sélectionnez donc la sortie) dans les périphériques GPIO.

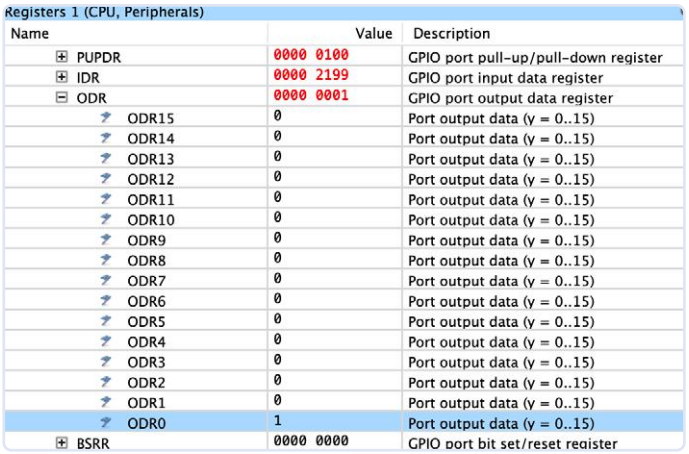


Figure 10. Activez ODR0 en sélectionnant la valeur 1 dans ODR (GPIO).

Questions ou commentaires ?

Envoyez un courriel à l'auteur (sergey.lyubka@cesanta.com) ou contactez Elektor (redaction@elektor.fr).

Produits

- **Dogan Ibrahim, *Nucleo Boards Programming with the STM32CubeIDE* (Elektor 2020)**
  https://elektor.fr/19530
- **Dogan Ibrahim, *Programming with STM32 Nucleo Boards* (Elektor 2015)**
  https://elektor.fr/18585

LIENS

[1] Dépôt GitHub pour cet article : https://github.com/cpq/bare-metal-programming-guide
[2] Sergey Lyubka, " Guide de programmation Bare-Metal (2) " Elektor 9-10/2023 : https://elektormagazine.fr/220665-04
[3] Manuel de référence de l'architecture Arm v7-M : https://developer.arm.com/documentation/ddi0403/ee
[4] Manuel de référence RM0090 pour STM32F429 : https://bit.ly/3neE7S7
[5] Step 2 SysTick folder: https://github.com/cpq/bare-metal-programming-guide/tree/main/steps/step-2-systick
[6] Manuel d'utilisation de la carte Nucleo-144 (UM1974) : https://bit.ly/3oIBXKZ
[7] Fiche technique de la STM32F429ZI : https://st.com/resource/en/datasheet/stm32f429zi.pdf
[8] PuTTY : https://putty.org
[9] Dossier de l'étape 3 UART : https://github.com/cpq/bare-metal-programming-guide/tree/main/steps/step-3-uart
[10] Bibliothèque C newlib : https://sourceware.org/newlib
[11] Dossier de l'étape 4 printf : https://github.com/cpq/bare-metal-programming-guide/tree/main/steps/step-4-printf
[12] Ozone — Débogueur J-Link et analyseur de performances :
https://segger.com/products/development-tools/ozone-j-link-debugger
[13] Conversion du ST-LINK intégré en J-Link :
https://segger.com/products/debug-probes/j-link/models/other-j-links/st-link-on-board

elektorstore

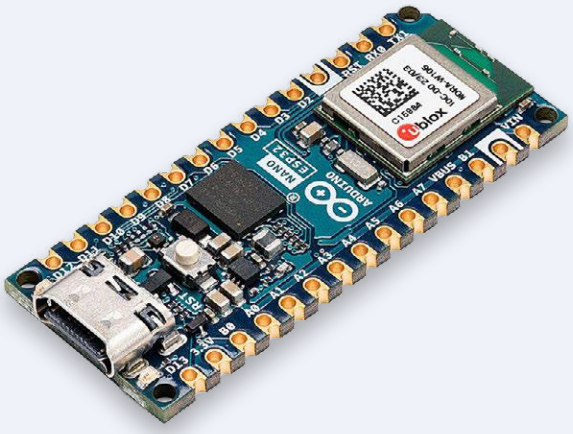
www.elektor.fr

e-choppe Elektor

des produits et des prix surprenants

L'e-choppe Elektor a évolué. D'abord simple commerce de proximité pour nos productions (livres, magazines, kits et modules), c'est désormais une boutique en ligne bien rodée qui propose des produits surprenants à des prix très étudiés. Ce sont les produits que nous aimons et testons nous-mêmes. Si vous avez une suggestion, n'hésitez pas : sale@elektor.fr.

Arduino Nano ESP32



L'Arduino Nano ESP32 (avec ou sans connecteurs) est une carte au format Nano basée sur l'ESP32-S3 (intégré dans le NORA-W106-10B de u-blox). Il s'agit de la première carte Arduino entièrement basée sur un ESP32, et elle dispose du Wi-Fi, du Bluetooth LE, du débogage via USB natif dans l'IDE Arduino ainsi que de la faible consommation d'énergie.

Prix : 23,95 €

Prix (membres) : 21,56 €

www.elektor.fr/20562

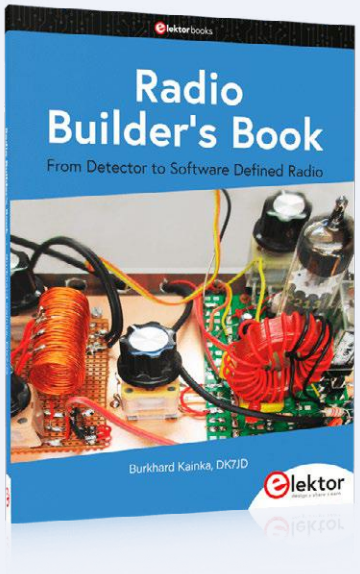
Radio Builder's Book

Radio frequency (RF) technology is one of the areas which still allows putting your own ideas into practice. Countless circuit variants with special objectives allow space for meaningful experiments and projects. Many things simply aren't available off the shelf.

Prix : 34,95 €

Prix (membres) : 31,46 €

www.elektor.fr/20553



Kit de démarrage officiel Raspberry Pi 4 (4 Go)



Prix : 104,95 €

www.elektor.fr/20556

QuantAsylum QA403 analyseur audio 24 bit



Prix : 799,00 €

Prix (membres) : 719,10 €

www.elektor.fr/20530

FNIRSI DSO-TC3 (3-en-1) oscilloscope, testeur de composants et générateur de signaux



Prix : ~~74,95 €~~

Prix spécial : 59,95 €

www.elektor.fr/20520

ESP Terminal (carte de développement basée sur un ESP32-S3 avec écran 3,5")



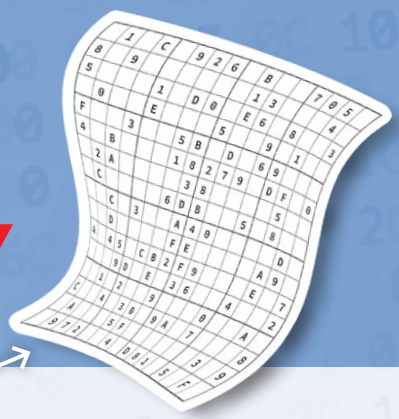
Prix : 44,95 €

Prix (membres) : 40,46 €

www.elektor.fr/20526

hexadoku

casse-tête pour elektorniciens



La dernière page de votre magazine propose toujours une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans, 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 €.

Où envoyer votre réponse ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **15 octobre 2023** à l'adresse **hexadoku@elektor.fr**

LES GAGNANTS

La solution de la grille du numéro de juillet/août 2023 est **23BDF.**

La liste des gagnants est publiée ici : www.elektormagazine.fr/hexadoku

Bravo à tous les participants et félicitations aux gagnants !

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

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

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.

TME®
Electronic Components

TRANSFER MULTISORT ELEKTRONIK

GLOBAL DISTRIBUTOR OF ELECTRONIC COMPONENTS



Utilisez ECAD Model chez TME

- Références et conceptions PCB
- Modèles 3D
- Accès rapides

SOLUTION PARFAITE POUR LES CONCEPTEURS

Transfer Multisort Elektronik Sp. z o.o.
Łódź, Poland, export@tme.eu

Rejoignez-nous :

tme.eu

YOU NEED IT, WE HAVE IT!

tme.com

Imaginez si les frères Wright s'étaient associés à Mouser



Avec la plus vaste sélection des produits les plus récents™, il n'y a pas de limite à la conception.

mouser.fr/new

+44 (0) 1494-427500

MOUSER ELECTRONICS