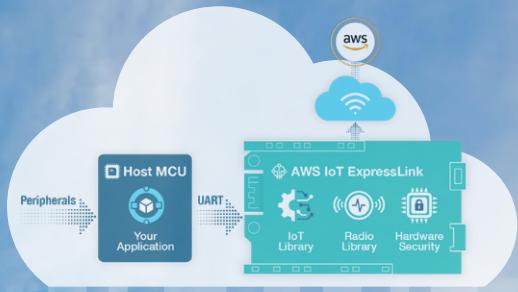
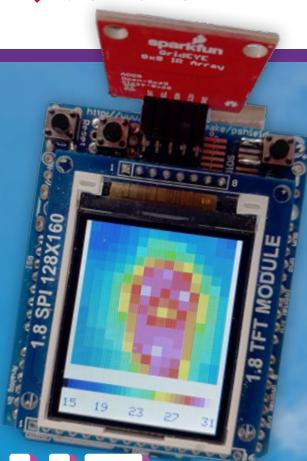


réalisée avec un
Arduino UNO

PETITE CAMÉRA THERMIQUE



AWS pour Arduino et Cie. (1)

utiliser AWS IoT ExpressLink en pratique



capteurs de stations météorologiques

lequel choisir ?

enregistreur universel de données de jardinage
un pas vers l'IA au Jardin



FOCUS SUR

IdO et capteurs



Raspberry Pi se met à l'IA
accélérateur avec 13 TOPS



compteur d'énergie ESP32
intégration dans Home Assistant



générateur analogique 1 kHz
ondes sinusoïdales à faible distorsion

L 19624 - 508 - F : 15,50 € - RD



Rejoignez la communauté Elektor



Devenez membre maintenant !



- 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
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

47^{ème} année n° 508
juillet/août 2024
ISSN 0181-7450

N° de TVA Intracommunautaire : FR90319937454

Dépôt légal : juillet 2024
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/publicite

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

Droits d'auteur
© 2024 Elektor International Media B.V.

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

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

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

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



Jens Nickel

rédacteur en chef d'Elektor Magazine

Décelter l'innovation



Lorsque je pense aux salons d'électronique, les premiers événements qui me viennent à l'esprit sont *embedded world* et *electronica* - deux rendez-vous incontournables pour Elektor, où nous avons bien entendu notre propre stand. Je peux déjà vous annoncer que nous avons quelque chose de très spécial en préparation pour *electronica* en novembre - vous en saurez plus dans les prochains numéros.

Mais, les salons plus modestes méritent aussi une visite. Cette semaine, je suis revenu du salon *Sensor+Test*, qui se tient à Nuremberg, parallèlement à *PCIM Europe* (Power Conversion and Intelligent Motion). Avec ses nombreux petits stands, ses exposants parfois très spécialisés et surtout son hébergement dans les halls d'exposition 1 et 2 quelque peu «poussiéreux», de nombreux visiteurs auront probablement l'impression de faire un voyage dans le passé. Vous ne trouverez pas ici des stands coûtant des millions, ou les entreprises présentent leur vision du pouvoir de marché basé sur l'IA pour les 10 prochaines années. Cependant, les petites entreprises jouent également un rôle crucial dans l'innovation. J'ai vu beaucoup de choses intéressantes au salon *Sensor+Test*, comme un appareil portable qui visualise les sources audio et de bruit, combiné à une image provenant d'une caméra ainsi qu'à une image thermographique, créant ainsi un nouveau type d'outil pour la maintenance des machines.

Outre *Sensor+Test* (en accord avec le thème de ce numéro), j'ai également visité le salon *PCIM*. Vous trouverez un bref compte rendu dans le numéro bonus numérique, que vous pouvez télécharger gratuitement sur le site d'Elektor (www.elektormagazine.fr/2407-bonus). Nous sommes ravis de vous présenter les numéros bonus (en version numérique) pour toutes nos éditions. Ces numéros, disponibles en téléchargement gratuit en format PDF sur le site d'Elektor, enrichissent nos publications habituelles avec des projets additionnels et des articles de fond approfondis. Vous avez peut-être déjà découvert ce format à travers notre récent «Circuit Special» ou l'édition rédigée en collaboration avec nos amis d'*Espressif*, qui a été téléchargée plus de 114 000 fois !

Vous pouvez trouver ces éditions bonus sur les pages thématiques spéciales que nous mettons à jour tout au long de l'année. Si vous avez un intérêt particulier pour l'*IdO* et les capteurs, visitez notre page thématique «*IdO et capteurs*» pour des nouvelles continues et des articles exclusifs tout au long de l'année (www.elektormagazine.com/iot-sensors). Dans ce numéro du magazine Elektor, mon collègue Saad Imtiaz présente son compteur d'énergie basé sur l'*ESP32* - qui fonctionne intégralement pour la première fois - utilisé en combinaison avec l'impressionnant cadre Home Assistant (page 12). Les fans de l'*IdO* devraient également explorer l'article de Tam Hanna (page 62) - un guide pratique et détaillé sur la manière de connecter de petites cartes à microcontrôleur au puissant Amazon Web Services.

Elektor Labs : idées et projets

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

www.elektormagazine.fr/labs

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

notre équipe

Rédacteur en chef : Jens Nickel | Rédaction : Asma Adhimi, Roberto Armani, Eric Bogers, Jan Buiting, Stuart Cording, Rolf Gerstendorf (RG), Ton Giesberts, Ouafae Hassani, Hedwig Hennekens, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Jean-François Simon, 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

PETITE CAMÉRA THERMIQUE

réalisée avec un
Arduino UNO



6

Rubriques

3 Édito

18 2024 : l'odyssée de l'IA

améliorer la détection d'objets : intégration de techniques avancées

44 sur le vif

montrez-moi là où ça fait mâle

80 drôle de composant, la série

le quartz

112 démarrer en électronique...

...connexions symétriques

Features

21 Raspberry Pi se met à l'IA

nouveau kit comprenant un accélérateur IA matériel et un adaptateur M.2 HAT+

24 capteurs de stations météorologiques

lequel choisir ?

38 livres d'Elektor

optimisation et contrôle des appareils Thread à faible consommation d'énergie

50 SparkFun Thing Plus Matter

carte de développement Ido polyvalente basée sur Matter

62 AWS pour Arduino et Cie. (1)

utiliser AWS IoT ExpressLink en pratique

94 Miletus : utiliser les applications Web hors ligne

accès aux fonctions de l'appareil et du système

102 Interview : de la 4G à la 5G

est-ce une étape si facile à franchir ?

CONTENU BONUS

Découvrez le numéro bonus du magazine Elektor consacré à l'ido et aux capteurs !

- Projet : Volet roulant intelligent
- Nouveaux produits présentés au salon Sensor+Test et PCIM à Nuremberg
- Initiation : Comment fonctionnent les capteurs tactiles capacitifs ?
- Revue : Caméra d'imagerie thermique HT-03
- Et bien plus encore !



www.elektormagazine.fr/2407-bonus

Projets

6 petite caméra thermique

réalisée avec un Arduino UNO

12 compteur d'énergie basé sur l'ESP32

intégration et test avec Home Assistant

30 relevé des compteurs d'eau basé sur l'IA (1)

intégrez votre ancien compteur dans l'IdO !

34 une alarme GSM

un module GSM protège votre garage à distance

Raspberry Pi se met à l'IA

kit officiel comprenant un accélérateur IA



21



capteurs de stations météorologiques

lequel choisir ?

24

46 chambre à brouillard à faire soi-même
visualiser les rayonnements invisibles

70 détecteur de flux d'air Arduino
aucun capteur externe n'est nécessaire !

73 détecteur de fuite d'eau
connecté à l'Arduino Cloud

82 enregistreur universel de données de jardinage
un pas vers l'IA au Jardin.

89 un générateur analogique 1 kHz
ondes sinusoïdales à faible distorsion



Plongez dans
l'IdO et les capteurs

Visitez la page « IdO et capteurs » d'Elektor pour découvrir des projets, des vidéos et des tutoriels !

www.elektormagazine.com/iot-sensors

Industrie

54 rétroéquipement IoT
adaptation des machines à interface RS232 à l'industrie 4.0

57 ajouter l'IoT grâce aux microcontrôleurs 8 bits

60 la technologie au service du développement durable
les avancées technologiques favorisent une utilisation plus efficace de l'énergie dans de nombreuses applications

Bientôt dans ces pages

Circuits de vacances 2024 (août et septembre 2024)

Dans la tradition des numéros « circuits de vacances », la prochaine édition sera extra épaisse, remplie de dizaines de projets à réaliser soi-même, de circuits rétro, de trucs et astuces et bien plus encore ! Le numéro « circuits de vacances 2024 » sera publiée aux alentours du 14 août 2024.

Le numéro de Septembre-octobre 2024

Vous retrouverez dans le prochain magazine Elektor l'habituel mélange stimulant de réalisations originales, de circuits, d'articles de fond, de sujets nouveaux, de trucs et d'astuces pour les électroniciens. Le thème de ce numéro sera « Applications sans fil ».

- ESP32 Range Extender
- Station LoRa avec le module compact WIO E5
- Carte d'expansion pour ESP32S3 XIAO
- AWS Cloud pour Arduino et Cie. : Transmission de données
- Sonde RF avec graphique à barres LED
- Filtre Notch configurable

Le numéro de septembre-octobre 2024 du magazine Elektor sera publié aux alentours du 11 septembre 2024. La date d'arrivée du magazine papier chez les abonnés dépend des aléas d'acheminement.

FOCUS SUR

IdO et capteurs

petite caméra thermique

réalisée avec un Arduino UNO

Source : Shutterstock

Roland Stiglmayr (Allemagne)

Quiconque a déjà eu besoin de détecter des points chauds sur un matériel électronique, de mesurer la température d'un composant de puissance ou de détecter des personnes dans une pièce, a probablement souhaité disposer d'une caméra thermique. Dans cet article, nous présentons une solution de bricolage peu coûteuse basée sur un capteur 8×8 pixels de Panasonic.

Un électronicien ambitieux se demandera certainement s'il est possible de construire soi-même une caméra thermique. La réponse est : « Oui, c'est possible, à condition d'utiliser les bons composants ». Le capteur thermique le plus approprié est le AMG88xx « Grid-EYE »

de Panasonic. Grâce à son traitement interne des signaux, qui se charge de tout le calcul des températures des 64 points de mesure, il réduit considérablement la charge de l'ordinateur externe, ce qui fait qu'un UNO avec un écran TFT de 1,8 pouce comme celui de la **figure 1** est tout à fait suffisant pour construire une caméra thermique simple.

Le capteur à rayons infrarouges

Le capteur AMG88xx [1] est une puce montée sur un support en céramique et fabriquée selon la technologie MEMS (*Micro Electro-Mechanical Systems*) avec 64 points de mesure (*pixels*) sensibles aux infrarouges. Les pixels forment une matrice carrée. Le capteur est accompagné d'un ASIC avec l'électronique de calcul déjà mentionnée et une thermistance pour la saisie de la température de référence. Le support en céramique est recouvert d'un capuchon métallique dans lequel est serrée une lentille, laquelle est en silicium afin de ne pas atténuer le rayonnement dans la gamme de longueurs d'onde de 5 à 12 µm.

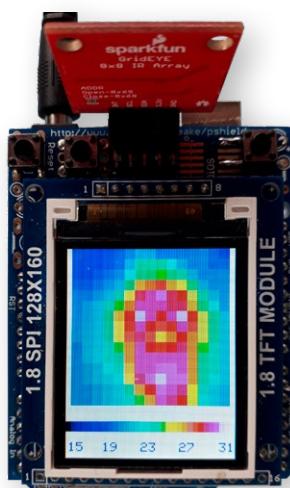


Figure 1. Un cerveau intelligent génère évidemment de la chaleur.

La lentille forme l'image de l'objet à mesurer sur le champ du capteur. On peut imaginer que la pointe d'une pyramide carrée est placée sur chaque pixel, qui ne reçoit que le rayonnement contenu à l'intérieur de cette pyramide. L'objet est ainsi représenté par 64 surfaces contiguës formant un carré.

La **figure 2** montre une représentation simplifiée et idéalisée d'un réseau de 4×4 points de mesure. Le champ de vision (*Field of View, FOV*) est la zone visible par le capteur, qui est déterminée exclusivement par les angles d'ouverture vertical et horizontal. Comme le capteur est une matrice carrée, ces angles, notés α , sont égaux. Pour chaque distance d d'objet a , il existe une surface visible dans le champ de vision avec des côtés de longueur S . La surface est elle-même constituée de 16 surfaces individuelles de côté s_{pix} , chacune associée à un pixel spécifique. La température de l'objet n'est mesurée avec précision que si l'objet couvre entièrement au moins l'une de ces surfaces. Cette condition permet de calculer la plus petite taille d'un objet détectable situé à une distance donnée a . Dans le cas d'une matrice carrée de $n \times n$ pixels, on a :

$$s_{pix} = 1/n \quad (2a \tan(\alpha/2))$$

$$A_{pix} = s_{pix}^2$$

Dans la figure 2, seule la partie de l'objet ayant les coordonnées (2,2) répond à cette exigence. Les zones du champ de vision ayant les coordonnées (2,1), (1,2), (2,3) ne sont couvertes qu'à moitié. L'écart de température mesuré des pixels associés n'est donc que la moitié de celui de la zone (2,2).

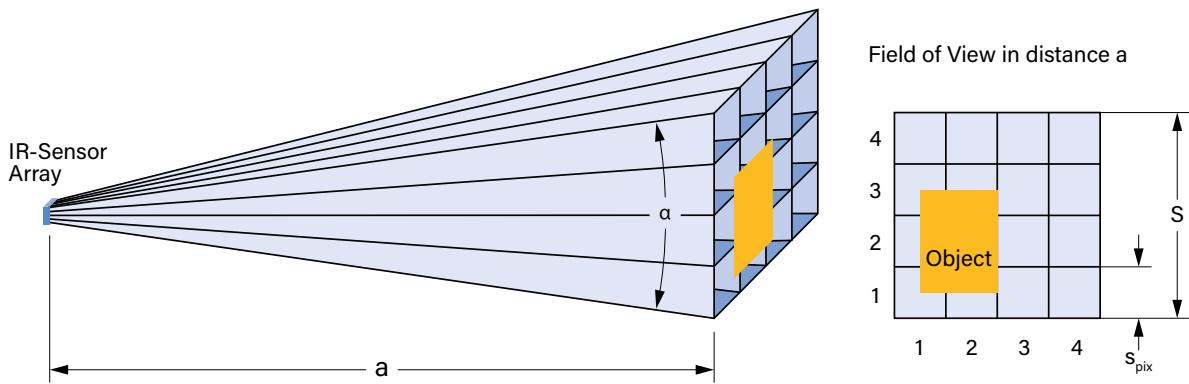


Figure 2. Zone visible d'un capteur carré 4×4 avec un champ de vision α à une distance a .

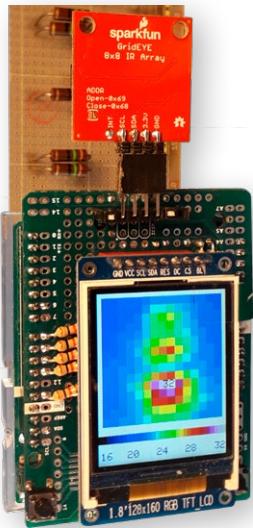


Figure 3. Mesure d'un objet à une distance de 50 mm, $\alpha = 60^\circ$.

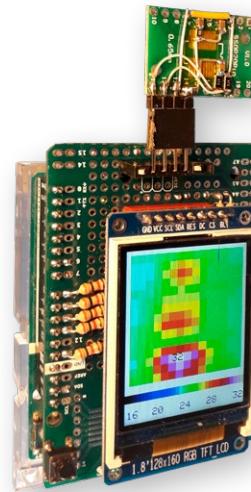


Figure 4. Mesure d'un objet à une distance de 100 mm, $\alpha = 30^\circ$.

Autrement dit, cela signifie que, pour que la zone projetée sur un pixel soit entièrement couverte, un objet doit être d'autant plus grand qu'il est plus éloigné. Cette condition doit toujours être respectée pour éviter des erreurs de mesure et d'interprétation. Les **figures 3** et **4** illustrent la relation entre le champ de vision et la distance de l'objet à mesurer.

La précision de la mesure dépend également de l'émissivité ϵ de l'objet. Elle dépend de la composition de la surface et caractérise sa capacité à émettre un rayonnement thermique (c'est le rapport entre sa luminance et celle du corps noir). Le capteur Grid-EYE suppose une émissivité constante de 0,93 dans son calcul interne de la température.

Thermopile

Chaque point de mesure individuel du champ du capteur est ce que l'on appelle une thermopile [2], qui se compose de nombreux thermocouples connectés électriquement

Les thermocouples génèrent une tension proportionnelle à l'écart de température. Pour déterminer la température absolue, il faut mesurer très précisément la température de référence et l'ajouter à cet écart. La courbe caractéristique d'un thermocouple n'est pas linéaire, mais dépend de la température de référence et de la tension mesurée. Les valeurs mesurées sont linéarisées à l'aide de courbes stockées dans la mémoire morte du capteur. La dispersion des différentes thermopiles est prise en compte par des données d'étalonnage individuelles stockées en interne. Il est facile d'imaginer à quel point la linéarisation de la courbe caractéristique consommerait de ressources en temps et en calcul si l'ASIC interne ne se chargeait pas de cette tâche. Sur son interface I²C, l'ASIC fournit les valeurs mesurées linéarisées et parfaitement traitées en degrés Celsius. Différents types de capteurs sont disponibles en fonction de l'application. Le **tableau 1** en donne un aperçu.

Tableau 1. Valeurs typiques des capteurs infrarouges de la série AMG88xx.

Module	V_{DD}	Plage de mesure	Angle total de détection	Angle de détection d'un pixel	Taille de l'objet relative à un pixel à une distance de 30 cm
AMG8832	3,3 V	-20°C ... 100°C	60°	7,5°	4 cm
AMG8833	3,3 V	0°C ... 80°C	60°	7,5°	4 cm
AMG8834	3,3 V	-20°C ... 100°C	60°	7,5°	4 cm
AMG8853	5 V	0°C ... 80°C	60°	7,5°	4 cm
AMG8854	5 V	-20°C ... 100°C	60°	7,5°	4 cm
AMG883642	3,3 V	-20°C ... 100°C	32° ... 34°	4°	2 cm
AMG883543	3,3 V	0°C ... 80°C	90°	11,25°	6 cm

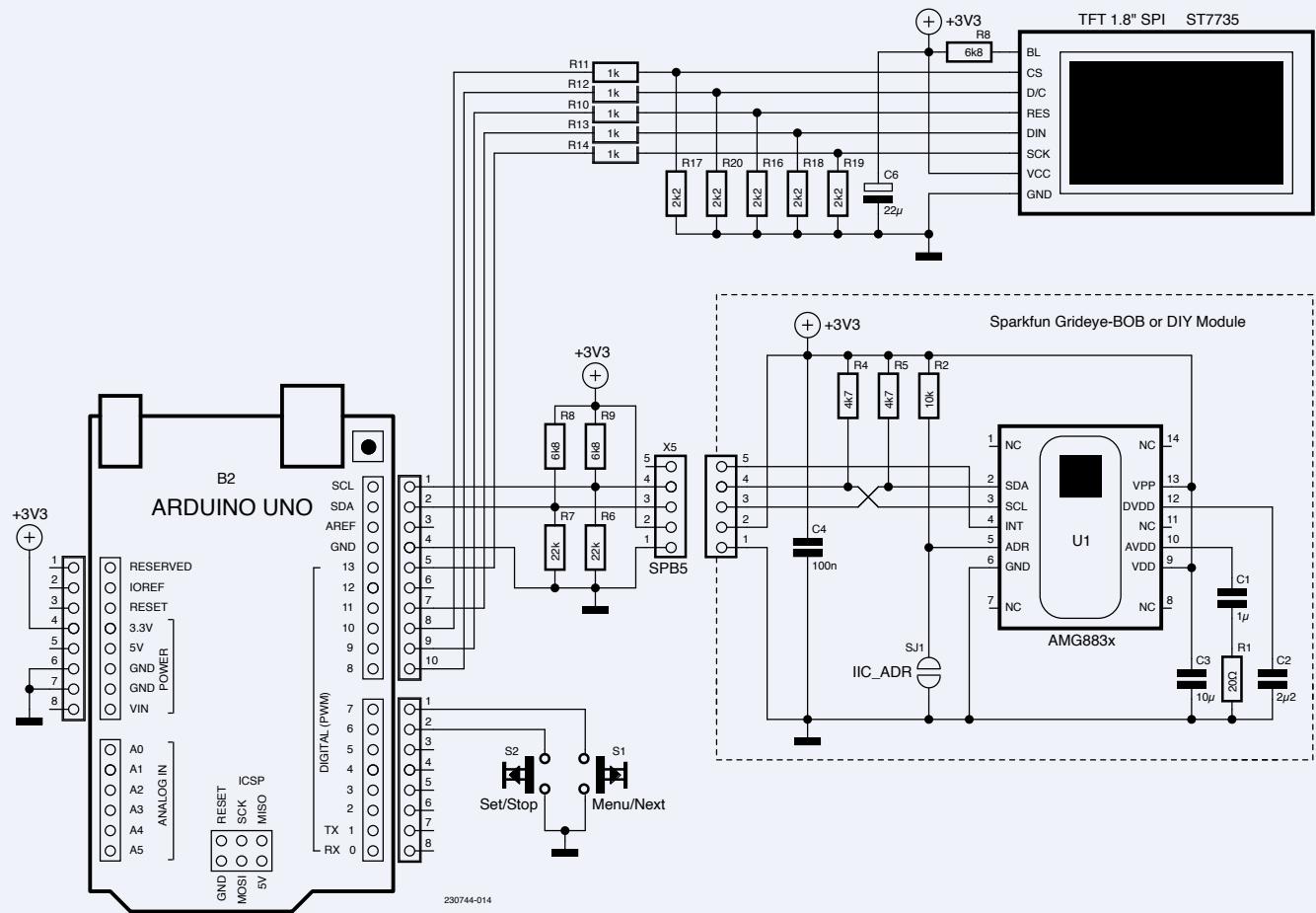


Figure 5. Schéma de la caméra.

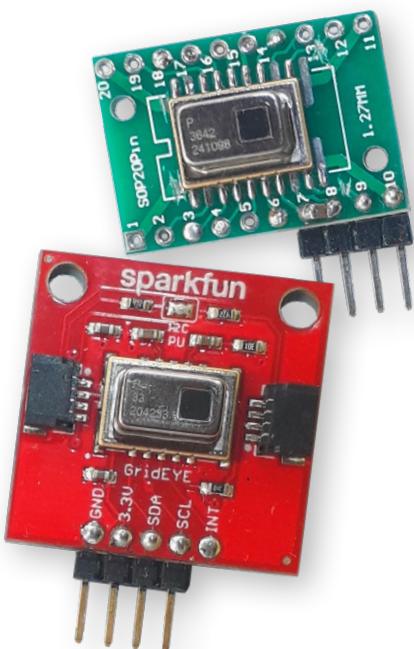


Figure 6. Module de capteur maison à côté d'un BoB de Sparkfun.

La structure de la caméra

Le circuit complet de la caméra thermique se compose d'un Arduino Uno, d'un bouclier avec un petit écran couleur TFT, de deux boutons et du module de la caméra. La **figure 5** montre le schéma complet de la caméra

Module de caméra

Si vous décidez d'utiliser un BoB (*Breakout Board*) [4] (**figure 6**) entièrement assemblé avec un AMG8833, vous avez déjà terminé. Il ne vous reste plus qu'à déterminer si le câblage doit être réalisé avec des Qwiic (la connectique de *SparkFun*) ou des barrettes à broches. Cependant, vous pouvez aussi construire le module vous-même assez facilement, car le capteur peut être soudé sans problème à l'aide d'un fer à souder fin. L'avantage de la version maison est que vous pouvez choisir un capteur approprié. Par exemple, le AMG883642 [3] est très intéressant pour de nombreuses applications en raison de son petit champ de vision et de sa large plage de température.

L'auteur a soudé le capteur sur un module adaptateur SSOP20 vers DIL (1,27 mm par

2,54 mm) en compagnie de quelques autres composants CMS (**figure 7**). L'adresse I²C de l'appareil n'a pas d'importance, car le logiciel détermine l'adresse correcte au démarrage, de sorte que la broche ADDR peut être connectée à la masse.

Bouclier TFT

Le circuit de la figure 5 étant très simple, il peut facilement être construit sur un bouclier Arduino Mega Proto Shield. Il y a suffisamment

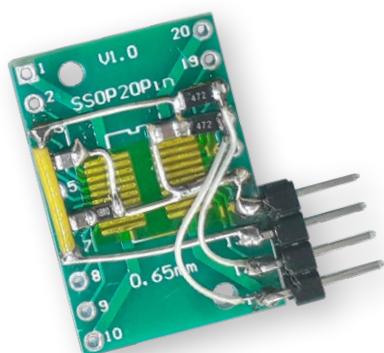


Figure 7. Module de capteur maison, côté arrière.

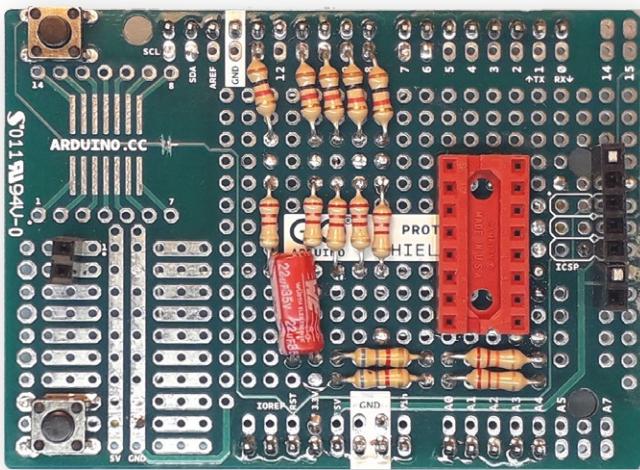


Figure 8. Bouclier maison, côté assemblage.

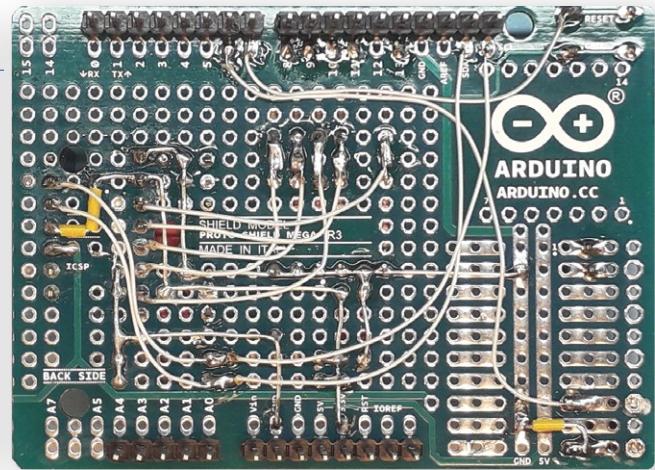


Figure 9. Bouclier maison, côté câblage.

d'espace pour utiliser des composants traversants (**figure 8, figure 9**). L'écran TFT, d'une diagonale de 1,8 pouce et de 128×160 pixels, doit être équipé d'un contrôleur ST7735 et doit pouvoir fonctionner sous 3,3 V. Un produit bon marché d'origine chinoise convient parfaitement. Mais attention : ces cartes ont des brochages variés. Ce n'est pas un problème, mais il faut y prêter attention. Une embase DIL est un très bon connecteur pour l'écran.

Alimentation et bus

Le module caméra est alimenté par l'alimentation 3,3 V de l'UNO. C'est pourquoi les résistances de rappel des lignes I²C sont connectées au 3,3 V. En plus des résistances sur le module de la caméra, il y a également sur le bouclier de l'auteur deux rappels vers le 3,3 V et deux rappels vers le 0 V pour compenser le courant interne de 140 µA provenant des ports. Bien que l'UNO fonctionne à 5 V, il lit les signaux de 3,3 V sans aucune erreur. L'écran TFT est contrôlé via l'interface SPI. Son contrôleur ST7735 travaille exclusivement avec des signaux de 3,3 V. Les signaux du port 5 V de l'UNO sont donc réduits à 3,3 V à l'aide des diviseurs de tension R10 et R14 ainsi que R16 et R20. Comme l'horloge SPI est de 10 MHz, la valeur de ces résistances ne doit pas être trop élevée pour limiter l'effet des parasites. Les valeurs spécifiées représentent un bon compromis entre la consommation de courant et la transmission de signaux sans distorsion. L'emplacement des deux boutons est illustré à la **figure 10**.

Mise en service

Commencez par installer les deux bibliothèques, *Adafruit_ST7735_and_ST7789_Library* et *Adafruit_GFX_Library* en utilisant le gestionnaire de bibliothèque de l'IDE Arduino.

Ensuite, compilez le croquis *Grideye_V2x.ino* à partir de la page du projet Elektor [5] et chargez le programme dans l'UNO.

L'alimentation étant éteinte, enfichez le bouclier, toujours sans le module caméra, et mettez l'UNO sous tension. Le TFT doit s'allumer et afficher le message d'erreur « *no valid device found* ». Vérifiez maintenant la polarité de la tension d'alimentation au niveau du connecteur du module caméra. Les signaux I²C peuvent être mesurés à l'aide d'un oscilloscope. Ensuite, éteignez à nouveau l'appareil, connectez le module de caméra au bouclier et remettez l'ensemble en service.

Si tout va bien, on obtient une image significative. Selon la version du contrôleur ST7735, la position et les couleurs peuvent être incorrectes, ce qui peut être corrigé dans le programme. Le croquis propose quatre routines d'initialisation différentes à cette fin. Quatre valeurs *TFT_TYPE_n* sont définies à cet effet, dont l'une doit être assignée à la constante *TFT_TYPE*. Avec la valeur correcte, l'image est cadrée en haut, à gauche et à droite. Si les couleurs rouge et bleue sont inversées, la valeur de *TFT_CHANGE_COLR* doit être fixée à *true*. Si l'image est à l'envers, il faut attribuer la valeur *02* à *TFT_ORIENTATION*. Enfin, la plage de température du capteur utilisé est sélectionnée en assignant la valeur *true* ou *false* à *AMG88x3*.

```
#define AMG88x3 false
//select sensor type, true if AMG88x3
#define PERMANENT_AUTO false
//if true autoranging always active
#define T_OFFSET 0
//offset for correcting T [°C]

#define TFT_ORIENTATION 00
//portrait format
```

```
//#define TFT_ORIENTATION 02
//portrait format overhead

#define TFT_TYPE_1 00
//module type 1 (w/o SD)
#define TFT_TYPE_2 01
//module type 2
#define TFT_TYPE_3 02
//module type 3 (with SD)
#define TFT_TYPE_4 03
//module type 4
const byte TFT_TYPE= TFT_TYPE_1;
//set used module type

#define TFT_CHANGE_COLR true
//change red-blue depending on tft
```

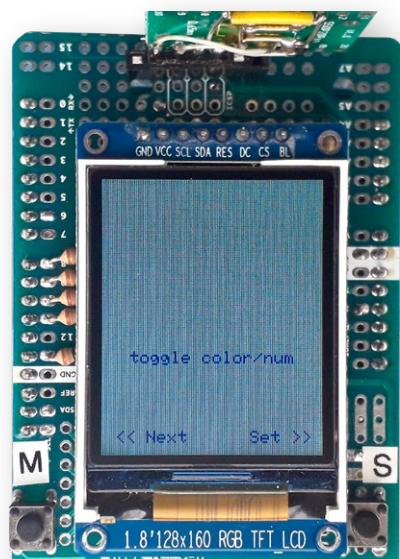


Figure 10. Positionnement des boutons, commutation entre l'affichage graphique et l'affichage numérique.



Listage 1. La fonction `read_GridEYE`.

```
if (acquire==true)
    (read_GridEYE (&lowestT, &highestT,
                  &highestTpix));                                //read all pixels and
                                                               //get lowest and highest T

...
void read_GridEYE (int *pl, int *ph, int *phnum)
{
    *ph= -512; *pl= 1024;                                //init values highest, lowest
    int T;                                                 //temporary
    byte pixnum;

    //  read data of 64 pixels
    //  -----
    for(byte y = 0; y <8; y++)                          //line address
    {
        for(byte x=0; x<8; x++)                        //column address
        {
            pixnum= (y+56)-(x*8);                      //due to orientation, start with pixel 56
            T= GridEye_getT (pixnum);                   //read this pixel
            pixel_Table[y*2][x*2]= T;                  //T to each second column, line position

            if (T>*ph)                                 //find highest temperature
                { *ph=T;
                  *phnum= y*256 + x;
                }
            if (T<*pl) *pl=T;                         //find lowest temperature
        }
    }

    //  interpolate between the horizontal pixels of each line
    //  -----
    for (byte y=0; y<8; y++)                          //line address counter
    {
        for (byte x=0; x<7; x++)                      //column address counter
        {
            T= (pixel_Table [2*y][2*x] +  pixel_Table [2*y][(2*x)+2])/2;
            pixel_Table [2*y][(2*x)+1]= T;
        }
    }

    //  interpolate between the vertical pixels of each column
    //  -----
    for (byte x=0; x<15; x++)                        //column address counter
    {
        for (byte y=0; y<7; y++)                      //line address counter
        {
            T= (pixel_Table [2*y][x] + pixel_Table [(2*y)+2][x])/2;
            pixel_Table [(2*y)+1][x]= T;
        }
    }
}
```

Logiciel

Le croquis a été créé avec l'EDI Arduino. Aucune bibliothèque externe n'est utilisée pour le capteur Grid-EYE ; toutes les fonctions nécessaires sont présentes dans le croquis. Il est donc facile d'apporter des modifications ou des extensions en fonction de vos propres idées.

Le programme principal appelle la routine de mesure `read_GridEYE` (**listage 1**) de manière cyclique (c'est-à-dire que le processus est déterministe). La routine de mesure lit les 64 températures via la fonction `GridEye_getT` (les températures la plus basse et la plus élevée sont également relevées lors de la lecture et enregistrées avec la position de la température la plus élevée).

Les températures ont une résolution de 0,25 K. Pour éviter les nombres fractionnaires, elles sont multipliées par un facteur 4 par le capteur. Le capteur fournit ensuite les valeurs sur 12 bits, le 12^e bit étant le signe. La routine de lecture les convertit en un type `int`, c'est-à-dire 16 bits en complément à deux.

Les valeurs du capteur sont écrites dans le tableau bidimensionnel `pixel_Table`. Ce tableau est de taille 15x15 `int` pour contenir aussi des valeurs additionnelles interpolées. Après avoir lu les valeurs du capteur, la routine calcule des valeurs intermédiaires entre deux valeurs de ligne voisines par interpolation, puis répète le calcul colonne par colonne, en incluant les valeurs déjà interpolées. Les 15x15 = 225 valeurs obtenues sont maintenant disponibles dans le tableau `pixel_Table`. La fonction `showT_as_Color` affiche les 225 valeurs en fausses couleurs. Les couleurs sont attribuées aux valeurs via la table de correspondance de couleur `color_Scale`. Pour qu'elles soient bien distinctes, la résolution est limitée à 32 couleurs.

Les valeurs mesurées doivent donc être redimensionnées de manière à être identifiables à un ensemble de 32 valeurs. C'est ce que fait la fonction `do_Auto_Ranging`.

En fonction de la température la plus basse mesurée, la fonction sélectionne la valeur la plus basse de la plage de mesure parmi six valeurs prédéfinies et la transmet à la variable `Tcoloffset`. La différence entre la température mesurée la plus élevée et `Tcoloffset` est utilisée pour sélectionner la résolution `Tcolrange` de la plage de mesure parmi cinq valeurs prédéfinies. Par soustraction de `Tcoloffset` et multiplication par `Tcolrange`, chaque valeur mesurée est transformée en un nombre représentant l'une des 32 couleurs. La représentation numérique des valeurs mesurées est gérée par `showT_as_Numeric` et leur sortie série est gérée par la fonction `Serial_send_T`. La routine `Set_menu` permet de sélectionner divers réglages et de les appliquer lorsqu'elle est activée. Ces réglages sont enregistrés dans l'EEPROM et restaurés à chaque redémarrage.

Utilisation

Le bouton `Menu` permet d'accéder aux réglages. En appuyant plusieurs fois sur ce bouton, vous pouvez faire défiler les éléments sélectionnables. Les réglages actuellement actifs sont affichés en vert. Appuyez sur le bouton `Set` pour appliquer l'entrée affichée. Le premier paramètre est le *calibre automatique*, qui permet généralement d'obtenir un affichage optimal. Avec l'*option Couleur/Num* (voir figure 10), les résultats peuvent être affichés numériquement ou en fausses couleurs.

Si l'on appuie sur le bouton `Set` pendant le processus de mesure normal, la mesure est arrêtée et figée. L'affichage peut alors être modifié dans le menu sans changer les valeurs mesurées. Appuyez à nouveau sur ce bouton pour poursuivre la mesure.

Dans l'affichage en fausses couleurs, la température du point le plus chaud est affichée sous forme de valeur numérique dans le graphique. Cette fonction peut être supprimée lors du redémarrage en maintenant le bouton `Menu`

enfoncé pendant la réinitialisation. Les résultats sont transmis en continu via l'interface série avec une résolution de 0,25 K.

En conclusion, on ne peut que s'émerveiller de ce qu'un si petit processeur ATmega est capable de faire ! 

Vf : Helmut Müller — 230744-04



À propos de l'auteur

Roland Stiglmayr a étudié les technologies de l'information dans les années 1970 et possède plus de 40 ans d'expérience en recherche et développement. Ses travaux ont porté sur le développement d'ordinateurs centraux, de systèmes de transmission de données par fibre optique, de RRH (Remote Radio Head) pour les communications mobiles et de systèmes de transmission d'énergie sans fil. Aujourd'hui, il travaille en tant que conseiller. Il est particulièrement attaché au transfert de connaissances.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (1134-715@online.de) ou contactez Elektor (redaction@elektor.fr).



Produits

➤ **SparkFun Grid-EYE Infrared Array Breakout - AMG8833**
www.elektor.fr/19605



LIENS

[1] Documentation du Grid-EYE : <https://industrial.panasonic.com/ww/products/pt/grid-eye>

[2] Thermopile (Wikipedia) : <https://fr.wikipedia.org/wiki/Thermopile>

[3] AMG883462 data sheet: <https://industrial.panasonic.com/cdbs/www-data/pdf/ADI8000/ast-ind-139049.pdf>

[4] Grid-EYE-BoB : <https://sparkfun.com/products/14607>

[5] Page du projet : <https://elektormagazine.fr/230744-04>



mise à jour du projet #3 : **compteur d'énergie**

basé sur l'ESP32

intégration et test avec
Home Assistant

Saad Imtiaz (Elektor)

Dans la précédente mise à jour du projet, nous vous avons présenté les améliorations apportées au schéma et au circuit imprimé du compteur d'énergie ESP32. Dans cet article, nous nous concentrerons sur la réalisation pratique et l'intégration de cette nouvelle version. Vous trouverez un guide détaillé sur la configuration du compteur avec ESPHome et Home Assistant pour un contrôle efficace de la consommation d'énergie. En outre, nous abordons le calibrage de l'appareil.

Dans l'épisode précédent [1], nous avons exploré les améliorations apportées au schéma et au circuit imprimé du compteur d'énergie ESP32, en mettant l'accent sur les améliorations de la modularité et des caractéristiques de sécurité. Avant de nous pencher sur la prochaine mise à jour du projet, faisons un bref rappel.

Dans les dernières avancées de notre projet de compteur d'énergie ESP32, nous avons opté pour le microcontrôleur ESP32-S3, ce qui a

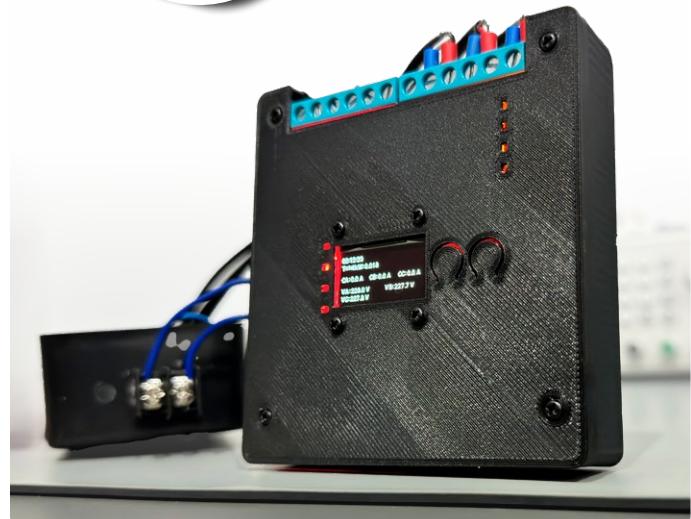


Figure 1. Le compteur d'énergie ESP32 assemblé avec son écran OLED et ses indicateurs d'état en direct.

permis d'améliorer la puissance de traitement et d'élargir les fonctionnalités. Cette mise à jour nous a permis de réduire la taille du circuit imprimé et d'intégrer un système d'alimentation basé sur un transformateur. Nous utilisons un transformateur abaissageur de tension 230 V à 12 V pour l'échantillonnage de la tension et pour alimenter le système. Cela renforce considérablement la sécurité tout en assurant la compatibilité avec les installations monophasées et triphasées.

Nous avons également introduit d'autres améliorations notables, telles que l'intégration d'un convertisseur buck AP63203WU-7 plus efficace, la modularité du circuit imprimé et le calibrage du circuit d'échantillonage du transformateur de courant. Ces améliorations permettent non seulement d'optimiser les performances et les fonctionnalités du compteur d'énergie, mais aussi de réduire son coût et sa taille.

Dans cet article, nous discuterons des étapes suivies pour rendre ce compteur d'énergie opérationnel et du chemin qu'il a parcouru depuis les étapes initiales sur l'établi jusqu'à son intégration dans la boîte à disjoncteurs. En outre, nous aborderons aussi la calibration et la configuration du système, et enfin, son intégration à Home Assistant avec ESP Home pour afficher et surveiller les données collectées à partir du compteur d'énergie. La **figure 1** montre le projet en action, logé dans un boîtier imprimé en 3D et équipé d'un écran OLED. L'image met en évidence des indicateurs d'état qui suivent et affichent la consommation d'énergie en temps réel de manière claire et instantanée.

Assemblage

Le nouveau circuit imprimé a été conçu pour être plus compact et plus facile à souder. La disposition assure un espace adéquat pour chaque composant, rendant ainsi le processus de soudure plus aisés. Pour faciliter la reproduction du projet et permettre des modifications tant que pour les passionnés que pour les professionnels, nous avons partagé la liste complète des composants (BOM) au format Mouser et les fichiers de production sur le dépôt GitHub d'Elektor Lab [2].

Pour les connexions d'échantillonnage de tension et de courant, nous avons utilisé des borniers à vis de CUI Devices. La qualité de ces borniers est bien meilleure que celle des borniers bleus bon marché souvent utilisés sur de nombreux modules de capteurs. Comme nous travaillons avec des tensions alternatives et des compteurs d'énergie, il est vital d'avoir des connexions sûres et fiables.

La réduction du bruit est un aspect critique dans la conception de ce circuit imprimé. Pour y répondre, nous avons intégré des condensateurs électrolytiques et céramiques autour de la puce de mesure de l'énergie ATM90E32S. Cette disposition permet de filtrer les bruits de basse et de haute fréquence, assurant ainsi une mesure d'énergie plus précise et plus stable. La carte est représentée dans la **figure 2**.

Comme mentionné précédemment, nous avons décidé d'utiliser un transformateur abaisseur de tension pour l'échantillonnage de la tension et comme source principale d'alimentation de l'ensemble du système. Trouver un tel transformateur est facile et abordable, mais la plupart de ces transformateurs abaisseurs prennent beaucoup de place lorsqu'ils sont utilisés dans un boîtier personnalisé, comme le montre la **figure 3**. Il est donc préférable d'utiliser des transformateurs DIN Rail Bell dans ce cas pour rendre l'installation plus propre et plus sûre ; de tels transformateurs peuvent être facilement trouvés en ligne. En outre, la précision des mesures de tension dépend des caractéristiques des transformateurs, notamment de la précision de leur rapport de tension, du déphasage et de la linéarité.

Vous avez peut-être remarqué dans les images qu'un unique transformateur est connecté au compteur d'énergie. Le compteur d'énergie a été configuré pour être utilisé en mode monophasé, en court-circuitant le cavalier JP8 situé sur la face arrière du circuit imprimé, comme le montre la **figure 4**. Pour faire fonctionner le compteur d'énergie en mode triphasé ou pour mesurer la tension de chaque phase dans un système triphasé avec trois transformateurs abaisseurs, vous devez brancher les côtés primaires de trois transformateurs aux phases correspondantes (L1, L2, L3). Du côté secondaire, relier une extrémité de l'enroulement de chaque transformateur à un point neutre commun pour former une configuration en étoile (Y). Les extrémités libres des enroulements secondaires (V1, V2, V3) fourniront alors les sorties de tension (UA, UB et UC) sur le circuit imprimé pour chaque phase. Il est essentiel de s'assurer que les transformateurs sont adaptés à la tension et au courant du système, de maintenir une isolation rigoureuse entre les circuits primaires et secondaires pour des raisons de sécurité et d'assurer une connexion neutre stable et équilibrée afin d'éviter toute imprécision dans les mesures.

Configuration dans ESPHome et Home Assistant

Dans le cadre de notre plan de développement, nous sommes en train de développer un micrologiciel spécialisé pour exploiter les capacités de la puce de mesure de l'énergie et les fonctions avancées d'intelligence artificielle de l'ESP32-S3. Bien que le développement d'un tel micrologiciel nécessite beaucoup de temps et soit toujours en cours, cela ne limite pas l'usage du compteur d'énergie. L'appareil est déjà plein fonctionnel avec les plateformes existantes telles que Home Assistant, offrant ainsi une solution immédiate pour la gestion de l'énergie. Dans cet article, nous allons nous concentrer sur

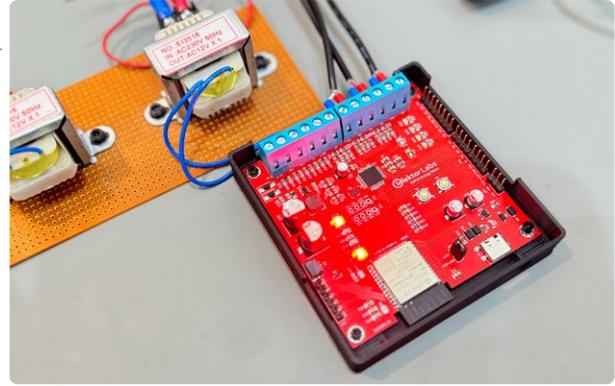


Figure 2. Circuit imprimé du compteur d'énergie ESP32 entièrement assemblé.



Figure 3. Transformateur abaïsseur de tension 220 V à 12 V dans un boîtier personnalisé.



Figure 4. Configuration des cavaliers pour un fonctionnement monophasé sur la carte du compteur d'énergie ESP32.

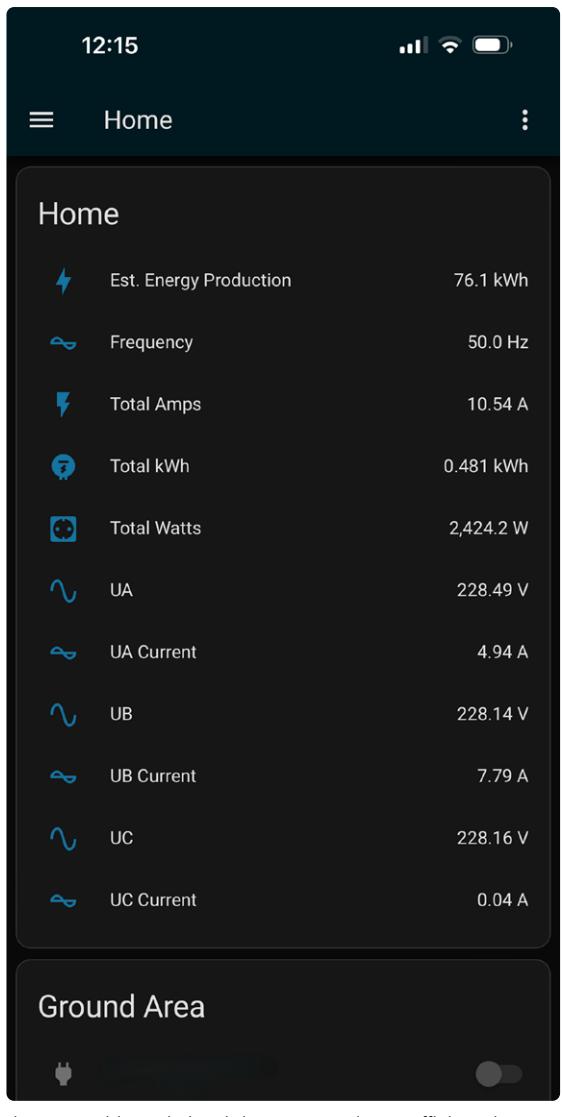


Figure 5. Tableau de bord du Home Assistant affichant les données énergétiques en temps réel du compteur d'énergie ESP32.



Figure 6. Graphe détaillé de l'historique de la consommation d'énergie dans Home Assistant.

tableau de bord de votre Home Assistant, il suffit d'attribuer l'appareil ESPHome à une zone spécifique dans Home Assistant. Cela permet d'organiser votre tableau de bord en regroupant les appareils selon leur emplacement physique ou logique dans votre domicile. Pour une illustration de ce que vous pouvez obtenir, consultez la **figure 5**, qui montre les données du compteur d'énergie sur le tableau de bord de Home Assistant.

L'intégration du compteur d'énergie ESP32 dans Home Assistant simplifie non seulement le processus de gestion de la consommation d'énergie, mais ouvre également la voie à de nombreuses fonctions puissantes offertes par la plateforme. Home Assistant offre une interface intuitive pour la visualisation des données en temps réel, l'automatisation de la gestion et l'intégration transparente avec d'autres appareils connectés dans votre maison. Grâce à cette intégration, il est possible de créer des graphiques temporels détaillés et d'effectuer des analyses dans Home Assistant, offrant une compréhension approfondie sur les modèles de consommation d'énergie au fil du temps, comme le montre la **figure 6**. Ces informations permettent aux utilisateurs de prendre des décisions éclairées concernant leur consommation d'énergie, d'identifier les économies potentielles et d'optimiser l'efficacité énergétique de leur maison.

En suivant le processus d'installation décrit ci-dessus, les utilisateurs peuvent tirer pleinement parti de ces capacités, transformant le compteur d'énergie ESP32 en un élément essentiel de leur écosystème de maison intelligente. Cette intégration ne se contente pas d'améliorer le fonctionnement du compteur d'énergie, mais enrichit également l'expérience globale de la maison intelligente grâce à des outils complets de surveillance et de gestion de l'énergie.

Configuration YAML

La configuration YAML proposée permet de configurer le compteur d'énergie ESP32 avec ESPHome, ce qui permet de surveiller les paramètres électriques clés tels que la tension, le courant et la puissance sur les trois phases. Elle tire parti des capacités du capteur ATM90E32, avec des définitions détaillées pour la communication SPI



Figure 7. Configuration pour tester et calibrer le compteur d'énergie ESP32 en utilisant une charge variable.



Figure 8. Pince ampèremétrique pour le calibrage.

et les capteurs spécifiques pour chaque phase. Cette configuration permet non seulement de mesurer mais aussi de calculer la consommation totale, en intégrant un compteur d'énergie quotidien en kWh et un écran OLED pour la visualisation des données en temps réel. Ces configurations sont réalisées conformément aux instructions de la page ESPHome pour le capteur ATM90E32 [6].

Pour garantir la précision des données fournies par le compteur d'énergie ESP32, le calibrage est une étape cruciale. Les détails de l'ajustement des paramètres de gain pour les transformateurs de courant et les entrées de tension seront détaillés dans la section suivante.

Configuration de test et calibrage

Lors de nos tests, nous avons utilisé un sèche-cheveux à plusieurs niveaux de chaleur et de vitesse comme charge, couvrant une gamme de 0,7 A à 8 A. Pour faciliter la mise en place des transformateurs de courant ouverts sur le fil sous tension ou le fil neutre, le cordon d'alimentation d'une rallonge électrique a été dénudé, facilitant ainsi la surveillance directe dans diverses conditions, comme le montre la **figure 7**.

J'ai réalisé le calibrage en courant et en tension du compteur d'énergie ESP32 à l'aide de mon multimètre UT201+. Ce multimètre offre une résolution de 0,001 A et précision de $\pm 4\%$ +10 chiffres pour le courant et une résolution de 0,001 V avec une précision de $\pm 1\%$ +5 chiffres pour la tension. Bien que ce niveau de précision soit suffisant pour la plupart des projets, mais il est légèrement moins précis que celui offert par les modèles de qualité professionnelle.

Lors du calibrage, nous avons comparé les mesures de courant : la pince ampèremétrique indiquait 1,692 A, comme le montre la **figure 8**, tandis que les valeurs calculées par le compteur d'énergie avaient

```

Logs esp32-energymeter.yaml

preferences to flash...
[22:53:59] [D][esp32.preferences:143]: Saving 1
preferences to flash: 0 cached, 1 written, 0 failed
[22:53:59] [D][sensor:094]: 'Total Watts': Sending
state 1207.07874 W with 1 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Total kWh': Sending
state 0.02185 kWh with 3 decimals of accuracy
[22:53:59] [W][atm90e32:366]: SPI power factor read
error.
[22:53:59] [D][sensor:094]: 'Line Voltage A':
Sending state 233.89999 V with 2 decimals of
accuracy
[22:53:59] [D][sensor:094]: 'Line Voltage B':
Sending state 0.42000 V with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Line Voltage C':
Sending state 0.15000 V with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'CT1 Current': Sending
state 1.73400 A with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'CT2 Current': Sending
state 1.72100 A with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'CT3 Current': Sending
state 1.70300 A with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Active Power CT1':
Sending state -392.60928 W with 2 decimals of
accuracy
[22:53:59] [D][sensor:094]: 'Active Power CT2':
Sending state 0.11808 W with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Active Power CT3':
Sending state 0.08352 W with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Power Factor CT1':
Sending state -0.96700 with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Power Factor CT2':
Sending state 0.15500 with 2 decimals of accuracy

```

DOWNLOAD LOGS EDIT CLOSE

NEW DEVICE

Figure 9. Résultats du calibrage final montrant une amélioration de la précision des mesures.



ATTENTION : Travailler avec un tableau électrique comporte des risques inhérents, notamment un risque d'électrocution ou d'incendie. Il est essentiel de couper l'alimentation électrique avant de commencer l'installation. Dans la plupart des pays, ce travail ne peut être effectué que par un électricien certifié !

entre 1,70 et 1,73 A après le calibrage, comme le montre la **figure 9**. Compte tenu des spécifications de l'UT201+ et du SCT-013-000, un transformateur de courant ouvrant de classe 1 qui garantit une précision de 1% par rapport à la valeur réelle, ce faible écart reste dans la marge d'erreur prévue. Toutefois, pour une précision encore plus grande, il est possible d'utiliser une pince ampèremétrique plus précise.

Pour améliorer encore la précision du compteur d'énergie ESP32, des ajustements ont été apportés aux paramètres de gain pour les mesures de tension et de courant. Pour la tension, le capteur a été calibré en utilisant la formule suivante :

$$\text{New gain_voltage} = (\text{your voltage reading} / \text{ESPHome voltage reading}) * \text{existing gain_voltage value}$$

De même, pour les ajustements de courant :

$$\text{New gain_ct} = (\text{your current reading} / \text{ESPHome current reading}) * \text{existing gain_ct value}$$

Nous avons ensuite mis à jour ces nouvelles valeurs de gain dans le fichier de configuration YAML de l'ESPHome, puis nous avons recompilé et chargé le micrologiciel. Vous pouvez répéter ce processus autant de fois que nécessaire pour garantir une précision optimale. Ces valeurs calibrées permettent d'affiner les mesures et sont cruciales pour l'établissement de rapports et d'analyses précis dans toute installation de surveillance de la consommation d'énergie.



Figure 10. Compteur d'énergie ESP32 installé dans un tableau de disjoncteurs, surveillant la consommation d'énergie en temps réel.

Installation du compteur ESP32 dans le tableau électrique

L'installation du compteur d'énergie ESP32 dans mon tableau de disjoncteurs a été un processus gérable qui a exigé une attention méticuleuse aux détails pour garantir la sécurité et le bon fonctionnement du système. J'ai commencé par sélectionner un circuit avec la limite d'ampérage la plus basse. Ce choix stratégique offrait une marge de sécurité ; le disjoncteur se déclencherait en cas de surtension imprévue ou de défaillance du transformateur, offrant ainsi une protection au système.

L'utilisation de transformateurs de courant ouvrants offre un avantage notable grâce à leur simplicité d'installation. Il est possible de fixer rapidement ces transformateurs sur n'importe quelle charge, mais il est crucial de prêter attention au sens du courant pour garantir la précision des mesures. Il est important de noter que si le sens du courant et l'orientation du transformateur de courant ne sont pas correctement alignés, les relevés de puissance apparaîtront négatifs, ce qui indique une installation incorrecte.

La **figure 10** présente une démonstration visuelle du compteur d'énergie ESP32 en action dans le tableau électrique. Cette image montre le compteur d'énergie affichant en temps réel le courant, les mesures de tension et la charge correspondante en kilowatts, illustrant ainsi son fonctionnement en pratique.

Développement et perspectives

Bien que la configuration logicielle actuelle soit fonctionnelle sur ESPHome, nous poursuivons le développement pour étendre les capacités du compteur d'énergie ESP32. Nous travaillons actuellement sur l'intégration d'un nouveau micrologiciel spécialement conçu pour exploiter tout le potentiel de la puce ESP32-S3. Ce micrologiciel devrait inclure des fonctions avancées telles que l'analyse détaillée de l'énergie et des fonctionnalités IA/ML potentiellement révolutionnaires qui pourraient prédire les schémas de consommation énergétique et identifier les appareils à partir de l'empreinte de leur charge.

Bien que les aspects fondamentaux et opérationnels du projet soient achevés, le développement de fonctionnalités plus sophistiquées est une tâche complexe qui prend du temps. Je suis enthousiasmé par les possibilités qui s'offrent et je suis déterminé à repousser les limites de ce que ce compteur d'énergie peut réaliser.

Le projet de compteur d'énergie ESP32 continue d'évoluer, intégrant de nouvelles fonctions à chaque mise à jour. Nous encourageons vivement les membres de notre communauté qui sont intéressés par les futures fonctionnalités d'IA et de ML, ou ceux qui souhaitent contribuer au développement à s'impliquer. La collaboration est essentielle pour accélérer les progrès et aboutir à une solution de surveillance d'énergie plus robuste et plus riche en fonctionnalités. Restez attentifs aux prochaines avancées, nous sommes déterminés à perfectionner et à améliorer cet outil polyvalent de gestion de l'énergie. ↗

240244-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (saad.imtiaz@elektor.com) ou contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Saad Imtiaz (ingénieur senior, Elektor) est un ingénieur mécatronicien expérimenté, spécialisé dans les systèmes embarqués, la mécatronique et le développement de produits.

Saad a également travaillé dans l'industrie aéronautique et a dirigé une startup technologique. Chez Elektor, il supervise le développement de projets dans les domaines du logiciel et du matériel.



Produits

➤ **PeakTech 4350 Pince ampèremétrique**
www.elektor.fr/18161

➤ **Siglent SDM3045X Multimètre**
www.elektor.fr/17892

LIENS

- [1] Saad Imtiaz, « mise à jour #2 : compteur d'énergie basé sur l'ESP32 », Elektor 5-6/2024 :
<https://www.elektormagazine.fr/magazine/elektor-344/62819>
- [2] Compteur d'énergie ESP32 | Dépôt Github : <https://github.com/ElektorLabs/esp32-energymeter>
- [3] Home Assistant : <https://home-assistant.io/>
- [4] ESPHome : <http://esphome.io>
- [5] Clemens Valens, « la domotique, c'est facile avec ESPHome, Home Assistant et MySensors », Elektor Magazine 9-10/2020 :
<https://www.elektormagazine.fr/magazine/elektor-156/58991>
- [6] ATM90E32 Power Sensor : <https://esphome.io/components/sensor/atm90e32.html>

2024 l'odyssée de l'IA

améliorer la détection d'objets : intégration de techniques avancées

Brian Tristam Williams (Elektor)

Après avoir réussi à déployer le principe fondamental de la détection d'objets sur un Raspberry Pi sans tête (*headless*), nous poursuivons notre aventure en nous concentrant sur l'amélioration de cette technologie. Dans cet épisode nous abordons les améliorations techniques visant à accroître la précision et l'efficacité, ainsi que l'intégration de capteurs supplémentaires afin d'améliorer le fonctionnement dans diverses conditions environnementales.

Après avoir rencontré des difficultés pour détecter des objets et du texte sur de vieilles vidéos d'archives, j'ai dû chercher des moyens d'améliorer les algorithmes de détection.

Dans mon exemple, les vidéos entrelacées et de faible résolution sur lesquelles je travaillais ont probablement semé la confusion. Les différences entre les demi-trames alternées sont susceptibles de perturber le modèle de détection, conçu pour analyser une image (composée de deux demi-trames) (**figure 1**). Il est essentiel que j'examine le désentrelacement de l'ancienne vidéo capturée avant de l'intégrer dans le flux de travail.

Amélioration de la détection

Malgré les défis de niche que j'ai rencontrés, cela ne signifie pas que nous ne pouvons pas améliorer la détection avec quelques modifications. Voici ce que j'ai trouvé utile en travaillant avec une vidéo numérique non entrelacée provenant du Raspberry Pi Camera Module 3 :

Seuil dynamique : pour pallier les variations de l'éclairage et des distances, j'ai établi un seuil de confiance adaptatif. Cette méthode permet d'ajuster les niveaux de confiance de la détection en temps réel en fonction de la luminosité moyenne détectée par la caméra, grâce à un algorithme de calcul de luminosité simple :

```
def adjust_threshold(lux):
    base_threshold = 0.5 # base confidence level
    if lux < 50: # low light conditions
```



Figure 1. La vidéo PAL et NTSC utilise une méthode entrelacée pour dessiner les lignes de balayage, ce qui affecte les images complètes en cas de mouvement.

```
return base_threshold - 0.1
elif lux > 500: # very bright conditions
    return base_threshold + 0.1
return base_threshold
```

Intégration du capteur de luminosité : En parlant d'éclairage, bien que la caméra puisse fonctionner comme un capteur de luminosité, mes essais passés d'ajuster les nombreux paramètres du module caméra pour obtenir des photos et vidéos « pafais », je sais que la relation entre la luminosité du signal vidéo de sortie et la luminosité réelle à l'extérieur sera ténue – c'est juste une variable confondante de plus dont je ne veux pas gérer la frustration, et j'ai donc opté pour l'utilisation d'un capteur à cet effet.

J'ai choisi le capteur de lumière TLS2561 (**figure 3**) pour fournir des données en temps réel au Raspberry Pi, qui ajuste alors dynamiquement l'exposition de la caméra ainsi les paramètres de traitement. Cela permet d'assurer une qualité d'image optimale pour les algorithmes de détection dans des conditions d'éclairage fluctuantes :

```
lux = read_lux_sensor()
camera.set_exposure(calculate_exposure(lux))
```

Encore une fois, je n'ai pas eu à écrire ces fonctions moi-même, car il y existe une excellente bibliothèque Adafruit-TSL2561 disponible [1]. L'installation est simple via le terminal :



Figure 2. Le capteur de luminosité TSL2561 sur un module Adafruit.

```
sudo apt-get update
sudo apt-get install python3-smbus
pip3 install Adafruit-TSL2561
```

Ce code Python vous permet ensuite de lire et d'éditer les valeurs du capteur :

```
import time
from Adafruit_TSL2561 import TSL2561

# Initialize the sensor
tsl = TSL2561()

while True:
    lux = tsl.calculate_lux()
    print("Current Lux: ", lux)
    time.sleep(1) # Delay for 1 second
```

De nos jours, les bibliothèques disponibles sur internet facilitent grandement les choses. À l'époque, il fallait consulter la fiche technique de chaque nouveau capteur acheté et créer ses propres fonctions ou sous-programmes.

Amélioration du traitement des images : En intégrant la fonction `cv2.GaussianBlur()` avant la détection d'objet, j'ai pu réduire l'impact du bruit et du grain dans les images vidéo à faible luminosité. J'ai trouvé cela particulièrement utile avec le Raspberry Pi Camera Module 3 NoIR (**figure 2**, à droite) que j'ai testé pour la surveillance nocturne. Ce modèle de caméra ne dispose pas de filtre infrarouge et peut donc capter bien les éclairages infrarouges. Cependant, son efficacité dépend toujours de la distance entre votre éclairage infrarouge et le sujet, et parfois les objets distants posent un problème.

L'utilisation de cette fonction a permis au modèle de traiter les images avec beaucoup moins de bruit, bien que cela ait affecté la netteté de l'image d'entrée. Cependant, lorsque vous essayez de détecter des objets à grande échelle sans avoir besoin de détails fins tels que la lecture de plaques d'immatriculation, la netteté des « humains » détectés importe peu au modèle – c'est à l'utilisateur de vérifier les détails du flux de la caméra lorsqu'une alerte est déclenchée. L'appel de la fonction se fait en une seule ligne de code :

```
frame = cv2.GaussianBlur(frame, (5, 5), 0)
```

Prise en charge de plusieurs modèles : Au début, j'ai opté pour un seul modèle après en avoir testé plusieurs, mais il était un peu frustrant de constater qu'aucun ne convenait à toutes les configurations et à tous les scénarios. Puis j'ai réalisé, lentement, qu'il était possible de basculer entre différents modèles à la volée.

L'intégration d'un modèle de détection secondaire spécialisé pour des

cibles spécifiques améliore la précision. Par exemple, l'intégration d'un modèle YOLO rationalisé augmente la robustesse de la détection des véhicules et des piétons. Le passage d'un modèle à l'autre en fonction du contexte de la scène est géré comme dans cet exemple :

```
if scene == 'urban':
    model.load('yolo_city.tflite')
else:
    model.load('tensorflow_lite_default.tflite')
```

Réduction de la latence : Malheureusement, l'augmentation de la complexité de la détection a entraîné une augmentation de la latence. J'ai essayé d'implémenter le traitement asynchrone des modèles, ce qui a permis d'accélérer les choses en termes de chiffres bruts, même si l'amélioration n'était pas évidente à mes yeux. Néanmoins, toute amélioration est bienvenue. Voici donc comment l'essayer :

```
from concurrent.futures import ThreadPoolExecutor

with ThreadPoolExecutor() as executor:
    future = executor.submit(process_frame, frame)
    result = future.result()
```

Nous voyons ici l'utilité de la fonction `ThreadPoolExecutor()` du module `concurrent.futures`, qui fait partie de la bibliothèque standard de Python. Cette méthode permet de gérer l'exécution asynchrone de tâches dans des threads distincts, ce qui peut s'avérer particulièrement utile pour des applications comme le traitement vidéo en temps réel où la réactivité est essentielle.

Après la première ligne, où l'on intègre la fonction, les trois lignes suivantes font respectivement ce qui suit :

1. Créer une instance de `ThreadPoolExecutor`, qui gère un pool de threads pour l'exécution asynchrone des appels. L'instruction `with` garantit que les ressources de l'exécuteur sont correctement gérées, en fermant automatiquement l'exécuteur lorsqu'il n'est plus nécessaire.
2. La méthode `submit()` planifie l'exécution de la fonction `process_frame()` en lui passant `frame` comme argument. La fonction est exécutée dans un thread distinct géré par l'exécuteur. `submit()` renvoie un objet `future`, qui représente le résultat potentiel de l'appel de la fonction.
3. Bloquer le thread principal jusqu'à ce que la fonction `process_frame()` s'exécute et renvoie un résultat. La méthode `result()` récupère le résultat de l'exécution de la fonction. Si une exception est soulevée durant l'exécution, celle-ci sera soulevée à nouveau ici.

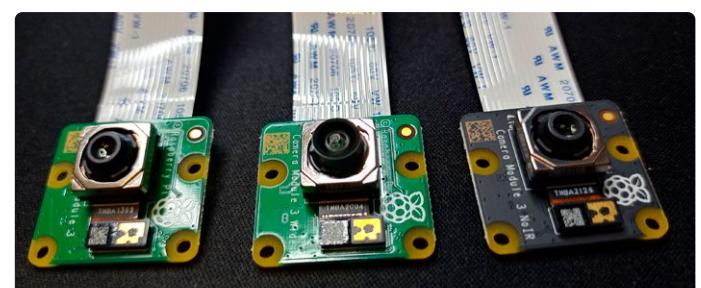


Figure 3. Outre la Camera Module 3 large et ordinaire, j'ai essayé la version NoIR pour sa capacité à bénéficier d'un éclairage infrarouge la nuit.

En déchargeant les tâches gourmandes en ressources sur des threads séparés, le thread principal du programme peut rester réactif. Ceci est particulièrement essentiel dans les applications graphiques ou en temps réel, où une latence élevée peut nuire à l'expérience de l'utilisateur. Le déploiement de ce système au sein d'un réseau de surveillance domestique a prouvé sa capacité à distinguer une personne d'un chien ou d'un arbre. Toutefois, il n'atteint pas le niveau de sophistication requis pour la détection de visages comme le ferait un système de reconnaissance faciale.

Pour aller plus loin

Outre le désentrelacement et le prétraitement nécessaires pour avancer sur mon projet de traitement de vieilles vidéos, de nombreuses autres possibilités s'offrent avec TensorFlow Lite, et je ne sais pas laquelle sera la plus intéressante. N'hésitez donc pas à me contacter pour me faire part de toute piste que vous aimeriez explorer ensemble !

Voici quelques tâches que nous pouvons explorer :

- **Traitement d'images et de vidéos :** Je continue à travailler sur des applications de traitement avancé d'images et de vidéos, comme la segmentation d'images, où le modèle identifie différents objets dans l'image et les sépare de l'arrière-plan, ou le transfert de style, qui permet d'appliquer le style d'une image au contenu d'une autre.
- **Détection et classification d'objets en temps réel :** TensorFlow Lite exécute des modèles qui détectent et classifient des objets en temps réel, ce qui est utile pour des applications telles que les caméras de sécurité, le contrôle qualité automatisé et la surveillance de la faune. Des modèles pré-entraînés tels que MobileNet peuvent être adaptés et déployés sur des appareils pour identifier des objets efficacement.
- **Reconnaissance de la parole et traitement audio :** TensorFlow Lite gère des modèles de reconnaissance vocale, ce qui permet de créer des applications à commande vocale, de convertir la parole en texte et d'effectuer d'autres tâches de traitement audio. Pour moi, cela pourrait être un moyen de préserver des histoires oubliées des archive, mais cela permet également de développer des commandes mains libres ou des outils d'assistance pour personnes handicapées.
- **Traitement du langage naturel (NLP) :** Le framework prend en charge des modèles NLP légers qui peuvent effectuer des tâches telles que l'analyse des sentiments, la détection du langage, ou même alimenter de simples chatbots. C'est particulièrement utile dans les applications nécessitant une interaction et traitement des retours utilisateurs.
- **Reconnaissance des gestes :** En utilisant TensorFlow Lite, vous pouvez développer des systèmes qui comprennent et interprètent les gestes humains comme des commandes, permettant des installations interactives ou améliorant les interfaces utilisateur dans les appareils.
- **Détection d'anomalies :** TensorFlow Lite peut être utilisé pour la détection d'anomalies ou de valeurs aberrantes dans les données

de séries temporelles, ce qui est précieux pour la maintenance prédictive dans les environnements industriels ou les systèmes de surveillance tels que la détection de battements cardiaques irréguliers dans les appareils de santé.

➤ **Estimation de la pose :** Le framework est capable d'exécuter des modèles d'estimation de la pose qui détectent les figures humaines et leurs postures dans les images ou les vidéos. Cela peut être appliqué à l'analyse sportive, aux applications de fitness et aux applications interactives avancées, comme démontré lors de notre récente visite au salon embedded world 2024.

Grâce aux améliorations apportées à TensorFlow Lite sur Raspberry Pi, nous avons réalisé des progrès significatifs en matière de détection d'objets et de texte, bien que les vidéos entrelacées et de faible résolution présentent toujours des défis. Ces améliorations pourraient aider les amateurs à gérer efficacement les applications en temps réel et sont particulièrement utiles pour des projets tels que la surveillance domestique et l'archivage de médias. Il reste encore beaucoup à explorer et à perfectionner, et j'ai hâte de voir comment nous pourrons adapter ces outils à nos besoins. ↗

230181-G-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur
(brian.williams@elektor.com).



À propos de l'auteur

Brian Tristam Williams est fasciné par les ordinateurs et l'électronique depuis qu'il a eu son premier « micro-ordinateur » à l'âge de 10 ans. Son aventure avec le magazine Elektor a commencé lorsqu'il a acheté son premier numéro à 16 ans, et depuis lors, il suit le monde de l'électronique et de l'informatique, explorant et apprenant sans cesse. Il a commencé à travailler pour Elektor en 2010 et, aujourd'hui, il s'attache à suivre les dernières tendances technologiques, en se concentrant notamment sur l'IA et les ordinateurs monocartes tels que le Raspberry Pi.



Related Products

- **Raspberry Pi 5 (4 GB)**
www.elektor.fr/20598
- **Raspberry Pi Camera Module 3**
www.elektor.fr/20362
- **Raspberry Pi Camera Module 3 NoIR**
www.elektor.fr/20363

LIEN

[1] Bibliothèque Adafruit TSL2561 pour Python sur le Raspberry Pi : <https://github.com/adafruit/TSL2561-Arduino-Library>



Figure 1. Contenu de la boîte du kit Raspberry Pi AI.

Raspberry Pi a lancé un kit comprenant le M.2 HAT+ associé à l'accélérateur d'IA Hailo-8L. Cette nouvelle combinaison offre un surplus de puissance conséquent pour l'IA embarquée sur Raspberry Pi 5, offrant aux développeurs et ingénieurs des performances d'inférence impressionnantes.

Pourquoi faire de l'IA périphérique ?

L'*edge computing* traite les données localement sur les appareils plutôt que de s'appuyer sur des serveurs basés sur le cloud. Cette approche réduit considérablement la latence, assurant des temps de réponse plus rapides et améliorant la fiabilité. Elle est cruciale pour les applications nécessitant une prise de décision en temps réel, telles que la robotique, les véhicules autonomes et les appareils domestiques intelligents. En gérant les données à la périphérie, on peut améliorer les performances, réduire les latences et les coûts en termes de bande passante, et renforcer la sécurité des données.

Le problème est que le matériel embarqué a généralement beaucoup moins de puissance que les fermes de serveurs dans le cloud. Le kit Raspberry Pi AI vise à résoudre ce problème, l'intégration du Hailo-8L avec le M.2 HAT+ du Raspberry Pi illustrant cette évolution vers des solutions plus efficaces et plus réactives basées sur l'IA.

Lors d'un récent entretien avec Eben Upton, cofondateur de Raspberry Pi, ce dernier a souligné l'importance croissante de l'informatique périphérique : « À mesure que de plus en plus d'appareils deviennent intelligents, la demande de traitement local de l'IA augmente. Notre objectif avec ce nouveau kit d'IA est de rendre le calcul haute performance plus accessible aux développeurs travaillant à la périphérie. »

Raspberry Pi se met à l'IA

un nouveau kit comprenant un accélérateur IA matériel et un adaptateur M.2 HAT+

Brian Tristam Williams (Elektor)

Le kit AI de Raspberry Pi, composé de leur M.2 HAT+ avec une carte accélératrice AI Hailo-8L, apporte une performance de 13 TOPS au Raspberry Pi 5. Idéal pour l'IA en temps réel ou IA en périphérie, la robotique et la vision par ordinateur. L'intégration est relativement simple et l'offre logicielle abondante.

Que contient la boîte ?

La boîte (**figure 1**) contient un Raspberry Pi M.2 HAT+ avec une carte accélératrice Hailo-8L M.2 déjà connectée, un connecteur d'empilage de 16 mm, quatre entretoises et huit vis. Le Hailo-8L est déjà monté sur le Raspberry Pi M.2 HAT+ (**figure 2**).



Figure 2. Le kit Raspberry Pi AI comprend le Raspberry Pi M.2 HAT+ avec une carte accélératrice Hailo-8L déjà montée.



Figure 3. La carte accélératerice permet de connecter le module Hailo-8L au port PCIe via un connecteur M.2.

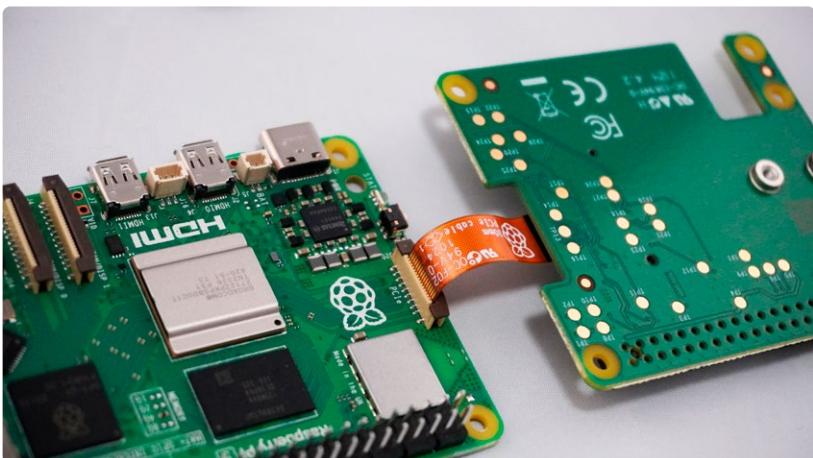
Le module Hailo-8L

Alors, qu'est-ce qui rend le Hailo-8L si spécial ? Pour commencer, cet accélérateur IA est une vraie bête de course, parfait pour les tâches d'IA en temps réel, ce qui signifie que vous pouvez entreprendre des projets en robotique, vision par ordinateur, appareils domestiques intelligents et automatisation industrielle sans être limité en puissance de calcul. La carte accélératerice Hailo au format M.2 de seulement 22 mm × 42 mm (**figure 3**) permet d'exploiter très facilement cette puissance, en tirant parti de la polyvalence du Raspberry Pi 5. Enfichez-la sur l'adaptateur M.2 HAT+, lui-même connecté au port PCIe du Raspberry Pi 5 via un câble plat de 30 mm.

L'accélérateur AI Hailo-8L est conçu pour offrir des performances locales comparables à celles dont on disposeraient dans le cloud, atteignant jusqu'à 13 TOPS (Téra-opérations par secondes). Il est idéal pour les produits d'entrée de gamme, et se distingue par son efficacité supérieure (à la fois volumétrique et énergétique), ce qui le rend très compétitif par rapport aux autres solutions de sa catégorie.

Il offre un traitement à faible latence et haute efficacité capable de gérer des pipelines complexes avec plusieurs flux en temps réel et le traitement simultané de divers modèles et tâches d'IA. Cet accélérateur est également

Figure 4. Le kit Raspberry Pi AI se connecte au port PCIe du Raspberry Pi 5.



compatible avec la suite logicielle complète de Hailo-8, ce qui permettra des mises à niveau transparentes pour des capacités encore plus élevées à l'avenir.

L'accélérateur IA Hailo-8L comprend une suite logicielle complète avec des pilotes de périphériques Hailo, HailoRT et HailoTappas, qui peuvent être facilement installés via le gestionnaire de paquets apt. Cela garantit une installation et un fonctionnement sans accroc.

L'accélération IA s'intègre parfaitement avec la pile logicielle de la caméra Raspberry Pi, y compris la prise en charge de libcamera, rpicam-apps et picamera2. Cela permet un traitement d'image avancé et des applications d'intelligence artificielle directement sur le Raspberry Pi 5.

Principales caractéristiques et avantages :

➤ **Haute efficacité :** Le Hailo-8L se concentre sur le traitement IA haute performance avec une consommation d'énergie minimale. Cela est crucial pour les appareils en périphérie, où il est nécessaire de trouver un équilibre entre performance et efficacité énergétique.

➤ **Traitement IA en temps réel :** Avec ses 13 TOPS, le Hailo-8L gère des tâches complexes comme la détection d'objets, la classification d'images et la reconnaissance vocale en temps réel. Vos applications peuvent être plus réactives et plus intelligentes.

➤ **Intégration transparente :** Le kit est pré-assemblé avec le Hailo-8L. Il suffit de le brancher sur votre Raspberry Pi 5, et vous êtes prêt à partir. Pas de tracas.

➤ **Support logiciel robuste :** Entièrement intégré dans l'environnement de Raspberry Pi OS, vous pouvez facilement installer les packages logiciels de Hailo via le gestionnaire de paquets apt. Cela inclut des pilotes de périphériques et des bibliothèques de réseaux neuronaux, de sorte que vous pouvez commencer vos projets IA immédiatement.

Cette configuration associe l'abordabilité et la polyvalence de Raspberry Pi aux capacités d'IA avancées du Hailo-8L. Le résultat ? Vous pouvez exécuter des modèles sophistiqués directement à la périphérie, réduisant la latence et la bande passante par rapport aux solutions basées sur le cloud.

Le Hailo-8L prend en charge des frameworks d'IA populaires tels que TensorFlow, TensorFlow Lite, Keras, PyTorch et ONNX, et est compatible avec les architectures hôtes ARM. Cela le rend idéal pour déployer des modèles sophistiqués sur la Raspberry Pi 5, et ouvre un tout nouveau domaine de possibilités pour le Raspberry Pi.

“

À mesure que de plus en plus d'appareils deviennent intelligents, la demande de traitement local de l'IA augmente. Notre objectif avec ce nouveau kit d'IA est de rendre le calcul haute performance plus accessible aux développeurs travaillant à la périphérie.

Eben Upton

M. Upton a expliqué la décision de collaborer avec Hailo : « Ce qui nous a enthousiasmés chez Hailo, c'est sa haute performance et son efficacité. Le Hailo-8L fournit 13 TOPS, bien plus que les solutions précédentes. Cela, combiné à ses excellents outils pour la conversion de modèles, en a fait un choix naturel pour nos ambitions d'edge AI. »

Développement et défis

Lorsqu'on lui a demandé s'il prévoyait que la HAT+ M.2 serait utilisée pour les accélérateurs d'IA en plus des disques SSD NVMe, il a répondu : « Lorsque nous avons lancé le Raspberry Pi 5, nous avons d'abord pensé que la HAT+ M.2 serait principalement utilisée pour le stockage. Cependant, nous avons rapidement réalisé qu'il y avait un intérêt significatif pour les accélérateurs d'IA, les cartes réseau et les solutions graphiques. »

Quant aux obstacles inévitables rencontrés lors de la mise en place d'un produit sophistiqué comme celui-ci, il se souvient : « Un défi auquel nous avons fait face à l'approche du lancement était l'optimisation thermique. Au départ, nous n'avions pas de pad thermique entre l'accélérateur et la carte de base, mais nous l'avons inclus dans la version finale pour améliorer la dissipation thermique. » Malgré ces obstacles, la collaboration entre Raspberry Pi et Hailo s'est avérée fructueuse, en utilisant les forces de chaque entreprise pour offrir un produit robuste et efficace.

Eben Upton a également mentionné la difficulté d'appliquer l'accélération IA en raison des architectures de modèles en rapide évolution : « L'un des défis actuels avec GenAI est qu'il est en réalité très difficile d'appliquer une accélération. Les choses doivent se stabiliser avant de pouvoir concevoir un accélérateur qui soit significativement plus performant qu'un CPU ou un GPU sans sacrifier la flexibilité. »

Cas d'utilisation et applications idéales

M. Upton a fait part de son enthousiasme quant à l'avenir de l'IA sur la plateforme Raspberry Pi : « Nous allons voir des applications étonnantes dans la robotique industrielle et les appareils domestiques intelligents. Le potentiel est énorme, et ce kit IA aidera à débloquer cela pour nos utilisateurs, » a-t-il déclaré. Il a également souligné la popularité des caméras comme accessoire pour la Raspberry

Pi : « Les caméras sont un accessoire incroyablement populaire pour Raspberry Pi, et pouvoir donner plus d'intelligence à ces applications de caméra est une grande avancée. Cet accélérateur IA est parfait pour des tâches comme la détection d'objets et la classification d'images. »

Le Hailo-8L est optimisé pour les tâches liées à la vision, ce qui le rend idéal pour des applications comme la surveillance, l'inspection de qualité automatisée dans la fabrication et l'imagerie médicale. « L'un des défis de la GenAI à l'heure actuelle est l'évolution rapide des architectures des modèles. Cependant, le Hailo-8L est conçu pour gérer efficacement des tâches plus stables liées à la vision. »

La collaboration entre Raspberry Pi et Hailo

Raspberry Pi vise à rendre le calcul haute performance accessible et abordable. Hailo, quant à lui, est spécialisé dans les processeurs d'IA qui apportent des performances dignes du cloud aux appareils à la périphérie. Ce partenariat montre ce qu'il est possible de faire lorsque l'on associe l'innovation matérielle de pointe à l'accélération de l'IA.

Il s'est également penché sur l'impact éducatif de leurs produits : « Notre objectif a toujours été de mettre du matériel programmable entre les mains des enfants et de voir ce qui se passe. Il s'agit de nourrir la curiosité et de permettre à la prochaine génération d'ingénieurs et de développeurs de s'épanouir. »

Pour ceux d'entre nous qui cherchent toujours à repousser les limites de l'IA, le nouveau kit Raspberry Pi AI change la donne. Il ouvre un tout nouveau champ de possibilités en termes de projets à réaliser avec nos Raspberry Pi. ↗

240298-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur
(brian.williams@elektor.com).



Produits

› **Raspberry Pi AI Kit**
www.elektor.fr/20879



capteurs de stations météorologiques

lequel choisir ?

Saad Imtiaz (Elektor)

Lorsque vous construisez une station météorologique, vous avez le choix entre de nombreuses options de capteurs. La station doit être précise, fiable, peu coûteuse et facile à contrôler par le microcontrôleur de votre choix. Cet article propose une comparaison approfondie de quelques capteurs environnementaux.

La sélection des capteurs est une étape critique dans la construction d'une station météorologique (personnelle), et elle influence grandement la précision et la fiabilité des données collectées. Compte tenu des nombreuses options disponibles, il est impératif de comprendre les forces et les limites des différents capteurs environnementaux. Cet article propose une comparaison détaillée des différents capteurs pouvant être utilisés dans les stations météorologiques, en mettant l'accent sur leurs performances, leur facilité d'intégration et leur fiabilité globale. En expliquant leurs aspects techniques, l'objectif est d'aider à prendre des décisions éclairées qui améliorent la précision des relevés environnementaux. La **figure 1** montre une station météorologique entièrement équipée présentée dans Elektor [1].

Critères de sélection pour la comparaison des capteurs

Avant de commencer la comparaison des capteurs environnementaux pour les stations météorologiques, il est nécessaire d'établir des lignes directrices précises pour s'y retrouver dans le large éventail de capteurs disponibles. Le champ d'application sera réduit à l'aide des critères suivants :

Qualité : l'analyse se limitera aux capteurs conçus pour des applications grand public. Ils offrent un équilibre entre performance et prix, ce qui les rend idéaux pour les amateurs et les passionnés qui cherchent à construire ou à améliorer leurs stations météorologiques personnelles.

Disponibilité : seuls les capteurs facilement disponibles sur le marché seront pris en considération. L'accessibilité est importante pour que les gens puissent utiliser les informations qu'ils



Figure 1. Projet de station météo d'Elektor [1].

obtiennent immédiatement, sans avoir à attendre qu'elles soient disponibles ou à les acheter à quelqu'un d'autre.

Solutions basées sur des modules : la comparaison portera sur les capteurs fournis sous forme de modules. Cette approche facilite l'intégration, car ils comprennent généralement le conditionnement des signaux et les circuits d'interface nécessaires, ce qui les rend adaptés à un large éventail d'applications sans nécessiter une conception électronique complexe.

Gamme de prix : la comparaison portera sur les capteurs dont le prix de vente se situe entre 2 € et 20 €. Cette fourchette de prix a été choisie pour répondre aux besoins d'un public large, y compris les amateurs et à des fins éducatives, en veillant à ce que le déploiement d'une station météorologique personnelle reste un projet abordable.

Interopérabilité et facilité d'intégration : un aspect essentiel de nos lignes directrices est la facilité d'intégration dans les systèmes existants. Les capteurs qui offrent une compatibilité directe avec les microcontrôleurs et les plateformes de développement les plus répandus, tels que Arduino et Raspberry Pi, seront privilégiés. Ainsi, les utilisateurs peuvent facilement intégrer ces capteurs dans leurs projets avec un minimum d'apprentissage supplémentaire ou de reconfiguration du système.

Capteurs

Dans une station météo typique, en particulier celles conçues pour un usage personnel ou éducatif, un ensemble de capteurs environnementaux est couramment utilisé pour surveiller les différents

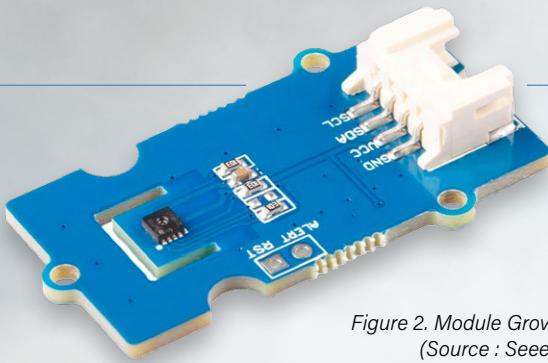


Figure 2. Module Grove SHT35.
(Source : Seeed Studio)

paramètres atmosphériques. L'accent sera mis sur les capteurs suivants, chacun jouant un rôle clé dans une station météo :

Capteurs de température et d'humidité : la température est un paramètre météorologique essentiel, tandis que les relevés d'humidité sont cruciaux pour comprendre le potentiel de précipitations et les niveaux de confort.

Capteurs de pression barométrique : les mesures de pression permettent de prévoir les changements météorologiques. Un capteur de pression barométrique offre un aperçu des tendances météorologiques, telles que l'approche de fronts ou de tempêtes, en détectant les variations de la pression atmosphérique.

Capteurs de précipitations : pour quantifier les précipitations, on utilise des capteurs de pluie. Ils mesurent généralement la quantité de pluie sur une période donnée, fournissant des données utiles pour comprendre les schémas de précipitations et la dynamique du cycle de l'eau.

Capteurs de vitesse et de direction du vent (anémomètres et girouettes) : ces capteurs mesurent la vitesse et la direction du vent, des données importantes pour les prévisions météorologiques et l'étude des régimes de vent dans un lieu donné.

Capteurs de rayonnement solaire : pour une analyse météorologique complète, il est utile de mesurer l'énergie solaire reçue à la surface de la Terre. Les capteurs de rayonnement solaire, ou pyranomètres, évaluent l'intensité de la lumière solaire, contribuant ainsi aux données sur les conditions météorologiques et le potentiel d'énergie solaire.

Capteurs d'indice UV : dans certaines stations météorologiques avancées ou spécialisées, les capteurs d'indice UV surveillent le niveau de rayonnement ultraviolet du soleil. Cette information est cruciale pour la protection de la santé et de l'environnement, car elle indique le risque de coup de soleil et l'effet du rayonnement UV sur les écosystèmes.

Ces capteurs constituent la base d'une station météorologique, fournissant une vue d'ensemble des conditions atmosphériques. Cette approche garantit que les capteurs sont à la fois abordables et faciles à intégrer dans les projets de stations météorologiques personnelles.

Température et humidité

Cette section compare plusieurs capteurs de température et d'humidité, en se concentrant sur leur précision, leurs plages opérationnelles et leur consommation d'énergie, afin de faciliter la sélection du plus approprié pour une surveillance fiable et efficace de l'environnement.

Les **SHT35** et **SHT40** de Sensirion [2] sont connus pour leur grande précision, avec des relevés de température précis à $0,1^{\circ}\text{C}$ et des relevés d'humidité précis à 1,5 % HR. La **figure 2** montre le module SHT35 de Seeed Studio. Les applications qui ont besoin

de mesures environnementales précises en bénéficieront. Cependant, cette précision a un prix plus élevé que les autres options. Le SHT40 est un choix pratique pour les projets où les contraintes de coût sont importantes, car il offre une solution plus abordable avec une précision légèrement inférieure.

Le capteur **AHT20** d'Aosong Electronic [3] fonctionne avec une tension d'alimentation de 2,2 V à 5,5 V et mesure l'humidité de 0 à 100% HR et la température de -40°C à $+85^{\circ}\text{C}$. Il offre des précisions de $\pm 2\%$ d'humidité relative et de $\pm 0,3^{\circ}\text{C}$, respectivement. Le capteur est très précis et peut répondre en 5 s pour la température et 8 s pour l'humidité. Il utilise le protocole série I²C pour la communication avec les microcontrôleurs, ce qui le rend compatible avec un large nombre.

Le capteur **SHTC3** de Sensirion [4] peut mesurer la température et l'humidité avec une précision de $0,2^{\circ}\text{C}$ et 2% HR. Il fonctionne dans une plage de température de -40°C à 125°C et une plage d'humidité de 0 à 100% d'humidité relative, avec des temps de réponse d'environ 5 s. Fonctionnant sur une plage de tension de 1,62 V à 3,6 V, il consomme un courant moyen ultra-faible de 0,5 μA . Ce capteur est compact et a une interface I²C, ce qui le rend adapté à une intégration directe. La **figure 3** montre un module SHTC3 de Soldered Electronics.

Le **HIH6130** d'Honeywell [5] offre un bon équilibre entre la précision, $\pm 0,5^{\circ}\text{C}$ pour la température et $\pm 3\%$ RH pour l'humidité, et la robustesse, avec une sortie numérique. Sa consommation électrique est modérée, autour de 450 μA en fonctionnement.

Le **MPL3115A2** de NXP Semiconductors [6] est un capteur altimétrique de précision conçu pour mesurer la pression et la température, permettant des calculs d'altitude précis. Il fonctionne dans une plage de pression de 20 kPa à 110 kPa et une plage de température de -40°C à 85°C , avec une précision de pression de $\pm 0,4$ kPa et une précision de température de $\pm 1,0^{\circ}\text{C}$. Le capteur présente une résolution en altitude de 30 cm et une résolution en température de $0,1^{\circ}\text{C}$, prenant en charge les interfaces de communication I²C et SPI. Il a une faible consommation d'énergie, seulement 40 μA en mode standard.

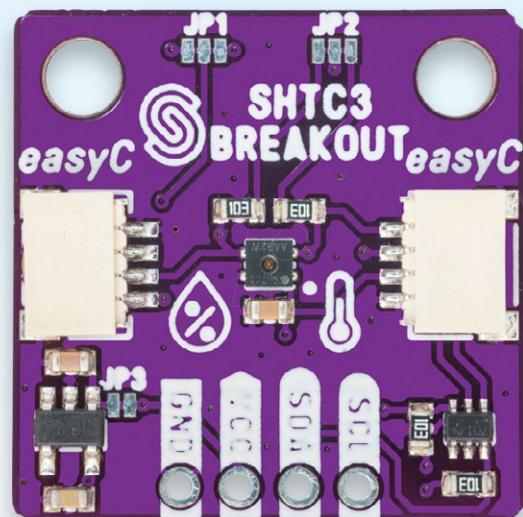


Figure 3. Module de température et d'humidité SHTC3.
(Source : Reichelt Elektronik)

Capteurs de pluie, de vitesse et de direction du vent

Il est nécessaire de mesurer la masse de la pluie dans un récipient de taille connue pour obtenir la quantité exacte de pluie qui est tombée. Pour ce faire, il existe plusieurs techniques et des solutions peuvent être mises en œuvre pour réduire les coûts. Il est, par exemple, possible d'imprimer en 3D un conteneur avec un accessoire doté d'un capteur de charge. Ce dernier mesurera la masse de l'eau, et en calculant la masse du réservoir, le volume peut être mesuré. Enfin, les quantités de pluie en ml peuvent être enregistrées. Il y a diverses solutions pour mesurer la vitesse et la direction du vent. Il est recommandé de se procurer un kit approprié, disponible dans le commerce, comme le montre la figure. Mais il est également possible de bricoler une solution pour mesurer la vitesse du vent. Pour cela, il faut imprimer en 3D les trois coupelles qui tournent dans le vent, ainsi qu'une girouette qui s'aligne sur sa direction. Pour mesurer la vitesse du vent, un encodeur magnétique conviendrait. L'aimant serait fixé à l'extrémité de l'arbre des trois coupes, et le codeur magnétique serait fixé sur le côté adjacent pour mesurer la vitesse de rotation du moulin, calculant ainsi la vitesse du vent. Pour déterminer la direction du vent, il est possible d'utiliser un capteur de boussole numérique, qui sera plus facile à mettre en œuvre qu'une méthode résistive équivalente utilisée dans la plupart des options commerciales disponibles.



Analyse comparative

L'évaluation de capteurs spécifiques sur la base d'indicateurs de performance clés permet d'identifier des choix optimaux adaptés à des besoins d'application distincts :

Précision : le Sensirion **SHT35** est le capteur le plus précis pour les mesures de température et d'humidité, idéal pour les applications où la précision est cruciale.

Rentabilité : le Sensirion SHT40 offre une grande valeur, en

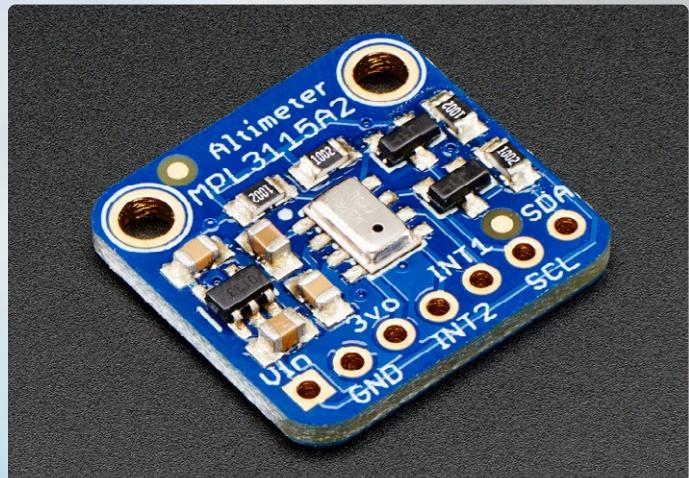


Figure 4. Module capteur Adafruit MPL3115A2. (Source : Adafruit)

fournissant une précision adéquate à un coût réduit. Il est donc particulièrement intéressant pour les projets à budget limité. Le **MPL3115A2** de NXP Semiconductors est également un choix approprié car il a un prix raisonnable et une bonne précision. La **figure 4** montre le module MPL3115A2 d'Adafruit.

Consommation d'énergie : en matière d'efficacité énergétique, le Sensirion **SHTC3** est en tête de peloton. Ne consommant que $0,5 \mu\text{A}$, il se distingue dans les applications qui requièrent une consommation d'énergie minimale, telles que les appareils fonctionnant sur batterie ou les unités de télédétection, ce qui en fait un bon choix pour les conceptions sensibles à la consommation d'énergie.

Chaque capteur a ses mérites et doit être choisi en fonction des exigences spécifiques du projet, notamment les conditions environnementales dans lesquelles il fonctionnera et le degré de précision requis. Ce choix doit également tenir compte de la facilité d'intégration dans les systèmes existants et du coût global du capteur et des périphériques associés.

Pour des comparaisons plus détaillées, il est essentiel d'examiner la fiche technique de chaque capteur. Celle-ci fournit des informations précises sur les caractéristiques électriques, la durabilité environnementale, la taille, les besoins en énergie et autres, ce qui permet de prendre une décision éclairée en fonction des besoins spécifiques du projet. Le **tableau 1** en présente une comparaison générale.

Capteurs multifonctions

Les capteurs multiparamètres tels que les séries BME280 et BME688 de Bosch Sensortec sont avantageux pour plusieurs raisons. Tout d'abord, ils offrent une solution rationalisée pour la surveillance de l'environnement, en mesurant des variables telles que la température, l'humidité, la pression et les niveaux de gaz dans un seul composant compact.

Le Bosch Sensortec **BME280** [7] est un capteur largement utilisé pour mesurer la température, l'humidité et la pression. Il atteint une précision de température de $\pm 1,0^\circ\text{C}$, une précision d'humidité

Tableau 1. Comparaison des capteurs de température et d'humidité.

Nom du capteur	Fabricant	Mesures	Précision (Temp/Hum/Pression)	Plage de fonctionnement (Temp/Hum/Pression)	Consommation d'énergie (Moyenne)	Caractéristiques supplémentaires
SHT35	Sensirion	Température, Humidité	$\pm 0,1^\circ\text{C}$ / $\pm 1,5\%$ RH	-40 à 125°C / 0 à 100% RH	1,5 µA	Haute précision, étalonnage en usine
DHT22	Aosong Electronics	Température, Humidité	$\pm 0,5^\circ\text{C}$ / $\pm 2,5\%$ RH	-40 à 80°C / 0 à 100% RH	~	Economique
HTU21D	TE Connectivity	Température, Humidité	$\pm 0,3^\circ\text{C}$ / $\pm 2\%$ RH	-40 à 125 °C / 0 à 100% RH	2,7 µA	Réponse rapide, interface I ² C
DS18B20	Maxim Integrated	Température	$\pm 0,5^\circ\text{C}$	-55 à 125°C / NA	1 µA	Modèles étanches disponibles
MPL3115A2	NXP Semiconductors	Température, Pression	$\pm 0,3^\circ\text{C}$ / $\pm 0,4$ kPa	-40 à 85°C / 20 à 110 kPa	2 µA	Fonctionnalité altimétrique
LPS22HB	STMicroelectronics	Pression	$\pm 0,1$ hPa	NA / 260 à 1260 hPa	3 µA	Capteur ultra-compact à haute résolution
MS5611	TE Connectivity	Pression	$\pm 1,5$ hPa	NA / 10 à 1200 hPa	1,2 µA	Haute résolution, faible consommation d'énergie
SHTC3	Sensirion	Température, Humidité	$\pm 0,2^\circ\text{C}$ / $\pm 2\%$ RH	-40 à 125°C / 0 à 100% RH	0,5 µA	Interface I ² C, temps de réponse de 5 s

de $\pm 3\%$ et une précision de pression de $\pm 1,0$ hPa. Fonctionnant sur une plage de température de -40°C à 85°C, d'humidité de 0 à 100% RH, et de pression de 300 hPa à 1100 hPa, il se caractérise également par une faible consommation d'énergie, ne consommant que 0,1 µA en mode veille. Ce capteur est idéal pour les stations météorologiques portables et les systèmes domotiques en raison de sa précision et de sa faible demande en énergie.

Le Bosch Sensortec **BME688** [8] s'appuie sur les capacités du BME280 en y ajoutant la détection des composés organiques volatils (COV), ce qui le rend adapté à la surveillance de la qualité de l'air intérieur. Ce capteur conserve la même précision de température et d'humidité que le BME280, mais offre une meilleure précision de pression de $\pm 0,6$ hPa. Le BME688 inclut également la détection de gaz ainsi que des capacités d'IA, ce qui nécessite un peu plus d'énergie mais conserve une efficacité adaptée aux appareils fonctionnant sur batterie. Le BME688 peut être parfait pour les environnements intérieurs, les nœuds de capteurs et les applications réseau.

La **série HPM** d'Honeywell [9] se concentre sur la détection des particules, mesurant les PM_{2,5} et les PM₁₀ avec une précision de $\pm 5 \mu\text{g}/\text{m}^3$ et $\pm 10 \mu\text{g}/\text{m}^3$, respectivement. Conçu pour la surveillance de la santé environnementale, il fonctionne dans une plage de 0 à 1 000 $\mu\text{g}/\text{m}^3$. Ce capteur utilise un ventilateur pour aspirer l'air pour l'échantillonnage, ce qui entraîne une consommation plus grande d'environ 80 mA pendant le fonctionnement, ce qui est plus élevé par rapport aux capteurs Bosch, mais justifié par son application spécifique dans l'évaluation de la qualité de l'air.

Le Sensirion **SGP40** [10] est un capteur compact qui mesure la qualité de l'air intérieur en détectant les composés organiques volatils totaux (COVT) et les niveaux équivalents de CO₂. Il offre

une précision typique de 15 ppb pour la mesure des TVOC. La consommation moyenne de courant du SGP40 est d'environ 2,6 mA, avec des pics de courant pouvant atteindre 3 mA à 3,3 V lors de la mesure.

Le Bosch Sensortec **BME680** [11] intègre les mesures de température, d'humidité, de pression et de composés organiques volatils (COV) dans un seul composant. Avec des précisions de $\pm 1,0^\circ\text{C}$ pour la température, $\pm 3\%$ RH pour l'humidité et $\pm 0,12$ hPa pour la pression, il excelle dans la précision. Il est également doté d'un capteur de gaz pour la détection des COV, ce qui améliore sa performance pour la surveillance détaillée de l'environnement. Fonctionnant dans une plage de -40°C à 85°C pour la température, de 0 à 100% RH pour l'humidité, et de 300 hPa à 1100 hPa pour la pression, le BME680 est économique en énergie, consommant moins de 0,1 mA en mode standard. La **figure 5** montre un module du BME 680 de Joy-IT.

Ces capteurs intégrés simplifient la conception et le déploiement, sont peu encombrants et réduisent la complexité des circuits. En outre, ils sont généralement plus économies en énergie et plus rentables que l'utilisation de capteurs distincts pour chaque variable environnementale. Ces capteurs dotés d'une interface unifiée, comme les connecteurs Grove, améliorent encore la facilité d'utilisation, permettant une intégration rapide dans les projets, sans câblage important ni configuration technique. C'est particulièrement bénéfique pour les applications où l'espace et l'énergie sont

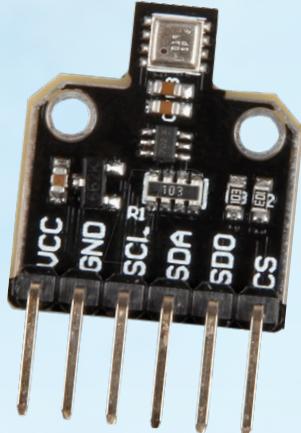


Figure 5. Module capteur Bosch BME680.
(Source : Reichelt Elektronik)

Tableau 2. Comparaison des capteurs multifonctions.

Nom du capteur	Fabricant	Mesures	Précision (Temp/Hum/Pression)	Plage de fonctionnement (Temp/Hum/Pression)	Consommation d'énergie (Moyenne)	Caractéristiques supplémentaires
BME280	Bosch Sensortec	Temp., Humidité, Pression	Temp. $\pm 1,0^{\circ}\text{C}$, Hum. $\pm 3\%$, Press. $\pm 1,0 \text{ hPa}$	Temp. -40 à 85°C , Hum. 0 à 100% RH, Press. 300 à 1100 hPa	<0,1 μA	Faible consommation, stations portables, domotique
BME688	Bosch Sensortec	Temp., Humidité, Pression, COV	Temp. $\pm 1,0^{\circ}\text{C}$, Hum. $\pm 3\%$, Press. $\pm 0,6 \text{ hPa}$	Similaire à BME280, avec détection de gaz	3,7 μA	Qualité de l'air intérieur, détection des COV, capacités IA
HPM Series	Honeywell	Matières particulières (PM2.5, PM10)	PM2,5 $\pm 5 \mu\text{g}/\text{m}^3$, PM10 $\pm 10 \mu\text{g}/\text{m}^3$	0 à 1,000 $\mu\text{g}/\text{m}^3$	~80 mA	Surveillance de la santé environnementale
SGP40	Sensirion	TCOV, niveaux de CO2 (équivalent)	TVOC $\pm 15 \text{ ppb}$	Temp. -10 à 50°C , Hum. 0 à 90% RH	2,6 mA	Qualité de l'air intérieur, conception compacte
BME680	Bosch Sensortec	Temp., Humidité, Pression, COV	Temp. $\pm 1,0^{\circ}\text{C}$, Hum. $\pm 3\%$, Press. $\pm 0,12 \text{ hPa}$	Temp. -40 à 85°C , Hum. 0 à 100% RH, Press. 300 à 1100 hPa	<0,1 mA	Surveillance détaillée de l'environnement, faible consommation

réduits, telles que les appareils portatifs et les stations compactes de surveillance de l'environnement. La comparaison de ces capteurs est présentée dans le **tableau 2**.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (saad.imtiaz@elektor.com) ou contactez Elektor (redaction@elektor.fr).

Dernières remarques

La sélection des capteurs pour une station météorologique nécessite une évaluation minutieuse des paramètres techniques, de la facilité d'intégration et de la rentabilité. Cet article a procédé à une évaluation approfondie de nombreux capteurs, en se concentrant sur ceux qui sont accessibles, peu coûteux et capables de fournir des informations fiables. Au fur et à mesure que la technologie progresse, la précision et l'utilité de ces capteurs sont susceptibles de s'améliorer, augmentant ainsi les possibilités de surveillance météorologique. Pour obtenir des informations précises et actuelles, consultez les fiches techniques les plus récentes et les communiqués de presse des fabricants. Cette stratégie permet à la station météorologique de rester un outil fiable pour la surveillance de l'environnement. ↪

VF : Maxime Valens — 240002-04



Produits associés

➤ Capteurs de vent et de pluie pour station météo (girouette, anémomètre, pluviomètre)
www.elektor.fr/20234

➤ Enviro+ Environmental Monitoring Station
www.elektor.fr/18975

➤ Bouclier météo SparkFun
www.elektor.fr/19089



LIENS

- [1] Mathias Claussen, « station météo en réseau ouvert V.2 » Elektor 5-6/2020 : <https://www.elektormagazine.fr/magazine/elektor-148/58640>
- [2] Sensirion SHT3X-DIS-F | fiche technique : <https://sensirion.com/resource/datasheet/sht3x-d>
- [3] Aosong Electronic AHT20 | fiche technique : <http://www-aosong.com/en/products-32.html>
- [4] Sensirion SHTC3 | fiche technique : <https://sensirion.com/resource/datasheet/shtc3>
- [5] Honeywell HIH6130 | fiche technique : https://eu.mouser.com/datasheet/2/187/HWSC_S_A0012940693_1-3073215.pdf
- [6] NXP MPL3115A2S | fiche technique : <https://www.nxp.com/docs/en/data-sheet/MPL3115A2S.pdf>
- [7] Bosch BME 280 | fiche technique : <https://tinyurl.com/BME-280-Datasheet>
- [8] Bosch BME 688 | fiche technique : <https://tinyurl.com/BME-688-Datasheet>
- [9] Série Honeywell HPM : <https://tinyurl.com/hpm-series>
- [10] Sensirion SGP40 | fiche technique : <https://sensirion.com/resource/datasheet/sgp40>
- [11] Bosch BME680 | fiche technique : <https://tinyurl.com/BME-680-Datasheet>

Microchip is...

IoT

Sensor Interface ◀
Microcontrollers ◀
Microprocessors ◀

Security ◀
Connectivity ◀
Cloud Services ◀



- Automotive
- Home Appliance
- Lighting
- Medical
- Smart Energy/Metering
- Wireless Audio

relevé des compteurs d'eau basé sur l'IA (1)

intégrez votre ancien compteur dans l'IdO !

Daniel Scaini (Italie)

Les compteurs d'eau intelligents sont déjà sur le marché depuis un certain temps, mais pour des raisons techniques ou administratives, le remplacement de nos anciens compteurs n'est généralement pas si simple. Ce projet permet de transformer n'importe quel compteur analogique en compteur numérique à l'aide d'un module ESP32-CAM et d'un système d'intelligence artificielle (IA). Parce que nous allons également dévoiler de nombreux éléments de référence dans cet article, nous l'avons divisé en deux parties.

L'Italie est l'un des pays européens qui prélève et consomme le plus d'eau à des fins civiles, juste derrière la Grèce. Ces chiffres sont inquiétants, surtout si l'on considère les pénuries d'eau qui frappent ponctuellement notre pays. En effet, la disponibilité moyenne a diminué d'environ 19 % au cours des trois dernières années, aggravée par des températures de plus en plus chaudes et un manque de précipitations. Ce problème n'est pas propre à l'Italie, et plusieurs pays prennent des mesures drastiques, comme la limitation de la consommation, ou le déblocage de financements pour rénover les installations de distribution privées ou publiques afin d'éviter les fuites.

De plus, compte tenu de l'évolution constante de la technologie, plusieurs fournisseurs ont pris des mesures depuis quelques années pour rendre la lecture des compteurs plus efficace chez les particuliers. Cette mesure permet d'éviter le gaspillage de la matière première, mais aussi de réduire les opérations de lecture effectuées maison par maison par les opérateurs.

Dans cet article, nous allons aborder cette thématique, qui consiste à faire un relevé d'un compteur analogique et le numériser, pour qu'il nous transmette les valeurs ou les transmette à un de nos serveurs. Le tout est basé sur une ESP32-CAM, une plateforme déjà connue de beaucoup, et un système d'IA qui peut détecter des photos et les convertir en valeurs probables.

Architecture

Les systèmes d'IA sont entrés massivement dans notre vie quotidienne, il suffit de penser aux assistants vocaux ou à la reconnaissance d'images. Pour les calculs complexes qu'un système d'IA doit effectuer, il est possible de s'appuyer sur l'informatique en nuage (*cloud computing*) sur des plateformes en ligne dédiées, ou de les effectuer directement sur la puce, dans ce que l'on appelle l'informatique en périphérie (*edge computing*). Avec le perfectionnement croissant des processeurs, ce deuxième mode de traitement se développe et constitue la base de notre projet.

Elettronica In
www.elettronicain.it

Dans celui-ci, un réseau d'IA et une ESP32-CAM coopéreront afin de pouvoir donner à l'utilisateur un résultat numérique, obtenu en photographiant numériquement un compteur d'eau analogique classique. La reconnaissance et la numérisation sont effectuées par l'ESP32-CAM à l'aide d'un réseau de neurones à convolution (CNN), dont nous parlerons plus loin dans l'article.

La première étape de la mise en œuvre de notre projet consiste à installer le micrologiciel sur notre ESP32-CAM. Ensuite, nous aurons besoin d'une phase de calibration, au cours de laquelle nous identifierons les zones dédiées à la reconnaissance des chiffres et des indicateurs de notre compteur. Une fois cette étape terminée, nous disposerons de toutes les informations nécessaires pour les envoyer numériquement là où nous le souhaitons.

Réseaux de neurones

Les réseaux de neurones sont des modèles informatiques inspirés du fonctionnement du cerveau humain. Ce sont des systèmes d'intelligence artificielle qui tentent d'émuler la façon dont le cerveau traite l'information. Les réseaux neuronaux sont utilisés dans l'apprentissage automatique, un domaine d'étude qui vise à apprendre aux ordinateurs à effectuer certaines tâches sans avoir été expressément programmés à cette fin.

Les réseaux neuronaux jouent un rôle essentiel dans les modèles d'apprentissage profond, une branche de l'apprentissage automatique qui se concentre sur le traitement de

données complexes. Ils sont composés d'unités de calcul appelées *neurones artificiels* ou *nœuds*. Ces neurones sont reliés entre eux par des connexions artificielles appelées *poids*. Chaque connexion est associée à une valeur numérique, qui représente l'importance de la connexion au modèle. Les réseaux neuronaux sont organisés en couches, avec une ou plusieurs couches qui sont cachées entre la couche d'entrée et la couche de sortie. La première couche, appelée *couche d'entrée*, reçoit les données d'entrée. Les couches intermédiaires, appelées *couches cachées*, traitent les informations par l'intermédiaire de leurs neurones. Enfin, la dernière couche, appelée *couche de sortie*, produit les résultats souhaités, comme le montre le schéma suivant à la **figure 1**.

Les réseaux neuronaux peuvent être utilisés dans un large éventail d'applications, telles que la traduction automatique, le traitement du langage naturel, le diagnostic médical, et dans ce qui nous intéresse ici, la reconnaissance d'images. Dans ce cas précis, au cours du processus d'apprentissage, des filtres sont appliqués à l'image à différentes résolutions, et la sortie de chaque image traitée est utilisée comme entrée pour la couche suivante. Les filtres commencent par des caractéristiques de base, comme la luminosité ou les bords, et deviennent de plus en plus complexes au fur et à mesure de leur évolution, en incluant des caractéristiques qui définissent l'objet de manière unique.

Les réseaux neuronaux convolutifs (CNN) sont un type de réseau neuronal spécifique, conçu pour le traitement efficace de données structurées, comme des images ou

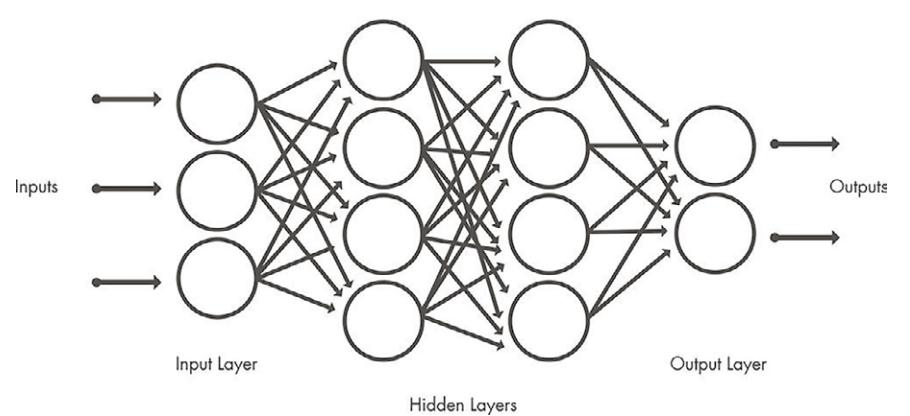


Figure 1. Les réseaux neuronaux sont composés d'une couche d'entrée, d'une couche de sortie et de nombreuses couches intermédiaires appelées couches cachées. (Source de toutes les images de cet article : Elettronica In — <https://futuronet.it>)

des vidéos. Ce qui distingue les CNN des réseaux neuronaux traditionnels, c'est l'utilisation d'une opération appelée *convolution*. Pendant la convolution, une petite fenêtre appelée *filtre* ou *noyau* glisse au-dessus de l'image d'entrée et une série d'opérations mathématiques est appliquée à chaque partie de l'image. Ce processus permet au CNN d'extraire automatiquement et efficacement les caractéristiques pertinentes des images, comme les bords, les textures ou les motifs. Ces réseaux se composent également d'une couche d'entrée, d'une couche de sortie et de nombreuses couches intermédiaires appelées couches *cachées*. Ils contiennent trois principaux types de couches, à savoir :

- La couche de convolution
- La couche de *Pooling*
- La couche complètement connectée (FC).

La couche convulsive est la première couche d'un réseau neuronal convolutif. Elle est chargée d'extraire les caractéristiques principales de l'image d'entrée. Les couches convolutives peuvent être suivies d'autres couches convolutives ou de couches de pooling. Les couches convolutives suivantes continuent de traiter les caractéristiques extraites des

couches précédentes, ce qui permet au réseau d'apprendre des fonctionnalités de plus en plus complexes et abstraites.

La couche complètement connectée, également appelée couche de sortie, est la dernière couche d'un CNN. À ce niveau, les caractéristiques extraites sont utilisées pour effectuer des prédictions ou des classifications. Ce niveau est chargé de fournir la sortie finale du réseau neuronal à convolution.

À chaque niveau, la complexité du réseau neuronal convolutif augmente au fur et à mesure que le réseau apprend des caractéristiques de plus en plus sophistiquées et abstraites de l'image d'entrée. De plus, à mesure que l'on progresse dans les niveaux de convolution, la partie de l'image identifiée et analysée en détail par le réseau neuronal augmente, comme illustré à la **figure 2**. Contrairement à un réseau neuronal traditionnel, un CNN a des poids et des biais partagés qui sont les mêmes pour tous les neurones cachés dans une couche donnée.

Après l'apprentissage des caractéristiques dans plusieurs couches, l'architecture d'un CNN passe à la classification. L'avant-dernière couche est une couche entièrement connectée qui génère un vecteur de taille K (où K est le nombre de classes

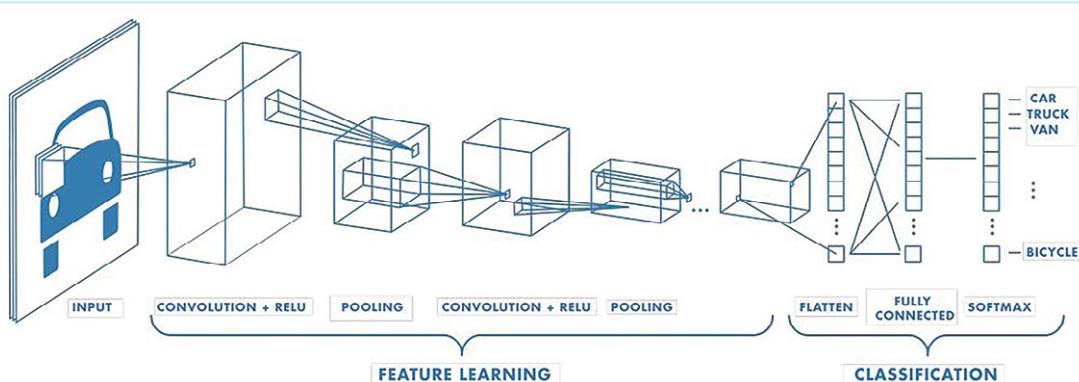


Figure 2. À chaque couche, la complexité du CNN augmente.

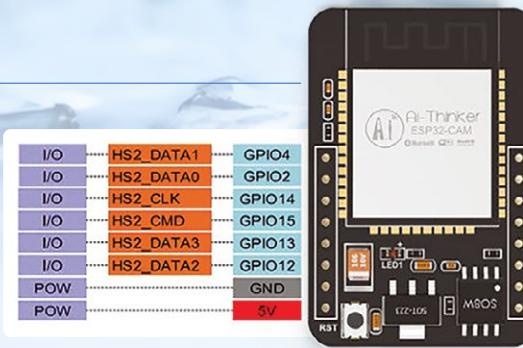


Figure 3. Brochage du module ESP32-CAM.

prévisibles) et contient les probabilités pour chaque classe de toute image classée. La dernière couche de l'architecture CNN utilise une couche de classification pour fournir la sortie de la classification finale. En général, vous disposez de modèles pré-entraînés de ces réseaux qui pèsent évidemment plusieurs Mo ou Go. Dans notre cas, il s'agira de quelques Mo, mais ils seront sauvegardés sur une carte SD afin de ne pas surcharger la mémoire interne de la puce.

Il y a plusieurs fichiers présents dans le dossier config, plus précisément avec l'extension *tflite*, ce qui signifie Tensorflow Lite. TensorFlow, né de Google Brain en 2015, est devenu une bibliothèque de référence pour la création de modèles de Deep Learning. Initialement développée en interne par Google, elle a été mise en open source et a rapidement gagné en popularité au sein de la communauté de l'apprentissage automatique. Elle offre un large éventail de modèles et d'algorithmes d'apprentissage automatique et d'apprentissage profond, connus sous le nom de réseaux neuronaux, et les met à la disposition des développeurs avec une API intuitive.

TensorFlow exploite la puissance de Python ou de JavaScript pour fournir une interface de programmation facile à utiliser pour créer des applications. Parallèlement, l'exécution de ces applications se fait en C++, ce qui permet d'obtenir des performances de calcul élevées. Cela fait de TensorFlow un choix polyvalent pour les projets d'apprentissage automatique à grande échelle.

Les modèles entraînés avec TensorFlow peuvent également être mis en œuvre sur des appareils mobiles ou de périphérie, comme dans notre cas, ainsi que sur des systèmes d'exploitation iOS ou Android. L'écosystème TensorFlow propose des outils tels que TensorFlow Lite, qui optimise les modèles TensorFlow pour qu'ils fonctionnent efficacement sur ces appareils. Avec TensorFlow Lite, il est possible de faire un compromis entre la taille du modèle et sa précision. Un modèle plus petit peut occuper moins d'espace, par exemple 12 Mo au lieu de 25 Mo ou même plus

de 100 Mo, mais il peut présenter une légère perte de précision. Cependant, cette perte de précision est souvent négligeable, compte tenu des avantages en termes de vitesse et d'efficacité énergétique qu'offre le modèle compressé. Dans notre cas ici, ce modèle entraîné avec TensorFlow sera utilisé pour distinguer et reconnaître les nombres dans le compteur et pour déterminer la direction des indicateurs. Avec l'aide de TensorFlow, nous pourrons optimiser la puissance de l'apprentissage profond pour l'analyse des données et l'IA, en améliorant l'efficacité et la précision de nos applications.

Matériel

Comme nous l'avons mentionné, le module choisi est l'ESP32-CAM d'AI-Thinker qui, grâce à sa taille compacte (40,5 × 27 × 4,5 mm), pourrait être la crème de la crème de tous les *makers*. Succédant au célèbre ESP8266, dont nous parlerons plus loin, il se compose d'un module ESP32 équipé d'un connecteur, dont le brochage est visible sur la **figure 3**, pour accueillir un module caméra séparé et un emplacement pour une carte SD d'une capacité maximale de 4 Go. Dans le détail, nous pouvons voir un microcontrôleur programmable avec Wifi et Bluetooth intégrés, et une mémoire RAM externe supplémentaire pouvant aller jusqu'à 4 Mo.

En plus de ça, le nouveau connecteur de caméra peut intégrer un module OV2640 ou OV7670, le premier est intégré au module lui-même et a une résolution de 2 mégapixels. Sa vitesse SPI est de 8 MHz et la taille de la mémoire tampon est de 384 Ko. Sa consommation normale est de 70 mA, tandis qu'en mode économie d'énergie, elle n'est que de 20 mA, ce qui la rend également intéressante pour sa fonction de faible consommation d'énergie.

Il y a également une LED à haute luminosité sur la carte (**figure 4**), qui peut servir de flash ou d'éclairage de la scène à filmer. Cette caractéristique est essentielle pour notre projet, puisqu'il sera placé dans des environnements qui sont normalement très



Figure 4. Le module ESP32-CAM est le cœur de ce projet.

sombres. Nous verrons qu'il sera possible de faire varier cette lumière pour obtenir le résultat souhaité. Bien entendu, si cela ne suffit pas, les nombreux GPIO montés sur la carte nous permettent éventuellement de monter une source lumineuse externe en cas de faible luminosité.

Pour les petites applications, cette puce se révèle très robuste et dotée d'une grande puissance de calcul, puisqu'elle s'appuie sur 2 cœurs 32 bits à 120 MHz. Cette dernière caractéristique lui permet d'avoir un taux de rafraîchissement assez élevé, bien sûr en fonction du format et de la taille. À titre indicatif, nous pouvons atteindre un débit de 8 JPEG en SVGA (800x600) par seconde.

La programmation de notre module ESP32-CAM peut se faire de plusieurs manières, la plus classique étant l'utilisation d'un adaptateur de FTDI (**figure 5**), qui grâce à l'interface FT232RL, simule les ports RS-232 et COM et permet ainsi un *plug and play* rapide du module.

Note de l'éditeur : Le module suggéré dans l'encadré « Produits associés » à la fin de cet article, entièrement compatible avec les broches et disponible dans la boutique Elektor, intègre un connecteur micro-USB avec l'interface série correspondante au chipset, ce qui rend inutile tout adaptateur série externe.

Si vous ne disposez pas de l'objet décrit ci-dessus, ne désespérez pas. En fait, il est possible de le faire à partir de n'importe quel module dans lequel les broches TX et RX sont présentes, donc à partir de différents modèles d'Arduino ou d'ESP.

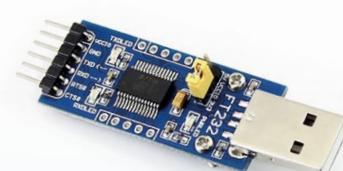


Figure 5. Le convertisseur USB/TTL avec FT232RL.

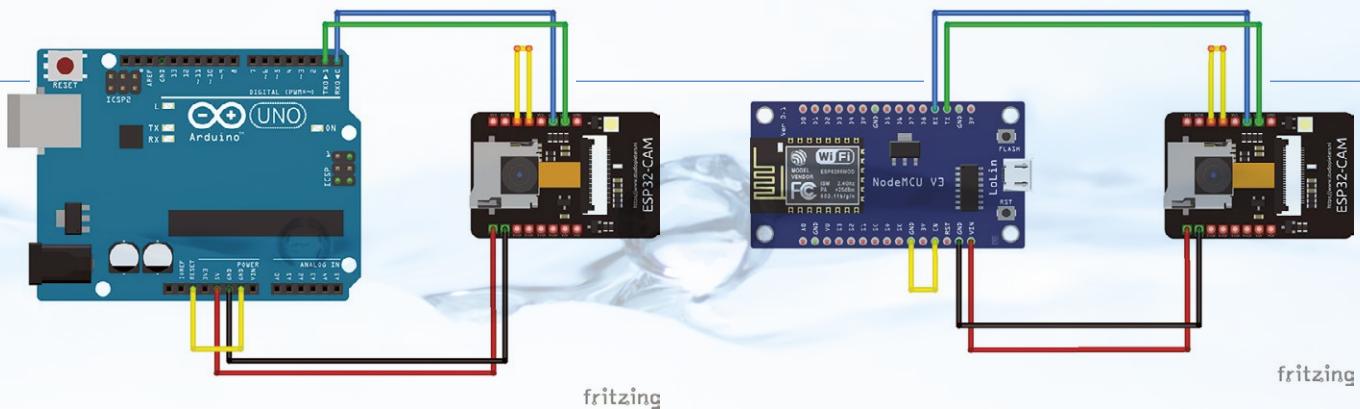


Figure 6. Le module ESP32-CAM peut être programmé avec un Arduino Uno.

Tableau 1. câblage de l'Arduino à l'ESP32-CAM.

Arduino → ESP32-CAM	Arduino	ESP32-CAM
TX → U0TXD	Reset → GND	GPIO0 → GND
RX → U0RXD		
5V → 5V		
GND → GND		

Figure 7. Câblage pour programmer le module ESP32-CAM avec un ESP8266.

Tableau 2. Câblage de l'ESP8266 à l'ESP32-CAM.

ESP8266 ⇒ ESP32-CAM	ESP8266	ESP32-CAM
TX → U0TXD	EN ⇒ GND	RX → U0RXD
5V ⇒ 5V		
GND ⇒ GND		

Câblage pour programmer le module ESP32-CAM

Pour la raison mentionnée ci-dessus, il peut y avoir différents types de câblage, similaires mais différents à certains égards. En général, le module ESP32-CAM n'a pas de port USB pour pouvoir se connecter directement au PC, et c'est précisément pour cette raison que nous devrons utiliser un module externe. Commençons par le plus classique des modules, un Arduino UNO. Dans un premier temps, nous identifions les broches d'émission et de réception de notre ESP32-CAM, appellés respectivement U0TXD et U0RXD. Ces broches devront être connectées directement aux broches TX et RX de notre Arduino. Dans le module de la caméra, ces broches correspondent également aux broches GPIO1 et GPIO3. Ensuite, nous court-circuitons les contacts GND et IO0 de l'ESP32-CAM pour assurer une communication correcte. Pour l'alimentation, nous connectons en parallèle les broches 5V et GND des deux cartes. Enfin, pour réinitialiser l'Arduino si nécessaire, nous court-circuitons la broche RESET et la broche GND. En résumé, nous obtenons le diagramme de la **figure 6**, également illustré dans le document **tableau 1**.

Si nous n'avons pas un Arduino UNO mais plutôt un ESP8266, le câblage change mais très peu. En fait, le seul changement qui mérite d'être mentionné concerne les contacts à court-circuiter sur notre module programmateur. Il ne s'agit plus de RESET et GND mais de EN et GND. Le branchement

est illustré dans la **figure 7** et indiqué dans le **tableau 2** également.

Et pour finir, le câblage le plus simple de tous. En fait, si nous possédons un module convertisseur USB/TTL basé sur la puce FTDI, tout est déjà réglé avec une connexion de moins. En effet, du côté du programmeur, nous n'aurons plus à court-circuiter quoi que ce soit, mais nous devrons échanger les contacts TX et RX l'un avec l'autre, comme le montrent la **figure 8** et le **tableau 3**.

Les connexions évoquées ne sont nécessaires que pendant la phase de programmation de l'ESP32-CAM. Une fois la programmation terminée, il suffit d'alimenter le module ESP32-CAM en utilisant uniquement les fils d'alimentation 5V et GND. Il n'est plus nécessaire de le connecter directement à l'Arduino ou d'effectuer d'autres câblages.

Dans la deuxième et dernière partie de cet article, nous verrons la procédure d'installation du micrologiciel, la configuration correcte de l'objectif de la caméra pour une mise au

point optimale, le positionnement du lecteur sur le compteur d'eau, et bien évidemment, l'ensemble du processus basé sur l'IA pour la reconnaissance et la lecture correctes des éléments du compteur. Restez à l'écoute !

VF : Laurent Rauber — 240213-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).



Produits

➤ **ESP32-Cam-CH340 Carte de développement**
www.elektor.fr/19333

➤ **FTDI Serial TTL RS232 USB Cable**
www.elektor.fr/20173

Tableau 3. Câblage du convertisseur FTDI vers l'ESP32-CAM.

FTDI	ESP32-CAM
RX	U0TXD
TX	U0RXD
5V	5V
GND	GND

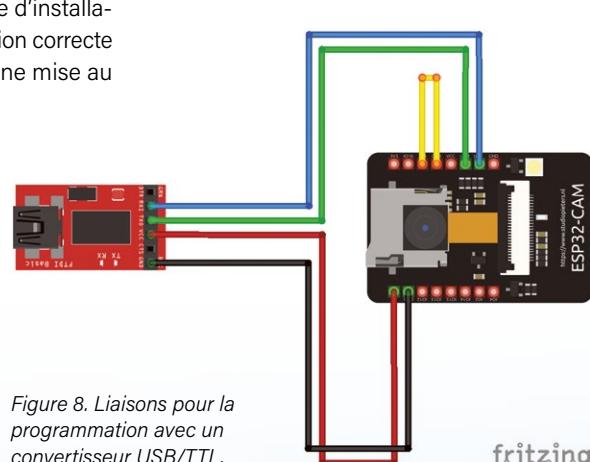


Figure 8. Liaisons pour la programmation avec un convertisseur USB/TTL.

une alarme GSM

un module GSM protège votre garage à distance

Pascal Rondane (France)

L'article décrit un système de sécurité et alarme incendie à faible consommation d'un coût modique pour un garage sans électricité. Il utilise un émetteur GSM pour les alertes et fonctionne environ deux ans sans recharger la batterie.

Ayant loué un box pour stocker mes motos et vélos, j'ai réalisé ce projet pour les protéger. Le local n'a pas l'électricité. Il est au sous-sol d'un immeuble, à quelques kilomètres de mon domicile. L'idée était de protéger ce local contre le vol et le feu, en recevant une alarme en cas d'ouverture de la porte ou de déclenchement du détecteur d'incendie. La **figure 1** montre le projet terminé. Voilà près de deux ans que mon système fonctionne sans recharger la batterie (12 V, 7 Ah). Je tiens à remercier Vincent Ruggieri pour son aide.

Conception du système

Un transmetteur GSM muni d'une carte SIM envoie les alertes. Bien sûr, il faut un abonnement, mais, en France, l'un des fournisseurs est réputé pour une offre à seulement 2 € par mois, qui convient pour ce projet. La consommation d'électricité doit être la plus basse possible, car l'installation en sous-sol rend impossible la pose d'un panneau solaire. Ne disposant pas du temps nécessaire pour concevoir et développer un matériel et un logiciel entièrement faits maison, j'ai cherché un module GSM clé en main, avec l'objectif d'en trouver un qui consomme peu d'énergie et à un prix abordable (autour de 40 €).

J'ai choisi un module GSM GL09 Plus. Différents vendeurs en proposent sur AliExpress ; voici ses possibilités et caractéristiques :

- Envoi de messages (SMS) et d'appels téléphoniques en cas d'alarme.
- Envoi de SMS périodiques pour vérifier que le transmetteur GSM fonctionne bien.
- Mesure configurable de l'état de la batterie, avec envoi d'un SMS d'alarme si la batterie faiblit.
- Tension d'entrée de 5 V à 16 V.
- Très faible consommation en veille (<4 mA).



Figure 1. Le projet terminé.

- Surveillance de 8 entrées configurables NO ou NC.
- Configuration par SMS, ne nécessitant pas de connexion PC.
- En option : version avec une sortie pour connecter une sirène.
- Antenne GSM et câble coaxial fournis.

Construction de l'alarme

Un connecteur latéral HE10-10 (double rangée de 10 contacts au pas de 2,54 mm) permet de raccorder les entrées du GL09 au moyen d'un câble plat. Ces entrées doivent être connectées à des contacts secs (sans tension) : toute source de tension les détruirait irrémédiablement. Cela m'a amené à concevoir une carte d'adaptation (**figure 2**) sur laquelle le module GL09 peut être vissé.

La carte d'interface permet d'utiliser le GL09 avec des contacts secs ou avec des capteurs externes ayant des sorties en tension (par exemple, 6 à 12 V pour un niveau logique haut, 0 V pour un niveau logique bas). Cette souplesse est obtenue en intercalant des optocoupleurs que l'on peut activer et désactiver à l'aide de cavaliers.



Figure 2. Vue rapprochée du circuit imprimé de l'adaptateur.

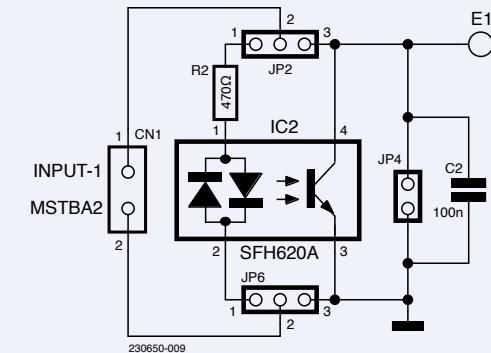


Figure 3. Une section des circuits optocoupleurs.

Les contacts ou capteurs qui déclencheront les alarmes sont connectés au circuit imprimé par des borniers enfichables. Un cavalier met le module en mode programmation (JP14) et une protection contre les tentatives de manipulations est prévue (connecteur JP9). De façon à raccorder soit un contact sec, soit un signal de tension, chaque optocoupleur se configure avec deux cavaliers installés sur des connecteurs mâles à trois broches. Le même schéma est utilisé pour chaque entrée (**figure 3**).

Pour une entrée de contact sec, placer les cavaliers entre les broches 2 et 3 au niveau de JP2 et JP6 et pour une entrée de tension entre 6 et 12 V, les placer entre les broches 1 et 2. On peut omettre l'optocoupleur et R2 si on utilise exclusivement des contacts secs. Je recommande néanmoins de souder le connecteur pour JP4 : il permet de simuler des alarmes pour les essais de configuration du module GL09. Le schéma complet (**figure 4**) montre sept blocs identiques d'optocoupleurs ainsi que deux autres cavaliers, deux condensateurs, et les connecteurs. Le circuit imprimé (**figure 5**) est simple. Tous les composants SMD sont en boîtier 1206 et tous les autres composants sont traversants. Je l'ai routé avec Eagle ; le fichier Gerber est disponible [1]. L'ensemble peut être monté dans une boîte de jonction électrique étanche.

Mise en place de l'émetteur

Le module GL09 s'accompagne d'instructions assez détaillées qui permettront d'être rapidement opérationnel. Sur déclenchement d'une

entrée d'alarme, jusqu'à six téléphones prédéfinis peuvent recevoir un SMS d'alarme. Vous pouvez personnaliser le message d'alarme (par ex., « porte de garage ouverte ») et son retour à l'état normal aussi (par ex., « porte de garage fermée »). Vous pouvez sélectionner trois modes d'alarme : appel téléphonique, SMS ou les deux. Il faut noter que lors d'un appel téléphonique, le transmetteur ne diffuse aucun message vocal et raccroche la ligne dès que vous décrochez.

Le module peut mesurer la tension d'alimentation et si elle tombe en dessous du seuil de tension réglé (11,6 V pour une batterie de 12 V), l'émetteur envoie un SMS d'alarme. J'ai configuré l'émetteur pour qu'il envoie un SMS de bon fonctionnement quotidien. L'un d'eux est illustré sur la **figure 6**. Il comprend la date et l'heure, l'état effectif des entrées, le masque des bits d'entrées « armées », ainsi que la tension de la batterie et son seuil programmé de basse tension.

Pour préserver la longévité de la batterie, sélectionnez le mode d'économie d'énergie. Selon le fabricant, dans ce mode, le module ne consomme que 30 µA environ sous 12 V.

Détection d'incendie

J'ai équipé mon garage d'un détecteur de fumée standard utilisant une pile de 9 V, tels que ceux utilisés pour la protection obligatoire des maisons dans de nombreux pays européens.

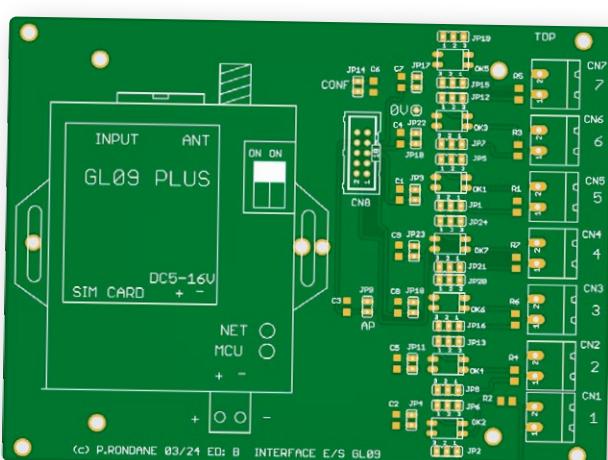


Figure 5. Vue de dessus du circuit imprimé de l'adaptateur.

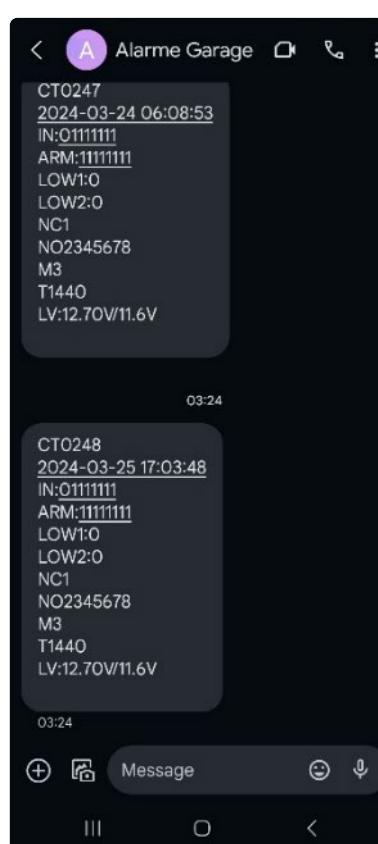


Figure 6. Exemples de messages SMS reçus. *T* est la périodicité des rapports d'état (ici T1440 signifie 1440 minutes = 24 h), *M* est le mode de fonctionnement (ici M3 signifie qu'une alarme déclenche à la fois un SMS et un appel téléphonique).

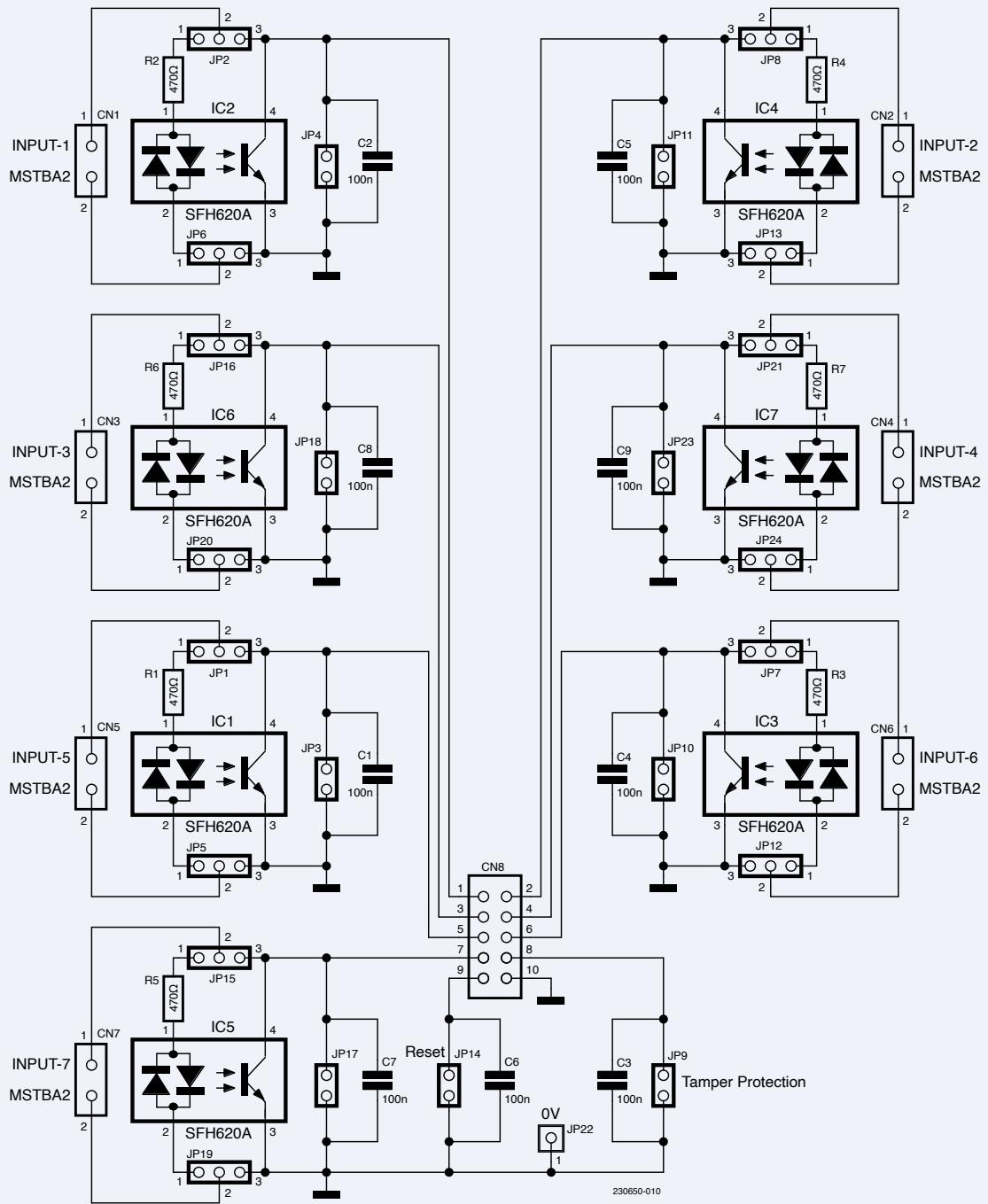


Figure 4. Schémas du projet.

Je l'ai modifié pour récupérer le signal lorsqu'un incendie est détecté. Il y a deux solutions pour cela. La première consiste à détecter si la LED intégrée au détecteur est allumée, en connectant un fil au +9 V de la batterie et un autre fil à l'une des bornes de la résistance en série de la LED. Ce signal peut être utilisé pour piloter la LED de l'un des optocoupleurs de la carte d'interface. La seconde option convient si votre détecteur d'incendie utilise (comme très souvent) le circuit intégré RE46C181 [2]. Une tension d'environ 8 V apparaît entre la broche 7

(TESTOUT) et la masse si de la fumée est détectée, et il est possible de l'utiliser pour piloter l'un des optocoupleurs décrits ci-dessus. ↗

VF : Yves Georges — 230650-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (pascal.tours@gmail.com) ou contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Pascal Rondane se passionne pour l'électronique depuis son adolescence et possède tous les numéros d'Elektor depuis le début de l'édition française en 1978. Il est électronicien de formation et a travaillé pendant 20 ans dans une entreprise de maintenance d'équipements radio Motorola et de réparation de cartes électroniques pour IBM France. Ensuite il a travaillé pendant 22 ans au service après-vente et à la conception de bancs d'essai dans un grand groupe français de signalisation routière. Retraité depuis un an, il peut s'adonner à son passe-temps favori et participer aux activités de l'*Association du Centre Historique de la Diffusion Radiophonique (ACHDR)* qui se consacre à la sauvegarde du patrimoine audiovisuel français [3].



Produits

- **OzzMaker BerryGPS-GSM pour Raspberry Pi**
www.elektor.fr/19326

- **D. Ibrahim, Projets GSM/GPRS à base de microcontrôleurs PIC et Arduino, Elektor, 2017 (E-Book, en anglais)**
www.elektor.fr/18203

- **Carte de connexion Module GPS Crowdtail-4G SIM A7670E**
www.elektor.fr/20542

— LIENS —

- [1] Téléchargements : <https://elektormagazine.fr/230650-04>
- [2] DéTECTEUR de fumée RE46C181 : <https://microchip.com/en-us/product/RE46C181>
- [3] Association ACHDR : <https://achdr.over-blog.com>

REJOIGNEZ NOTRE COMMUNAUTÉ



Abonnez-vous maintenant à
elektormagazine.fr/ezine-24

TÉLÉCHARGEZ GRATUITEMENT



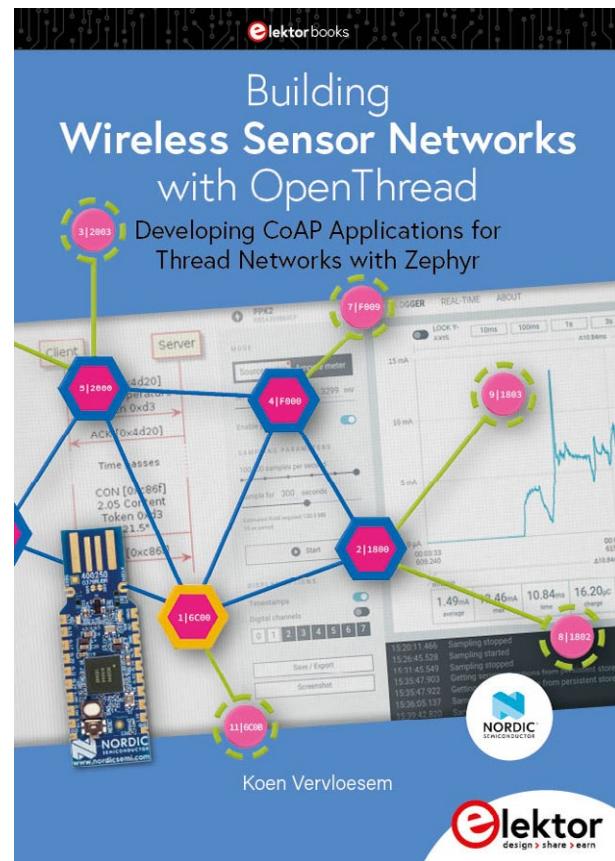
elektor
design > share > earn

optimisation et contrôle des appareils Thread à faible consommation d'énergie

faible consommation... peu d'effort ?

Koen Vervloesem (Belgique)

Le nouveau livre d'Elektor, intitulé *Building Wireless Sensor Networks with OpenThread*, ne se contente pas de présenter un ensemble riche d'informations sur la construction et la programmation de vos réseaux personnels pour lire différents capteurs. Il explore également des applications pratiques en utilisant le dongle nRF52840 de Nordic Semiconductor, qui est fourni avec le livre. Examinons un exemple pratique pour découvrir si naviguer dans le domaine des technologies à faible consommation, telles que Zephyr, CoAP, Wireshark, et Nordic est aussi simple.



Note de la rédaction : cet article est un extrait du livre d'Elektor *Building Wireless Sensor Networks with OpenThread* formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. L'auteur et l'éditeur seront heureux de répondre aux questions – pour les contacter, voir l'encadré « **Questions ou commentaires** ? ».

Thread est souvent considéré comme un « réseau maillé sans fil à faible consommation », il est donc essentiel de disposer des données précises sur la consommation d'énergie des applications Thread ! Pour combler cette lacune, cet article vous propose de :

- mesurer la consommation énergétique du micrologiciel Thread de base sur un dongle nRF52840 de Nordic Semiconductor
- réduire la consommation d'énergie en désactivant le matériel non utilisé
- réduire la consommation énergétique en transformant l'application en un appareil en veille (*Sleepy End Device*, SED).

Pour commencer, chargez l'application de Thread (vous trouverez plus de détails dans le chapitre 5 du livre) sur un dongle nRF52840 (le code est disponible sur [4]). Assurez-vous ensuite que le dongle est bien connecté à votre réseau Thread, par exemple en surveillant les paquets réseau transmis dans Wireshark. Vous êtes maintenant prêt à étudier et à optimiser la consommation d'énergie de l'application sur le dongle.

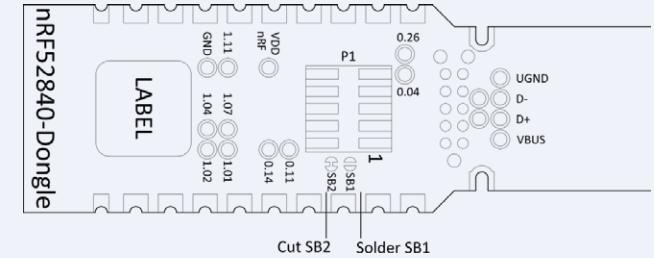


Figure 1. Couper SB2 et souder SB1 sur le dongle nRF52840 pour recevoir l'alimentation d'une source externe régulée de 1,8 à 3,6 V à travers VDD OUT. (Source de l'image : Nordic Semiconductor)

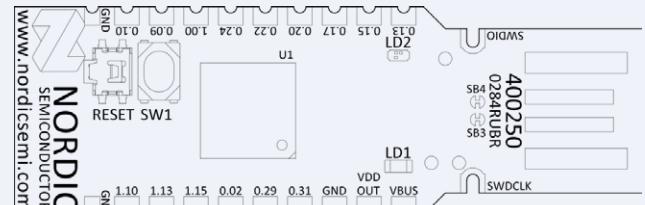


Figure 2. Brochage du Dongle nRF52840. (Source de l'image : Nordic Semiconductor)

- PPK2 VOUT à la sortie VDD du dongle ;
- PPK2 GND à GND du dongle.

Les connexions sont illustrées dans la **figure 3**. Dans cette configuration, le Power Profiler Kit II alimente le dongle nRF52840 et mesure simultanément sa consommation en courant (**figure 4**).

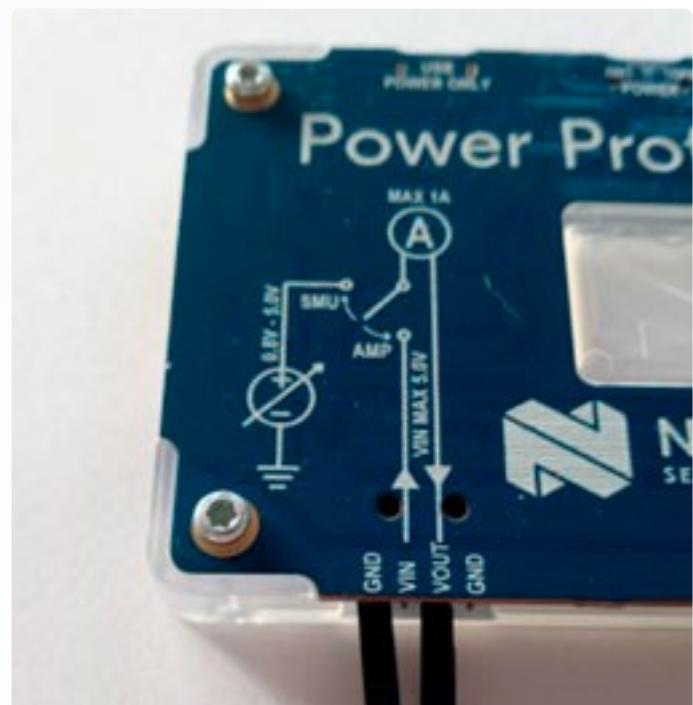


Figure 3. Connectez le Power Profiler Kit II au dongle nRF52840 en mode Source Meter.

Mesurer la consommation d'énergie avec le kit Power Profiler

Afin d'optimiser votre application Thread pour réduire la consommation d'énergie, il est essentiel de pouvoir mesurer cette consommation. Le Power Profiler Kit II (PPK2) de Nordic Semiconductor est un kit matériel intéressant idéal pour cet objectif. Ce dispositif permet de mesurer et, potentiellement de fournir des courants allant de quelques μ A à 1 A, avec des tensions inférieures à 5 V. L'appareil dont vous mesurez la consommation de courant (dans ce cas, le dongle nRF52840) est appelé l'appareil à tester (DUT).

Côté logiciel, vous utiliserez l'application Power Profiler incluse dans nRF Connect for Desktop [1], le logiciel de développement multiplateforme de Nordic Semiconductor. Téléchargez le dernier package compatible avec votre système (Windows/Linux/macOS) et exécutez-le. Note : Sous Linux, assurez-vous de rendre d'abord le fichier AppImage téléchargé exécutable, par exemple, en utilisant la commande :

```
chmod +x nrfconnect-4.2.1-x86_64.appimage
```

Ouvrez ensuite nRF Connect for Desktop et cliquez sur *Install* à côté de Power Profiler.

Pour le matériel, configurez le Dongle nRF52840 pour pouvoir l'alimenter par une source externe via sa broche VDD OUT. Pour ce faire, vous devez couper la piste du circuit imprimé qui court-circuite le pont de soudure SB2, et vous devez souder SB1, comme expliqué dans [2]. SB1 et SB2 se trouvent sur la face inférieure de la carte, comme illustré dans la **figure 1**. Attention : après cette modification, votre dongle nRF52840 nécessitera une alimentation externe en permanence, même pour flasher une nouvelle image du micrologiciel via USB.

Le PPK2 offre deux modes de mesure de la consommation d'énergie :

- **Mode Ampere Meter** : l'appareil testé est alimenté par une source externe (par exemple, USB ou une batterie). Vous pouvez utiliser ce mode pour tester la consommation d'énergie du dongle lorsqu'il fonctionne sur une batterie de 3 V.
- **Mode Source Meter** : l'appareil testé est alimenté par le PPK2.

Dans les deux modes, vous devez connaître l'emplacement des broches VDD OUT et GND du dongle, voir la **figure 2**.

En mode Ampere Meter, connectez :

- PPK2 VIN à BAT+ de la batterie ;
- PPK2 VOUT à la sortie VDD OUT du dongle ;
- PPK2 GND à GND du dongle ;
- BAT de la batterie à GND du dongle.

Dans cette configuration, la batterie alimente le dongle nRF52840, mais avec le courant circulant à travers le Power Profiler Kit II, permettant ainsi sa mesure.

En mode Source Meter, connectez :

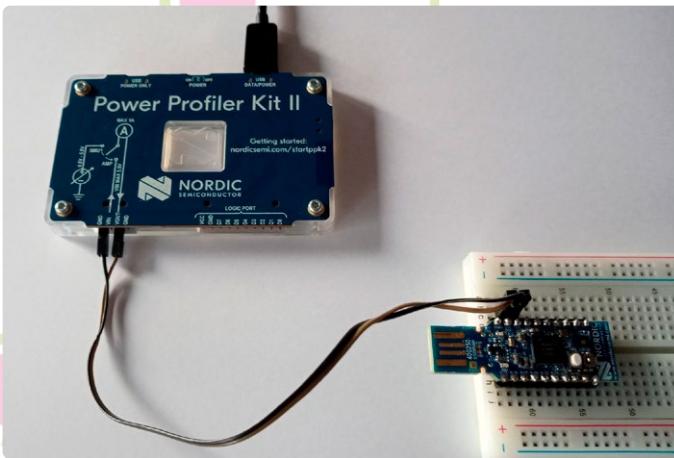


Figure 4. Le Power Profiler Kit II est prêt à mesurer la consommation de courant du Dongle nRF52840 en mode Source Meter.



Figure 5. Connectez le Power Profiler Kit II à votre PC.

Consommation d'énergie de l'application Thread

Connectez le PPK2 à votre PC via un câble USB branché sur le port USB DATA/POWER, et mettez l'interrupteur POWER situé à côté du port USB sur ON, comme illustré sur la **figure 5**. Lancez l'application Power Profiler dans nRF Connect for Desktop et cliquez sur Select Device (électionner un appareil) en haut à gauche. Choisissez PPK2. L'application pourrait vous demander de programmer le dispositif. Confirmez en cliquant sur Program .

Choisissez la mode Source meter ou Amper meter, en fonction de la façon dont le PPK2 est connecté à votre dongle. Réglez la tension d'alimentation à 3300 mV et activez l'interrupteur Enable power output (même en mode Amper meter). Dès que cet interrupteur est activé, le dongle reçoit la tension et vous pouvez voir ses paquets sur le réseau Thread dans Wireshark.

Cliquez ensuite sur START dans Power Profiler. Comme illustré dans la **figure 6**, l'onglet Data Logger affichera en continu la consommation de courant de l'appareil.

D'après les données statistiques affichées sous la courbe, la consommation moyenne de courant est de 6,29 mA. Cette valeur élevée est attendue, étant donné que vous n'avez pas encore appliqué une

optimisation de la consommation énergétique. Si vous laissez le Power Profiler fonctionner pendant un moment avec Wireshark, vous observerez un pic plus élevé à chaque fois que l'appareil Thread reçoit une annonce d'établissement de lien Mesh à l'adresse ff02::1 Link-Local All-Nodes. Si vous zoomez sur ce pic, vous verrez quelque chose qui ressemble à la **figure 7**.

Réduire la consommation d'énergie

De nombreux éléments du dongle que vous n'utilisez pas sont activés par défaut. Mais chaque composant activé consomme de l'énergie ! Par conséquent, examinons attentivement toutes les définitions de l'arborescence des périphériques dans :

```
~/zephyrproject/zephyr/boards/arm/
nrf52840dongle_nrf52840/nrf52840dongle_nrf52840.dts
```

Désactivez ceux dont vous n'avez pas besoin dans le fichier de superposition de l'arborescence du projet :

```
boards/nrf52840dongle_nrf52840.overlay:
```

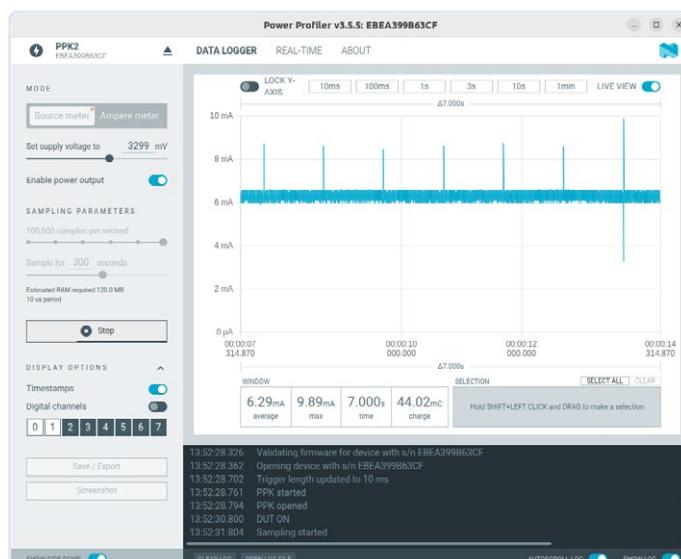


Figure 6. L'application Thread de base consomme en moyenne 6,29 mA.

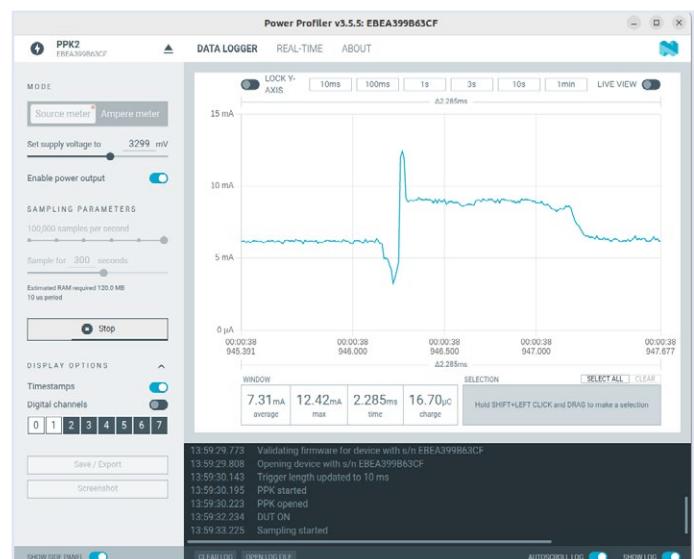


Figure 7. La réception d'une annonce MLE entraîne un pic élevé de consommation d'énergie.



Listage 1. Device tree definitions.

```
/*
 * Copyright (c) 2024 Koen Vervloesem
<koen@vervloesem.eu>
 *
 * SPDX-License-Identifier: MIT
 */

/*
 * Disabled unused hardware
 */

&adc {
    status = "disabled";
};

&uart0 {
    status = "disabled";
};

&i2c0 {
    status = "disabled";
};

&i2c1 {
    status = "disabled";
};

&pwm0 {
    status = "disabled";
};

&spi0 {
    status = "disabled";
};

&spi1 {
    status = "disabled";
};

&usbd {
    status = "disabled";
};
```



Listage 2. Kconfig configuration file (prj.conf).

```
#
# Copyright (c) 2024 Koen Vervloesem
#
# SPDX-License-Identifier: Apache-2.0
#

# Enable networking and OpenThread
CONFIG_NETWORKING=y
CONFIG_NET_IPV6_NBR_CACHE=n
CONFIG_NET_IPV6_MLD=n
CONFIG_NET_L2_OPENTHREAD=y
CONFIG_OPENTHREAD_THREAD_VERSION_1_3=y
CONFIG_OPENTHREAD_SLAAC=y
CONFIG_OPENTHREAD_PING_SENDER=y
CONFIG_OPENTHREAD_DNS_CLIENT=y
CONFIG_OPENTHREAD_MLR=y

# Kernel options
CONFIG_MAIN_STACK_SIZE=2560

# Enable power management
CONFIG_PM_DEVICE=y

# Create Sleepy End Device
CONFIG_OPENTHREAD_MTD=y
CONFIG_OPENTHREAD_MTD_SED=y
CONFIG_OPENTHREAD_POLL_PERIOD=1000

# Disable USB
CONFIG_USB_DEVICE_STACK=n
CONFIG_BOARD_SERIAL_BACKEND_CDC_ACM=n
```

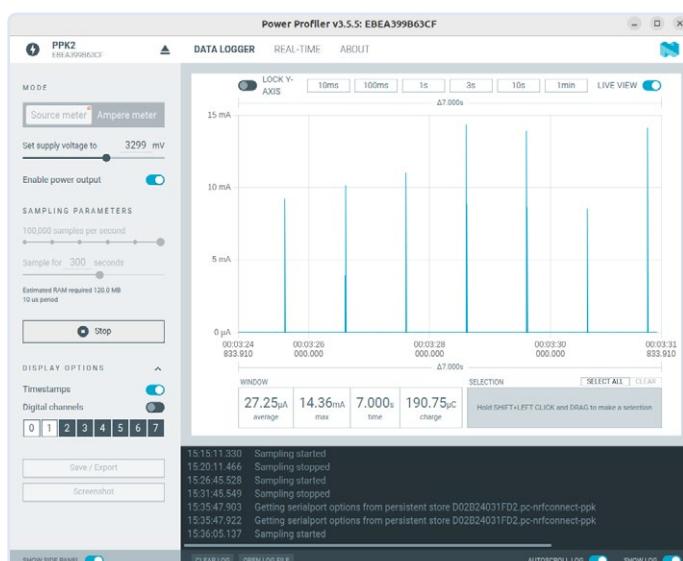


Figure 8. L'application Thread de base qui envoie des requêtes à son routeur en tant que Sleepy End Device toutes les secondes consomme en moyenne 27 µA.

Le programme permettant de le faire est présenté dans le **listage 1**. Note : L'interpréteur de commandes Zephyr ne sera plus accessible via USB si l'on désactive `uart0`.

De plus, le fichier de configuration Kconfig `prj.conf` ressemble au **listage 2**. Il permet essentiellement d'activer la gestion de l'alimentation et de désactiver l'USB. Il manque également les options pour activer la journalisation (logging) et l'interpréteur de commandes Zephyr. En outre, il configure l'appareil en tant que SED, de sorte que la radio peut rester en veille la plupart du temps, et il définit la période de polling (l'intervalle entre les requêtes du SED à son routeur) à 1000 ms.

Reconstruisez maintenant le micrologiciel :

```
$ west build -p auto -b nrf52840dongle_nrf52840
```

Créez le paquet et flashez le micrologiciel sur le dongle. Si vous mesurez la consommation de courant maintenant, la moyenne est considérablement inférieure à environ 27 µA (**figure 8**). Vous observerez clairement un pic de plus de 10 mA dans la consom-

mation de courant chaque seconde, et en même temps Wireshark montre une requête de données du SED à son routeur. Si vous envoyez une requête ping au SED, vous observerez que la réponse peut prendre jusqu'à une seconde, ce qui correspond à la période de polling de l'appareil (**figure 9**).

Plus la période de polling est longue, plus la consommation de courant est faible, mais moins l'appareil est réactif. Vous pouvez tester cela en réglant la variable de configuration `CONFIG_OPENTHREAD_POLL_PERIOD` à 5 000 millisecondes. Reconstruisez le micrologiciel et chargez-le sur l'appareil Thread. Si vous mesurez ensuite sa consommation de courant, vous trouverez une moyenne de 7,5 µA, avec des pics de courant espacés de cinq secondes, comme illustré dans la **figure 10**. L'inconvénient de cette période prolongée est que la requête ping au SED peut prendre jusqu'à cinq secondes, ce qui le rend beaucoup moins réactif.

Résumé et autres

Dans cet article, nous avons exploré la méthode de mesure de la consommation énergétique d'une application Thread fonctionnant sur un dongle nRF52840. Grâce au Power Profiler Kit II de Nordic Semiconductor et à l'application Power Profiler accompagné dans nRF Connect for Desktop, vous pouvez obtenir des informations détaillées sur la consommation d'énergie de votre application.

Comme exemple, nous avons analysé la consommation d'énergie de l'application Thread de base présentée dans ce livre. Vous avez appris comment réduire la consommation d'énergie en désactivant le matériel non nécessaire et en configurant le périphérique comme un SED. Vous avez également appris l'impact de la période de polling sur la consommation d'énergie du périphérique : plus la période de polling est longue, plus la consommation de courant est faible, mais moins le périphérique est réactif.

Cet article n'est qu'un point de départ dans cette démarche d'optimisation. Si vous souhaitez réduire au maximum la consommation d'énergie de votre application Thread, vous devez tirer parti de l'API de gestion de l'énergie de Zephyr [3]. Cela vous permet d'utiliser les fonctions d'économie d'énergie du SoC et d'autres appareils, tels que les capteurs connectés. Vous pouvez ainsi programmer votre application pour qu'elle lise les mesures d'un capteur, envoie les données sur le réseau Thread, désactive le capteur, se met en veille, se réactive, réactive le capteur et recommence tout le cycle. ↗

240225-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (koen@vervloesem.eu) ou contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Koen Vervloesem écrit depuis plus de 20 ans sur Linux, les logiciels libres, la sécurité, la domotique, l'intelligence artificielle (IA), la programmation et l'Internet des objets (IoT). Il est titulaire d'un Master en ingénierie informatique, d'un Master en philosophie et d'un certificat de sécurité LPIC-3 303. Il donne des cours de Linux et de Python aux étudiants qui aspirent à un diplôme d'associé en IdO.

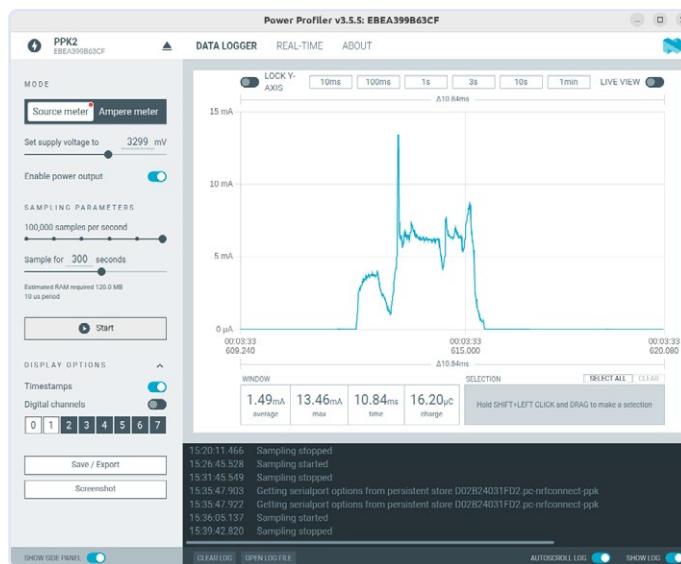


Figure 9. À chaque requête de données adressée à son routeur, l'appareil en veille subit un pic de consommation de courant.

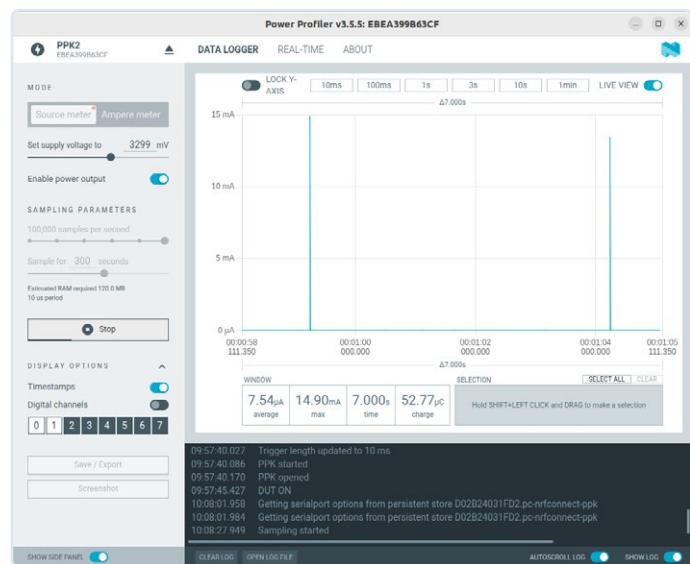
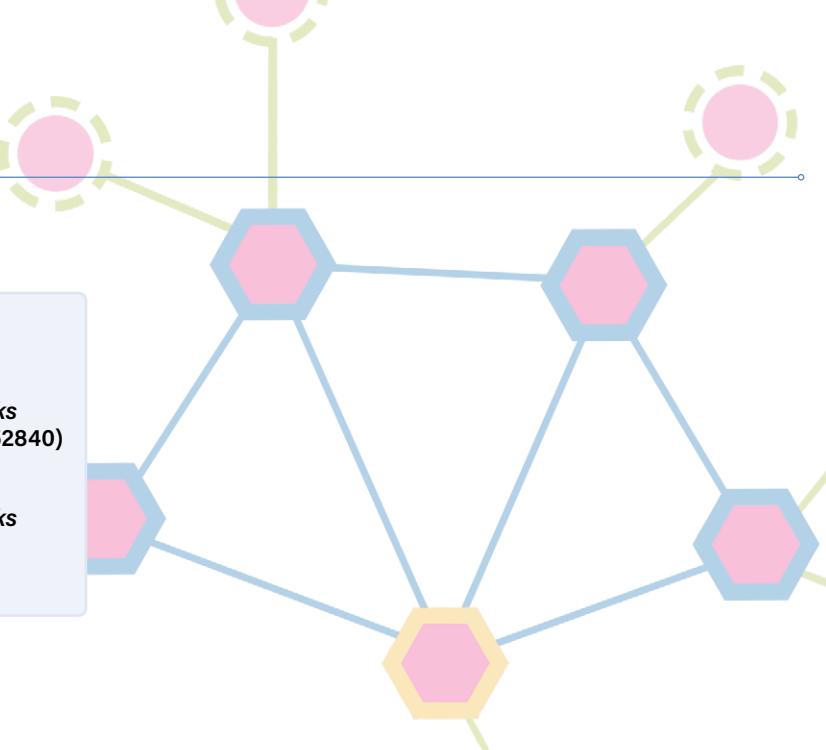


Figure 10. Avec une période de polling de 5 s, le SED ne consomme que 7,5 µA en moyenne.



Produits

- Koen Vervloesem, *Building Wireless Sensor Networks with OpenThread*, Elektor 2024 (livre et dongle nRF52840)
www.elektor.fr/20860
- Koen Vervloesem, *Building Wireless Sensor Networks with OpenThread*, Elektor 2024 (livre numérique)
www.elektor.fr/20861



LIENS

- [1] nRF Connect for Desktop : <https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-Desktop>
- [2] nRF52840 Dongle hardware mod : <https://tinyurl.com/rvj5mp47>
- [3] Zephyr's power-management API : <https://docs.zephyrproject.org/latest/services/pm/index.html>
- [4] Main.c, CMakeLists.txt, and configuration overlay for nRF52840 dongle :
<https://github.com/koenvervloesem/openthread-applications>

The screenshot shows the ElektorLabs website interface. At the top, there is a navigation bar with links like ACCUEIL, NOUVELLES, BANCS D'ESSAI, ARTICLES, MAGAZINES, LABS, INDUSTRIE, and E-CHEMIE. Below the navigation is a search bar and a 'Commencer' button. The main area displays several project cards:

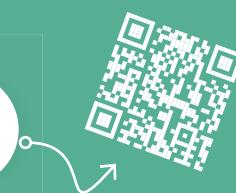
- "A 'Tube-Theremin'" by Kinkade: A project using tubes like the original and hate voltages.
- "6-digit VFD Clock with ESP32" by Andis: A project using Russian IV-22 VFD tubes and an ESP32.
- "SMALL & EFFICIENT IOR" by Dr. Gert Chesi: A project to make an LDR auto light controller.
- "LCR Meter Revisited [AU2019]" by freddy: A project using a 500 ppm LCR Meter AU2019.
- "RF Power Meter with TMHz-10GHz bandwidth and 55dB dynamic range [160193]" by breadj: A project using a RF Power Meter.
- "High-end propeller clock [120732]" by Beldiophob: A project using a propeller clock.
- "Volumetric Acoustic Amplifier with the Current Output Signal" by V2A77: A project using a Volumetric Acoustic Amplifier.

Each project card includes a thumbnail image, a brief description, a rating, and a 'Published in Elektor' badge.

Partagez vos projets dès maintenant !
www.elektormagazine.fr/e-labs

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



elektor
design > share > earn

zone D

Astuces, bonnes pratiques et autres informations pertinentes



sur le vif

montrez-moi là où ça fait mâle

Marie et Pierre Curie dans leur laboratoire.
(Source : Shutterstock/Morphart Creation)

Ilse Joostens (Belgique)

Les femmes entretiennent une relation compliquée avec la technologie, notamment en raison du manque de modèles à suivre, de préjugés sociaux et de stéréotypes sur la répartition des rôles homme-femme. Regardez attentivement des clichés pris lors de salons consacrés au marché de l'électronique, et vous y verrez une prédominance d'hommes d'âge mûr, au front dégarni, à l'attitude posée et à la bedaine tranquille. Les femmes y sont rares. Cette même disparité s'observe lors de grand-messes financières annuelles réunissant épargnants et investisseurs.

Si les lieux et événements « tendance » tels que les fablabs ou la *Maker Faire* attirent un public plus diversifié, ici aussi les hommes restent la majorité. Les compétences techniques ou financières ne dépendent pourtant pas du sexe, et quiconque, en théorie, devrait pouvoir les acquérir.

Orgueil et préjugés

Des lointaines guerrières vikings [1] [2] aux scientifiques et ingénierues d'aujourd'hui, les femmes ont toujours joué un rôle social primordial. Nous vivons pourtant dans un monde où des hommes n'ont pas hésité à s'attribuer le mérite de travaux réalisés par des

femmes, où la masculinité reste la norme, et où toutes les découvertes et révolutions technologiques majeures semblent avoir été le fait de la seule gent masculine. Ah, la vanité de ces messieurs ! Si Pierre Curie n'était pas intervenu pour que le nom de sa femme soit ajouté aux nobélisables de l'année 1903, la renommée de Marie Curie n'aurait pas dépassé la note en bas de page d'un livre d'histoire des sciences [3].

Dès la fin du 19^e siècle en astronomie, puis dans le domaine militaire durant la Seconde Guerre mondiale, plus tard encore à la NASA, les calculs étaient réalisés par des femmes. Lorsque les ordinateurs tels que l'ENIAC prirent le relais, la programmation devint naturellement un travail de femme (**fig. 1**) [4] [5]. La situation changea à la fin des années 1960 lorsque les employeurs réalisèrent que programmer était bien plus qu'une tâche administrative ordinaire. L'activité gagnant en prestige, plus d'hommes furent formés, et même discrètement favorisés durant les entretiens d'embauche. De cette



Figure 1. À l'époque où programmer était un travail de femme.



Figure 2. Un geek surpris en plein travail.
(Source : Shutterstock/Arsenii Palivoda)

période émergea le stéréotype du crack en informatique négligé et peu doué pour les interactions sociales. L'activité reste de nos jours encore une affaire masculine : environ 82 % d'hommes pour 18 % de femmes. Le cliché du geek accro aux jeux vidéos et aux pizzas, vissé à son ordinateur la nuit, est tenace (fig. 2).

En électronique, les femmes sont surtout employées sur des chaînes d'assemblage de produits ou de circuits imprimés. Elles sont heureusement de plus en plus nombreuses à occuper des postes d'ingénieries ou de techniciennes, et aussi à avoir leur propre chaîne YouTube. Le siècle précédent comptait déjà des femmes impliquées dans des travaux de haut niveau technique. Ainsi Marthe Douriau (1899–1968) [6], que la figure 3 montre portant une élégante coupe à la garçonne et un collier de perles, était-elle ingénier chez Philips et Ferrix, et aussi rédactrice attitrée de la revue *L'Antenne*. Elle a également écrit plusieurs ouvrages sur l'électricité, dont l'un, *L'Électricité et l'automobile*, fut publié sous le pseudonyme de Marc Dory par crainte de ne pas être prise au sérieux. Les compétences techniques de la femme en matière automobile restent aujourd'hui encore douteuses aux yeux de certains : l'annonce sur eBay [7] d'un testeur de batterie précise sans complexe que l'appareil est facile à utiliser,

et que les conductrices aussi sauront s'en servir...

Bouclez-la quand même

It's A Man's Man's Man's World : James Brown le chantait en 1966, et cela reste hélas vrai. Dans son livre *Femmes Invisibles, comment le manque de données sur les femmes dessine un monde fait pour les hommes*, Caroline Criado Perez explique comment une moitié de l'humanité est systématiquement ignorée en raison du biais (in)conscient « être humain = homme ». Les ceintures de sécurité ne sont par exemple pas faites pour les femmes, et durant des années n'ont été testées que sur des mannequins mâles. Celle de ma voiture me cisaille au niveau du cou et des épaules, et même si l'ajout de coussinets en a réduit les frottements, ils gênent l'accroche et l'enroulement de la ceinture, et parfois la font glisser, ce qui tourmente ma poitrine.

Le monde de la tech, surtout en Europe de l'Ouest, reste lui aussi un bastion masculin. Prenons l'exemple de Winter Circus [8], un consortium créé à l'initiative d'entrepreneurs et d'organisations de la ville de Gand pour favoriser le développement des jeunes entreprises. L'architecture du bâtiment qui l'abrite est belle et imposante, mais humainement peu chaleureuse. Devoir convaincre un jury (essentiellement masculin) de la valeur d'une



Figure 3. Marthe Douriau (en haut, au centre).
(Source : forum.retrotechnique.org)

idée – de préférence liée à l'IA – en un tel lieu est la méthode la plus sûre pour faire fuir les candidatures féminines. Tout aussi répulsifs sont les termes « esprit commercial », « résistant au stress », « passionné », ou « culture du résultat » dont sont parsemées bon nombre d'offres d'emploi pour des postes techniques. De nombreuses femmes font le choix d'une carrière ou d'une formation dans un domaine technique, mais s'arrêtent en cours de chemin. Être sous-estimée, devoir sans cesse lutter contre des préjugés ambients, ou subir un environnement de travail hostile, voire peu sûr, ne sont pas étrangers à ces abandons. Un monde plus inclusif, pas uniquement conçu pour les hommes, qui laisserait aussi place aux idées novatrices venant de femmes et conçues pour elles, nécessitera de surmonter les préjugés actuels et de rendre l'ingénierie plus attractive aux yeux féminins. ▶

VF : Hervé Moreau — 240247-04

LIENS

- [1] La guerrière viking de Birka : https://fr.wikipedia.org/wiki/Guerri%C3%A8re_viking_de_Birka
- [2] YouTube: Efin Reality — So you want to be a Warrior... : https://youtu.be/_gfo0peYu6o
- [3] When Marie Curie was almost excluded from winning the Nobel Prize, Liz Heinecke : <https://lithub.com/when-marie-curie-was-almost-excluded-from-winning-the-nobel-prize>
- [4] Femmes et tech : allons-nous revivre l'époque des « calculatrices » ? [en néerlandais] : <https://feminier.nl/magazine/rolmodellen/vrouwen-in-tech-interview-chantal-schinkels>
- [5] YouTube: Computer History Archives Project — IBM 701 Rare promo 1953 first of IBM 700 Series Mainframes : <https://www.youtube.com/watch?v=fsdLxarwmTk>
- [6] Marthe Douriau ou Marc Dory ? (forum Retrotechnique) : <https://forum.retrotechnique.org/t/marthe-douriau-ou-marc-dory/79440>
- [7] Annonce Car Battery Tester Analyzer for Automobile : https://ebay.co.uk/item/294919416437?chn=ps&_ul=GB&mkevt=1&mkcid=28
- [8] Consortium Winter Circus : <https://wintercircus.be>



chambre à brouillard à faire soi-même

visualiser les rayonnements invisibles



Matthias Rosezky (Autriche)

Avez-vous déjà imaginé voir de vos propres yeux la radioactivité et les rayonnements ionisants ? Dans cet article, je vous propose de découvrir un dispositif à réaliser vous-même, grâce à quelques composants de base disponibles sur le marché. Avec cette chambre à brouillard « maison », vous pouvez réellement voir le rayonnement ambiant (dit « rayonnement de fond »), et même visualiser la radioactivité de certains échantillons.

Qu'est-ce qui vous vient tout de suite à l'esprit lorsque vous entendez les termes « radioactivité » ou « rayonnements ionisants » ? Probablement rien de très positif, voire même un sentiment de danger, consécutif aux catastrophes de Tchernobyl ou de Fukushima. Pourtant, ce type de rayonnement existe depuis bien plus longtemps que nous ne l'utilisons et que nous ne sommes en mesure de le détecter. Ainsi, le rayonnement ambiant (aussi appelé « rayonnement de fond ») est en permanence présent autour de nous, et ce, depuis que l'humanité existe. Il n'y a là rien d'extraordinaire, même si nous n'en sommes pas toujours conscients. Concrètement, il existe deux sources de rayonnement ambiant : l'une est terrestre, et provient du sous-sol de la planète elle-même. L'autre est cosmique et correspond à tous les rayonnements à haute énergie provenant principalement de l'espace extra-atmosphérique. Le rayonnement cosmique n'est cependant pas aussi intense pour nous que la source terrestre, car il n'affecte pour l'essentiel que l'aviation, les

astronautes et les satellites. La partie terrestre du rayonnement ambiant est beaucoup plus importante sur Terre. Elle provient d'une multitude d'isotopes radioactifs naturels qui se trouvent pratiquement partout (roches, minéraux, air, eau, aliments ingérés, corps). La plupart de ces isotopes proviennent de la désintégration radioactive de l'uranium et du thorium, et il existe même, par exemple, un

isotope instable naturel du potassium. Cela vous confère donc une légère radioactivité.

Une fenêtre ouverte sur le rayonnement

Vous êtes donc sûrement en train de vous demander : « Si ce rayonnement est présent tout autour de nous, existe-t-il un moyen de le voir de nos propres yeux ? ». Il y en a évidemment un, et il a joué un rôle important dans la physique des particules au début du XX^e siècle : la chambre à brouillard. Cet appareil permet en effet non seulement de voir les rayonnements ionisants par vous-même, mais aussi de les différencier. Il existe deux types d'appareils ; celui dont il est question dans l'article présent s'appelle la « chambre à brouillard à diffusion ». Inventée en 1936 par un physicien américain, elle peut être utilisée en continu, pratiquement sans arrêt. Avec la

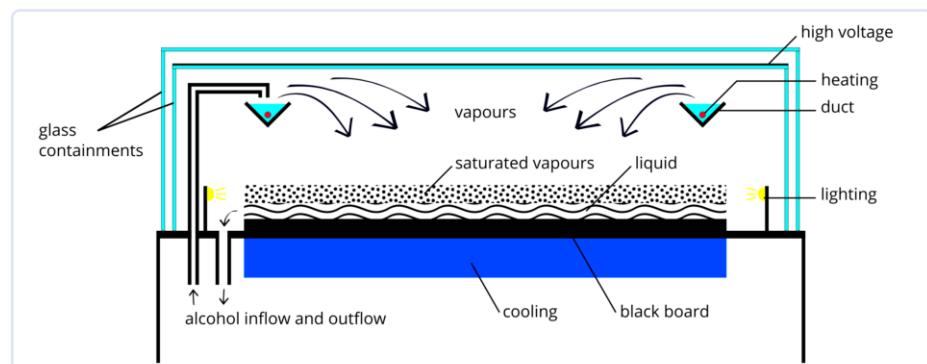
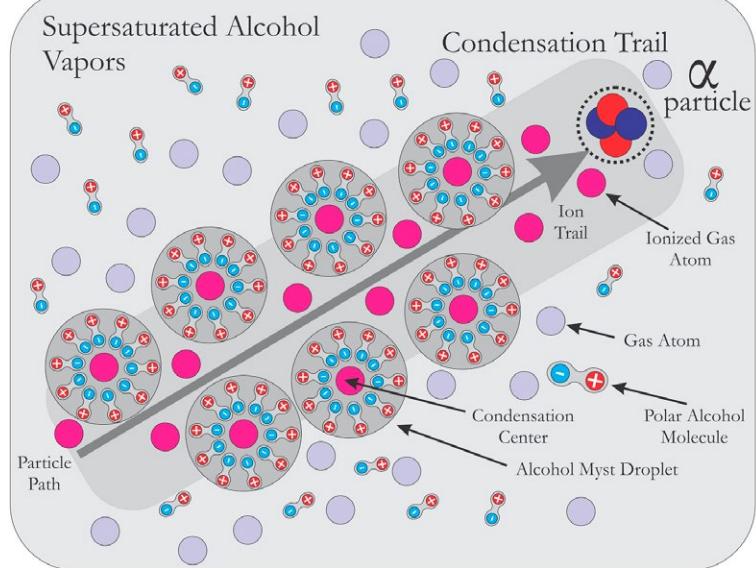


Figure 1. Schéma d'une chambre à brouillard à fonctionnement continu. (Source : Nuledo, CC BY-SA 4.0 — <https://creativecommons.org/licenses/by-sa/4.0> — via Wikimedia Commons)

How Diffusion Cloud Chamber Works



A. Stoev

Figure 2. Formation de traces de condensation d'alcool dans la chambre à brouillard à diffusion. (Source : Kotarak71, CC BY-SA 4.0 — <https://creativecommons.org/licenses/by-sa/4.0> — via Wikimedia Commons)

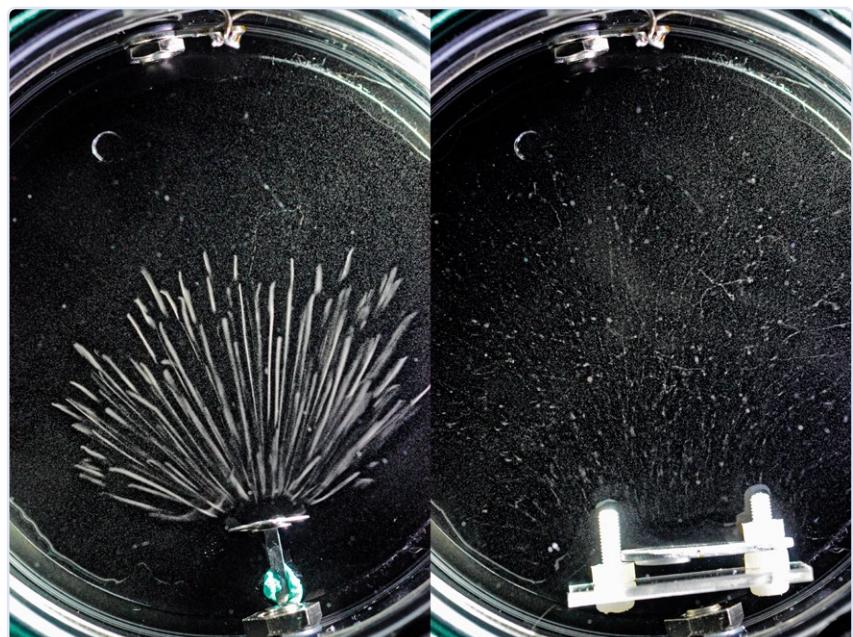


Figure 3. Exemples d'images : (à gauche) traces alpha issues d'une source Am-241. (à droite) traces bêta issues d'une source Sr-90/Y-90. (Source : Kebuk awan, CC BY-SA 4.0 — <https://creativecommons.org/licenses/by-sa/4.0> — via Wikimedia Commons)

technologie actuelle, elle est réalisable assez facilement chez vous, et c'est ce que j'ai fait.

La chambre à brouillard

Pour une chambre à brouillard à diffusion, il vous faut essentiellement deux éléments : une plaque métallique très froide formant le fond, et, au-dessus, de l'alcool, par exemple de l'alcool isopropylique (**figure 1**). Lorsque l'alcool s'évapore, il sature l'air qui, en se refroidissant, redescend également. Lorsque ce mélange d'air saturé s'approche de la plaque froide, l'air se condense sur la plaque froide pour former à nouveau un liquide. Cependant, juste au-dessus de la plaque, se trouve une petite section d'air où ce mélange air-alcool est sursaturé et sur le point de se condenser, n'attendant qu'un événement pour le perturber. C'est là que les rayonnements ionisants entrent en jeu. Comme leur nom l'indique, ces rayonnements ionisent les molécules d'air lorsqu'ils traversent cette couche, et les ions laissés sur place constituent des noyaux de condensation pour le mélange (**figure 2**). La condensation se produit donc le long de la ligne d'ionisation et forme un léger brouillard, révélant ainsi la trajectoire de la particule ionisante. Selon le type de particule qui a traversé la chambre à brouillard, apparaissent différents types de traces. La plupart d'entre elles sont des particules alpha ou bêta issues de désintégrations respectives. Les particules alpha, beaucoup plus lourdes, laissent des traces courtes et épaisses, d'apparence très rectiligne. Pour leur part, les particules bêta laissent des traces beaucoup plus longues, plus fines et parfois très irrégulières dans la chambre à brouillard (**figure 3**). Vous pourrez également placer vous-même de petits échantillons radioactifs dans l'appareil, et vous pourrez éventuellement voir, dans ce cas, d'autres types de radioactivité.

Construire vous-même votre propre appareil

Alors, comment ai-je procédé pour réaliser de toutes pièces ma propre chambre à brouillard ? Comme je l'ai mentionné, l'idée de base de cette chambre à brouillard repose sur une plaque métallique très froide et un peu de vapeur d'alcool. Commençons par la plaque métallique. Pour obtenir une distribution optimale de la chaleur sur toute la surface, j'ai choisi une grande plaque de cuivre de 10x10 cm, d'une épaisseur de 4 mm (**figure 4**). Le cuivre est un excellent conducteur thermique, de sorte que la température est très homogène sur l'ensemble de la plaque. Cela me permet de refroidir complètement la



Figure 4. Plaque de cuivre au-dessus du refroidisseur de CPU.

zone de 10×10 cm jusqu'à environ -30 °C et de l'utiliser dans son intégralité pour observer les traces, ce qui est utile lorsque l'on insère certains échantillons dans la chambre à brouillard. Si vous le souhaitez, vous pouvez également utiliser de l'aluminium, qui est beaucoup moins cher, mais qui ne sera probablement pas aussi performant. Pour refroidir la plaque, j'utilise un refroidisseur thermoélectrique, également connu sous le nom de refroidisseur Peltier, qui est essentiellement une petite pompe à chaleur sans aucune pièce mobile. Lorsqu'on lui applique une tension, un côté de la cellule Peltier refroidit tandis que l'autre se réchauffe, ce qui maintient un gradient de température constant entre les deux faces. Le côté froid est directement relié à la plaque de cuivre à l'aide de pâte thermique. Pour sa part, le côté chaud doit être refroidi de manière optimale pour ne pas trop se réchauffer. Ce processus est nécessaire parce que le gradient est constant. La température de la face froide dépend donc de la température de la face chaude. J'utilise un puissant refroidisseur de CPU pour tour double, normalement utilisé dans les PC, avec deux ventilateurs. Avec un refroidissement classique par air ou par liquide, vous ne pourrez jamais descendre au-dessous de la température ambiante de la face chaude. Le gradient de température requis sera donc d'au moins 50 °C pour atteindre -30 °C si vous utilisez le refroidisseur thermoélectrique (et plus tard, la chambre à brouillard) à température ambiante. Pour y parvenir, vous aurez besoin de plusieurs refroidisseurs Peltier en série. Vous pouvez cependant aussi utiliser directement un modèle déjà cascadé, comme le TEC2-25408. Il s'agit d'un refroidisseur composé de deux modules simples superposés en interne de sorte que vous pouvez l'utiliser exactement comme une unité unique (plus épaisse). Il est capable d'atteindre un gradient de température beaucoup plus élevé atteignant 80 °C (certes, dans les circonstances les plus idéales). Avec ce module unique, un gradient de 50 °C à 60 °C est facilement possible si votre plaque de cuivre n'est pas trop grande.

Surmonter les défis de l'assemblage

Pour monter solidement la plaque de cuivre, le refroidisseur thermoélectrique et le refroidisseur à air, je voulais un dispositif qui puisse être vissé et dévissé au cas où j'aurais besoin de changer un élément. Je voulais aussi pouvoir appliquer une bonne pression de montage sur le refroidisseur thermoélectrique avec toute la pâte thermique. Ainsi, j'ai utilisé les supports

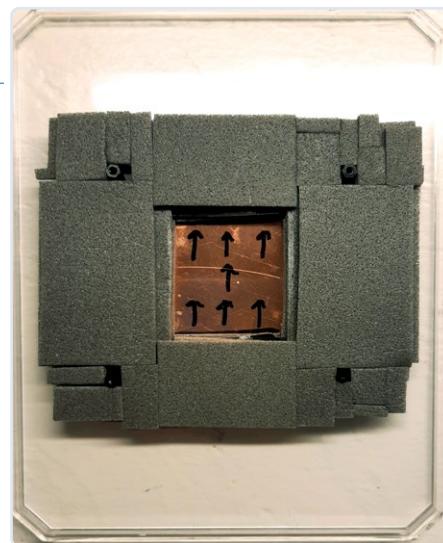


Figure 5. Plaque de cuivre avec isolation thermique et fixations en matière plastique.



Figure 6. Plaque de cuivre noire entourée d'une feuille d'acrylique. J'ai recouvert l'acrylique d'un peu plus d'isolant pour assurer l'étanchéité de la cloche transparente.

de montage Intel fournis avec le refroidisseur de CPU, j'ai percé les trous de fixation dans la plaque de cuivre et j'ai utilisé des vis en matière plastique et des entretoises pour solidariser tous les éléments. Les vis en matière plastique apportent un compromis évident, même si elles sont loin d'être aussi solides et fiables que les vis métalliques normales. Cependant, utiliser des vis en métal à proximité de la plaque de cuivre froide et du refroidisseur de CPU chaud est une mauvaise idée, ne serait-ce qu'en raison de l'énorme transfert de chaleur qui se produirait entre les faces chaude et froide. Le plastique, en revanche, a une très faible conductivité thermique, ce qui est essentiel dans ce cas. Pour réduire davantage la charge thermique, j'ai également ajouté une isolation thermique en bas de la plaque et autour de l'élément Peltier, comme le montre la **figure 5**. De même, pour minimiser les perturbations de la couche sensible nébulisée (brouillard) et veiller à ce que le mélange d'air se sature ensuite d'alcool, une cloche transparente est installée autour de la plaque de cuivre, ce qui nécessite un certain espace. À cette fin, j'ai utilisé une simple feuille d'acrylique plus grande que le diamètre de la cloche et je l'ai fixée à la plaque de cuivre par le bas à l'aide de fixations en plastique (**figure 6**). Cela permet également de disposer d'un espace pour installer des diodes électroluminescentes de bonne puissance, destinées à éclairer tangentielle-ment la plaque de cuivre et illuminer la fine nappe de brouillard. Cette lumière contribue à rendre les traces clairement visibles sur la plaque de cuivre en augmentant le contraste. Pour améliorer encore la visibilité, j'ai également recouvert de ruban noir isolant électrique la surface supérieure exposée de ma plaque de cuivre.

Questions d'alimentation, et autres

Pour alimenter ce dispositif, j'ai utilisé un bloc d'alimentation ATX standard, récupéré d'un ancien PC. Ces alimentations peuvent délivrer jusqu'à 12 V, injectés directement dans le refroidisseur thermoélectrique. Il est possible d'affecter directement un certain nombre de lignes 5 V aux ventilateurs, sans contrôleur, et de les faire tourner silencieusement tout en appliquant un flux d'air au dissipateur thermique géant. Dans mon cas, j'ai ajouté un interrupteur entre le 5 V et le 12 V pour changer la vitesse du ventilateur. Les LED que j'ai utilisées sont des modèles RVB adressables en 5 V, uniquement pour expérimenter différentes couleurs et luminosités. Pour cette raison, je les ai également connectées à l'alimentation et j'ai utilisé un Raspberry Pi Pico qui sert à configurer les valeurs de luminosité et les nuances de couleur. J'ai fini par utiliser un blanc légèrement froid avec la luminosité maximale, ce qui est idéal pour observer des traces complexes. Vous avez sans doute remarqué que je n'ai pas encore parlé de l'alcool. Il doit pouvoir passer au-dessus de la plaque froide sans goutter.

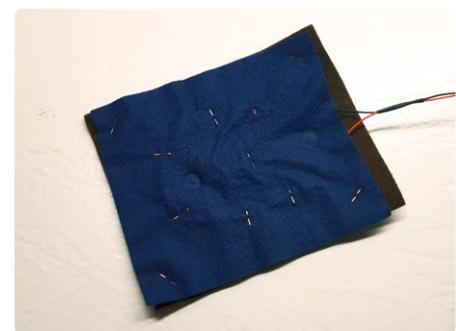


Figure 7. Morceaux de feutre agrafés ensemble avec le dispositif chauffant résistif.

Pour ce faire, j'ai pris quelques morceaux de feutre (**figure 7**) et je les ai montés à l'intérieur à l'aide d'aimants, en haut de la cloche, comme le montre l'image en début d'article. Ces aimants assurent un maintien suffisamment solide pour ne pas compliquer le montage, et avec l'avantage supplémentaire de permettre le démontage des éléments. J'ai apporté une dernière amélioration, cependant, avant de tout mettre en place en installant un petit dispositif chauffant pour l'alcool. En effet, pour obtenir une nappe de brouillard encore plus épaisse au-dessus de la plaque froide, on peut soit diminuer sa température (ce qui nécessite une tension plus élevée ou plusieurs refroidisseurs Peltier), soit augmenter la température de l'alcool pour accroître l'évaporation. J'ai donc simplement ajouté quelques résistances aux morceaux de feutre, je les ai agrafées solidement, puis solidement connectées au rail 12 V de mon alimentation. **Pour des raisons de sécurité, vous devez être absolument sûr(e) que rien ne peut être court-circuité à proximité de l'alcool, car il est hautement inflammable !** Dans mon cas, les résistances ont une valeur assez élevée, ce qui ne permet de dépasser la température ambiante que d'environ 5 à 10 °C, mais, même avec cela, on peut déjà constater une nette amélioration.

Mise en service et résultats

Pour démarrer la chambre à brouillard, je mets un peu d'alcool isopropylique sur le feutre de manière à bien le saturer, mais sans qu'il goutte. Je place ensuite la cloche transparente autour de la plaque supérieure et je démarre le dispositif. Il faut quelques minutes pour que la chambre refroidisse complètement jusqu'à environ -30 °C. Une petite flaque d'alcool se forme au-dessus de la plaque et empêche la formation de glace sur la face froide. Ensuite, une couche de brouillard de plus en plus épaisse commence à se former au-dessus de la plaque et, peu de temps après, des traces commencent à se former dans tout cet espace. Les résultats sont visibles dans cette vidéo [1], où je place également un petit échantillon légèrement radioactif dans la chambre à brouillard. Par ailleurs, le projet est disponible sur la page du projet Elektor Labs [2], ainsi que sur Hackaday [3].

Et voilà ! Ce n'est pas très compliqué et ce dispositif fonctionne très bien pour le budget alloué. Le fait de pouvoir construire une chambre à brouillard aussi performante en « bricolant » un petit nombre de composants standard ne cesse de m'étonner. N'hésitez pas à me faire savoir ce que vous en pensez ! 

VF : Pascal Godart — 230495-04

À propos de l'auteur

Étudiant diplômé, Matthias Rosezky vit à Vienne, en Autriche. Spécialisé dans la physique des rayonnements, c'est un maker et un bricoleur dans l'âme. Il aime programmer et travailler avec l'électronique pour créer des choses intéressantes et améliorer les réalisations existantes. Dans certains de ses autres projets [4], Matthias a repris cette philosophie et a construit du matériel de détection de rayonnement open source abordable.

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr)



Produits

➤ Kit de compteur Geiger MightyOhm (avec boîtier)
www.elektor.fr/18509

➤ Testeur de rayonnement électromagnétique WT3122
www.elektor.fr/20521



LIENS

- [1] Introduction de thorianite dans ma chambre à brouillard « maison » : <https://youtu.be/BRjhpcfuWA>
- [2] Page Elektor Labs de ce projet : <https://elektormagazine.fr/labs/peltier-cloud-chamber>
- [3] Le projet sur Hackaday : <https://hackaday.io/project/192146-peltier-cloud-chamber>
- [4] Autres projets de cet auteur : <https://nuclearphoenix.xyz>

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 / worldwide coverage
- Multi-band support

#GOCELLULAR

SparkFun Thing Plus Matter

carte de développement IdO polyvalente basée sur Matter

Saad Imtiaz (Elektor)

Le SparkFun Thing Plus Matter MGM240P marque son entrée impressionnante dans le domaine de l'IdO et des maisons intelligentes, offrant des fonctionnalités avancées et robustes destinées aux développeurs et aux passionnés.

Matter [1], autrefois connu sous le nom de "Project CHIP" (Connected Home over IP), est un protocole développé pour permettre l'interopérabilité entre les appareils domestiques intelligents et les appareils IdO, ce qui fait de la carte SparkFun Thing Plus Matter [2] un complément intéressant dans le paysage du développement IdO.

La carte (**figure 1**) présente un design compact, (5.84 x 22.9 cm) compatible avec le facteur de forme "Thing Plus" de Sparkfun. Vous pouvez écrire vos programmes pour le MGM240P en utilisant l'outil de débogage Simplicity Studio, qui est disponible pour Windows, Mac OSX et Ubuntu. Le manuel de mise en œuvre [3] fournit des informations détaillées sur le matériel ainsi qu'un guide pas à pas pour vous aider à démarrer avec l'EDI Simplicity Studio.

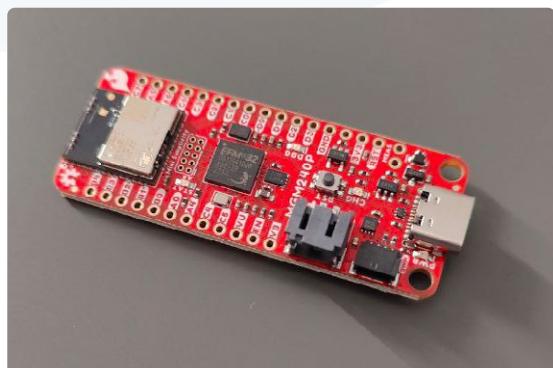


Figure 1. SparkFun Thing Plus Matter MGM240P.

Voici les caractéristiques de la carte SparkFun Thing Plus Matter MGM240P:

- Module sans fil MGM240P [4]
- Protocoles sans fil — 802.15.4 (Zigbee et Open Thread) et Bluetooth Low Energy 5.3 ; Matter-ready
- SoC EFR32MG24 de Silicon Labs [5]
- Mémoire — 1,536 Ko de mémoire flash et 256 Ko de RAM
- Stockage — socle microSD
- Deux rangées de connecteurs avec 21× broches GPIO
- Connecteur JST Qwiic à 4 broches
- Microcontrôleur : EFM32GG12B410F1024GL120-A utilisé comme programmeur J-Link et débogueur
- Connecteur : Mini Simplicity de Silicon Labs non équipé, pour connecter un débogueur externe.
- Connecteur : JST 2 broches pour une batterie LiPo battery (non incluse) avec chargeur LiPo MC73831 et état de charge LiPo MAX17048
- Consommation d'énergie : 15 µA lorsque le MGM240P est en mode basse consommation.

Prise en charge du protocole Matter

La carte est conçue pour prendre en charge le protocole Matter, ce qui constitue un avantage significatif pour ceux qui cherchent à réaliser des appareils domestiques intelligents et des appareils IdO qui communiquent de manière transparente avec d'autres appareils compatibles avec Matter. La capacité du protocole à unifier différents écosystèmes IdO change la donne dans le développement IdO.

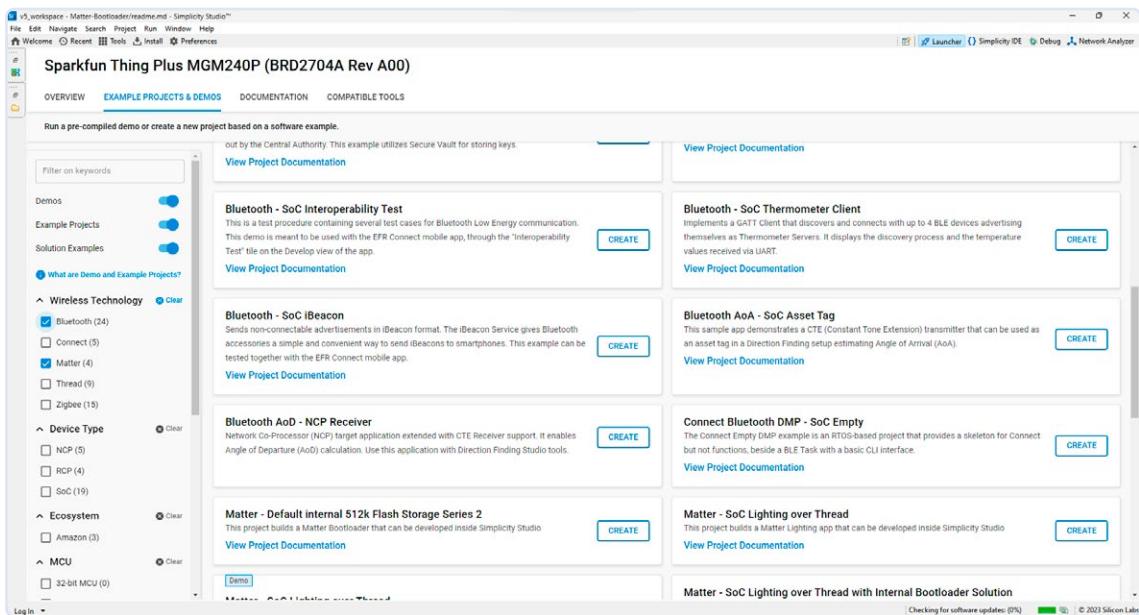


Figure 2. Ce module prend en charge différents protocoles sans fil tels que Bluetooth Low Energy (BLE) et IEEE 802.15.4 (la base de Thread et Matter), dont j'ai testé une sélection.

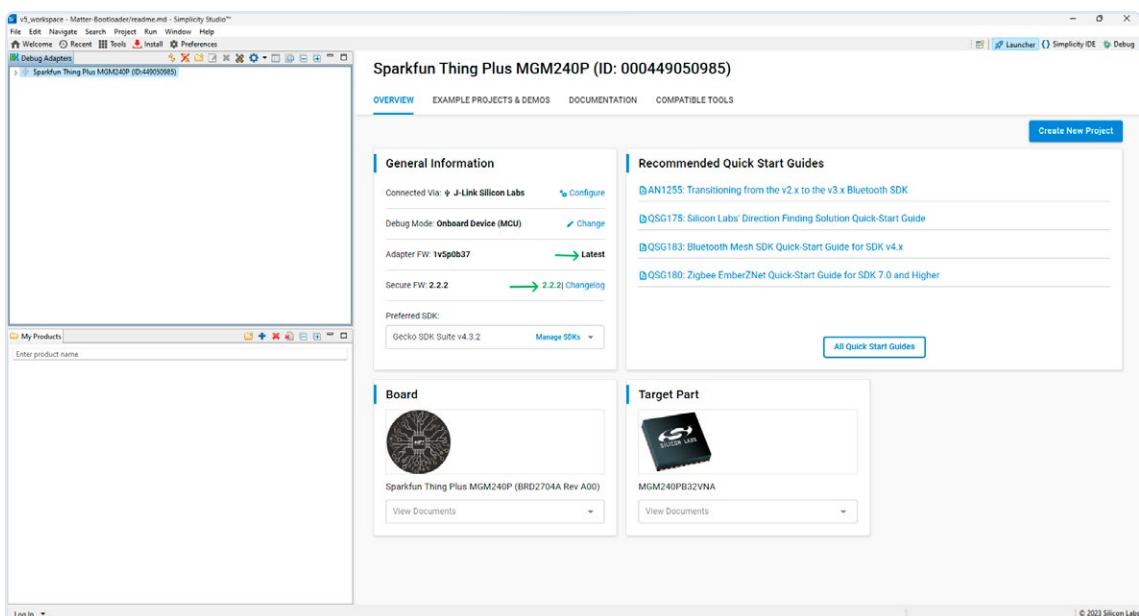


Figure 3. Le chargement du code sur cette carte est très simple : installez Simplicity Studio, branchez votre carte et elle sera automatiquement détectée.

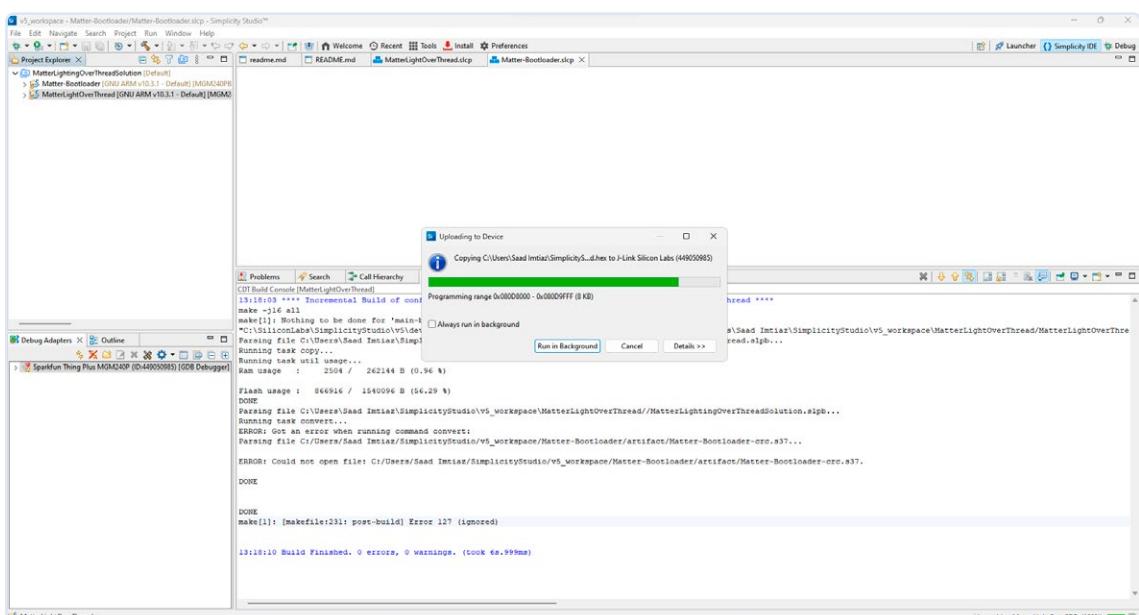


Figure 4. J'ai choisi l'exemple Matter Light Over Thread, j'ai compilé le code, puis je l'ai chargé sur la carte.

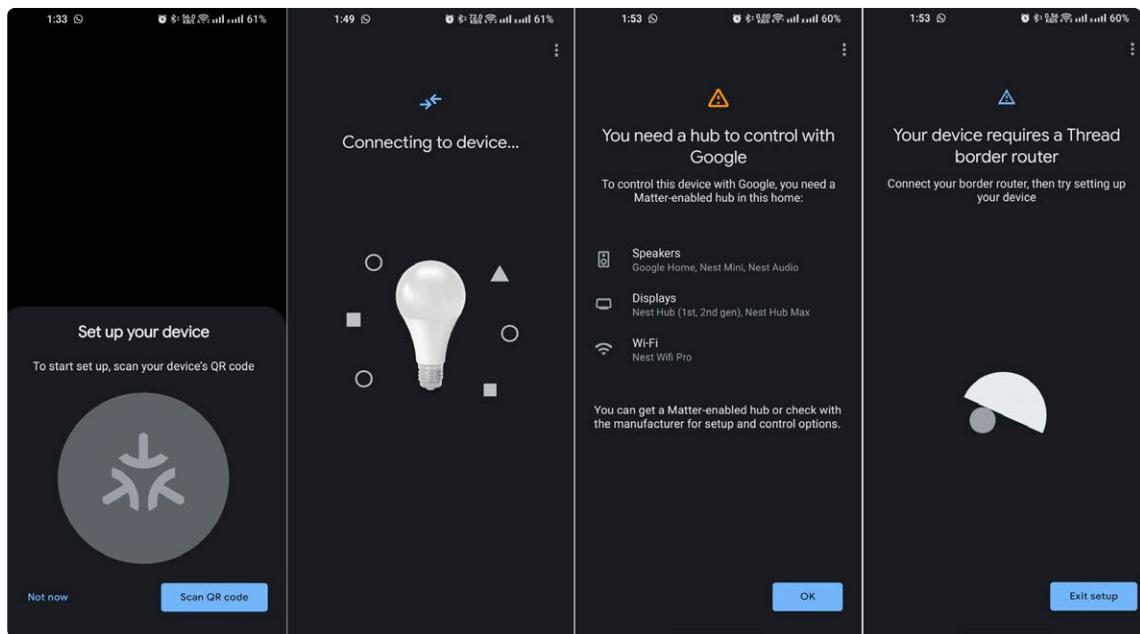


Figure 5. Pour configurer la carte avec votre application Google Home, votre téléphone détecte l'appareil juste après le chargement du code. Cependant, pour contrôler les appareils Matter avec Google, vous devez disposer d'un Hub Google.

Connectivité sans fil et tests

Cette carte est équipée d'un module MGM12P, basé sur la famille EFR32MG12 de Silicon Labs. Ce module supporte plusieurs protocoles sans fil, dont "Bluetooth Low Energy (BLE) et IEEE 802.15.4 (la base de Thread et Matter) (**figure 2**). Il existe de nombreux exemples pour cette carte sur Simplicity Studio de Sparkfun. Lors de mes tests j'ai vérifié certains de ces exemples, entre autres également l'exemple *Matter Light Over Thread*, qui correspond essentiellement à une application d'éclairage via Matter.

Le chargement du code sur cette carte est simple. Vous devez d'abord installer Simplicity Studio, et lorsque que vous branchez votre carte, celle-ci est automatiquement détectée. Toutefois, avant d'utiliser votre carte, assurez-vous d'avoir la version la plus récente du micrologiciel. Il est toujours important qu'elle soit à jour pour éviter les bugs et problèmes qui n'étaient pas encore corrigés dans les versions précédentes. (**figure 3**).

Après avoir mis à jour le micrologiciel, vous pouvez simplement cliquer sur *Create New Project*, et vous disposez alors d'une tonne d'exemples adaptés à la plupart des applications de cette carte. J'ai sélectionné l'exemple *Matter Light Over Thread*. J'ai composé le firmware, et l'ai ensuite chargé sur la carte. (**Figure 4**). Vous devez configurer le compte Google Home Developer [6]. Lorsque vous chargez le code sur votre carte,

votre smartphone détecte automatiquement cet appareil comme un appareil Matter et vous demande de l'appairer. J'ai suivi les instructions disponibles sur la page de Sparkfun [7], mais malheureusement avec Google, vous devez avoir un Google Hub pour contrôler les appareils Matter. Comme le montre la **figure 5**, il est indiqué que "vos appareils nécessitent un *Thread border router*". Quoi qu'il en soit, je n'ai eu aucune inquiétude en cours de route, et si j'avais eu un hub j'aurais pu dire en toute confiance qu'il aurait fonctionné.

Options flexibles d'alimentation

La carte (**figure 6**) offre des options flexibles pour l'alimentation, vous permettant de l'alimenter via l'USB, une batterie Li-Po ou une alimentation externe. Elle possède également des broches PTH connectées au réseau 3,3 V, "V_USB et V_BATT". La carte est équipée d'un connecteur JST à 2 broches, offrant ainsi une prise pour la connexion d'une batterie LiPo à une cellule. Elle est de ce fait appropriée aux applications alimentées par batterie. Pour garantir un fonctionnement stable, la tension d'entrée est régulée par un régulateur de tension de 3,3 V. De plus, la carte intègre deux composants clés : le "MCP73831 Single-Cell LiPo Charge IC", qui gère la recharge de la batterie connectée, lorsqu'elle est branchée via USB-C, et le "MAX17048 Single-Cell fuel gauge", qui surveille en permanence le niveau de charge de la batterie.

Par défaut, le courant de charge est fixé à 500 mA. Cependant, un cavalier à trois positions nommé CHG permet à l'utilisateur de choisir entre un courant de charge de 500 mA, de 100 mA, ou de désactiver complètement le circuit intégré de charge si cette fonction n'est pas souhaitée. Lors de mes tests, le circuit de charge par défaut a fonctionné sans aucun problème, comme prévu. Cette fonctionnalité vous permet également d'alléger votre réalisation puisque vous n'avez plus besoin d'inclure un module de charge dans votre projet.

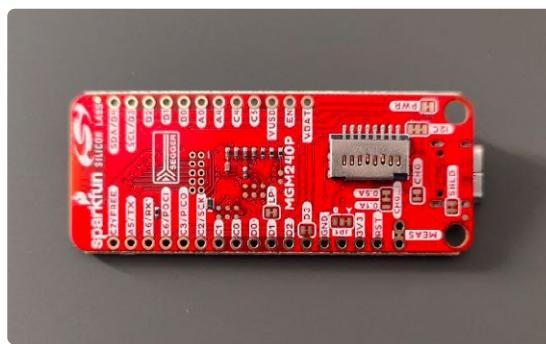


Figure 6. La carte prend en charge plusieurs options d'alimentation. Elle comprend des fonctionnalités qui améliorent la mobilité des appareils IoT.

Possibilités d'extension avec les connecteurs Qwiic

Une autre particularité de cette carte est sa compatibilité avec les connecteurs Qwiic. Qwiic est le système *plug-and-play* de SparkFun pour connecter divers capteurs et périphériques. Il simplifie le processus de configuration du matériel et permet d'ajouter facilement des composants supplémentaires à votre projet IdO.

Débogueur

Cette carte est équipée du microcontrôleur EFM32GG-12B410F1024GL120-A, servant de programmeur et de débogueur J-Link. Elle est livrée avec un connecteur Mini Simplicity pour ceux qui souhaiteraient utiliser un débogueur externe. Par défaut, le débogueur est configuré en mode standard, avec la broche WAKE du débogueur connectée à V_USB via le cavalier LP. Cependant, les utilisateurs ont la possibilité d'enlever ce cavalier pour faire passer le débogueur en mode basse consommation.

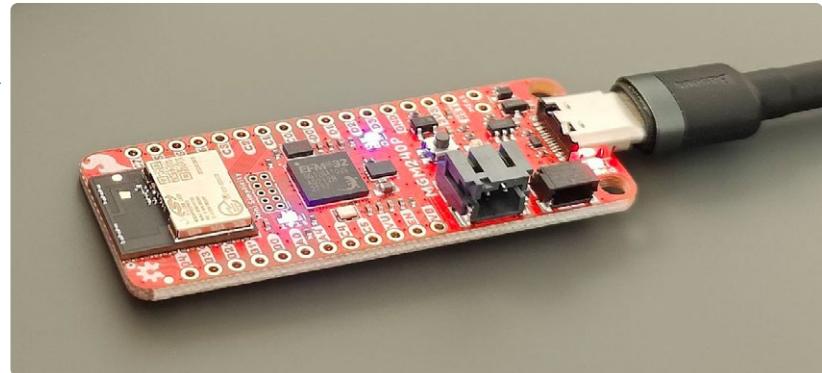
Ce débogueur offre de puissantes possibilités pour un débogage de bas niveau lorsqu'il est associé à l'outil de débogage de "Simplicity Studio". Vous pouvez effectuer un large éventail d'opérations de débogage standard, comprenant la visualisation de données, l'introduction de points d'arrêt dans le code, et même l'exploration du code assembleur pour une analyse plus détaillée.

Interopérabilité

L'un des principaux objectifs du protocole "Matter" est d'assurer l'interopérabilité entre les différents appareils IdO, quel que soit le fabricant. Avec la carte SparkFun Thing Plus Matter, vous vous trouvez dans une excellente position pour réaliser des appareils qui s'intègrent en toute transparence aux écosystèmes IdO existants et futurs basés sur Matter.

Personnalisation et prototypage

Pour ceux qui souhaitent personnaliser leurs solutions IdO, la carte offre la possibilité de souder des barrettes de connexion et de configurer divers paramètres. De plus, l'écosystème Qwiic de SparkFun fournit une gamme de capteurs et de périphériques pour perfectionner votre projet.



Idées de projets

Avec ses fonctionnalités étendues et la prise en charge du protocole Matter, la carte SparkFun Thing Plus Matter - MGM240P est idéale pour réaliser divers appareils IdO. Explorez divers projets, depuis les appareils domestiques intelligents jusqu'aux capteurs environnementaux, et exploitez les capacités de cette carte pour donner vie à vos idées innovantes.

Le SparkFun Thing Plus Matter - MGM240P est une bonne carte de développement conçue pour les solutions IdO basées sur "Matter". Sa compatibilité avec de nombreux protocoles sans fil, ses connecteurs "Qwiic", sa fonction permettant de charger une batterie ainsi que les ressources étendues de "SparkFun" et "Silicon Labs" aident les développeurs à relever divers défis IdO. Que vous lanciez un nouveau projet IdO ou que vous amélioriez un projet existant, cette carte constitue une excellente base pour l'expérimentation dans le développement IdO. ↗

VF : Jean-Philippe Nicolet — 240251-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (saad.imtiaz@elektor.com) ou contactez Elektor (redaction@elektor.fr).



Produits

➤ **SparkFun Thing Plus Matter (MGM240P)**
www.elektor.fr/20442

— LIENS —

- [1] What Is Matter? : <https://developers.home.google.com/matter/overview>
- [2] SparkFun Thing Plus Matter (MGM240P) : <https://elektor.fr/20442>
- [3] Guide de branchement du MGM240P : <https://tinyurl.com/mgm240phookup>
- [4] Module sans fil MGM240P : <https://tinyurl.com/mgm240modules>
- [5] Silicon Labs BG24 and MG24 announcement: <https://tinyurl.com/bg24mg24news>
- [6] Google Home Developer Console: <https://console.home.google.com/projects>
- [7] Tutoriel - connecter Thing Plus Matter au Nest Hub de Google : <https://tinyurl.com/thing-plus-matter-to-google>

Rétroéquipement IoT

Adaptation des machines à interface RS-232 à l'industrie 4.0

Matthias Lay (Würth Elektronik eiSos)

Dans les environnements industriels, on trouve souvent des vieilles machines encore parfaitement opérationnelles, mais auxquelles il manque une interface avec un système de bus industriel. Cette situation peut être élégamment résolue par un rétroéquipement (retrofit) IoT, comme le montre l'exemple d'une machine à souder automatique dotée d'une interface RS-232.

La numérisation croissante de l'industrie, souvent nommée Industrie 4.0, et la généralisation de l'internet industriel des objets (IIoT) nécessitent une mise en réseau toujours plus poussée des machines dans les usines de production. Les avantages de la mise en réseau dans l'environnement industriel sont déjà bien connus [1] et son importance dans le succès futur des entreprises est incontestée. Néanmoins, sa mise en œuvre effective pose toujours des difficultés. Selon une enquête du Vogel Communications Group, le problème de l'interfaçage est l'un des défis à relever lors de la mise en œuvre de projets IIoT. Les machines de production sont en effet incapables de communiquer avec d'autres systèmes en raison d'un manque de normalisation des protocoles et des interfaces, si ce n'est de leur absence totale.

Une solution possible est d'appliquer la stratégie IIoT dite Greenfield qui consiste à remplacer toutes les vieilles machines par de nouvelles machines compatibles IIoT dotées d'interfaces normalisées. Mais cette approche n'est évidemment guère raisonnable d'un point de vue économique ou écologique [2].

Prototypage rapide avec les modules Feather

Les modules Feather et FeatherWing sont idéaux pour le prototypage rapide dans

le cadre de projets de rétroéquipement. Les cartes d'évaluation sont de conception modulaire et peuvent être enfichées les unes dans les autres. Grâce au format unique de la plateforme Adafruit Feather et à la possibilité d'empiler plusieurs cartes les unes sur les autres, elles offrent la possibilité d'ajouter d'autres fonctions et interfaces au moyen de n'importe quel microcontrôleur, ce qui permet de réaliser des prototypes matériels opérationnels et d'évaluer différentes configurations en un temps très court.

Würth Elektronik propose une gamme de cartes d'évaluation de ce type - open source et entièrement compatibles avec le format Feather. Outre l'utilisation de capteurs et de divers protocoles radio (WiFi/radio mobile) et le fonctionnement du prototype sous différentes tensions d'alimentation, ces cartes permettent également d'ajouter des interfaces industrielles. Il existe un dépôt GitHub [3] pour toutes les cartes open source, y compris leurs schémas, listes de composants, logiciels et descriptions de la connectivité cloud pour Azure et AWS.

Lecture de l'interface RS-232

Dans les projets de rétroéquipement, il est particulièrement intéressant de réutiliser les interfaces de machines existantes non compatibles avec l'IIoT. Cela permet d'enregistrer et d'analyser les paramètres

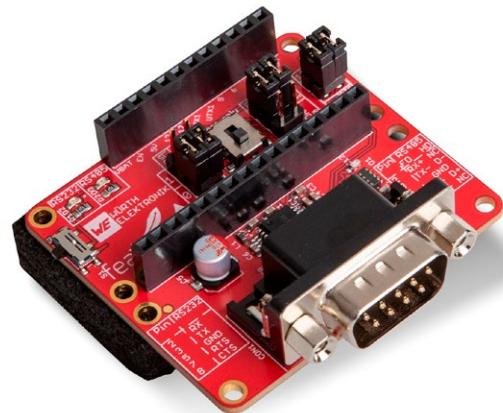


Figure 1. Pont série FeatherWing de Würth Elektronik pour les protocoles RS-232, RS-422 and RS-485.

de la machine sans qu'il soit nécessaire de rajouter des capteurs externes.

L'interface RS-232 est celle qu'on trouve le plus fréquemment sur les anciennes machines. Elle est très robuste malgré sa simplicité. Une maîtrise de ses paramètres de base : la vitesse de transmission, la parité et les bits d'arrêt, suffit en général pour arriver à transmettre les données avec succès.

Pour connecter une interface RS-232 au microcontrôleur du prototype de mise à niveau, on utilise un pont série FeatherWing de Würth Elektronik (**figure 1**). Il convertit une interface série en une interface UART, disponible sur la plupart des microcontrôleurs. En configurant l'interface UART du microcontrôleur par logiciel, on peut utiliser des protocoles de communication courants tels que MODBUS ou ASCII et, en configurant le pont série, les normes de communication physique RS-232, RS-422 et RS-485 peuvent être opérées dans les modes de fonctionnement half et full duplex.

Connexion IoT via Wi-Fi/MQTT

Une interface IoT appropriée est nécessaire pour intégrer les prototypes de mise à niveau dans un système IoT existant. Le protocole MQTT, l'un des protocoles les plus largement utilisés dans le monde de l'IoT, peut être utilisé comme protocole pour l'interface. En raison de sa simplicité et de

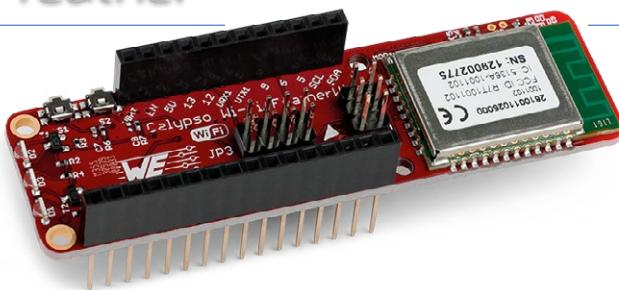


Figure 2. Il est facile de réaliser une communication sans fil WiFi avec Calypso FeatherWing de Würth Elektronik.

ses faibles exigences en matière de bande passante, il est particulièrement adapté à la réception de données provenant de nombreuses machines différentes [4]. Son avantage est qu'il peut être basé sur n'importe quel type de communication TCP. L'écosystème Feather permet d'utiliser différentes méthodes de transmission en fonction des besoins du rétroéquipement : par exemple, une transmission par câble, en utilisant un FeatherWing Ethernet ou une transmission sans fil avec un FeatherWing radio.

Comme le prototype devait être mis en œuvre et testé le plus rapidement possible, le choix s'est porté sur le FeatherWing Calypso WiFi de Würth Elektronik (**figure 2**). Il peut être facilement intégré dans le réseau local d'une usine de production via les points d'accès WiFi existants (**figure 3**) et élimine la nécessité de tirer des câbles Ethernet supplémentaires.

Mise en œuvre

La numérisation de l'unité de soudage d'une chaîne de production sert d'exemple pratique pour la mise en œuvre d'un prototype (**figure 4**). Ce maillon de la chaîne de production a été identifié à priori comme le plus apte à être modernisé. L'idée de base est qu'en évaluant les paramètres de soudage, il est possible de tirer des conclusions sur l'état courant de la thermode. Ces informations peuvent indiquer au personnel de maintenance si une thermode risque d'atteindre sa limite de fonctionnement au cours de la prochaine période de production. De cette manière, la thermode peut

être remplacée avant que la défaillance réelle ne se produise, afin d'éviter un arrêt imprévu de la chaîne de production.

Le pont série FeatherWing est configuré par logiciel en mode RS-232 en fonction de la machine à équiper et adapté à la vitesse de transmission et à la configuration du convertisseur de soudage. Après chaque soudure par point, le convertisseur transmet tous les paramètres enregistrés pour cette soudure en utilisant le protocole ASCII.

Les données de soudage sont déjà transférées au sein de la machine via RS-232 du convertisseur de soudage à l'automate programmable afin de créer un rapport d'assurance qualité local. Pour pouvoir lire cette transmission de données, la voie de réception RS-232 du côté de l'automate programmable et le potentiel de référence RS-232 doivent être matériellement accessibles et être connectés au pont série FeatherWing. Le prototype lit alors en continu les transmissions RS-232 et convertit les données transmises dans le format de données requis par le système IoT.

Le microcontrôleur transmet ensuite les données au module WiFi FeatherWing Calypso. Celui-ci est connecté au réseau IIoT local et peut donc transmettre les données directement au système IoT.

Les données maintenant disponibles dans le système IoT (**figure 5**) peuvent être utilisées pour obtenir une vue en temps réel des paramètres de soudage et de l'utilisation de la ligne de production et, à l'étape suivante, pour effectuer une analyse des données sur la base des paramètres de soudage.



Figure 4. Machine à module de soudage pour le rétroéquipement.

Amélioration des processus par l'analyse

Le prototype fonctionne et permet de contrôler en continu l'utilisation de la machine, le temps de cycle et les paramètres de soudage. Il permet également d'estimer l'état de la thermode.

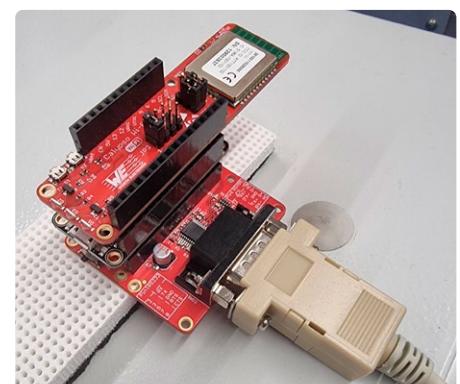


Figure 5. Modules Feather empilés, dessus : Calypso WiFi module, milieu : microcontrôleur M0, dessous : pont série.

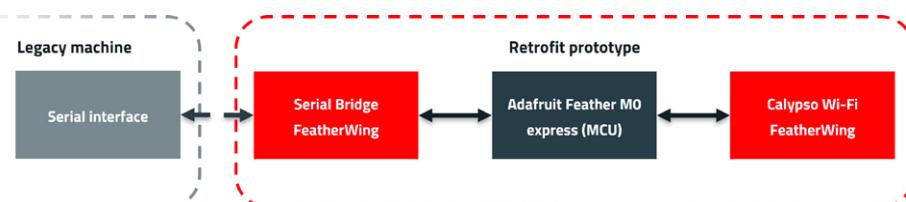


Figure 3. Diagramme de flux de données de la connexion de l'ancienne machine au système IIoT existant.

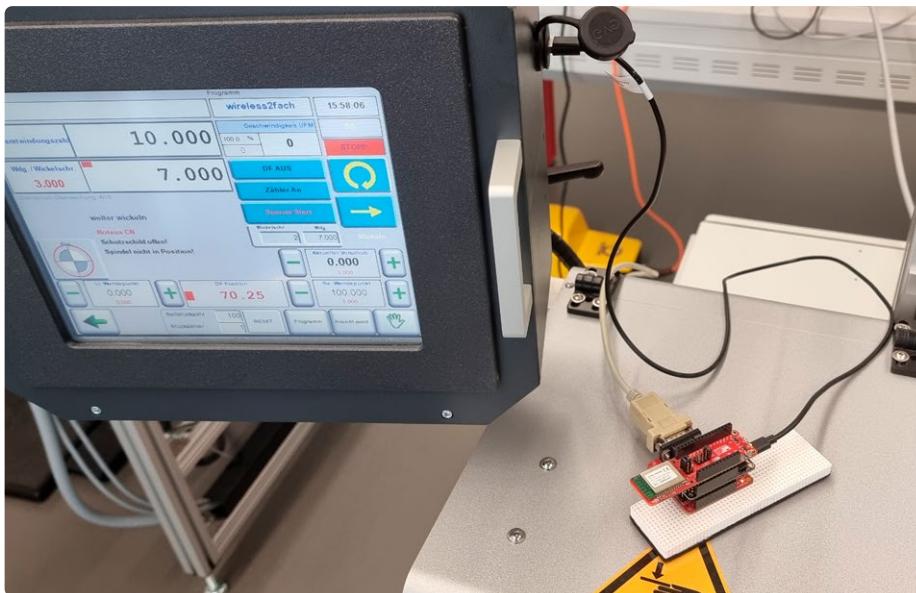


Figure 6. Le même prototype équipant une machine à bobiner.

L'analyse des données enregistrées a déjà permis de déterminer que l'ajustement de certains paramètres de soudage a des effets bénéfiques sur la durée de vie de la thermode. Par exemple, il est apparu clairement qu'en réduisant la puissance de soudage et en augmentant la durée du soudage, il est possible d'allonger nettement cette durée de vie avec une augmentation minime de la durée du cycle.

Cependant, la mise en œuvre du prototype de rétroéquipement n'a pas été sans poser de problèmes. Comme le convertisseur de soudage est logé dans la base de la machine et que cette base est entièrement recouverte de panneaux métalliques, elle agit comme une cage de Faraday. Le prototype a donc dû être monté à l'extérieur de la base pour garantir une liaison sans fil stable au point d'accès.

En outre, la configuration actuelle ne

permet aux données de circuler que dans une seule direction, à savoir du convertisseur de soudage au système IoT. La raison en est que la communication RS-232 n'utilise que deux lignes de données et qu'il ne peut y avoir qu'un émetteur actif par ligne de données. Pour pouvoir modifier à l'avenir les paramètres de soudage à l'aide du système IoT, la communication doit être bidirectionnelle. À cette fin, une extension du prototype avec un pont série FeatherWing supplémentaire est envisagée. Cela permettrait non seulement de lire la communication, mais aussi d'envoyer des ajustements de paramètres au convertisseur de soudage.

Le prototype présenté, utilisé sur un convertisseur de soudage, peut l'être de différentes manières : par exemple, la **figure 6** montre un prototype identique installé sur une machine à bobiner.

FeatherWing, kit de base du prototypage

En conclusion, on peut dire que l'utilisation du format Feather permet de raccourcir substantiellement le temps de développement lors de la création du prototype matériel. Au lieu de concevoir une carte avec tous les composants nécessaires, de l'assembler et de la mettre en service, les cartes d'évaluation au format Feather sont simplement spécifiées selon la fonction requise et branchées ensemble. Cela signifie que le développement du logiciel peut commencer très rapidement et que le prototype matériel peut être modifié beaucoup plus facilement en cas de problèmes imprévus ou de changement du cahier des charges. En bref, l'utilisation du format Feather rend le développement du matériel plus souple et accélère considérablement l'ensemble du processus de prototypage. ▶

Vf: Helmut Müller — 240280-04



À propos de l'auteur

Matthias Lay a étudié le génie électrique à l'Université des Sciences Appliquées de Heilbronn, où il s'est spécialisé dans la technologie de l'automatisation et a obtenu une maîtrise en sciences. Il travaille chez Würth Elektronik eiSos depuis 2019, d'abord dans les domaines du développement matériel et logiciel, puis, depuis 2023, comme ingénieur système IoT, avec un accent sur l'IoT et le rétroéquipement.

LIENS

- [1] Sniderman, B. et al.: Industry 4.0 et les écosystèmes de fabrication – Exploration du monde des entreprises connectées. Deloitte : <https://tinyurl.com/5745usv8>
- [2] Pietrangeli, I. et al.: Le rétroéquipement intelligent : une solution innovante et durable. MDPI : <https://mdpi.com/2075-1702/11/5/523>
- [3] Dépôt GitHub : <https://github.com/WurthElektronik/FeatherWings>
- [4] Ries, U.: Le protocole MQTT : Échanges IoT librement accessibles dans les réacteurs ou les prisons. Heise Security [en Allemand] : <https://tinyurl.com/388yrykb>

ajouter l'IoT grâce aux microcontrôleurs 8 bits

Joshua Bowen (Microchip Technology)

Depuis leur avènement dans les années 1970, les microcontrôleurs (MCU) sont utilisés pour gérer différents produits automobiles, de grande consommation et industriels. Leur utilisation s'est aujourd'hui étendue et comprend également les objets sans fil portables et les objets connectés portés sur soi, applications de l'Internet de Objets (IoT). De plus, le secteur de la santé a connu une croissance significative et a intégré des microcontrôleurs 8 bits dans de nombreux systèmes embarqués.

L'électronique embarquée utilisant des microcontrôleurs 8 bits doit être à la fois dotée de grandes capacités et être économiquement viable, dans le contexte d'échelles de production atteignant bien souvent des centaines de milliers, voire des millions d'unités par application. Sur les applications automobiles, les microcontrôleurs 8 bits gèrent de nombreux sous-systèmes, depuis la motorisation des sièges, les lève-vitres électriques ou l'ouverture des portières électriques, sans oublier les capteurs de pression des pneumatiques. En raison de cette utilisation à grande échelle, la moindre différence de coût, aussi infime soit-elle, se révèle capitale. Les coûts de maintenance de millions d'appareils, souvent sous-estimés pendant la conception, peuvent être limités en améliorant la fiabilité et la durabilité grâce à la simplification du code et à des améliorations matérielles, réduisant ainsi le besoin de redondance logicielle.

La popularité durable des microcontrôleurs 8 bits est principalement due à leur innovation continue sur le plan de la mémoire, de la consommation énergétique, des boîtiers et des périphériques indépendants du cœur.

Améliorations significatives des 8 bits

Face à la multiplication des applications IoT et à la transformation des villes à l'aide d'équipements intelligents, l'intelligence évolutive devient désormais vitale pour de nombreux secteurs. Des évolutions telles que l'éclairage public intelligent et les détecteurs de places de parking individuels nécessitent des microcontrôleurs capables de collecter des données, de les traiter et de les transmettre. Souvent, ces tâches peuvent être gérées par un microcontrôleur 8 bits doté d'un convertisseur A/N intégré, tandis que le cœur reste dans un état de faible consommation. Cette approche convient pour les parkings intelligents, les réverbères connectés et le jardinage urbain automatisé, où l'efficacité énergétique s'avère critique.

Les avantages des composants plus petits sont évidents, comme leur consommation énergétique réduite et leur taille compacte, qui leur permettent d'être intégrés dans des espaces réduits sur des applications d'objets connectés portables.

Les microcontrôleurs les plus récents sont conçus avec le rapport coût/efficacité à l'esprit, offrant les fonctionnalités nécessaires tout en étant attentif aux préoccupations budgétaires de l'utilisateur. De plus, les progrès en matière de technologie de mémoire ont significativement augmenté les capacités des microcontrôleurs modernes.

Mémoire

Les microcontrôleurs modernes ont évolué de façon significative, poussés par les avancées dans le domaine des mémoires Flash. Les applications actuelles exigent des programmes plus complexes, nécessitant davantage de capacité de mémoire sur les microcontrôleurs. La mémoire Flash embarquée sur ces composants est durable, capable de supporter les tests automobiles rigoureux et de nombreux cycles de lecture/écriture. Les microcontrôleurs 8 bits offrent des capacités de mémoire allant de 384 bits à plus de 128 ko, permettant un large éventail d'applications.

Consommation énergétique

L'efficacité énergétique est devenue une préoccupation centrale dans la conception des microcontrôleurs 8 bits, en particulier pour les applications alimentées par batterie. Par exemple, les microcontrôleurs PIC eXtreme Low Power XLP nanoWatt [1] intègrent des circuits de supervision des systèmes conçus pour une consommation minimale. Ces innovations débouchent sur les courants les plus faibles du marché pour le mode actif comme pour le mode veille, dans lequel les composants passent 90 à 99 % du temps. Des technologies telles que la désactivation des modules périphériques

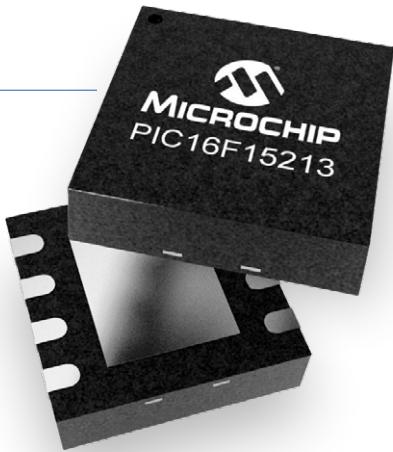


Figure 1. De nombreux nouveaux produits des familles PIC et AVR se déclinent en différents types de boîtiers, pour des composants VQFN atteignant 3x3 mm, idéaux pour les applications avec des contraintes d'espace.

améliorent encore les économies d'énergie en supprimant totalement les périphériques du rail d'alimentation et de l'arbre d'horloge. Parmi les avantages de la technologie XLP nanoWatt, on peut citer :

- Courants de veille inférieurs à 20 nA
- Réinitialisation BOR jusqu'à 45 nA
- Timer du Watchdog jusqu'à 220 nA
- Horloge/calendrier temps réel jusqu'à 470 nA
- Courants de fonctionnement jusqu'à 50µA/MHz
- Capacité d'auto-écriture et capacité « full analogique » jusqu'à 1,8 V

Puisque de nombreux microcontrôleurs 8 bits sont utilisés sur les applications alimentées par batterie, il est possible de réaliser encore plus d'économies d'énergie en utilisant des périphériques optimisés, que nous présenterons plus tard dans le présent article.

Boîtier

L'une des principales caractéristiques qui distinguent les microcontrôleurs 8 bits est leur capacité à pouvoir être encapsulés dans des boîtiers réduits, idéaux pour les applications ayant des contraintes d'espace sur les produits sans fil, portables et portés sur soi. Des boîtiers tels que le SOIC à 8 broches ou le DFN à 8 broches, ainsi que le populaire VQFN à 20 broches (Very Thin Flat Pack No-Leads), constituent des solutions compactes (**Figure 1**). Pour les applications plus complexes, des boîtiers plus gros comme le PDIP à 40 broches ou le TQFP à 44 broches sont disponibles.

Périphériques indépendants du cœur (CIP)

Les périphériques indépendants du cœur (CIP) améliorent la fonctionnalité des microcontrôleurs en fonctionnant de façon autonome par rapport au cœur, ce qui constitue un avantage pour les systèmes faible consommation économiques. Les périphériques CIP peuvent gérer de nombreuses tâches de façon indépendante, ce qui réduit le besoin d'intervention du CPU (Central Process Unit) et augmente par conséquent l'efficacité et la fiabilité du système. Cette approche modulaire simplifie, entre autres, la mise en œuvre des interfaces tactiles et du traitement des données issues de capteurs.



Figure 2.
Les périphériques indépendants du cœur répondent aux attentes d'un grand nombre de domaines d'utilisation des microcontrôleurs 8 bits.

8-bit PIC® and AVR® Microcontrollers	
CPU	Memory
(Enhanced) Capture/ Compare/ PWM	Input Capture
8-/10-/12-bit ADC	Direct Memory Access Controller
ADC with Gain Stage	Configurable Custom Logic
Complementary Output Generator	Configurable Logic Cell
Angular Timer	High Endurance Flash (Data)
Complementary Waveform Generator	Crypto Engine AES/DES
Charge Time Measurement	Event System
Data Signal Modulator	CAN
RTC/C	IDLE & DOZE
Numerically Ctd Oscillator	(E)USART
Programmable Switch Mode Cntlr	ETHERNET MAC
High Speed Comparators*	I²C/TWI
DAC	LIN
10b/12b/16b PWM	SPI™
Operational Amplifiers*	Keeloq® Sub-GHz RF
8-/12-/16-/20-/24-bit Timers	Crystal Free USB
Programmable Switch Mode Cntlr	Full-Speed USB Device w/w/o OTG
10b/12b/16b PWM	IRCOM
Quadrature Decoder	
Ramp Generator*	
Waveform Extension	
Slope Compensation*	
Clock Failure Detection	
Voltage Reference	
Cyclical Redundancy Check	
Zero Cross Detect*	
Hardware Limit Timer	
mTouch® solution	
Qtouch Solution	
Peripheral Touch Controller	
Windowed WDT	
Brown-Out Detection	
LCD	
	Math Accelerator

Cette façon de procéder fournit un moyen pré-intégré de programmer les événements, basé sur les périphériques. Par exemple, le Système de gestion des événements peut déclencher des événements à partir des entrées/sorties à usage général (GPIO) ou de l'interruption d'un programme sur plusieurs canaux.

Sur la **Figure 2**, les CIP actuellement disponibles pour les microcontrôleurs PIC et AVR® 8 bits sont indiqués par un code de couleur, chaque couleur correspondant à une catégorie de périphériques. Les huit catégories et leurs sous-catégories permettent de mettre en œuvre la plupart des fonctionnalités attendues sur un contrôleur embarqué au bon rapport coût/efficacité. À noter que les éléments en vert permettent des possibilités de réduction de la consommation supplémentaires par rapport à celles indiquées précédemment.

Ils offrent une fiabilité accrue en réduisant la quantité de code nécessaire. Les fonctions mises en œuvre par les structures matérielles permettent d'éviter les conflits logiciels potentiels. De plus, l'interconnectivité des périphériques sur le plan matériel réduit les connexions externes, augmentant la fiabilité du système final. La fiabilité accrue des composants réduit par ailleurs les coûts tout au long de la durée de vie du projet.

Les nouvelles familles de microcontrôleurs 8 bits offrent de nombreuses options en termes de mémoire et de nombre de broches, ce qui permet aux développeurs de démarrer avec des composants plus gros et de descendre en taille ensuite une fois le code optimisé. Par exemple, la famille PIC16F152XX est conçue pour les applications sensibles au coût, de capteurs et de contrôle en temps réel, intégrant un convertisseur analogique-numérique (convertisseur A/N) 10 bits, la sélection PPS, des périphériques de communication numérique et des temporiseurs. Les fonctionnalités de mémoire comprennent la partition de l'accès mémoire (MAP), pour aider les utilisateurs dans les applications de protection des données et de bootloader.

Outils de conception pour accélérer et simplifier les applications

Les progrès en matière d'outils de développement ont permis de simplifier le processus de développement, en diminuant le besoin en codage. Des outils tels que le générateur de code MPLAB Code Configurator [2] (MCC) permettent de générer un code compact et efficace, réduisant significativement les temps de développement. Le kit de la carte d'évaluation Curiosity

Nano PIC16F15244 [3] (Réf. : EV09Z19A) en est un bon exemple, puisqu'il offre une assistance complète pour développer de nouveaux systèmes grâce à des capacités complètes de programmation et de débogage (**Figure 3**).

L'environnement de développement MPLAB X (IDE) fournit une plateforme polyvalente pour développer du code pour des microcontrôleurs 8, 16 et 32 bits, permettant la simulation matérielle et l'intégration avec des outils tiers.

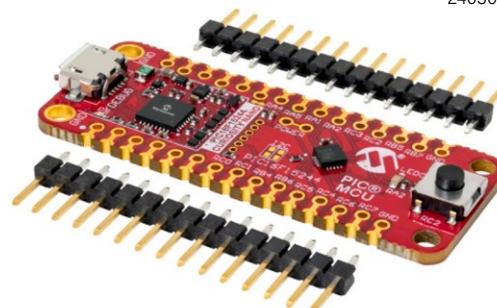
Un bel avenir (au bon rapport coût/efficacité)

Les microcontrôleurs ont remarquablement évolué, les microcontrôleurs 8 bits menant la charge sur le plan de la mémoire, de la consommation énergétique, des boîtiers et des périphériques. Ils permettent de créer des applications complexes avec davantage de mémoire et des processus de développement simplifiés, réduisant le temps de développement comme les coûts de production. L'adaptabilité et l'efficacité des microcontrôleurs 8 bits actuels en fait un choix privilégié pour de nombreuses applications IoT, garantissant un avenir radieux et au bon rapport coût/efficacité pour les systèmes embarqués.

Les microcontrôleurs 8 bits modernes font bien plus que collecter des données : ils sont capables de collecter, traiter et envoyer les données vers de nombreuses applications IoT. Les derniers modèles 8 bits résolvent la complexité croissante de ces applications en offrant des capacités de mémoire significativement plus grandes ainsi que des périphériques améliorés. Cependant, pour les systèmes compacts et au bon rapport coût/efficacité, tels que les capteurs et applications de contrôle en temps réel, les caractéristiques simplifiées de la famille PIC16F152xx 8 bits se révèlent particulièrement avantageuses. Dotés de périphériques indépendants du cœur, ces microcontrôleurs sont le choix numéro un de nombreux développeurs. ▶

240306-04

Figure 3. La carte d'évaluation Curiosity Nano PIC16F15244, ainsi que les deux barrettes d'embase de 10 mil, 1x15 broches fournies dans le kit d'évaluation de la carte Curiosity Nano, permettent de simplifier la conception.



LIENS

- [1] Microcontrôleurs PIC eXtreme Low Power XLP nanoWatt : <https://ww1.microchip.com/downloads/en/devicedoc/39941d.pdf>
- [2] Générateur de code MPLAB Code Configurator : <https://microchip.com/en-us/tools-resources/configure/mplab-code-configurator>
- [3] Kit de la carte d'évaluation Curiosity Nano PIC16F15244 : <https://microchip.com/en-us/development-tool/EV09Z19A>

la technologie au service du développement durable

les avancées technologiques favorisent une utilisation plus efficace de l'énergie dans de nombreuses applications

Mark Patrick (Mouser Electronics)

Améliorer l'efficacité est une quête constante dans l'industrie électronique. Qu'elle soit obtenue par le biais de solutions nouvelles ou innovantes qui consomment moins d'énergie ou nécessitent moins de matières premières ou via une amélioration du développement de produits ou de la maintenance, une efficacité accrue se traduit souvent par des conceptions plus durables.

Cette tendance continue de l'industrie électronique s'inscrit dans le cadre de nombreuses initiatives lancées par les gouvernements et les groupes de travail du monde entier. Certaines de ces initiatives visent à produire de l'énergie de manière plus durable, tandis que d'autres incitent à une utilisation raisonnée de l'énergie en réduisant le gaspillage, comme on peut notamment le retrouver dans les Objectifs de développement durable (ODD) des Nations Unies [1]. Parallèlement, la course à la technologie et à l'automatisation se poursuit. Ce secteur est pourtant un grand consommateur d'énergie, comme en atteste le fait que 40 % de l'énergie produite mondialement sont consommés par des moteurs électriques [2]. Malgré tout, le secteur technologique facilite considérablement le passage à un mode de vie durable, et ce, dans de nombreux domaines.

La durabilité commence par une conception de qualité

Un produit qui n'a pas été conçu pour être durable ne le sera jamais. C'est ce que les concepteurs ne doivent jamais perdre de vue pour créer des conceptions qui fonctionnent de manière efficace et qui consomment très peu d'énergie en mode veille ou veille prolongée. Afin de les soutenir dans cette voie, de nombreux fabricants de semi-conducteurs développent aujourd'hui des solutions permettant d'améliorer l'efficacité énergétique. Des dispositifs tels que les semi-conducteurs de puissance, les microprocesseurs et les modules de communication (pour n'en citer que quelques-uns) participent grandement à augmenter les performances tout en réduisant la consommation d'énergie.

Étant donné l'importance grandissante que prend la réduction de



Figure 1. Grâce à l'outil Otii Ace Pro, les concepteurs peuvent acquérir une compréhension complète de la consommation électrique de leurs conceptions. (Source : Mouser Electronics)

la consommation d'énergie dans le processus de conception, un certain nombre d'outils sont récemment apparus pour faciliter la tâche des concepteurs. Par exemple, le fabricant Qoitech propose l'outil Otii Ace Pro, qui combine une alimentation et un instrument de mesure [3] (voir **figure 1**). Ce petit module de laboratoire mesure et enregistre les valeurs de courant et de tension, ce qui lui permet en outre de réaliser des analyses en temps réel et d'estimer la durée de vie de la batterie.

Cet outil mesure, enregistre et analyse la consommation électrique globale, le courant de fuite et le courant en mode veille. Les concepteurs disposent ainsi de toutes les informations dont ils ont besoin pour améliorer la durabilité de leurs conceptions.

Les semi-conducteurs durables sont essentiels à l'efficacité

La production d'énergie renouvelable est l'un des domaines qui présentent les plus grands défis pour les semi-conducteurs, car, suivant les objectifs de développement durable, la moindre quantité d'énergie utilisée doit être convertie en énergie électrique. Un autre domaine sensible est celui des véhicules électriques, où l'impératif est de maximiser l'autonomie que peut offrir une batterie dont la capacité est fixe.

Depuis fort longtemps, les dispositifs semi-conducteurs sont essentiellement fabriqués à partir de silicium. Même s'il demeure le matériau de prédilection pour la plupart des applications, le silicium peine pourtant à fournir le niveau d'efficacité attendu dans les conditions d'utilisation les plus difficiles. C'est pourquoi les concepteurs tendent aujourd'hui à se tourner vers d'autres matériaux,

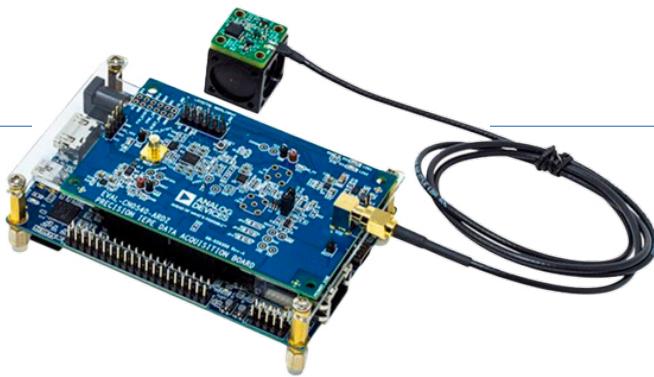


Figure 2. La plateforme de développement CN0549 d'Analog Devices surveille les machines afin que le personnel de maintenance intervienne uniquement lorsque cela s'avère nécessaire. (Source : Mouser Electronics)

parmi lesquels le carbure de silicium (SiC), qui a pour avantage d'offrir une fiabilité élevée avec des pertes ultra-faibles, même lors d'opérations à haute fréquence.

Prenons l'exemple des onduleurs. Ils sont notamment utilisés pour convertir le courant continu généré par les panneaux solaires photovoltaïques en courant alternatif à injecter dans le réseau. Dans cette application, le silicium peut atteindre une efficacité de 98 %, ce qui est tout à fait honorable. Cela dit, dans cette même application, une solution SiC réduit les pertes de moitié tout en affichant une efficacité de 99 %. Dans un pays comme les États-Unis, qui produit chaque année 60 W d'électricité d'origine photovoltaïque, la conversion de tous les onduleurs classiques en onduleurs SiC permettrait d'économiser 600 MW de pertes. En Europe, où la production annuelle d'électricité d'origine photovoltaïque dépasse les 200 GW, le passage au SiC permettrait d'économiser plus de 2 GW !

Si les dispositifs SiC hautes performances ne sont pas encore très répandus, de nombreux fournisseurs en possèdent déjà dans leur catalogue. Ainsi, Onsemi [4] propose une large gamme de produits SiC, dont les MOSFET M3S EliteSiC [5] qui promettent une réduction de 40 % des pertes de commutation totales par rapport aux MOSFET de génération précédente.

La durabilité dans la maintenance industrielle

Essentielles à la fabrication des produits de notre quotidien, les usines devront désormais produire des biens de consommation durables. Par le passé, elles étaient également de grosses consommatrices d'énergie, mais la hausse des prix de l'énergie et les impératifs de développement durable ont amené les industriels à être plus attentifs aux économies d'énergie.

Par exemple, dans beaucoup d'usines et d'entrepôts, l'éclairage est commandé par des détecteurs de présence à très faible consommation afin de limiter le gaspillage d'énergie.

Tenter d'économiser sur la maintenance est une entreprise délicate. Réaliser des opérations de maintenance trop fréquemment est un gaspillage de ressources, mais les retarder trop longtemps, c'est s'exposer à un plus grand risque de pannes et par conséquent à des coûts de réparation élevés. Il existe heureusement des appareils tels que la plateforme de développement de surveillance conditionnelle CN0549 d'Analog Devices Inc. [6][7] (voir **figure 2**), intégrant la carte d'acquisition de données IEPE CN0540 [8], la carte d'évaluation de circuit CN-0532 [9], ainsi que le bloc de montage EVAL-XL-MOUNT1 [10]. Cette plateforme de développement est capable de distinguer la présence de vibrations qui indiquent une usure de la machine et de déterminer si une opération de maintenance est nécessaire.

Conclusion

Le développement durable devient un sujet incontournable et l'industrie des technologies en sera un des principaux acteurs. C'est en effet le développement durable qui pousse déjà à rechercher des moyens de produire de l'électricité de manière plus efficace et à faire un usage raisonnable de cette précieuse ressource grâce à de bonnes pratiques dans le domaine de la conception, à l'automatisation et à la surveillance. 

240281-04



À propos de l'auteur

En tant que directeur du contenu technique chez Mouser Electronics dans la région EMEA, Mark Patrick est responsable de la création et de la diffusion du contenu technique dans la région – un contenu essentiel à la stratégie de Mouser pour soutenir, informer et inspirer le public technique. Avant de prendre la direction du contenu technique, Mark Patrick faisait partie de l'équipe EMEA Supplier Marketing de Mouser et a joué un rôle important dans l'établissement et le développement de relations avec les principaux partenaires de fabrication. L'expérience antérieure de Mark comprend des tâches pratiques d'ingénieur, l'assistance technique, les ventes techniques de semi-conducteurs et divers postes de marketing. Mark est un ingénieur pratique et possède un diplôme de premier ordre en génie électrique de l'université de Coventry. Il est passionné par les vieux synthétiseurs et les motos britanniques, et n'hésite pas à les entretenir ou à les réparer.

— LIENS —

- [1] United Nations' Sustainable Development Goals (SDGs) : <https://sdgs.un.org/goals>
- [2] IEA, "Energy Efficiency Policy Opportunities for Electric Motor-Driven Systems," 2011 : <https://iea.org/reports/energy-efficiency-policy-opportunities-for-electric-motor-driven-systems>
- [3] Qoitech Otii Ace Pro Power Supply and Measuring Instrument : <https://tinyurl.com/qoitech-otii>
- [4] Onsemi at Mouser : <https://tinyurl.com/manufacturer-onsemi>
- [5] M3S EliteSiC MOSFETs : <https://tinyurl.com/M3S-EliteSiC>
- [6] CN0549 Dev Platform: <https://tinyurl.com/ADCN0549>
- [7] Analog Devices at Mouser : <https://tinyurl.com/manufacturer-AD>
- [8] EVAL-CN0540-ARDZ: <https://tinyurl.com/ADCN0540>
- [9] EVAL-CN0532-EBZ : <https://tinyurl.com/ADCN-0532>
- [10] EVAL-XLMOUNT1 : <https://tinyurl.com/EVAL-XLMOUNT1>

AWS

pour Arduino et Cie. (1)

Utiliser AWS IoT ExpressLink en pratique

Tam Hanna (Hongrie)

Se connecter à Amazon Web Services (AWS) n'est pas toujours une tâche aisée. Bien que cela soit toujours possible avec un Raspberry Pi, les ordinateurs monocartes sont souvent trop encombrants pour de nombreuses applications IoT pratiques. C'est là qu'intervient AWS IoT ExpressLink, qui offre aux petits appareils un moyen simple de se connecter à AWS. Le module de connectivité basé sur AWS IoT ExpressLink gère les communications avec le cloud, ce qui permet au microcontrôleur hôte de réaliser des mesures et des tâches de contrôle. Dans ce premier article d'une série en deux parties, nous allons voir comment configurer un module ExpressLink avec un ESP32.

Les services en nuage (cloud) tels que Microsoft Azure et Amazon Web Services (AWS) offrent de puissantes solutions de stockage, de traitement et de visualisation de données. L'interaction avec les services cloud n'est pas vraiment simple, mais possible avec les ordinateurs monocartes tels que le Raspberry Pi ou l'Orange Pi.



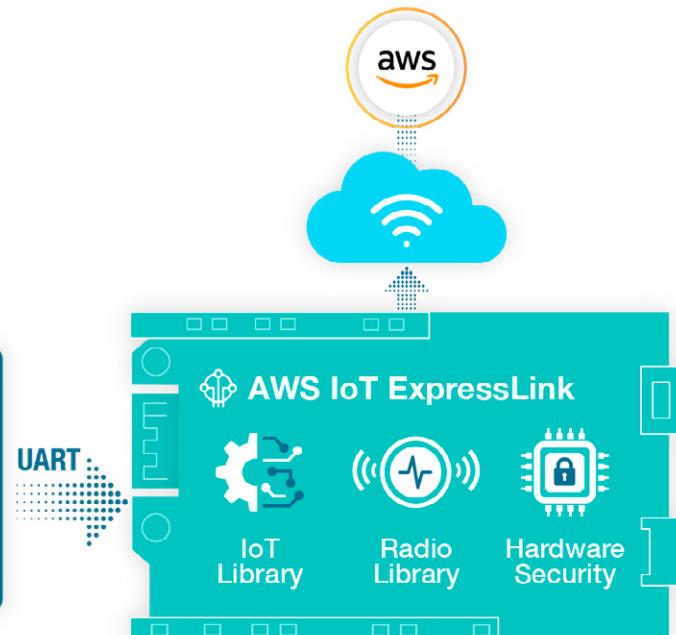
Figure 1. Certains utilisateurs appellent le module ExpressLink le "modem AWS." (Source : [1])

Cependant, ces systèmes peuvent s'avérer trop encombrants pour de nombreuses applications pratiques. Pour les cartes à microcontrôleur, l'implémentation d'une pile TCP/IP complète sur un contrôleur 32 bits est possible, mais elle consomme beaucoup de ressources. Sur des appareils plus petits et moins performants, cette tâche devient encore plus difficile.

Avec la technologie AWS IoT ExpressLink, Amazon vise à aider les concepteurs de petits appareils à surmonter cette difficulté. Dans cet article, nous allons explorer les possibilités et les limites de cette technologie avant de procéder à des essais avec un Arduino Uno R4 WiFi et un ESP32-C3-AWS-ExpressLink-DevKit d'Espressif. L'objectif est de présenter un « flux de travail de bout en bout » utile pour vos propres projets.

Protocole AT

Le protocole AT, à l'origine développé par Hayes Microcomputer Co. pour la gestion des modems, reste utile même dans une application d'accès à distance. En fait, le protocole de contrôle reste largement utilisé, notamment dans le contrôle de modules radio modernes, comme peuvent en témoigner les professionnels du domaine.



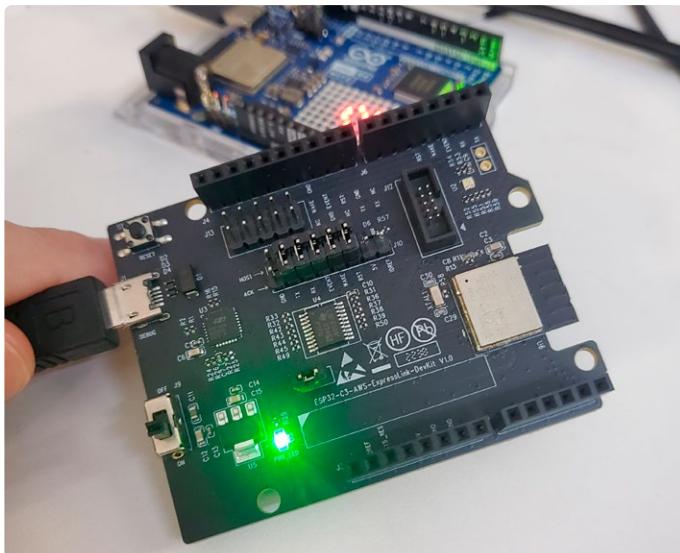


Figure 2. Avec ce module, cela fonctionne !

Le fonctionnement d'un module ExpressLink est expliqué dans la documentation d'Espressif. La **figure 1** montre comment les différents éléments fonctionnent ensemble. Le module ExpressLink communique avec le microcontrôleur hôte via une interface simple, dédié uniquement à la gestion de l'interaction avec les capteurs et les actionneurs. Le logiciel du microcontrôleur hôte se charge d'envoyer des commandes correctement formatées au module, qui prend en charge la communication avec le cloud.

Il convient de noter que le module ExpressLink gère aussi l'interface radio. Ceux qui souhaitent combiner un tel module avec un SoC radio autonome, tel que le GigaDevice GD32W515 ou une autre carte ESP32, doivent s'assurer que la communication dans le cloud soit effectuée via l'émetteur-récepteur du module ExpressLink.

L'un des avantages de ce choix est que la plupart des modules ExpressLink sont dotés d'une antenne. Cela nous permet d'éviter la complexité liée à la conception de l'antenne, de l'analyse du réseau vectoriel, etc.

L'ESP32-C3-AWS-ExpressLink-DevKit présenté dans la **figure 2** est un kit de démarrage intéressant et compatible avec diverses cartes Arduino. Selon oemsecrets.com, il est disponible au prix approximatif de 28 euros (voir [2]). Il constitue un outil abordable pour commencer à utiliser AWS IoT ExpressLink. Dans les étapes suivantes, nous utiliserons un Arduino UNO R4 WiFi comme carte de base, sans utiliser son module Wifi (voir ci-dessus).

Configuration du matériel

En théorie, la configuration de la carte et son utilisation en ligne sont simples — Espressif propose le module ExpressLink sous la forme d'un shield Arduino, il suffit donc de connecter les deux cartes.

Cependant, la figure 1 peut prêter à confusion, car l'interface de communication entre le microcontrôleur hôte et le module radio ne se limite pas à une simple liaison série. La **figure 3** montre quelques connexions supplémentaires nécessaires pour transmettre l'état opérationnel de certains signaux de contrôle appropriés.

ExpressLink Pin	ESP32-C3 GPIO Pin	ESP32-C3-MINI-1-N4-A Module Pin
TX	IO19	27
RX	IO18	26
EVENT	IO10	16
WAKE	IO3	6
RESET	EN	8

Figure 3. Certaines broches GPIO sont nécessaires pour signaler des états opérationnels critiques ou importants (Source : [14])

Comme nous nous concentrons sur l'utilisation d'Arduino et du shield Arduino d'Espressif dans les étapes suivantes, nous n'aborderons pas ces broches ici. Nous les mentionnons surtout parce qu'Espressif souligne leur importance dans la documentation - leur absence limitera les capacités du circuit à une simple communication « Hello World », rendant toute communication pratique avec AWS impossible.

Avant de connecter l'Arduino et la carte Espressif, il est utile de mettre à jour le micrologiciel du module. Nous ne savons pas depuis combien de temps le module a été en stock chez le distributeur et nous ne savons pas si la dernière version du logiciel est préchargée. En théorie, il est possible d'effectuer cette mise à jour à distance, mais cela nécessiterait une connexion préétablie entre le module et les services AWS IoT cloud.

En alternative, Espressif nous propose un script Python qui nécessite Python 3 installé sur un PC. Dans mon cas, un poste de travail fonctionnant sous Ubuntu 20.04 LTS, indique la version comme suit :

```
tamhan@TAMHAN18:~$ python3 --version  
Python 3.8.10
```

Pour établir une communication, vous devez exécuter la commande `pip3 install pyserial==3.5`, qui installe une bibliothèque qui assure le lien entre l'environnement d'exécution Python et les ports série du PC.

Ensuite, vous devrez télécharger l'outil nécessaire au chargement du micrologiciel, disponible sous la forme d'un fichier `.py` sur GitHub [3]. Cherchez le bouton indiqué dans la **figure 4**, qui permet de télécharger directement l'outil sans avoir recours au processus habituel de copier-coller.

Figure 4. Ce bouton permet de gagner du temps.

```

[ 1613.074190] usb 1-1.1: new high-speed USB device number 7 using ehci-pci
[ 1613.191003] usb 1-1.1: New USB device found, idVendor=0403, idProduct=6010, bcdDevice= 7.00
[ 1613.191006] usb 1-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1613.191007] usb 1-1.1: Product: FT2232H-56Q MiniModule
[ 1613.191009] usb 1-1.1: Manufacturer: FTDI
[ 1613.191010] usb 1-1.1: SerialNumber: FTL6Z5PC
[ 1613.235698] usbcore: registered new interface driver ftdi_sio
[ 1613.235776] usbserial: USB Serial support registered for FTDI USB Serial Device
[ 1613.235898] ftdi_sio 1-1.1:1.0: FTDI USB Serial Device converter detected
[ 1613.236007] usb 1-1.1: Detected FT2232H
[ 1613.237935] usb 1-1.1: FTDI USB Serial Device converter now attached to ttyUSB0
[ 1613.238114] ftdi_sio 1-1.1:1.1: FTDI USB Serial Device converter detected
[ 1613.238940] usb 1-1.1: Detected FT2232H
[ 1613.239673] usb 1-1.1: FTDI USB Serial Device converter now attached to ttyUSB1
[ 1676.337202] usb 1-1.4: new full-speed USB device number 8 using ehci-pci
[ 1676.452057] usb 1-1.4: New USB device found, idVendor=10c4, idProduct=ea60, bcdDevice= 1.00
[ 1676.452062] usb 1-1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 1676.452065] usb 1-1.4: Product: CP2102N USB to UART Bridge Controller
[ 1676.452067] usb 1-1.4: Manufacturer: Silicon Labs
[ 1676.452069] usb 1-1.4: SerialNumber: 5a66658d521fed11a7015cb1dcc2a201
[ 1676.452682] cp210x 1-1.4:1.0: cp210x converter detected
[ 1676.455540] usb 1-1.4: cp210x converter now attached to ttyUSB2
tamhan@TAMHAN18:~$ █

```

Figure 5. Les deux adaptateurs USB-série apparaissent dans le système de fichiers /dev du poste de travail.

```

tamhan@TAMHAN18:~/Desktop/Stuff/2024Mar/EAG-ExpressLink$ python3 otw.py /dev/ttyUSB1 v2.5.0.bin
File size 1511424
Uploaded 100.0%
Done...
tamhan@TAMHAN18:~/Desktop/Stuff/2024Mar/EAG-ExpressLink$ █

```

Figure 6. Assurez-vous que le module est à jour. Dans le cas contraire, OTW.py affiche des messages d'erreur.

Vous devrez ensuite télécharger le micrologiciel - pour ce faire, rendez-vous à nouveau sur GitHub [4]. Nous avons réussi à télécharger la version v2.5.0.bin du micrologiciel - assurez-vous simplement de ne pas télécharger le fichier hash par erreur.

La mise à jour du logiciel d'exploitation est impossible via l'interface USB intégrée à la carte car celle-ci se connecte à un « UART d'application » et ne permet pas la programmation de l'ESP32 qui sert ici de module radio. Pour résoudre ce problème, nous avons utilisé un mini-module FTDI, plus précisément le FT2232H MINI MODULE, qui est également efficace pour programmer les micro-contrôleurs ESP32 dans votre labo.

Il convient de noter que la connexion série et l'affectation des broches ne sont pas standards. Pin BD1 est connecté au TX du module ESP32, tandis que Pin BDO est connecté à la broche RX. Les broches TX et RX sont situées au milieu du module et ne sont pas conformes à la disposition habituelle des broches Arduino. Enfin, nous avons besoin d'un fil de connexion pour relier les deux potentiels de masse ensemble. Pour le montage, nous commençons par connecter le module FTDI à l'ordinateur, puis nous connectons l'alimentation de la carte ExpressLink. Cette configuration de connexion est séquencée dans dmesg comme le montre la **figure 5**. Il est important de noter que les ports ttyUSB0 et ttyUSB1 sont responsables de la communication PC-FTDI. Si vous utilisez les broches mentionnées, le chargement de programme s'effectue comme suit :

```

tamhan@TAMHAN18:~/Desktop/Stuff/2024Mar/EAG-ExpressLink$ python3 otw.py /dev/ttyUSB1 v2.5.0.bin

```

Le succès est visible, comme le montre la **figure 6**. Une fois cette étape terminée, nous n'aurons plus besoin d'utiliser le module FTDI. Le module ExpressLink sera connecté uniquement à l'Arduino.

Configuration de la connexion au cloud

Après la mise à jour du micrologiciel du module, nous sommes maintenant prêts à le mettre en service. Dans les étapes suivantes, je suppose que vous avez créé et configuré votre propre compte AWS et que vous vous connectez en tant qu'utilisateur root à l'adresse <https://aws.amazon.com>. Pour des raisons de sécurité, nous n'aborderons pas l'utilisation recommandée d'IAM dans les étapes suivantes. Si vous n'avez pas encore de compte AWS, consultez le guide de démarrage rapide [5]. Veuillez noter que la création d'un compte nécessite une carte de crédit.

Une fois mis à jour avec le dernier micrologiciel, il est possible d'initialiser le module via le cloud ou en envoyant des commandes AT avec des paramètres de configuration spécifiques. Pour les étapes suivantes, vous aurez besoin de la console AWS IoT, disponible sur <http://console.aws.amazon.com/iot>. Assurez-vous de la lancer sur votre poste de travail. Vous pouvez la garder ouverte dans une fenêtre de navigateur.

Dans les étapes suivantes, nous utilisons l'EDI Arduino 2 comme

```

tamhan@TAMHAN18: ~
I (784) wifi_init: tcp mss: 1440
I (784) wifi_init: WiFi IRAM OP enabled
I (794) wifi_init: WiFi RX IRAM OP enabled
I (804) phy_init: phy_version 970,1856f88,May 10 2023,17:44:12
I (844) wifi:mode : sta (dc:54:75:90:4e:44)
I (844) wifi:enable tsf
I (854) PF_WIFI: Station started event
I (854) PF_TIME: Initializing the NTP
I (854) esp_secure_cert: Partition found, current format is cust flash legacy.
I (864) esp_secure_cert: Current partition format: CUST_FLASH_LEGACY, partition name: pre_prov
I (874) esp_pre_init: Device Certificate:
Length: 1318
-----BEGIN CERTIFICATE-----
MIIDnzCCAoegAwIBAgIRAMQdVFip+k0jsu6eqj1obXswDQYJKoZIhvcNAQELBQAw
diELMAkGA1UEBhMC004xCzATU...../OYDV00HDAJTSDEsMCoGA1UE
.../HmS0k
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDnzCCAoegAwIBAgIRAMQdVFip+k0jsu6eqj1obXswDQYJKoZIhvcNAQELBQAw
diELMAkGA1UEBhMC004xCzATU...../OYDV00HDAJTSDEsMCoGA1UE
.../HmS0k
-----END CERTIFICATE-----
I (994) esp_pre_init: Thingname is: 05cdbf4f-ab1f-4bdf-969
I (1004) EL: Starting ExpressLink
I (1004) PF_KVS: Get Failed DefenderPeriod
I (1014) ELEVNT: Setting Event pin high.
I (1014) PF_MAIN: UART task started, input is now processed
I (1014) main_task: Returned from app_main()

```

Figure 7. Le module ExpressLink est communicatif au démarrage.

environnement de développement pour écrire le croquis Arduino - nous ne couvrirons pas ici son installation sous Linux avec un fichier AppImage.

Il est important de noter que pendant cette opération, seul l'Arduino doit être connecté à l'ordinateur - le module est alimenté par l'Arduino lui-même. Si le PC doit communiquer directement avec le module en envoyant des commandes AT, vous devez d'abord déconnecter l'Arduino et utiliser un câble Micro-USB pour relier le module.

Le matériel communiquant avec AWS est représenté sur le serveur sous la forme d'une Thing. Pour l'authentification de notre Thing avec le serveur, un nom (*Thing Name*) et un certificat cryptographique sont requis. Ces deux éléments se trouvent dans la mémoire du module ExpressLink et nous devons maintenant les déterminer. Bien qu'Amazon propose dans sa documentation l'utilisation des commandes `AT+CONF? ThingName` et `AT+CONF? Certificate pem`, en pratique, il est plus simple de connecter directement le module à l'ordinateur, de lancer une application de terminal, puis d'appuyer sur le bouton de réinitialisation. Le micrologiciel implémenté par Espressif fournit dans ce cas les informations nécessaires - comme le montre la **figure 7** - que nous pouvons ensuite copier directement dans le clipboard.

Je préfère utiliser Screen comme application de terminal sous Linux - bien que ce soit un peu difficile, les commandes de configuration de la connexion sont les suivantes :

```
tamhan@TAMHAN18:~$ screen /dev/ttyUSB0 115200
```

Pour quitter Screen, ouvrez une deuxième fenêtre de terminal et entrez la commande suivante (cette méthode permet de conserver le contenu de la boîte de dialogue Screen dans l'autre fenêtre de terminal de sorte que vous pouvez y accéder si nécessaire) :

```
tamhan@TAMHAN18:~$ sudo killall screen -s9
```

Comme prochaine étape dans notre configuration, nous devons retourner à la console AWS IoT mentionnée précédemment. L'URL exacte dépend de la région géographique que vous avez choisi comme « base » pour votre système IdO - sur mon poste de travail, le chemin est le suivant <https://eu-west-1.console.aws.amazon.com/iot/home?region=eu-west-1#/home>.

Create a thing resource

AWS IoT thing resources represent your physical devices in the cloud and maintain a record of their properties and data. Thing resources also let your devices access AWS IoT services. Create a thing resource to register a single device with AWS IoT.

[Create thing](#)

Figure 8. Seule cette option permet de créer de nouvelles Thing.

Cartes de démarrage

Dans cet article, nous avons utilisé le module Espressif pour sa disponibilité. Amazon a signé des contrats avec divers autres partenaires, tels que U-blox, Infineon, Realtek et Telit - vous trouverez une liste actualisée des cartes de démarrage de tous les fabricants sous [15].

La navigation est un peu délicate, car l'interface de navigation de gauche n'est pas toujours complète et peut défiler. Néanmoins, la prochaine étape consiste à cliquer sur le bouton *Link Manage* (*Gérer les liens*) -. Vous devrez alors cliquer sur le bouton *Create Thing* (*Créer une Thing*) dans l'écran qui apparaît ensuite, comme le montre la **figure 8**.

Après tous ces efforts, nous accédons à l'assistant de configuration, qui va maintenant nous guider à travers les trois étapes permettant d'intégrer une nouvelle Thing dans le service AWS IoT.

Dans la première étape, le seul champ à remplir est *Thing Name*, où vous entrerez la chaîne de caractères obtenue plus tôt. Ensuite, il suffit de cliquer sur *Next Step*, où Amazon nous invitera à « télécharger » les fichiers cryptographiques.

Dans la première étape, nous optons pour l'option *Use my certificate* et ensuite nous sélectionnons *CA is not registered with AWS IoT*.

Au fil des ans, le format de fichier *.pem* est devenu un conteneur quasi-standard pour divers types de clés cryptographiques. Pour ceux qui souhaitent en savoir plus, une description détaillée de ce format de fichier est disponible sur [6]. Cependant, nous avons juste besoin de générer un fichier *.pem* valide :

```
tamhan@TAMHAN18:~/Desktop/ExpressLink$ touch tamscert.pem  
tamhan@TAMHAN18:~/Desktop/ExpressLink$ gedit tamscert.pem
```

Après avoir entré la commande *gedit*-, un nouveau fichier s'ouvre dans l'éditeur de texte, dans lequel nous collons le texte du certificat que nous avons copié plus tôt. Il est important d'inclure les délimiteurs **-----BEGIN CERTIFICATE-----** et **-----END CERTIFICATE-----** dans le fichier texte, sans quoi l'analyseur ne pourra pas récupérer les informations du certificat.

Une fois que vous avez sauvegardé le fichier *.pem*, vous pouvez le télécharger lors de la dernière étape. Dans la section *Certificate Status*, sélectionnez l'option *Actif*. Cliquez ensuite sur *Next Step* pour passer à la troisième étape, puis sur *Attach policies to certificate* (Attacher des politiques au certificat).

AWS IoT met en œuvre un système d'autorisation basé sur des certificats. Cela signifie que les certificats sont associés à des documents de politique, qui dictent ensuite les actions que la ressource respective dans le réseau AWS est autorisée à effectuer.

Dans les clusters AWS nouvellement configurés, les politiques prédefinies ne sont plus utilisées depuis un certain temps. Amazon adopte probablement cette approche pour éviter toute responsabilité en cas d'incidents de sécurité causés par des configurations non sécurisées.

Pour cette raison, dans la première étape, nous pouvons cliquer sur le bouton *Create Policy*, qui ouvre une nouvelle fenêtre de navigateur montrant le *Policy Editor*. Dans cette première étape, nous donnons à la politique un nom par lequel elle sera connue par la suite.

En dessous, une deuxième option permet de définir les autorisations accordées par la politique. Ici, nous cliquons sur l'option *JSON* pour activer la saisie directe de texte. Le code déjà présent est généralement prêt à l'emploi - il suffit de veiller à insérer les caractères génériques suivants dans les champs *Action* et *Resource* :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "*",  
      "Resource": "*"  
    }  
  ]  
}
```

La politique présentée permet à l'appareil d'avoir un accès universel à toutes les ressources du backend AWS. Bien entendu, cette approche n'est pas recommandée dans les déploiements pratiques. Il convient de noter que, pour des raisons juridiques, vous devez supprimer ces politiques dès que possible après vos tests. L'éditeur et l'auteur ne peuvent être tenus responsables des conséquences des erreurs commises.

Après avoir cliqué sur l'option *Create*, la fenêtre *Policy Editor* affiche une liste de politiques - il suffit maintenant de fermer la fenêtre et de revenir au processus de génération d'une Thing. La politique nouvellement créée est maintenant prête à être sélectionnée. Ensuite, il est nécessaire de cliquer sur le bouton *Create* pour compléter le processus de saisie dans le backend.

Pour terminer la configuration, il faudra défiler tout en bas dans l'interface de navigation de gauche de la console AWS IoT et cliquer sur l'option *Settings*.

Dans la section *Device Data Endpoints*, vous trouverez une chaîne au format suivant : *a3gw111abcdefgdl-ats.iot.eu-west-1.amazonaws.com*. Vous devez enregistrer cette chaîne, car elle détermine le point d'entrée par lequel le module se connectera au cloud.

Établissement de la liaison Wifi

La configuration correcte du côté du cloud AWS ne représente que la moitié de la tâche, car le module ESP32 doit encore réussir à communiquer avec votre réseau local. Pour ce faire, vous devez entrer vos informations d'identification Wifi dans le *Secure Element*. Dans l'ensemble de commandes AT spécifié par Amazon, vous trouverez les commandes **AT+CONF SSID=<replace-with-your-router-ssid>** et **AT+CONF Passphrase=<replace-with-your-router-passphrase>**, qui nous permettent de transférer les informations d'identification Wifi appropriées via une liaison série.

Espressif a toutefois doté ses modules ExpressLink d'une fonction utile et pratique : des applications Android et iOS, qui permettent une configuration plus « graphique » des modules, sont disponibles sur [6] et [7].

Le seul problème dans ce contexte est que l'exposition de l'interface Bluetooth nécessaire nécessite d'abord la transmission de la commande **AT+CONFMODE**. Pour cela, nous pouvons utiliser le croquis d'exemple fourni par Amazon [8].

Normalement, l'Arduino utilise le port série matériel pour afficher des messages **printf()** et d'autres informations d'état. Après

l'installation d'un module ExpressLink, ce port n'est (logiquement) plus disponible. L'équipe de développement d'AWS s'est donc appuyée sur la bibliothèque *SoftwareSerial*. Au moment de la rédaction de cet article, cette bibliothèque ne fonctionne pas bien avec l'Arduino R4 (voir [10]), mais elle n'est de toute façon pas nécessaire. Notre première tâche consiste donc à nettoyer le croquis en supprimant tous les appels qui y sont liés. Il est très utile que le Renesas Arduino offre deux interfaces UART matérielles utilisant des broches différentes [11]. *Serial1* envoie des données au module radio, tandis que *Serial* peut être utilisé pour la communication via le moniteur série de l'EDI Arduino fonctionnant sur le PC. Cela signifie que la méthode responsable de la communication avec le module ESP32 doit être modifiée avec les commandes suivantes :

```
String execute_command(String command,
    unsigned long timeout_ms) {
    Serial.print("EXC : ");
    Serial.println(command);

    unsigned long saved_timeout_value_ms =
        Serial1.getTimeout();
    Serial1.setTimeout(timeout_ms);
    Serial1.println(command);
    String s = Serial1.readStringUntil('\n');
    s.trim();
    Serial1.setTimeout(saved_timeout_value_ms);
    return s;
}
```

Veillez à modifier la configuration conformément aux définitions suivantes avant d'exécuter le programme :

```
#define EVENT_PIN 2
#define RESET_PIN 4

#define MY_AWS_IOT_ENDPOINT "change_me-ats.iot.eu-west-1.
    amazonaws.com"
```

En raison d'un conflit rapporté par l'auteur entre les parties Renesas et Arduino de la bibliothèque, comme documenté sous [12], une modification de la fonction *process_event()* est nécessaire. Spécifiquement, la valeur renournée par *c_str()* ne doit pas être passée directement à *scanf()* :

```
event_t process_event()
{
    String response;

    int val = 0;
    event_t event_number = EVENT_NONE;
    val = digitalRead(EVENT_PIN);
    if(val)
    {
        response = execute_command("AT+EVENT?",
            3000);
```

```
        if (response.equals("OK"))
        {
            return EVENT_NONE;
        }
        else {
            char ok_string[3];
            int topic_index;
            int total_read;
            char someResponse[300];
            strcpy(someResponse, response.c_str());
            total_read = sscanf(someResponse,
                "%s %d %d %*s", ok_string,
                &event_number, &topic_index);
            return event_number;
        }
    }
}
```

Le module ESP32 est connecté à l'Arduino, alors que seul l'Arduino utilise la liaison USB. L'alimentation électrique du kit ExpressLink doit être éteinte. Une double pression rapide sur le bouton RST de l'Arduino UNO R4 est nécessaire pour permettre au bootloader d'accepter de nouvelles compilations.

Après avoir modifié et flashé avec succès le croquis de test [13], le module reçoit la commande nécessaire s'il n'est pas déjà dans un état provisionné. Vous pouvez vérifier ceci en essayant de relire le SSID stocké dans le *Secure Element* :

```
void loop() {
    event_t event = EVENT_NONE;
    String response;
    static state_t state = STATE_INIT;
    if (state != STATE_INIT) {
        event = process_event();
    }
    switch (state) {
        ...
        case STATE_EL_READY:
            response = execute_command(
                "AT+CONF Endpoint=\"MY_AWS_IOT_ENDPOINT\"", 3000);
            response = execute_command(
                "AT+CONF Topic1=TEST", 3000);
            response = execute_command(
                "AT+CONF? SSID", 3000);
            state = process_ssid(response);
            break;
        case STATE_UNPROVISIONED:
            response = execute_command(
                "AT+CONFMODE", 5000);
            if (response.equals("OK CONFMODE ENABLED"))
            {
                state = STATE_PROVISIONING;
            }
            break;
```

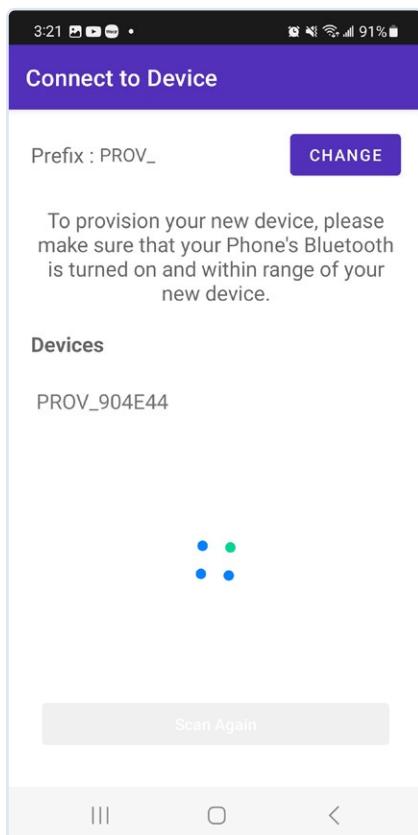


Figure 9. Le scanner Bluetooth LE a détecté une nouvelle cible !

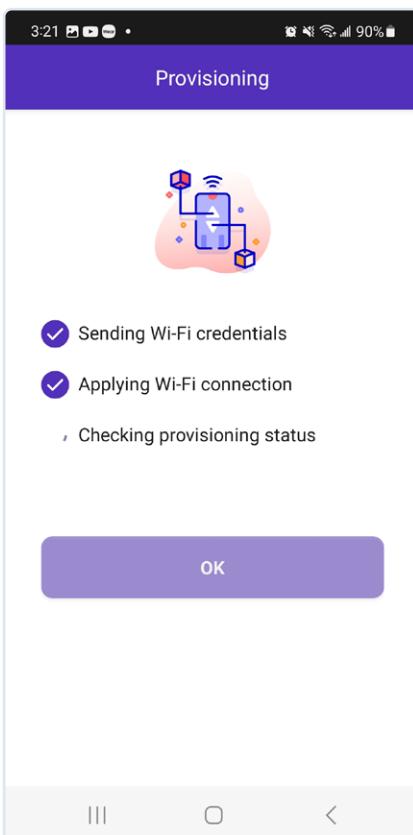


Figure 10. L'approvisionnement peut prendre jusqu'à 2 minutes...

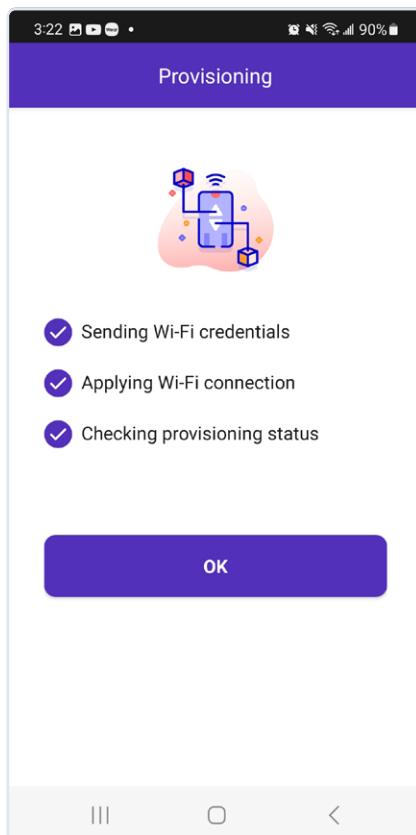


Figure 11. ...et n'est terminé que lorsque la fenêtre ci-contre apparaît.

Amazon met en œuvre le programme présenté ici sous la forme d'une machine à états colossale. La méthode responsable du traitement du SSID définit ensuite l'état correct comme suit, si le SSID renvoyé est inutilisable :

```
state_t process_ssid(String response)
{
    int event_number = 0;
    char ok_string[3];
    char ssid_string[33] = {0};
    int total_read;

    total_read = sscanf(response.c_str(),
        "%s %s" , ok_string, ssid_string);

    if(total_read > 1) {
        return STATE_PROVISIONED;
    }
    else {
        return STATE_UNPROVISIONED;
    }
}
```

Une fois que le module atteint le statut de provisionnement, il devient visible pour un scanner Bluetooth LE. Une méthode plus pratique consiste à utiliser l'application pour smartphone mention-

née précédemment. En raison des particularités du système de permissions d'Android, le système d'exploitation vous demandera de nombreuses autorisations lors du premier lancement. Si vous les accordez, vous accéderez à l'interface de l'appareil photo - Espresif utilise la même application non seulement pour le provisionnement Bluetooth LE, mais aussi pour capturer les informations des modules ExpressLink qui fournissent leur nom de module via un code QR.

La **figure 9** montre comment le module est détecté. L'exécution proprement dite du programme est ensuite réalisée par un assistant qui vous invite à saisir le mot de passe après avoir sélectionné le nom du Wifi. Notez que la transition entre les captures d'écran présentées dans les **figures 10** et **11** peut prendre un peu de temps.

Conclusion

Dans cet article, nous avons principalement démontré les étapes de configuration d'un module AWS IoT ExpressLink. Dans la deuxième partie de cette série, nous approfondirons la transmission des données. Les avantages de cette configuration particulière du système devraient déjà être évidents. Avec un investissement matériel aussi minime, il serait autrement assez difficile de se connecter à un grand fournisseur de services cloud. Le coût supplémentaire des modules radio peut être justifié par les promotions occasionnelles offertes en collaboration avec Amazon, ce qui peut être intéressant. ↗

240240-04

À propos de l'auteur

Avec plus de 20 ans d'expérience, Tam Hanna est un ingénieur spécialisé dans l'électronique, l'informatique et les logiciels. Il est designer indépendant, auteur de plusieurs ouvrages et journaliste — instagram.com/tam.hanna. Durant ses moments libres, il se consacre à la conception et à la production de solutions imprimées en 3D. Il nourrit également une passion pour le commerce et la dégustation de cigares haut de gamme.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (tamhan@tamoggemon.com) ou contactez Elektor (redaction@elektor.fr).



Produits

- Arduino Uno R4 WiFi
www.elektor.fr/20528



LIENS

- [1] Documentation d'ExpressLink par Espressif : <https://espressif.com/en/solutions/device-connectivity/esp-aws-iot-expresslink>
- [2] Distributeurs de l'ESP32-C3-AWS-ExpressLink-DevKit : <https://oemsecrets.com/compare/ESP32-C3-AWS-ExpressLink-DevKit>
- [3] Télécharger l'outil : <https://github.com/espressif/esp-aws-expresslink-eval/blob/main/tools/otw.py>
- [4] Micrlogiciel du module : <https://github.com/espressif/esp-aws-expresslink-eval/releases>
- [5] AWS getting started guide : <https://tinyurl.com/elgsg-set-up>
- [6] Description du format de fichier .pem : <https://tinyurl.com/PEM-file>
- [7] Outil de configuration pour Android : <https://play.google.com/store/apps/details?id=com.espressif.provble>
- [8] Outil de configuration pour iOS : <https://apps.apple.com/app/esp-ble-provisioning/id1473590141>
- [9] Croquis Arduino pour l'activation du Bluetooth : <https://tinyurl.com/sketches-arduino>
- [10] Problèmes liés à l'UART logiciel : <https://github.com/arduino/uno-r4-library-compatibility/issues/12>
- [11] Two Hardware UARTs : <https://forum.arduino.cc/t/uno-r4-and-serial-rx-tx/1146022/2>
- [12] Questions concernant les bibliothèques : <https://github.com/espressif/esp-aws-expresslink-eval/issues/23>
- [13] Croquis de l'auteur : <https://elektormagazine.fr/240240-0>
- [14] Getting Started Guide ExpressLink Evaluation Kit : <https://github.com/espressif/esp-aws-expresslink-eval>
- [15] ExpressLink Starter Boards : <https://tinyurl.com/ExpressLink-Starter>

Rejoignez notre communauté



www.elektormagazine.fr/community

elektor
design > share > earn

détecteur de flux d'air

Arduino

Source : Adobe Stock

aucun capteur externe n'est nécessaire !



Raymond Schouten (Pays-Bas)

Ce circuit de détection utilise uniquement l'Arduino Nano, sans capteur supplémentaire. Il permet de détecter les flux d'air provenant d'une distance de 30 cm, en mesurant la température interne de la puce chauffée avec une résolution de 0,01°C et en détectant les variations rapides. Ce circuit utilise une technique de dithering, permettant d'augmenter la résolution d'un CAN au-delà de celle définie par le LSB.

La détection de flux d'air dans ce circuit repose sur un principe relativement simple. Lorsqu'il est en fonction, le contrôleur ATmega328P de l'Arduino Nano chauffe à environ 6...10°C supérieure de la température ambiante. Une augmentation du flux d'air autour de cette puce entraîne son refroidissement : c'est ainsi détection qu'un flux d'air soudain autour de la carte Arduino Nano est détecté.

Cette détection se fait en comparant la nouvelle température mesurée avec celle relevée précédemment, deux fois par seconde. Outre la détection du flux d'air, le système est capable d'identifier d'autres variations de température, tels que celles dues au déplacement de la

carte ou au contact avec le boîtier de la puce (**figure 1**). Veuillez noter que :

- Pour que l'Arduino fonctionne en tant que détecteur, après un boot à froid, il faut lui laisser le temps de se réchauffer (5...10 min). Une fois qu'il a atteint une température « stable », il devient possible de détecter ces flux d'air soudains.
- Ce détecteur est très sensible – les fluctuations de lecture peuvent être <0,02°C - mais un flux d'air provoque des fluctuations plus élevées. Pour évaluer la stabilité de lecture, vous pouvez protéger le détecteur des flux d'air, par exemple en le plaçant dans un sac en plastique. Voir la **figure 2**.

Le concept de lecture de la température

Le CAN interne de la puce peut non seulement lire ses broches d'entrée analogique, mais peut également être configuré pour lire le capteur de température intégré à la puce. Cette tension sera toujours <1 V. Pour augmenter la sensibilité du CAN, la tension de référence est réglée (via le logiciel) à 1,1 V au lieu de l'alimentation standard de 5 V. Avec un CAN de 10 bits (1 024 pas), le plus petit pas mesurable passe alors de 5 mV à 1 mV (chiffres arrondis).

Ce plus petit pas est exprimé par le LSB (bit de poids faible). Selon la fiche technique, la sortie du capteur est de 1 mV/°C, ce qui signifie qu'avec un pas de 1 mV, nous obtenons une résolution de mesure de 1°C - une valeur trop imprécise pour nos besoins. Le paragraphe suivant explique comment améliorer cette résolution à environ 0,01°C/pas.

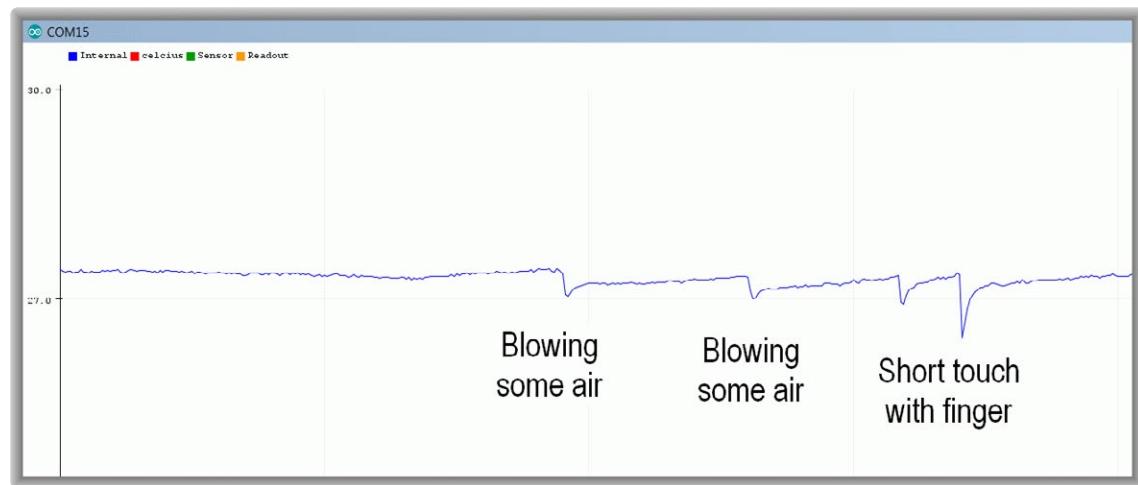


Figure 1. La réponse rapide aux relevés de température du capteur thermique intégré au contrôleur ATmega328 de la carte Arduino Nano.

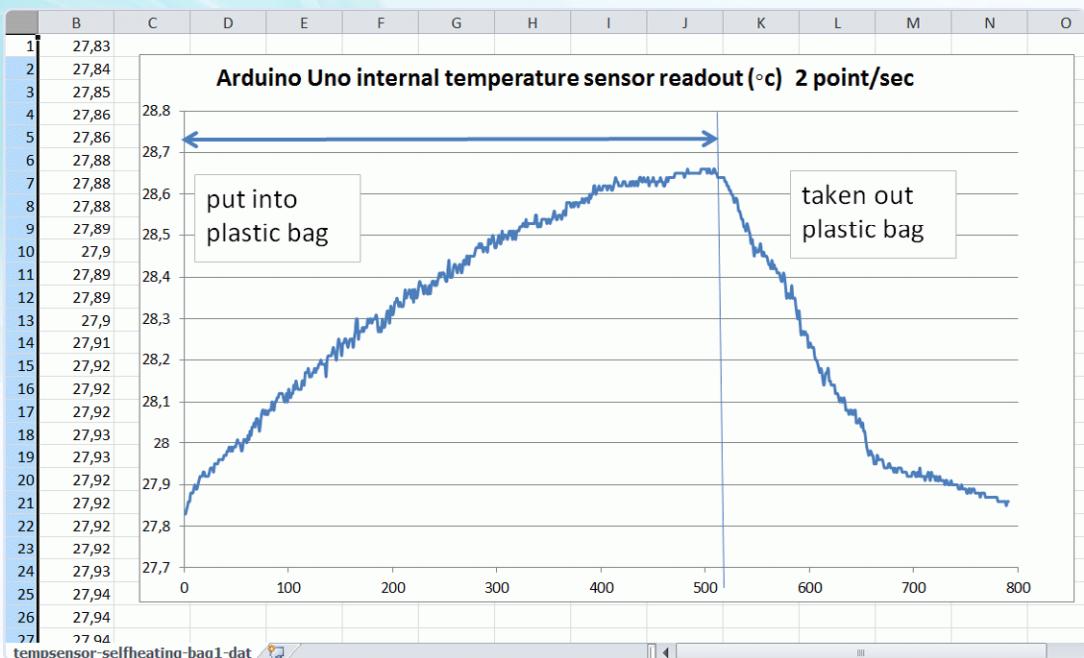


Figure 2. Pour évaluer la stabilité de la lecture, il est conseillé de placer la carte dans un sac en plastique, afin de la protéger de tout changement de flux d'air indésirable.

Augmentation de la résolution de lecture du CAN par dithering

Comme mentionné précédemment, un CAN est généralement limité à la détection des variations (pas) équivalentes à un LSB. Ayant réduit la tension de référence à 1,1 V, chaque LSB correspond maintenant à environ 1 mV.

Le CAN donne alors une lecture échelonnée en multiples de 1 mV. Si. Par exemple, si la tension d'entrée est entre 99,5 mV et 100,5 mV, il donnera toujours des lectures de « 100 mV ». Ainsi, si nous avons un signal d'entrée de 100,1 mV, la lecture indiquera simplement « 100 mV ». Ici, le concept de dithering s'avère utile. Cette technique est l'une des rares applications où la présence de bruit peut être une bénédiction. Pour mieux comprendre le processus du dithering, examinons un exemple pour notre cas spécifique, comme le montre la **figure 3**.

Si nous ajoutons un bruit aléatoire au signal de 100,1 mV (supposons que nous ajoutons un bruit d'une amplitude 1 LSB = 1 mV) et effectuons de nombreuses mesures rapides et répétées, nous obtiendrons environ 90 % de lectures à « 100 mV » et 10 % de lectures à « 101 mV ». En calculant la moyenne, nous pouvons déduire que le signal d'entrée est en fait de 100,1 mV. Pour des bruits d'amplitudes plus élevées, la moyenne des mesures converge également vers cette valeur.

Les fluctuations causées par le bruit rendent ce résultat légèrement instable, mais, en général, le calcul de la moyenne de 100 échantillons permet une amélioration de 10x. L'amélioration correspond à la racine carrée du nombre d'échantillons. Dans cette application, la moyenne est calculée sur 6 400 échantillons pour chaque mesure de température. La racine carrée de ce chiffre permet de prévoir une amélioration de 80 fois.

La résolution initiale de notre mesure était de 1°C, de sorte que nous pourrions théoriquement améliorer la résolution à 0,0125°C. Les données de test visibles à droite de la **figure 4** ont tendance à fluctuer de 0,01...0,02°C, ce qui démontre que nous nous en sommes approchés de cette valeur dans la pratique. Vous pouvez télécharger le fichier de données sur la page de ce projet sur Elektor Labs [1] et consulter les résultats dans la vidéo de démo [2].

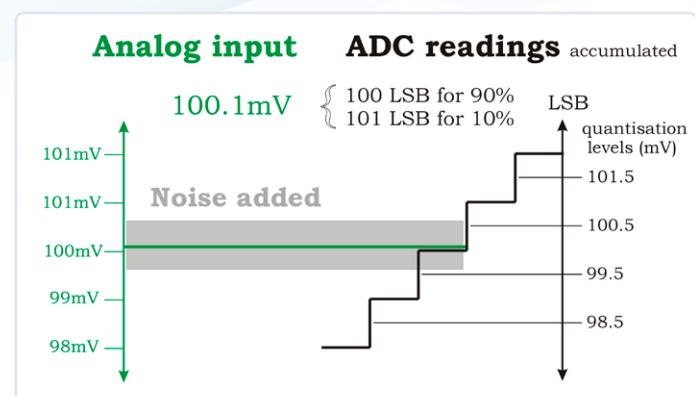


Figure 3. Les relevés de la composante bruit sont ajoutés à ceux du signal primaire, puis une moyenne est calculée à partir de ces valeurs.



Figure 4. La carte Arduino Nano utilisée dans ce projet, avec les lectures indiquées à droite.

Précision et exactitude

Il est important de noter que les lectures se caractérisent par une grande précision (fluctuations <0,02°C) mais elles ne sont pas très exactes à ce niveau, ce qui signifie que les lectures à faible fluctuation (précision) peuvent être décalées de plusieurs degrés par rapport à la valeur réelle (manque d'exactitude), comme illustré dans la **figure 5**. Pour les applications détectant des variations relatives de la température, sans nécessiter une précision absolue, une grande précision est suffisante.

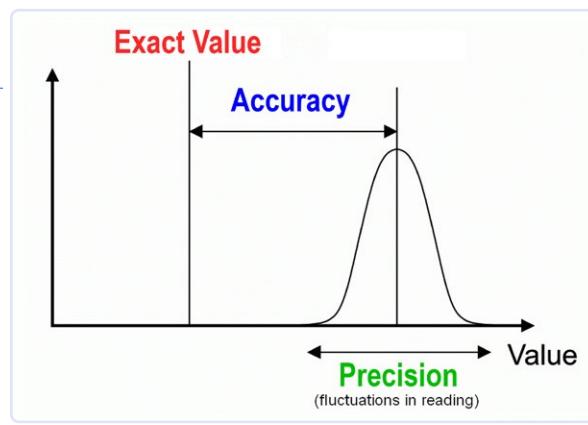


Figure 5. Relation entre la précision et l'exactitude (Source : Wikimedia Commons, Pekaje / Anthony Cutler, https://commons.wikimedia.org/wiki/File:Accuracy_and_precision.svg)

Logiciel

Nous avons utilisé un Arduino Nano, mais vous pouvez utiliser d'autres cartes avec le même contrôleur ATmega328P (Arduino UNO ou Arduino Pro Mini). Le code source de ce projet est disponible sur [1], et les étapes sont plutôt simples :

1. Initialiser en définissant la tension de référence du CAN et l'entrée du multiplexeur de manière appropriée pour la lecture du capteur de température interne.
2. Accumuler 6 400 lectures du CAN afin de déterminer la température avec une haute précision.
3. Envoyer le résultat via USB.
4. Vérifier si la lecture a chuté soudainement. Si c'est le cas, éteindre le voyant indicateur ; sinon, l'allumer.
5. Passer à la phase 2.

Limites

- Ce projet fonctionne mieux avec les cartes Arduino équipées de la version CMS du microcontrôleur. Le boîtier DIL, plus volumineux, résiste plus longtemps aux variations de température, a une meilleure isolation thermique et n'est donc pas adapté à cette application.
- Le processus de dithering ralentit la vitesse de mesure, car nous devons collecter de nombreuses lectures. Pour cela, nous utilisons un CAN de 10 kHz, ralenti à environ deux lectures par seconde, ce qui est suffisant pour les mesures de température.
- Un CAN n'est pas parfait, et peut présenter des inégalités dans ses pas les plus fins, ce qui peut entraîner des erreurs supplémentaires dans les résultats. Pour en savoir plus sur la DNL (non-linéarité différentielle), consultez [3].
- Pour les utilisateurs avancés : un bruit d'amplitude supérieure à 1 LSB contribue également à atténuer les erreurs de DNL du CAN.
- Si le bruit n'est pas uniformément réparti parmi toutes les valeurs mesurables, cela peut affecter les résultats.

Comment avons-nous introduit la source de bruit nécessaire au dithering ?

En réalité, nous n'avons pas eu besoin de le faire. À ces faibles niveaux de LSB de 1 mV, le bruit est naturellement présent dans les composants du circuit, y compris le capteur lui-même. À titre de comparaison : lors de la lecture d'une tension d'entrée en utilisant la tension de référence de 5 V du CAN, j'ai constaté que le bruit ambiant « naturel » était trop faible pour que le dithering soit efficace.

Comment en suis-je arrivé à cette conclusion ? La plupart des lectures restaient centrées autour des pas de 5 mV LSB. Je compte démontrer comment résoudre cela pour réaliser un voltmètre à haute résolution dans un futur article.

Quelques idées (non testées)

Utilisation comme thermomètre

En réduisant la dissipation thermique moyenne du contrôleur, vous pouvez utiliser ce circuit comme thermomètre. Pour cela, mettez le contrôleur en mode veille et réactivez-le toutes les quelques secondes pour effectuer des mesures.

Lecture des capteurs externes

Vous pourriez connecter des diodes externes aux broches d'entrée analogique, en utilisant une faible résistance de tirage vers le haut pour la polariser avec un faible courant. Ainsi, vous pourriez mesurer plusieurs températures externes avec une grande précision (mais une faible exactitude) en utilisant le même concept de code. ↗

220615-04



À propos de l'auteur

Outre son emploi principal où il développe des instruments électroniques à faible bruit, Raymond Schouten s'adonne à des projets personnels en concevant de petits synthétiseurs musicaux et d'autres circuits compacts. La plupart de ses projets visent à obtenir des résultats optimaux avec le matériel le plus simple. Il partage souvent ses projets sur Elektor Labs, YouTube et son site web personnel rs-elc.nl.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (rs.elc.projects@gmail.com) ou contactez Elektor (redaction@elektor.fr).



Produits

- **Arduino Nano**
www.elektor.fr/17002

- **Livre « Kickstart to Arduino Nano », Ashwin Pajankar (Elektor, 2022)**
version papier : www.elektor.fr/20241
Version numérique : www.elektor.fr/20242

LIENS

- [1] Page du projet sur Elektor Labs : <https://tinyurl.com/airflowdetector>
- [2] Vidéo de démonstration sur YouTube : <https://youtu.be/0p6hssAf7Xs>
- [3] Short wiki-note about DNL : https://en.wikipedia.org/wiki/Differential_nonlinearity

détecteur de fuite d'eau

connecté à l'Arduino Cloud

Source : Adobe Stock



Figure 1. Mon compteur de consommation d'eau avec le capteur de débit attaché.

Yves Bourdon (France)

J'ai subi un préjudice de plusieurs milliers d'euros lorsqu'une conduite d'eau sous ma maison s'est fissurée sans que je m'en aperçoive. J'ai alors décidé de créer et d'installer un système de surveillance de la consommation d'eau afin d'éviter que des incidents aussi coûteux ne passent inaperçus à l'avenir.

Grâce à l'Arduino Cloud, je peux désormais surveiller ma consommation d'eau sur mon téléphone portable à tout moment de la journée et depuis n'importe où dans le monde. Voici comment j'y suis parvenu.

L'eau est précieuse et devient de plus en plus chère. Il y a un an, une conduite d'eau s'est fissurée sous ma maison sans que je m'en aperçoive immédiatement, avec à la clé une facture d'eau de plusieurs milliers d'euros ! J'ai donc cherché un moyen d'être prévenu au cas où cela se reproduirait. Le service des eaux a installé un compteur intégrant un capteur de débit (**figure 1**) et un petit afficheur à distance sans fil.

Malheureusement, la fiabilité n'était pas au rendez-vous, et il n'y avait aucun moyen de m'alerter, si ce n'est un « F » sur l'écran censé indiquer une fuite...

Un article d'Elektor présentait un détecteur de fuite d'eau développé par Denis Lafourcade [1]. Avec son aimable autorisation, j'ai décidé de reprendre son projet en le modifiant assez largement pour l'intégrer à mon système existant. De plus, j'ai voulu utiliser l'Arduino Cloud présenté dans l'édition 2022 d'Elektor Arduino Guest Edition [2].

Conditions préalables

Je voulais pouvoir accéder à mes données sur mon téléphone portable, et l'environnement de l'Arduino Cloud est particulièrement adapté à cet effet. Cela me permettait d'avoir un tableau de bord sur mon téléphone sans avoir à développer une application spécifique qui nécessite des mises à jour constantes. Soucieux de la sécurité, je ne voulais surtout pas ouvrir un port sur mon routeur pour accéder à mes données. Enfin, en cas de fuite, je souhaitais être alerté par une notification push accompagnée d'une alarme sonore.

Spécifications

- Utilisation de l'Arduino Cloud pour collecter des données et afficher des statistiques. Un graphique montre la consommation d'eau quotidienne en temps réel. Jusqu'à trois mois de données sont stockés dans le nuage. En outre, il est facile de télécharger un fichier CSV (valeurs séparées par des virgules) pour traiter les données dans un tableur, par exemple.

- Utilisation d'un processeur ESP32, et plus particulièrement de la carte de développement Heltec WiFi Kit 32 [3], qui comprend notamment un écran OLED et un chargeur de batterie.
- Connexion directe (via un optocoupleur pour éliminer les interférences) au capteur Reed intégré à mon compteur d'eau situé à environ 40 mètres, contre le mur d'enceinte de ma propriété. Notez que si vous n'avez pas de compteur équipé d'un capteur, vous pouvez facilement trouver des adaptateurs de relais Reed pour n'importe quel compteur existant (**figure 2**).
- Le système mesure le débit d'eau en litres par unité de temps au moyen d'interruptions en provenance de mon compteur.
- Chaque heure, la consommation d'eau en litres est enregistrée dans un tableau.
- Chaque jour à minuit, la valeur est enregistrée pour mesurer la consommation journalière (consommation à l'instant t moins la consommation à minuit précédent).
- Chaque année, le 31 décembre à minuit, la valeur de la consommation est également enregistrée. Cela permet de calculer facilement la consommation annuelle (consommation à l'instant t moins la consommation au 1er janvier).
- Les mesures sont enregistrées par tranches de 24 heures. Cela permet de vérifier la consommation d'eau même dans des tranches horaires inhabituelles. Par exemple, mon adoucisseur d'eau nettoie sa résine à 2h15 du matin tous les 15 jours.
- Toutes les 15 minutes, la consommation des dernières 24 heures est calculée. Si elle dépasse le seuil d'alarme (750 l / jour dans mon cas), une alarme sonore se déclenche, un indicateur virtuel sur le tableau de bord Arduino s'allume, et un email (ou une

notification push) est envoyé via l'Arduino Cloud. Tout cela pour s'assurer d'être alerté à temps !

- De même, la présence de deux tranches horaires successives sans consommation d'eau (souvent le cas la nuit) est vérifiée. Si cette condition n'est pas remplie, une fuite d'eau peut être suspectée et des alertes sont envoyées.
- La seule opération de maintenance nécessaire est la saisie périodique de la valeur réelle affichée par le compteur d'eau. La différence entre les mesures effectuées par le relais Reed et la valeur réelle affichée par le compteur est très faible. Je le réajuste environ tous les deux mois.

Se connecter à l'Arduino Cloud

On peut se passer de l'EDI classique d'Arduino sur son ordinateur lorsqu'on travaille avec l'Arduino Cloud [4]. Le compilateur en ligne est suffisant pour développer confortablement une application, et les mises à jour successives et transparentes améliorent considérablement l'environnement de développement. La mise à jour OTA (Over-the-Air) étant native dans cet environnement, on peut choisir de télécharger le logiciel via un port USB (obligatoire pour le premier téléchargement) ou à distance, ce qui est très pratique.

Il existe de nombreux tutoriels sur internet pour apprendre à maîtriser le cloud, mais dans l'ensemble, ce n'est pas si compliqué. Tout est gratuit si on ne dépasse pas cinq variables à rafraîchir sur l'application. C'est peu, mais suffisant pour le détecteur de fuite d'eau si l'on se contente d'un affichage limité : consommation journalière, deux tableaux de tranches horaires, consommation annuelle et valeur du compteur. La version gratuite est limitée à deux applications et il existe des

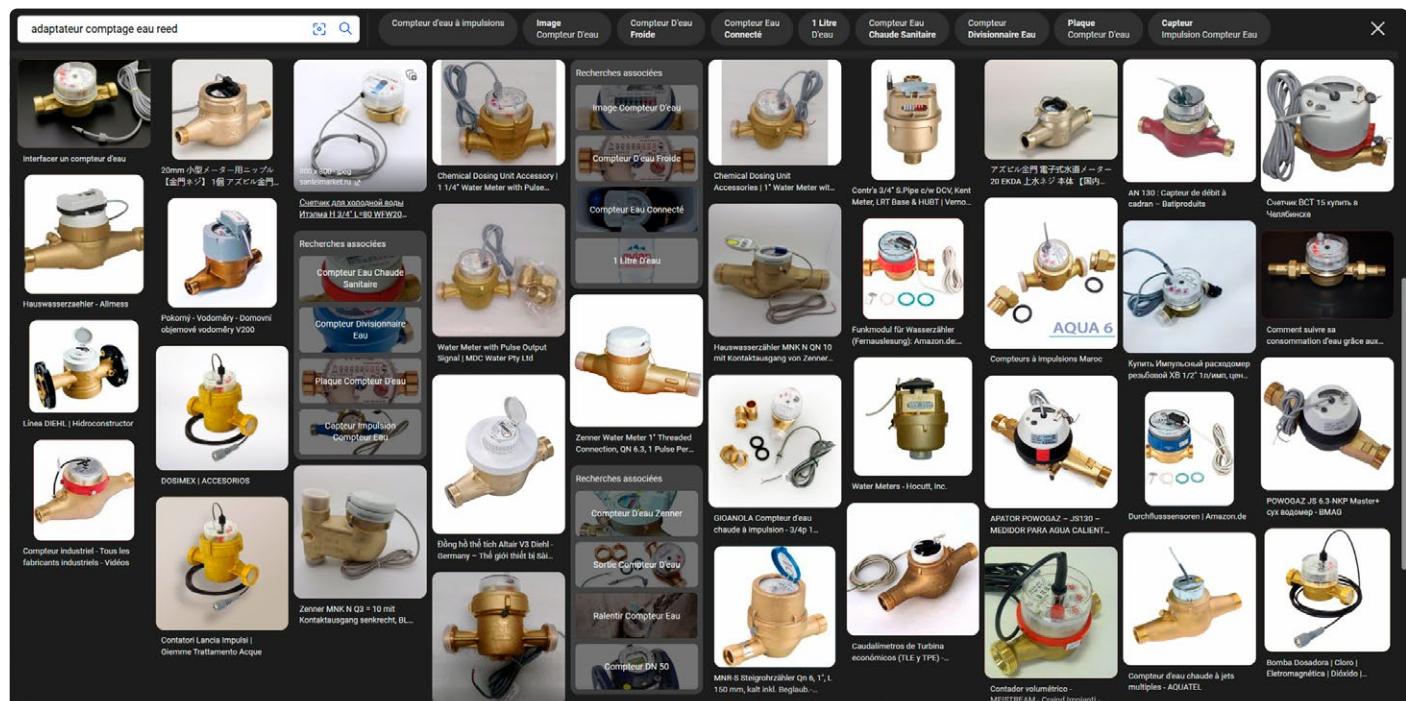


Figure 2. On trouve facilement des capteurs Reed pour tous les types de compteurs.



The screenshot shows the Arduino Cloud interface. On the left, under 'Cloud Variables', there is a table listing various variables with their names, types, last values, and last update times. Some variables include 'affCons1' (String), 'alarmLED1' (bool), and 'valActuelle' (float). On the right, under 'Associated Device', it shows a device named 'Heltec_Eau' with its ID, type (Heltec WiFi Kit 32), and status (Online). Below that, under 'Network', it shows the Wi-Fi name 'Home_IoT' and secret key.

Figure 3. Étape 2 - configuration d'une Thing.

```

const char DEVICE_LOGIN_NAME[] = "b4101540-7f5a-4c33-a20e-a24cc0867106";
const char $SSID[] = SECRET_SSID; // Network SSID (name)
const char PASS[] = SECRET_OPTIONAL_PASS; // Network password (use for WPA, or use as key for WEP)
const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // Secret device password

void onReleveChange();
void onValActuelleChange();
void onSeuilAlarmeChange();
void onRazConsChange();
void onRazStatsChange();
void onRebootChange();

String affCons1;
String affCons2;
String consGloss;
String consolAjour;
String displayHeure;
String jourCons;
int jourCons;
int nbDisconnect;
int nbReboot;
int razCons;
bool razCons;
bool razStats;
bool reboot;
float releve;
float valActuelle;
int seuilAlarme;
String startRun;
String versionSoft;
float valactuelle;
bool alarmE01;
bool alarmE02;
bool alarmE03;
bool razStats;
bool reboot;
int seuilalarme;
bool alarmLED1;
bool alarmLED2;
bool razCons;
bool razStats;
bool reboot;
String versionsoft;
void initProperties(){
    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);
    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);
    ArduinoCloud.addProperty(affCons1, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(affCons2, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(consGloss, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(consolAjour, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(displayHeure, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(jourCons, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(nbDisconnect, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(nbReboot, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(releve, READWRITE, ON_CHANGE, onReleveChange);
    ArduinoCloud.addProperty(valActuelle, READWRITE, ON_CHANGE, onValActuelleChange);
    ArduinoCloud.addProperty(jourCons, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(nbDisconnect, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(nbReboot, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(seuilAlarme, READWRITE, ON_CHANGE, onSeuilAlarmeChange);
    ArduinoCloud.addProperty(alarmE01, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(alarmE02, READ, ON_CHANGE, NULL);
    ArduinoCloud.addProperty(alarmE03, READWRITE, ON_CHANGE, onRazConsChange);
    ArduinoCloud.addProperty(razCons, READWRITE, ON_CHANGE, onRazStatsChange);
    ArduinoCloud.addProperty(reboot, READWRITE, ON_CHANGE, onRebootChange);
}
WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);

```

Figure 4. Voici ce que vous devriez obtenir après avoir défini toutes les variables.

offres spéciales. Pour ma part, j'ai souscrit un abonnement Maker pour 5,99 € par mois car j'utilise cette application pour de nombreux projets. Avec mon abonnement Maker, mes données sont conservées pendant trois mois, et je peux télécharger un fichier CSV contenant les données horodatées.

Si vous souhaitez bénéficier de toutes les fonctionnalités que j'ai développées (y compris les statistiques sur le fonctionnement et les commandes de connexion avec réinitialisation), je vous recommande de souscrire au moins un mois au plan Maker pour l'essayer.

J'ai optimisé la fiabilité de la connexion comme suit :

- Si la connexion Wi-Fi échoue, le système le détecte et tente de se reconnecter.
- Si la connexion à l'Arduino Cloud est perdue (ce qui peut arriver), le programme le détecte et, après plusieurs tentatives de reconnexion, l'unité centrale est redémarrée.

J'ai également géré l'heure locale comme suit :

- Lorsque le système est connecté à l'Arduino Cloud, il récupère périodiquement l'heure UTC du nuage.
- Comme j'ai besoin de l'heure locale en tenant compte des changements d'heure d'été, l'heure locale de l'ESP32 est mise à jour en tenant compte de ces paramètres.

Étape 1 : Ajouter un dispositif

Cette étape prépare votre système à l'environnement cloud en lui attribuant un Device ID et une Secret Key. Commencez par sélectionner *third-party device* dans le menu Setup Device, puis *ESP32*, et enfin *Heltec WiFi Kit 32* dans la liste déroulante. Regardez bien, car la liste n'est peut-être pas triée par ordre alphabétique, mais ça y est. Donnez-lui un nom. Le nuage vous fournit alors les deux variables qui doivent être sauvegardées correctement en utilisant le PDF téléchargeable.

Étape 2 : Configurer une Thing

Il faut ensuite associer les variables que l'on veut afficher ou modifier (attention à respecter la casse !), voir la **figure 3**.

Vous devez définir les types *bool*, *int*, *float*, *string* lors de la création, ainsi que l'attribut *Read & Write*. L'attribut *Read & Write* correspond à une variable qui peut être modifiée dans l'application mobile, comme le seuil d'alarme. L'attribut *Read Only* (lecture seule) concerne les valeurs qui doivent uniquement être affichées, comme l'état de l'alarme. Conservez *Variable update Policy – On change* pour toutes les variables.

Vous devez également indiquer le SSID et le mot de passe du menu *Network*, ainsi que la *Secret Key* précédemment récupérée. La **figure 4** montre le résultat de cette étape.

Étape 3 : Le croquis

Allez dans l'onglet *Sketch* et choisissez *Open full editor*. Vous êtes maintenant dans l'environnement de développement.

Chargez les bibliothèques nécessaires à ce projet : *SSD1306.h*, *OLEDDisplayUi.h*, *TimeLib.h*, *math.h*, et *Preferences.h*. Je les ai placées dans le dossier *Libraries*. Il suffit de les télécharger en utilisant la flèche dans l'onglet *Libraries*.

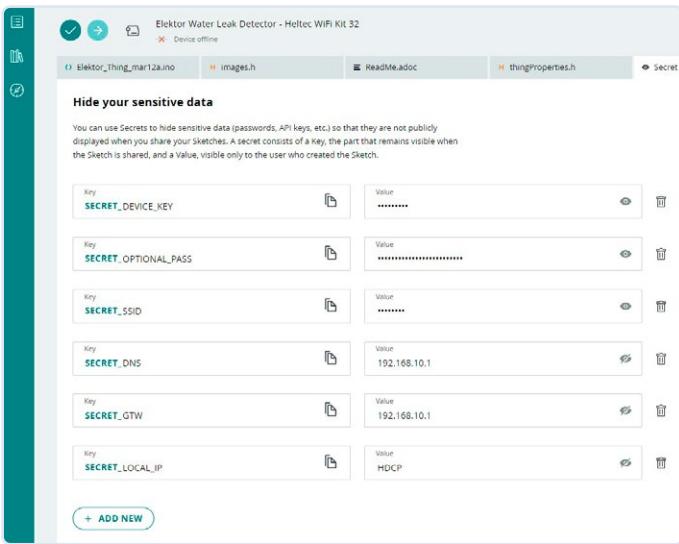


Figure 5. Complétez la page de l'onglet Secret ainsi.

Téléchargez le code source depuis la page de ce projet sur Elektor Labs [5]. Ouvrez le fichier *Elektor_Thing_mar12a.ino* dans l'EDI Arduino classique ou dans un éditeur de texte pour copier facilement le programme dans l'EDI Arduino Cloud. Il suffit de copier et coller le code source dans le premier onglet.

Créez l'onglet *images.h* en utilisant le + de la barre principale, et copiez-collez le contenu de mon fichier. Vous devez également compléter l'onglet *Secret*, comme le montre la **figure 5**.

Compilez le programme en utilisant le bouton correspondant. Si tout est correct, vous pouvez télécharger le programme sur votre carte (après avoir installé Cloud Agent, comme le suggère l'EDI). La compilation est très rapide - rien à voir avec l'EDI classique.

Note : J'ai regroupé plusieurs paramètres dans l'onglet *Secret*. En plus des paramètres par défaut **SECRET_SSID**, **SECRET_PASSWORD**, et **SECRET_DEVICE_KEY**, vous pouvez définir des paramètres réseau (IP,

Un bogue ?

J'ai récemment remarqué un problème sur mon iPhone, mais pas sur ma tablette. Dans le tableau de bord Arduino, il est très compliqué d'entrer une nouvelle valeur pour le compteur d'eau, la dernière lecture du 1^{er} janvier ou le seuil d'alarme. Pour une raison quelconque, la nouvelle valeur est immédiatement effacée avant même que vous n'ayez la possibilité de toucher le bouton Terminé.

Pour contourner ce problème, apparu après une mise à jour d'Arduino Cloud, il suffit d'appuyer sur le bouton de redémarrage, d'entrer la nouvelle valeur et d'attendre que le système redémarre. La nouvelle valeur sera alors prise en compte.

Il se peut que ce problème ait été résolu au moment où vous lirez cet article.

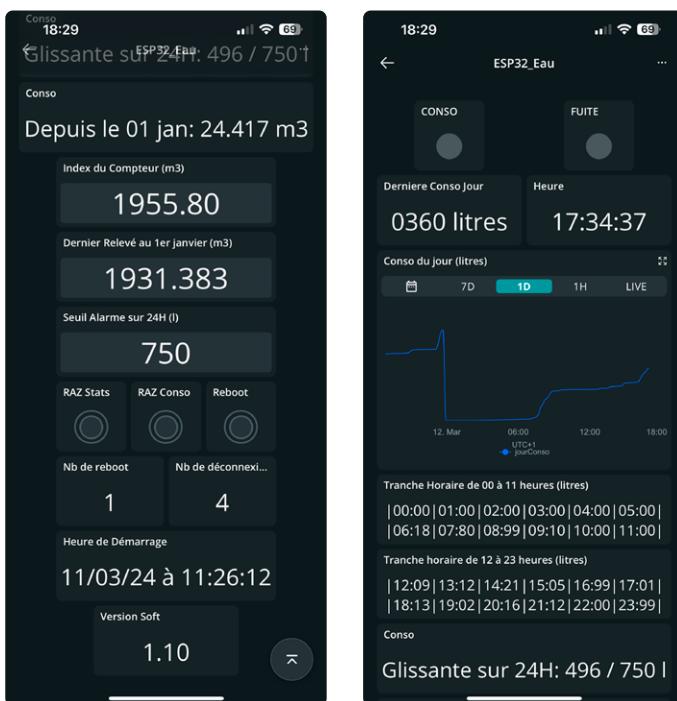


Figure 6a et b. Voici ce que je vois sur mon téléphone portable !

Gateway, DNS). Si l'adresse IP est *dhcp* ou *DHCP*, alors la connexion Wi-Fi est établie à l'aide de DHCP.

Le tableau de bord

La création d'un tableau de bord comme le mien est simple (**figures 6a** et **b**). Cliquez sur + *Dashboard* dans le menu principal. Vous pouvez le renommer. Appuyez ensuite sur *Add*. Sélectionnez le type de widget que vous souhaitez afficher. Il ne reste plus qu'à faire *Link Variable* pour attribuer une valeur à ce widget. Validez par *Done* et répétez cette procédure pour toutes les variables que vous souhaitez afficher sur le tableau de bord.

J'ai également assigné la variable **jourConso** à un widget graphique pour avoir un historique de la consommation d'eau.

Une fois toutes les variables assignées, sélectionnez la fonction d'édition (deux flèches croisées) pour déplacer et redimensionner vos widgets. Cliquez sur l'icône du téléphone pour obtenir une présentation différente sur votre mobile. La création d'un tableau de bord est amusante et ne prend que quelques minutes pour obtenir un résultat impressionnant.

Notifications par email et push

Là encore, rien de plus simple. Il suffit d'aller sur la page des *Triggers* dans le menu principal. Cliquez sur + *Trigger* et liez la variable **alarmLED1** (par exemple), puis choisissez d'envoyer un email ou une notification push, ou les deux.

Le circuit

J'ai réutilisé le même circuit (**figure 7**) que celui utilisé dans d'autres projets, que vous pouvez trouver sur ma page Elektor Labs [6] (ESP32 Thermostat, VMC Regulator, Fuel Measurement, Linky Analyzer and Load Shedding, NTP Server).

J'ai placé mon appareil près de mon panneau de brassage, secouru par un onduleur, mais vous pouvez également connecter une batterie au lithium rechargeable (le module Heltec dispose d'un connecteur à cet effet) qui se chargera automatiquement et maintiendra l'alimentation pendant environ 20 minutes. Comme les données sont sauvegardées périodiquement, seule la consommation d'eau des 15 dernières minutes est perdue si vous n'avez pas d'alimentation électrique de secours.

Le compteur d'eau se trouve à près de 40 m de ma maison. Par conséquent, pour garantir la fiabilité du comptage des impulsions, j'ai préféré utiliser un optocoupleur au lieu d'un filtre RC pour bloquer les interférences. Non seulement il fournit une isolation galvanique, mais il filtre

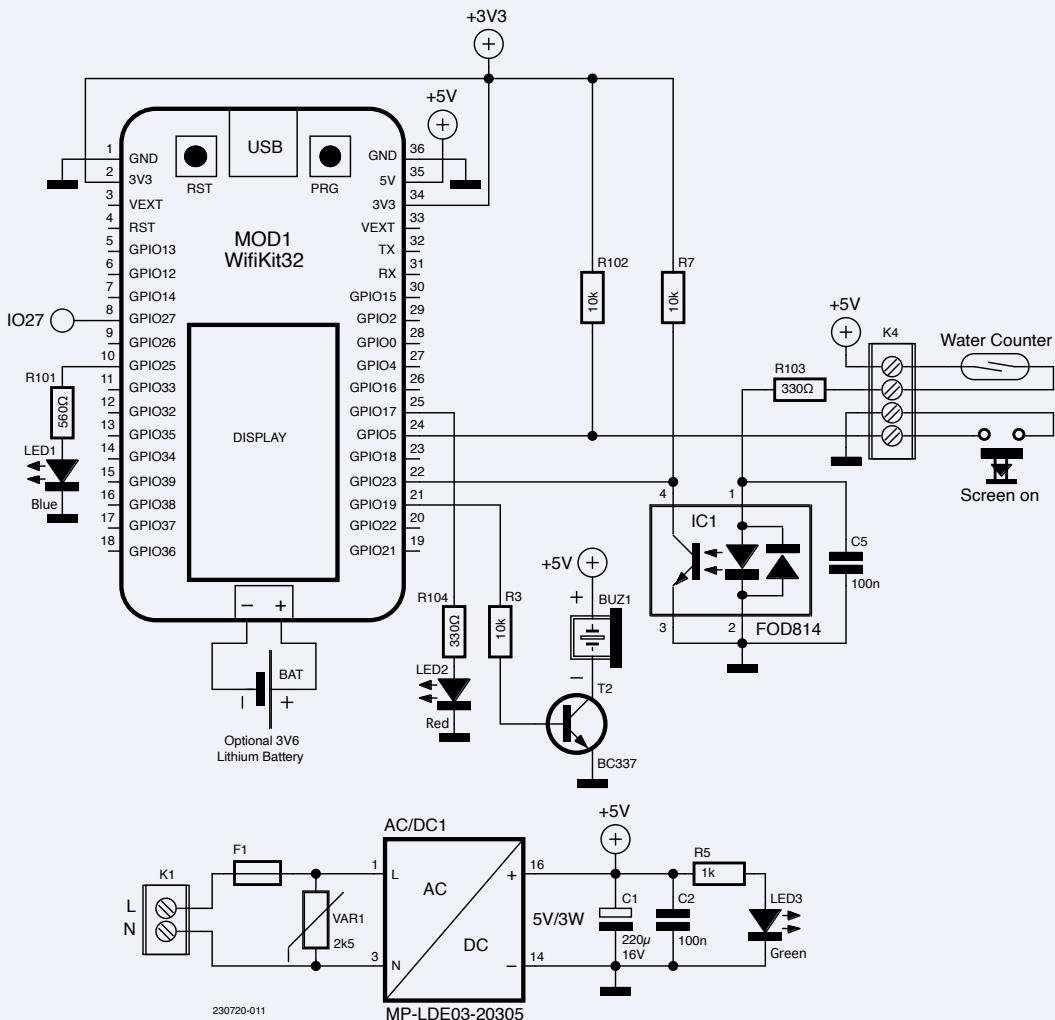


Figure 7. Le schéma du détecteur de fuites d'eau est une version adaptée du thermostat ESP32 [7]. Les résistances R101, R102 et R103 ont été ajoutées. La sortie de l'optocoupleur a été redirigée vers GPIO23.



Listage 1. Routine d'interruption.

```
/* Interrupt counting routine
masking over 750 ms (the transmitter produces pulses of approximately 200 ms for each 0.51 of water consumed)
at the fastest, 1 pulse every 725 ms, which corresponds to about 2.5 m3 per hour (physical limit of my counter)
-----*/
void IRAM_ATTR handle_Interrupt () {
    static unsigned long lastInterrupt = 0;
// debouncing of the pulse - we must wait 750 ms before processing another interrupt
    unsigned long interrupt = millis(); // start of interrupt
    if (interrupt - lastInterrupt >= 750) { // last interrupt more than 750 ms ago?
        portENTER_CRITICAL_ISR(&mux);
        indexCounter++; // cumulative counter index, 1 pulse = 0.5 liter
        indexJour++; // daily counter index
        portEXIT_CRITICAL_ISR (&mux);
        flagInterrupt = true; // an interrupt has been detected
        digitalWrite(SENSOR_LED, HIGH);
// blue LED (sensor) is ON - will be put to OFF 80 ms later in the loop program
    }
    lastInterrupt = interrupt; // reset and memorize the last interrupt to manage debouncing and blue LED
}
```

également les impulsions parasites qui sont trop faibles pour allumer la LED de l'optocoupleur. Environ 10 mA à 3,3 V sont nécessaires pour allumer correctement la LED. Un effet secondaire positif d'un tel courant est qu'il évite l'oxydation des contacts du relais Reed. Un condensateur de 100 nF est suffisant pour filtrer la connexion au capteur Reed. Deux LED fournissent des informations sur l'état de l'appareil. La LED1 bleue, connectée à IO25 (SENSOR_LED), s'allume pendant environ 100 ms pour chaque impulsion reçue sur IO23. Lorsque la LED2 rouge, connectée à IO17 (CONNECT_LED), s'allume, cela signifie que le système n'est plus connecté à l'Arduino Cloud.

Logiciel

La principale difficulté du projet était de ne pas manquer les impulsions du compteur d'eau. C'est pourquoi une grande attention a été portée à l'écriture de la routine d'interruption (**Listage 1**).

À chaque interruption, la LED bleue s'allume. La variable globale `Last_interrupt` enregistre l'heure de l'interruption, ce qui permet à la boucle principale d'éteindre la LED environ 100 ms plus tard (flash). La variable `indexCounter` est de type `volatile int`. Elle contient le nombre d'impulsions, c'est-à-dire le nombre de demi-litres d'eau consommés. Cette variable ne peut être modifiée que dans le cadre de la routine de service d'interruption à l'aide des commandes `portENTER_CRITICAL_ISR(&mux)` et `portEXIT_CRITICAL_ISR(&mux)`.

Mon capteur envoie deux impulsions par litre. On peut le configurer dans le logiciel comme 1, 2 ou 4 impulsions par litre. En fonction des caractéristiques du capteur, la valeur est divisée par 1, 2, 4, 8, etc. (par décalage de bits) dans la boucle principale :

```
Total_Counter = indexCounter >> pulsParLitre ; // 2
impulsions / l
```

La variable `indexJour`, également un `volatile int`, correspond à la variable `jourConso` dans l'Arduino Cloud. Elle n'est pas mise à jour si la valeur est modifiée manuellement dans le tableau de bord Arduino. Comme `indexCounter`, elle est divisée par `pulsParLitre`.

Toutes les 15 minutes

- Les variables `indexCounter` et `indexJour` sont sauvegardées dans la mémoire non volatile (NVS).
- Un tableau `conso[24]` contient la consommation d'eau pour chaque heure (pour des raisons d'affichage, seuls 99 l/h peuvent être comptés). Cela permet de vérifier s'il y a une fuite : s'il n'y a pas deux périodes horaires consécutives avec une consommation à zéro, cela signifie que de l'eau coule en permanence. Dans ce cas, une alarme est déclenchée et la variable `alarmLED2` est mise à 1.
- La quantité totale d'eau consommée depuis minuit est également vérifiée. Si elle dépasse un seuil défini par l'utilisateur, il y a surconsommation. Dans ce cas, une alarme est déclenchée et la variable `alarmLED1` est mise à 1.
- En cas d'alarme, une notification push et un email sont envoyés via l'Arduino Cloud.

Toutes les 60 minutes

La consommation pour l'heure suivante est remise à zéro, et l'écran est allumé pendant une période définie par l'économiseur d'écran (5 mn, voir ci-dessous). Le petit écran OLED permet d'afficher trois pages

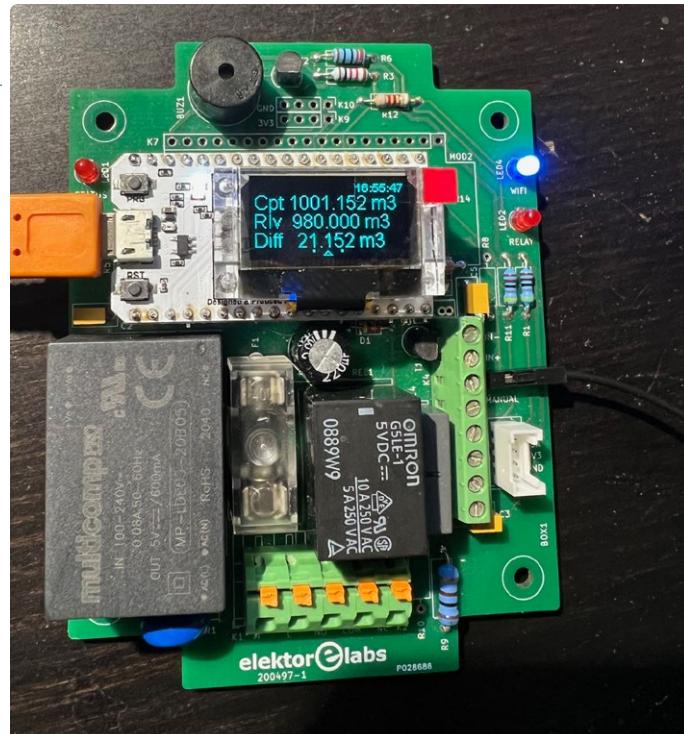


Figure 8. L'assemblage de la carte électronique portant le module Heltec ESP32.



Figure 9. La carte dans son boîtier.

qui défilent toutes les cinq secondes. Il permet d'afficher les caractéristiques du réseau, l'heure, les différents comptages de consommation et, le cas échéant, le type d'alarme déclenchée. On utilise la routine `formatData()` pour n'afficher la consommation sur 24 heures que sur deux lignes.

Certaines fonctions supplémentaires permettent :

- La remise à 0 de toutes les périodes horaires (normalement utilisé uniquement pendant les tests).



- Vérifier si l'Arduino Cloud s'est déconnecté (ce qui, bien sûr, n'entraîne pas d'erreurs de comptage). Si après 600 s (10 minutes) le système ne peut toujours pas se connecter, un problème de réseau est supposé et le système redémarre. Le nombre de redémarrages peut être consulté et, à chaque redémarrage, l'heure et la date de la reconnexion sont indiquées.
- Le 31 décembre à minuit, la consommation annuelle d'eau est remise à zéro et la nouvelle valeur du compteur de consommation d'eau ([Last_Counter](#)) est enregistrée.

Pour vérifier que tout fonctionnait comme prévu, j'ai placé un compteur d'impulsions en parallèle et je n'ai observé aucun écart entre ce que je mesurais et ce que le compteur affichait. Comme j'utilise un écran OLED qui a tendance à vieillir rapidement lorsqu'il est allumé en permanence, j'ai également intégré une routine d'économiseur d'écran pour que l'écran s'éteigne au bout de cinq minutes. Il se rallume en cas d'alarme ou si l'on appuie sur une touche connectée au système.

Mon installation

Pour le matériel, j'ai utilisé la carte que j'avais développée pour un thermostat ESP32 [7], voir les **figures 8 et 9**. L'avantage de cette carte est que l'alimentation secteur 230 VAC est intégrée à la carte, et que toutes les connexions sont effectuées à l'aide de borniers de 3,81 mm.

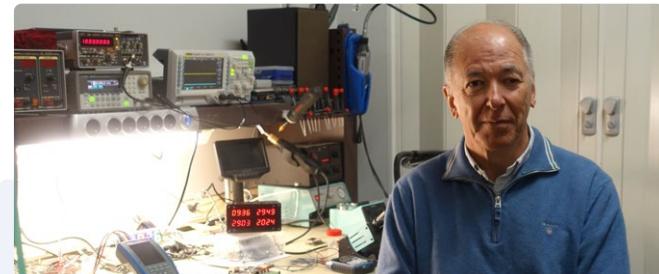
Une surveillance efficace

Le détecteur de fuites d'eau décrit dans cet article est opérationnel depuis plus d'un an. Jusqu'à présent, je n'ai pas remarqué de bogues car j'ai régulièrement mis à jour le système pour y apporter des améliorations et des corrections.

Il y a quelques mois, des toilettes situées à l'extérieur de la maison ont commencé à fuir très discrètement. Le système a détecté une fuite continue d'environ vingt litres par heure de minuit à 7 heures du matin. J'ai remarqué l'alarme à mon réveil. Sans le détecteur, cet événement aurait pu facilement se transformer en un désastre environnemental et financier.

Le système me permet également de savoir si mon adoucisseur d'eau a nettoyé sa résine à 2 heures du matin et quelle quantité d'eau a été utilisée à cet effet. De même, il peut indiquer s'il fonctionne mal. Le détecteur de fuites d'eau est devenu un atout précieux de notre système électrique et électronique domestique. ↵

VF : Denis Lafourcade — 230720-04



À propos de l'auteur

Passionné d'électronique et de technologie depuis son plus jeune âge, Yves Bourdon a suivi des études d'ingénieur à l'INSA de Lyon. Avant même d'avoir terminé ses études, il crée sa première entreprise, ERIM, en 1981. En 1991, Yves crée le premier PC français avec un processeur Intel 286 d'un encombrement de seulement 10 cm x 10 cm. Après avoir vendu ses entreprises en 2009, Yves a consacré une grande partie de son temps à soutenir les entreprises innovantes dans leur développement. Il a notamment été président bénévole de Cap'Tronic (structure publique) pendant plus de 10 ans. Aujourd'hui âgé de 66 ans, Yves est à la retraite et passe une grande partie de son temps dans son laboratoire d'électronique à explorer les applications basées sur l'ESP32.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (yb.electronique@orange.fr) ou contactez Elektor (redaction@elektor.fr).



Produits

- **Carte de développement LILYGO T-Display ESP32 (16 MB)**
www.elektor.fr/19774
- **D. Ibrahim, *The Complete ESP32 Projects Guide, Elektor 2019***
www.elektor.fr/18860

LIENS

- [1] D. Lafourcade, « surveillance de la consommation d'eau avec l'ESP32 », Elektor 7-8/2019 :
<https://www.elektormagazine.fr/magazine/elektor-101/50930>
- [2] S. Romero, « les projets connectés simplifiés », Elektor Guest Edition 12-2022/1-2023 :
<https://www.elektormagazine.fr/magazine/elektor-271>
- [3] Heltec WiFi Kit 32 : <https://heltec.org/project/wifi-kit32-v3/>
- [4] Arduino Cloud : <https://cloud.arduino.cc>
- [5] Ce projet sur Elektor Labs : <https://elektormagazine.fr/labs/esp32-water-leak-detector-connected-to-iot-arduino>
- [6] Projets de l'auteur sur Elektor Labs : <https://www.elektormagazine.fr/labs/13670/ybourdon/projects>
- [7] Y. Bourdon, « Thermostat connecté à ESP32 », Elektor 9-10/2021 : <https://elektormagazine.fr/200497-04>

le quartz

drôle de composant, la série

David Ashton (Australie)

Du quartz naturel aux versions synthétiques avancées, les cristaux jouent un rôle essentiel dans l'électronique, en assurant un contrôle précis des fréquences pour des appareils tels que les microcontrôleurs et les radios. Explorons leurs applications pratiques dans le domaine de l'électronique, depuis les systèmes informatiques jusqu'aux équipements radio.

Le quartz représente un fascinant sujet d'étude pour un débutant en électronique. Ce composant est essentiellement constitué d'un cristal de quartz avec des électrodes plaquées des deux côtés. Le quartz est l'un des composants électroniques les plus utilisés et les plus courants. Il se retrouve presque sur toutes les cartes comportant un microcontrôleur, ainsi que dans tous les systèmes radio. Ces deux dispositifs partagent le besoin d'une référence de fréquence stable – pour les microcontrôleurs, il s'agit du signal d'horloge qui régule leur fonctionnement et synchronise leurs communications avec le monde extérieur. Pour les systèmes radios, cette référence permet de définir la fréquence à laquelle ils fonctionnent.

Évolution du quartz

Avant 1925 environ, la plupart des oscillateurs utilisaient des circuits accordés – composés généralement d'une bobine d'inductance fixe et d'un condensateur variable. Bien que souvent soigneusement conçus, ces circuits présentaient des dérives en fréquence.

En 1880, Jacques et Pierre Curie ont découvert l'effet piézoélectrique : ils ont observé qu'une substance cristalline munie d'électrodes de part et d'autres se déforme lorsqu'une tension est appliquée aux électrodes. Si on l'utilise pour fournir une rétroaction positive dans



Figure 1. Un petit disque de quartz de quelques MHz datant de 1975 et un plus grand quartz de 100 kHz en forme de barre datant de 1959.

un circuit d'amplification, on obtient un oscillateur. Pendant la Première Guerre mondiale, Paul Langevin a exploité cet effet pour générer des fréquences ultrasoniques. En 1925, Westinghouse a intégré un oscillateur à cristaux dans sa station de radio, marquant ainsi le début de l'adoption généralisée des quartz dans les oscillateurs.

Jusqu'à la Seconde Guerre mondiale, on utilisait des cristaux de quartz naturel, dont la majorité était extraite au Brésil. Les Laboratoires Bell développé une méthode innovante permettant la culture de cristaux de quartz synthétiques. En 1970, la plupart des cristaux utilisés étaient devenus synthétiques.

Comprendre les propriétés des quartz

Un cristal fonctionne comme un circuit accordé précis caractérisé par un facteur « Q » élevée. Les quartz présentent des fréquences de résonance séries (basse impédance) et parallèles (haute impédance). En dessous de 30 MHz, on utilise généralement une fréquence intermédiaire, tandis qu'au-delà de cette valeur, la résonance en série est la norme. Les propriétés des cristaux varient en fonction la « coupe » effectuée, c'est-à-dire l'endroit où le cristal est coupé par rapport à ses axes. De nombreuses autres variables interviennent et la physique des cristaux est extrêmement complexe. Certains cristaux, notamment ceux destinés aux hautes fréquences, peuvent être conçus pour fonctionner à une harmonique de leur fréquence naturelle comme la 3^e, la 5^e ou même la 7^e harmonique.



Figure 2. Différents quartz anciens et modernes. En haut : les plus anciens. Au milieu (de gauche à droite) : HC6/U, HC49/U (dont un avec un fusible rearmable PTC rouge utilisé comme « four ») et HC49S à trous traversants à. En bas (de gauche à droite) : Cristaux de montre 32,768 Hz, HC49SMD, et autres quartz CMS.

Les quartz sont sensibles aux variations de la température et de nombreuses techniques sont utilisées pour pallier ce problème, telles que la compensation de la température dans le circuit de l'oscillateur, ou même l'utilisation d'un « four » pour maintenir le cristal à une température fixe supérieure à la température ambiante. On recourt à la « traction » pour régler la fréquence avec précision – il s'agit généralement d'une bobine ou d'un condensateur ajustable, en série ou en parallèle, qui modifie légèrement la fréquence de résonance naturelle. Le quartz est un composant essentiel, quoique souvent mal compris. La **figure 1** montre la construction interne de certains quartz, et la **figure 2** montre des boîtiers de quartz anciens et modernes. ↵

240214-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

David Ashton est né à Londres, a grandi en Rhodésie (aujourd'hui Zimbabwe), a vécu et travaillé au Zimbabwe et vit aujourd'hui en Australie. Depuis son plus jeune âge, il a été passionné par l'électronique. La Rhodésie n'étant pas un centre majeur pour l'électronique, il a dû apprendre à s'adapter et à trouver des solutions créatives. Il a acquis très tôt des compétences en adaptation, substitution et recherche de composants, des compétences dont il est toujours fier. Il a dirigé un laboratoire d'électronique mais a principalement travaillé dans les télécommunications.



enregistreur universel de données de jardinage

un pas vers l'Intelligence Artificielle au Jardin.

Gamal Labib (Egypte)

L'irrigation intelligente contrôlée par des capteurs, contribue à la conservation de l'eau, et aide à l'amélioration de la santé des plantes en situation de stress hydrique, du fait d'un programme d'arrosage inadapté. Un système d'irrigation intelligente, basée sur l'apprentissage machine, est un premier pas vers cette approche. Dans cet article, je présente un enregistreur de données constituant un premier élément du jardinage intelligent.

Source : Adobe Stock

L'automatisation de l'irrigation d'un jardin, soulage son propriétaire du souci quotidien de l'arrosage manuel. Malgré l'automatisation, si la distribution d'eau n'est pas adaptée, des zones décolorées peuvent apparaître dans les pelouses et les plantes peuvent présenter alors des signes de stress hydrique.

L'irrigation intelligente est un sujet sur lequel fabricants et chercheurs se sont focalisés depuis des décennies. La collecte des données par des capteurs, est un moyen magique permettant d'activer ou désactiver l'arrosage de façon adéquate et atténuer les variations des conditions de développement des plantations. Le capteur le plus utilisé pour cela est une sonde mesurant l'humidité du sol au niveau des racines des plantes. Des conditions de sécheresse du sol activent l'arrosage, jusqu'à ce que les racines soient correctement hydratées. [1].

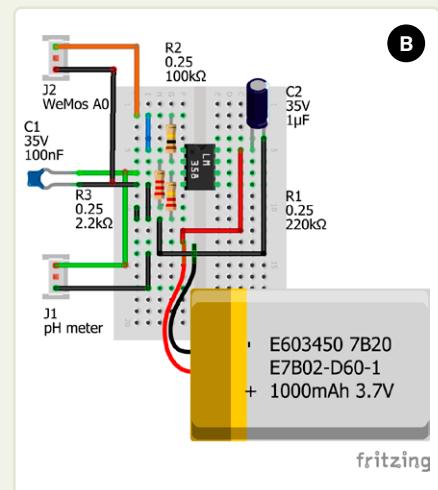
L'Intelligence Artificielle (IA) donne une autre dimension au processus d'irrigation. Plusieurs approches sont possibles. Dans cet exemple [2], une caméra couleur ordinaire est utilisée, en conjonction avec des techniques d'apprentissage afin d'identifier si la plante est en situation de stress en raison d'un manque d'arrosage. Dans un autre exemple [3], l'apprentissage machine utilise les données de capteurs d'humidité, de température, et d'acidité (pH) [4], afin de prévoir les besoins en humidification du sol.

En général, les applications d'apprentissage machine (ML ou Machine Learning), se réfèrent à une grande quantité de données significatives pour un processus, afin de définir le modèle à appliquer. Des nouvelles données, jamais rencontrées précédemment, peuvent être injectées au modèle qui en retour fournit des informations importantes à leur sujet, et détermine les actions appropriées pouvant être prises. Les données utilisées dans les applications d'apprentissage machine peuvent se présenter sous différents formats, le format le plus populaire étant les valeurs séparées par des virgules (CSV ou Comma Separated Values).

Interfaçage d'un pH-mètre

L'acidité des sols varie lentement, selon des grandes périodes, des semaines et parfois de mois sont nécessaires pour constater des changements mesurables, en particulier si elle est modifiée par un jardinier. Ainsi, j'ai considéré que le suivi au jour le jour du pH était un objectif secondaire qui pouvait être accompli manuellement, ou être un complément à ce projet. En recherchant des capteurs de pH numériques disponibles sur le marché, je suis arrivé à la conclusion qu'ils étaient coûteux, particulièrement ceux pouvant être interfacés dans des projets utilisant un microcontrôleur Arduino.

Le capteur le plus économique disponible lors de la création de ce projet était le modèle manuel que je possédais déjà dans mon jardin, qui bizarrement, ne nécessite aucune pile pour fonctionner (voir **figure A**). Le capteur possède deux électrodes métalliques de matériaux différents et le sol agit comme un électrolyte. Une telle configuration constitue une pile électrochimique, qui est bien entendu relativement faible, mais



Dans un projet d'Intelligence Artificielle pour le jardinage, il est nécessaire de se concentrer sur la collecte des données. C'est pour cela que j'ai prévu une grande variété de capteurs couvrant une large gamme de paramètres du jardinage.

Un enregistreur « universel » doit produire des fichiers de données CSV contenant les valeurs ambiantes de température et d'humidité, d'exposition au soleil, d'humidité du sol, de pH et d'évaluation du taux d'engrangement et nutriments NPK (nitrogen-phosphorus-potassium fertility and nutrients), etc. Le volume d'eau atteignant le sol doit également être mesuré, permettant la détermination de l'efficacité des arroseurs utilisés par le système d'arrosage. Malheureusement, les capteurs de pH et de NPK sont trop coûteux pour pouvoir être expérimentés. J'ai cependant réussi à interfaçer un capteur de pH économique afin de lui permettre de fonctionner avec des microcontrôleurs, comme le montre l'encadré « **Interfaçage d'un pH mètre** ».

Réalisation du projet

L'enregistreur des données de culture est réalisé autour du module WeMos D1 Mini, basé sur un microcontrôleur ESP8266. Il inclut les possibilités de connectivité Wifi, il supporte l'intégration de l'enregistreur au système de domotique (IoT) de mon logement. Son implémentation actuelle utilise quatre capteurs analogiques : un capteur de luminosité TEMT6000 permettant de déterminer l'ensoleillement des plantes, un capteur de pluie évaluant les averses de pluie (qui ne sont pas très fréquentes dans ma région), que je me prépare à utiliser

suffisamment puissante pour être mesurable par un multimètre analogique. En utilisant un amplificateur opérationnel et une petite pile, j'ai pu amplifier le signal afin de le diriger vers l'entrée analogique du chip WeMos, comme les autres capteurs (figure B).

également pour déterminer la zone couverte des arroseurs, un capteur de niveau d'eau pour déterminer la pluviométrie ou la quantité d'eau délivrée par les arroseurs au point de mesure des données, et un capteur d'humidité du sol pour déterminer le degré d'humidité des racines. La configuration comprend un capteur de température et humidité DHT11, pour mesurer les conditions ambiantes au niveau de l'enregistreur. Il est important de noter que certains capteurs analogiques possèdent une sortie digitale qui fournit un niveau HAUT ou BAS, selon un seuil déterminé par un potentiomètre de réglage présent sur la carte. L'enregistreur est représenté sur la **figure 1**.



Figure 1. Vue de la réalisation du prototype.



L'ESP8266 ne possédant qu'une seule entrée analogique (A0), mais je dois connecter plusieurs capteurs analogiques. J'ai utilisé un chip multiplexeur analogique 4052, afin de permettre à l'enregistreur d'utiliser jusqu'à 8 capteurs. Le multiplexeur analogique utilise deux interfaces digitales GPIO ; les broches 16 (sortie D0) et 14 (D5) du Mini D1 sont utilisées à cette fin, permettant de sélectionner l'un des quatre capteurs afin de le relier à l'entrée analogique du microcontrôleur. L'enregistreur étant alimenté par une batterie, et prévu pour fonctionner sur une longue période, l'économie d'énergie est primordiale. Durant les périodes où l'enregistreur est désactivé, j'ai prévu de stopper le fonctionnement des capteurs en interrompant leurs connexions à la masse par un transistor NPN BD139. Pouvant supporter un courant de 1,5 A, sa base qui peut accepter une tension de 5 V, est pilotée directement par la broche GPIO 15 (D8) du D Mini. En fournissant un niveau HAUT au transistor, celui-ci est saturé, permettant le fonctionnement des capteurs. Dans le cas contraire, un niveau BAS permet d'économiser l'énergie consommée.

L'enregistreur devant être universel, j'ai choisi de délivrer les valeurs analogiques directement fournies par les capteurs, donnant ainsi à l'utilisateur, la possibilité de calibrer les capteurs et d'interpréter les valeurs le plus souplement possible. Le capteur de pluie est une exception, puisque je me base sur ses sorties analogique et numérique, comme je le décris plus loin. J'ai utilisé la broche GPIO13/D pour acquérir sa sortie numérique.

La **figure 2** montre le câblage des composants du projet. Les capteurs de pluie et d'humidité du sol sont totalement passifs, ils nécessitent la présence d'un module de contrôle pour fournir leurs données analogiques, alors que les autres capteurs fonctionnent de façon autonome. La liste complète des composants est fournie à la fin de l'article. La **figure 3** montre le câblage physique des composants du projet, avant leur installation dans le boîtier de l'enregistreur.

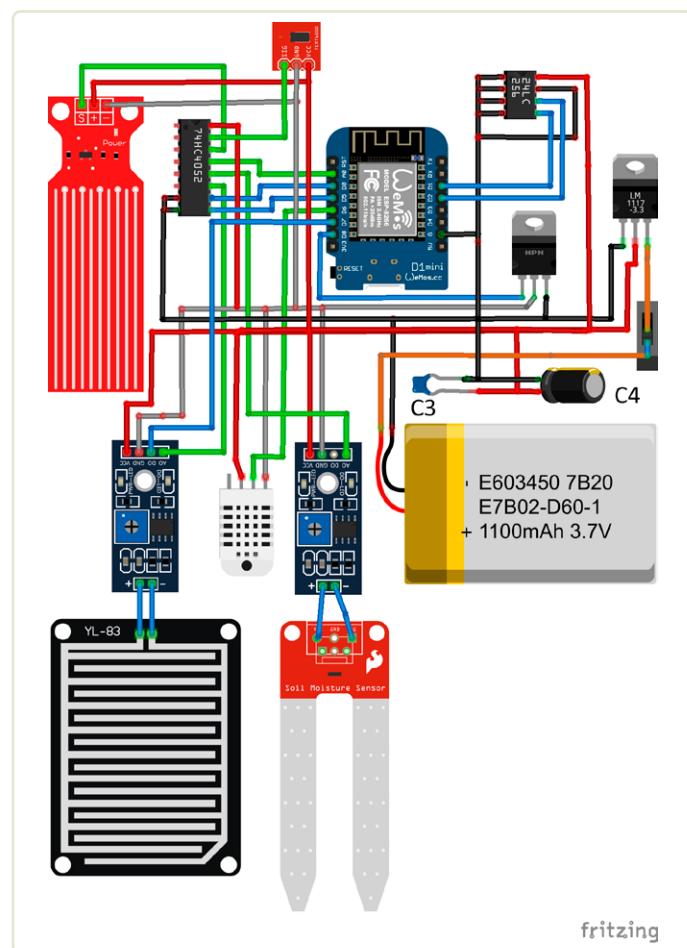


Figure 2. Câblage du circuit de l'enregistreur de jardinage.

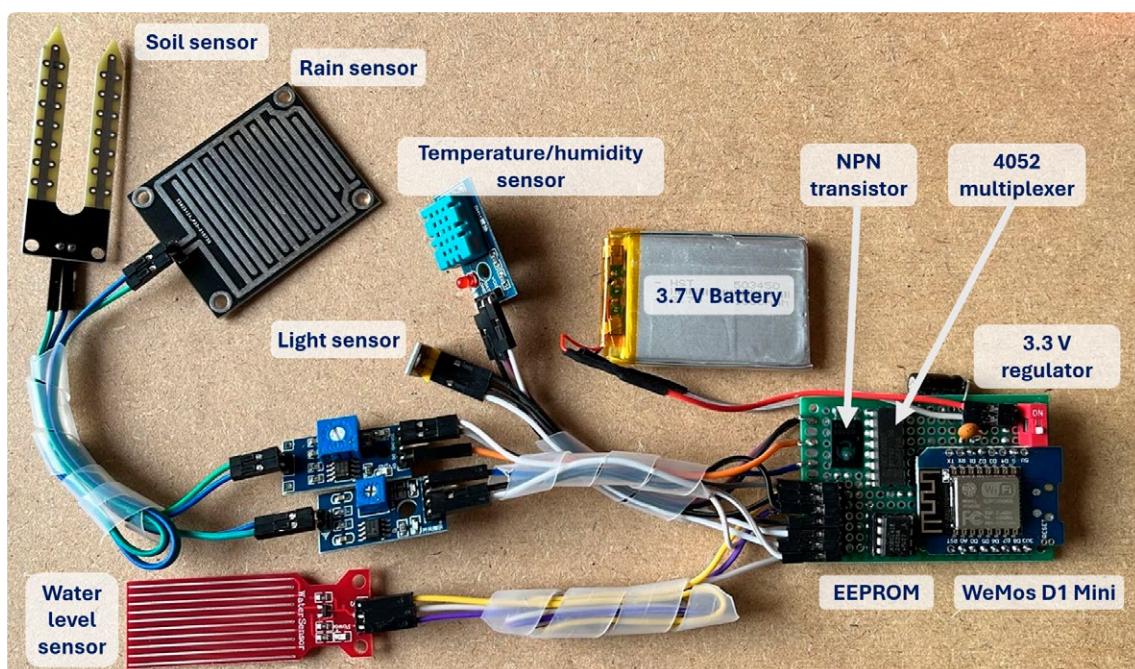


Figure 3. Ensemble des éléments reliés à l'extérieur du boîtier du projet.

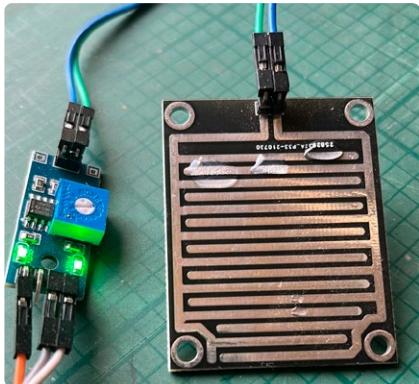
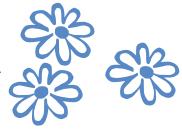


Figure 4. Le capteur de pluie.

```

13:55:15.169 -> Analog output: 538 → New reading
13:55:16.187 -> Analog output: 538
13:55:17.186 -> Analog output: 540
13:55:18.190 -> Analog output: 538
13:55:19.169 -> Analog output: 538
13:55:20.165 -> Analog output: 538
13:55:21.159 -> Analog output: 538
13:55:22.187 -> Analog output: 537
13:55:23.189 -> Analog output: 537
13:55:24.182 -> Analog output: 537
  }
```

Readings induced by water viscosity

Figure 5. Interprétation des mesures délivrées par le capteur de pluie.

Après être mis en place, le capteur des paramètres d'ambiance est accessible par le fond du boîtier, afin de se situer le plus près possible du sol, tout en étant protégé de l'eau d'arrosage et de la pluie. Sur la partie haute de l'enregistreur, les capteurs de luminosité et de pluie sont exposés à l'environnement pour une meilleure acquisition de l'ensoleillement et de la quantité de pluie. Une trappe en verre est nécessaire pour isoler le capteur de lumière de l'extérieur. De façon identique aux capteurs d'eau et d'humidité du sol, ils sont reliés à l'enregistreur par un câble de 20 cm de long afin de faciliter leur positionnement. L'enregistreur est alimenté par une batterie au lithium de 3,7 V. Cette batterie a une capacité moyenne de 1100 mAh et ses dimensions lui permettent d'être facilement logée dans le boîtier du projet. Le module WeMos D1 fonctionnant sous une tension de 3,3 V, j'ai utilisé un régulateur de tension à faible chute (LDO) pour diminuer la tension de 0,4 V afin de respecter les spécifications de WeMos. J'ai choisi un régulateur prévu pour fournir une intensité de 800 mA permettant d'alimenter les composants de l'enregistreur. L'utilisateur devra choisir une batterie dont la capacité permet d'assurer le fonctionnement de l'enregistreur selon la durée souhaitée.

Estimation de la quantité d'eau reçue

La quantité d'eau reçue par la zone considérée du jardin est un facteur important lorsqu'il convient de résoudre les problèmes de stress hydrique des plantes. J'ai utilisé un capteur de niveau d'eau pour déterminer cette quantité. C'est la méthode traditionnelle pour déterminer la quantité d'eau reçue, provenant de la pluie ou des asperseurs d'arrosage, dans un entonnoir alimentant un réservoir gradué. Le principe d'évaluation de ce dispositif simple est de considérer que la hauteur d'eau H dans le réservoir (par exemple cylindrique), dont la surface de base S est connue correspond au volume $V = H \times S$. Ainsi, lors de l'écriture du sketch Arduino, j'ai défini S par une constante et j'enregistre l'évolution du volume d'eau en fonction du temps selon la hauteur d'eau H indiquée par le capteur de hauteur d'eau.

Le capteur de pluie (**figure 4**) utilise une approche différente pour déterminer la quantité d'eau. La sonde utilise le même principe qu'un capteur de niveau d'eau, plus il y a d'eau reçue, plus la résistance du capteur diminue. Il possède différentes pistes métalliques qui lui permettent de déterminer la quantité de gouttes lui parvenant en un temps donné. Toutefois, l'accumulation de gouttes entre les pistes métalliques qui dépend de la viscosité de l'eau, est problématique.

Pour éviter ce problème, j'ai installé l'enregistreur en l'inclinant de façon à ce que les pistes métalliques du capteur soient orientées vers le sol afin de favoriser l'écoulement des gouttelettes et éviter de comptabiliser l'eau déjà prise en compte. Cela impose également la détection momentanée des gouttes d'eau au moment où elles atteignent le capteur et d'ignorer la détection des gouttes qui glissent sur la sonde en maintenant le contact. Les gouttelettes qui restent bloquées indéfiniment en bas des pistes du capteur constituent un problème supplémentaire.

C'est là que la sortie numérique du capteur vient à mon secours. En ajustant le potentiomètre de la figure précédente, cet effet peut être neutralisé, et la sortie numérique indique alors «NO RAIN» (pas de pluie, niveau HAUT). Il reste maintenant à résoudre le problème du ruissellement des gouttelettes sur les pistes métalliques. La mise en œuvre expérimentale du capteur de pluie montre de légères variations sur la sortie analogique du capteur. J'ai pris en compte ce phénomène en éliminant les valeurs subsistantes avant l'apparition d'une nouvelle valeur. Ce fonctionnement est illustré sur la **figure 5**, la sortie analogique d'une seule gouttelette indique une valeur de 538 et que les valeurs suivantes sont pratiquement identiques durant le ruissellement de la gouttelette provoqué par la gravité.

Lors de l'utilisation du capteur de pluie pour effectuer le réglage des asperseurs rotatifs, le système doit prendre en compte les deux catégories d'asperseurs utilisés pour les pelouses : les arrosoirs fixes qui irriguent le gazon selon un cercle de rayon constant et les modèles rotatifs qui utilisent la force de l'eau pour provoquer leur rotation. Afin de prendre en considération les deux types d'asperseurs, le micrologiciel du microcontrôleur doit pouvoir distinguer entre la dispersion continue de l'eau et la dispersion intermittente.

Utilisation des données enregistrées

L'enregistreur de Jardin est équipé d'une mémoire externe EEPROM de 32 ko pour la mémorisation des données captées pendant toute la durée de fonctionnement. Je l'ai installée sur un support DIP plutôt que de la souder, afin de permettre, si nécessaire, de la remplacer par un chip de capacité plus importante. La mémoire EEPROM interne de l'ESP8266 peut également être utilisée, mais sa capacité est limitée à 4 ko. De plus, l'écriture répétée de l'EEPROM interne aurait pour effet de réduire la durée de vie de l'ESP, c'est pourquoi je ne recommande pas de l'utiliser pour enregistrer les données.

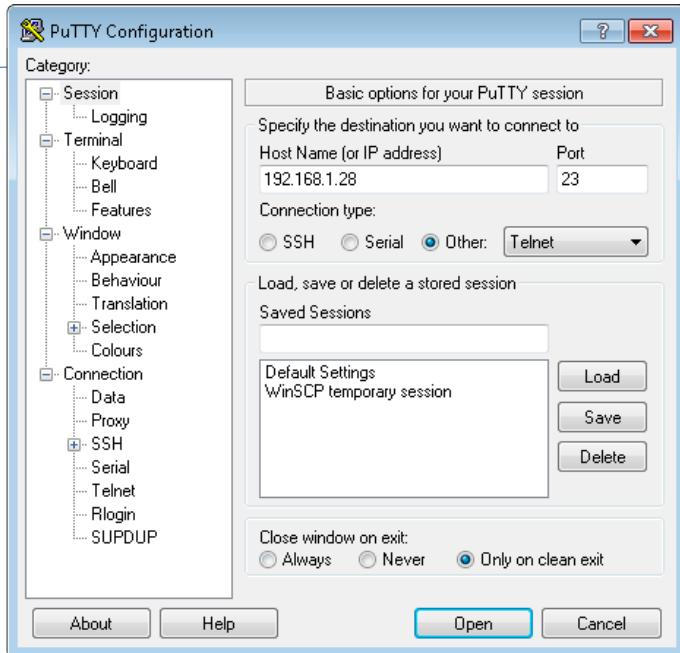


Figure 6. Configuration de PuTTY (1/2).

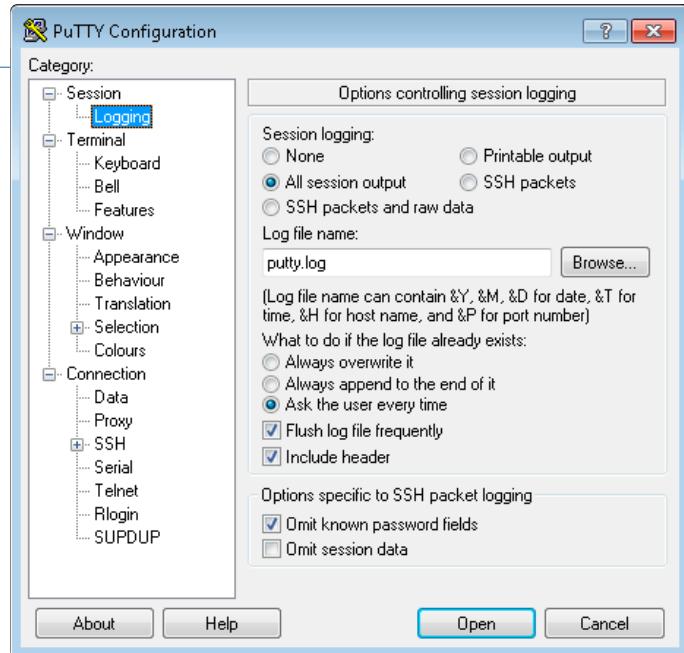


Figure 7. Configuration de PuTTY (2/2).

Chaque enregistrement des données recueillies est formaté selon le format CSV (valeurs séparées par des virgules). J'ai décidé que toutes les valeurs seraient représentées par des nombres entiers, cette application ne nécessitant pas la précision des nombres décimaux. C'est pourquoi le code transpose toutes les valeurs flottantes `Float` en entiers `Integer`, puis la fonction `String()` est utilisée pour les convertir en chaînes de caractères (`strVal`). L'appel en cascade des fonctions `strcat(strVal, ",")` est ensuite prévu pour insérer les virgules entre les différentes valeurs d'un enregistrement, les chaînes ainsi créées sont précédées par un en-tête contenant l'horodatage de l'enregistrement. Le projet ne comprend pas d'horloge RTC (Real Time Clock) pour maintenir une heure précise. Le dispositif se met à l'heure par consultation d'un serveur Internet NTP, lors de la réinitialisation intervenant à minuit chaque jour, afin d'obtenir les informations horaires. La fonction `millis()` est ensuite utilisée pour déterminer l'heure en cours et générer l'horodatage. La structure d'un enregistrement effectué est : *horodatage, température, humidité, éclairement, volume d'eau collecté, volume d'eau déversé, humidité du sol.*

Concernant la transmission à un ordinateur des valeurs enregistrées, il existe plusieurs approches possibles. Une des options peut être de disposer d'un serveur web s'exécutant sur l'enregistreur. Il est également possible de programmer le microcontrôleur pour transmettre les données à un PC en mode batch, après avoir été enregistrées dans la mémoire EEPROM. Un client Telnet tel que PuTTY peut être utilisé pour enregistrer les données dans un fichier CSV. Une troisième possibilité peut être de transmettre les enregistrements à un PC dès qu'une ligne de données CSV est créée. J'ai implémenté ce dernier mode d'opération dans la version courante de ce projet. La **figure 6** montre la fenêtre de configuration PuTTY sur le PC.

La procédure à suivre est assez simple : s'assurer que l'enregistreur de jardinage et le PC sont connectés au même réseau sans-fil, entrer l'adresse IP de l'enregistreur dans la zone *Host Name (nom de l'ordinateur hôte)*, définissez le port 23, puis choisissez le protocole Telnet depuis la fenêtre de saisie. Avant de cliquer sur *Open (Ouvrir)* pour lancer la transmission Telnet, il faut définir les paramètres de sauvegarde des données reçues de l'enregistreur. Cette étape est illustrée sur

```

28 2024-03-07 21:41:03,21.40,50.00,0.00,0,1024,9
29 2024-03-07 21:41:09,21.40,50.00,0.00,0,1024,9
30 2024-03-07 21:41:16,21.40,53.00,0.00,0,1024,9
31 2024-03-07 21:41:22,22.20,55.00,0.00,0,1024,9
32 2024-03-07 21:41:28,23.40,58.00,0.00,0,1024,9
33 2024-03-07 21:41:35,24.50,60.00,0.00,0,1024,9
34 2024-03-07 21:41:41,25.30,62.00,0.00,0,1024,9
35 2024-03-07 21:41:48,26.20,62.00,0.00,0,1024,9
36 2024-03-07 21:41:54,27.10,59.00,0.00,0,1024,9
37 2024-03-07 21:42:00,27.10,56.00,0.00,0,1024,9
38 2024-03-07 21:42:07,27.60,53.00,0.00,0,1024,9
39 2024-03-07 21:42:13,27.20,51.00,0.00,3,1024,9
40 2024-03-07 21:42:20,27.10,50.00,0.00,3,1024,9
41 2024-03-07 21:42:26,27.10,49.00,0.00,3,1024,9
42 2024-03-07 21:42:32,27.00,48.00,0.00,2,1024,9
43 2024-03-07 21:42:39,26.70,48.00,0.00,3,1024,9
44 2024-03-07 21:42:45,26.70,47.00,0.00,3,1024,9
45 2024-03-07 21:42:52,26.20,47.00,0.00,3,1024,9
c 2024-03-07 21:42:58,26.20,47.00,0.00,3,1024,9
bye bye

```

Figure 8. Fenêtre Telnet d'envoi des données au PC.

la **Figure 7**. Vous devez choisir *All session output (Envoyer des données des sessions)* et *Ask the user every time (Demander à l'utilisateur à chaque transmission)* afin de ne pas réécrire un fichier log créé lors d'une session précédente. L'emplacement du fichier CSV destination doit alors être défini en utilisant le bouton *Browse (Parcourir)*. Vous pouvez alors cliquer sur *Open* pour démarrer le protocole Telnet. Après avoir été lancé, l'utilitaire Telnet débute la transmission des enregistrements en provenance de l'enregistreur, comme décrit sur la **figure 8**. Il est possible de mettre fin à la session de transmission en entrant « C » ; l'enregistreur retournera alors le message « bye bye » (Au revoir...) généré par la fonction `telnetAction()`. Les données des capteurs sont également disponibles sur le port série, au fur et à mesure qu'elles sont assemblées dans les enregistrements. Elles peuvent être lues par le Moniteur série de l'EDI Arduino selon la **figure 9**.



```

COM39

rain detected!
Water Level (cm): 657 Water Volume (cm3): 6
Soil Moisture (unscaled): 1024 Soil Moisture (scaled): 9
Humidity: 47.00% Temperature: 26.20°C 79.16°F Heat index: 26.10°C 78.99°F
Raw ADC data: 22.00 Volts: 0.07 Lux: 46.79
Analog Input: 380 Digital Input: 0
-62
rain detected!
Water Level (cm): 667 Water Volume (cm3): 6
Soil Moisture (unscaled): 1024 Soil Moisture (scaled): 9
Humidity: 47.00% Temperature: 26.20°C 79.16°F Heat index: 26.10°C 78.99°F
Raw ADC data: 27.00 Volts: 0.09 Lux: 57.43
Analog Input: 385 Digital Input: 0
-62
rain detected!
Water Level (cm): 649 Water Volume (cm3): 6
Soil Moisture (unscaled): 1024 Soil Moisture (scaled): 9
Humidity: 47.00% Temperature: 26.20°C 79.16°F Heat index: 26.10°C 78.99°F
Raw ADC data: 27.00 Volts: 0.09 Lux: 57.43
Analog Input: 387 Digital Input: 0
-62
rain detected!
Water Level (cm): 639 Water Volume (cm3): 6
Soil Moisture (unscaled): 1024 Soil Moisture (scaled): 9
Humidity: 47.00% Temperature: 26.20°C 79.16°F Heat index: 26.10°C 78.99°F
Raw ADC data: 27.00 Volts: 0.09 Lux: 57.43
Analog Input: 397 Digital Input: 0
-61
rain detected!
Water Level (cm): 634 Water Volume (cm3): 6
Soil Moisture (unscaled): 1024 Soil Moisture (scaled): 9
Humidity: 47.00% Temperature: 25.66°C 78.44°F Heat index: 25.66°C 78.19°F
Raw ADC data: 28.00 Volts: 0.09 Lux: 59.55
Analog Input: 399 Digital Input: 0
-61
rain detected!
Water Level (cm): 629 Water Volume (cm3): 6

```

Autoscroll Show timestamp Newline 9600 baud Clear output

Figure 9. Données de sortie de chacun des capteurs dans le Moniteur série de l'IDE Arduino.

Programme

La programmation du projet est réalisée en utilisant l'IDE Arduino. Le programme est composé du fichier principal *Garden.ino*, accompagné de cinq fichiers *.ino* auxiliaires. Chacun de ces fichiers comporte les fonctions pilotant un périphérique commun. *Timing.ino* contient les fonctions relatives au serveur d'horodatage NTP, *telnet-streaming.ino* concerne le formatage des enregistrements et leur envoi vers le PC, *analog-sensors.ino* comporte quatre fonctions de gestion des capteurs analogiques de l'ensoleillement, de la pluie, du volume d'eau et de l'humidité du sol. Enfin *digital-sensors.ino* contient une seule fonction qui fait l'acquisition de la température et de l'humidité ambiantes, *external-eeprom.ino* contient les fonctions utilisées pour lire et écrire dans la mémoire EEPROM.

Ce style de codage facilite l'interprétation et la mise au point de chaque composante fonctionnelle du projet et améliore la réutilisabilité. Le **listage 1** comporte la section `loop()` du sketch ne comportant que dix instructions, dont six sont des appels aux fonctions décrites précédemment. La liste décrit également la section `setup()` dans laquelle je



Listage 1. Main loop.

```

void loop() {
    delay(2000);
    digitalWrite(SENSORS, HIGH); // enable sensors power
    delay(1000);
    senseAmbient(); // read ambient humidity & temperature
    senseLight(); // read ambient light intensity
    senseRain(); // read rain status
    senseWater(); // read irrigation water level
    senseSoil(); // read soil moisture level
    telnetAction(); // assemble logged data and send to telnet app
    digitalWrite(SENSORS, LOW); // disable sensors power
}

// this is part of setup()
pinMode(MUXA, OUTPUT); // part of 2-bit analog sensor selection address
pinMode(MUXB, OUTPUT); // part of 2-bit analog sensor selection address
pinMode(SENSORS, OUTPUT); // sensors GND control pin
pinMode(DHTPIN, INPUT_PULLUP); // ambient DHT sensor feed
pinMode(RAIN, INPUT); // rain status sensor digital feed
digitalWrite(SENSORS, LOW); // disable sensors power

// see [5] for the complete code

```





Liste des composants

Module Microcontrôleur

WeMos D1 Mini (ESP8266), <https://amazon.com/dp/B07W8ZQY62>

Capteurs

Capteur de niveau d'eau, <https://amazon.com/dp/B09J2NK21Y>
Capteur de pluie, <https://amazon.com/dp/B01DK29K28>
Capteur d'humidité du sol, <https://amazon.com/dp/B083Q8DFHX>
Capteur d'ambiance DHT11, <https://amazon.com/dp/B0CCF2C2CF>
Capteur de luminosité TEMT6000, <https://amazon.com/dp/B00L8DW8L2>

Divers

C3 = 100 nF, céramique
C4 = 470 µF, 10 V électrochimique
24LC256 EEPROM (Mouser)
LM1117MP-3.3-NOPB régulateur 3,3V (Mouser)
Transistor BD139 (Mouser)
Multiplexeur analogique CD4052BE (Mouser)
Batterie au lithium 1,100 mAh 3.7V, <https://amazon.com/dp/B06WRRQGR6>
Boîtier du projet 85×50×21 mm, <https://amazon.com/dp/B0B4VGGN6J>
Circuit Imprimé vierge double face FR-4 PCB 40×60 mm, <https://amazon.com/dp/B0968F3748>

que l'on retrouve dans un jardin. Une mise à jour du projet pourrait consister en l'enregistrement de l'état des plantes intéressantes, en utilisant une caméra ou d'autres types de capteurs, et l'ajout de ces données aux enregistrements effectués dans le cadre de ce projet. Cela permettrait de créer un modèle d'irrigation utilisant des techniques d'apprentissage machine avec supervision. 

VF : Jean Boyer — 230629-04



À propos de l'auteur

Gamal Labib est un ingénieur passionné, enthousiasmé par les systèmes embarqués depuis plus de vingt ans, il est actuellement mentor (à codementor.io). Il est titulaire d'une maîtrise en engineering (Meng) et d'un Doctorat (PhD) en informatique. En plus d'écrire des articles techniques pour des magazines, il est professeur associé dans des Universités Égyptiennes et certifié Consultant en Informatique.

Questions ou commentaires ?

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



Produits

- **WeMos D1 mini Pro**
www.elektor.fr/19185
- **Module ESP8266 ESP-01 Wi-Fi**
www.elektor.com/17326
- **Kit de capteurs SparkFun**
www.elektor.com/19620

LIENS

- [1] Exemple de projet d'irrigation : <https://circuitdigest.com/microcontroller-projects/automatic-irrigation-system-using-arduino-uno>
- [2] Surveillance des plantations par l'IA : <https://smellslike.ml/posts/tf-microcontroller-challenge-droopthereitis/>
- [3] Projet d'irrigation intelligente à base d'Arduino : <https://tinyurl.com/2kkbc2zx>
- [4] Information sur le pH du sol : <https://esf.edu/eis/eis-soil-ph.php>
- [5] Téléchargements : <https://elektormagazine.com/230629-01>

un générateur analogique 1 kHz

ondes sinusoïdales à faible distorsion

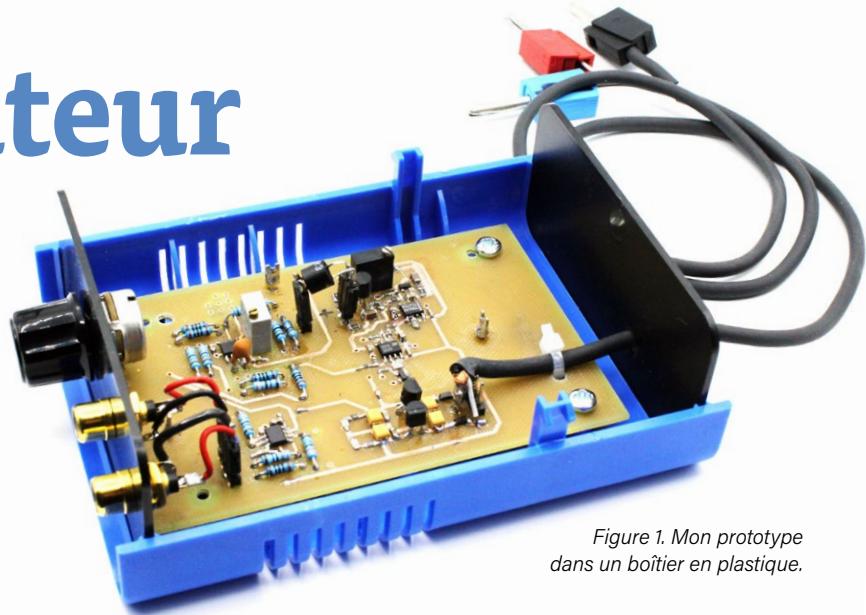


Figure 1. Mon prototype dans un boîtier en plastique.

Alfred Rosenkränzer (Allemagne)

Un signal sinusoïdal de haute qualité à 1 kHz est indispensable pour les mesures en électronique audio. Les générateurs numériques ne sont pas encore en mesure de pouvoir rivaliser avec leurs cousins analogiques. Comme toujours dans le domaine du haut de gamme, les appareils prêts à l'emploi ne sont pas vraiment bon marché. Avec le circuit proposé ici, construire soi-même un tel générateur revient à un prix raisonnable, sans avoir à trop sacrifier la qualité du signal.

Ma revue du QA403 de QuantAsylum [1] montre que les générateurs d'ondes sinusoïdales numériques sont tout à fait utilisables. Malheureusement, cette société a retiré son modèle analogique QA480 – un complément au QA403 (voir le blog sous [2]). Cet appareil combinait un générateur analogique de 1 kHz avec un filtre coupe-bande approprié et un amplificateur de 12 dB de haute qualité. L'amplitude et le filtre coupe-bande pouvaient être contrôlés via USB à partir d'un PC. Mais c'est de l'histoire ancienne. Selon le blog, l'abandon est en partie dû à l'effort excessif requis pour le réglage.

Le circuit était basé sur la conception d'une autre société [3] et a heureusement été publié dans ce blog. Comme il n'existe actuellement aucun appareil équivalent disponible à l'achat, j'ai légèrement révisé le schéma du circuit et développé un circuit imprimé approprié. Le prototype est présenté à la **figure 1**. L'alimentation est fournie par un bloc d'alimentation externe symétrique de ± 15 V. Par souci de simplicité, j'ai utilisé un potentiomètre pour le réglage de l'amplitude. Il n'y a donc ni USB ni logiciel et donc aucun parasite provenant d'une électronique numérique.

Le circuit

Le schéma de principe de la **figure 2** montre cinq unités fonctionnelles. L'oscillateur proprement dit se compose d'un filtre passe-bande mis en oscillation par une rétroaction positive. L'amplitude est stabilisée par un régulateur dont la valeur de mesure est obtenue à partir du signal de sortie par redressement et filtrage, et la valeur de consigne à partir

Analog 1-kHz-Generator

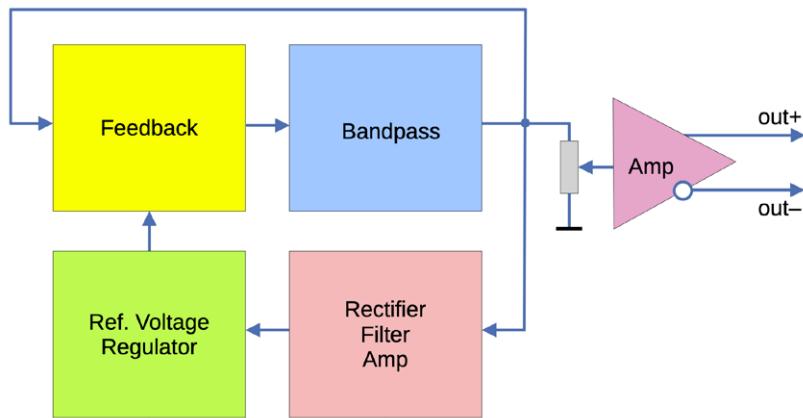


Figure 2. Schéma fonctionnel du générateur de 1 kHz.

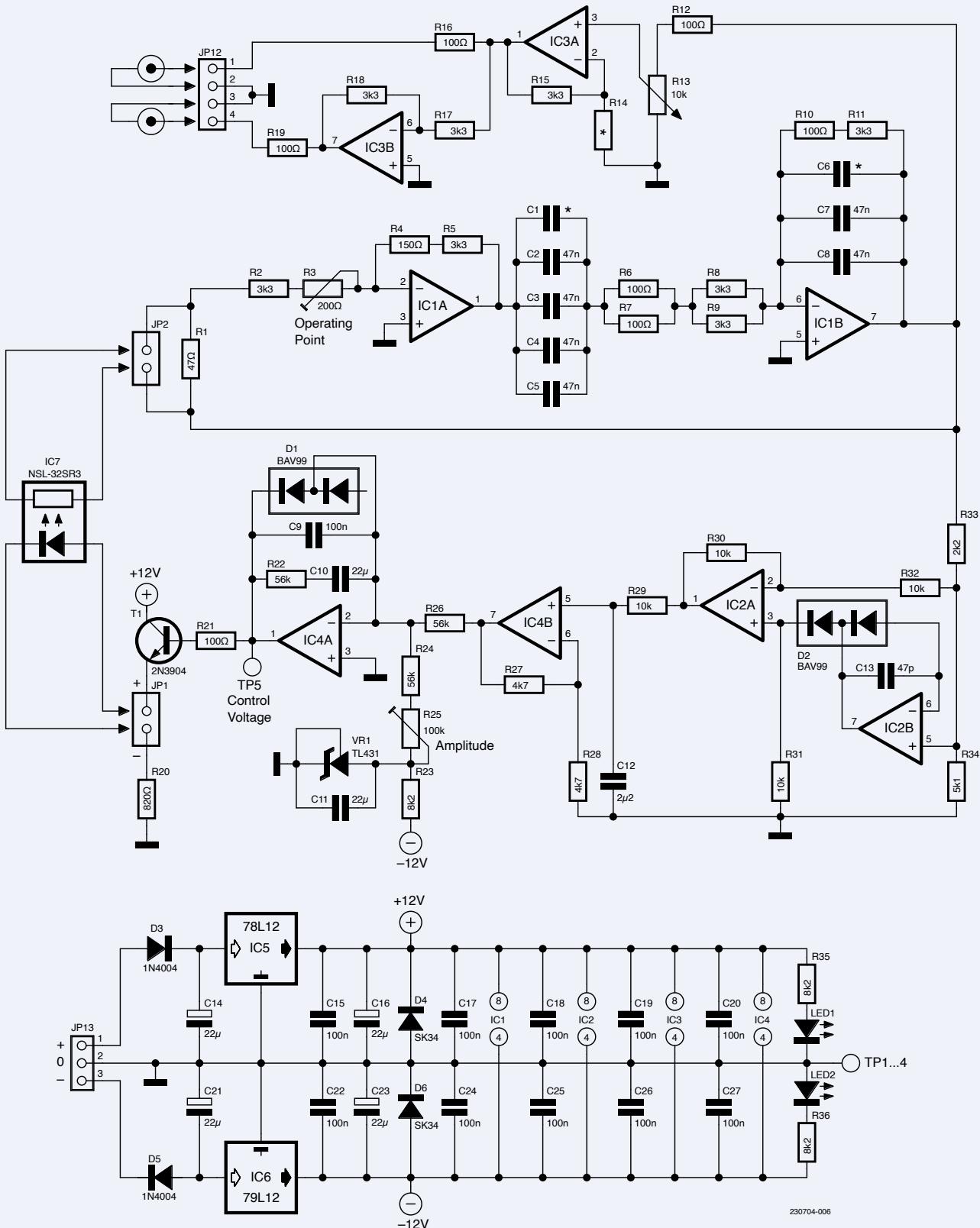


Figure 3. Le circuit ne nécessite que quatre amplificateurs opérationnels doubles, deux régulateurs de tension et un optocoupleur.

d'une tension de référence qui est nécessaire pour obtenir un signal stable. Un amplificateur de sortie fournit un signal différentiel d'amplitude réglable. Le circuit de la **figure 3** montre comment les blocs fonctionnels sont traduits en électronique réelle.

Filtre passe-bande

Le filtre passe-bande est réalisé par la combinaison de IC1B avec R6...R11 et C1...C8. Sa fréquence centrale détermine la fréquence d'oscillation. La qualité des composants utilisés influence la distorsion du signal sinusoïdal obtenu. On peut utiliser des condensa-

teurs COG (à coefficient de température nul) en céramique dans des boîtiers CMS ou des condensateurs à film traversant. Le circuit présente donc deux fois plus de condensateurs, dont seulement la moitié, c'est-à-dire des condensateurs CMS ou des condensateurs traversants, doivent être installés.



Liste des composants

Résistances

Sauf indication contraire :

MELF 0204 ou couche mince CMS 1206, 1%.

R1 = 47 Ω

R2, R5, R8, R9, R11, R15, R17, R18 = 3k3

R3 = 200 Ω , potentiomètre multitours,
vertical, RM 1/10".

R4 = 150 Ω

R6, R7, R10, R12, R16, R19 = 100 Ω^*

R12, R16, R19 = 100 Ω

R13 = 10 k, potentiomètre linéaire

R14 = optionnel*

R20 = 820 Ω , CMS 0603

R21 = 100 Ω , CMS 0603

R22, R24, R26 = 56 k Ω , CMS 0603

R23, R35, R36 = 8k2, CMS 0603

R25 = 100 k Ω , potentiomètre multitours,
vertical, RM 1/10".

R27, R28 = 4k7, CMS 0603

R29, R30, R31, R32 = 10 k Ω , CMS 0603

R33 = 2k2, CMS 0603

R34 = 5k1, CMS 0603

Condensateurs

C1, C6 = optionnel*, C0G, CMS 0603

C2, C3, C8 = 47 nF, film, RM 2/10" (à la
place de C4, C5, C7)*

C4, C5, C7 = 47 n, C0G, CMS0805 (à la
place de C2, C3, C8)*

C9, C15, C17...C20, C22, C24...C27 = 100 nF,
X7R, CMS 0603

C10, C11 = 22 μ F / 16 V, CMS 1206

C12 = 2 μ 2 / 16 V, CMS 1206

C13 = 47 pF, COG, CMS 0603

C14, C16, C21, C23 = 22 μ F / 25 V,
électrolytique au tantalum, SMC-B

Semi-conducteurs

LED1, LED2 = LED, CMS 0805

D1, D2 = BAV199, SOT23

D3, D5 = 1N4004

D4, D6 = SK34, DO214AC

T1 = 2N3904, SOT23

IC1, IC3 = OPA2211*, SOIC8

IC2, IC4 = TL072, SOIC8

IC5 = 78L12, TO92

IC6 = 79L12, TO92

IC7 = NSL-32SR3*

VR1 = TL431, TO92

Divers

JP1, JP2 = barrette à 2 broches, LP 1/10".

JP12 = barrette à 4 broches, LP 1/10".

JP13 = barrette à 3 broches, LP 1/10".

Carte de circuit imprimé 230704-01

Bouton pour le potentiomètre R13

Bornes de sortie audio*

* Voir le texte

Sur le circuit imprimé, on voit clairement que seuls deux condensateurs connectés en parallèle sont raccordés à l'entrée de IC1B (plus C1 si l'on veut ajuster la fréquence). Dans la boucle de retour, il n'y a même qu'un seul condensateur (plus éventuellement C6 pour l'ajustement de la fréquence). Les pastilles de soudage des résistances sont adaptées au format CMS1206, facile à souder à la main, ainsi qu'au format des résistances cylindriques MELF204, de qualité comparable à celle des versions traversantes à film métallique. À la place des MELF, on peut aussi utiliser des résistances à couche mince.

La capacité dans la branche d'entrée est le double de celle dans la boucle de retour, alors que c'est l'inverse pour les résistances. Cette configuration devrait permettre de réduire le bruit, du moins mathématiquement. Il est conseillé de mesurer les condensateurs avec précision et que les condensateurs mis en parallèle soient aussi identiques que possible, ce qui facilitera le réglage fin ultérieur de la fréquence par C1 et C6 ou R6/R7 et R10. Le filtre passe-bande a un gain d'environ -1 à la fréquence centrale.

Rétroaction positive

Comme pour obtenir des oscillations stables le gain global doit être égal à +1, le gain (réglable) du circuit composé de IC1A et R1...

R5 doit aussi être d'environ -1. Si le gain est trop faible, le circuit n'oscillera pas, et s'il est trop élevé, les signaux seront si grands qu'ils pousseront les ampli-op à la saturation. Une amplitude constante avec un signal sinusoïdal non écrété ne peut être obtenue qu'en contrôlant la rétroaction globale positive.

L'optocoupleur NSL-32SR3 d'Advanced Photonix [4], facilement disponible, sert d'élément de contrôle. Le courant traversant la LED du côté de l'entrée agit sur la résistance de la LDR intégrée du côté de la sortie. Cet optocoupleur assure l'isolation électrique entre l'entrée et la sortie.

Mesure de l'amplitude

Environ 70% du signal de sortie de IC1B atteint l'entrée de IC2A via le diviseur de tension R33/R34. Cette atténuation permet d'éviter un comportement transitoire bizarre lors de la mise sous tension, lorsque l'oscillateur est brièvement poussé à la limite d'amplitude. Le circuit du redresseur de précision construit autour d'IC2 est décrit dans une note d'application de Texas Instruments [5]. Le signal redressé est lissé par le filtre passe-bas

composé de R29 et C12, puis doublé par IC4B. Si vous le souhaitez, un calculateur en ligne adéquat [6] vous permettra de calculer les valeurs numériques. Avec ses composants, IC4A fonctionne comme un régulateur.

Stabilisation de l'amplitude

La puce TL431 fournit une tension de référence stabilisée de -2,5 V, dont le bruit est réduit par C11. Le courant provenant de IC4B et de R26 est compensé par R24 et R25, cette dernière servant à régler l'amplitude. D1 empêche la tension de commande en sortie de IC4A de devenir négative. Commandé via R21, le transistor T1 pilote la LED de l'optocoupleur dont la résistance de sortie agit sur la rétroaction globale positive. La boucle de régulation est ainsi fermée. Sa constante de temps est déterminée par R26, R22, C9 et C10.

Étape de sortie

Via R12, le signal sinusoïdal atteint le potentiomètre R13, qui est utilisé pour régler l'amplitude du signal de sortie entre 0 et la valeur maximale fixée par R25. IC3A sert d'amplificateur de sortie. R14 permet d'augmenter légèrement l'amplification si nécessaire. Avec son gain de -1, IC3B fournit le signal de sortie négatif. Les bornes de sortie peuvent être de type RCA, BNC, jack audio TS ou XLR, selon les besoins.

Alimentation électrique

Le circuit est alimenté par du ± 15 V. Deux régulateurs de tension en font du ± 12 V stabilisé. D3/D5 empêchent les dommages causés par l'inversion de polarité, et D4/D6 protègent IC5 et IC6 d'un blocage lors de la mise sous tension. Les deux LED s'allument lorsque l'alimentation est active.

Choix des composants

La qualité des ampli-op IC1 et IC3 a une influence majeure sur la qualité du signal. C'est pourquoi j'ai utilisé ici les types OPA2211, légèrement plus chers. Bien entendu, le circuit fonctionne également avec d'autres ampli-op doubles à broches compatibles. Pour IC2 et IC4, pratiquement n'importe quel op-amp double convient. Mon prototype utilise le TL072. Les condensateurs C1 à C8 influencent également la distorsion. Comme déjà mentionné, il convient d'utiliser ici des condensateurs de type C0G dans des boîtiers CMS ou des condensateurs à film traversants. Attention : un seul «mauvais» condensateur selon les critères audiophiles, avec un diélectrique XR7

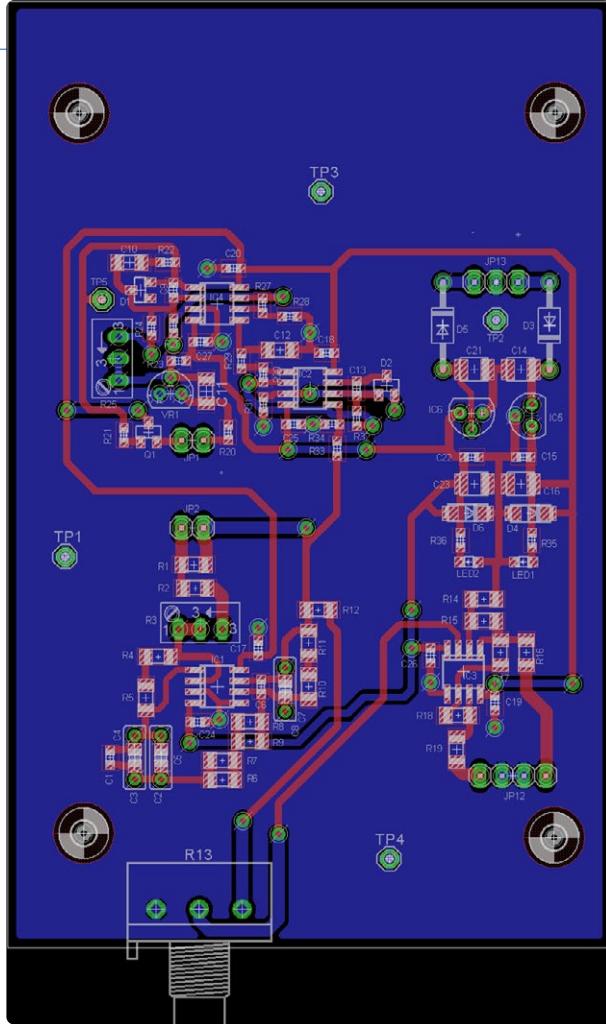


Figure 4. Disposition de la carte. Les fichiers correspondants peuvent être téléchargés au format Eagle sous [7].

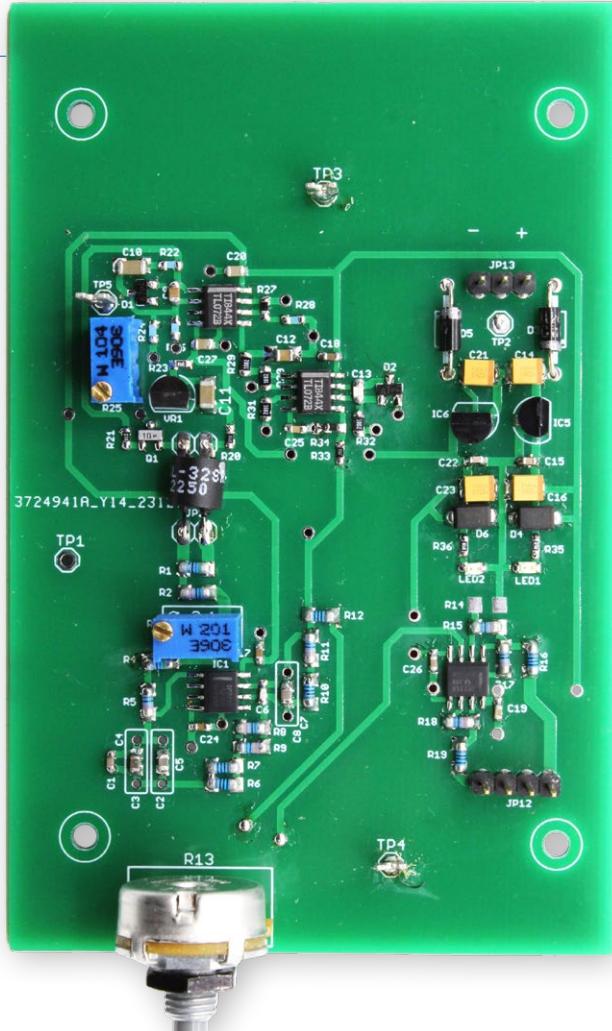


Figure 5. Le circuit imprimé entièrement assemblé de mon prototype.

ou similaire, peut tout gâcher ! Les résistances sur le chemin du signal doivent être du type à couche mince CMS 1206 ou MiniMELF204.

Assemblage et mise en service

L'optocoupleur IC7 ne doit pas être installé tout de suite. La figure 1 montre que je l'ai rendu enfichable à l'aide de deux barrettes à 2 broches. Cela facilite le test du circuit : si vous court-circuitez les broches JP2 et tournez le potentiomètre R3, le circuit doit se mettre à osciller pour un certain réglage et produire un signal limité à $20 V_{cc}$.

Cela permet de vérifier le fonctionnement de l'étage de sortie, du redresseur et du filtre, y compris l'amplificateur. La fréquence devrait être proche de 1 kHz. Le réglage fin est effectué ultérieurement à une amplitude contrôlée et stable. La fréquence change légèrement lorsque les ampli-op atteignent la saturation. Lorsque le signal atteint ses limites, la sortie de IC4A doit être à 0 V. Si l'oscillateur est arrêté, elle augmente lentement jusqu'à environ +10 V. Les caractéristiques de transfert de l'optocoupleur peuvent varier considérablement.

J'en ai acheté quatre et chacun a eu un comportement différent. En alimentant la LED d'entrée avec une alimentation de laboratoire et une résistance en série, vous obtiendrez la résistance de sortie en la mesurant à l'aide d'un multimètre. Un courant de LED de 3 à 5 mA s'est avéré être un bon point de fonctionnement.

Ensuite, connectez une résistance avec la valeur mesurée en parallèle à R1 et réglez R3 de manière à ce que l'oscillateur commence tout juste à osciller. Si nécessaire, vous pouvez ajuster R4. Si vous retirez la résistance et insérez l'optocoupleur, le circuit doit fournir un signal sinusoïdal avec une amplitude constante et stable. R25 peut être utilisé pour ajuster l'amplitude souhaitée à des valeurs de 1 à $3 V_{cc}$. Des amplitudes plus grandes entraînent une distorsion plus élevée.

La constante de temps du circuit de régulation a été délibérément choisie assez grande, ce qui rend le temps de stabilisation assez long, mais garantit un niveau très constant. Le point de fonctionnement du régulateur est réglé avec R3.

Assemblage et mise en service

L'optocoupleur IC7 ne doit pas être installé tout de suite. La figure 1 montre que je l'ai rendu enfichable à l'aide de deux barrettes à 2 broches. Cela facilite le test du circuit : si vous court-circuitez les broches JP2 et tournez le potentiomètre R3, le circuit doit se mettre à osciller pour un certain réglage et produire un signal limité à $20 V_{cc}$.

Cela permet de vérifier le fonctionnement de l'étage de sortie, du redresseur et du filtre, y compris l'amplificateur. La fréquence devrait être proche de 1 kHz. Le réglage fin est effectué ultérieurement à une amplitude contrôlée et stable. La fréquence change légèrement lorsque les ampli-op atteignent la saturation. Lorsque le signal atteint ses limites, la sortie de IC4A doit être à 0 V. Si l'oscillateur est arrêté, elle augmente lentement jusqu'à environ +10 V.

Les caractéristiques de transfert de l'optocoupleur peuvent varier considérablement. J'en ai acheté quatre et chacun a eu un comportement différent. En alimentant la LED d'entrée avec une alimentation de laboratoire

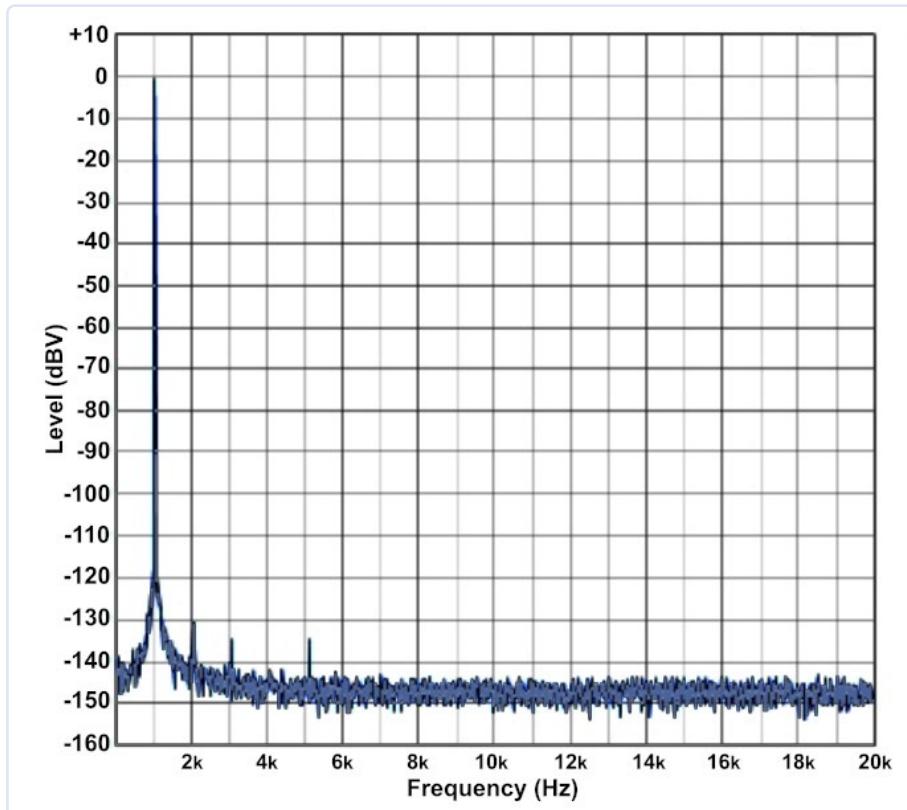


Figure 6. Spectre du signal de sortie à un niveau de 0 dBV.

et une résistance en série, vous obtiendrez la résistance de sortie en la mesurant à l'aide d'un multimètre. Un courant de LED de 3 à 5 mA s'est avéré être un bon point de fonctionnement.

Ensuite, connectez une résistance avec la valeur mesurée en parallèle à R1 et réglez R3 de manière à ce que l'oscillateur commence tout juste à osciller. Si nécessaire, vous pouvez ajuster R4. Si vous retirez la résistance et insérez l'optocoupleur, le circuit doit fournir un signal sinusoïdal avec une amplitude constante et stable. R25 peut être utilisé pour ajuster l'amplitude souhaitée à des valeurs de 1 à 3 V_{cc}. Des amplitudes plus grandes entraînent une distorsion plus élevée. La constante de temps du circuit de régulation a été délibérément choisie assez grande, ce qui rend le temps de stabilisation assez long, mais garantit un niveau très constant. Le point de fonctionnement du régulateur est réglé avec R3.

Fréquence et autres

La fréquence peut maintenant être réglée avec précision. Vous pouvez modifier la capacité en ajustant C1 et C6 (C1 = 2 × C6) ou ajuster les valeurs de R6/R7 et R10. La fréquence est calculée à l'aide de la formule habituelle $f = 1/(2\pi RC)$. Après le réglage de la fréquence, il peut être nécessaire de retoucher le point de fonctionnement. Si vous disposez d'un filtre coupe-bande, les fréquences du générateur et du filtre doivent être identiques.

J'ai réalisé un circuit imprimé approprié au circuit. La **figure 4** montre sa disposition et la **figure 5** la carte assemblée. La **figure 6** montre le spectre au niveau nominal de 0 dBV. Toutes les harmoniques ont un niveau inférieur à -130 dBV : le signal de sortie présente donc une distorsion extrêmement faible. Ce spectre a été mesuré à l'aide de l'analyseur audio APx555 d'Audio Precision. La qualité de ce générateur de signaux maison est donc tout à fait respectable !

Les valeurs des composants mentionnées dans le circuit fonctionnent assez bien avec mon prototype, mais il y a encore de l'espace pour une optimisation. Il me reste également quelques cartes nues. Si vous êtes intéressé, veuillez me contacter directement. ↗

VF : Helmut Müller — 230704-04



À propos de l'auteur

Alfred Rosenkränzer a travaillé pendant de nombreuses années comme ingénieur en développement, d'abord dans le domaine de la technologie de la télévision professionnelle. Depuis la fin des années 1990, il développe des circuits numériques à grande vitesse et des circuits analogiques pour les testeurs de circuits intégrés. L'audio est son passe-temps favori.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (alfred_rosenkraenzer@gmx.de) ou contactez Elektor (redaction@elektor.fr).



Produits

➤ QuantAsylum QA403 Analyseur audio 24 bits
www.elektor.fr/20530

➤ FNIRSI 2C23T (3-en-1)
Oscilloscope 2 voies (10 MHz)
+ Multimètre + Générateur de signaux
www.elektor.fr/20717

LIENS

- [1] A. Rosenkränzer, "Comparing the QuantAsylum QA403 to the Gold Standard," 2023: <https://tinyurl.com/yudmy5sk>
- [2] Blog QA403 : <https://tinyurl.com/yckdwe8e>
- [3] Site web de JanasCard : <http://www.janascard.cz/aHome.html>
- [4] Optocoupleur NSL-32SR3 (Mouser) : <https://tinyurl.com/4c5c8nss>
- [5] Redresseur de précision (TI) : <http://www.ti.com/lit/ug/tidu030/tidu030.pdf>
- [6] Calcul des paramètres de la tension alternative : <https://tinyurl.com/vsrbpzbff>
- [7] Page web de l'article : <https://www.elektormagazine.fr/230704-04>

Miletus : utiliser les applications Web hors ligne

accès aux fonctions de l'appareil et du système

Veikko Krypczyk (Allemagne)

De nos jours, les applications web constituent un standard dans de nombreux domaines. Elles sont exécutées dans un navigateur et peuvent donc être déployées sur presque tous les appareils, y compris un Raspberry Pi. Le framework Miletus offre la possibilité de transformer ces applications web peuvent en applications natives, permettant ainsi leur utilisation hors ligne. Il vous permet également d'accéder aux interfaces du système local, par exemple pour lire et émettre des signaux via les broches GPIO.

Les applications de contrôle des appareils ou composants connectés à un PC ou à un Raspberry Pi doivent être adaptées au système d'exploitation utilisé et nécessitent l'accès aux interfaces du système. Elles sont appelées "applications natives" et sont spécialement créées et exécutées sur le système cible. D'autre part, il existe des applications web, qui, s'exécutant dans un navigateur, peuvent être lancées sur presque tous les systèmes. Cependant, ces applications présentent certaines limites par rapport aux applications natives. L'utilisation hors ligne et

l'accès aux interfaces du système ne sont pas facilement possibles. Le **tableau 1** présente une comparaison des principales caractéristiques des applications natives et des applications web.

Pour les applications destinées aux électroniciens, des avantages tels que l'utilisation hors ligne, l'accès aux interfaces du système et l'intégration de pilotes individuels rendent les applications de bureau natives attrayantes. Si vous souhaitez développer une application qui fonctionne sur différents systèmes, tels que Windows, macOS et

Tableau 1. Caractéristiques importantes des applications natives par rapport aux applications web.

Caractéristiques	Application native	Application Web
Interface utilisateur	native	basée sur HTML, CSS et JavaScript
Installation	nécessaire - les fichiers exécutables doivent être copiés sur le système	non nécessaire - les fichiers sont chargés par le navigateur via l'internet
Mises à jour	doivent être installées localement sur chaque appareil	gérées de manière centralisée sur le serveur
Exécution	directement sur le système	dans le navigateur in the browser
Multiplateforme	non — une application distincte doit être créée pour chaque système	oui
Fonctionnement hors ligne	oui	non
Accès aux fonctions de l'appareil et du système	oui	non
Accès aux bases de données installées localement, par exemple SQLite	oui	non
Contrôle de matériel spécifique via des pilotes	oui	non
Accès au système de fichiers	oui	limité

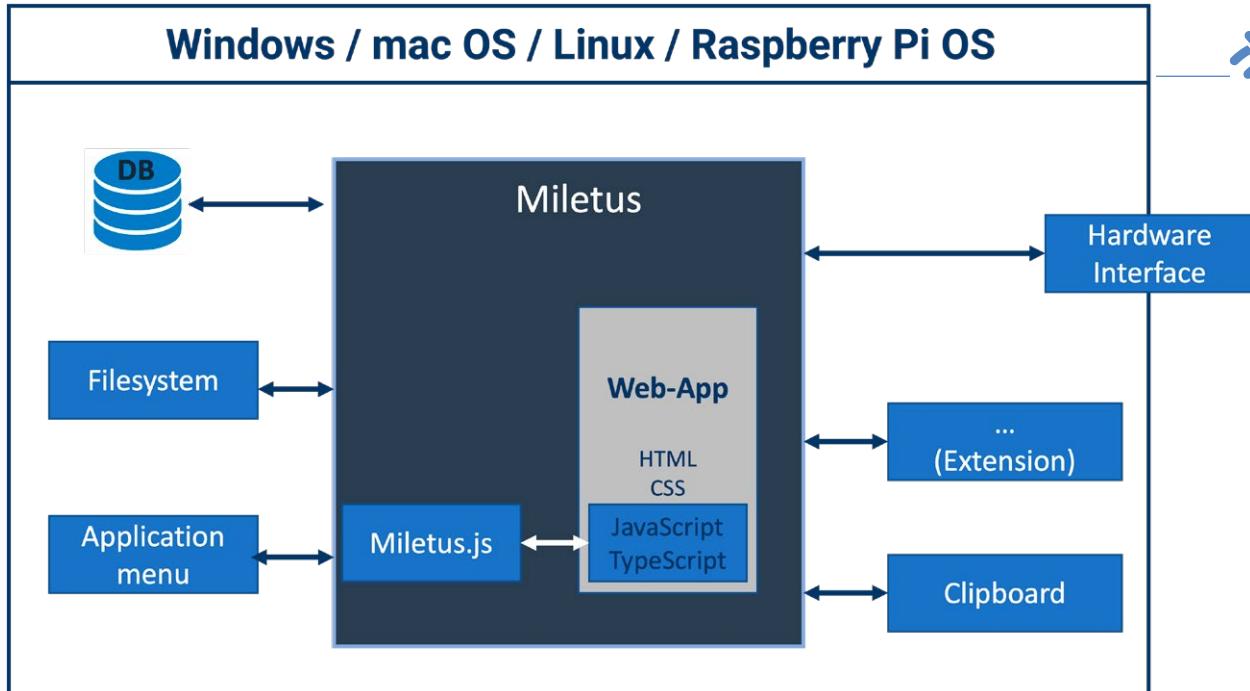


Figure 1. Architecture d'une application Miletus.

Raspberry Pi, les choses peuvent rapidement devenir compliquées. Chaque système a ses propres règles pour la création de l'interface utilisateur par exemple. Cela signifie que pour trois systèmes cibles, il faut développer trois applications, triplant ainsi les efforts de développement. De plus, le savoir-faire technique requis varie, car la programmation d'une application Windows native diffère de celle d'une application pour le Raspberry Pi ou pour macOS. Une application web, en revanche, est créée en HTML, CSS et JavaScript et s'exécute dans un navigateur ; en outre, de puissantes bibliothèques telles que *Bootstrap* simplifient la conception de l'interface.

Dans cet article, nous présentons un cadre intéressant appelé Miletus [1]. Avec Miletus, vous pouvez empaqueter une application web, nouvelle ou existante, dans un paquetage d'application pour différents systèmes cibles (y compris le Raspberry Pi) et l'exécuter en tant qu'application native. Ainsi, il est possible d'utiliser l'application hors ligne et d'avoir accès aux fonctions système.

Le cadre Web Miletus

Avant de nous pencher sur l'utilisation pratique, nous allons présenter les fonctionnalités les plus importantes de Miletus. Les applications créées fonctionneront sur les systèmes d'exploitation Windows, macOS et Linux, y compris le Raspberry Pi OS. L'API de Miletus permet d'accéder aux fonctions système et de l'appareil, au système de fichiers et au matériel externe connecté à l'appareil via des pilotes.

Pour un électronicien, une fonctionnalité particulièrement intéressante est la possibilité d'accéder aux interfaces matérielles du Raspberry Pi, notamment aux ports GPIO, I²C, SPI et UART, ainsi qu'au tampon de mémoire. Ainsi, presque toutes les limitations de l'application web sont éliminées, tout en conservant sa compatibilité multiplateforme. Pour illustrer cela, la **figure 1** présente l'architecture d'une application basée sur le cadre Miletus.

L'intégration de l'application web dans un package d'applications natives pour le système cible correspondant ouvre des horizons pour de nouvelles utilisations. Il s'agit notamment d'applications pour le contrôle de machines ou d'applications de l'Internet des objets (IoT) où il est nécessaire d'envoyer et de recevoir des signaux via les inter-

faces matérielles du Raspberry Pi. Une autre particularité du cadre est l'extensibilité de l'API. Cela est possible grâce à des "bibliothèques partagées" - sous Windows, par exemple, sous la forme de fichiers DLL (*dynamic linked library*). Si une telle bibliothèque existe pour le système cible, par exemple un pilote pour un périphérique spécifique, il est possible d'intégrer cette bibliothèque dans Miletus et d'accéder à ses fonctions depuis l'application web. Miletus utilise le moteur de navigation standard fourni par le système cible :

- **Windows** : *WebView2*
- **Linux/Raspberry Pi OS** : *WebKitGTK*
- **macOS** : *WebKit (Safari)*

On peut généralement supposer que les moteurs de navigation mentionnés sont déjà installés sur les systèmes cibles, de sorte qu'il n'est pas nécessaire que l'application en fournisse d'autres. Le package d'applications à distribuer reste léger, ce qui accélère la distribution de l'application tout en utilisant les ressources du système avec modération. Ceci est particulièrement important pour l'exécution sur le Raspberry Pi. Comme le navigateur existant du système est utilisé, les applications bénéficient aussi automatiquement des mises à jour fonctionnelles et de sécurité du navigateur.

Cependant, vous devez vérifier la configuration requise sur le système cible et vous devrez éventuellement installer des bibliothèques système supplémentaires. Les bibliothèques suivantes doivent être disponibles sur les systèmes cibles :

- **Windows 32/64-bit** : La dernière version du navigateur Edge Chromium doit être installée. En outre, les fichiers *WebView2Loader_x86.dll* (32 bits) ou *WebView2Loader_x64.dll* (64 bits) doivent être copiés dans le dossier *System32* ou *SysWow64*, respectivement. Ces fichiers sont fournis avec l'installation de Miletus.
- **Linux/Raspberry Pi OS** : L'interface graphique *GTK3* est utilisée. Vous pouvez l'installer en ligne de commande avec la commande `sudo apt install libwebkit2gtk-4.0-dev`.
- **macOS** : Aucune autre condition préalable n'est requise.

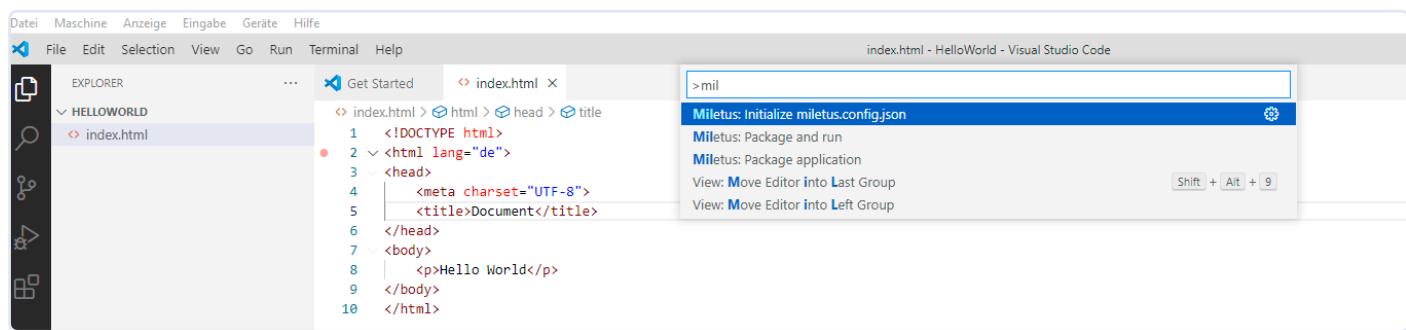


Figure 2. Commandes pour le Packager Miletus dans Visual Studio Code.

Alternative : Electron

Le cadre Electron [2] est également utilisé pour convertir une application web en une application de bureau. Par exemple, l'éditeur Visual Studio Code, utilisé ici, a été développé avec ce framework (c'est-à-dire qu'il s'agit en fait d'une application web). Contrairement à Miletus, Electron fournit son propre moteur de navigation, ce qui signifie que les packages d'applications sont beaucoup plus volumineux. En outre, les applications Electron ne sont pas compatibles avec le Raspberry Pi et l'API ne permet pas d'extensions.

Installation et premiers essais

Ce dont nous avons besoin :

- **Application Web** : Pour les premiers essais, un simple fichier HTML (*index.html*) qui ne produit que le texte "Hello, world" est suffisant. Nous ajoutons également un bouton pour démontrer le fonctionnement de l'API Miletus (voir ci-dessous).
- **Visual Studio Code** : Cet éditeur est téléchargeable à partir de [3] pour votre propre système d'exploitation.
- **Extension pour Visual Studio Code** : Vous pouvez également la télécharger depuis la page web de Miletus [1] et l'installer manuellement dans Visual Studio Code (fichier *vsix*).
- **Packager (optionnel)** : Vous pouvez le télécharger pour Windows, macOS ou Linux à partir de [4] et l'installer. Il permet d'utiliser le Packager en ligne de commande.

Remarque pour les développeurs web expérimentés : si le gestionnaire de paquets *npm* est installé sur votre système, vous pouvez également installer Miletus avec la commande `npm install miletus`. L'API est alors installée sur le système de développement et vous pouvez l'intégrer dans vos propres projets web.

Voici le fichier *index.html* de notre exemple "Hello, world". Le code source des exemples présentés ici est disponible sur le site web de l'auteur [5].

```
<!DOCTYPE html>
<html lang="de">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="miletus.js"></script>
```

```
<script>
  function save() {
    miletus.dialogs.showSaveDialog();
  }
</script>
</head>

<body>
  <p>Hello World</p>
  <button onclick="save()">Save to File</button>
</body>
</html>
```

Tout est désormais en place. Si vous avez installé l'extension pour Visual Studio Code, le packager sera configuré automatiquement. Les commandes supplémentaires suivantes seront alors disponibles dans Visual Studio Code (**figure 2**) :

- **Miletus — Package application** : crée les packages d'application pour les systèmes cibles Windows, macOS et Linux / Raspberry Pi OS.
- **Miletus — Initialize config** : crée un modèle pour le fichier de configuration *miletus.config.json*.
- **Miletus — Package and run** : crée les packages pour les systèmes cibles sélectionnés et démarre l'application sur le système de développement. Un Raspberry Pi ne peut pas être utilisé comme environnement de développement ; l'application ne peut y être exécutée qu'après le packaging.

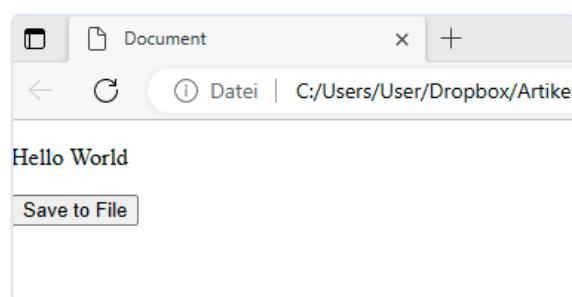


Figure 3. Une application web avec un bouton.



Ouvrez le dossier contenant le fichier *index.html*. Vous pouvez ouvrir ce fichier HTML dans votre navigateur pour le tester (**figure 3**).

Nous envisageons maintenant packager cette appli web en une appli de bureau. Le Packager est contrôlé par le fichier de configuration *miletus.config.json*. Le nom et la version de l'application, les systèmes cibles, le point d'entrée de l'application, et bien d'autres détails sont spécifiés ici. La description détaillée des variables de configuration est disponible dans la documentation [6]. Vous pouvez créer le fichier manuellement ou le générer automatiquement avec Visual Studio Code. Pour ce faire, vous pouvez utiliser le raccourci ([*Ctrl + Shift + P*]) et la commande *Miletus: Initialize config*. Complétez les paramètres, c'est-à-dire attribuez un nom à l'application, attribuez le fichier de démarrage (*index.html*) et sélectionnez les systèmes cibles. Le fichier *miletus.config.json* généré ressemble à ceci :

```
{  
  "name": "Dialogs",  
  "output": "output",  
  "debug": true,  
  "main": {  
    "html": "index.html"  
  },  
  "target": [  
    "win_ia32"  
  ],  
  "include": [  
    "index.html",  
    "miletus.js"  
  ]  
}
```

Les packages d'application destinés aux systèmes cibles sont générés à partir des informations contenues dans le fichier de configuration, par exemple *win_ia32* pour Windows 32 bits. Ceci est réalisé grâce à la commande *Miletus: Package application*. Pour Windows, il s'agit d'un fichier *exe*, pour macOS d'un dossier *.app* et du fichier *entitlements* nécessaire pour gérer les autorisations et pour Linux d'un fichier exécutable et d'un fichier *.desktop* (**figure 4**).

Passons maintenant à la première exécution de l'application web en tant qu'application de bureau sur le système cible. Pour ce faire, nous devons copier les fichiers exécutables sur le système cible, c'est-à-dire :

- **Windows** : Exécutez le fichier *.exe*. Si vous travaillez avec Visual Studio Code sous Windows, vous pouvez exécuter l'application directement avec la commande *Miletus: Package and run*.
- **Linux** : Copiez le dossier *Output* sur le système Linux. Attribuez les droits aux fichiers avec la commande `sudo chmod -R 755 /[Path of the Application]`. Lancez ensuite l'application.
- **macOS** : Vous devez copier les fichiers de l'application depuis le dossier *Output*, attribuer les droits, puis vous pouvez lancer l'application. Remarque : sous un système macOS basé sur ARM, la signature du code de l'application est nécessaire.

Vous pouvez voir l'application "Hello, world" en cours d'exécution dans la **figure 5**.

Remarque : si vous recevez un message d'erreur ou si une fenêtre vide s'affiche, vérifiez les conditions requises pour votre système.

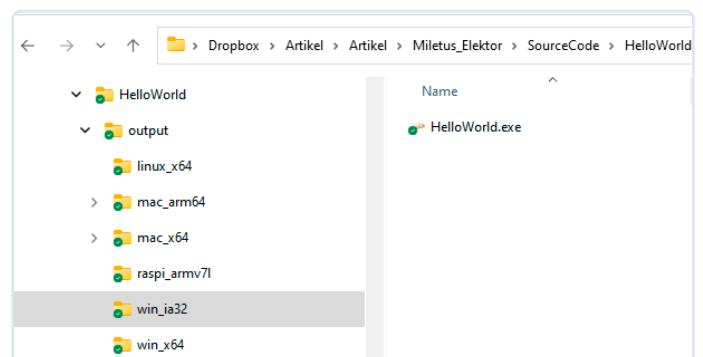


Figure 4. Packages d'applications pour les différents systèmes cibles.

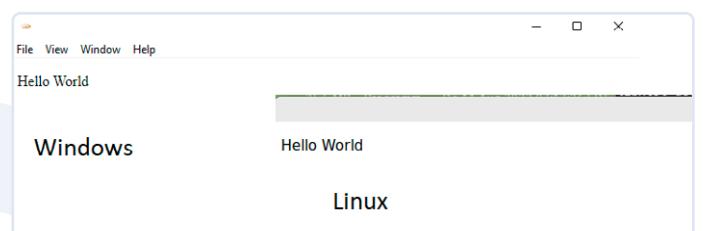


Figure 5. Application "Hello, world" sur Windows et Linux.

Utiliser l'API Miletus

Une application "Hello World" ne suffit pas pour démontrer l'utilité du cadre Miletus. Nous allons donc explorer ses fonctionnalités avec un exemple simple. L'interface utilisateur de l'API Miletus offre, entre autres, les fonctions suivantes :

➤ **Dialogues** : L'API Miletus fournit des fonctions pour afficher des boîtes de dialogue pour ouvrir et enregistrer des fichiers (`showOpenDialog(...)`, `showSaveDialog(...)`), signaler des erreurs (`showErrorDialog(...)`) et afficher des messages (`showMessageBox(...)`). Les fonctions sont accessibles dans le code JavaScript ou dans le code source *TypeScript*. Vous devez d'abord inclure la bibliothèque correspondante *miletus.js*. Ensuite, il est possible d'appeler, par exemple, la boîte de dialogue `Save as...` avec la fonction :

```
miletus.dialogs.showSaveDialog()
```

Pour ce faire, cette ligne de code est intégrée dans une fonction nommée `save()`, qui est à son tour déclenchée par l'événement `onClick()` d'un bouton. La **figure 6** montre l'appel de la boîte de dialogue du fichier.

➤ **Accès aux fichiers** : Miletus propose des méthodes pour charger et écrire des fichiers texte et binaires `loadTextFile(...)`, `saveTextFile(...)`, `loadBinaryFile(...)`, et `saveBinaryFile(...)`. Il existe des méthodes pour le suivi des modifications des fichiers, à savoir `watchFile(...)`, `removeWatch(...)` et `removeAllWatches(...)`, ainsi que la méthode `startDrag(...)` pour les opérations de glisser-déposer. Des méthodes permettant d'afficher tous les éléments d'un dossier de fichiers (`openFile(...)`) et de supprimer des fichiers (`moveToBin(...)`) sont également disponibles.

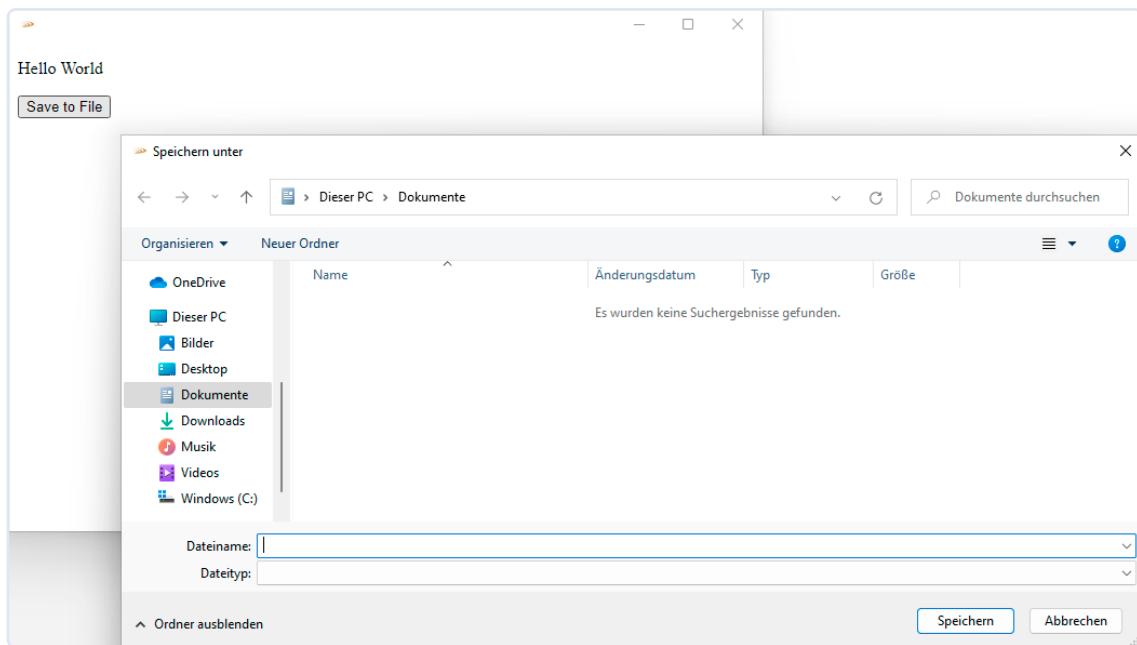


Figure 6. Dialogue de fichier dans Windows - appelé via l'API Miletus.

- **Menus de l'application :** Vous pouvez ajouter des éléments de menu à la fenêtre de l'application. Un menu peut comporter des sous-menus. Une réaction peut être déclenchée lorsque l'utilisateur interagit avec ces éléments. Un menu précédemment défini est attribué à l'application via `application.setMenu(menu)`.
- **Fonctions système :** Vous pouvez charger un URL externe à afficher dans le navigateur via `shell.openURL(...)`. Vous pouvez exécuter une commande système via `execute(...)`.
- **Accès aux bases de données :** Les bases de données suivantes sont prises en charge : SQLite, MySQL, MSSQL, PostgreSQL, MS Access (Windows), Firebird et Interbase. La connexion à la base de données peut simplement être établie via une "chaîne de connexion" sous la forme suivante :

```
let db = new DataBase('sqlite', {database: 'path_to_my/sqlitedb.db'})
```

- **Registre Windows :** Sous Windows, vous pouvez accéder au registre, c'est-à-dire lire, écrire et vérifier les valeurs clés.

D'autres fonctions de l'API Miletus (fenêtre d'application, lecture et écriture de fichiers *ini*, messages système) sont décrites dans la documentation. Une fonctionnalité remarquable est l'accès aux interfaces matérielles du Raspberry Pi que nous aborderons dans la section suivante.

ntation. A special feature is access to the hardware interfaces of the Raspberry Pi. We will look at how this works in the next section.

Une application pour le Raspberry Pi

Nous souhaitons maintenant créer une application capable de lire les capteurs connectés au Raspberry Pi. L'application est développée initialement en tant qu'application web puis sera convertie en un package d'application pour le Raspberry Pi avec Miletus. Nous utilisons la configuration de test suivante :

- Raspberry Pi version 2 ou supérieure
- Capteur environnemental BME280 de Bosch Sensortec (voir le tableau 2 pour les spécifications techniques) ; pour cet exemple, nous utilisons une carte d'expansion de Waveshare.

- Un capteur connecté au Raspberry Pi via l'interface I²C
- Un écran, une souris et un clavier connectés au Raspberry Pi.

Le capteur BME280 mesure la température, l'humidité et la pression atmosphérique. Nous souhaitons lire ces valeurs avec JavaScript et les afficher dans l'application. Le capteur est connecté directement aux broches de la carte Raspberry Pi. Seuls quatre fils sont nécessaires :

Tableau 2. Caractéristiques techniques du capteur BME280 [7].

Caractéristiques	Valeurs
Tension de fonctionnement	5 V/3 V
Interface	I ² C/SPI
Plage de température	-40 à 85 °C, résolution 0.01 °C, précision ±1 °C
Humidité de l'air	0 à 100 RH, résolution 0,008 % RH, précision ±3 % RH, temps de réponse 1 s, retard ≤2 % RH
Pression atmosphérique	300 à 1100 hPa, résolution 0.18 Pa, précision ±1 hPa
Dimensions	27 mm × 20 mm × 2 mm



Figure 7. Montage expérimental avec un Raspberry Pi et un capteur BME280.



Listage 1. index.html of the sensor app.



```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <script src="../js/miletus.js"></script>
    <script src="../js/bme280.js" defer></script>
    <link rel="stylesheet" type="text/css" href="../css/style.css" />
  </head>
  <body>
    <h1>Weather Station</h1>
    <div class="container">
      <div class="child">
        
        <p class="label">Temperature</p>
        <div class="value" id="temperature"></div>
      </div>
      <div class="child">
        
        <p class="label">Pressure</p>
        <p class="value" id="pressure"></p>
      </div>
      <div class="child">
        
        <p class="label">Humidity </p>
        <p class="value" id="humidity"></p>
      </div>
      <div class="buttonscontainer">
        <button id="start-reading" class="btn">Start reading</button>
        <button id="stop-reading" disabled class="btn">Stop reading </button>
      </div>
    </div>
    
  </body>
</html>
```

VCC (rouge) et GND (noir) du module BME280 sont connectés aux broches 2 et 39 du Raspberry Pi. SDA (bleu) et SCL (jaune) sont connectés aux broches 3 et 5 du Raspberry Pi. La **figure 7** montre le montage de test.

Nous installons la dernière version de Raspberry Pi OS sur le Raspberry Pi sur une carte SD depuis un PC. La communication entre les deux composants s'effectue via l'interface I²C, qui doit être activée dans le Raspberry Pi OS. Pour ce faire, ouvrez un terminal sur le Raspberry Pi et entrez la commande :

```
sudo raspi-config
```

Dans l'étape suivante, sélectionnez l'option :

Interfacing Options -> I²C

pour activer le pilote I²C. Après avoir sauvegardé les paramètres et redémarré le Raspberry Pi en utilisant :

```
sudo reboot
```

Il est maintenant possible d'échanger des données via l'interface I²C. Les étapes préliminaires sont terminées. Passons maintenant à la programmation. Le projet se compose des fichiers suivants :

- *index.html* : Ce fichier est le point de départ de l'application web (**listage 1**). Le fichier *index.html* contient le fichier CSS (*style.css*) et les deux fichiers JavaScript *miletus.js* et *bme280.js*.
- *style.css* : La mise en page et la conception de l'application web sont spécifiées en CSS.
- *miletus.js* : contient l'API Miletus.
- *bme280.js* : définit toutes les fonctions de communication avec le capteur.

Tout d'abord, nous avons créé un design initial minimaliste (**figure 8**). Vous pouvez utiliser n'importe quel outil pour ce faire. Une simple esquisse dessinée à la main peut également suffire. Ensuite, nous avons implémenté l'application web avec HTML (structure), CSS (mise en page, conception) et JavaScript (fonctions). Vous pouvez lancer l'application dans un navigateur sur n'importe quel système (**figure 9**) afin de tester son design. Cependant, la fonction-

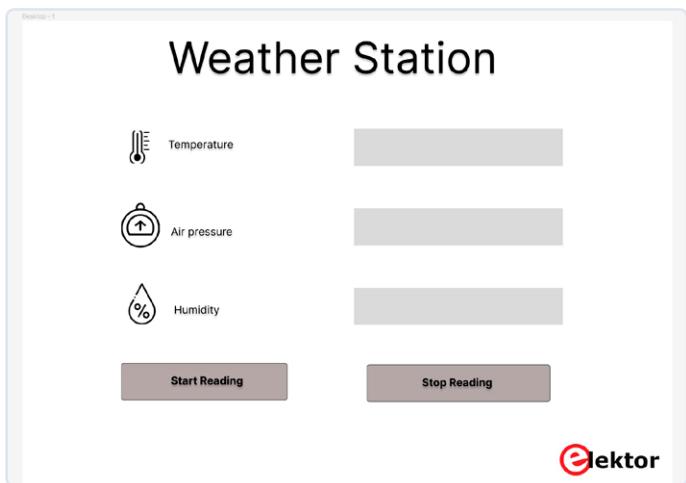


Figure 8. Prototype de l'application (layout et design).

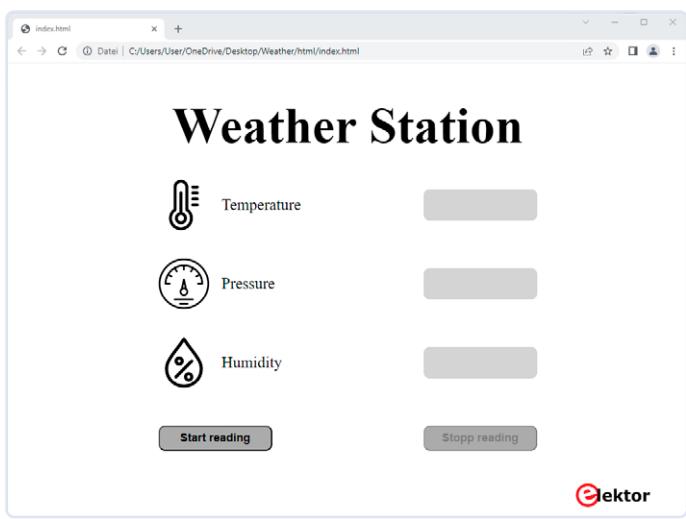


Figure 9. Application météo fonctionnant dans le navigateur.

nalité clé, c'est-à-dire la lecture des données du capteur, ne fonctionne que sur le Raspberry Pi.

Examinons de plus près le code source. Un bouton intitulé *Start reading* est défini dans le fichier HTML :

```
<button class="start" id="start-reading">Start reading</button>
```

Lorsque vous cliquez sur ce bouton, les données du capteur sont censées être lues. L'événement *onclick* est défini dans le fichier *bme280.js* à cette fin :

```
document.getElementById("start-reading").onclick = startReading;
```

Il s'agit de la fonction JavaScript *startReading()* du **listage 2**. Les données du capteur sont calibrées, le bouton *Start* est désactivé, le bouton *Stop* est activé, les relevées sont lus en continu. Cette opération est conforme aux spécifications du capteur BME280, détaillées dans la documentation et dans les exemples [7] et [8]. Par exemple, en examinant la fonction *readSensorData()*, nous trouvons la ligne de code suivante :

```
let rawValues = await i2c.readBuffer(0xFA, 3)
```

Elle est utilisée pour lire les données de température. L'objet *i2c* est défini comme suit :

```
i2c = new miletus.I2C()
```

L'API Miletus permet d'accéder à l'interface I²C du Raspberry Pi. Cela nous permet de lire les données du capteur dans l'application web avec JavaScript. Créons maintenant le fichier de configuration *miletus.config.json* pour notre projet dans Visual Studio Code comme décrit ci-dessus (commande : *Miletus: Initialize config*). Saisissez également les chemins aux fichiers d'images qui doivent apparaître dans l'interface utilisateur (par exemple, les icônes).

Nous pouvons alors générer le package d'application pour le Raspberry Pi (commande : *Miletus: Package application*). Nous copions ensuite les fichiers d'application générés sur le Raspberry Pi, par exemple en utilisant une clé USB. L'attribut "exécutable" doit être défini pour le fichier d'application. Ensuite, démarrez l'application sur le Raspberry Pi. Elle devrait s'ouvrir dans une fenêtre séparée. Cliquez sur le bouton pour lire et afficher les données du capteur (**figure 10**).

Autres possibilités avec le Raspberry Pi

Nous avons étudié la communication via l'interface I²C. Avec Miletus, nous pouvons également accéder aux interfaces suivantes sur le Raspberry Pi [9] :

- **GPIO** : Nous pouvons lire et écrire des données via les 26 broches GPIO.
- **SPI** : Ce bus série nécessite trois fils pour la communication et repose sur le principe maître/esclave.
- **UART** : Ce bus est notamment utilisé pour la communication avec les interfaces RS-232 ou RS-485, par exemple pour le transfert de données avec des microcontrôleurs.

Il est également possible d'accéder à la mémoire tampon. Cela permet de lire et d'écrire la mémoire tampon au niveau du shell de l'application. Voyons un exemple d'échange de données via des GPIO. Importez la bibliothèque *gpio* en utilisant :

```
import { gpio } from 'miletus'
```

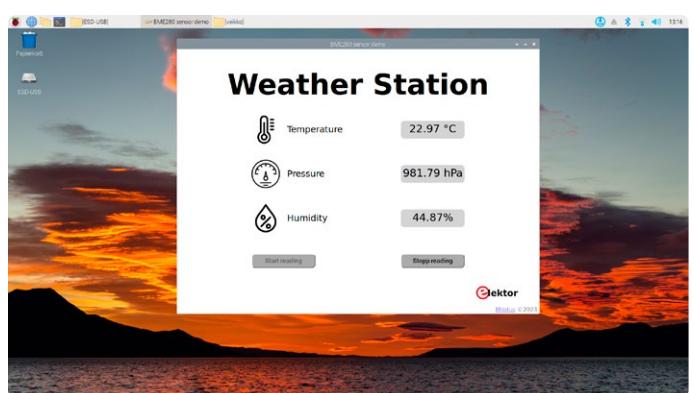


Figure 10. Application météo sur le Raspberry Pi.



Listage 2. Function for reading out the sensor data.

```
async function startReading() {
    await readCalibrationData();
    document.getElementById("start-reading").disabled = true;
    document.getElementById("stop-reading").disabled = false;
    doRead = true;
    while (doRead) {
        await readSensorData();
        await new Promise(resolve => setTimeout(resolve, 1000));
    }
}
```



Produits

➤ **Raspberry Pi 4 B (2 GB RAM)**
www.elektor.fr/18965

➤ **Raspberry Pi 5 (4 GB RAM)**
www.elektor.fr/20598

Nous pouvons maintenant utiliser les méthodes `setPin(...)` pour définir le mode de la broche (entrée ou sortie), utiliser `write(...)` pour activer ou désactiver une broche (signal haut ou bas) et utiliser `read(...)` pour lire une broche. Pour garantir que l'interface utilisateur de l'application web reste réactive pendant l'échange de données avec l'interface matérielle, l'accès doit être asynchrone. Par exemple, l'accès doit être asynchrone :

```
await gpio.setPin(1, 'read')
```

Ceci met la broche du port GPIO en mode lecture.

Extensibilité

Le cadre Miletus offre une grande flexibilité d'extension. Vous pouvez charger n'importe quelle bibliothèque de Miletus et ensuite accéder aux fonctions qu'elle exporte depuis l'application web. Cela inclut l'accès aux fichiers `.dll` (Windows), `.dylib` (macOS) et `.so` (Linux/Raspberry Pi). Ainsi, les fonctions système des systèmes cibles peuvent être utilisées dans une application web packagée avec Miletus, par exemple pour contrôler le matériel connecté au PC via des pilotes système. Vous pouvez trouver plus d'informations sur ce sujet à l'adresse suivante : [10].

Réduction de l'effort de programmation

La méthode, qui peut sembler inhabituelle, et qui consiste à transformer une application web en une application native offre de nombreux avantages. Elle permet à l'application de s'exécuter sur tous les systèmes (multiplateforme), de se comporter comme une application native, d'être utilisée hors ligne et d'avoir accès aux fonctions du

système. Sur le Raspberry Pi, cela permet à l'application d'envoyer et de recevoir des données via les interfaces. Cela signifie que vous pouvez utiliser une application web pour contrôler des circuits électroniques. Cette approche peut considérablement simplifier et réduire l'effort de programmation pour de nombreuses applications, surtout si vous souhaitez couvrir plusieurs systèmes simultanément. ↗

220616-04



À propos de l'auteur

Veikko Krypczyk est développeur de logiciels, formateur et auteur spécialisé. Il partage sa passion et son expertise lors de séances de coaching, de séminaires, de cours de formation et des ateliers. Vous pouvez demander des ateliers personnalisés ou obtenir du soutien en visitant <https://larinet.com> et consulter l'agenda des événements à venir.

Questions ou commentaires ?

Contactez elektor (redaction@elektor.fr).

LIENS

- [1] Miletus : <https://miletus.org>
- [2] Electron : <https://electronjs.org>
- [3] Visual Studio Code : <https://code.visualstudio.com>
- [4] Miletus : téléchargement du Packager : <https://miletus.org/download.html>
- [5] Téléchargement du code d'exemple : https://wp.larinet.com/?page_id=663
- [6] Miletus : variables de configuration du Packager : <https://miletus.org/doc/gettingstarted/packager>
- [7] BME280 Fiche technique : https://waveshare.com/w/upload/9/91/BME280_datasheet.pdf
- [8] BME280 Bibliothèque du pilote : https://github.com/BoschSensortec/BME280_driver/blob/master/bme280.c
- [9] Miletus : Accès aux GPIO: <https://miletus.org/doc/reference/gpio>
- [10] Extensions de Miletus : <https://miletus.org/doc/gettingstarted/extensibility>



de la 4G à la 5G

est-ce une étape si facile à franchir ?

Questions de Roberto Armani (Elektor)

Chaque jour, nous sommes inondés de publicités sur les merveilles des récentes concessions 5G. Il ne fait aucun doute qu'elles apportent une contribution essentielle à la vitesse du réseau, mais tout ce qui brille n'est pas vraiment de l'or, non ? Dans cet entretien avec Nemo Galletti, directeur italien d'une entreprise spécialisée dans la production d'antennes pour les réseaux de télécommunications, nous examinons les différences entre la 5G et l'ancienne technologie 4G. Nous vérifions également si quelque chose n'a pas été « caché sous le tapis » par les fournisseurs.

Roberto Armani : Nemo, pouvez-vous nous parler de votre expérience dans le domaine des télécommunications ?

Nemo Galletti (Radio Frequency Systems) : 2024 est ma 40^e année dans l'industrie des télécommunications. Je travaille actuellement chez Radio Frequency Systems, une société dirigée par Nokia, et je suis spécialisé dans les composants passifs principalement déployés dans les réseaux mobiles, tels que les alimentations spécifiques, les antennes, les guides d'ondes et les câbles rayonnants.

Roberto : Comment décririez-vous la 5G en quelques mots ?

Nemo : La 5G désigne la cinquième génération de réseau mobile sans fil. Par rapport aux réseaux mobiles de la génération précédente, ses avantages les plus importants peuvent être résumés en trois caractéristiques clés : la vitesse, la capacité et une latence plus faible.

Roberto : Dans le détail, en quoi la technologie 5G diffère-t-elle des générations précédentes (c'est-à-dire la 4G) ?

Nemo : Il y a des aspects technologiques essentiels

apportés par les réseaux 5G, qui diffèrent des générations de réseaux mobiles précédentes :

➤ **Vitesse de transfert des données :** Par rapport à la 4G, la 5G offre des vitesses de transfert de données nettement plus élevées. Alors que les réseaux 4G offrent des vitesses de téléchargement allant jusqu'à plusieurs centaines de mégabits par seconde en général, les réseaux 5G peuvent atteindre des débits de plusieurs gigabits par seconde. Toutefois, au lieu de « peuvent atteindre », nous devrions plutôt dire « pourraient atteindre », comme nous l'expliquons plus loin.

➤ **Temps de latence :** La latence fait référence au temps nécessaire pour que les données voyagent d'un appareil à l'autre sur un réseau. Alors que les réseaux 4G ont généralement un temps de latence de l'ordre de quelques dizaines de millisecondes, les réseaux 5G peuvent atteindre un temps de latence très faible de l'ordre de quelques millisecondes seulement.

➤ **Capacité du réseau :** Les réseaux 5G offrent une capacité de réseau nettement supérieure à celle de la 4G. Cela signifie que les réseaux 5G peuvent gérer simultanément un nombre beaucoup plus important d'appareils connectés sans une dégradation des performances.

➤ **Efficacité spectrale :** La technologie 5G est conçue pour être plus performante au niveau spectral que les technologies précédentes. Cela signifie qu'elle peut transmettre plus de données en utilisant la même quantité de spectre radio. La 5G permet cela grâce à des techniques de modulation avancées, à une utilisation plus efficace des bandes de fréquences disponibles et à l'utilisation de bandes de fréquences plus élevées.

➤ **Découpage du réseau :** La 5G introduit le concept de découpage du réseau, qui permet aux opérateurs de créer plusieurs réseaux virtuels au sein d'une seule infrastructure de réseau physique. Chaque tranche de réseau peut être optimisée pour des cas d'utilisation ou des applications spécifiques, telles que le haut débit mobile amélioré, la communication ultra-fiable à faible latence et les déploiements massifs de l'IdO.



Figure 1. Tableau de l'utilisation mondiale des fréquences 5G. (Source de toutes les images, sauf indication contraire : Radio Frequency Systems)

➤ **Edge computing :** Les réseaux 5G permettent des capacités d'*Edge computing*, en rapprochant les ressources de calcul de la périphérie du réseau. Cela réduit la distance que les données doivent parcourir, améliorant la latence et permettant un traitement plus rapide pour les applications qui nécessitent une analyse des données en temps réel ou des réponses à faible latence.

➤ **Amélioration de la connectivité :** La 5G prend en charge des fonctions de connectivité avancées, telles que le filtrage spatial et les systèmes d'antennes MIMO massives. Ces technologies améliorent la qualité du signal, augmentent la couverture du réseau et améliorent la capacité et les performances globales du réseau. Le filtrage spatial permet au réseau de concentrer son signal sur des appareils spécifiques, tandis que le système MIMO utilise plusieurs antennes pour améliorer la qualité et la capacité du signal.

➤ **Fréquences d'ondes millimétriques :** La 5G introduit l'utilisation de bandes de fréquences plus élevées, y compris les fréquences d'ondes

millimétriques (mmWave). Ces bandes à haute fréquence offrent des largeurs de bande beaucoup plus importantes, ce qui permet des débits de données plus rapides.

Roberto : On associe souvent la 5G au déploiement de nouvelles fréquences radio. Quelles sont les bandes de fréquences existantes ou qui seront déployées pour les réseaux 5G ? Sont-elles différentes selon les pays ?

Nemo : Les organismes de normalisation de la 5G ont défini des bandes spécifiques qui seront utilisées par la 5G. Elles sont identifiées par la lettre *n* suivie d'un chiffre. Les principales bandes de fréquences *nXX* existantes ou qui seront attribuées à la 5G sont énumérées dans la **figure 1** (mais de nombreuses autres bandes peuvent être utilisées). Pour *n78*, la bande la plus utilisée, des valeurs plus précises ont été indiquées dans le même tableau.

Le tableau de la figure 1 est une simplification. L'Europe n'adopte pas les mêmes fréquences dans tous les pays. Par exemple, l'Allemagne ne prévoit pas d'utiliser la bande *n41* pour la 5G, tandis que

Tableau 1. Codes de bandes de fréquences mondiales et attribution des fréquences.

Code de bande de fréquence	Fréquence (GHz)	Europe	USA/Canada	Chino	Japan	Corée	Australie
28	0,7	x	x		x		x
n40	2,3						x
n41	2,5		x	x			
n78	3,5	3,4-3,8	3,7-4,3	3,3-4,9	3,6-4,9	3,4-3,7	3,5
n257	28	x	x	x	x	x	x
n258	26	x	x				x



Figure 2. Une tour 4G (5G) typique pour les bandes 700 MHz et 1 800 MHz, avec une vue détaillée de l'intérieur d'un réflecteur.

les sous-bandes de la bande la plus utilisée, la n78, seront probablement différentes d'un pays à l'autre. La tendance en Europe est de rester en dessous de 3,8 GHz dans cette bande. Étant donné que les diverses fréquences utilisées par la 5G sont très différentes, on pourrait plutôt se demander laquelle est la meilleure. Nous avons vu que la 5G utilise des fréquences basses, moyennes et hautes. Une basse fréquence (définie comme une émission électromagnétique avec une fréquence de l'ordre de centaines de MHz), a la capacité de pénétrer les obstacles et d'atteindre une distance beaucoup plus grande qu'une haute fréquence, mais peut transporter moins de données par unité de temps (typiquement exprimées en bits par seconde ou bit/s). Une haute fréquence (de l'ordre du GHz), au contraire, a une portée beaucoup plus faible mais peut transporter beaucoup de données par unité de temps. C'est pourquoi une connexion peu fiable (c'est-à-dire perturbée) à une bande de haute fréquence peut parfois être plus lente qu'une bande de fréquence plus basse mais avec une bonne connexion.

Par conséquent, nous ne pouvons pas affirmer qu'une bande de fréquences est meilleure qu'une autre, car il existe différentes fréquences pour différents usages, et le meilleur réseau 5G est celui qui fait bon usage des

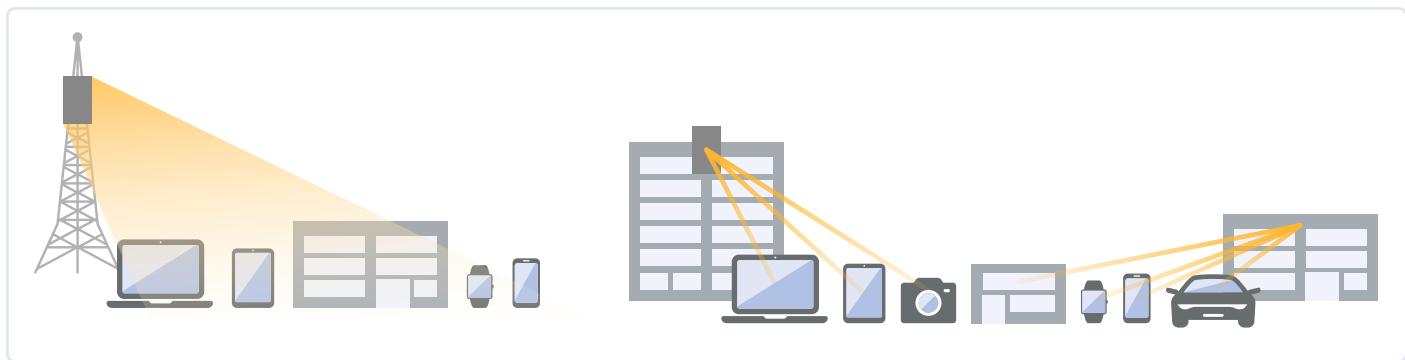
trois bandes de fréquences en fonction des besoins spécifiques du moment. Un flux vidéo, l'accès à des e-mails, ou un thermostat intelligent connecté à une maison intelligente, ont des exigences différentes en termes de bande passante !

C'est aussi pour cette raison que les nouveaux smartphones 5G utilisent une technologie appelée « commutation adaptative de faisceau », qui leur permet de passer d'une bande de fréquence à l'autre pour maintenir une connexion stable. Cette fonction permet de maximiser les avantages d'une connexion via plusieurs fréquences en même temps. C'est particulièrement vrai pour les fréquences millimétriques, comme la bande haute n257/n258, la très haute fréquence permet un débit de données extrêmement élevé, mais même un temps nuageux ou brumeux peut avoir un impact négatif sur un faisceau d'ondes millimétriques. Sans commutation adaptative du faisceau, nous ne pourrions pas confier notre connexion à une transmission millimétrique. Les fréquences 5G les plus utilisées, du moins dans un premier temps, seront les fréquences moyennes, ou « sub 6 », qui sont comprises entre 1 et 6 GHz. Par rapport aux ondes millimétriques, ces fréquences permettent un bon débit de données, une pénétration suffisante pour traverser les murs, et sont moins sujettes aux interférences. Enfin, les basses fréquences seront idéales pour les appareils, les capteurs, l'IdO, les contrôleurs et les maisons intelligentes, où un faible débit de données suffit, mais où une forte pénétration et une faible latence sont nécessaires.

Roberto : Le débat sur les fréquences inclut également celui des antennes 5G. Sont-elles différentes de celles de la 4G ?

Nemo : En principe, les antennes 4G et 5G ne sont théoriquement pas différentes. Pourtant, elles sont impactées par certaines technologies innovantes introduites avec la 5G. Nous nous concentrerons ici sur deux éléments, dont l'un est physique, c'est-à-dire que la taille du dipôle doit être aussi proche que possible de la moitié de la longueur d'onde (λ). La formule simplifiée pour calculer la longueur optimale (en mm) d'un dipôle $\lambda/4$ est la suivante : vitesse de la lumière [m/s] / fréquence [kHz] / 2.

Les dipôles peuvent avoir la même taille ou un multiple de la longueur d'onde, leur forme peut être linéaire, en V ou coudée, mais leur taille est toujours proportionnelle à la longueur d'onde. Nous avons vu que la fréquence la plus basse pour la 5G est de 700 MHz, ces dipôles seront donc plus longs que ceux utilisés à 800 MHz par la 4G, alors que pour les ondes millimétriques, la 5G déploie une matrice composée de plusieurs réseaux de dipôles dont la taille est minimale. Pour donner un ordre d'idée, à 700 MHz, la longueur typique d'un dipôle $\lambda/4$ est de 214 mm, alors qu'à 28 GHz, elle est de 5,35 mm.



Le deuxième élément est lié à la technologie, car l'antenne 4G « traditionnelle » se compose d'un réseau vertical de dipôles pour chaque fréquence, monté sur un réflecteur. Sur la **figure 2**, vous pouvez voir une station cellulaire typique avec un ensemble de réflecteurs montés selon le schéma d'émission classique de 120°, tandis que l'image encadrée dans le coin inférieur droit montre la structure interne d'un réflecteur. Dans ce cas, nous voyons à l'intérieur les dipôles à polarisation croisée de 700 MHz (éléments dorés), et à l'intérieur de ceux-ci, les dipôles blancs plus petits pour la bande de 1,8 GHz. À partir de la bande moyenne, les antennes 5G sont souvent constituées d'un réseau de matrice d'antennes, avec plusieurs antennes verticales en parallèle, initialement 6×6 ou 8×8, jusqu'à 32×32 dans la version 15, et plus dans les versions ultérieures. La quantité et la taille des matrices augmentent avec la fréquence, gérées en mode actif de manière à réaliser le MIMO Massif et la formation de faisceaux. Il en résulte que les antennes pour les bandes basses ne seront pas très différentes de celles que nous voyons pour la 4G, mais les antennes typiques de la 5G pour les bandes plus élevées seront comme des boîtes rectangulaires ou carrées, devenant de plus en plus petites avec l'augmentation de la fréquence.

Roberto : Vous avez mentionné le MIMO Massif, la formation de faisceaux et les matrices d'antennes dipolaires. S'agit-il de nouvelles technologies ?

Nemo : Le MIMO massif et la formation de faisceaux par le biais de matrices de dipôles ne sont pas des technologies nouvelles. Cependant, lorsque nous parlons de déploiement mobile commercial, la première apparition de ces technologies a eu lieu il y a environ 20 ans avec les réseaux Wimax. Pour plusieurs raisons, les réseaux Wimax n'ont pas évolué comme prévu, de sorte que cette approche est restée confinée à des applications spécifiques. Ce n'est qu'avec la 5G que nous voyons l'utilisation étendue de ces solutions. MIMO (entrées multiples, sorties multiples) utilise les techniques de la diversité spatiale et du multiplexage spatial. Les signaux de données indépendants et codés séparément, appelés « flux », sont transmis par différentes antennes en réutilisant la même période de temps et la même ressource de fréquence. Dans le MIMO multi-utilisateurs (MU-MIMO), l'émetteur envoie simultanément différents flux à différents utilisateurs, en utilisant les mêmes ressources temporelles et fréquentielles. L'efficacité spectrale et

la capacité peuvent être améliorées en ajoutant des antennes supplémentaires pour prendre en charge davantage de flux. L'effet MIMO repose sur le fait qu'un signal radio entre l'émetteur et le récepteur est influencé par son environnement, les réflexions sur les bâtiments et autres obstacles entraînant de multiples trajets de signaux. Les divers signaux réfléchis atteindront l'antenne de réception avec des retards, une atténuation et une direction de déplacement variables. Lorsque plusieurs antennes de réception sont déployées, chacune d'entre elles reçoit une version légèrement différente du signal, qui peut être combinée mathématiquement pour améliorer la qualité du signal transmis. On parle alors de « diversité spatiale ». On y parvient également en transmettant le signal radio par l'intermédiaire de plusieurs antennes, chaque antenne envoyant, dans certains cas, des versions du signal modifiées.

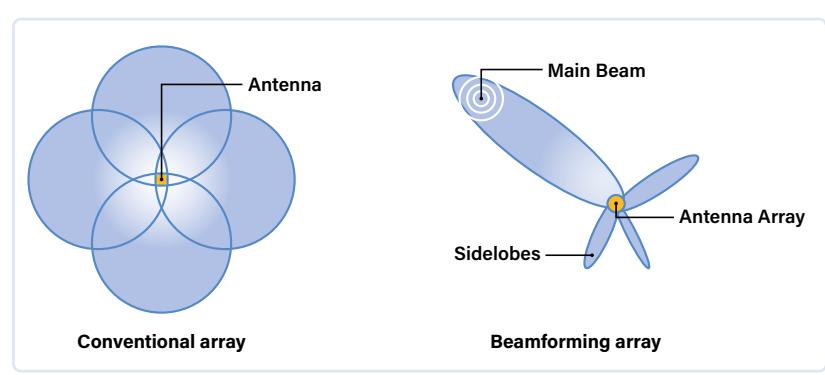
En plus de la diversité spatiale, le multiplexage spatial augmente la capacité de la liaison radio en utilisant les multiples chemins de transition comme des canaux supplémentaires pour le transport des données. Le multiplexage spatial permet d'envoyer des flux de données multiples et uniques entre l'émetteur et le récepteur, ce qui augmente considérablement le rendement, et permet également à plusieurs utilisateurs du réseau d'être pris en charge par un seul émetteur. Comme le montre la **figure 3**, la formation de faisceaux utilise des technologies d'antenne avancées pour concentrer un signal mobile dans une direction spécifique, plutôt que de le diffuser dans une zone étendue.

Roberto : Pourriez-vous nous expliquer plus en détail son principe de fonctionnement ?

Nemo : La **figure 4** montre les diagrammes de rayonnement d'une antenne dipôle standard (à gauche) et

Figure 3. Différence entre l'empreinte de diffusion standard d'une tour 4G (à gauche) et celle d'une tour 5G moderne et optimisée, dotée de capacités d'orientation du faisceau (à droite).

Figure 4. Diagrammes de rayonnement d'une antenne réseau dipôle standard (à gauche) et d'une antenne réseau 5G de dernière génération (à droite) avec capacité de formation de faisceau.



d'une antenne réseau 5G de dernière génération (à droite) avec des capacités de formation de faisceau. La technologie de formation de faisceau permet de concentrer le signal mobile dans une direction spécifique, plutôt que de le diffuser dans une zone étendue. Cette technique réduit les interférences entre les faisceaux dirigés dans des directions différentes, ce qui permet de déployer des réseaux d'antennes plus importants, et est donc associée à la technologie MIMO massive.

La formation de faisceaux tridimensionnels, facilitée par le grand nombre d'antennes dans un système MIMO massif, crée des faisceaux horizontaux et verticaux vers les utilisateurs, augmentant les débits de données à la demande, en « suivant » la requête d'un utilisateur mobile. Des algorithmes complexes ont été développés pour coordonner les informations spatiales obtenues à partir d'un *signal de référence sur l'état du canal* (CSI-RS) afin de permettre à la station de base de communiquer avec plusieurs appareils en même temps et de manière indépendante. Le CSI-RS est un signal envoyé par la station de base à l'*élément utilisateur* (UE), permettant à l'UE de calculer les informations sur l'*état du canal* (CSI) et de les renvoyer à la station de base. Ces informations sont utilisées par le système MIMO pour effectuer un grand nombre de traitements de signaux, en utilisant les CSI pour représenter la fonction de transfert du canal sous la forme d'une matrice.

Roberto : Quelle est la situation de l'Europe par rapport aux autres pays en ce qui concerne le déploiement de la 5G ?

Nemo : Depuis 2016, l'UE a établi un plan général de déploiement de la 5G, fixant les étapes suivantes :

- Premiers réseaux d'essai d'ici à la fin de 2018.
- Services commerciaux 5G complets dans au moins une grande ville d'ici à la fin de 2020.

Figure 5. Prévision mondiale de la progression de la 5G d'ici à 2025. (Source : GSMA).

➤ Couverture 5G complète dans les zones urbaines et continuité de la couverture sur les principaux axes de transport (routes, chemins de fer) d'ici à la fin de 2025.

➤ En mars 2021, une nouvelle étape a été franchie : couverture 5G de toutes les zones habitées d'ici à la fin de 2030.

À la fin de l'année 2020, 23 pays de l'UE ont atteint l'objectif d'avoir au moins une grande ville entièrement couverte par la 5G. Quatre pays seulement n'ont pas pu atteindre cet objectif.

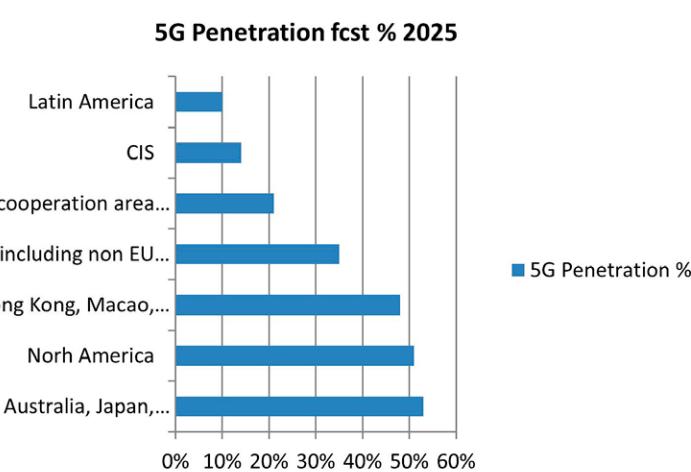
Selon les dernières études, la Commission européenne prévoit que 11 pays seulement atteindront la fin de l'année 2025 avec une couverture 5G complète de toutes les zones urbaines et des principaux axes de transport. En d'autres termes, le déploiement de la 5G en Europe progresse lentement, et la liste des 13 pays qui seront à la traîne comprend également des noms inattendus (à vous d'identifier les « surprises ») : l'Autriche, la Tchéquie, l'Estonie, l'Allemagne, l'Irlande, la Pologne, la Lituanie, la Slovénie, la Belgique, la Bulgarie, la Croatie, Chypre et la Grèce.

Selon les prévisions de la GSM Association, d'ici à la fin de 2025, le pourcentage de connexions mobiles 5G par rapport au total des connexions sera illustré dans le diagramme de la **figure 5**.

Est-ce que c'est bon ? Certainement pas, dirais-je. L'Europe obtiendra un score misérable de 35 %. En plus, il est important de considérer que la définition de « la pénétration de la 5G » en termes de nombre de connexions 5G est un concept générique, puisque la 5G est mise en œuvre en plusieurs phases (ou étapes) le long d'un calendrier. Donc le « nombre de connexions 5G » ne signifie pas que les services 5G seront disponibles et utilisés sur une connexion 5G en réalité.

Roberto : Quelles sont les étapes de la mise en œuvre de la 5G ?

Nemo : Elles sont conçues par le 3GPP, l'organisme de normalisation des télécommunications mobiles le plus important au monde. Il regroupe sept organismes d'élaboration de normes de télécommunications comme ARIB, ATIS, CCSA, ETSI, TSDSI, TTA et TTC, qui fournissent les spécifications pour les technologies de télécommunications cellulaires, y compris l'accès radio, le réseau central et les fonctionnalités de service. Le 3GPP a basé le déploiement de la 5G sur des phases qui peuvent coexister et qui aboutiront à la mise en œuvre « finale » de la 5G dans un délai qui dépend de la rapidité des investissements dans l'infrastructure. Aujourd'hui, chaque opérateur est positionné à une étape intermédiaire sur ce chemin dans chaque pays. Malheureusement, je peux dire que l'Europe est généralement en retard par rapport à d'autres pays, tels que les États-Unis, la Chine et la Corée du Sud,

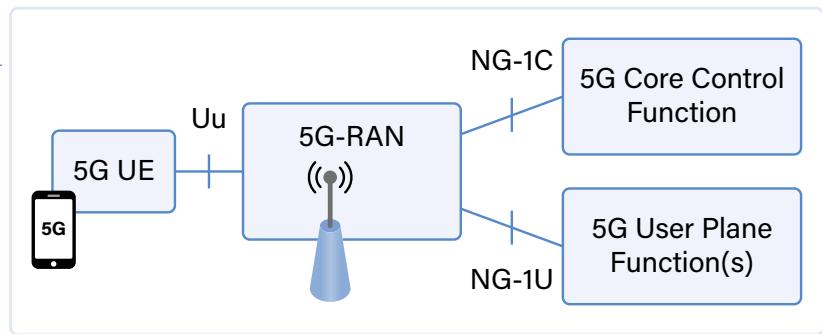


qui sont plus avancés dans le déploiement de la 5G. En ce qui concerne les phases 15, 16, 17 et 18 de l'évolution de la 5G, dont la normalisation a été achevée par le 3GPP, l'Europe devrait avoir commencé le déploiement de la phase 18 d'ici à la fin de 2023 (voir le **tableau 2**).

Roberto : Alors, où en est la 5G aujourd'hui ?

Nemo : Dans la plupart des cas, à l'exception des zones d'essai particulières, la 5G se situe entre la phase 15 et la phase 16.

La première étape de la 5G est appelée 5G NR (New Radio). Il s'agit en effet d'une mise à niveau complexe, mais ce n'est que la première étape. Pour simplifier, la 5G NR est liée à l'accès radio et à tous ses protocoles et interfaces de contrôle. Comme le montre la **figure 6**, l'élément utilisateur 5G est connecté au réseau d'accès radio 5G via l'interface NR-Uu. Le réseau d'accès radio (Radio Access Network) de la 5G, constitué de gNB (nœud B de la 5G, l'unité de bande de base), a été conçu pour avoir ses propres fonctions de contrôle, mais cette fonction de contrôle est mise en œuvre de manière native par quelques réseaux seulement. Étant donné que le déploiement complet d'une fonction de contrôle centrale 5G nécessitera du temps et des investissements massifs, la plupart des opérateurs mobiles mettent en œuvre la première phase de la fonction de contrôle 5G sur la fonction de contrôle centrale 4G. Nous parlons de 5G-NSA (non-standalone) pour faire la différence avec la (future) 5G-SA (standalone). Dans le cas de la 5G-NSA, l'expérience



de l'utilisateur ne bénéficiera que d'une amélioration de 15 à 50 % du débit de données et d'une réduction de la latence, ce qui est encore loin des améliorations spectaculaires promises par la 5G-SA finale. Mais, car il y a un « mais ». Même pour la 5G-NSA, il faut une condition préalable, avec une couverture réseau 4G pleinement efficace. Or, certains pays sont loin de fournir une couverture 4G complète, dans les zones rurales ou même le long des principaux axes de communication à l'intérieur de ces pays (autoroutes, chemins de fer).

Roberto : D'après vos propos, je comprends que la transition ne se fait pas sans difficulté. Y a-t-il peut-être un excès de prudence de la part des fournisseurs, qui entraîne des retards dans la mise en œuvre de la 5G ?

Nemo : La 5G est très ambitieuse et nécessite d'énormes investissements. Pour les entreprises privées, ces investissements sont directement liés aux revenus escomptés. Et c'est là que se trouve le problème, du moins en Europe, car le retour sur investissement n'est pas satisfaisant, du fait que dans plusieurs pays, les gens sont satisfaits des

Tableau 2. Phases de déploiement de la 5G (lancements).

Phase (Lancement)	Année de mise en œuvre ou de définition	Objectif	Statut (mars 2024)
15	2018	Prise en compte des concepts d'accès à la nouvelle radio (NR).	Terminé.
16	2020	Achèvement de la NR, y compris les fréquences sans licence et l'accès par satellite.	Terminé.
17	2022	Définition de services supplémentaires tels que MIMO amélioré, amélioration du partage du spectre, amélioration de la couverture, inclusion de bandes de fréquences jusqu'à 71 GHz, prise en charge améliorée des réseaux privés, IoT industriel, dispositifs NR à faible complexité, edge computing, pilotage du trafic d'accès, prise en charge de la commutation et du fractionnement, automatisation du réseau pour la 5G, découpage du réseau, service V2X avancé, prise en charge de plusieurs USIM et autres services.	Non achevé. Son activation est toujours en cours et affectée par plusieurs retards.
18	2023	Également définie comme la Advanced 5G. Les nouvelles spécifications comprennent l'edge computing, l'énergie et l'infrastructure intelligentes, les relais embarqués, l'évolution du service de téléphonie multimédia IMS, le positionnement à faible consommation et à haute précision pour les appareils IoT industriels, l'amélioration du découpage du réseau.	Définition des spécifications achevée. Reste encore à mettre en œuvre.
19	2024	Le 3GPP travaille actuellement à la définition des caractéristiques de cette version Advanced 5G.	Ni entièrement définie ni mise en œuvre.

Figure 6. L'UE (élément utilisateur) 5G connecté avec le 5G-RAN (Radio Access Network) avec ses propres fonctions de base (-1C) et d'utilisateur (-1U).

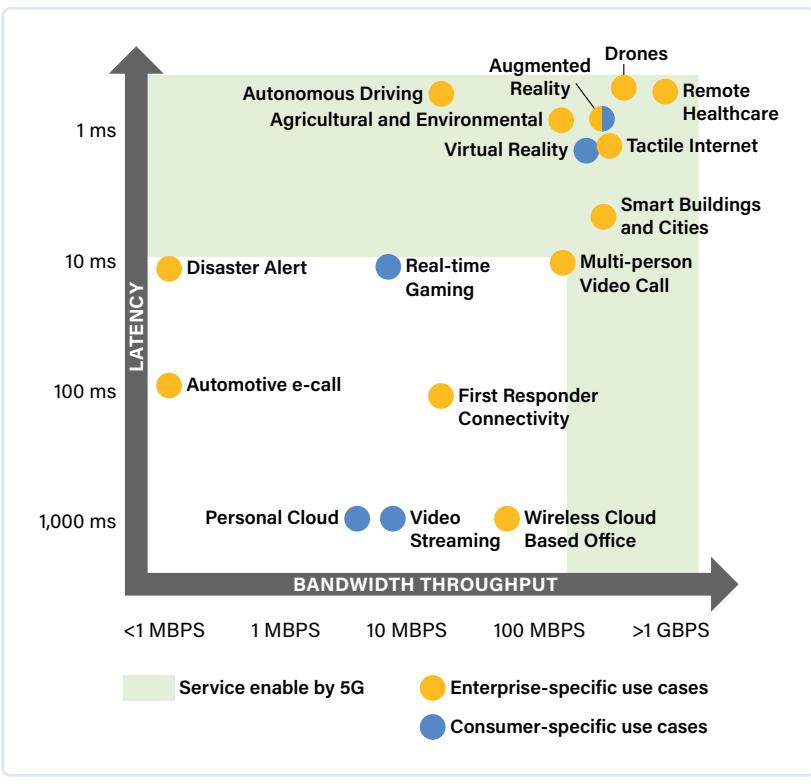


Figure 7. Domaines d'application de la 4G (blanc) et de la 5G (vert clair), en fonction de la combinaison de la latence et du débit de la connexion. (Source : GSMA)

performances de la 4G et ne sont pas prêts à dépen-
ser plus pour les quelques améliorations proposées
par la 5G-NSA. D'autre part, les recettes attendues
provenant principalement des entreprises et des
nouveaux marchés, nécessitent des services qui ne
seront développés qu'avec la 5G-SA.

Le diagramme de la **figure 7**, dérivé de l'analyse du
groupe GSMA, montre où se situent les véritables
attentes de la 5G en termes de « fournisseur de
services ». La zone blanche correspond à ce que l'on
peut obtenir avec un réseau 4G efficace. La zone vert
clair correspond aux services que la 5G permettra
d'offrir.

Aujourd'hui, nous vivons le paradoxe de « l'oeuf et la
poule », avec des revenus massifs qui proviennent
de cas d'utilisation spécifiques aux utilisateurs, et
nous pouvons voir que pour l'instant, il n'y a pas de
demande massive de la part de ce groupe d'utilisateurs
pour des services 5G spécifiques. Dans le même

temps, une guerre des prix en cours dans plusieurs
pays européens (principalement en France et en
Italie) fait baisser l'ARPU (revenu moyen par utilisa-
teur). La conséquence est que les consommateurs
habituel sont réticents à payer un supplément, pour
une augmentation des fonctionnalités dont ils n'ont
pas vraiment besoin. Lorsque nous examinons les
besoins typiques des entreprises (ou des gouverne-
ments), nous nous intéressons à des services qui ne
sont pas encore mis en œuvre par la 5G-NSA et qui
seront disponibles principalement avec la 5G-SA.
Cela nécessite un développement et des investisse-
ments supplémentaires, exigeant un financement que
plusieurs opérateurs de réseaux ont du mal à obtenir
à partir des niveaux de revenus actuels.

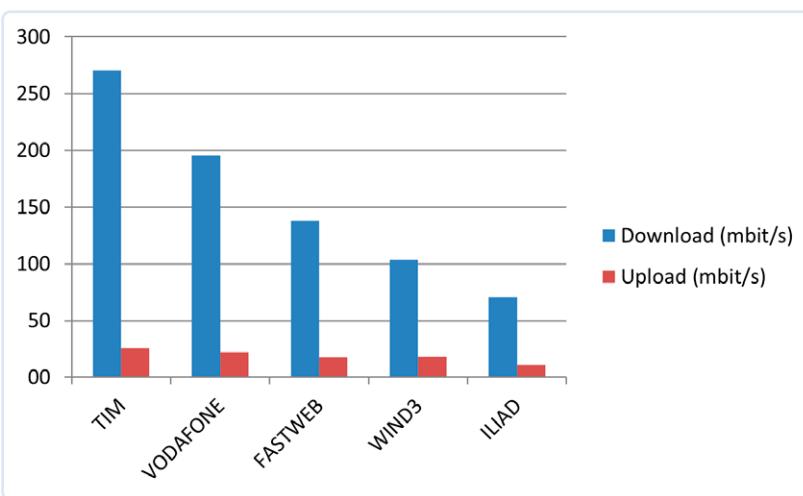
Pour en revenir au sujet initial sur la différence entre la
4G et la 5G, je dirais qu'aujourd'hui la situation est pour
le moins floue. Les réseaux 4G ont des performances
très différentes dans le monde, tandis que les perfor-
mances (débit de données par heure) des réseaux 5G,
qui sont pour la plupart toujours basés sur le cœur
4G, diminuent à mesure que le nombre d'utilisateurs
augmente. Et ces performances montrent d'énormes
différences dans le monde : de 500 Mbit/s en Corée
à 300 Mbit/s en Chine jusqu'à la moyenne actuelle
d'environ 200 Mbit/s en Italie, mais avec d'énormes
différences entre les opérateurs, comme le montre le
tableau de la **figure 8**.

En général, la 5G « pourrait être » trois à cinq fois plus
rapide que la 4G, ou deux fois plus rapide que la 4.5G
dans l'agrégation de portées. Mais en essayant de
faire une comparaison efficace, le problème est à la
fois la vitesse de référence de la 4G et celle de la 5G.
Les avantages de la 5G se mesurent en fait en combi-
nant trois facteurs qui sont la vitesse, la latence et
la capacité du réseau. Les considérer séparément ne
donne pas une idée du progrès que cela représente
par rapport à la 4G.

Un autre exemple de l'état de la couverture 5G, mise
à jour en mars 2024, est présenté à la **figure 9**. La
carte de la couverture 5G de Vodafone Allemagne
entre Hanovre, Hambourg et Berlin (le violet corres-
pond à la 5G, l'orange et le rouge à la 4G et à la 4.5G,
respectivement) montre qu'ils sont encore loin des
objectifs fixés.

Roberto : Devrais-je dire, alors, « Oublions le déploiement de la 6G pour l'instant » ?

Nemo : Pas vraiment. La situation que j'ai présen-
tée semble peut-être pessimiste, mais je pense que
nous ne vivons qu'un ralentissement temporaire du
déploiement de la 5G. La 5G offre, en principe, des
fonctionnalités et des services fantastiques. Ce qu'il
faut maintenant, c'est attendre la croissance inévi-
table de la demande de ces services, afin de dispo-
ser des revenus nécessaires pour que les réseaux 5G



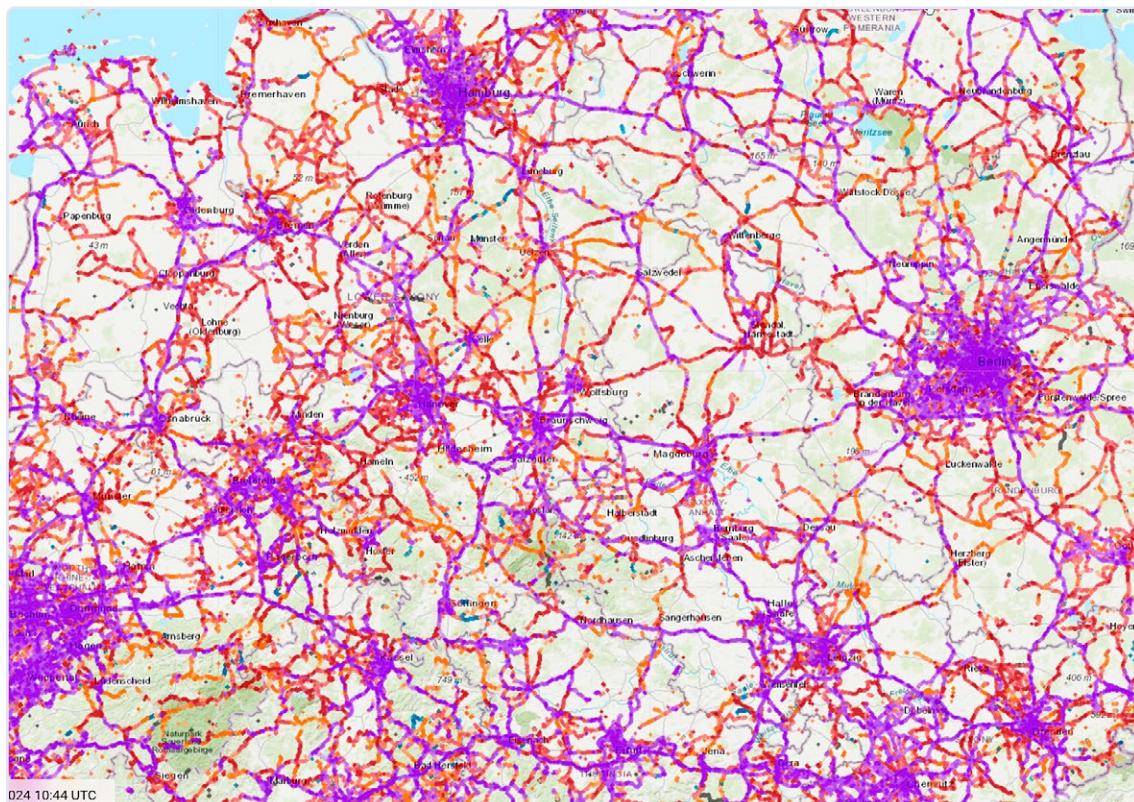


Figure 9. Déploiement des réseaux 4G et 5G dans la région Hanovre-Berlin-Hambourg
(Source : NPERF, nperf.com/fr/map/DE/)

redeviennent rentables. La demande de nouveaux services 5G n'a pas progressé au rythme prévu jusqu'à présent, mais elle progresse et exercera bientôt la dynamique nécessaire pour que la 5G puisse reprendre sa course.

Pendant ce temps, en décembre 2023, le 3GPP a annoncé qu'il commencerait à travailler sur les spécifications de ce qui sera la 6^e génération, la 6G, avec l'objectif de lancer les premiers essais avant 2030. Ainsi, bien qu'il soit trop tôt pour prédire la composition et l'architecture exactes de la 6G, nous pouvons néanmoins prévoir que nous contribuerons au développement d'une sorte de 5.5G. Elle pourrait être disponible peu avant 2030 puisque plusieurs nouvelles fonctionnalités, qui seront intégrées dans les futurs réseaux 6G, font déjà l'objet d'un financement et d'un développement publics dans le cadre de la course à la suprématie technologique. En anticipant et en intégrant certaines des fonctionnalités conçues pour faire partie de la 6G, ces standards seraient adoptés comme norme de facto.

Par exemple, plusieurs annonces associées aux réseaux 5G, comme la prise en charge de systèmes de conduite entièrement automatisés, ne connaîtraient probablement une véritable impulsion massive que grâce à l'amélioration de la vitesse, de la couverture et de l'intégration de différents éléments de réseau qui feront théoriquement partie des réseaux 6G.

À ce jour, les principaux objectifs de la 6G sont un nouveau bond en avant jusqu'à 1 Tbit/s du débit de données, une nouvelle réduction de la latence vers 100 microsecondes, une couverture totale du territoire intégrant des réseaux cellulaires et non cellulaires, y compris des constellations de satellites de télécommunications, de nouveaux types de hotspots et de

nouveaux appareils. Avec le soutien de systèmes d'IA pour optimiser toutes les ressources du réseau, un véritable service personnalisé est offert, et c'est une autre exigence importante, également la réduction de la pollution électromagnétique et la consommation d'énergie. Si la 5G devrait contribuer à l'explosion de l'IdO, la 6G intégrera l'intelligence artificielle dans le réseau, ce qui permettra une interaction plus étroite entre la réalité et son extension dans les métavers.

VF : Laurent Rauber — 240233-04



À propos de Nemo Galletti

Nemo Galletti est le chef de site de RFS Italia, la branche italienne de Radio Frequency Systems GmbH, une société multinationale contrôlée par Nokia, au cours de cet entretien. Il a étudié l'ingénierie électronique à l'École polytechnique de Milan et a acquis 39 ans d'expérience dans le domaine des télécommunications, d'abord dans la R&D pour le groupe Alcatel, puis dans le secteur commercial à l'étranger en vendant des réseaux de télécommunications, avant de rejoindre RFS.



À propos de Roberto Armani

Roberto Armani est ingénieur en électronique. Après ses études à l'École polytechnique de Milan, il a acquis plus de 35 ans d'expérience dans divers secteurs. Avant de rejoindre l'équipe d'*Elektor* en tant que rédacteur en chef, il a travaillé dans l'industrie informatique, l'imagerie électronique, les télécommunications, les équipements d'essai des matériaux et la publication sur le web. Outre l'électronique, il aime écouter (et chanter) de la musique classique et faire des promenades en haute montagne.

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.

Andonstar AD210 10,1" Microscope numérique



L'Andonstar AD210 est un microscope numérique doté d'un grand écran IPS de 10,1 pouces qui offre un angle de vision de 178° et prend en charge la vidéo 1080P et la capture photo de 12 MP. Avec un grossissement de 260x, le microscope offre une vue claire des côtés des composants des circuits imprimés. Le microscope est également livré avec une carte mémoire de 32 Go et une télécommande.

Prix : ~~179,95 €~~

Prix spécial : 129,95 €

www.elektor.fr/20802

Kit de développement Waveshare Jetson Nano Lite



Le kit de développement Waveshare Jetson Nano, basé sur les ordinateurs IA Jetson Nano (avec 16 Go eMMC) et Jetson Xavier NX, fournit presque les mêmes E/S, taille et épaisseur que le kit de développement Jetson Nano (B01), plus pratique pour mettre à niveau le noyau module.

Prix : 269,00 €

Prix (membres) : 242,10 €

www.elektor.fr/20761



UNI-T UT512D Testeur de résistance d'isolation (2,5 kV)

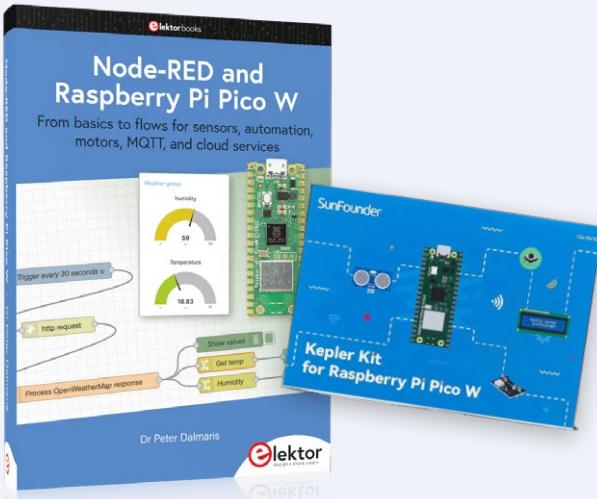


Prix : 289,00 €

Prix (membres) : 260,10 €

www.elektor.fr/20786

Kit de développement Node-Red



Prix : 104,95 €

Prix spécial : 84,95 €

www.elektor.fr/20849

CrowVision 11,6" écran tactile capacitif IPS (1366 x 768)

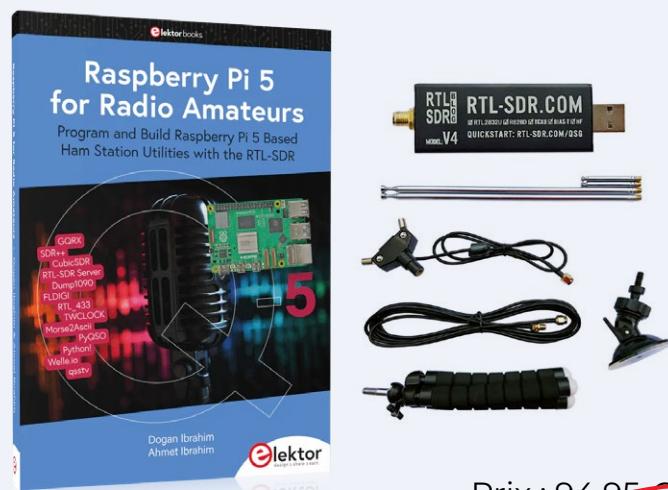


Prix : 139,95 €

Prix (membres) : 125,96 €

www.elektor.fr/20792

Raspberry Pi 5 RTL-SDR V4 (offre groupée)



Prix : 104,95 €

Prix spécial : 84,95 €

Prix : 94,95 €

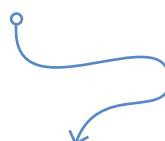
Prix spécial : 79,95 €

www.elektor.fr/20878



démarrer en électronique

...connexions symétriques



Eric Bogers (Elektor)

Dans cet épisode, nous aborderons quelques montages à amplificateurs opérationnels (ampli-op) importants, notamment l'amplificateur différentiel et l'amplificateur de mesure. Nous aborderons également un thème important, en particulier dans les applications audio : les connexions équilibrées.

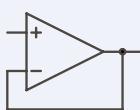


Figure 1. Suiveur de tension.

Suiveur de tension

Une application courante de l'amplificateur non inverseur est le suiveur de tension (**figure 1**). En affectant une valeur « zéro » à R₁ et une valeur « infinie » à R₂ dans la formule du gain, le résultat est exactement 1. Tout comme l'émetteur suiveur abordé précédemment, le suiveur de tension a une résistance d'entrée élevée et une faible résistance de sortie - surpassant l'émetteur suiveur de plusieurs ordres de grandeur. Le suiveur de tension est également connu sous le nom de « transformateur d'impédance ».

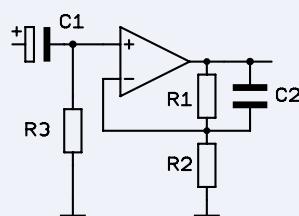


Figure 2. Limitation de la bande passante

Limitation de largeur de bande

Dans l'amplificateur non inverseur, la bande passante doit être limitée, et nous utilisons ici aussi des condensateurs à cette fin (**figure 2**).

Contrairement à l'amplificateur inverseur, il est possible de régler la résistance d'entrée indépendamment du gain, ce qui permet d'utiliser des condensateurs de capacités plus faibles. Par ailleurs, les condensateurs

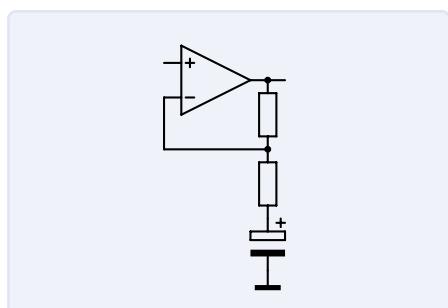


Figure 3. Réduction du gain en continu.

sont dimensionnés de la même manière que dans l'amplificateur inverseur. Une méthode répandue consiste à ajouter un condensateur en série avec R₂ ; dans ce cas, pour les tensions continues, le gain est alors de 1. Le gain en courant alternatif (au-delà de la fréquence de coupure) est défini par R₁ et R₂ (**figure 3**).

Amplificateur sommateur

Une version de l'amplificateur inverseur est l'amplificateur sommateur (**figure 4**), qui additionne les tensions de plusieurs sources.

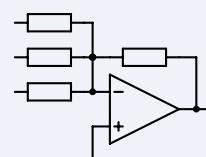


Figure 4. Amplificateur sommateur.

En général, nous utilisons résistances d'entrée de valeurs égales, mais le circuit fonctionne également avec des résistances d'entrée différentes – les tensions individuelles à additionner doivent alors être multipliées par le facteur adéquat.

Les montages sommateurs sont couramment utilisés dans les tables de mixage pour combiner les différents canaux - non seulement pour le master ou le sous-groupe, mais aussi dans les pistes de signaux auxiliaires. Comme le nœud (point de sommation) est une masse virtuelle, les canaux individuels peuvent être mixés librement sans risque d'interférence. La faible impédance du point de sommation rend également le signal de sommation relativement insensible aux interférences.

Bien entendu, cette connexion « rail » n'est pas exactement au potentiel zéro et son impédance n'est pas nulle, ce qui limite son insensibilité aux interférences. Pour minimiser davantage les interférences, les lignes de signal individuelles sont idéalement placées entre deux connexions de masse. Il est également possible de rendre le signal symétrique, ce qui nécessite naturellement l'utilisation de deux connexions de bus.

Connexions symétriques

Avant d'aborder l'amplificateur différentiel, nous devons d'abord comprendre la différence entre les connexions symétriques et asymétriques.

Comme nous l'avons évoqué au début de cette série d'articles, nous avons besoin d'un chemin de sortie et d'un chemin de retour pour permettre à un courant électrique de circuler. En courant continu, on parle d'une ligne positive et d'une ligne négative, et en courant alternatif, on parle d'une phase et d'une masse (ou zéro).

Pour connecter des haut-parleurs, on utilise souvent un câble bifilaire (non blindé) à noyau torsadé, avec l'un des fils servant de phase et l'autre de masse. En pratique, le choix du fil n'affecte pas la connexion ; la connexion du câble au connecteur est standardisée et l'un des fils est marqué.

Les câbles de haut-parleurs sont peu sensibles aux signaux parasites : ils transmettent généralement un niveau de

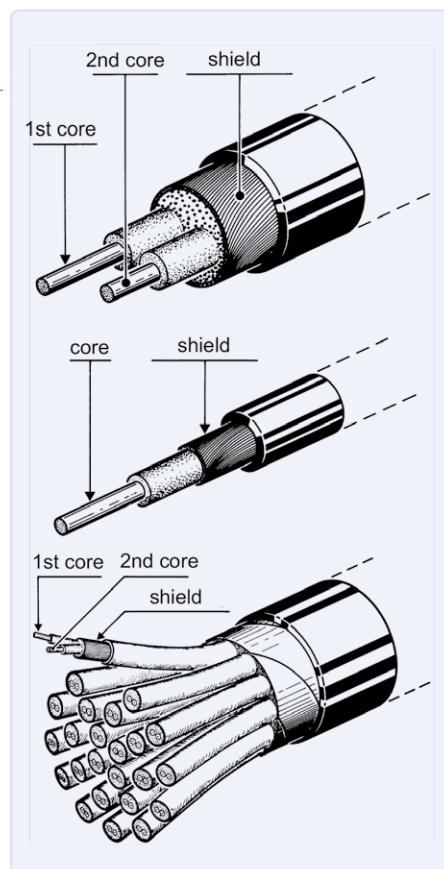


Figure 5. Lignes symétriques et asymétriques et un câble multiconducteur (en bas).

signal élevé et possèdent une très faible impédance.

Cependant, les câbles transportant des signaux de niveau nettement inférieur se terminent par une impédance beaucoup plus élevée, ce qui nécessite l'utilisation d'un câble blindé (**figure 5**). Le blindage peut être constitué d'une bande conductrice ou d'une gaine tressée et agit comme une cage de Faraday - les signaux d'interférence ne peuvent alors (du moins en théorie) pas

atteindre les fils conducteurs.

Dans sa forme la plus simple, un tel câble blindé est composé d'un conducteur enveloppé dans un blindage ; le conducteur intérieur transporte le signal tandis que le blindage est relié à la masse. Pour des câbles courts transportant un signal de niveau élevé, une telle connexion asymétrique est tout à fait suffisante et est souvent utilisée. Tous les câbles équipés de connecteurs RCA ou Cinch ou jack mono utilisent une connexion asymétrique. Les connecteurs jack stéréo peuvent transporter un signal mono symétrique mais peuvent aussi être utilisés différemment.

La **figure 5** montre un exemple typique de câble asymétrique au centre.

Pour les câbles longs et/ou transportant un signal de faible niveau, un câble symétrique est toujours utilisé. Ici, le signal est transmis par un conducteur en phase et par un autre conducteur en phase opposée, les deux étant enveloppés dans un blindage commun. La partie supérieure de la figure 5 présente un exemple d'un tel câble. Quel est alors l'avantage réel de cette transmission « double » de signaux ?

La **figure 6a** montre un exemple de connexion asymétrique entre deux appareils. Si des interférences se produisent, elles sont effectivement dirigées vers la masse par le blindage, bien qu'une petite

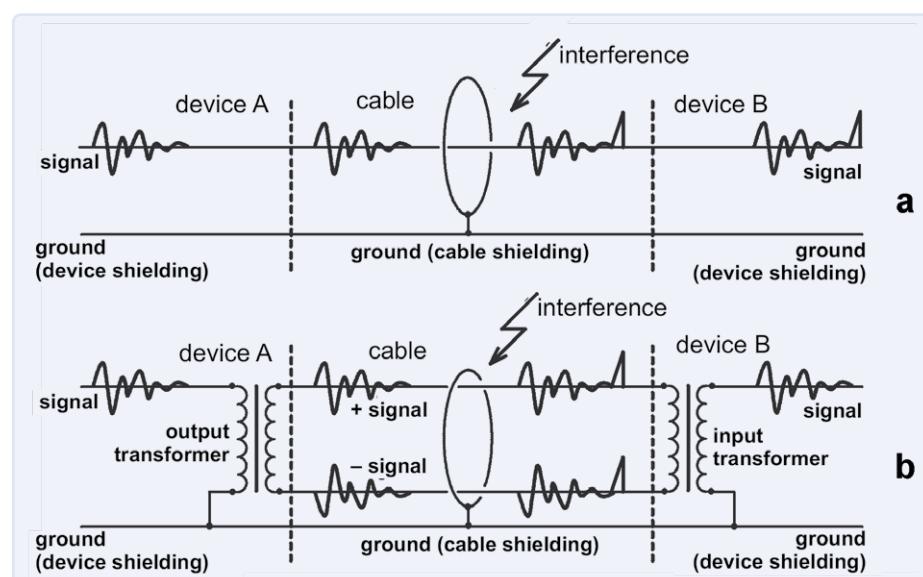


Figure 6. Transmission de signaux asymétrique (a) et symétrique (b)



partie puisse atteindre le conducteur du signal. Pour la suite de l'article, il n'est pas important de déterminer si cela résulte d'un blindage imparfait ou de la pénétration des interférences à travers la résistance ohmique du blindage. Nous supposerons ici que le blindage n'est pas parfaitement efficace, pour simplifier la démonstration. Dans une connexion équilibrée (**figure 6b**), un conducteur transporte le signal dans la bonne phase, tandis que l'autre transporte le même signal en phase opposée. Les interférences affectent les deux conducteurs, mais ce signal d'interférence (contraire au signal « utile ») a la même phase sur les deux conducteurs. Dans l'appareil récepteur, les signaux sur les deux conducteurs sont soustraits l'un de l'autre (formant ainsi un signal différentiel) annulant ainsi le signal d'interférence.

Vous vous demandez peut-être sur la nécessité du blindage si dans une connexion symétrique les signaux d'interférence sont complètement supprimés. Premièrement, les deux conducteurs porteurs de signaux ne captent jamais le signal d'interférence de la même manière, même lorsque les conducteurs sont torsadés. Deuxièmement, la formation du signal différentiel n'est jamais parfaite.

Les étages d'entrée différentiels utilisant de résistances de 1% atteignent une réjection en mode commun d'environ 40 dB ; les transformateurs de signaux professionnels (selon la fréquence) atteignent environ 60 dB. Dans la pratique, cela n'est souvent pas suffisant. Cependant, lorsque nous combinons une transmission de signal équilibrée avec un blindage adéquat, nous obtenons un excellent rapport signal/bruit.

Amplificateur différentiel

L'amplificateur différentiel donne la différence de deux signaux d'entrée (**figure 7**). Le gain se détermine de la même manière que pour l'amplificateur inverseur ; la tension de sortie est donc :

$$U_{\text{out}} = (U_{\text{in+}} - U_{\text{in-}}) \cdot \frac{R_2}{R_1}$$

L'un des problèmes de l'amplificateur différentiel est l'impédance d'entrée extrêmement déséquilibrée. Ce problème peut être

contourné non pas en utilisant des résistances de valeurs égales mais en maintenant leur rapport égal. Dans la pratique, cela a un effet préjudiciable sur la suppression en mode commun (suppression ou atténuation des signaux en phase) : si nous utilisons des valeurs de résistance égales, dans de nombreux cas, elles proviendront de la même série de production - ce qui signifie que, bien qu'elles diffèrent légèrement de la valeur indiquée sur l'étiquette, elles présentent donc des caractéristiques similaires.

En utilisant des valeurs de résistance différentes, les deux rapports R_2/R_1 ne seront jamais parfaitement égaux, ce qui entraîne une amplification simultanée d'un signal décalé.

Avec des résistances à film métallique de 1%, la suppression maximale du mode commun atteint généralement 40 dB.

Amplificateur d'instrumentation

Plus le gain délivré par un amplificateur différentiel est élevé, plus les problèmes d'impédances d'entrée inégales deviennent significatifs. Pour des amplifications de 20 dB ou plus - c'est-à-dire 10 fois ou plus - l'utilisation des amplificateurs d'instrumentation est recommandée (**figure 8**). Le premier avantage de cette configuration est que les deux entrées ont une impédance d'entrée égale, qui est généralement réglée à une valeur utilisable (inférieure) via une résistance à la masse.

Veuillez noter la connexion particulière des résistances à l'amplificateur d'entrée : lors de l'application de signaux symétriques, un point de masse virtuel se forme au milieu de R_4 , de sorte que la moitié de la valeur de R_4 doit être introduite dans la formule du gain. En revanche, pour les signaux de mode commun, on peut considérer que R_4 est absent, ce qui fait que le gain des composantes du signal de mode commun est égal à un.

L'augmentation de l'atténuation asymétrique est de l'ordre de grandeur du gain de l'étage d'entrée. Comme notre principale préoccupation est de maximiser l'atténuation asymétrique, dans la plupart des cas, nous attribuerons à R_1 et R_2 la même valeur - disons 10 kΩ - et laisserons l'étage d'entrée fournir le gain. Si nous le réglons à

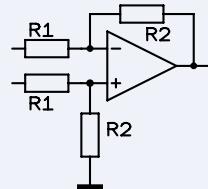


Figure 7. Amplificateur différentiel.

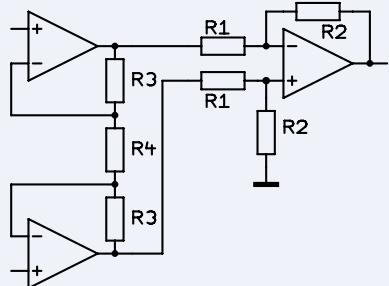


Figure 8. Amplificateur d'instrumentation.

20 dB, une atténuation de mode commun de 60 dB est réalisable pour le circuit dans son ensemble.

Nous en resterons là pour l'instant ; le prochain article couvrira de nombreux montages à ampli-op pratiques et intéressants. ↗

240237-04

Note de la rédaction : 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

➤ Livre «Basic Electronics for Beginners», B. Kainka (Elektor, 2020)

Version papier :

www.elektor.fr/19212

Version numérique :

www.elektor.fr/19213

Ils nous font confiance, n'est-ce pas ?



Nouveaux produits

- Multimètre intelligent FNIRIS DM1-09
- Raspberry Pi 5 (8 Go de RAM)
- Kit Innocent Disport Connect
- PNIRIS 103BD 2-en-1 Tablette Oscilloscopique (10G MSa/s)

Meilleures ventes

- UT603 LCD-mètre
- Boîtier d'Assemblage 1978-2023 d'Elstart sur élstart
- Elstart Labo
- DYSON Multimètre OWON XDS1114I
- PNA1014B Oscilloscope 100 MHz à 2 canaux & fonction de mesure

“Produit livré rapidement, bien emballé et conforme à la commande”

★★★★★ by Albert Jouret

Rated 4.6 / 5 | 398 reviews

★ Trustpilot

NOUVELLE
E-CHOPPE

“Excellent Transaction – RAS ! Sur une dernière commande, d'un montant de 50 euros, tout s'est bien déroulé et plus rapidement que prévu. L'objet est en excellent état et conforme à l'annonce.”

★★★★★ by Aida Alda

Rated 4.6 / 5 | 400 reviews

★ Trustpilot

Nous aimons l'électronique et les projets, et nous faisons tout notre possible pour répondre aux besoins de nos clients.

Le magasin Elektor : **Jamais cher, toujours surprenant**

“Très satisfait – Envoi très rapide, bien emballé. Je recommande vivement !”

★★★★★ by FORTIN GUILLAUME

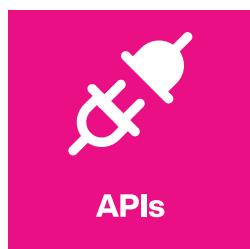
Rated 4.6 / 5 | 544 reviews

★ Trustpilot

Consultez d'autres avis sur notre page Trustpilot : www.elektor.com/TP/fr

Vous pouvez également vous faire votre propre opinion en visitant notre Elektor Store, www.elektor.fr





Passez commande en toute simplicité

Passez commande en toute simplicité

mouser.fr/servicesandtools

