

nœud de capteurs LoRa

transmission de données
longue portée et
alimentation solaire

FOCUS SUR

Sans-fil et communication



extension de couverture WiFi sur ESP32

comment modifier simplement une antenne



Sonde RF avec bargraphe à LED

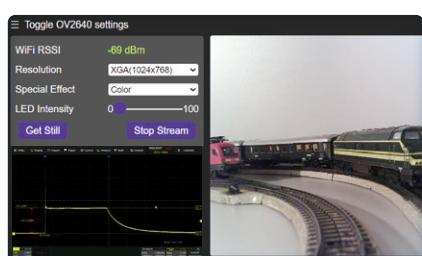


antenne magnétique à large bande pour les grandes ondes

tous les émetteurs sans aucun accord



filtre coupe-bande audio à fréquence réglable



train miniature avec caméra

installation d'un module ESP32 CAM



carte eXpansion d'Elektor pour ESP32-S3 et autres modules XIAO



NOTRE GAMME PAR DES TECHNICIENS POUR LES TECHNICIENS



Tirer le meilleur parti de votre projet: www.reichelt.com

Uniquement le meilleur pour vous – provenant de plus de 1500 marques.

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



OFFICIAL
RESELLER 2023



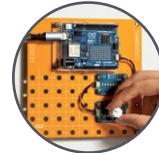
ARDUINO – La plateforme open source polyvalente de prototypage électronique et de projets interactifs

Kit Arduino Plug and Make

avec carte Arduino Uno R4 WiFi, capteurs et actionneurs Modulino®, câble USB-C et bien plus encore.

Libérez votre créativité avec le kit Arduino Plug and Make. Il constitue votre point de départ idéal et vous offre tout ce dont vous avez besoin pour explorer l'écosystème Arduino.

- Aucune carte d'expérimentation et aucune soudure requise



NOUVEAU

N° de commande :
ARD AKX00069

95,70
(79,75)

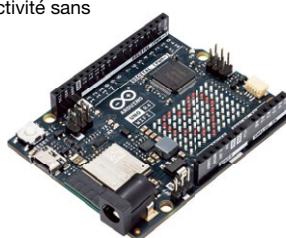
Carte Arduino UNO R4 WiFi

La carte Arduino Uno R4 WiFi combine la puissance de calcul et les nouveaux périphériques passionnantes du microcontrôleur RA4M1 de Renesas avec la puissance de connectivité sans fil de l'ESP32-S3 d'Espressif.

- Renesas RA4M1 (Cortex M4)
- Wi-Fi/BT LE (ESP32-S3)

BESTSELLER

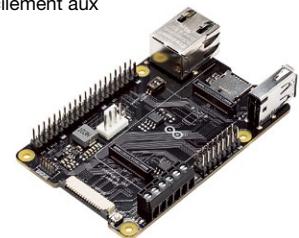
N° de commande :
ARD UNO R4 WIFI 25,01 (20,84)



Carte porteuse Arduino Portenta HAT

Créez des prototypes comme un professionnel, automatisez votre projet ou approfondissez vos connaissances en robotique. La carte porteuse Portenta Hat vous permet d'accéder facilement aux hautes performances !

- Réalisez vos projets Portenta avec les Hats Raspberry Pi®
- Embase à 16 broches, Gigabit Ethernet, microSD, USB-A



N° de commande :
ARD PORTENTA HAT 45,88 (38,24)

 **reichelt**
elektronik MAGAZINE

Arduino dans le magazine Reichelt :
projets Arduino, conseils et tutoriels. À vous de jouer !



A lire ►
<https://rch.lt/m-arduino>



Arduino - Microcontrôleurs puissants pour fonctions de commutation et de commande

Idéal pour ceux qui veulent programmer des objets et environnements interactifs.

Découvrir maintenant ►
www.reichelt.com/arduino



Types de paiement :      

PRIX DU JOUR! Prix à la date du: 20. 8. 2024

■ Excellent rapport qualité prix

■ Plus de 150 000 produits sélectionnés

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

www.reichelt.com

Assistance téléphonique: +33 9 75 18 03 04

Les réglementations légales en matière de résiliation sont applicables. Tous les prix sont indiqués en € TVA légale incluse, frais d'envoi pour l'ensemble du panier en sus. Seules nos CGV sont applicables (sur le site <https://rch.lt/CG-FR> ou sur demande). Semblables aux illustrations. Sous réserve de coquilles, d'erreurs et de modifications de prix.
reichelt elektronik GmbH, Elektronikring 1, 26452 Sande (Allemagne), tél. +33 9 75 18 03 04

 **reichelt**
elektronik - Tirer le meilleur parti de votre projet

47^{ème} année n° 509
septembre-octobre 2024
ISSN 0181-7450

N° de TVA Intracommunautaire : FR90319937454

Dépôt légal : septembre 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



L'ère du sans-fil

Il est indéniable que le 20^{ème} siècle a été marqué par le règne du câble électrique, alors que notre siècle est caractérisé par l'avènement de la technologie sans fil. Les bénéfices de la communication sans fil sont considérables, facilitant une mobilité accrue des appareils et rendant possibles des innovations telles que l'internet des objets.

Toutefois, la communication sans fil présente une complexité inhérente qui entraîne souvent des retards significatifs, dus à la nécessité de garantir une transmission à la fois sécurisée et fiable. La synchronisation des flux de données et des appareils représente un défi particulier avec des technologies numériques telles que le Bluetooth et le Wifi, alors que cela serait bien plus simple avec une connexion filaire. Cette complexité rend difficile l'obtention de temps de réponse courts ou la synchronisation précise entre différents canaux. Par exemple, la communauté des technophiles attend avec impatience le lancement de Bluetooth LE Audio, qui promet une synchronisation impeccable des canaux gauche et droit pour les signaux stéréo. Avec l'un de nos contributeurs réguliers, nous avions envisagé de rédiger un article détaillé sur ce sujet. Cependant, en raison du coût élevé des premières cartes de développement et de la complexité du logiciel, nous avons préféré reporter la publication de cet article à une édition ultérieure. Dans ce numéro, nous explorons d'autres techniques sans fil innovantes que vous pouvez mettre en œuvre immédiatement dans vos propres projets. Nous vous présentons notamment la technologie LoRa, une solution de transmission de données à faible consommation énergétique. Vous découvrirez deux projets utilisant des nœuds capteurs autonomes qui collectent des mesures et les transmettent via LoRa. Les deux articles se complètent parfaitement. L'auteur Claus Kühnel se concentre sur l'alimentation électrique et ses mesures (page 94), tandis que mon collègue Saad détaille le transfert de données vers le cloud pour une visualisation et une récupération aisées, dans son article en page 6. L'aspect le plus remarquable de ces projets est leur utilisation efficace de modules prêts à l'emploi, permettant de réduire considérablement le temps de développement. Comme de coutume, en plus de notre édition imprimée, nous proposons une édition numérique bonus qui inclut des projets supplémentaires. Vous pouvez la consulter sur notre page dédiée au « sans-fil » (www.elektormagazine.fr/sans-fil-communication). Parmi les projets présentés, l'auteur Sebastian Westerhold explique comment modifier une tête TV pour l'utiliser en radio amateur. Ne manquez pas de découvrir ce projet passionnant !

En savoir plus sur le sans-fil

Si vous êtes particulièrement intéressé par les technologies et les applications sans fil, nous vous invitons à consulter notre page dédiée aux technologies sans fil. Vous y trouverez des mises à jour continues et des articles exclusifs tout au long de l'année.

www.elektormagazine.fr/sans-fil-communication



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 Joosten, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | Maquette : Harmen Heida, Sylvia Sopamena, Patrick Wielders | Des questions techniques : redaction@elektor.fr



un nœud de capteurs autonome

transmission de données via
LoRa

6

Articles de fonds

- 3 **Édito**
- 48 **démarrer en électronique...**
... plus de montages à ampli-op
- 58 **sur le vif**
à propos de prendre les choses pour acquises
- 109 **projet 2.0**
corrections, mises à jour et courrier des lecteurs
- 112 **2024 : l'odyssée de l'IA**
accélérateurs d'IA : une comparaison

Features

- 32 **TensorFlow Lite pour microcontrôleurs**
par un débutant, pour les débutants
- 44 **livres d'Elektor**
sonde RF
- 86 **le système LeoINAGPS**
surveillez votre véhicule électrique
- 100 **AWS pour Arduino et cie. (2)**
transmission de données avec AWS IoT ExpressLink

CONTENU BONUS

Découvrez le numéro bonus du magazine
Elektor consacré au « sans-fil et
communication ».

- Projet : accès à la bande X avec une tête TV modifiée
- Revue: le récepteur ATS25
max-Decoder
- Projet : convertisseur ArtNet-DMX
basé sur ESP32
- Infographie : sans-fil et communication



[www.elektormagazine.fr/
sans-fil-communication](http://www.elektormagazine.fr/sans-fil-communication)

Projets

- 6 **un nœud de capteurs autonome**
transmission de données via LoRa et alimentation par cellules solaires
- 18 **la carte eXpansion V1.0 d'Elektor**
pour ESP32S3 et autres cartes XIAO
- 23 **une caméra dans un train miniature**
installation d'un module ESP32 CAM
- 28 **antenne magnétique à large bande pour les grandes ondes**
tous les émetteurs sans aucun accord
- 38 **un concentrateur pour les appareils RS-422 et RS-485**
câbler votre bus comme une star
- 52 **Open Vario**
variomètre multifonction open source pour vol en parapente
- 60 **relevé des compteurs d'eau basé sur l'IA (2)**
intégrer votre ancien compteur dans l'IoT !



comment modifier simplement une antenne

118



pour ESP32S3 et autres cartes XIAO

118

- 81 **filtre coupe-bande audio à fréquence réglable**
solution universelle de suppression de fréquences dans le domaine audio
- 94 **nœud LoRa alimenté par énergie solaire**
une solution IdO modulaire, compacte et polyvalente
- 118 **extension de couverture Wi-Fi sur ESP32**
comment modifier simplement une antenne

Plongez dans le sans-fil et la communication

Visitez la page « sans-fil et communication » pour découvrir des projets, des vidéos et des tutoriels !
[www.elektormagazine.fr/
sans-fil-communication](http://www.elektormagazine.fr/sans-fil-communication)



Industrie

- 68 **agriculture intelligente**
détection des nuisibles basée sur l'apprentissage machine avec connectivité IdO
- 72 **Anybus CompactCom est le choix idéal pour la communication industrielle embarquée - voici pourquoi**
- 75 **norme de communication IQRF**
fiabilité des réseaux maillés sans fil à faible débit avec perte
- 78 **comment construire un robot agricole intelligent**
considérations essentielles et défis techniques

Bientôt dans ces pages

Le numéro de novembre-décembre 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 « production et composants ».

- Conception de circuits imprimés
- Carte processeur Elektor Audio DSP FX
- Testeur de quartz autonome
- MultiCalculateur pour l'électronique
- KiCad 8 : nouvelles et dernières caractéristiques
- Projets ESP32 avec PlatformIO
- Testeur d'amplificateurs pour applications audio et autres
- Circuit imprimé de sapin de Noël en 3D
- Afficheur à LED avec le MAX7219

Le numéro de novembre-décembre 2024 du magazine Elektor sera publié aux alentours du 13 novembre 2024. La date d'arrivée du magazine papier chez les abonnés dépend des aléas d'acheminement.



un nœud de capteurs autonome

transmission de données via LoRa et alimentation par cellules solaires

Saad Imtiaz (Elektor)

Ce projet de nœud de capteurs, exploite la technologie LoRa pour la transmission de données à faible consommation énergétique et à longue portée, le rendant idéal pour la surveillance environnementale à distance. Utilisant le module LoRa WIO E5 et une carte contrôleur ESP32-C3 XIAO, il intègre un capteur de CO₂ SCD30 et un capteur d'humidité du sol, tous alimentés par énergie solaire. Les données collectées sont transmises au réseau The Things Network (TTN) et peuvent être consultées via Datacake ou Home Assistant, offrant des informations en temps réel et des possibilités d'automatisation. Ce guide détaillé vous accompagnera à travers toutes les étapes, depuis la mise en place de votre passerelle LoRaWAN jusqu'à son intégration fluide dans TTN, Datacake, et Home Assistant.

Source: Adobe Stock



Figure 1. Nœud de capteurs LoRaWAN alimenté par l'énergie solaire.

distance, en utilisant la communication LoRa à longue portée et l'énergie solaire pour assurer un fonctionnement continu. En intégrant des capteurs personnalisables, le système offre une grande flexibilité pour répondre aux différents besoins de surveillance environnementale. L'une des caractéristiques de ce projet est son intégration dans la plateforme d'automatisation open-source Home Assistant, qui offre des capacités d'automatisation avancées. Par exemple, en mesurant l'humidité du sol, il est possible d'automatiser l'irrigation du jardin, ce qui peut soulager les propriétaires du fardeau de l'arrosage manuel. Dans cet article, nous allons explorer le processus d'installation d'un nœud de capteurs LoRaWAN, illustré à la **figure 1**, depuis la configuration d'une passerelle LoRaWAN et l'intégration au réseau The Things Network (TTN) jusqu'à la visualisation des données sur des plateformes telles que Datacake et Home Assistant. Ce projet montre comment une technologie abordable et adaptable peut offrir des solutions de surveillance complètes. Il pallie les insuffisances des options actuellement disponibles sur le marché et ouvre de nouvelles perspectives pour des applications plus étendues.

Aperçu du système

Le cœur de ce projet est la carte microcontrôleur XIAO ESP32-C3 de Seeed Studio. Vous pouvez acquérir cette carte ainsi que les autres modules nécessaires pour ce projet chez Elektor (voir l'encladré Produits). Le choix de la carte XIAO a été fait en raison de sa taille compacte et de ses capacités d'E/S adéquates pour ce projet. Pour étendre ses options d'E/S, nous avons utilisé la carte eXpansion d'Elektor [1] qui augmente les fonctionnalités du microcontrôleur avec

La surveillance à distance de l'environnement est essentielle pour relever les défis du changement climatique et de la gestion des ressources. Cependant, de nombreux nœuds de capteurs LoRa disponibles sur le marché sont coûteux et manquent d'options de personnalisation, ce qui les rend peu pratiques pour des applications spécifiques et des zones éloignées. Pour surmonter ces limitations, ce projet présente un nœud de capteur LoRa polyvalent et rentable. Ce système personnalisé offre une solution efficace pour la collecte de données à

LoRa, LoRaWAN, une passerelle et TTN ?

LoRa : LoRa (Long Range) est une technologie de communication sans fil spécialement conçue pour permettre la transmission de données à longue distance tout en minimisant la consommation énergétique. Opérant sur des fréquences sub-gigahertz (868 MHz en Europe et 915 MHz en Amérique du Nord), cette technologie est capable de transmettre des données jusqu'à 15 km dans des zones rurales et jusqu'à 5 km en milieu urbain. Elle est particulièrement adaptée pour des applications IoT telles que la surveillance environnementale ou l'agriculture intelligente.

LoRaWAN : LoRaWAN (Long Range Wide Area Network) est un protocole de communication qui exploite la technologie LoRa pour établir des connexions entre les appareils LoRa et les passerelles. Ce protocole est idéal pour les déploiements IoT à grande échelle, car il assure une communication sécurisée et bidirectionnelle, supporte des débits de données adaptatifs et

optimise l'utilisation du réseau.

Passerelle : Une passerelle LoRaWAN sert de pont entre les appareils LoRa et l'internet, recevant les données des appareils et les relayant à un serveur central via des connexions Wi-Fi, Ethernet ou cellulaires. Ces passerelles peuvent couvrir de vastes zones et gérer de multiples transmissions simultanées d'appareils.

TTN (The Things Network) : The Things Network est une initiative mondiale qui propose un réseau LoRaWAN open-source. Il fournit une infrastructure complète pour connecter des appareils LoRaWAN, incluant l'enregistrement des appareils, le routage des données et l'intégration avec des plateformes telles que Datacake et Home Assistant. Grâce à son large soutien communautaire, TTN constitue une ressource précieuse pour le déploiement d'applications IoT via LoRaWAN.

six connexions I²C. Bien que ce projet n'utilise qu'une seule de ces connexions I²C, la flexibilité de ce nœud de capteurs permet l'intégration d'autres capteurs au besoin. La **figure 2** présente le schéma fonctionnel du projet.

Pour la communication LoRa, nous avons choisi le module E5 WIO LoRa de Seeed Studio, contrôlable via une connexion UART et des commandes AT pour l'envoi de données et la configuration. Sa facilité de configuration et son coût raisonnable en ont fait une option appropriée. Ce module assure une transmission fiable des données à longue distance. Au cours de mes essais, j'ai pu obtenir une portée d'environ 700 m dans un environnement urbain malgré la présence d'arbres et de bâtiments. Il est important de noter que ma passerelle était située à l'intérieur, près d'une fenêtre, ce qui est considéré comme une bonne performance. Comme pour tout système de communication sans fil, la ligne de vue est cruciale. La portée pourrait être considérablement augmentée en plaçant la passerelle en hauteur avec une antenne appropriée. L'efficacité du module pourrait également être améliorée en utilisant une meilleure antenne.

Le système de recharge solaire utilise un panneau solaire 3 W de Seeed Studio, connecté à un module de gestion de l'énergie solaire de Waveshare qui est un tracker MPPT avec quelques circuits de protection. Cette configuration permet de charger deux batteries 18650 en parallèle. Le tracker MPPT (*Maximum Power Point Tracking*) est essentiel pour optimiser la puissance produite par le panneau solaire.

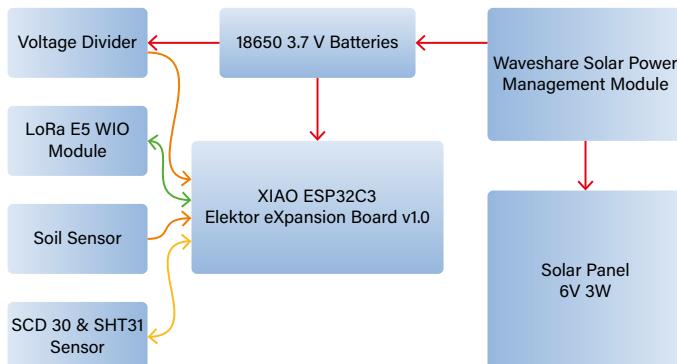
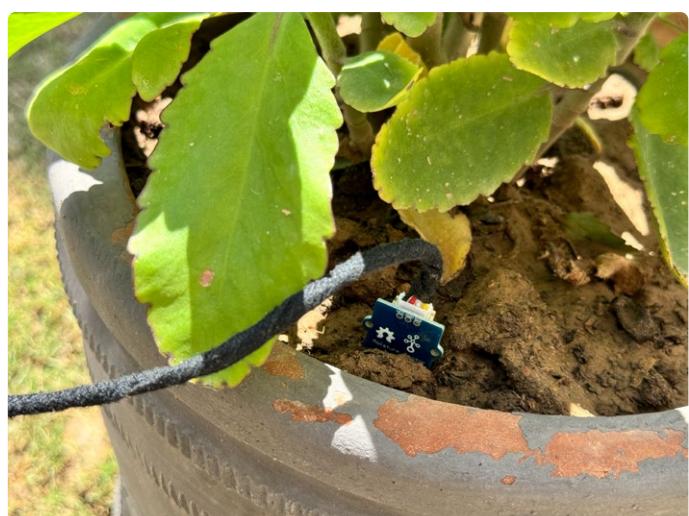


Figure 2. Schéma fonctionnel du projet.

Il ajuste continuellement le point de fonctionnement électrique du panneau solaire, pour maximiser son efficacité. Après avoir testé plusieurs modules de suivi MPPT, le module Waveshare s'est avéré être le plus performant et le plus fiable pour cette application.

Nous avons choisi le capteur SCD30 de Seeed Studio sa grande précision dans la détection du CO₂. Le SCD30 est un capteur NDIR, ce qui en fait un véritable capteur de CO₂, capable de mesurer la concentration en CO₂ en ppm dans l'air ambiant. Un autre aspect intéressant de ce capteur est qu'il est livré avec un capteur de température et d'humidité SHT31 déjà intégré. Le SCD30 n'étant pas conçu pour être utilisé à l'extérieur, un boîtier Stevenson imprimé en 3D a été créé pour le protéger des éléments extérieurs.

Le capteur d'humidité du sol utilisé dans ce projet est également un produit de Seeed Studio. Bien qu'il ait été choisi en raison de sa disponibilité, il n'est pas idéal pour les applications extérieures car ses connecteurs et son circuit imprimé ne sont pas étanches, comme le montre la **figure 3**. Dans les prochaines itérations du projet, il sera envisageable de choisir un capteur d'humidité du sol plus adapté aux conditions extérieures.



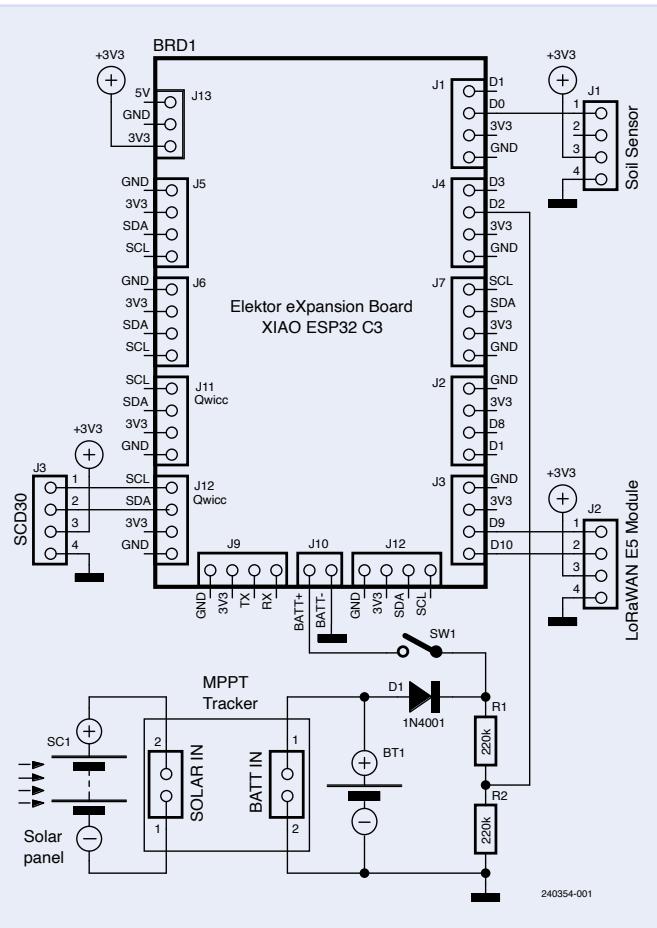


Figure 4. Schéma du projet.

Le microcontrôleur XIAO ESP32-C3 de Seeed Studio communique avec le capteur SCD30 via I²C et avec le module LoRa E5 via UART. Le capteur d'humidité du sol est connecté à une broche analogique et la tension de la batterie est régulée avec un circuit diviseur de tension. Le schéma de circuit du projet est représenté dans la **figure 4**.

Configuration de la passerelle LoRaWAN

Avant d'aborder les aspects techniques plus complexes du projet, commençons par l'élément principal : la configuration de la passerelle LoRaWAN. Comme nous utilisons LoRaWAN pour transmettre les données des capteurs, il est essentiel d'avoir une passerelle LoRaWAN à proximité afin que notre nœud de capteurs puisse envoyer ses données collectées sur Internet. Bien qu'il soit possible de trouver des passerelles LoRaWAN disponibles autour de vous, il se peut que vous ayez besoin d'installer votre propre passerelle si aucune n'est présente à proximité. La configuration de votre propre passerelle LoRaWAN est simple. Tout d'abord, vous devez acquérir une passerelle ; j'ai choisi la passerelle LoRaWAN LPS8v2 de Dragino. Pour utiliser votre passerelle, vous devez la connecter à un réseau LoRaWAN. J'ai choisi le réseau The Things Network (TTN), mais il est également possible de se connecter à d'autres réseaux tels que Helium ou Datacake, et même utiliser deux réseaux simultanément.

Figure 5. Ajout de votre passerelle au réseau.

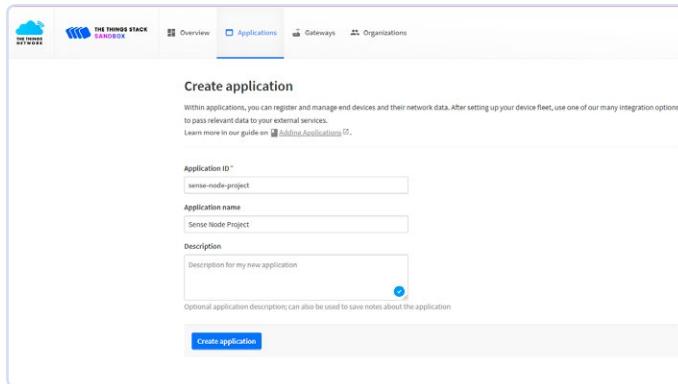


Figure 6. Création de l'application de la console TTN.

Voici un bref aperçu de la procédure de configuration :

- **Connexion au WiFi** : Commencez par mettre votre passerelle LoRaWAN sous tension et connectez-la au réseau WiFi de votre domicile. Cela permet à la passerelle d'accéder à Internet.
- **S'enregistrer sur TTN** : Ensuite, rendez-vous sur la console The Things Network et créez un compte si vous n'en avez pas déjà un. Une fois connecté, enregistrez votre passerelle en fournissant les informations nécessaires telles que l'EUI (identifiant unique de votre passerelle) et en sélectionnant votre région, comme le montre la **figure 5**.
- **Configuration de la passerelle** : Suivez les instructions fournies par TTN pour configurer votre passerelle. Cette étape inclut généralement la saisie des paramètres réseau, la sélection du plan de fréquences approprié et la configuration de la passerelle pour qu'elle communique avec les serveurs de TTN.
- **Dernières étapes** : Après la configuration, la passerelle devrait se connecter au TTN et être prête à recevoir des données de vos nœuds LoRa. Vous pouvez surveiller l'état de votre passerelle sur la console TTN pour vous assurer qu'elle fonctionne correctement.

Il existe de nombreux guides détaillés disponibles en ligne [2][3] qui fournissent des instructions étape par étape pour configurer différents types de passerelles LoRaWAN. Suivre ces guides peut vous aider à résoudre les problèmes et à vous assurer que votre passerelle est correctement configurée et connectée. Une fois la passerelle LoRaWAN configurée, nous passerons à l'enregistrement de notre nœud de capteurs sur le réseau The Things Network.

Intégration de votre projet au réseau The Things Network

Une fois votre passerelle configurée, l'étape suivante consiste à intégrer le module LoRa E5 au réseau The Things Network (TTN) [4]. Pour ce faire, commencez par enregistrer votre appareil sur la console TTN, vous devez d'abord créer une nouvelle application, puis ajouter votre appareil. Sur la console TTN cliquez sur *Create application*, donnez un ID et un nom à votre application, puis cliquez sur *Create application*, comme le montre la **figure 6**.

Ensuite, il est nécessaire d'enregistrer votre appareil au sein de l'application que vous venez de créer. Rendez-vous dans votre application puis cliquez sur *Register end device*. Au cours de cette étape, vous obtiendrez des informations d'identification essentielles telles que *Device EUI*, *Application EUI*, et *App Key*, comme le montre la **figure 7**. Ces informations sont indispensables pour configurer votre appareil afin qu'il puisse communiquer avec le TTN et seront utilisées dans le code Arduino.

Figure 7. Enregistrement de l'appareil final sur l'application.

Après avoir enregistré votre appareil dans la console TTN, vous devez configurer le format de la charge utile afin que le TTN interprète correctement les données envoyées par votre appareil. TTN vous permet de définir un formateur de charge utile personnalisé en JavaScript, qui décide les données brutes envoyées par votre appareil.

Voici un formateur de charge utile en JavaScript :

```
function Decoder(bytes, port) {
  var decoded = {};
  if (port === 8) {
    decoded.soilMoisture = (bytes[0] << 8) | bytes[1];
    decoded.temp = ((bytes[2] << 8) | bytes[3]);
    decoded.humi = (bytes[4] << 8) | bytes[5];
    decoded.co2 = (bytes[6] << 8) | bytes[7];
    decoded.battery = (bytes[8] << 8) | bytes[9];
  }
  return decoded;
}
```

Pour mettre cela en œuvre :

- **Accédez à Payload Formats** : Dans la console TTN, accédez à votre application et sélectionnez l'onglet *Payload Formats*.
- **Sélectionnez Decoder Function** : Choisissez le type de fonction *Decoder*.
- **Insérez le code** : Copiez et collez le code JavaScript fourni dans l'éditeur de la fonction de décodage.
- **Enregistrez les modifications** : Enregistrez les modifications pour appliquer le décodeur à votre application.

Cette fonction de décodage traite les octets reçus de votre nœud de capteur et les convertit en valeurs lisibles, telles que l'humidité du sol, la température, l'humidité, les niveaux de CO₂ et la tension de la batterie.

Logiciel

Le logiciel est conçu pour collecter et transmettre efficacement les données des capteurs tout en gérant la consommation d'énergie grâce à des modes de veilles. Examinons de plus près les principaux composants du code et la manière dont ils interagissent.



Listage 1. Croquis Arduino (extrait).

```
#include <Arduino.h>
#include <SCD30.h>
#include <HardwareSerial.h>
#include <config.ino>

// Defining Hardware the second internal UART -
// Serial2 for the LoRaWAN E5 Module - Pin 9 and 10
HardwareSerial Serial2(1);

...
//*** Initializing variables ***//

void setup() {

    ...
    //*** Initializing Sensors, Serial 1 & Serial 2 ***//

    // Check if the AT command returns OK
    if (at_send_check_response("+AT: OK", 100, "AT\r\n")) {
        // Set the flag to indicate the LoRa module exists
        is_exist = true;
        // Send AT command to get AppEUI and check the response
        at_send_check_response("+ID: AppEui", 1000, "AT+ID\r\n");
        // Set the LoRa module to LWOTAA mode
        at_send_check_response("+MODE: LWOTAA", 1000, "AT+MODE=LWOTAA\r\n");
        // Set the data rate to EU868
        at_send_check_response("+DR: EU868", 1000, "AT+DR=EU868\r\n");
        // Set the channel number range
        at_send_check_response("+CH: NUM", 1000, "AT+CH=NUM,0-2\r\n");
        // Set the APP Key for authentication, replace with your generated APP Key from TTN
        at_send_check_response("+KEY: APPKEY", 1000,
            "AT+KEY=APPKEY,\"C2XXXXXXXXXXXXXXXXXXXXXXXXXX\"\r\n");
            // Enter your generated APP Key here.
        // Set the LoRa module to Class A
        at_send_check_response("+CLASS: C", 1000, "AT+CLASS=A\r\n");
        // Set the port number to 8
        at_send_check_response("+PORT: 8", 1000, "AT+PORT=8\r\n");
        // Delay to ensure all commands are processed
        delay(200);
        // Print confirmation that the LoRaWAN setup is complete
        Serial.println("LoRaWAN");
        // Set the flag to indicate the module has joined the network
        is_join = true;
    }
    else {
        is_exist = false;
        Serial.print("No LoRa E5 module found.\r\n");
    }
}

void loop() {
    getSensors();
    sendData();
    // Configure the wake-up source and duration for deep sleep
    esp_sleep_enable_timer_wakeup(10 * 60 * 1000000);
        // 10 minutes in microseconds
    // Enter deep sleep mode
    esp_deep_sleep_start();
}
```

 **sense-node-garden**
ID: sense-node-garden

↑ n/a ↓ n/a ⓘ • Last activity 25 seconds ago ⓘ

[Overview](#) [Live data](#) [Messaging](#) [Location](#) [Payload formatters](#) [General settings](#)

General information

End device ID	sense-node-garden	
Frequency plan	Europe 863-870 MHz (SF9 for RX2 - recommended)	
LoRaWAN version	LoRaWAN Specification 1.0.3	
Regional Parameters version	RP001 Regional Parameters 1.0.3 revision A	
Created at	Jul 8, 2024 17:29:25	

Live data

See all activity →

- ↓ 17:56:39 Schedule data downlink for transmission on **Gateway Server** DevAddr: 26 08 FA F3
- ↑ 17:56:39 Forward uplink data message DevAddr: 26 08 FA F3 ⓘ  Payload: { c
- ↑ 17:56:39 Successfully processed data message DevAddr: 26 08 FA F3 ⓘ 
- ↑ 17:56:31 Forward join accept message DevAddr: 26 08 FA F3 ⓘ  JoinEUI: 52
- ↑ 17:56:29 Successfully processed join-request DevAddr: 26 08 37 5A ⓘ  Join
- ↑ 17:56:29 Join-request to cluster-local Join Server succeeded DevAddr: 26 08 37 5A ⓘ

Le code commence par l'inclusion des bibliothèques nécessaires et définir les configurations matérielles. Dans la fonction `setup`, nous initialisons la communication série, configurons les capteurs et le module LoRa. Ceci implique de définir les paramètres du module LoRa E5, notamment d'entrer l'*App Key* générée sur la console TTN (voir ci-dessous). Le **listage 1**, présente une version simplifiée du code, tandis que l'intégralité du code et des fichiers matériels est disponible sur le dépôt GitHub du projet [5].

Les principales opérations se déroulent dans la fonction `loop`. Elle recueille les données des capteurs, les envoie au TTN, puis met le microcontrôleur en mode « deep sleep » pour économiser de l'énergie. L'utilisation du mode « deep sleep » réduit considérablement la consommation d'énergie en désactivant la plupart des fonctionnalités du microcontrôleur et en ne le réveillant uniquement pour effectuer les tâches essentielles. Lorsqu'il est en veille profonde, le XIAO ESP32-C3 consomme environ 142 µA au lieu de 8 mA, ce qui augmente significativement l'autonomie de la batterie. En mode « deep sleep », l'ESP32C3 consomme environ 5 µA et le reste du courant est consommé par le circuit intégré de gestion de la tension et de la batterie intégré. Le microcontrôleur s'active à intervalles réguliers pour collecter et transmettre des données avant de retourner en veille.

La fonction `getSensors()` lit les valeurs du capteur d'humidité du sol et du capteur SCD30. Elle calcule également la tension de la batterie en utilisant un diviseur de tension et effectue la moyenne de plusieurs lectures pour améliorer la précision. En effectuant plusieurs lectures, le système assure des résultats de mesure stables et précis.

La fonction `sendData()` formate les données de capteur collectées dans une charge utile et les envoie au TTN via le module LoRa E5. Elle convertit essentiellement les données du capteur en une chaîne hexadécimale prête à être transmise via LoRa, comme le montre l'extrait de code ci-dessous. Cette fonction garantit que les données sont correctement formatées et transmises via LoRa en établissant d'abord une connexion à la passerelle la plus proche et en envoyant les données du capteur, ce qui permet une surveillance en temps réel. Après l'envoi des données, le module LoRa E5 est mis en mode veille pour réduire la consommation énergétique.

```
// Prepare and send the sensor data
// as a hexadecimal string via LoRa
char cmd[128];
sprintf(cmd, "AT+CMMSGHEX=
 \"%04X%04X%04X%04X%04X\"\\r\\n",
 (int)soilMoisturePercent, (int)temp,
 (int)humid, (int)co2, (int)battery);
```

App Key

Avant de charger le code, vous devez ajouter la clé d'application générée lors de l'ajout de votre appareil à l'application sur The Things Network.

Network (TTN). Cette clé est essentielle car elle sert à authentifier votre appareil auprès du réseau TTN. Elle est saisie dans la fonction `setup` et utilisée dans la fonction `at_send_check_response`.

```
at_send_check_response("+KEY: APPKEY", 1000,  
"AT+KEY=APPKEY, \"XXXXXXXXXXXXXXXXXX\"\\r\\n");  
// Enter your generated APP Key here
```

This section of the code ensures that your device is registered and authenticated, enabling it to send data to TTN. If all is done correctly, you can see the forward uplink messages from your LoRa Sense Node on your TTN device live data section, as shown in **Figure 8**. You can find the entire project repository on the GitHub [5] which includes all the files for this project.

Consommation d'énergie

Tout système fonctionnant sur batterie nécessite une gestion rigoureuse de la consommation d'énergie afin d'éviter les recharges fréquentes, toutes les deux ou trois jours par exemple. De plus, les systèmes destinés à un usage extérieur devraient nécessiter un minimum d'entretien. Mesurer la consommation d'énergie, tester le courant consommé par chaque composant et évaluer le courant moyen du système, puis calculer les coulombs pour estimer la durée de vie de la batterie sont des tâches complexes. Pour faciliter ces tâches, il est possible d'utiliser des outils tels que les profileurs de puissance ou les analyseurs d'énergie DC pour obtenir des mesures précises et analyser la consommation d'énergie de chaque composant.

J'ai récemment mis la main sur un Joulescope JS200 [6], qui a grandement simplifié ce processus. Sa taille compacte et son interface conviviale en font l'outil idéal pour tous les tests de consommation électrique.

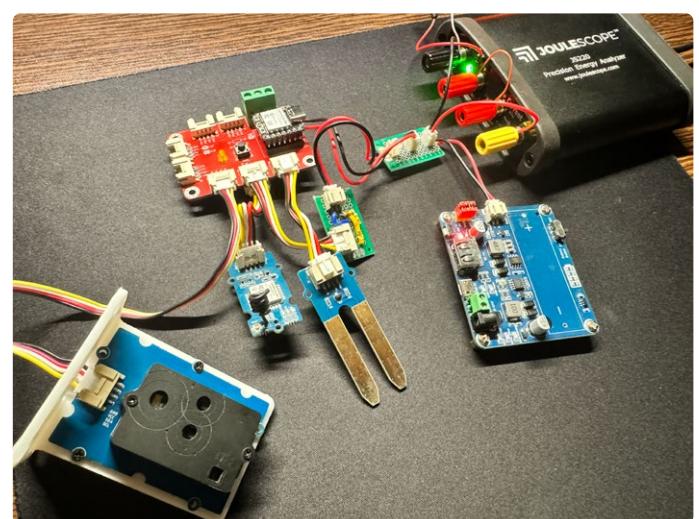


Figure 9. Consommation d'énergie du projet testé avec un JouleScope.



Figure 10. Forme d'onde du courant du SCD30 et du module LoRa.

Initialement, comme tout ingénieur, mon système n'était pas optimisé en termes de consommation d'énergie ; la priorité était de vérifier que tous les composants fonctionnaient correctement ensemble. L'optimisation du code et de la consommation d'énergie vient ensuite.

Après avoir assemblé le système initial, je l'ai connecté au Joulescope, comme le montre la **figure 9**. J'ai immédiatement remarqué que la LED IR du capteur SCD30, utilisée pour la mesure du CO₂, consommait près de 68 mA toutes les 2 secondes, ce qui résultait en un courant moyen de 25,3 mA sur 30 secondes, comme le montre la **figure 10**. En examinant la fiche technique et la bibliothèque fournie, j'ai trouvé que l'intervalle de mesure par défaut était réglé sur 2 s. C'était une solution facile car l'intervalle de mesure peut être contrôlé, de sorte que la LED IR ne s'allume que lorsqu'une mesure est nécessaire. Cela réduit les pics de courant, diminuant ainsi l'appel de courant moyen. Cependant, le capteur SCD30 consomme toujours environ 4,5 mA même lorsqu'il est inactif, et je n'ai pas encore trouvé de solution logicielle pour le réduire.

Ensuite, je me suis concentré sur le module LoRa E5 WIO, qui consomme un courant significatif de la batterie. Lors des opérations LoRa, le module consomme jusqu'à 110 mA, ce qui est acceptable, car cela ne dure qu'une seconde. Mais l'objectif est de minimiser la consommation de courant lorsque le système est en veille et que l'ESP32-C3 est en mode deep sleep. Lorsque le module est inactif, il peut consommer jusqu'à 10 mA, ce qui représente une valeur élevée pour un système alimenté par batterie sur de longues périodes. En explorant différentes ressources en ligne, j'ai découvert l'existence d'une commande AT qui peut être utilisée, pour mettre le module en mode veille, la consommation de courant de 51,7 µA : une amélioration notable. La **figure 11** illustre la forme d'onde de courant du module LoRa E5 avant et après l'activation du mode veille. Il est également possible de réduire davantage la consommation de courant du module LoRa peut être encore réduit jusqu'à 3 µA, en retirant le LDO intégré sur le module. Lors de tests ultérieurs, j'ai constaté que le capteur d'humidité du sol augmentait la consommation de courant du système jusqu'à 4 mA lorsqu'il mesurait l'humidité. Ce capteur utilise la capacité pour mesurer l'humidité, ce qui entraîne une consommation d'énergie excessive à mesure que le sol devient plus conducteur. Cela provoque également de la corrosion et l'accumulation de dépôts minéraux sur les sondes, du fait que les pistes nues du capteur fonctionnent comme un dispositif d'électrolyse. Pour atténuer le problème de la corrosion, il est possible d'utiliser un capteur de sol revêtu. Cependant, pour éliminer le courant excessif lorsque le niveau d'humidité augmente, il faut couper l'alimentation électrique du capteur lorsqu'il n'est pas utilisé, une solution

principalement matérielle. Dans la première version du projet, j'ai privilégié une approche simplifiée, axée sur les optimisations logicielles. Les versions futures prévoiront l'intégration de solutions matérielles.

Autonomie de la batterie

Cette étape nous a permis de calculer la durée de vie de la batterie du système, ce qui a été grandement facilité grâce à l'utilisation du Joulescope. Après avoir appliqué toutes les optimisations de consommation d'énergie via le logiciel, j'ai connecté le système au Joulescope. Dans la **figure 12**, vous pouvez voir la forme d'onde de courant de l'ensemble du système après ces optimisations. Notez que le SCD30 allume la LED IR uniquement lorsque l'ESP32-C3 est éveillé, puis arrête de prendre des mesures une fois que l'ESP32-C3 passe en mode veille profonde. En utilisant sa fonction avancée dans son mode multimètre, j'ai mesuré le courant moyen et les coulombs consommés par le système sur 10 minutes, période durant laquelle de nouvelles valeurs des capteurs étaient envoyées via LoRa chaque minute. Comme le montre la **figure 13**, le courant moyen était de 19,10 mA, ce qui est élevé pour mon objectif de maintenir le système en fonctionnement pendant une semaine sur une seule charge. La réduction de la fréquence de lecture des

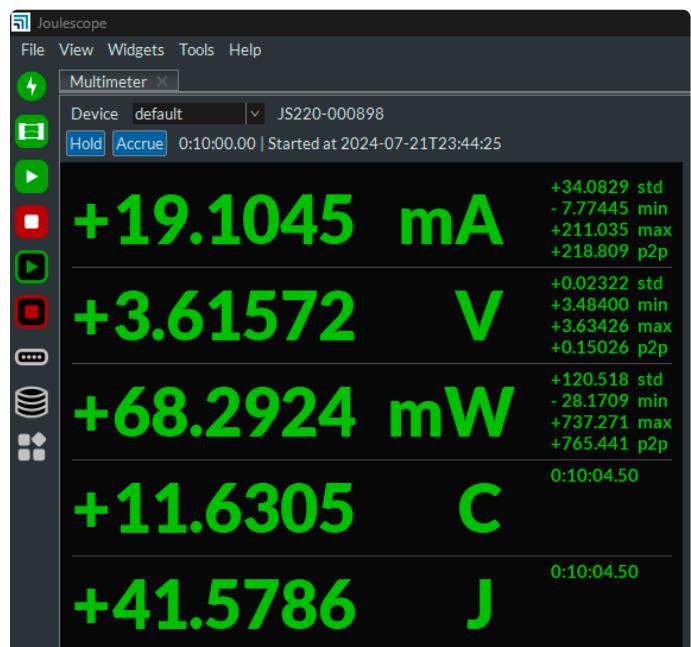


Figure 13. Capture d'écran du mode Multimètre dans JouleScope.

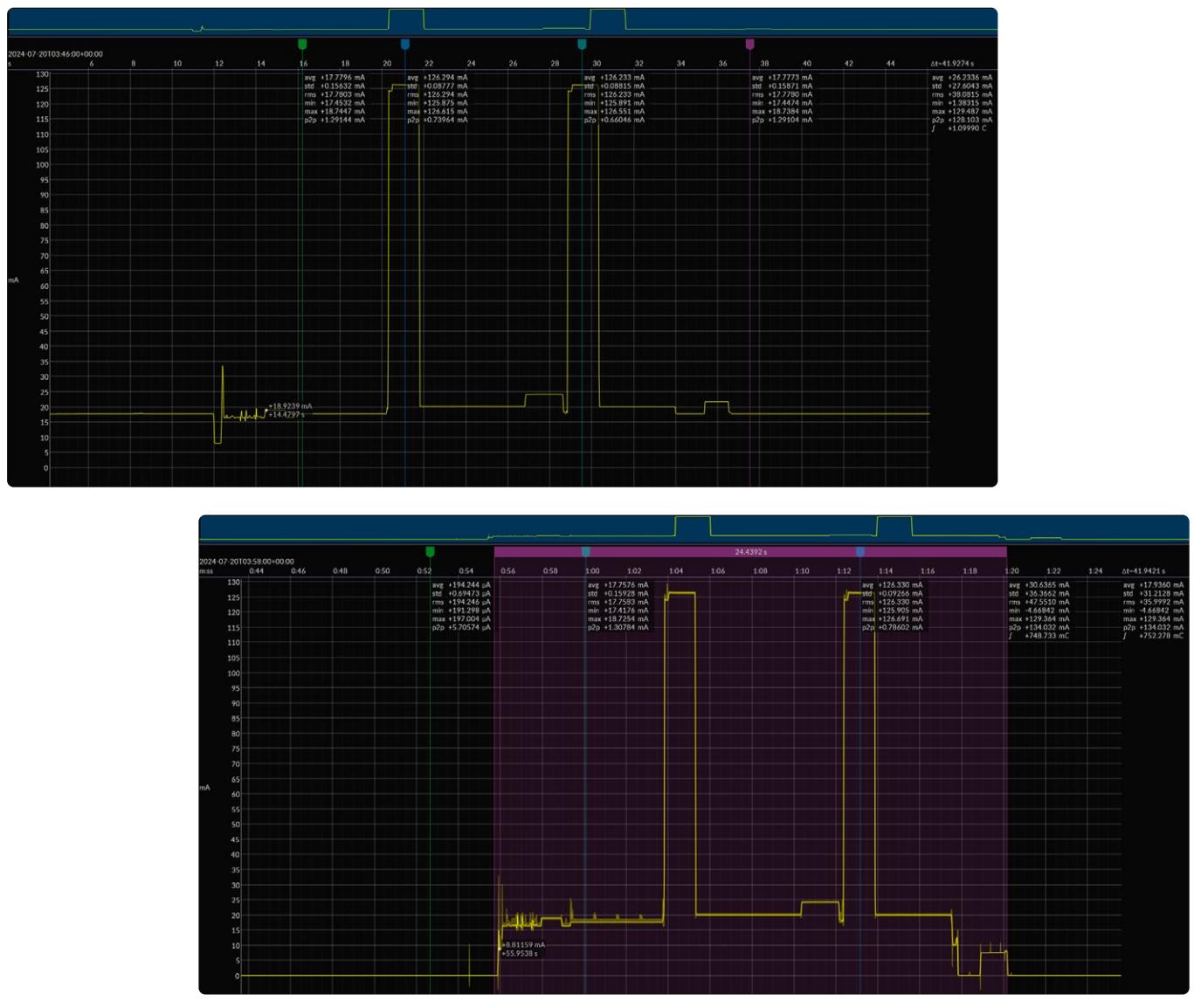


Figure 11. Consommation de courant du module LoRa E5 Modoule, avant (à gauche) et après (à droite) la mise en œuvre de la fonction sommeil.



Figure 12. Forme d'onde de courant du système entier, après les optimisations de puissance.

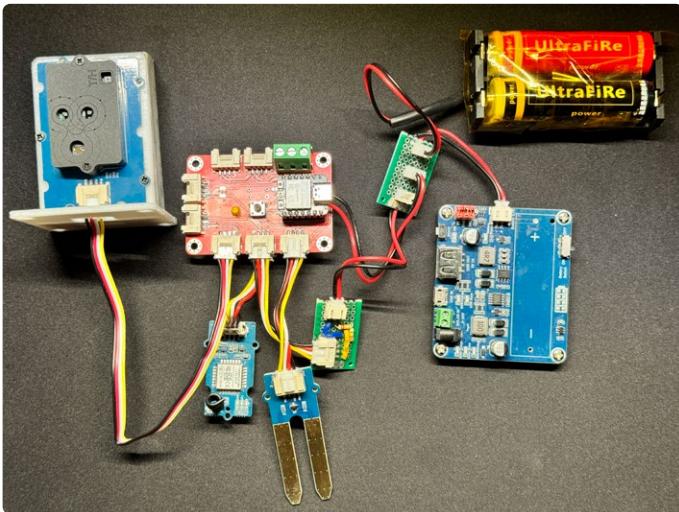


Figure 14. Tous les composants du projet assemblés et prêts à être placés dans le boîtier.

données à toutes les 10 minutes a réduit le courant moyen à 4,905 mA, rendant ainsi la consommation plus acceptable pour cette application. Le nombre de coulombs consommés au cours du temps est nécessaire pour mesurer la durée de vie de la batterie, qui était de 28,1403 C en 1 h et 35 min (5700 s), cela a été mesuré sur le Joulescope. Ensuite, il faut convertir la capacité de nos batteries en coulombs. J'utilisais deux batteries Li-ion 18650 en parallèle ayant chacune une capacité de 1800 mAh.

Capacité de la batterie en C (coulombs) :

$$3.6 \text{ Ah} \times 3600 \text{ s/h} = 12960 \text{ C}$$

Comme la tension de coupure est d'environ 3,3 V, ce qui est plus élevé que les 3 V (qui est la tension lorsque la batterie est complètement vidée), nous devons supposer environ 80 à 90% de capacité utilisable. En gardant à l'esprit le taux d'autodécharge des batteries, utilisons 85% comme approximation :

$$0.85 \times 12960 \text{ C} = 11016 \text{ C}$$

Calculons maintenant le taux de consommation en coulombs par seconde :

$$281403 \text{ C} / 5700 \text{ s} = 0.004936 \text{ C/s}$$

Enfin, pour calculer l'autonomie de la batterie, il faut diviser la capacité utilisable de la batterie par le taux de consommation pour obtenir l'autonomie de la batterie en heures :

$$(11016 \text{ C} / 0.004936 \text{ C/s}) / 3600 \text{ s/h} = 619.93 \text{ h.}$$

Ce qui représente environ 25 jours d'autonomie sur une seule charge. Sans aucune optimisation de la consommation d'énergie, l'ensemble du système n'allait durer que six jours ! Une fois que j'ai été satisfait de la consommation d'énergie après avoir éliminé d'autres LED inutiles, j'ai préparé le système pour le déploiement, comme on peut le voir sur la **figure 14**.

Composants et boîtier

J'ai conçu un boîtier personnalisé imprimé en 3D [5] pour abriter tous les composants du projet de noeud de capteurs LoRa. Résistant aux intempéries, ce boîtier principal contient efficacement tous les éléments,

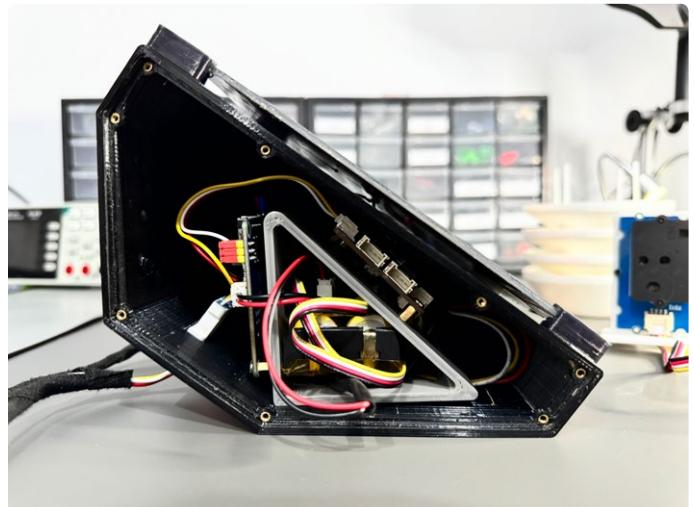


Figure 15. Composants placés dans le boîtier principal avec le panneau solaire monté sur le dessus.



Figure 16. Vue latérale de la noeud de capteur LoRa d'installation, montrant le connecteur étanche de type USB Type-C et les connecteurs étanches JST.

comme l'illustre la **figure 15**. Plusieurs facteurs ont dû être pris en compte lors de sa conception. Le boîtier assure l'étanchéité de toutes les connexions électriques, utilisant pour cela des connecteurs JST JWPF étanches spéciaux pour les capteurs externes et l'alimentation du panneau solaire. De l'époxy a été utilisé pour sceller les espaces entre les connecteurs et la structure du boîtier

Un bouton étanche a été intégré pour permettre le redémarrage, ainsi que l'allumage ou l'extinction du système. Un connecteur USB-C étanche a été utilisé pour charger de nouveaux micrologiciels ou déboguer des problèmes sans avoir à ouvrir le boîtier, comme le montre la **figure 16**. Ce port permet également de recharger la batterie en cas d'urgence, si le mauvais temps persiste plusieurs jours et que les batteries s'épuisent faute de soleil.

Pour le capteur de température, d'humidité et de CO₂ SCD30, un boîtier Stevenson personnalisé (**figure 17**), a été conçu pour protéger le capteur de la pluie, étant donné qu'il n'est pas conçu pour un usage extérieur. Ce boîtier Stevenson peut également abriter d'autres capteurs, grâce à un support de composants interne adapté à la disposition du circuit imprimé du module Grove, permettant l'installation de jusqu'à quatre petits capteurs Grove.

Après l'installation du système, comme le montre la figure 18, ses capacités d'étanchéité ont été rapidement mises à l'épreuve. Malgré une semaine de fortes pluies et une tempête, le système est resté sec à l'intérieur, et tous les composants ont continué à fonctionner parfaitement.



Figure 17. Vue en contre-plongée du nœud de capteur LoRa, présentant le boîtier Stevenson.



Figure 18. Nœud de capteurs LoRa déployé.

Datacake sous la section API.

Ensuite, créez un compte sur Datacake si ce n'est pas déjà fait, et configurez un nouveau dispositif correspondant à votre nœud de capteur LoRa. Lors de la configuration du dispositif, fournissez des détails tels que le Device EUI et connectez-le à The Things Network. Cela garantit que les données envoyées depuis TTN sont reçues par Datacake. Dans la console TTN, naviguez vers votre application et allez à l'onglet *Payload Formats*. Pour Datacake, nous utiliserons le même format de charge utile pour décoder les données reçues de TTN. Collez simplement le même code de formatage de charge utile JavaScript dans la section de décodeur de charge utile dans l'onglet de configuration du dispositif sur Datacake et enregistrez les modifications une fois la configuration terminée.

Revenez ensuite à Datacake et naviguez vers le tableau de bord du dispositif que vous avez créé. Ajoutez des champs pour afficher les valeurs des capteurs, telles que l'humidité du sol, la température, l'humidité, les niveaux de CO₂ et l'état de la batterie. Définissez le type de données approprié pour chaque champ et mappez les champs de charge utile de TTN aux champs correspondants sur Datacake comme illustré à la **figure 19**. Cette cartographie assure que les données reçues de TTN sont correctement affichées sur votre tableau de bord Datacake.

Fields
Fields describe the data the device will store.

[+ Add Field](#) [Live data](#)

NAME	IDENTIFIER	TYPE	ROLE	CURRENT VALUE	LAST UPDATE
Temperature	TEMP	Float	Primary	0	19 minutes ago
Humidity	HUMI	Float	N/A	0 %	19 minutes ago
CO2	CO2	Integer	Secondary	0 ppm	19 minutes ago
soilMoisture	SOILMOISTURE	Integer	N/A	0 %	19 minutes ago

Configuration Fields
Configuration fields hold a static value and can have a product-wide default value, that can be overwritten on a device level. They can be accessed in decoders.

[+ Add Configuration Field](#)

NAME	IDENTIFIER	FIELD TYPE	DESCRIPTION	VALUE
!				
No fields have been created, yet Create configuration fields to define configuration variables.				

Figure 19. Configuration sur le terrain de Datacake des données de capteurs reçues du TTN.

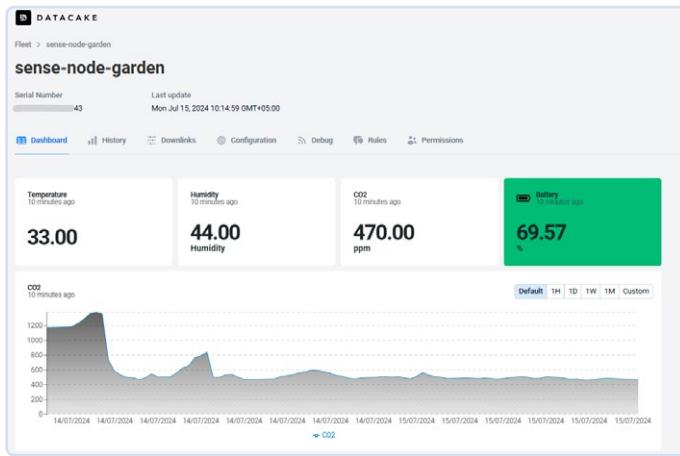


Figure 20. Tableau de bord Datacake montrant toutes les données de capteurs collectées.

Enfin, vérifiez que les données de votre nœud de capteur LoRa sont bien transmises à TTN et redirigées vers Datacake. Votre tableau de bord Datacake devrait maintenant afficher les valeurs des capteurs en temps réel, vous permettant de surveiller et d'analyser les données efficacement, comme le montre la figure 20. Cette intégration offre une plateforme puissante pour visualiser et gérer vos données de capteurs, facilitant le suivi des conditions environnementales et la prise de décisions éclairées basées sur les données collectées.

Intégration dans Home Assistant

Intégrer votre nœud de capteur LoRa dans Home Assistant [8] offre une excellente solution pour le stockage et l'analyse des données à long terme, sans les limitations de stockage imposées par des plateformes telles que Datacake. Puisque Home Assistant fonctionne sur votre propre installation domestique, il n'y a pas de limites en termes d'espace de stockage ou de points de données, ce qui le rend idéal pour une surveillance continue.

Pour intégrer Home Assistant avec The Things Network (TTN), commencez par générer une clé API dans la console TTN. Naviguez vers votre application, accédez à la section *API Keys*, et créez une nouvelle clé avec les permissions nécessaires. Assurez-vous également d'activer l'intégration de stockage dans les intégrations du tableau de bord de l'application TTN ; sinon, les données du capteur ne s'afficheront pas dans Home Assistant.

Ensuite, dans Home Assistant, allez dans *Devices & Services* et ajoutez l'intégration de The Things Network. Lorsqu'on vous le demande, entrez le nom de votre application dans TTN et la clé API que vous avez générée. Cela reliera votre instance Home Assistant à votre application TTN, permettant ainsi le flux de données de votre nœud de capteur LoRa vers Home Assistant.

Pour visualiser les valeurs des capteurs dans un graphique linéaire comme vu dans la **figure 21**, vous devez définir l'unité de mesure pour chaque entité de capteur reçue de l'intégration The Things Network. Ajoutez les entrées suivantes à votre fichier configuration.yaml dans Home Assistant :

```
sensor.lora_sense_node_temperature:
  unit_of_measurement: "°C"
sensor.lora_sense_node_co2:
  unit_of_measurement: "ppm"
sensor.lora_sense_node_humidity:
  unit_of_measurement: "%"
sensor.lora_sense_node_soilmoisture:
  unit_of_measurement: "%"
```



Figure 21. Graphique linéaire de l'historique dans Home Assistant.

Dans ces entrées, `sensor.lora_sense_node_temperature` est le nom de l'entité où Home Assistant reçoit les valeurs de température du TTN. De même, définissez l'unité de mesure pour les capteurs de CO₂, d'humidité et d'humidité du sol. Cette configuration permet de s'assurer que les valeurs des capteurs sont affichées correctement sur le tableau de bord de Home Assistant.

En complétant ces étapes, vous pouvez avoir une carte détaillée et interactive sur le tableau de bord de votre Home Assistant, comme le montre la **Figure 22**, montrant toutes les valeurs des capteurs que votre nœud de capteur LoRa envoie. Cette configuration fournit une solution robuste et flexible pour la surveillance et l'analyse des données environnementales, en exploitant les capacités de stockage et l'interface personnalisable de Home Assistant.

Améliorations futures et potentiel pour diverses applications

Bien que le système actuel de nœud de capteur LoRa soit efficace, il présente un potentiel important d'amélioration, notamment en termes d'efficacité énergétique. Au repos, le système consomme environ 4,32 mA, ce qui est relativement élevé pour des applications alimentées par batterie sur le long terme. En intégrant une horloge en temps réel (RTC) externe avec un circuit de verrouillage de puissance, nous pourrions réduire considérablement la consommation au repos jusqu'à atteindre 50 nA.

Dans cette configuration améliorée, l'horloge RTC externe serait contrôlée par le microcontrôleur, et cette RTC gérerait le circuit de verrouillage de puissance. Ce circuit couperait l'alimentation de l'ensemble du système lorsqu'il n'envoie pas activement des valeurs de capteurs et la réactiverait à des intervalles prédefinis. Cette approche prolongerait considérablement la durée de vie de la batterie, rendant le système plus adapté pour des déploiements à distance et de longue durée. Cette solution pourrait être mise en œuvre dans la prochaine version du tableau d'expansion ou sous forme d'un module séparé pouvant être ajouté à la configuration existante. Restez à l'écoute pour un article à ce sujet !

De plus, d'autres optimisations matérielles peuvent être explorées. Par exemple, remplacer les composants à haute consommation d'énergie par des alternatives plus efficaces, optimiser le firmware pour assurer une utilisation minimale de l'énergie, et améliorer l'efficacité du système de charge solaire sont toutes des améliorations potentielles.

La nature modulaire de ce système permet l'ajout de divers capteurs pour étendre sa fonctionnalité. En intégrant davantage de capteurs, cette configuration peut être utilisée pour une large gamme d'applications, y compris, mais sans s'y limiter :

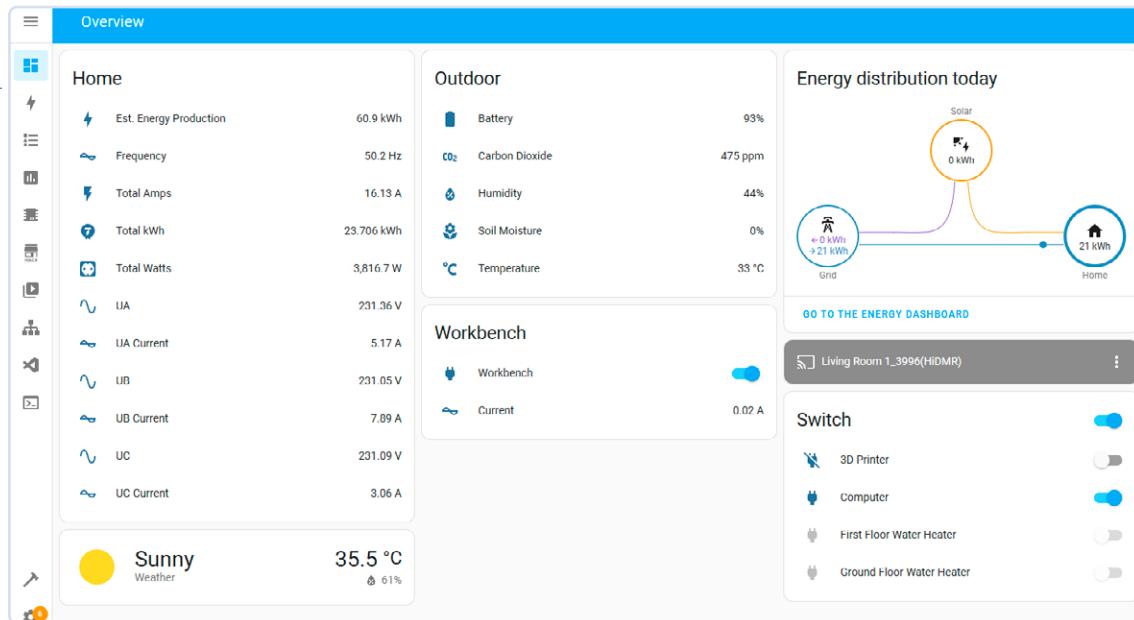


Figure 22. Données du nœud de détection LoRa affichées sur le tableau de bord de Home Assistant.

- **Surveillance environnementale :** L'ajout de capteurs de qualité de l'air, d'intensité lumineuse et de niveaux sonores peut faire de ce système une station de surveillance environnementale complète.
- **Applications agricoles :** L'intégration de capteurs de pH du sol, de capteurs de nutriments et de capteurs météorologiques peut fournir des données précieuses pour l'agriculture de précision.
- **Surveillance industrielle :** Inclure des capteurs de fuite de gaz, de vibration et de pression peut aider à surveiller les environnements industriels pour la sécurité et l'efficacité

En conclusion, le système actuel de nœud de capteur LoRa offre une plateforme robuste et flexible pour la surveillance environnementale à distance. Avec des améliorations futures axées sur l'efficacité énergétique et l'intégration de capteurs supplémentaires, ce système a le potentiel d'être adapté à diverses applications, en faisant un outil inestimable pour divers besoins de surveillance. 

240354-04

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 avec une expertise approfondie en systèmes embarqués et en

développement de produits. Son expérience lui a permis de collaborer avec une variété d'entreprises, allant de startups innovantes à des multinationales renommées, en menant des projets de prototypage et de développement à la pointe de la technologie. Fort d'une riche expérience dans l'industrie aéronautique et ayant dirigé une startup technologique, Saad apporte à Elektor un mélange unique d'expertise technique et un esprit entrepreneurial affirmé. Il contribue au développement de projets dans les domaines du logiciel et du matériel.

Produits

- **Seeed Studio XIAO ESP32C3**
www.elektor.fr/20265
- **Seeed Studio Grove SCD30 CO2**
www.elektor.fr/20012
- **Seeed Studio LoRa-E5 STM32WLE5JC**
www.elektor.fr/19956
- **Waveshare Solar Power Management Module**
www.elektor.fr/20488
- **Dragino LoRa/LoRaWAN IoT Kit v3 (EU868)**
www.elektor.fr/20775

LIENS

- [1] Saad Imtiaz, "Elektor eXpansion Board v1.0," Elektor 7-8/2024: <https://elektormagazine.com/240250-01>
- [2] Setting up The Things Network V3 on Dragino: <http://wiki.dragino.com/xwiki/bin/view/Main/Notes%20for%20TTN/>
- [3] Dragino LPS8N - Setup with The Things Network: <https://www.thethingsindustries.com/docs/gateways/models/dragino-lps8/>
- [4] The Things Network: <https://www.thethingsnetwork.org/>
- [5] LoRa Sensor Node Github Repository: <https://github.com/ElektorLabs/lora-sensor-node>
- [6] Joulescope JS220: Precision Energy Analyzer: <https://www.joulescope.com/products/js220-joulescope-precision-energy-analyzer>
- [7] Datacake: <https://datacake.co/>
- [8] Home Assistant: <https://www.home-assistant.io/>



la carte eXpansion v1.0 d'Elektor

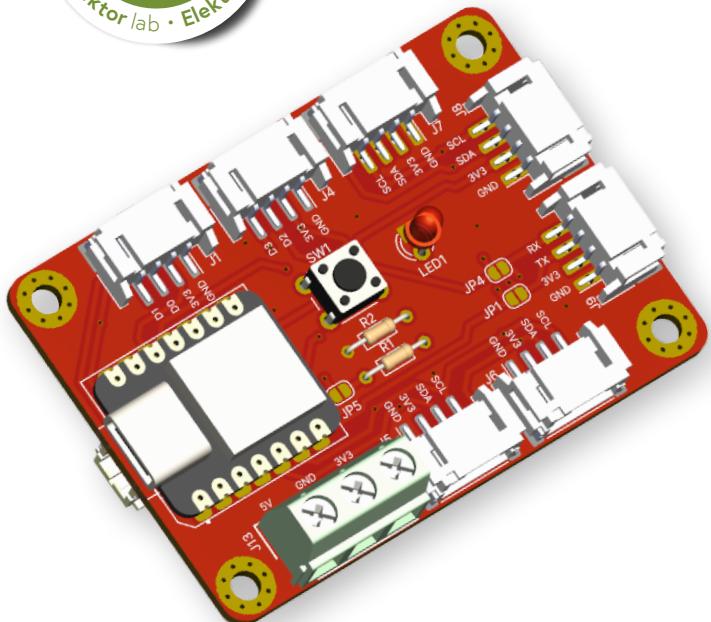
pour ESP32S3 et autres cartes XIAO

Saad Imtiaz et Jens Nickel (Elektor)

Compactes, abordables et performantes ! La famille Seeed XIAO comprend sept cartes breakout de la taille d'un timbre avec des contrôleurs variés tels que l'ESP32, le RP2040 et le SAMD21. Toutefois, pour faciliter la connectivité durant la phase de prototypage, l'utilisation d'une carte breakout s'avère indispensable. La carte eXpansion d'Elektor intègre de nombreux connecteurs (bien au-delà de l'écosystème Seeed), ainsi qu'un bouton utilisateur, une LED et des fonctions de protection.

Dans le monde en constante évolution de l'électronique, disposer d'outils de développement à la fois flexibles et fiables est crucial. De nombreux développeurs ont éprouvé la frustration de travailler avec des cartes d'extension qui ne répondent pas adéquatement aux besoins de connectivité et de protection. Cette carte d'extension pour les minuscules cartes d'extension de microcontrôleur de Seeed Studio XIAO [1] propose une solution versatile et robuste, conçue pour surmonter efficacement ces défis.

Nous avons conçu notre carte pour le projet de mesure de puissance AmpVolt [2], mais, la carte eXpansion peut être utilisée dans divers autres projets nécessitant un ESP32-C3 ou -S3 (ou tout autre contrôleur de la famille XIAO - voir l'encadré), et peu de broches. Les cartes XIAO offrent 11 broches GPIO au total (D0..D10), utilisables pour diverses fonctions telles que SPI, I²C, et UART, ainsi que pour des entrées et sorties numériques et analogiques. Les cartes XIAO présentent une compatibilité étendue au sein de la gamme. Par exemple, vous trouvez



rez toujours les broches I²C aux positions D4 et D5. Cela facilite le prototypage !

Il est important de noter que D4 et D5 correspondent à l'emplacement des broches sur la carte XIAO, et non aux broches GPIO 4 et 5 du microcontrôleur. Lors de la programmation avec l'EDI Arduino, le numéro de broche à utiliser pour des fonctions telles que `digitalRead(...)` dépend de la carte XIAO que vous utilisez - veuillez vous référer au **tableau 1** et à la documentation de la carte XIAO.

Schéma

La conception de notre carte d'extension est axée sur l'optimisation de la connectivité et des fonctionnalités. Le schéma de la carte est illustré à la **figure 1**. Les éléments principaux sont les connecteurs d'extension, qui rendent les 11 broches GPIO (D0..D10) d'une carte XIAO de votre choix facilement accessibles. Les cartes XIAO font partie de l'écosystème Grove [3], et sont compatibles avec une vaste gamme de modules périphériques Grove. Pour cette raison, notre

Cartes XIAO

Actuellement, 7 cartes sont disponibles, intégrant des contrôleurs de différents fabricants. Pour en savoir plus sur leurs caractéristiques, voir [1].

- XIAO ESP32S3 Sense
- XIAO ESP32S3
- XIAO ESP32C3
- XIAO nRF52840 Sense
- XIAO nRF52840
- XIAO RP2040
- XIAO SAMD21

Tableau 1. Connexions des broches de la carte XIAO aux GPIO de la carte microcontrôleur.

XIAO Board pin	ESP32-C3	SMD21	ESP32-C6	ESP32-S3
D0	GPIO2	D0	GPIO0	GPIO1
D1	GPIO3	D1	GPIO1	GPIO2
D2	GPIO4	D2	GPIO2	GPIO3
D3	GPIO5	D3	GPIO21	GPIO4
D4 (SDA)	GPIO6	D4	GPIO22	GPIO5
D5 (SCL)	GPIO7	D5	GPIO23	GPIO6
D6 (TX)	GPIO21	D6	GPIO16	GPIO43
D7 (RX)	GPIO20	D7	GPIO17	GPIO44
D8	GPIO8	D8	GPIO19	GPIO7
D9	GPIO9	D9	GPIO20	GPIO8
D10	GPIO10	D10	GPIO18	GPIO9

Lors de la programmation avec l'EDI Arduino, sélectionnez la variante XIAO correcte. Pour la définition des broches, vous devez utiliser les numéros GPIO spécifiques afin de sélectionner correctement la broche sur la carte.

Par exemple, pour faire clignoter la LED1 (connectée à D3) sur la carte eXpansion tout en utilisant une ESP32 C3, vous devez configurer LED1 comme sortie en utilisant le numéro GPIO correspondant. Pour D3, cela correspond au GPIO 5, comme démontré dans l'extrait de code ci-dessous.

```
#define LED D3 ou #define LED 5
// As GPIO5 corresponds with D3

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

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

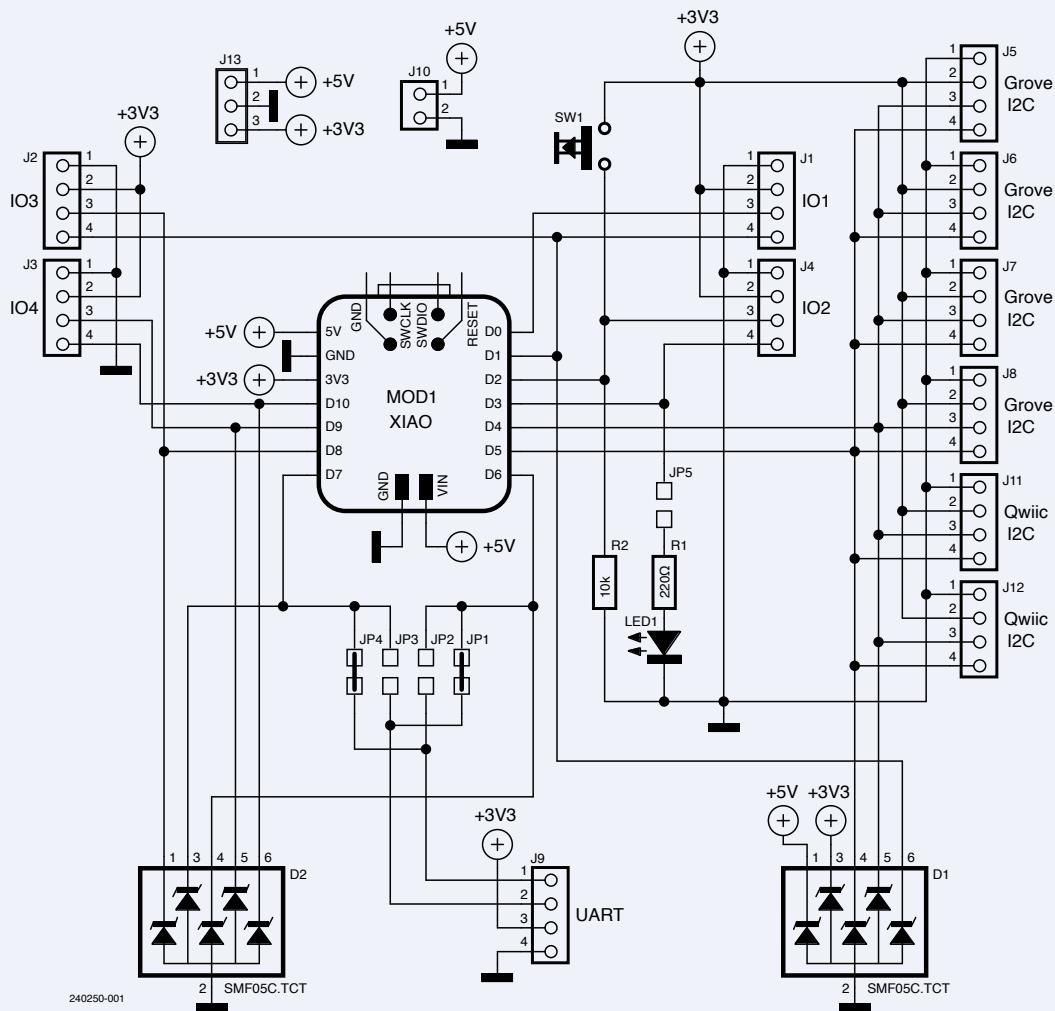


Figure 1. Schéma de la carte eXpansion V1.0.

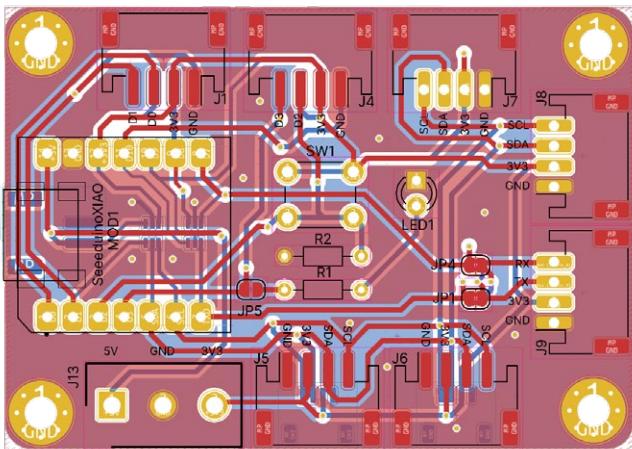


Figure 2. La couche supérieure du circuit imprimé.

carte inclut de nombreux connecteurs Grove, enrichissant ainsi ses possibilités d'utilisation. Cette caractéristique distingue nettement notre carte sur le marché.

Il existe une incroyable diversité de puces et de modules I²C, allant des capteurs aux actionneurs en passant par les éléments d'interface utilisateur comme les écrans. Comme vous le savez peut-être, il est possible de connecter de nombreux modules I²C en parallèle à l'interface I²C d'un contrôleur, à condition qu'ils possèdent des adresses I²C différentes. C'est pour faciliter cette connexion que nous avons rendu accessibles les broches I²C (D4 et D5) sur plusieurs de nos connecteurs. Nous avons ajouté quatre connecteurs I²C Grove (J5 à J8) [4] et deux connecteurs I²C Qwiic (J11 et J12) [5], permettant une intégration facile avec divers modules I²C Grove et Qwiic (voir [6]). Il va de soi qu'à la place des connecteurs Grove, vous pouvez également souder d'autres connecteurs au pas de 2 mm, tels que les connecteurs JST PH2.0 à 4 broches.

Bien entendu, tous les modules périphériques ne se connectent pas via l'interface I²C. Par exemple, les capteurs sont souvent reliés via une interface à un seul fil. Pour cette raison, nous avons également connecté les broches D0..D3 et D8..D10 à des connecteurs Grove. Les broches D8, D1, D9 et D10 peuvent ainsi constituer une interface SPI complète avec Chip Select (CS). Ces broches sont accessibles sur J2 et J3, mais il est nécessaire d'utiliser ces deux connecteurs simultanément pour exploiter cette fonctionnalité.

J9, associé aux broches D6 et D7, forme un connecteur UART compatible Grove. Afin d'accroître la polyvalence de la carte, les broches RX et TX peuvent être interchangées grâce à des cavaliers (JP1 à JP4). Cela renforce la flexibilité de la carte, permettant une communication série efficace non seulement avec des modules UART « périphériques », tels qu'un adaptateur RS-232 ou RS-485, ou un module GPS Grove [7]. En permutant les broches, vous pouvez également connecter une autre carte contrôleur avec un connecteur UART Grove ici, ou même une seconde carte eXpansion d'Elektor, pour faciliter la communication entre eux. Pour permuter les connexions RX/TX, il suffit de retirer les ponts de soudure JP1 et JP4 et de connecter les cavaliers JP2 et JP3. La carte est équipée d'une LED (LED1) et d'un bouton (SW1) destinés aux fonctions d'entrée et de sortie de base. Ils sont connectés aux broches D2 et D3, respectivement. La LED sert d'indicateur d'état, tandis que le bouton peut être utilisé pour les entrées de l'utilisateur. De plus, LED1 peut être isolée en dessoudant le cavalier JP5, ce qui

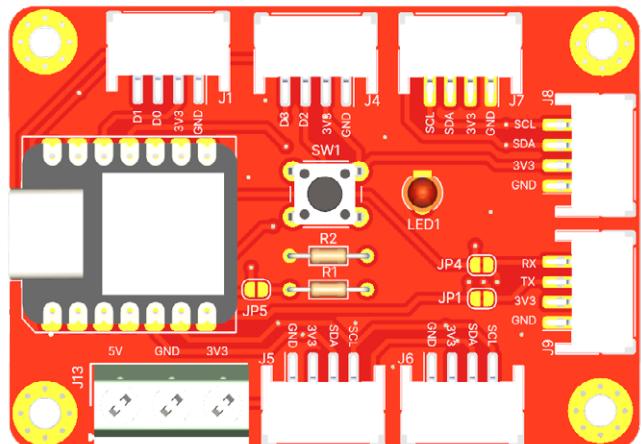


Figure 3. Les faces supérieure et inférieure du circuit imprimé.

permet aux utilisateurs de déconnecter la LED si elle n'est pas nécessaire, libérant ainsi la broche GPIO correspondante.

De plus, nous avons ajouté une protection ESD sur les lignes 3,3 V, 5 V, I²C et SPI en utilisant les diodes TVS SMF05C.TCT (D1 et D2) - Vous trouverez des détails supplémentaires sur cette protection dans les sections suivantes.

La carte est aussi dotée d'un connecteur à bornes à vis pour les tensions 3,3 V, 5 V, et pour la masse, ce qui facilite l'alimentation et/ou la mise à la masse d'autres modules et composants électroniques. Pour les applications nécessitant une alimentation par batterie, un connecteur au pas de 2 mm est aussi disponible, rendant cette carte idéale pour les projets à la fois fixes et portables.

Circuit imprimé

La disposition du circuit imprimé est illustrée à la **figure 2**. Pour réduire la taille du circuit imprimé tout en préservant la connectivité, nous avons également ajouté des connecteurs I²C et IO sur la face arrière du circuit imprimé, comme le montre la **figure 3**. Cela a été possible grâce à l'utilisation de connecteurs CMS qui sont plus compacts que



Liste des composants

Résistances

R1 = 220 Ω

R2 = 10 k Ω

Semi-conducteurs

D1, D2 = SMF05C.TCT

LED1 = LED 3 mm 3 V

MOD1 = Carte microcontrôleur série XIAO

Divers

J1...J9 = Connecteur Grove à 4 broches (pas de 2 mm)

J10 = connecteur JST PH à 2 broches (pas de 2 mm)

J11, J12 = Connecteur Qwiic à 4 broches (pas de 1 mm)

J13 = Bloc terminal à 3 broches (pas de 5,08 mm)

SW1 = Bouton poussoir de 6 mm

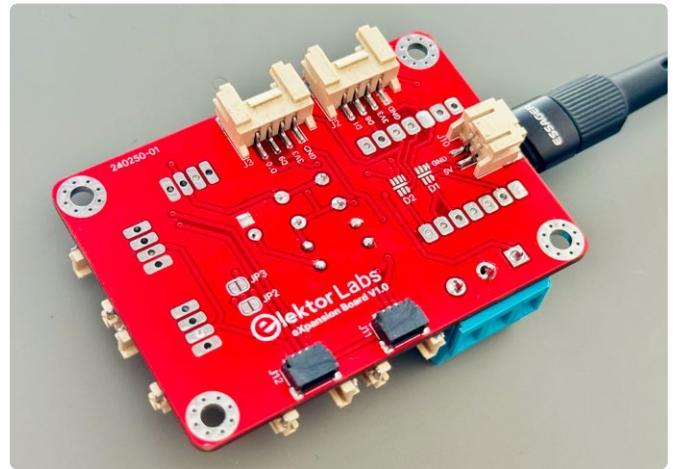
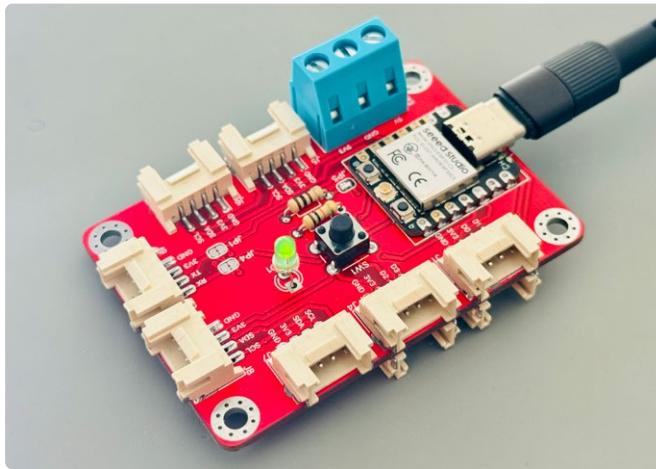


Figure 4. La carte eXpansion V1.0 d'Elektor

les connecteurs traversants ordinaires. De plus, pour l'UART et deux des connexions I²C, un support THT est également prévu. Ainsi, si vous n'avez plus de connecteurs CMS, vous pouvez utiliser des connecteurs THT.

L'agencement des composants et des connecteurs garantit que la carte reste compacte tout en offrant une haute fonctionnalité, répondant ainsi à une vaste gamme de besoins, comme illustré à la **figure 4**. Nous avons également conçu un boîtier imprimé en 3D pour la carte eXpansion, comme le montre la **figure 5**. Ce boîtier assure un accès facile à tous les connecteurs, ce qui facilite son intégration dans divers projets. Dans la **figure 6**, la carte eXpansion est alimentée par une batterie LiPo.

Importance de la protection ESD

En industrie, les appareils électroniques sont souvent exposés à diverses contraintes environnementales, parmi lesquelles la décharge électrostatique (ESD). Les décharges électrostatiques se produisent lorsqu'un flux soudain de courant est transféré entre deux objets chargés électriquement, généralement à la suite d'un contact. Sans

une protection adéquate contre les décharges électrostatiques, les microcontrôleurs et autres composants sensibles peuvent subir des dommages irréversibles.

Les microcontrôleurs (MCU) sont particulièrement sensibles aux décharges électrostatiques (ESD) en raison de leurs structures internes complexes et fragiles. Une ESD peut générer des pics de haute tension qui dépassent la capacité de tolérance des circuits, provoquant des défaillances immédiates ou latentes. Ces défaillances peuvent se manifester par une dégradation des performances, une perte des données ou une défaillance de l'appareil. Dans le contexte des applications industrielles où la fiabilité est cruciale, comme dans l'automatisation industrielle, les dispositifs médicaux ou les capteurs en environnement extérieur, les conséquences peuvent être désastreuses. Pour atténuer ces risques, la carte est équipée d'une protection ESD sur les lignes 3 V, 5 V, I²C et SPI. Cette protection garantit que toute décharge électrostatique produite sur le terrain est efficacement dissipée, évitant ainsi qu'elle n'atteigne et n'endommage le microcontrôleur sensible. Cette caractéristique est essentielle pour maintenir l'intégrité et la fiabilité du dispositif dans les applications pratiques.



Figure 5. Boîtier imprimé en 3D.



Figure 6. La carte eXpansion alimentée par une batterie LiPo.

Améliorations futures et applications

Cette carte d'extension est compatible avec tous les microcontrôleurs de la série XIAO. Elle représente un atout pour les développeurs, les éducateurs et les amateurs, en facilitant particulièrement le prototypage rapide, les projets éducatifs et les applications IdO. Elle offre une connectivité étendue et des fonctions avancées de gestion de l'alimentation. Tous les fichiers de conception KiCad et les fichiers de production sont disponibles sur le dépôt GitHub d'Elektor Lab dédié à ce projet [8].

À l'avenir, nous envisageons d'exploiter pleinement cette carte pour développer une variété de projets, en utilisant ses multiples connecteurs I²C et Qwiic pour intégrer aisément divers capteurs et modules. De plus, étant donné que la série XIAO ESP32 est équipée de capacités Wifi, cette carte est particulièrement adaptée pour le contrôle à distance et les applications IdO. Elle ne répond pas seulement aux besoins actuels mais pave également la voie à des innovations futures dans le domaine de l'électronique et des applications de microcontrôleurs. 

240250-04

Questions ou commentaires ?

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



Produits

- **Seeed Studio XIAO ESP32C3**
www.elektor.fr/20265
- **Seeed Studio XIAO ESP32S3 Sense**
www.elektor.fr/20705
- **Seeed Studio Arduino Sensor Kit (Base)**
www.elektor.fr/20068

LIENS

- [1] Seeed Studio XIAO: https://wiki.seeedstudio.com/SeeedStudio_XIAO_Series_Introduction
- [2] Saad Imtiaz, « AmpVolt : module de mesure de puissance (1) », Elektor 5-6/2024 :
<https://www.elektormagazine.fr/magazine/elektor-344/62820>
- [3] Grove Ecosystem Introduction : https://wiki.seeedstudio.com/Grove_System/
- [4] Seeed Studio Grove 10x Universal 4-pin Connector (2 mm Pitch) : <https://elektor.fr/18671>
- [5] Qwiic JST Connector - SMD 4-pin (Horizontal) : <https://sparkfun.com/products/14417>
- [6] SparkFun Environmental Combo Breakout - CCS811/BME280 (Qwiic) : <https://elektor.fr/19580>
- [7] Seeed Studio Grove GPS Module : <https://elektor.com/20022>
- [8] Elektor eXpansion Board v1.0 - Dépôt Github : <https://github.com/ElektorLabs/eXpansion-board>

Rejoignez notre communauté



www.elektormagazine.fr/community

elektor
design > share > earn

☰ Toggle OV2640 settings



une **caméra** dans un **train miniature**

installation d'un module ESP32 CAM

Gilbert Ghyselbrecht (Belgique)

Découvrez comment améliorer votre train miniature grâce au module ESP32-CAM, abordable et polyvalent. Suivez étape par étape l'installation de la caméra dans votre locomotive, la création d'une carte mère sur mesure pour l'alimentation électrique et la configuration de l'indispensable logiciel.



Figure 1. Une locomotive qui convient pour le module caméra.

Lorsqu'on m'a demandé s'il était possible d'installer une caméra dans la locomotive d'un train miniature, j'ai immédiatement pensé à un module ESP32-CAM. Il est disponible partout à très bas prix et avec différentes options d'objectifs et de caméras, et des câbles plats de différentes longueurs pour la connexion de la caméra au module. Cet article décrit l'installation, une carte mère pour l'alimentation électrique et le logiciel nécessaire.

Tout d'abord, j'ai dû trouver une locomotive adaptée avec suffisamment de place pour le module ESP32-CAM. Elle doit de préférence être munie d'un capot en plastique si on veut loger l'antenne Wi-fi à l'intérieur. J'ai trouvé un modèle approprié dans la série HO de Märklin, à savoir la locomotive électrique allemande de type B103 (figure 1). En déplaçant le module de décodage numérique du train, j'ai libéré suffisamment d'espace pour le module caméra et une carte mère de ma conception (figure 2).



Figure 2. L'espace disponible dans la locomotive pour le module camera.

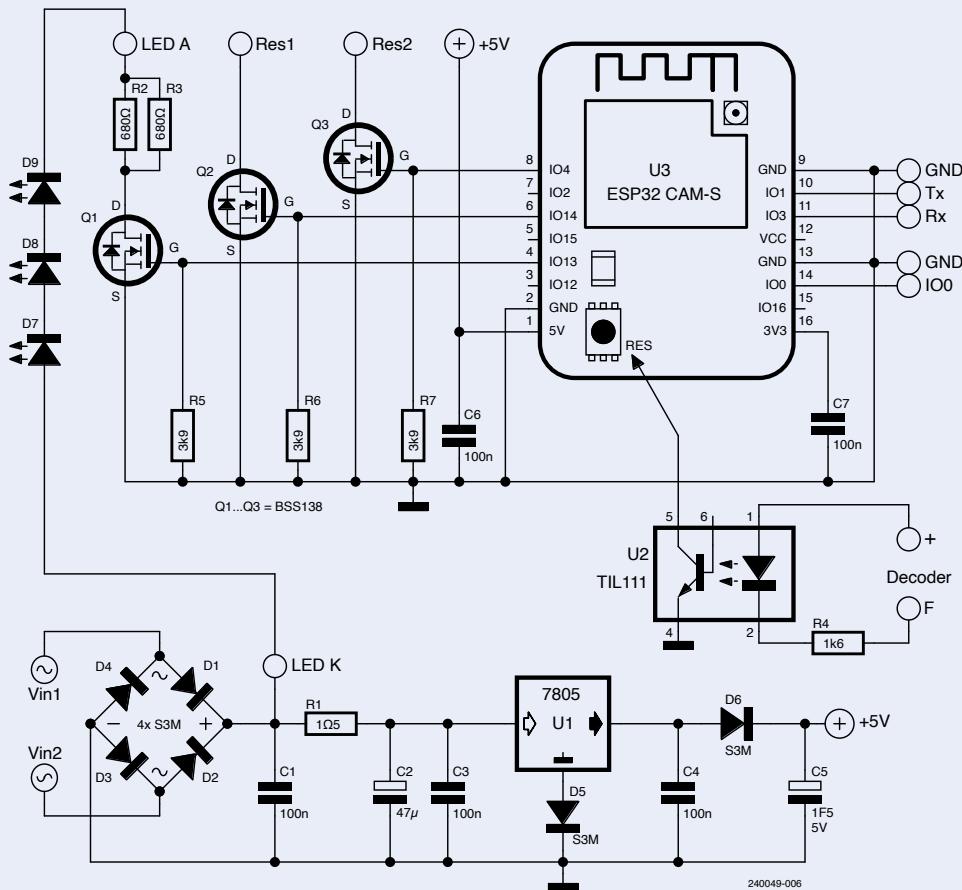


Figure 3. Schéma de la carte d'alimentation.

Alimentation électrique

Un deuxième défi consistait à assurer une alimentation fiable de 5 V pour l'ESP32. Nous supposons ici que les voies sont commandées numériquement. Avec une voie analogique à l'ancienne utilisant un transformateur, il n'y a pas de tension sur les rails lorsque le train est immobile. Avec un système numérique, en revanche, il y a toujours une tension alternative d'environ 18 V sur les rails, en provenance de l'unité de commande. Peu importe le système ou la marque utilisés. Le courant est fourni au train par les roues (ou, dans le système Märklin, par les roues et un contact coulissant sur le rail central). Cependant, cette alimentation n'est pas stable à 100 % lorsque le train roule, car de brèves interruptions peuvent parfois se produire au niveau des aiguillages ou lorsque la voie est sale. Ceci peut conduire à la réinitialisation de l'ESP32 ou à la détection d'une baisse de tension : La connexion Wi-fi est interrompue et tout doit être redémarré.

J'ai rapidement rejeté l'idée d'utiliser une petite batterie comme « alimentation de secours », car cela aurait nécessité un circuit supplémentaire pour la charger en toute sécurité. Il aurait aussi fallu un circuit pour couper l'alimentation au bout d'un moment afin d'empêcher la batterie de se décharger complètement. Vu l'espace limité dans la locomotive, impossible d'y intégrer tout cela. Une solution consisterait à loger l'alimentation à batterie dans un wagon séparé et à l'atteler derrière la locomotive, c'est-à-dire en d'autres termes, la renaissance du tender arrière.

Après de nombreuses tentatives, j'ai décidé d'utiliser un supercondensateur de 5 V et 1,5 F de dimensions $25 \times 17 \times 9$ mm [1]. Grâce à cela, la caméra continue à fonctionner normalement pendant environ huit secondes après une interruption de l'alimentation électrique, ce qui est suffisant pour un fonctionnement fiable dans des conditions normales.

Le circuit

La **figure 3** montre le schéma complet du circuit. En bas à gauche, la tension de 18 V provenant des rails est connectée au pont redresseur (D1...D4). Le reste du circuit ne doit pas entrer en contact avec les parties métalliques de la locomotive ! La résistance R1 de 1,5 Ω limite les pointes de courant excessives et sert également de fusible. Le régulateur de tension U1, un 7805 ordinaire, est relié à la surface métallique de la locomotive par une plaque isolante qui sert à dissiper la chaleur. La diode D5 dans la ligne de masse du 7805 compense la perte dans la diode en série D6 vers le supercondensateur C5, qui est ainsi chargé à 5 V. Le supercondensateur est chargé très rapidement, en 3 s environ, et le courant de charge est limité à environ 1,5 A par la protection contre les courts-circuits du régulateur 7805 (**figure 4**).



Figure 4. Charge du supercondensateur mesuré sur R1.

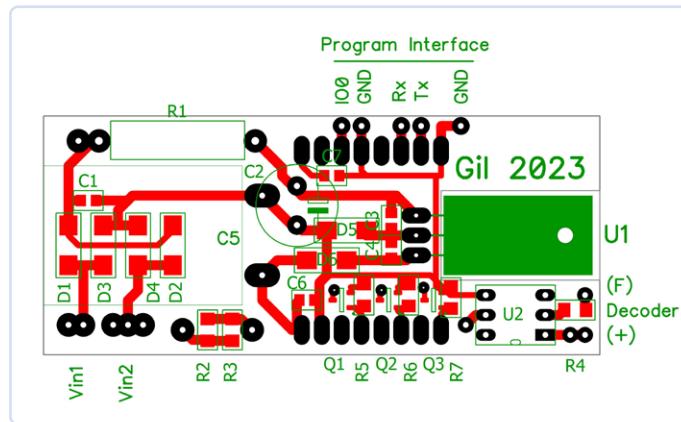


Figure 5. Conception du circuit imprimé de la carte d'alimentation.



Figure 7. Trois LED blanches à l'avant de la locomotive.

Pour l'alimentation électrique, un circuit imprimé simple face a été conçu pour s'adapter à l'espace disponible dans la locomotive. Les pièces principales sont des composants traversants, mais ils sont montés en surface. Le circuit imprimé est monté sur le châssis avec le côté cuivre vers le haut et fixé en place avec le régulateur 7805 coudé (voir **figures 5 et 6**).

Les broches de deux barrettes à huit broches sont pliées et soudées à la carte pour brancher le module ESP32-CAM. J'ai commandé deux types de caméra OV2640 avec un câble de connexion de 75 mm de long, l'une avec un objectif d'un angle de vue de 66° et l'autre de 120°. Selon l'espace disponible, la caméra appropriée est installée à l'avant de la locomotive, derrière le pare-brise. Toutefois, la fenêtre en plastique d'origine du train devant la caméra doit être retirée, car elle n'est pas suffisamment transparente.

Sorties et autres

Trois LED blanches lumineuses (D7...D9) sont fixées à l'avant de la locomotive pour bien éclairer la voie dans les sections sombres et les tunnels (**Figure 7**). Les trois LED sont connectées en série et sont commandées par un signal MLI (broche IO13 du module ESP) via le tampon MOSFET Q1 (un BSS138) et une résistance en série de 340 Ω (deux de 680 Ω en parallèle). La luminosité des LED peut être réglée via l'ESP32.



Figure 6. La carte d'alimentation dans la locomotive, fixée par le 7805.

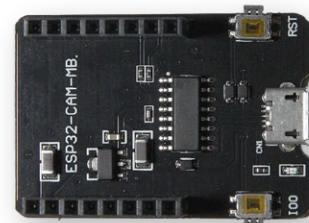


Figure 8. Module de programmation pour l'ESP32-CAM.

Deux autres sorties avec des tampons BSS138 sont disponibles pour d'autres applications, commandées via IO14 et IO4. La consommation de courant du module est de 250 mA lorsque la connexion Wi-fi est active et tombe à 15 mA au repos.

Une barrette dédiée à 3 broches sert à la programmation via un adaptateur USB-FTDI. Connectez IO0 à GND et appuyez sur le bouton de réinitialisation du module pour le mettre en mode de programmation. Il est cependant plus facile de retirer le module de la locomotive et de le brancher sur une carte de programmation ESP32-CAM (ESP32-CAM-MB) pour le programmer (**figure 8**). Ce module spécial est connecté à l'EDF Arduino via USB et comprend les boutons BOOT et RST nécessaires à l'ESP32-CAM.

Dès que le module ESP32-CAM est installé dans le train, le bouton de réinitialisation situé sur le module lui-même n'est plus accessible. Un fil supplémentaire est donc soudé au bouton de réinitialisation et connecté à la broche 5 de l'optocoupleur U2 sur la carte mère (**figure 9**). Une autre option consiste à installer un bouton de réinitialisation supplémentaire accessible de l'extérieur. La sortie AUX du décodeur du

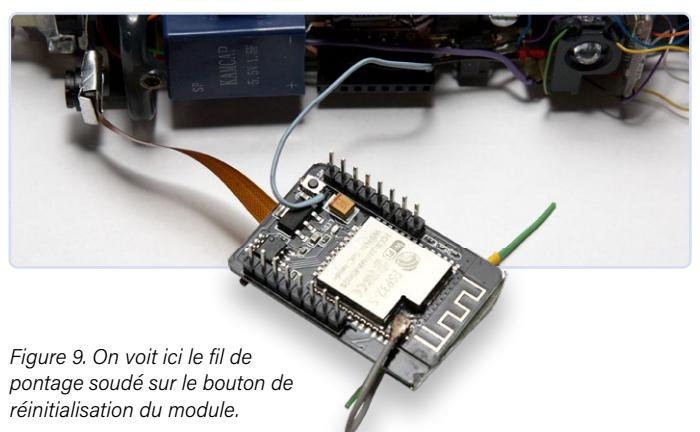


Figure 9. On voit ici le fil de pontage soudé sur le bouton de réinitialisation du module.



train sert à commander U2, qui met l'entrée de réinitialisation du module ESP à l'état bas, ce qui met la caméra hors tension ou la redémarre. J'ai remplacé l'antenne Wi-fi du module ESP par une petite antenne dipôle fabriquée à partir d'un morceau de fil et reliée par un court câble coaxial doté d'un minuscule connecteur IPEX approprié. Ce câble est parfois fourni avec le module ESP, mais vous pouvez aussi facilement l'acheter (avec un connecteur SMA à l'autre extrémité) dans un magasin spécialisé. Une petite modification doit être apportée à la carte du module : Il faut couper un pont de soudure à côté du connecteur d'antenne et en placer un autre [2]. La longueur d'onde à 2,4 GHz est de 12,5 cm. Le dipôle se compose de deux morceaux de fil de 3 cm de long, dont l'un est relié au conducteur intérieur et l'autre au blindage du câble coaxial. Une gaine thermo rétractable maintient le tout en place. L'antenne peut être montée à l'intérieur ou à l'extérieur de la locomotive, mais de préférence pas trop près du moteur ou d'autres pièces métalliques (**figure 10**).

Logiciel

Le module caméra est programmé avec l'EDI Arduino 1.8 ou 2.0. On trouve des instructions d'installation à [3]. Ouvrez *Fichier / Exemples* dans l'EDI Arduino et sélectionnez **ESP32 Camera CameraWebServer**. Après avoir adapté les champs SSID et PASSWORD, le code peut être compilé et chargé dans le module caméra.

Placez ensuite la locomotive sur les rails et allumez votre unité de commande du train. Utilisez l'unité de commande pour activer et désactiver la sortie AUX afin de réinitialiser et de démarrer l'ESP-CAM. Le module se connecte à votre réseau local sans fil (WLAN). Dans le

navigateur de votre ordinateur, entrez l'adresse IP que le routeur a attribuée au module (normalement 192.168.1.xxx ou 198.168.0.xxx) afin de vous connecter à la caméra. L'adresse correcte est affichée dans le moniteur série de l'EDI Arduino, réglé sur 115200 bauds. Vous pouvez jouer avec tous les paramètres du module caméra dans l'exemple de logiciel de test, mais le programme est complexe et parfois difficile à comprendre. J'ai donc développé un programme simple [6] qui ne contient que les paramètres les plus importants et dans lequel les pages de l'interface web sont stockées dans des fichiers d'en-tête séparés (lisibles et accessibles).

Si la connexion au WLAN échoue, le module passe en mode point d'accès (AP) en utilisant l'adresse 192.168.4.1 avec le SSID **ESP32-CAM** (sans mot de passe) et permet ainsi la connexion avec un navigateur sans routeur ni connexion Internet.

La page web est créée à l'aide des trois fichiers compilés avec les fichiers d'en-tête :

1. *index.html* in `#include "indexFile.h"`
2. *espcam.css* in `#include "cssFile.h"`
3. *espcam.js* in `#include "jsFile.h"`

Dès que le module est démarré et que la connexion Wi-fi est établie, les LED blanches à l'avant de la locomotive s'allument très faiblement. L'interface web, qui figure en tête de l'article, offre la possibilité de prendre une photo, de la sauvegarder ou de lancer un flux vidéo. La résolution de l'image peut être réglée, une résolution plus faible entraînant une fréquence d'images plus élevée.

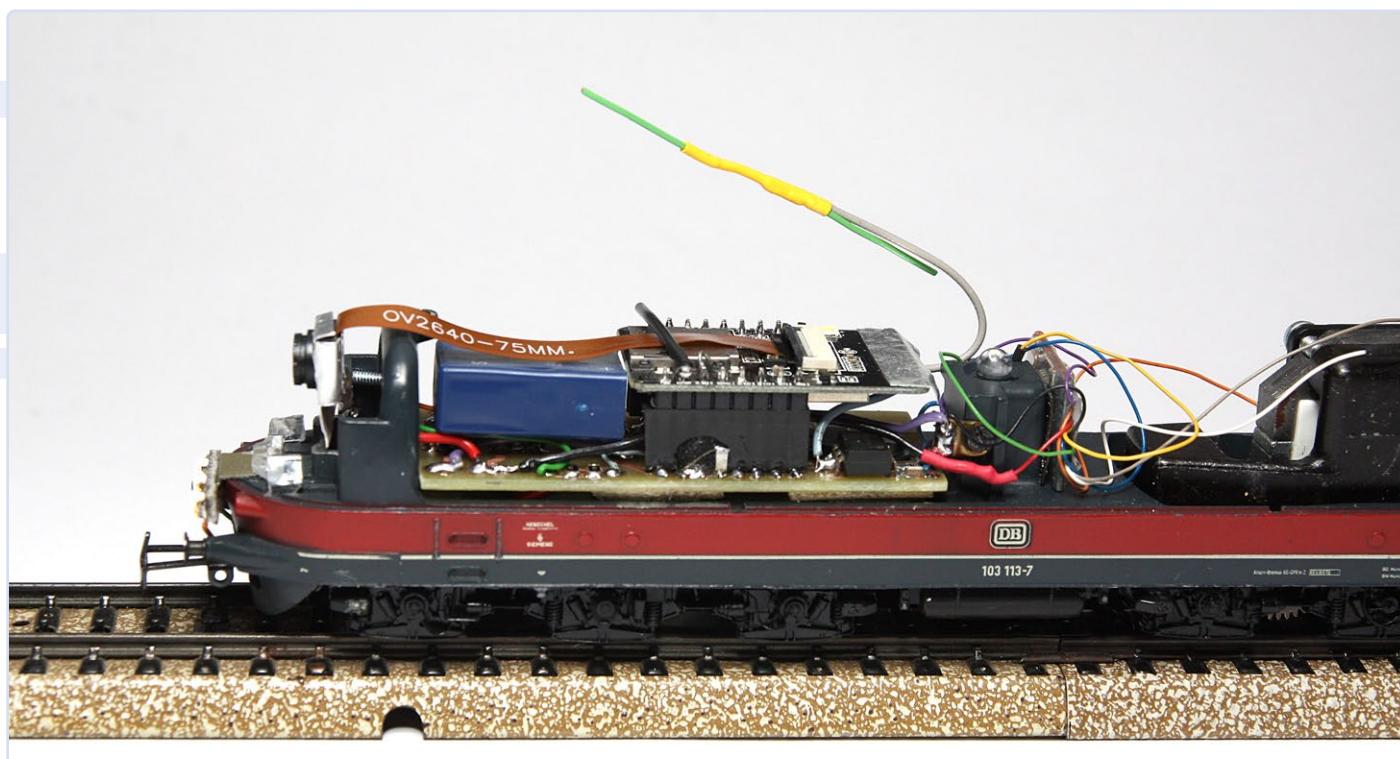


Figure 10. Le projet terminé avec l'antenne dipôle.



Flux vidéo

Le flux vidéo est un flux MJPEG [4] qui dépend du client (le navigateur web). Le navigateur maintient la connexion ouverte et effectue des requêtes image par image via JavaScript. La vitesse du navigateur et la qualité de la connexion Wi-fi déterminent le nombre d'images traitées par seconde. Si vous le souhaitez, ce nombre peut être contrôlé à l'aide d'un oscilloscope sur la broche IO12.

Un menu permet de sélectionner des images en couleur, en noir et blanc ou en sépia. Enfin, la luminosité des LED blanches situées à l'avant de la locomotive peut être réglée à l'aide d'un curseur. Notez que les LED sont très lumineuses lorsqu'elles sont réglées au maximum. Enfin, j'aimerais mentionner le programme supplémentaire, encore plus puissant, esp32-cam-webserver sur GitHub [5], qui fournit toutes les fonctions de la caméra et permet de charger le programme *over the air* (OTA) directement dans la caméra de la locomotive sans avoir à ouvrir la locomotive ou à la retirer des rails.

Ce projet

Il ne s'agit pas d'un produit clé en main, car beaucoup de choses dépendent du modèle de locomotive spécifique dans lequel vous souhaitez intégrer la caméra. Toutefois, le concept d'une alimentation sans interruption (ASI) avec supercondensateur pourrait également être intéressant pour d'autres applications. 

VF : Denis Lafourcade — 240049-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (gilbert.ghyselbrecht@skynet.be), ou contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

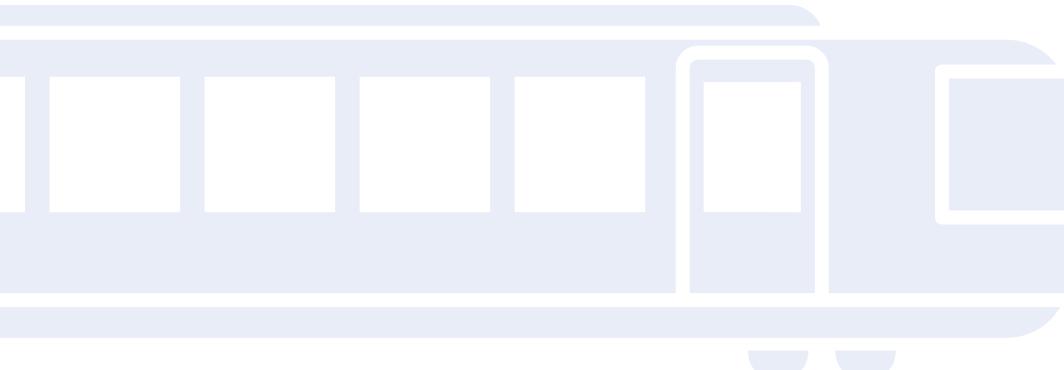
Gilbert Ghyselbrecht a longtemps travaillé comme technicien à la station radio-maritime Ostende Radio. À l'époque, il était responsable de la maintenance des équipements de diffusion. Il s'est intéressé en outre au développement et à la réalisation de systèmes de télécommande qui permettaient de commander ces équipements à distance. Son expertise a débuté avec le microprocesseur 8085 et le logiciel CP/M puis sur les commandes vocales avancées (VoIP).



Produit

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

Cette version du module ESP32-CAM ne nécessite pas d'adaptateur de programmation additionnel, mais elle est légèrement plus grande. Veuillez vérifier avant !

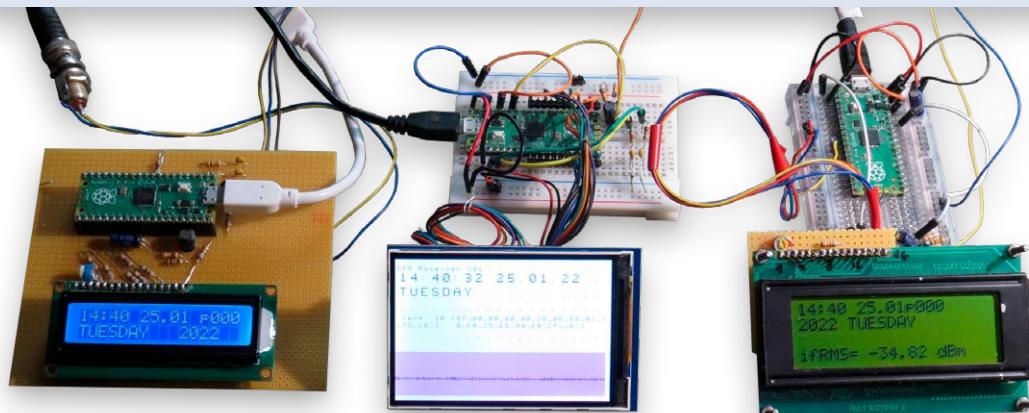


LIENS

- [1] Kamcap: 5.5V Fully Sealed (Waterproof) Supercapacitor :
[https://www.kamcappower.com/products/5.5v-finally-sealed-waterproof-super-capacitor/](https://www.kamcappower.com/products/5.5v-fully-sealed-waterproof-super-capacitor/)
- [2] Random Nerds Tutorials: ESP32-CAM Connect External Antenna:
<https://randomnerdtutorials.com/esp32-cam-connect-external-antenna/>
- [3] Random Nerds Tutorial: ESP32-CAM Video Streaming ... (details on the ESP32-CAM module) :
<https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/>
- [4] Wikipedia: Motion JPEG : https://en.wikipedia.org/wiki/Motion_JPEG
- [5] GitHub: ESP32 CAM Webserver : <https://github.com/easytarget/esp32-cam-webserver>
- [6] Circuit imprimé et logiciel : <https://www.elektormagazine.fr/240049-04>

antenne magnétique à large bande pour les grandes ondes

tous les émetteurs sans aucun accord



Réception simultanée sur une seule antenne de signaux horaires à diverses fréquences avec trois récepteurs.

Martin Ossmann (Allemagne)

L'antenne magnétique utilise la composante magnétique (et non l'électrique comme le dipôle) du champ électromagnétique des ondes radio pour les capter. L'important ici, c'est la caractéristique large bande d'une telle antenne. Notre antenne magnétique associe donc sensibilité élevée et grand confort, puisqu'il n'y a pas besoin de la régler laborieusement sur la fréquence de réception.

Pour capter des émissions en GO (grandes ondes) ou en PO (petites ondes) avec une antenne ferrite ou une bobine cadre, vous élaborerez généralement un circuit résonnant L/C avec par ex. un condensateur variable et réalisez ainsi une antenne accordée comme le circuit illustré (**figure 1**).

L'avantage est une sensibilité élevée due à la forte résonance. Mais le revers de la médaille, est qu'il faut réaccorder le circuit résonant chaque fois que l'on change de station. Nous allons voir comment construire

une antenne magnétique qui n'a pas besoin d'être accordée car elle est à large bande. On obtient une antenne large bande en associant la bobine réceptrice à un amplificateur à transimpédance : voir **figure 2**. L'ampli à transimpédance (idéal) a une impédance d'entrée de 0Ω et court-circuite donc la bobine réceptrice L. Cela élimine aussi l'effet de la capacité parasite de la bobine. Le signal reçu est le courant dans la bobine virtuellement court-circuitée.

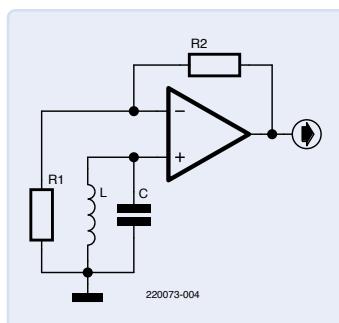


Figure 1. Circuit de récepteur avec circuit résonant L/C et ampli op.

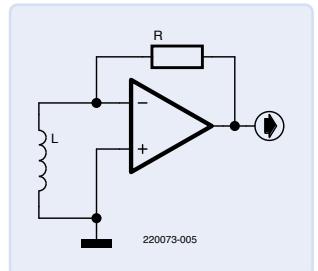


Figure 2. Un amplificateur à transimpédance court-circuite la bobine réceptrice L.

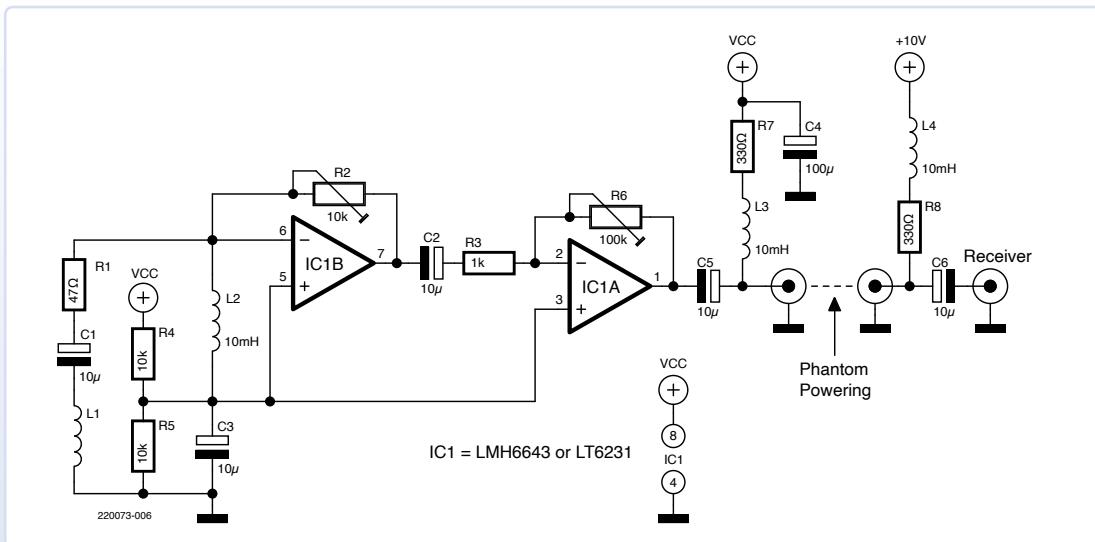


Figure 3. Schéma du circuit de l'antenne à large bande mise en œuvre.

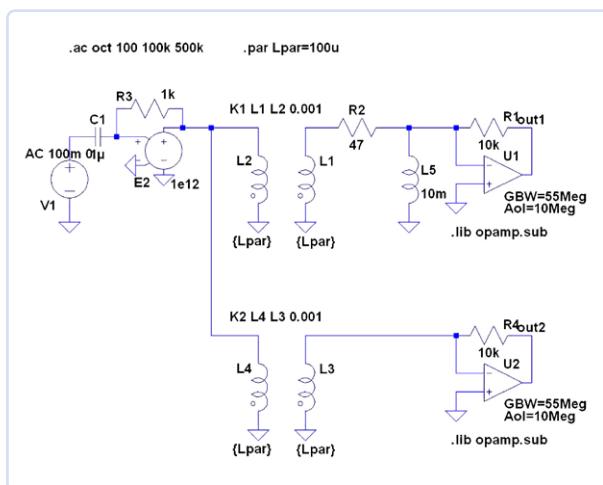


Figure 4. Circuit simulateur LTspice de l'amplificateur à transimpédance.

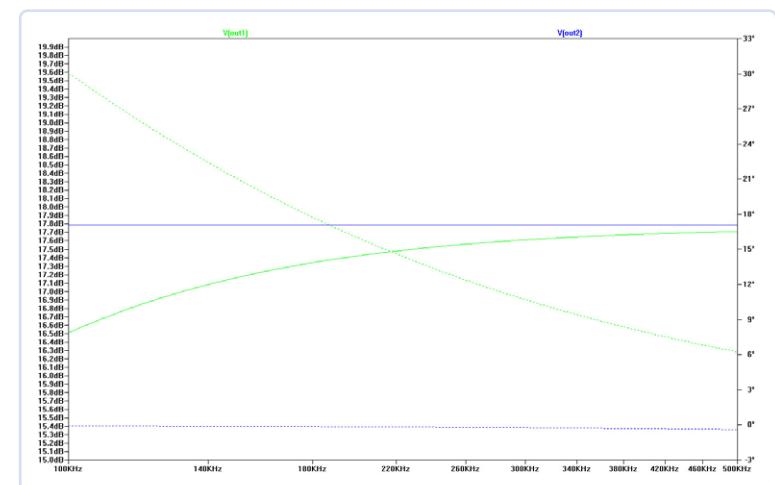


Figure 5. Réponse en fréquence des circuits simulés : En vert le circuit réel, en bleu le circuit idéal.

Circuit avec amplificateur à transimpédance

Observez le circuit final avec amplification à 2 étages et alimentation fantôme (figure 3). Avec un ampli à transimpédance à 1 étage simple (fig. 2), le circuit a tendance à osciller. Pour pallier ce phénomène, R1 et L2 ont été ajoutés pour réduire le gain de la boucle et donc sa tendance à osciller.

Nous avons simulé ce schéma dans LTspice pour étudier l'influence de R1 et L2 sur le fonctionnement du circuit. Observez le circuit de simulation (figure 4). La numérotation des composants est différente de la fig. 3 ! Le filtre passe-haut composé de C1 et R3 - ainsi que de E2 - simule la loi de l'induction, selon laquelle la tension induite par le champ magnétique reçu est proportionnel à sa fréquence. L'effet du champ magnétique sur la bobine réceptrice L1 est simulé par la bobine émettrice L2 faiblement couplée. Le script LTspice est téléchargeable depuis la page de cet article [1].

Examinez la courbe tension de sortie V(out1) / fréquence (figure 5). Pour comparer, la tension de sortie du circuit original V(out2) est aussi représentée. On note que L5 et R2 ne causent qu'une faible chute de la réponse en basse fréquence.

Cadre ou barreau de ferrite ?

Observez le circuit assemblé (figure 6). La bobine L1 est enroulée sur un barreau de ferrite (Ø10 cm long, 200 mm, 20 tours de fil de cuivre émaillé de 0,3 mm de diamètre). Notez la réponse en fréquence de notre circuit, mesurée avec un analyseur de spectre HP3585A (figure 7).

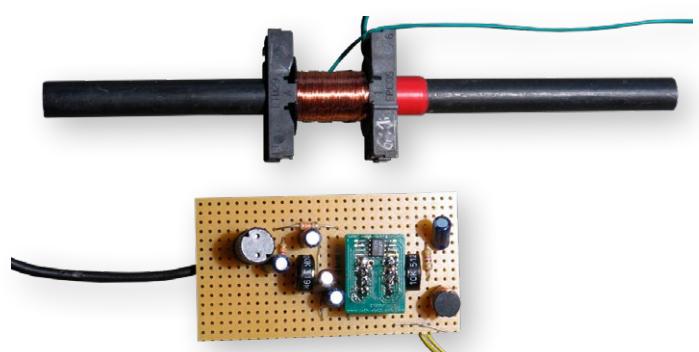


Figure 6. Barreau de ferrite avec bobine et amplificateur à transimpédance sur plaque d'essai.

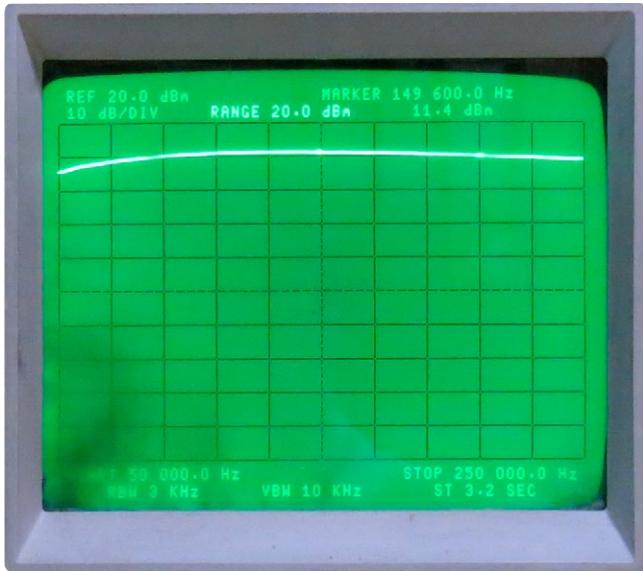


Figure 7. Réponse en fréquence (50 à 250 kHz) de l'antenne large bande avec barreau de ferrite.

Le générateur d'excitation dont la fréquence change en synchronisme avec le balayage du récepteur émet un signal sinusoïdal continu qui alimente une bobine de Helmholtz, au centre de laquelle est fixé le barreau de ferrite (**figure 8**). En GO, de 50 kHz à 250 kHz, la réponse en fréquence est relativement constante.

Au lieu de l'antenne en ferrite, on peut utiliser une antenne à cadre aérien pour la réception. Nous avons enroulé 20 tours de fil de cuivre émaillé de 0,3 mm sur une croix en bois, (**figure 9**), ce qui donne un carré de 30 cm de côté.

Nous avons procédé à des essais de réception dans la gamme GO de 60 kHz à 234 kHz avec les variantes antenne ferrite et bobine cadre. Le **tableau 1** donne la tension de sortie pour les stations reçues. Comme on le note, une série de stations de radio européennes de la gamme GO sont reçues parfaitement dans la ville privilégiée d'Aix-la-Chapelle, en Allemagne.

L'antenne magnétique à large bande peut recevoir plusieurs stations en même temps. C'est une caractéristique très intéressante. L'illustration de tête montre, respectivement, la station DCF77 à 77,5 kHz, DCF49 (service EFR) à 129,1 kHz, et TDF à 162 kHz, bien reçues simultanément sur une seule antenne. Les trois émetteurs envoient les données horaires que nous voyons sur les écrans LCD. Trois récepteurs SDR simples basés sur Raspberry Pi Pico sont mis en œuvre [2].

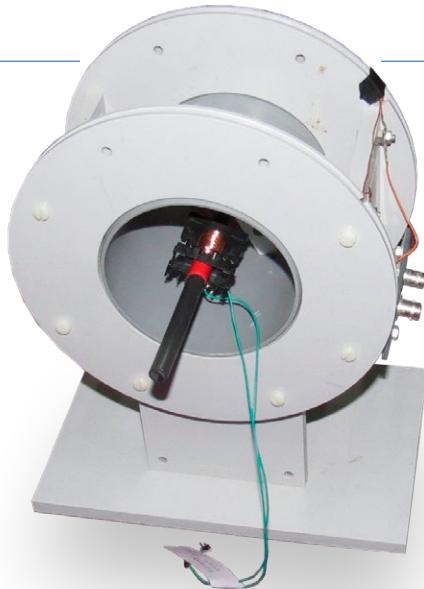


Figure 8. Dispositif de mesure de la réponse en fréquence à bobine de Helmholtz d'émission.

Ceci conclut la présentation de notre antenne magnétique à large bande. Avec les composants idoines, ce concept peut certainement être transposé en PO et en OC. Patience à ce sujet, un article est en gestation. 

VF : Yves Georges — 220073-04



Figure 9. Bobine de 30x30 cm à 20 tours de fil de cuivre émaillé.

Tableau 1. Données de réception de divers émetteurs de signaux horaires.

Émetteur	Fréquence (kHz)	Tension antenne ferrite (dBm)	Tension bobine cadre (dBm)	Puissance de l'émetteur (kW)	Distance depuis Aix-la-Chapelle (km)
MSF	60	-52	-49	17	500
DCF77	77,5	-35	-25	50	220
DCF49	129,1	-35	-43	100	220
DWD	147,3	-44	-39	20	410
TDF	162	-30	-25	2.000	490
BBC	198	-35	-26	500	580
RTL	234	-30	-22	750/375	120



Questions ou commentaires ?

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

À propos de l'auteur

Martin Ossmann commença à lire Elektor à douze ans - et à bricoler, bien sûr. Il fit des études d'électrotechnique et occupa quelques années un poste d'ingénieur de développement, puis embrassa une carrière de professeur au département d'électrotechnique et d'infotechnologie de l'université des sciences appliquées d'Aix-la-Chapelle. Il est l'auteur de d'articles scientifiques, mais publie aussi régulièrement des circuits et des projets logiciels avec une grande expertise technique dans Elektor depuis plus de trente ans. published circuits and software projects with a great deal of technical expertise in Elektor for more than three decades.



Produits

- **Raspberry Pi Pico RP2040 WH**
www.elektor.fr/20225
- **Raspberry Pi 5 RTL-SDR V4 (Bundle)**
Dogan and Ahmet Ibrahim, Raspberry Pi 5 for Radio Amateurs (Elektor 2020)
www.elektor.fr/20878

LIENS

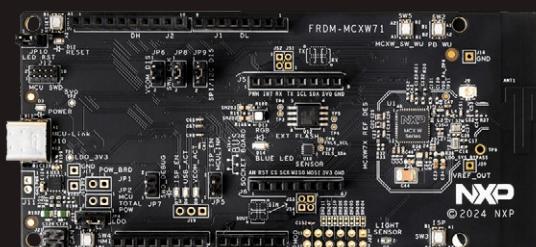
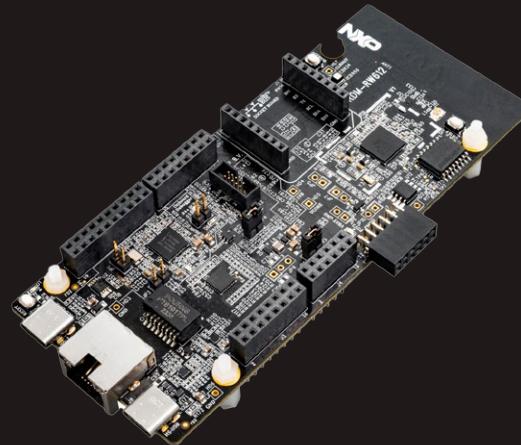
- [1] Script LTspice sur la page de cet article :
www.elektormagazine.fr/220073-04
- [2] M. Ossmann, "Radio logicielle MSF utilisant un Raspberry Pi Pico", [elektormagazine.fr, 07/2022](http://elektormagazine.fr/magazine/elektor-267/60603) :
www.elektormagazine.fr/magazine/elektor-267/60603



Passez au réseaux sans fil

Les derniers microcontrôleurs pour les réseaux sans fil de NXP permettent un développement rapide vers les technologies Bluetooth, Matter, Wi-Fi, Thread et Zigbee avec les cartes faible cout FRDM boards et outils logiciels MCUXpresso developer experience

Visitez www.nxp.com/mcxw et www.nxp.com/rw612 pour en savoir plus



TensorFlow Lite pour microcontrôleurs

par un débutant, pour les débutants

Jean-François Simon (Elektor)

TensorFlow Lite pour microcontrôleurs permet de faire tourner des modèles d'apprentissage automatique (Machine Learning, ou ML) sur des plateformes aux ressources limitées. Voyons cela de plus près et utilisons-le avec Edge Impulse pour la reconnaissance vocale sur un Arduino Nano 33 BLE Sense.

L'intelligence artificielle (IA) et l'apprentissage automatique (ML) sont les nouveaux mots à la mode, parfois utilisés de manière un peu abusive. Facebook, Amazon, Google et bien d'autres utilisent des systèmes d'apprentissage automatique pour vous proposer des contenus les plus adaptés à vos goûts et à vos habitudes. ChatGPT est un autre exemple de service très spectaculaire et populaire utilisant la ML. Le point commun de ces entreprises est qu'elles ont accès à des serveurs dotés d'une énorme puissance de calcul pour entraîner les modèles en traitant de très grands volumes de données, et pour répondre de manière fluide aux requêtes d'un grand nombre d'utilisateurs.

C'est toutefois en train de changer avec l'émergence de l'IA « en périphérie », c'est-à-dire l'utilisation d'algorithmes d'intelligence artificielle en périphérie du réseau, à savoir plus près de la source de données et loin d'un serveur. Cela permet d'analyser les données et de prendre des décisions en temps réel, en réduisant la latence et les besoins en bande passante. Bien que la notion de réseau soit souvent mise en avant, le concept fonctionne également sans réseau du tout - par exemple, sur une modeste carte à microcontrôleur, qui n'est pas nécessairement connectée à Internet.

TensorFlow Lite pour microcontrôleurs

Une évolution intéressante s'est produite dans ce domaine il y a quelques années avec l'apparition de TensorFlow Lite pour microcontrôleurs (TFLite Micro [1]). Il s'agit d'une version allégée de TensorFlow, un framework d'apprentissage machine open-source

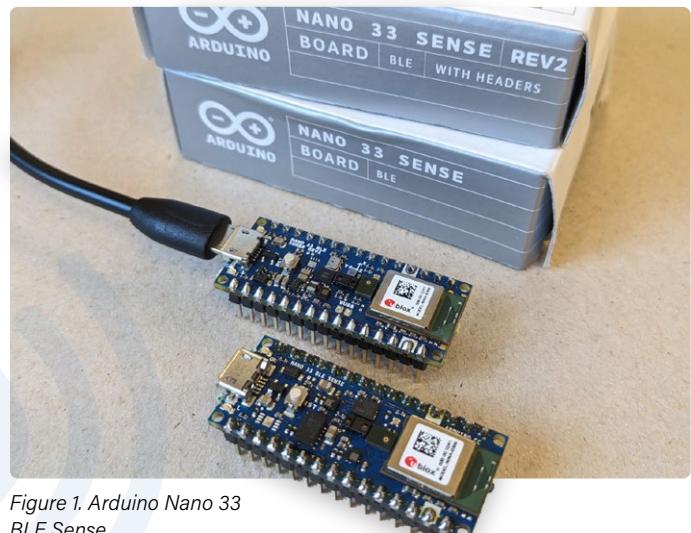


Figure 1. Arduino Nano 33 BLE Sense.

développé par Google. Cette version est conçue pour exécuter des modèles de ML sur des microcontrôleurs aux ressources limitées. Alors, pouvez-vous utiliser TFLite Micro sur votre carte Arduino ? Oui, mais pas sur tous les Arduino. Il est écrit en C++ 17 et nécessite une plateforme 32 bits, ainsi que quelques kilooctets de RAM. Il peut être utilisé avec de nombreux microcontrôleurs Arm Cortex-M, ainsi qu'avec l'ESP32. La liste complète des plateformes compatibles est disponible sur [1]. Ainsi, si le vénérable Arduino Uno n'est pas à la hauteur, l'Arduino Nano 33 BLE Sense (figure 1) peut être utilisé. Cette carte est en fait idéale pour expérimenter, car elle est puissante et contient de nombreux capteurs : un accéléromètre, un capteur d'humidité, de température, de couleur et d'intensité de la lumière, un capteur de pression et un microphone. Bien que cette carte Arduino soit puissante, elle ne l'est pas encore assez pour entraîner le modèle directement sur la carte. Dans la plupart des projets de ML basés sur des microcontrôleurs, la méthode habituelle consiste à préparer les données sources et à entraîner un modèle sur une machine puissante, telle que votre PC ou un serveur distant. Le modèle entraîné obtenu prend la forme d'un fichier binaire, qui doit être converti ultérieurement en un fichier d'en-tête en langage C. Enfin, un programme Arduino peut être écrit en utilisant les fonctions fournies dans la bibliothèque TFLite Micro et compilé avec l'IDE Arduino.

Figure 2. Les fichiers et dossiers avant l'exécution du script Python.

Pour ceux qui aiment tout faire par eux-mêmes, voyez la documentation officielle de TensorFlow Lite [2]. J'ai également trouvé des articles intéressants publiés par DigiKey [3]. Ils recommandent d'utiliser un ordinateur sous Linux doté de Python, puis d'installer, entre autres, TensorFlow, Keras, Anaconda, Jupyter Notebook et d'autres paquets. Une autre solution consiste à exécuter le code Python dans Google Colab [4], une plateforme gratuite basée sur le cloud qui permet aux utilisateurs d'écrire et d'exécuter du code Python dans un environnement en ligne. En tant que novice, j'ai trouvé la documentation de TensorFlow particulièrement difficile à suivre. Il faut également avoir une bonne compréhension des réseaux neuronaux pour pouvoir faire quoi que ce soit de fonctionnel, ce qui est vite décourageant.

Exemples simples

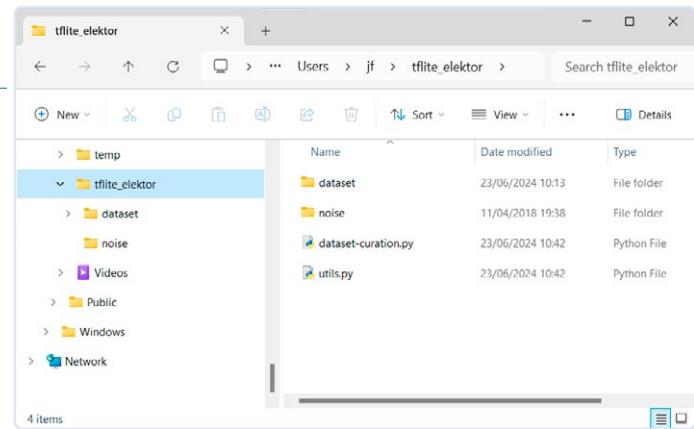
Les tutoriels sur Internet ont tendance à montrer toujours un peu les mêmes choses, dont certaines manquent d'utilité pratique pour être vraiment stimulantes. Par exemple, il est souvent montré comment entraîner un modèle à produire une valeur de sortie correspondant à une approximation du sinus de la valeur d'entrée. Cette méthode utilise, bien entendu, les valeurs précalculées de la fonction sinus comme ensemble de données d'apprentissage. Ainsi, une fois correctement entraîné, le modèle peut donner une valeur approximative de $\sin(x)$ à sa sortie, pour une valeur d'entrée x comprise entre 0 et 2π . Cela, bien sûr, sans utiliser de fonction sinus implémentée mathématiquement. Évidemment, il s'agit probablement de la manière la plus absurde et la moins pratique de calculer un sinus, en particulier sur un microcontrôleur où la puissance de calcul est limitée.

Un autre exemple, plus utile, est celui de la reconnaissance vocale. Le microcontrôleur peut ainsi écouter ce qui se passe dans son environnement à l'aide d'un microphone, discerner quelques mots (par exemple, *oui* et *non*, *ou chat* et *chien*, etc.) et déclencher diverses actions. Cet article, écrit par un débutant pour les débutants, reste simple et montre l'utilisation de la reconnaissance vocale sur un Arduino Nano 33 BLE Sense.

Reconnaissance vocale

À cet effet, le Google Speech Command Dataset sera utilisé. Il s'agit d'un dossier contenant 65 000 échantillons audio d'une seconde ; au total, une trentaine de mots différents prononcés par des milliers de personnes différentes. Pour entraîner le modèle, j'utiliserais Edge Impulse [5]. Il s'agit d'une plateforme qui permet de créer, d'entraîner et de déployer des modèles d'apprentissage automatique sur du matériel embarqué, tel que des microcontrôleurs, en mettant l'accent sur la facilité d'utilisation, sans nécessiter trop de programmation. Edge Impulse utilise TensorFlow Lite pour Microcontrôleurs et fournit un moyen facile de déployer le modèle et la bibliothèque TFLite sur la carte Arduino elle-même, ce qui est très pratique.

Pour commencer, vous aurez besoin de quelques échantillons audio. Créez un dossier, qui sera votre dossier de travail. J'ai appelé le mien *tflite_elektor*. Téléchargez le Google Speech Command Dataset [6]. Assurez-vous d'avoir une bonne connexion Internet, car le fichier pèse 2,3 Go. Comme il s'agit d'un fichier avec une extension *.tar.gz*,



il est compressé deux fois. Utilisez 7-Zip ou équivalent (je ne recommande pas l'utilitaire intégré de Windows pour traiter des fichiers aussi volumineux) pour obtenir le fichier *.tar* à l'intérieur, puis décompressez son contenu. Le résultat est un dossier *speech_commands_v0.02*. Placez ce dossier dans votre dossier de travail. Vous pouvez renommer le dossier *speech_commands_v0.02* pour lui donner un nom plus simple, dans mon cas : *dataset*.

Préparation des données

Ensuite, il faut préparer les données. Pour cela, je suggère d'utiliser l'excellent script Python développé par Shawn Hymel, qu'il propose généreusement sous licence libre. Téléchargez les fichiers *dataset-curation.py* et *utils.py* depuis son dépôt GitHub [7] et enregistrez-les dans votre dossier de travail. Ce script nécessite que le dossier *_background_noise* à l'intérieur du dossier *dataset* soit séparé des mots-clés. Glissez-déposez donc ce dossier en dehors du dossier *dataset* pour le placer dans votre dossier de travail. Vous pouvez également le renommer : *noise*. Votre dossier de travail contient maintenant les deux dossiers *dataset* et *noise* ainsi que les deux fichiers Python (figure 2).

Le script Python facilite grandement l'utilisation de l'énorme quantité de données contenue dans le dossier précédemment téléchargé. En outre, comme vous le verrez plus loin, le script est flexible et peut être utilisé avec d'autres données audio que celles-ci, y compris avec des fichiers audio que vous avez enregistrés vous-même. Il ne serait pas pratique de devoir télécharger plusieurs gigaoctets de fichiers vers les serveurs d'Edge Impulse. Pour commencer, choisissez un ou plusieurs mots-clés, qui seront les mots cibles que l'Arduino sera chargé de détecter. Pour cet exemple, j'ai choisi le mot *zéro*. Le script va créer un ensemble de dossiers : un dossier pour chaque mot-clé cible, donc dans ce cas un seul dossier nommé *zero*, ainsi qu'un dossier *_noise*, contenant du bruit aléatoire, et un dossier *_unknown* contenant des mots aléatoires autres que les mots-clés cibles.

Le script ajoute du bruit de fond dans les fichiers audio contenant les mots-clés, via un ré-échantillonnage. Il crée d'abord les dossiers nécessaires, puis extrait de petits échantillons de bruit de fond, qui sont ensuite mélangés avec des échantillons de mots-clés cibles et de mots-clés non cibles. Cela améliorera la résistance du modèle aux bruits de fond ; le résultat est un ensemble de données préparées, de taille beaucoup plus réduite (environ 140 mégaoctets), qu'Edge Impulse peut facilement importer.

Utilisation de Python

Le code a été testé avec Python 3.7. Pour gérer plusieurs environnements Python de différentes versions, avec différents paquets installés, il est possible d'utiliser Anaconda [8], qui facilite la création

```
(jf) C:\Users\jf\tflite_elektor>python dataset-curation.py -t "zero" -n 1500 -w 1.0 -g 0.1 -s 1.0 -r 16000 -e PCM_16 -b "./noise" -o "./keywords_curated" "./dataset"
Keyword Dataset Curation Tool
v0.1
Gathering random background noise snippets (1500 files)
Progress: |██████████| 100.0% Complete
Mixing: zero (1500 files)
Progress: |██████████| 100.0% Complete
Mixing: _unknown (1500 files)
Progress: |██████████| 100.0% Complete
Done!
```

Figure 3. Exécution du script Python.

d'une installation « toute neuve » de la version souhaitée. Ici, je crée un nouvel environnement appelé *jf* :

`conda create -n jf python=3.7`

Ensuite, vous devez installer les paquets *librosa*, *numpy* et *soundfile* :

`python -m pip install librosa numpy soundfile`

Le paquet *shutil* est également nécessaire, mais il est normalement inclus avec Python 3.7.

Depuis la ligne de commande Anaconda ou celle de votre système, naviguez jusqu'à votre répertoire de travail et exécutez le script à l'aide de la commande :

```
python dataset-curation.py -t "zero" -n 1500 -w 1.0 -g 0.1 -s 1.0 -r 16000 -e PCM_16 -b "./noise" -o "./keywords_curated" "./dataset"
```

Et attendez quelques minutes que l'opération se termine (**figure 3**).

Jetons un rapide coup d'œil sur les arguments pris par le script :

`-t` sert à lister les mots-clés cibles. Ici, j'utilisera `-t "zero"`.

`-n` est le nombre d'échantillons de sortie par catégorie. 1500 est un bon point de départ.

`-w` et `-g` sont les volumes sonores du mot parlé et du bruit de fond, respectivement. `-w 1.0 -g 0.1` sont des valeurs recommandées. `-s` et `-r` sont la longueur de l'échantillon (1 s) et le taux de rééchantillonnage (16 kHz). Utilisez `-s 1.0 -r 16000`.

`-e` est le nombre de bits pour l'échantillonnage, ici le PCM 16 bits est utilisé.

`-b` est l'emplacement du dossier du bruit de fond, `-o` est celui du dossier de sortie et enfin, le dernier argument non étiqueté est la liste des dossiers d'entrée. Ici, il s'agit du dossier *dataset*.

Lorsque le script a fini son traitement, il devrait avoir créé un dossier *keywords_curated* contenant trois dossiers : *_noise*, *_unknown* et *zero*. (**figure 4**)

Importation dans Edge Impulse

L'étape suivante consiste à importer ces fichiers dans Edge Impulse. Rendez-vous sur leur site web et créez un compte si vous n'en avez pas déjà un. Après avoir ouvert une session, créez un nouveau projet. Dans le menu de gauche, naviguez vers *Data Acquisition*, puis cliquez sur *Add Data* et *Upload Data*. Cochez *Select a folder* et choisissez le premier dossier, par exemple *_noise*.

Veillez à cocher l'option *Automatically split between training and testing*. De cette façon, Edge Impulse utilisera d'abord 80 % des échantillons téléchargés pour entraîner le modèle. Ensuite, nous pourrons tester les performances du modèle entraîné en utilisant des données qu'il n'a jamais vues auparavant ; les 20 % restants sont réservés à cet effet.

Cochez également l'option *Label: infer from filename* afin que Edge Impulse reconnaîsse, par le nom du fichier, les échantillons contenant le(s) mot(s) à reconnaître ainsi que ceux contenant du bruit. Enfin, cliquez sur le bouton *Upload data* dans le coin inférieur droit et attendez que le transfert soit terminé. Répétez l'opération pour les deux dossiers restants *_unknown* et *zero*.

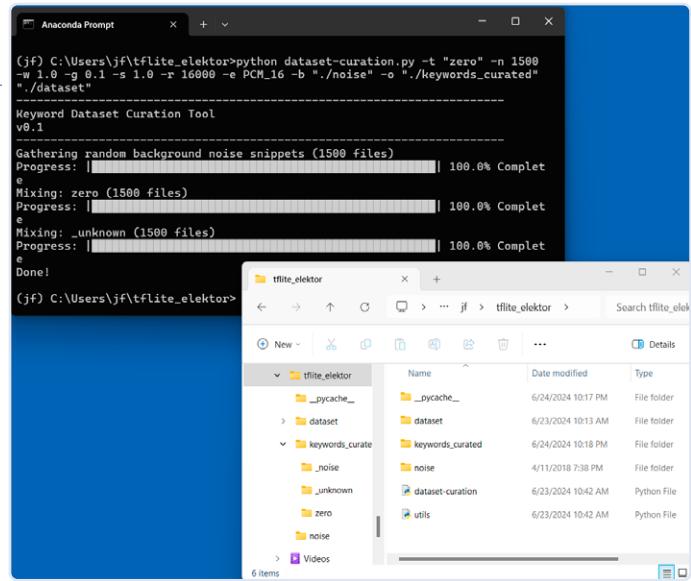


Figure 4. Après l'exécution du script Python.

Une fois le transfert terminé, retournez dans *Data Acquisition* pour visualiser tous les échantillons téléchargés. Assurez-vous que les fichiers sont bien répartis en 80 % de données d'entraînement et 20 % de données de test, et que les étiquettes ont été correctement lues (**figure 5**).

Ensuite, il est nécessaire d'ajouter un *Processing Block*. Dans Edge Impulse, il s'agit d'un composant utilisé pour transformer les données brutes en un format adapté à l'entraînement et à l'inférence des modèles d'apprentissage automatique. Sous ce simple « bloc » se cachent plusieurs étapes plus complexes, comme le prétraitement des données d'entrée, l'extraction des caractéristiques (voir ci-dessous), des étapes optionnelles telles que des transformées de Fourier, etc.

Dans le contexte de la ML, les *caractéristiques* ou *features* sont des propriétés distinctes et quantifiables des données observées. Ici, les caractéristiques à extraire sont les *Mel-Frequency Cepstral Coefficients* (MFCC) [9], qui sont couramment utilisés dans le traitement des signaux audio et la reconnaissance vocale. Ils représentent le spectre de puissance à court terme d'un signal sonore sur une échelle non linéaire de fréquence.

Ainsi, dans le menu de gauche cliquez sur *Impulse Design* puis sur le bouton *Add a Processing Block*. Sélectionnez la première option, *Audio (MFCC)*, en cliquant sur *Add* à droite. Ensuite, cliquez sur le

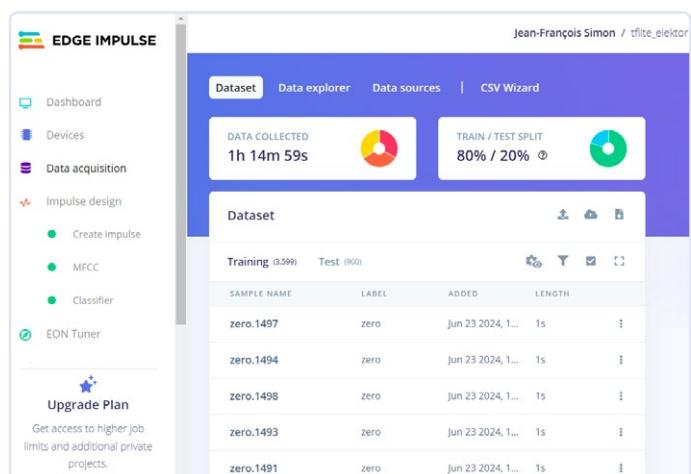


Figure 5. Les échantillons audio sont correctement stockés par Edge Impulse.

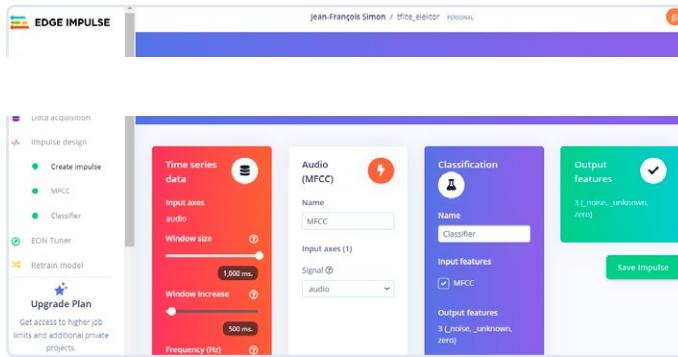


Figure 6. Configuration du modèle.

bouton *Add a Learning Block* (Ajouter un bloc d'apprentissage) et choisissez la première option, *Classification*, qui est celle recommandée. Enfin, cliquez sur *Save Impulse* à droite (**figure 6**).

Entraînement du modèle

Dans le menu de gauche, sous *Impulse Design*, sélectionnez *MFCC*. Naviguez jusqu'à l'onglet *Generate Features* et cliquez sur *Generate Features* (**figure 7**). Attendez que la génération des caractéristiques soit terminée. Ceci fait, allez dans la section *Classifier*, située juste en dessous de *MFCC* dans le menu de gauche. Dans le coin supérieur droit, cliquez sur *target* et sélectionnez *Arduino Nano 33 BLE Sense*. Vous pouvez ajuster les paramètres du réseau de neurones, mais les paramètres par défaut sont, sans surprise, meilleurs que tout ce que j'aurais pu faire moi-même. Notez que vous pouvez éditer le réseau neuronal à l'aide de l'outil graphique ou passer en mode expert via le menu contextuel si vous êtes familier avec Keras. Pour cet exemple, cliquez simplement sur *Start Training* en bas de la page pour commencer à entraîner le modèle. Une fois l'entraînement terminé, examinez les résultats dans le cadre *Model* en bas à droite. Vous verrez un score général de précision, et 90 % est considéré comme un très bon score ; ici, le système a obtenu 92,8 % (**figure 8**).

Il existe également une matrice, appelée *matrice de confusion*, qui vérifie les performances du modèle. Les lignes représentent les étiquettes réelles et les colonnes les étiquettes prédictives. Les nombres situés sur la diagonale, où l'étiquette prédictive correspond à l'étiquette réelle, devraient être beaucoup plus élevés que les autres valeurs. Ici, la diagonale affiche 98,8 %, 87 % et 92,8 %, ce qui devrait être suffisant. Un test plus difficile consiste à évaluer le modèle en lui fournissant des données qu'il n'a jamais vues auparavant. Pour ce faire, accédez à la section *Model testing* dans le menu de gauche. Cliquez sur *Classify All* et laissez l'opération se dérouler. Dans le cadre *Results*, en bas, le score est inférieur de quelques pourcents au score précédent, mais c'est normal. Ici, on obtient 90,56 % (**figure 9**).

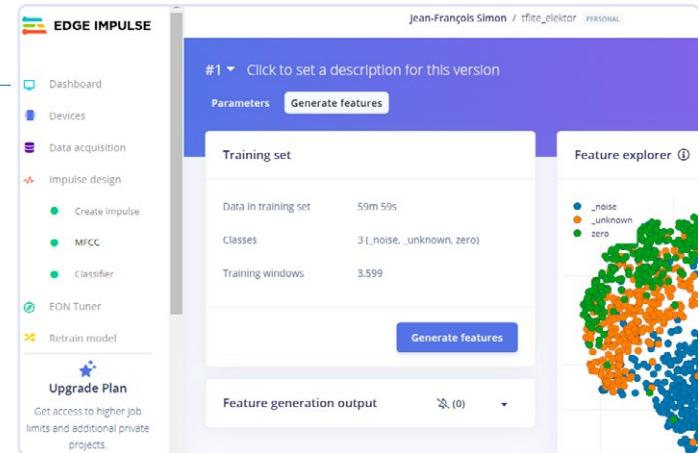


Figure 7. La section *Generate Features*, où les données audio sont traitées.

Déploiement pour Arduino

Passons maintenant à la page *Deployment*. Edge Impulse offre plusieurs options pour empaqueter le modèle : une bibliothèque C++ générique pour un usage général sur microcontrôleurs, Cube MX pour les composants STM32, WebAssembly pour les environnements JavaScript et bien d'autres encore. Cliquez sur *Search deployment options* et sélectionnez *Arduino library*. Cliquez ensuite sur le bouton *Build* en bas de la page. Quelques secondes plus tard, votre navigateur téléchargera un fichier ZIP contenant la bibliothèque Arduino.

J'utilise l'IDE Arduino version 1.8.19. Ouvrez l'Arduino IDE et connectez votre Arduino Nano 33 BLE Sense à votre ordinateur. Si c'est la première fois que vous utilisez votre Nano 33 BLE, l'IDE vous proposera de télécharger un paquet appelé *Arduino Mbed OS Nano Boards package*, qui est effectivement nécessaire. Ensuite, vous pouvez ajouter la bibliothèque en utilisant la technique habituelle, en cliquant sur *Croquis*, *Inclure une bibliothèque*, *Ajouter la bibliothèque ZIP* et en sélectionnant le fichier .zip que vous venez de télécharger depuis Edge Impulse. Ensuite, allez dans *Fichier*, *Exemples* et localisez la bibliothèque que vous venez d'installer. Il se peut que vous deviez recharger l'IDE Arduino pour qu'elle apparaisse. Le nom doit correspondre au nom de votre projet Edge Impulse, c'est donc *tflite_elektor_inferencing* dans mon cas.

Notez qu'il y a deux dossiers séparés, *nano_ble33_sense* et *nano_ble33_sense_rev2* (**figure 10**). L'exemple *microphone_continuous* utilisé ici n'apparaît que dans le premier, mais je l'ai testé avec succès sur les deux révisions de carte. En revanche, vous devrez probablement choisir la bonne version en fonction de la carte dont vous disposez si vous voulez expérimenter avec les autres exemples de programmes qui utilisent l'accéléromètre intégré. Ouvrez l'exemple *microphone_continuous*.

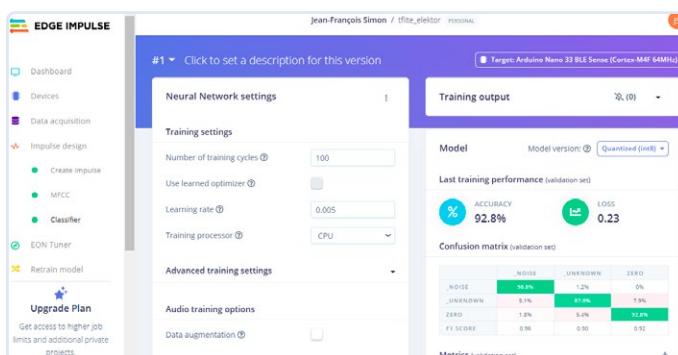


Figure 8. Le modèle a terminé son entraînement.

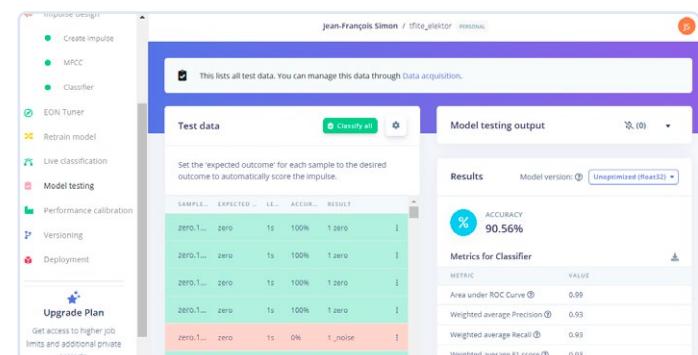


Figure 9. Test du modèle sur de nouvelles données.

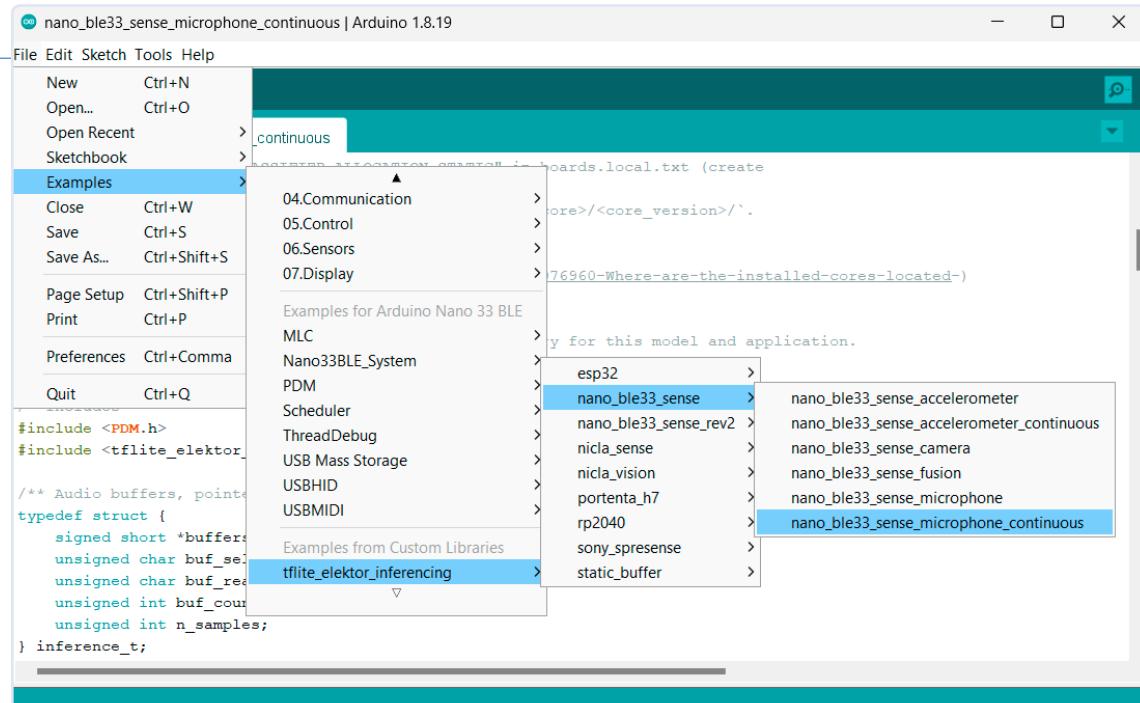


Figure 10. Utilisation de l'exemple intégré microphone_continuous.

Si vous le souhaitez, vous pouvez explorer le programme exemple pour comprendre comment tout est mis en place et quelles fonctions sont appelées pour l'inférence. Dans la boucle, le microcontrôleur attend que la mémoire tampon du microphone se remplisse, puis appelle la fonction `run_classifier_continuous` pour effectuer l'inférence. Les résultats ne sont affichés sur le Moniteur Série qu'une fois par seconde. Le code de la bibliothèque fournie n'est parfois pas facile à suivre, mais essayer de voir ce qu'il y a sous le capot peut être un exercice enrichissant.

Programmation de la carte

Dans le menu Outils de l'IDE Arduino, assurez-vous que la bonne carte (Nano 33 BLE Sense) est sélectionnée, ainsi que le bon port COM, que vous pouvez vérifier à l'aide du Gestionnaire de périphériques si vous utilisez Windows. Cliquez sur le bouton *Téléverser* et attendez ! Gardez à l'esprit que la compilation du projet prend un certain temps car la bibliothèque, qui contient les fonctions TensorFlow Lite pour l'inférence ainsi que le modèle créé sur Edge Impulse au format binaire, est assez conséquente. Une fois

Un sujet complexe, et des options plus simples

Pourquoi ne voit-on pas une plus grande variété de projets et d'exemples utilisant la ML sur de petits microcontrôleurs ? À mon avis, il y a plusieurs raisons.

- Pour entraîner un modèle, vous avez besoin d'un grand volume de données pré-classifiées. Pour la fonction `sinus`, c'est facile : les résultats attendus sont connus à l'avance. Pour les mots-clés anglais, vous pouvez utiliser le jeu de données gratuit Google Speech Command Dataset. Mais pour faire autre chose que de la reconnaissance de mots, il faudra acquérir une grande quantité de données adaptées à votre projet, ce qui peut constituer une difficulté majeure.
- L'apprentissage automatique est un domaine complexe à aborder. Pour sortir des sentiers battus, il faut être très à l'aise avec Python et Linux et avoir une bonne compréhension de la théorie de la ML et des réseaux neuronaux. Personnellement, j'ai trouvé cela très difficile d'accès, et je n'ai clairement pas les qualifications ni l'expérience requises pour aller plus loin que cet exemple de base.
- La documentation existante pour TensorFlow et TensorFlow Lite s'adresse principalement aux développeurs, et il semblerait que nombre d'entre eux soient davantage

intéressés par les plateformes plus puissantes, telles que le Raspberry Pi 5 équipé d'un accélérateur d'IA externe, ou le Jetson Nano de Nvidia. Cela dit, Pete Warden, un contributeur clé de TensorFlow et maintenant PDG de Useful Sensors, a récemment fait une mise à jour majeure de la version TFLite Micro pour le Raspberry Pi Pico, qui a été créée en 2021 mais assez peu utilisée depuis. Espérons que cela revitalise le projet et incite les gens à l'utiliser !

Si vous souhaitez simplement ajouter quelques fonctions de contrôle vocal à votre projet, sans utiliser les processus complexes décrits dans cet article, vous pouvez envisager trois autres options : le Voice Recognition Module V3d d'Elechouse, le Speech Recognizer de Grove et le Offline Language Learning Voice Recognition Sensor de DFRobot. Vendus respectivement 30, 20 et 17 euros, ces modules sont un peu plus chers que ce à quoi on s'attendrait, mais ils peuvent constituer une solution simple à mettre en œuvre. Ces modules n'utilisent pas le même principe que celui décrit dans cet article ; vous pouvez les entraîner à reconnaître votre propre voix en répétant les commandes un petit nombre de fois. Je ne les ai jamais utilisés moi-même, mais ils semblent assez efficaces, et vous trouverez de nombreuses vidéos de démonstration sur YouTube.



```
Predictions <DSP: 76 ms., Classification: 6 ms., Anomaly: 0 ms. >:
noise: 0.15234
unknown: 0.84766
zero: 0.00000
Predictions <DSP: 76 ms., Classification: 6 ms., Anomaly: 0 ms. >:
noise: 0.98438
unknown: 0.01172
zero: 0.00391
```

Figure 11. L'Arduino écoute les bruits aléatoires dans la pièce.

```
Predictions <DSP: 76 ms., Classification: 6 ms., Anomaly: 0 ms. >:
noise: 0.00000
unknown: 0.00391
zero: 0.99609
```

Figure 12. Le mot a été détecté avec une excellente certitude.

le programme compilé, vous verrez qu'il utilise environ 171 kilooctets de mémoire Flash et environ 47 kilooctets de RAM pour les variables globales.

Ça marche !

Ouvrez maintenant le Moniteur Série pour observer la sortie. Chaque seconde, il donne trois nombres qui sont les probabilités qu'un motif donné ait été détecté, parmi : un bruit aléatoire, un mot qui n'est pas zéro et enfin le mot zéro. La **figure 11** est un exemple où rien de spécial ne se produit. Si je prononce le mot « zéro » relativement près de la carte Arduino, le troisième score atteint une valeur très élevée, presque 100 % (**figure 12**).

C'est un bon début ! L'étape suivante consisterait à faire en sorte que la carte Arduino fasse quelque chose d'utile avec cette information. Je suis sûr que vous trouverez des applications intéressantes pour envoyer des commandes vocales à des gadgets utilisant un Arduino. Le processus décrit ci-dessus et le script Python conçu par Shawn Hymel peuvent également être utilisés pour détecter plus d'un mot. Le nombre maximum sera limité par l'espace de stockage dans la mémoire Flash et la puissance de calcul disponibles sur l'Arduino. Dans le code, la ligne `#define EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW 4` nous indique que chaque fenêtre d'une seconde est divisée en quatre tranches de 250 ms, et la sortie dans le Moniteur Série nous indique que le temps utilisé par le programme est de 76 + 6 = 82 ms par tranche de 250 ms, ce qui correspond approximativement à 33% d'utilisation du CPU. Il reste donc du temps de traitement disponible pour ajouter votre propre programme.

Pour aller plus loin

Par souci de simplicité, j'ai utilisé l'un des mots déjà disponibles dans le Google Speech Command Dataset. Pour entraîner un modèle à repérer un mot qui ne fait pas partie de cet ensemble, vous devez enregistrer un grand nombre d'échantillons audio avec

le mot prononcé, de préférence par un grand nombre de personnes avec des voix, des âges et des intonations différents. Alors que le dossier Google contient des milliers d'échantillons par mot, lorsque vous enregistrez vous-même des mots personnalisés, cinquante à cent échantillons peuvent constituer un bon début. Bien entendu, je n'ai fait qu'effleurer le sujet avec cet exemple simple. Je recommande aux personnes intéressées de l'explorer plus en profondeur ! Avez-vous une idée en tête pour un projet utilisant la ML ? 

240357-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (jean-francois.simon@elektor.com), ou contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Jean-François Simon a une passion de longue date pour l'électronique et s'intéresse à des sujets aussi variés que la conception de circuits, le test et la mesure, le prototypage, la radio logicielle (SDR), et plus encore. Il aime créer, modifier et améliorer ses outils et autres systèmes. Il a une formation d'ingénieur et aime aussi la mécanique, l'usinage et tout ce qui est technique. Jean-François a rejoint l'équipe d'Elektor en 2023.



Produits

› **Arduino Nano 33 BLE Sense Rev2 avec connecteurs**
www.elektor.fr/20404

LIENS

- [1] TensorFlow Lite pour microcontrôleurs : <https://www.tensorflow.org/lite/microcontrollers>
- [2] TFLite Micro sur GitHub : <https://github.com/tensorflow/tflite-micro/tree/main/tensorflow/lite/micro/examples>
- [3] Digikey, TensorFlow Lite Tutorial Part 1 : Wake Word Feature Extraction : <https://tinyurl.com/bdf4dkfm>
- [4] Google Colab : <https://colab.research.google.com/>
- [5] Edge Impulse : <https://edgeimpulse.com/>
- [6] Google Speech Command Dataset : http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz
- [7] Scripts Python de Shawn Hymel : <https://github.com/ShawnHymel/ei-keyword-spotting>
- [8] Anaconda : <https://www.anaconda.com/download>
- [9] MFCCs sur Wikipedia : https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

un concentrateur pour les appareils RS-422 et RS-485

câbler votre bus comme une star

Sébastien Guerreiro de Brito (France)

Les bus RS-485 ou RS-422 sont encore largement utilisés dans l'industrie pour le contrôle d'accès, la vidéosurveillance et de nombreuses autres applications. Toutefois, la norme exige un câblage en guirlande, ce qui peut s'avérer contraignant. Voici comment s'adapter au câblage en étoile.

Le bus RS-422 [1] est défini par la norme EIA-422-B-1994 comme un bus *simplex* (c'est-à-dire qu'il n'y a qu'un seul émetteur à la fois) qui peut piloter jusqu'à 10 unités de charge (UL). L'acheminement des signaux est différentiel pour assurer une transmission robuste des données sur de longues distances, même dans des environnements bruyants. La norme spécifie notamment les caractéristiques électriques de la ligne de transmission différentielle, en utilisant une paire torsadée.

La ligne de transmission étant unidirectionnelle, si l'on souhaite seulement commander des récepteurs distants sans recevoir de retour d'information de leur part, une seule paire de fils suffit. Cependant, le cas le plus courant est celui où les appareils connectés communiquent dans les deux sens, par exemple lorsqu'un contrôleur central s'adresse à un capteur distant pour lui demander une mesure, à la suite de quoi le capteur transmet sa mesure. Dans ce cas, deux paires de fils sont nécessaires : la première (TX+/TX-) pour les transmissions de l'appareil maître vers les appareils esclaves, et la seconde (RX+/RX-) pour les transmissions des appareils esclaves vers l'appareil maître. Comme plusieurs esclaves doivent transmettre sur la même paire de fils, ils doivent activer et désactiver leurs émetteurs de manière à ce qu'un seul émetteur soit actif à la fois.

Débit en bauds face à la portée

L'émetteur est capable de transmettre des données sur une distance d'environ 1200 m, mais la portée augmente au détriment du débit de données. La norme spécifie des débits allant jusqu'à 10 Mbps, mais uniquement sur de courtes distances, comme le montre la **figure 1**. En raison de la différence

d'impédance entre le câble et le récepteur, une réflexion de l'onde transmise peut se produire sur les lignes lors de la transmission sur de longues distances ou à des débits de données élevés. Pour pallier ce phénomène, une résistance de terminaison est nécessaire pour adapter la charge à la résistance caractéristique du câble (typiquement 120 Ω). D'autres méthodes de terminaison existent, mais comme elles sont moins courantes, je ne les présenterai pas ici.

De RS-422 à RS-485

Le bus RS-485 [2] est basé sur le bus RS-422, avec quelques caractéristiques supplémen-

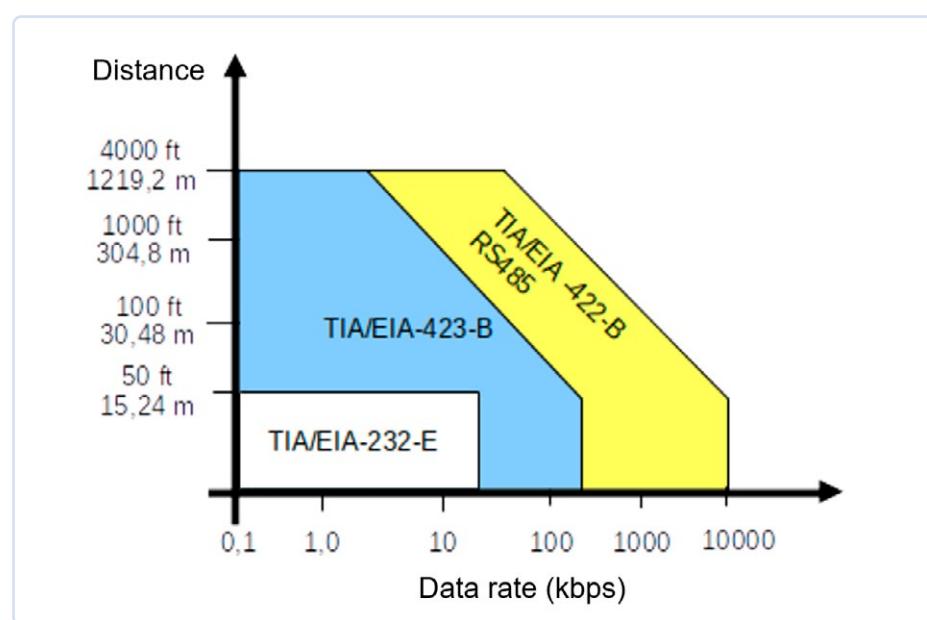


Figure 1. Débit de données en fonction de la longueur du conducteur.

taires, telles que la possibilité d'augmenter le nombre d'UL jusqu'à un maximum de 32. Le bus RS-485 peut prendre en charge une configuration multipoint sur une seule paire de fils (souvent appelée Data+/Data- ou A/B), pour la communication entre un émetteur et plusieurs récepteurs (**figure 2**). Cette configuration, dite half-duplex, est de loin la plus répandue. Contrairement à une idée reçue, le bus RS-485 nécessite également un fil de terre pour la communication, soit un total de trois fils. Plus rarement, les connexions RS-485 full-duplex sont réalisées sur deux paires torsadées, comme pour le RS-422. Je n'entrerai pas dans les détails ici. Pour plus d'informations, deux documents de Texas Instruments, le *RS-485 Design Guide* [3] et le *RS-422 and RS-485 Standards Overview and System Configurations* [4], ainsi que le *RS-485/RS-422 Circuit Implementation Guide d'Analog Devices* [5] sont très utiles.

En mode semi-duplex, chaque appareil transmet et reçoit des données sur la même paire de fils, ce qui signifie que chaque appareil doit activer son émetteur uniquement lorsqu'il doit transmettre. Cela implique que les récepteurs doivent être en mesure de reconnaître que la trame transmise leur est destinée. Un système d'adressage est donc nécessaire. Les commandes de *Broadcast* peuvent également être envoyées en envoyant une trame à un récepteur virtuel avec une adresse réservée (par exemple, 255). De cette manière, tous les récepteurs prennent en compte la trame reçue.

En pratique, dans ce type de communication, le maître émet et les esclaves (récepteurs) écoutent. Lorsque le maître a demandé une information spécifique à l'un des récepteurs, ce dispositif spécifique répond. Si le maître a envoyé des informations *Broadcast* à tous les récepteurs, ces derniers ne doivent pas répondre.

Les pilotes RS485 ou RS422 fonctionnent tous sur des principes similaires. À titre d'exemple, le schéma du circuit intégré MAX489, utilisé dans ce projet, est présenté dans la **figure 3**. Sa fiche technique est disponible sur le site [6]. Les données à transmettre sur le bus sont connectées à la broche *Driver Input* (DI). Les données reçues du bus sont alors disponibles sur la broche de sortie du récepteur (RO). Les broches *Receiver Enable* et *Driver Enable* (/RE et DE, respectivement) sont utilisées pour contrôler le sens de la communica-

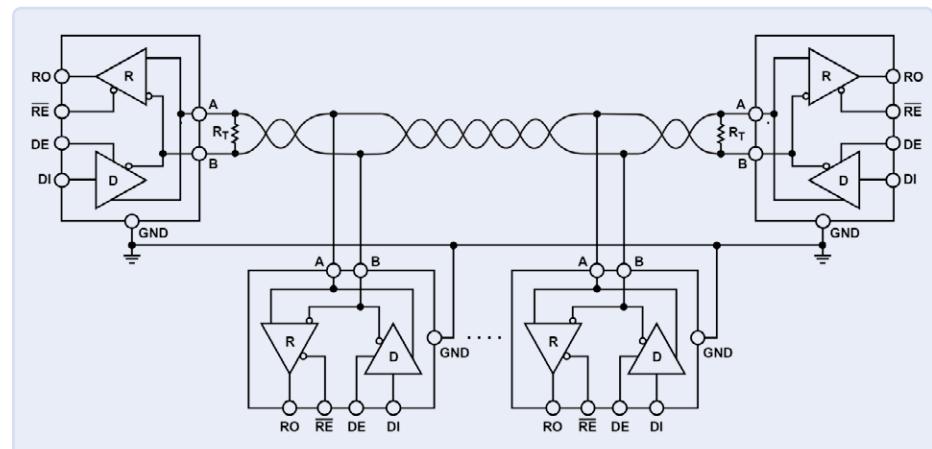


Figure 2. Configuration typique d'une liaison RS-485 semi-duplex. Source : Analog Devices

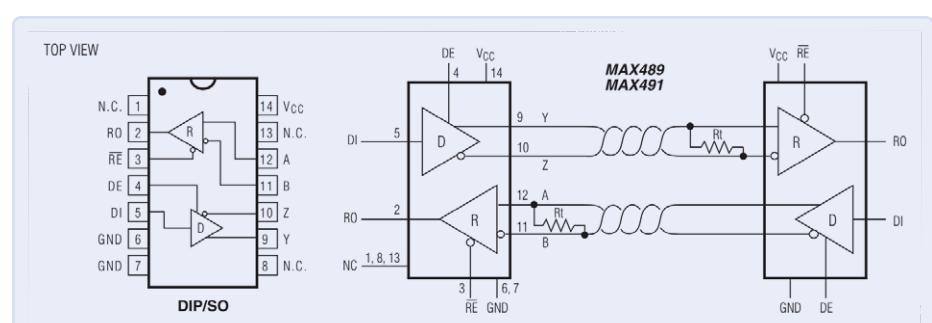


Figure 3. Configuration des broches du MAX489 et circuit de fonctionnement typique. Source : Analog Devices [6]

tion (réception ou transmission). Dans le cas d'une communication semi-duplex, ces deux broches peuvent être contrôlées par le même signal. Dans le cas d'une communication full-duplex, elles peuvent être commandées séparément. Les signaux de sortie Y (TX+), Z (TX-), B (RX-), A (RX+) sont connectés au bus de communication. Dans le cas de connexions semi-duplex, typiquement dans les installations RS-485, il est habituel de connecter Z à B (Data-) et A à Y (Data+).

Charges unitaires

La norme pour le RS-485 spécifie que pour un émetteur donné, un maximum de 32 émetteurs/récepteurs peuvent être connectés, chacun représentant une charge unitaire. Cette notion est liée à la résistance d'entrée des récepteurs (considérée comme 12 kΩ par la norme). Comme certains dispositifs RS-485 sont maintenant disponibles avec des résistances d'entrée huit fois supérieures, ne représentant donc que 1/8 UL, il est théoriquement possible de connecter 8 × 32, soit 256 récepteurs, sur le même bus.

Acheminement des câbles

Les configurations possibles et interdites sont présentées dans la **figure 4**. Les réseaux en étoile (a, c) sont interdits, de même que les connexions tronquées ou les configurations en anneau (b). Les réseaux en chaîne avec

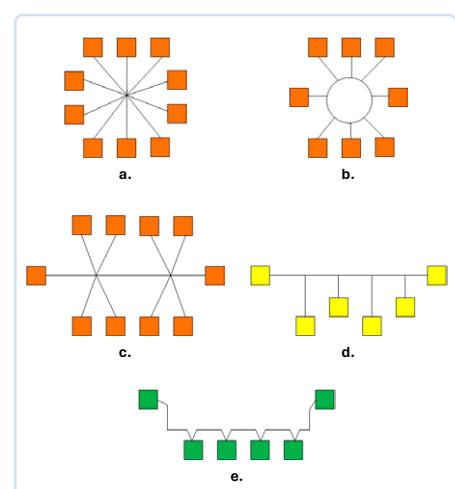


Figure 4. Configurations du réseau, dont certaines sont interdites.

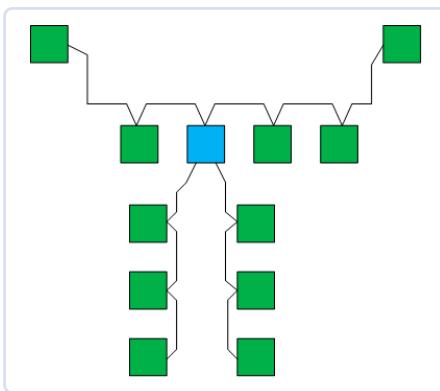


Figure 5. Exemple de réseau avec notre concentrateur RS-485/RS-422.



Liste des composants

Résistances (0805, 0,1 W)

R1, R2, R7...R9, R31, R37...R40 = 4,7 kΩ, 1%
 R3, R41...R45 = 1 kΩ, 1%
 R4, R10 = 22 kΩ, 1%
 R5, R6, R13, R20, R25, R30, R32 = 120 Ω, 1%
 R11, R12, R14...R19, R21...R24, R26...R29,
 R33...R36 = 47 Ω, 1%
 R46...R55 = 560 Ω, 1%

Condensateurs

C1 = 47 µF 35V
 C2 = 220 µF 10V
 C3...C5, C8, C12...C14, C24, C30 = 100 nF,
 0805
 C6, C7 = 1 nF, 0805
 C20 = 10 nF, 0805

Inducteurs

L1 = 100 µH

Semi-conducteurs

D1, D2, D4...D6, D28 = PML4148
 D3 = 1N4004
 D7 = MBRS130LT3
 D8...D27 = PESD5V2S2UT
 IC1, IC2 = 74HC14D
 IC3, IC6...IC9 = MAX489CSD
 LED1, LED3...LED8 = 5 mm LED vert
 LED2 = 5 mm LED orange
 U1 = LM2674-5

Divers

JP1...JP5 = En-tête mâle à 2 broches, pas de 2,54 mm
 X1, X7 = connecteur à 2 broches, pas de 5,08 mm, MSTBVA 2,5/2-G-5,08
 X2...X6 = connecteur à 5 broches, pas de 3,81 mm

des stubs (d) sont déconseillés. En résumé, les réseaux doivent être câblés en guirlande (e). Le bus doit passer par chaque appareil aussi près que possible de son point de connexion, puis se connecter au suivant et ainsi de suite jusqu'à une résistance de terminaison.

Principe de fonctionnement

Pour permettre le câblage en étoile, j'ai développé une carte qui divise le réseau en plusieurs bus. Le principe de fonctionnement de ce circuit est illustré par la **figure 5**. La boîte bleue représente notre carte qui permet d'avoir un réseau en étoile à l'entrée (en haut de la figure) et plusieurs réseaux en étoile à la sortie (en bas). La carte est relativement simple et ne nécessite pas de microcontrôleur ou de FPGA pour fonctionner. Son fonctionnement est illustré dans la **figure 6**, selon le principe suivant.

Le bloc à gauche de la figure représente le côté entrée, tandis que les quatre blocs à droite sont les sorties 1, 2, 3, 4, chacune sur son pilote RS-485 dédié. Comme le montre le schéma fonctionnel, l'émetteur (TX, en violet) du côté de l'entrée est connecté à tous les récepteurs (RX) sur les sorties 1 à 4, afin de transmettre des informations à tous les

récepteurs, comme on peut s'y attendre. De même, toutes les sorties des émetteurs sur les bus 1 à 4 (en vert) sont reliées au récepteur d'entrée (RX). Pour éviter les collisions, des blocs logiques sont ajoutés. Lorsqu'une activité est détectée du côté du TX d'entrée, tous les TX de sortie sont désactivés, et vice versa.

Schémas détaillés

Le schéma détaillé est présenté dans la **figure 7**. Comme mentionné dans le paragraphe précédent, le principe général est de transmettre le signal reçu ; ainsi, tous les signaux entrants (RX) en provenance des drivers sont assemblés par les diodes D4, D5, D6, D28 et envoyés au driver émetteur. En ce qui concerne les données sortantes (TX), le signal provenant du pilote d'entrée est simplement remodelé par les portes de déclenchement de Schmitt IC1A et IC1B, puis envoyé aux pilotes de sortie.

Les diodes D8...D27 sont des diodes de protection doubles unidirectionnelles qui protègent les pilotes contre les décharges électrostatiques (ESD) sur la ligne de bus. Même si les entrées et sorties du MAX489 sont assez robustes, une résistance en série

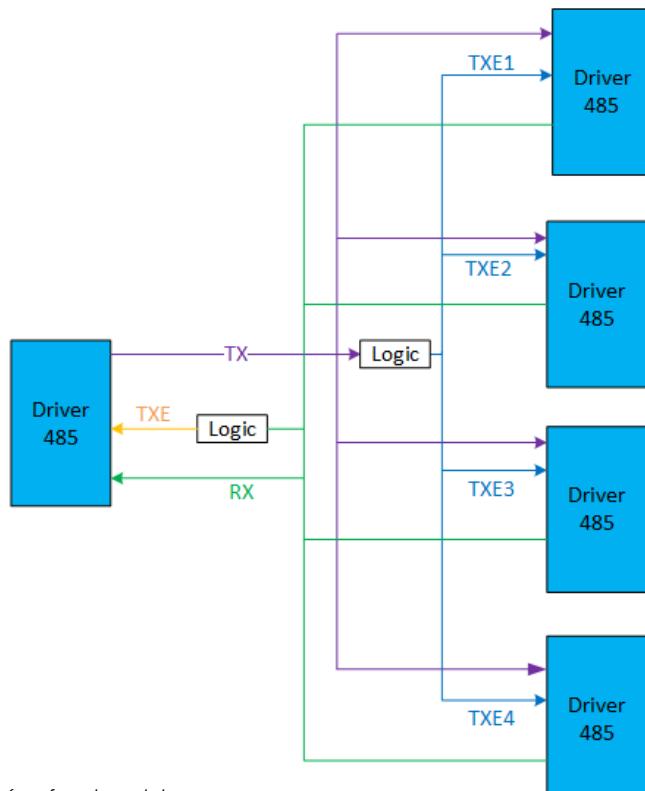


Figure 6. Schéma fonctionnel du concentrateur.

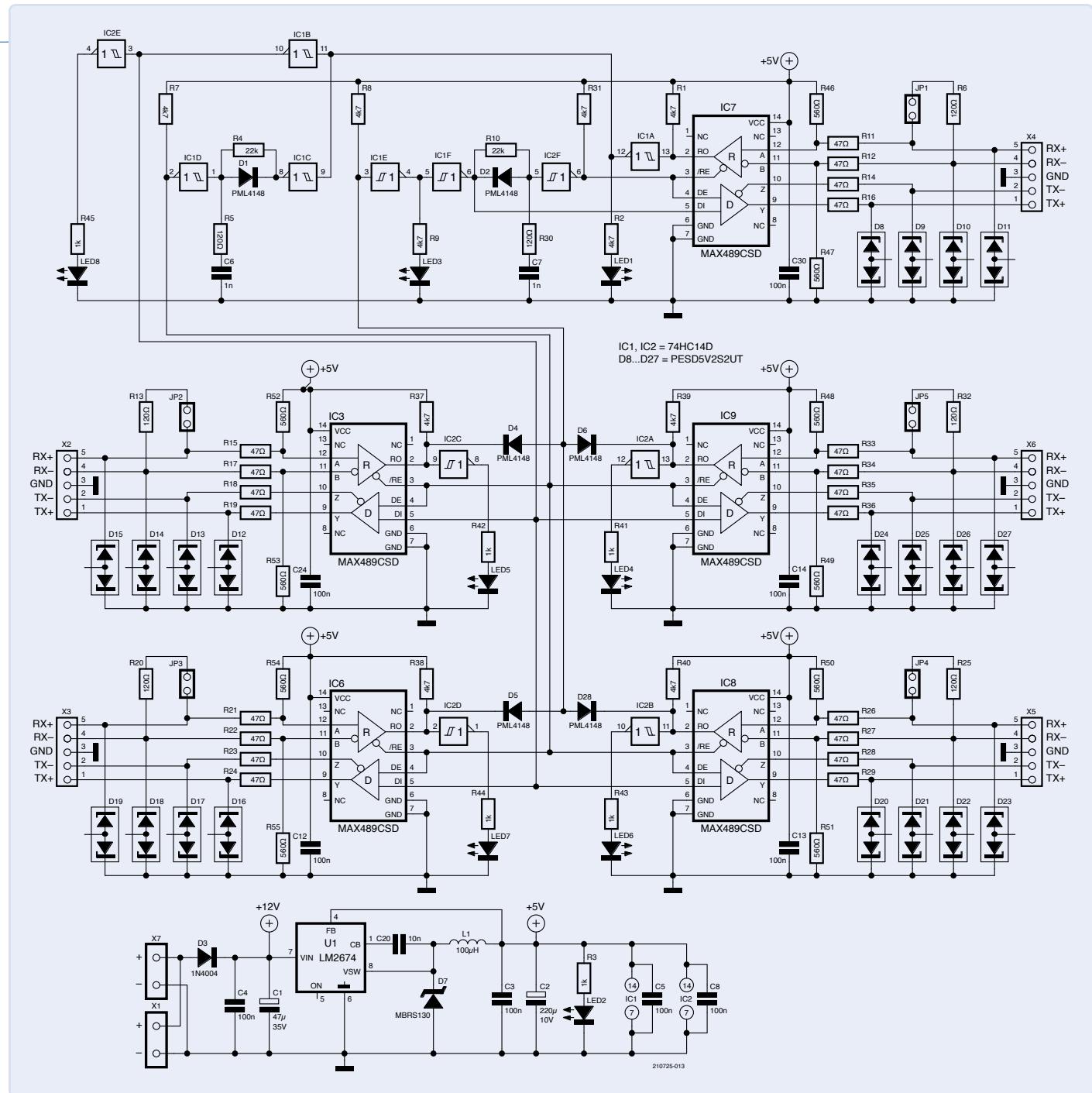


Figure 7. Schémas complets.

de 47Ω est ajoutée à chaque ligne, pour une protection supplémentaire contre les transistors venant de l'extérieur. Elles sont au nombre de quatre par circuit intégré, pour un total de vingt : R11, R12, R14...R19, R21...R24, R26...R29, R33...R36. Les cavaliers JP2, JP3, JP4 et JP5 sont utilisés pour connecter les résistances de terminaison embarquées (R6, R13, R20, R25, R32) au bus, si nécessaire. La carte dispose également de plusieurs LED : La LED2 indique que la carte est sous tension. Les autres LED indiquent l'activité des pilotes, soit en transmission, soit en réception

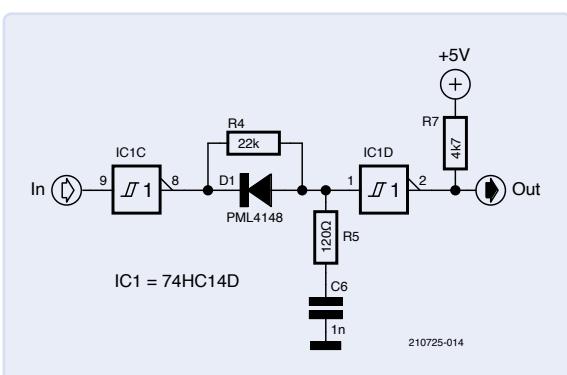


Figure 8. Circuit de détection/transmission.



Figure 9. Simulation du circuit de détection.

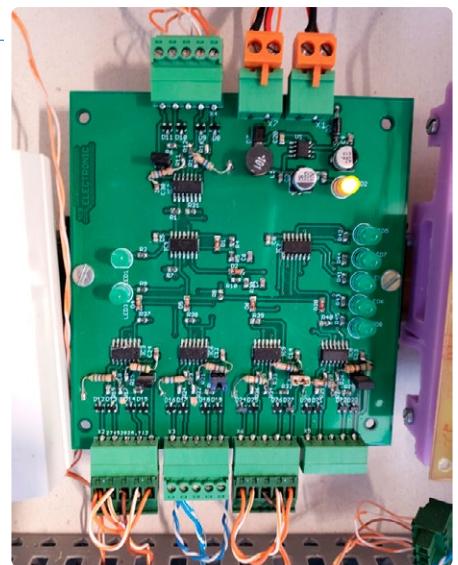


Figure 10. Photo de mon prototype.

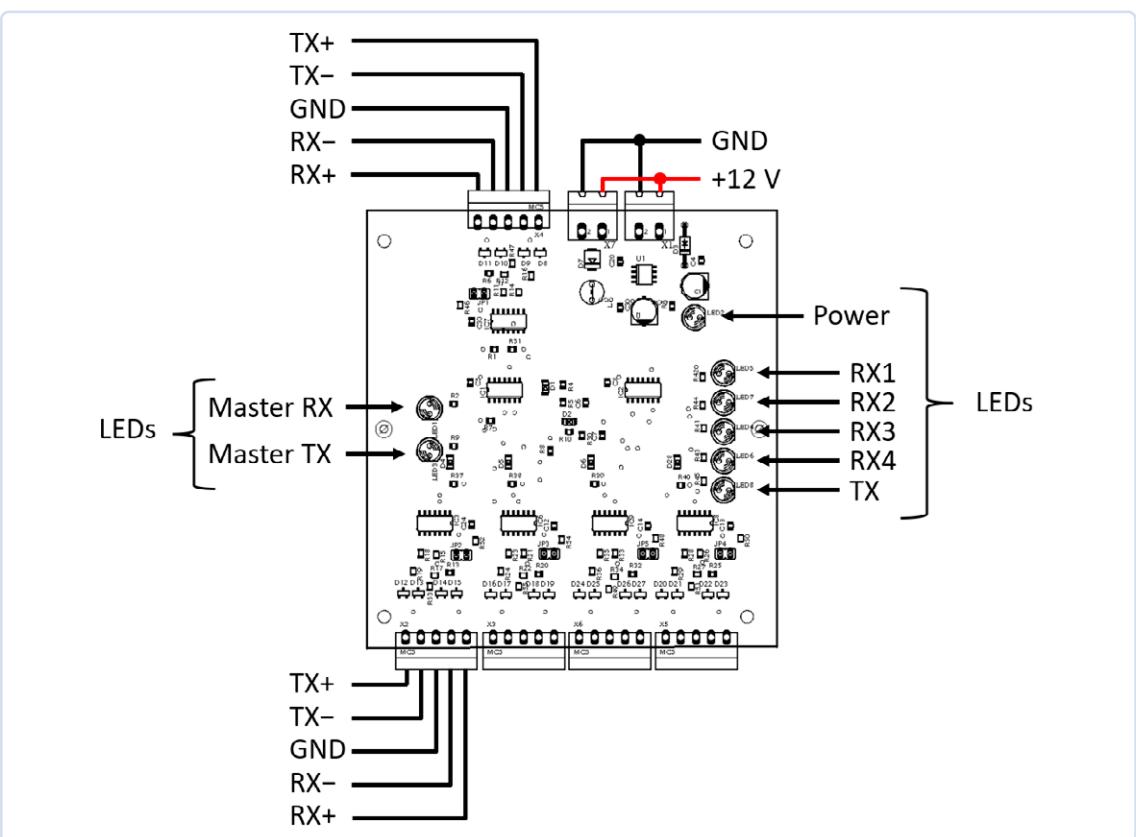
de données (voir le schéma de câblage à la fin de cet article).

Le bloc logique utilisé pour éviter les collisions et gérer automatiquement la direction du signal (TXE) est basé sur la mise en œuvre d'un délai, au moyen d'un simple circuit RC et de deux inverseurs à déclenchement de Schmitt. Ce sous-circuit est visible sur la **figure 8** ; deux de ces sous-circuits sont utili-

sés. Le premier utilise le signal provenant de la sortie de IC7 et le second est lié à la sortie du groupe IC3/IC6/IC8/IC9. Une simulation du comportement est présentée dans la **figure 9**. Pendant tout le temps où une transmission est active (en bleu), la sortie (en rouge) reste haute.

Cependant, pour que le système fonctionne correctement, quelques règles doivent être

respectées : premièrement, il ne doit y avoir qu'un seul appareil qui envoie des données sur le bus à un moment donné, et deuxièmement, le débit de données doit rester suffisamment élevé pour que le filtre RC puisse jouer son rôle. L'alimentation de la carte repose sur un convertisseur CC/CC abaisseur conventionnel basé sur un Texas Instruments LM2674-5, fournissant 5 V.



Mise en œuvre pratique

La carte a été réalisée sur un circuit imprimé double face (**figure 10**) ; ce prototype fonctionne sans problème chez moi depuis sept ans pour permettre à mon serveur domotique (qui est fait maison et compatible avec Home Assistant) de communiquer avec plusieurs cartes situées dans plusieurs bâtiments différents. D'autres vues du circuit imprimé sont disponibles sur la page du projet Elektor Labs [7]. Le schéma de câblage des connecteurs de la carte est présenté dans la **figure 11**. Le câblage des trois sorties restantes en bas est le même que celui de la sortie la plus à gauche. ↵

VF : Maxime Valens — 210725-04



À propos de l'auteur

Sébastien Guerreiro de Brito a découvert l'électronique à l'adolescence en achetant le numéro double d'Elektor de juillet/août. Après avoir obtenu un diplôme d'ingénieur en électronique et informatique industrielle à Polytech Nantes, il a développé diverses cartes pour son usage personnel. Pour partager sa passion, il publie fréquemment certaines de ces cartes dans Elektor.

Produit

➤ **Module de relais industriel 6 canaux Waveshare pour Raspberry Pi Zero**
www.elektor.fr/20085



Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).

LIENS

- [1] RS-422 sur Wikipedia : <https://fr.wikipedia.org/wiki/EIA-422>
- [2] RS-485 sur Wikipedia : <https://fr.wikipedia.org/wiki/EIA-485>
- [3] Texas Instruments, Guide de conception RS-485 : <https://ti.com/lit/an/slla272d/slla272d.pdf>
- [4] Texas Instruments, Aperçu des normes RS-422 et RS-485 et configurations du système (Aperçu des normes RS-422 et RS-485 et configurations du système) : <https://ti.com/lit/an/slla070d/slla070d.pdf>
- [5] Analog Devices, Guide de mise en œuvre du circuit RS-485/RS-422 : <https://analog.com/en/resources/app-notes/an-960.html>
- [6] Fiche technique du MAX489 : <https://analog.com/media/en/technical-documentation/data-sheets/MAX1487-MAX491.pdf>
- [7] Page du projet sur Elektor Labs : <https://www.elektormagazine.fr/labs/rs422485-hub>

YOUR KEY TO CELLULAR TECHNOLOGY



**WÜRTH
ELEKTRONIK**
MORE THAN
YOU EXPECT

WE meet @ electronica
Hall A6-502

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

#GOCCELLULAR

sonde RF

avec bargraphé à LED

Glen Popiel, KW5GP (ÉTATS-UNIS)

Grâce à cet article, les radioamateurs et les passionnés d'Arduino de tous niveaux pourront concevoir un renifleur de signaux RF à large bande. Ce projet utilise l'amplificateur logarithmique AD8307 d'Analog Devices, un ruban LED RGB WS2812, et un Arduino Nano, accompagné d'un programme spécifique. Glen, KW5GP, c'est à vous de prendre le relais !

Note de la rédaction. Cet article est un extrait du livre d'Elektor Arduino for Radio Amateur Applications, 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'encladré **Questions ou commentaires** ?

L'un des atouts de l'Arduino est sa capacité à être utilisé comme un outil pour créer d'autres outils. Si vous êtes comme la plupart des radioamateurs, vous n'avez généralement pas besoin de beaucoup d'équipement de test, à part un multimètre et un wattmètre. Lorsqu'un test nécessite un matériel spécifique que vous ne possédez pas, il est souvent possible d'emprunter ce dont vous avez

besoin à un autre radioamateur. La raison principale pour laquelle vous ne possédez pas votre propre arsenal d'équipement de test est que certains peuvent être assez onéreux, ou qu'il ne semble pas justifié d'investir dans des appareils que vous utiliserez très rarement.

Il est rare que j'aie besoin d'une sonde RF (**figure 1**), mais elle peut s'avérer utile lorsqu'il s'agit de vérifier le fonctionnement d'un émetteur, sans disposer d'un wattmètre ou d'un ROS mètre à proximité, ou encore lorsqu'il est néces-

saire de détecter des interférences RF parasites. En effectuant des recherches pour un autre projet, j'ai découvert l'article RF Driven On-Air Indicator de Keith Austermiller, KB9STR, paru dans le numéro d'août 2004 de QST Magazine. Cet article s'inspire lui-même du No Fibbin' RF Field Strength Meter de John D. Noakes, VE7NI, publié dans le numéro d'août 2002 du même magazine. Avec quelques modifications mineures, ces projets pourraient être adaptés pour créer une sonde RF qui permettrait à un Arduino de piloter des LED adressable WS2812 remplaçant ainsi un compteur classique pour indiquer l'intensité relative du signal RF.

Idée de conception et composants principaux

Bien que ce projet puisse paraître relativement simple et direct, je m'attache toujours à réaliser une documentation exhaustive. Disposer de cette documentation est extrêmement bénéfique si jamais je décide de revenir sur ce projet ultérieurement pour le modifier, y intégrer de nouvelles fonctionnalités, etc.

La **figure 2** montre le schéma fonctionnel du projet de sonde RF. Outre l'entrée de détection RF, l'Arduino est programmé pour afficher l'intensité du signal sur un ruban à 8 LED RGB WS2812. Cet aspect simplifie grandement ce projet. Au lieu de devoir contrôler des LED RGB individuelles, chacune nécessitant trois broches d'E/S numériques et un total de 24 broches d'E/S pour un affichage à 8 LED, les barres et bandes de LED RGB WS2812 adressables ne requièrent qu'une seule broche d'E/S numérique. Il est également possible d'utiliser une puce de contrôle de bande de LED tel que le MAX7219 d'Analog Devices qui n'utilise qu'une seule broche d'E/S numérique. Mais pourquoi introduire la complexité d'une puce à 24 broches quand elle n'est pas nécessaire ?

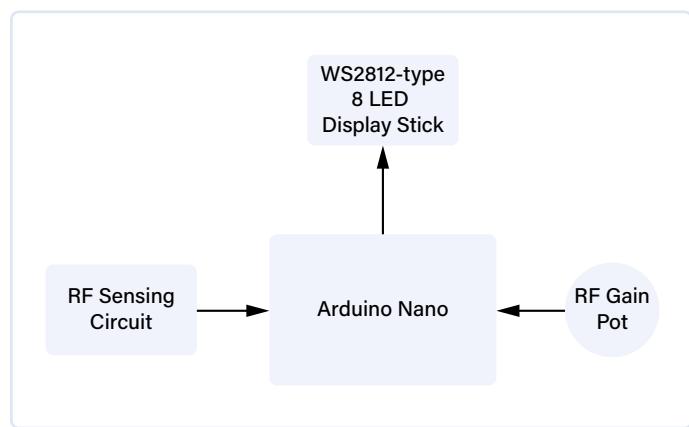


Figure 2. Schéma fonctionnel de la sonde RF.

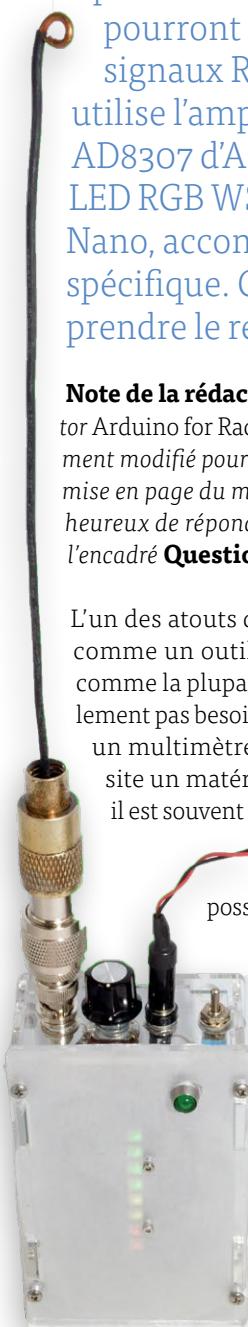
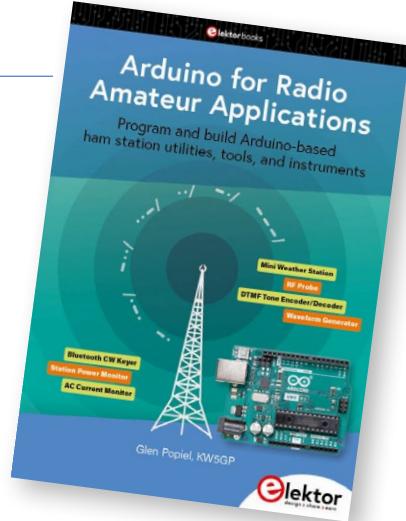


Figure 1. La sonde RF terminée.



RF Probe

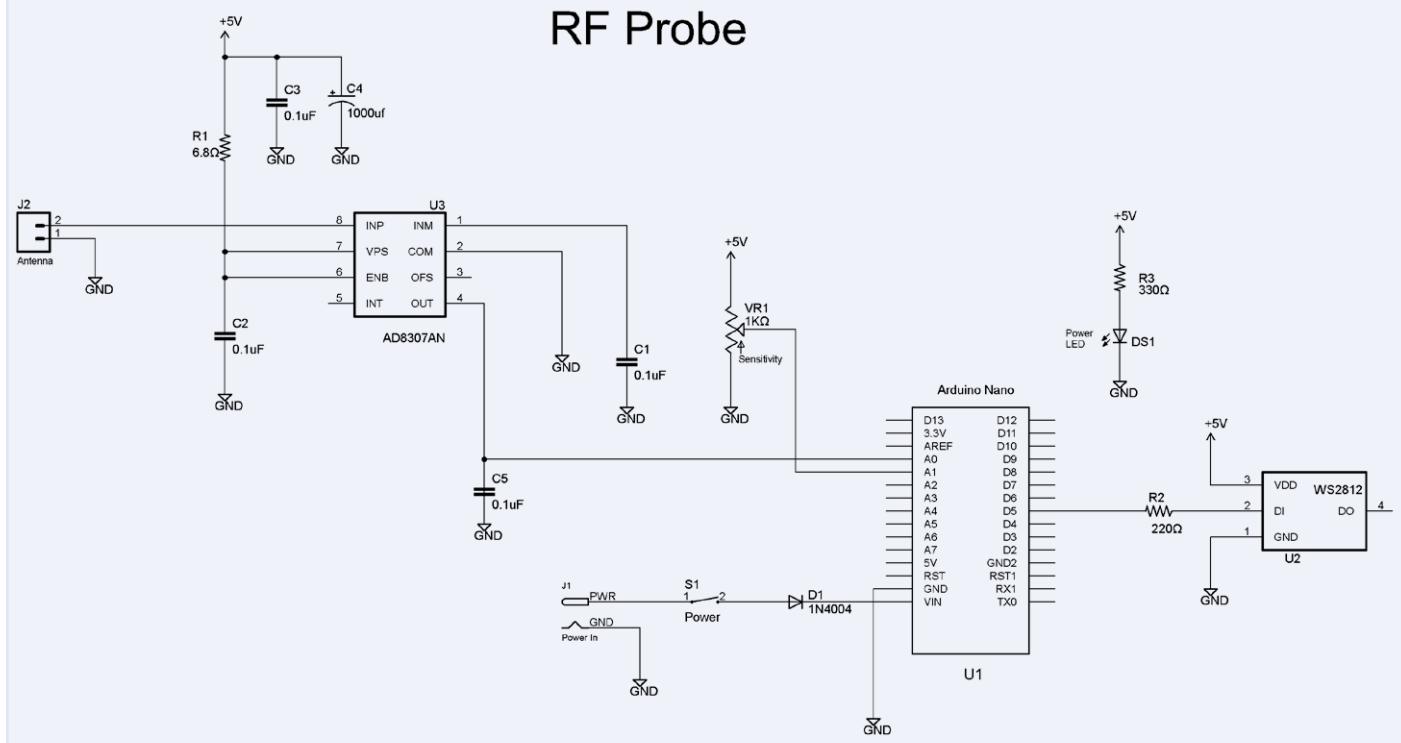


Figure 3. Schéma de la sonde RF.

Schéma et construction

La construction de la sonde RF diffère légèrement de l'approche habituelle décrite dans le reste du livre. Au lieu de prototyper l'unité de détection RF sur une plaque d'essai et de se préoccuper des interférences RF dues au câblage, le circuit de détection RF est assemblé et monté sur la même plaque que l'Arduino Nano. Plutôt que d'utiliser le circuit d'ampli-op présenté dans l'article sur l'*indicateur on-air piloté par RF*, j'ai décidé de moderniser un peu le circuit en utilisant un amplificateur logarithmique AD8307 d'Analog Devices, comme illustré dans le schéma de la sonde RF à la **figure 3**. Capable de détecter des signaux RF jusqu'à 500 MHz, la sonde RF est ainsi adaptée à presque tous vos besoins de détection hors des bandes de fréquences des téléphones cellulaires et du Wifi. Puisque la sortie de l'AD8307 est une tension analogique représentant l'intensité du signal, nous pouvons connecter la sortie de l'AD8307 directement à l'entrée analogique A0 de l'Arduino Nano.

Le connecteur d'antenne est fixé sur un panneau latéral d'un boîtier Solarbotics et se connecte à la carte de circuit imprimé de la sonde RF en utilisant un connecteur mâle à pas de 0,1 pouce (2,54 mm) et un connecteur femelle DuPont. Cette configuration permet de retirer facilement la carte pour rectifier toute erreur de câblage qui pourrait survenir. La puce de l'amplificateur logarithmique AD8307 est montée dans un socket. Cette précaution est utile pour les occasions où elle pourrait être accidentellement exposée à une puissance de 100 watts, au risque de la faire surchauffer et de laisser échapper de la fumée.

La sonde RF est construite sur une carte de prototypage de 60 mm × 80 mm (**figure 4**) et est installée dans un boîtier Solarbotics Mega SAFE transparent. Un connecteur de châssis coaxial SO-239 a été monté sur le boîtier, ce qui vous permet d'utiliser différentes antennes pour la capture des signaux RF. Nous avons soudé un segment de 10 pouces (25 cm) de fil solide #14 AWG au conducteur central d'un connecteur coaxial PL-259 pour faire office d'antenne pour circuit de détection RF.

Entrée, sortie et commandes

Pour ce projet, nous utiliserons le convertisseur analogique-numérique (ADC) 10 bits intégré de l'Arduino Nano pour convertir la sortie de tension analogique de l'AD8307 en une valeur numérique. Cette valeur sera ensuite convertie pour déterminer le nombre de LED à allumer sur une bande de 8 LED RGB WS2812, indiquant ainsi l'intensité du signal. Pour la sonde RF, l'utilisation d'un écran TFT pour l'affichage n'est pas nécessaire car nous recherchons uniquement une indication relative de l'intensité du champ RF, ce à quoi se prêtent idéalement les bandes de LED WS2812. Nous utiliserons un potentiomètre de 1 kΩ pour régler la sensibilité RF de la sonde.

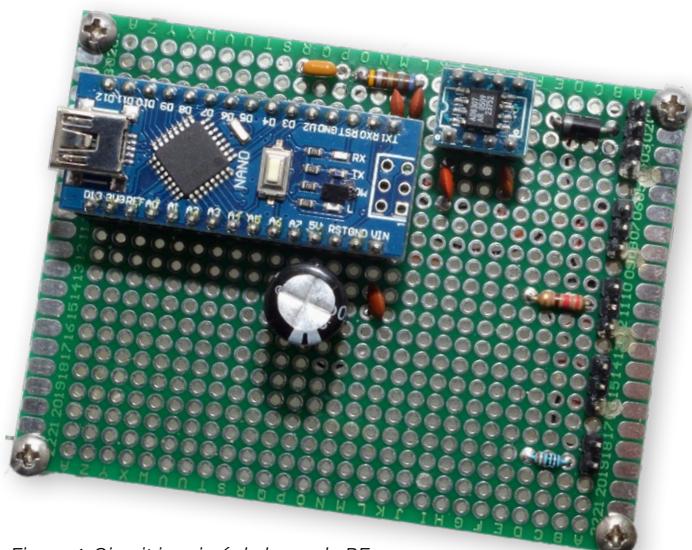


Figure 4. Circuit imprimé de la sonde RF.

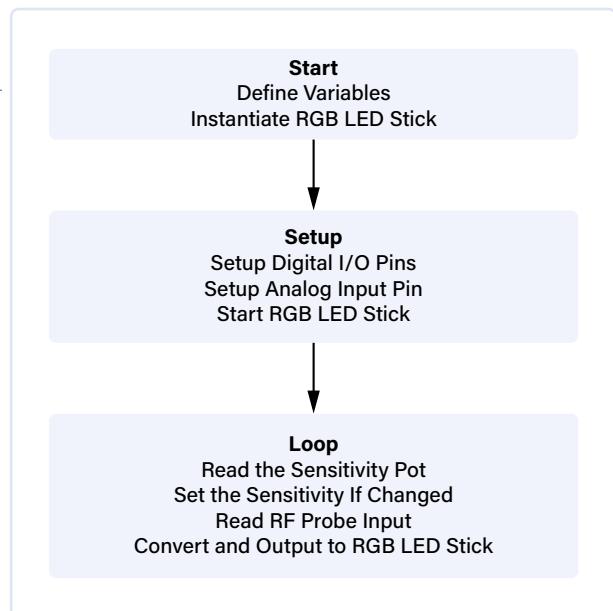


Figure 5. Organigramme de la sonde RF.

Croquis Arduino

Le programme pour la sonde RF est en fait assez simple. Grâce à l'organigramme du projet illustré dans la **figure 5**, le code est assez concis, ce qui démontre l'efficacité de l'Arduino pour gérer les tâches d'entrée/sortie simples requises dans ce projet. Le programme se contente de lire la tension analogique de l'AD8307 et de sortir une représentation graphique de cette tension d'entrée sur la bande de LED.

Le croquis complet ainsi que les bibliothèques de ce projet sont disponibles en ligne [1]. Le croquis utilisera quatre LED vertes, deux jaunes et deux rouges, pour indiquer l'intensité du signal détecté, comme le montre la **figure 6**. Vous remarquerez que j'ai utilisé un petit morceau de papier de soie collé à l'intérieur du couvercle transparent du boîtier comme diffuseur. Vous pourriez également utiliser l'un des boîtiers translucides de Solarbotics.

Concernant le code, nous commençons par inclure incluant la bibliothèque Adafruit_NeoPixel pour piloter le ruban de LED RGB adressable. Il existe de nombreuses bibliothèques pour les LED RGB adressables de type WS2812, mais je préfère toutes les fonctionnalités et la flexibilité de la bibliothèque Adafruit.

```
#include <Adafruit_NeoPixel.h>
```

Ensuite, nous allons définir le nombre de LED sur le ruban de LED adressable et la broche d'E/S à laquelle il sera connecté. L'un des avantages du WS2812 est sa capacité à gérer en interne la modulation de largeur d'impulsion (PWM) nécessaire pour contrôler l'intensité de chaque LED, ce qui permet d'utiliser n'importe quelle broche d'E/S numérique de l'Arduino pour la connexion à la ligne de données. Les rubans et bandes de LED RGB adressables WS2812 ne nécessitent qu'une seule broche d'E/S numérique pour fonctionner. Nous utiliserons également les directives `#define` pour simplifier la configuration des couleurs des LED.

```
#define NUM_LEDS 8
#define LED_PIN 5
#define RF_input_pin A0
#define gain_pin A1
#define led_Red 255, 0, 0
#define led_Green 0, 255, 0
#define led_Yellow 255, 255, 0
#define led_brightness 50
#define update_delay 200
```



Figure 6. Le boîtier de la sonde RF.

Après avoir déclaré les variables nécessaires à ce croquis, nous procéderons à l'instanciation du ruban de LED.

```
int signal_strength = 0;
int effective_strength = 0;
int max_strength = 1000;
int LEDsToLight = 0;
int rf_gain;

Adafruit_NeoPixel pixels
  (NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
```

La fonction `setup()` de la sonde RF est conçue pour être aussi concise et efficace que possible. Elle se limite à initialiser le ruban de LED RGB adressable, à éteindre toutes les LED, puis à invoquer la méthode `show()` de la classe NeoPixel pour activer l'effacement des LED.

```
pixels.begin(); // INITIALIZE NeoPixel strip object
pixels.clear(); // Set all pixel colors to 'off'
pixels.show();
```

Dans la boucle `loop()`, nous commençons par lire la tension analogique du potentiomètre de gain RF connecté à la broche d'entrée analogique 1 de l'Arduino. Ensuite, nous mesurons l'intensité du signal RF sur la broche d'entrée analogique 0. Nous utilisons la valeur obtenue du réglage de gain pour ajuster la fonction `map()` de l'Arduino, ce qui permet de régler la sensibilité des LED de la sonde RF.

Il s'agit d'un moment idéal pour introduire la fonction `map()` d'Arduino, utilisée pour modifier l'échelle de vos données. Le signal RF détecté, dont l'amplitude varie de 0 à 5 V et est mesurée par l'AD8307, est converti d'une plage numérique possible de 0 à 1023 (correspondant à la résolution de 10 bits de l'Arduino) à une échelle de 0 à 8, qui représente le nombre de LED sur le ruban. La valeur obtenue nous indique ainsi le nombre de LED à allumer pour indiquer l'intensité du champ RF.



Liste des composants

Résistances

R1 = Résistance de 6,8 Ω, 1/8 W

R2 = Résistance de 220 Ω, 1/8 W

R3 = Résistance de 330 Ω, 1/8 W

VR1 = Potentiomètre de 1 kΩ

Condensateurs

C1, C2, C3, C5 = 01 µF, 25 V condensateur

C4 = 1000 µF, condensateur de 25 V

Semi-conducteurs

D1 = diode 1N4004

DS1 = LED de 5 mm

U2 = bâton WS2812 à 8 LEDs

U3 = AD8307 CI amplificateur logarithmique

Divers

J1 = Prise d'alimentation DC

S1 = Interrupteur SPST

U1 = Arduino Nano

Boîtier : Solarbotics Mega S.A.F.E.

```
LEDsToLight = map(effective_strength, 0, max_strength,
0, NUM_LEDS);
```

```
if (LEDsToLight < 0)
{
    LEDsToLight = 0;
}
```

```
if (LEDsToLight > NUM_LEDS)
{
    LEDsToLight = NUM_LEDS;
}
```

Maintenant que nous avons déterminé le nombre de LED à allumer, il nous faut choisir la couleur des LED allumées. Comme convenu, les quatre premières LED s'illumineront en vert, les deux suivantes en jaune, et les deux dernières en rouge. Nous commençons par effacer la configuration précédente des LED, puis nous activons les LED correspondant à l'intensité actuelle du signal RF

```
pixels.clear();
```

Ensuite, les couleurs et le nombre souhaité de LED à allumer sont définis.

```
pixels.setBrightness(led_brightness);

for (int led = 0; led < LEDsToLight; led = led + 1)
{
    if (led <= 3)
    {
        pixels.setPixelColor(led, pixels.Color(led_Green));
    }
    if (led > 3 && led <= 6)
    {
        pixels.setPixelColor(led, pixels.Color(led_Yellow));
    }
    if (led > 5)
    {
        pixels.setPixelColor(led, pixels.Color(led_Red));
    }
}
```

Enfin, nous utilisons la fonction `pixels.show()` pour allumer le nombre désiré de LED dans les couleurs spécifiées.

```
pixels.show();
```

La valeur `update_delay` est utilisée pour contrôler l'intervalle entre les itérations du croquis. Je l'ai réglée à 200 ms dans le programme, ce qui permet d'obtenir cinq mises à jour par seconde. Vous pouvez ajuster cette valeur selon vos besoins. Une valeur de 0 mettra à jour les LED sur le ruban aussi rapidement que l'Arduino le permet, offrant ainsi un affichage presque en temps réel de l'intensité du signal. Il est également possible d'utiliser cette technique pour créer un indicateur de niveau audio dynamique pour les applications audio.

```
delay(update_delay);
```

Idées d'amélioration

Ce projet se prêterait parfaitement à une réalisation dans une boîte de bonbons Altoids, en utilisant un Arduino Nano. En optant pour un connecteur SMA plutôt qu'un SO-239, il serait possible de tout assembler à l'intérieur de la boîte, avec juste une petite ouverture pour le ruban de LED adressables. Une autre modification envisageable serait de remplacer le ruban de LED par un écran OLED compact, ce qui permettrait à la sonde RF d'afficher une représentation numérique de l'intensité relative du champ RF.

240326-04

Questions ou commentaires ?

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



Produits

➤ **Glen Popiel, Arduino for Radio Amateur Applications (Elektor 2024)**

Version papier : www.elektor.fr/20814

version numérique : www.elektor.fr/20815

LIEN

[1] Croquis et bibliothèques : <https://tinyurl.com/kw5gpelektor>



démarrer en électronique...

...plus de montages à ampli-op

Eric Bogers (Elektor)

Dans le dernier épisode, nous avons examiné le montage de l'amplificateur d'instrumentation. Ayant suffisamment couvert la théorie, il est maintenant temps de nous pencher sur les circuits pratiques.

Mais avant de poursuivre, ajoutons un petit complément à notre dernier article. Sous le paragraphe « Connexions symétriques », nous avons mentionné ce qui suit :

« Pour connecter des haut-parleurs, il est courant d'utiliser un câble bifilaire (non blindé) à âme torsadée. L'un des fils sert de phase et l'autre de masse. En général, le choix du fil n'a pas d'importance ; la connexion du câble au connecteur suit une norme, et l'un des fils est marqué pour faciliter l'identification. »

Comme l'a justement souligné mon collègue Brian Tristam Williams, cette explication est un peu trop simpliste. Il est vrai que, d'un point de vue électrique, la manière dont les enceintes sont connectées aux sorties d'un amplificateur peut sembler indifférente. Cependant, dans le cas d'un amplificateur stéréo, cela peut faire une grande différence pour l'auditeur. Pour une expérience d'écoute optimale, il est crucial que les deux enceintes (ou boîtiers d'enceintes) soient connectées en phase afin d'éviter les courts-circuits acoustiques, qui peuvent affecter les basses fréquences. Sur la plupart des amplificateurs et des enceintes, les sorties et les entrées sont clairement marquées, tout comme l'un des fils du câble d'enceinte, pour faciliter cette connexion correcte.

Les amplificateurs opérationnels ne se limitent pas aux simples circuits d'amplifi-

cation ; dans l'électronique basse fréquence, ils sont utilisés comme des « semi-conducteurs polyvalents » pour une multitude d'applications.

Redresseur de précision

Lorsqu'il est utilisé dans les alimentations, la précision du redresseur n'est généralement pas cruciale ; notre objectif principal est de convertir une tension alternative en une tension continue de manière aussi efficace que possible. Cependant, l'utilisation d'un circuit simple de diodes comme redresseur dans un instrument de mesure pose un problème spécifique : la tension de sortie est systématiquement inférieure à la tension d'entrée, qui est égale à la tension directe de la diode.

Pour contourner ce problème, il est possible de connecter la diode de redressement à la sortie d'un ampli-op et de placer la rétroaction après la diode, comme illustré dans la **figure 1**. L'ampli-op amplifie le signal suffisamment pour compenser précisément la chute de tension au niveau de la diode.

Courant de sortie plus élevé

Les amplificateurs opérationnels sont généralement capables de fournir seulement quelques mA de courant de sortie. Bien qu'ils soient équipés de protections contre les courts-circuits, prévenant ainsi les dommages dus aux faibles résistances de charge, ils ne sont pas conçus pour fournir

des courants de sortie élevés. Toutefois, il existe une solution à cette limitation : associer un ampli-op à un étage de sortie équilibré, comme le montre la **figure 2**.

Une fois de plus, la rétroaction est connectée après les transistors pour s'assurer que le circuit ne génère pas une distorsion excessive, malgré les tensions BE directes des transistors.

Toutefois, cette faible distorsion est uniquement due au gain élevé en boucle ouverte, qui, nous le savons désormais, diminue significativement avec l'augmentation de la fréquence. Le circuit, illustré dans la **figure 2**, est généralement très performant pour des applications en courant continu. Mais, il gère moins bien les signaux audio. Dans ces conditions, il est préférable d'opter pour un étage de sortie de classe B (voir **figure 3**). Dans ce cas, une diode (alimentée en

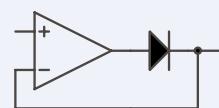


Figure 1. Redresseur de précision.

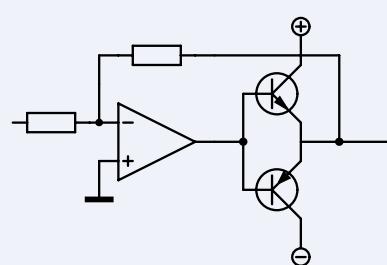


Figure 2. Augmentation du courant de sortie d'un ampli-op.

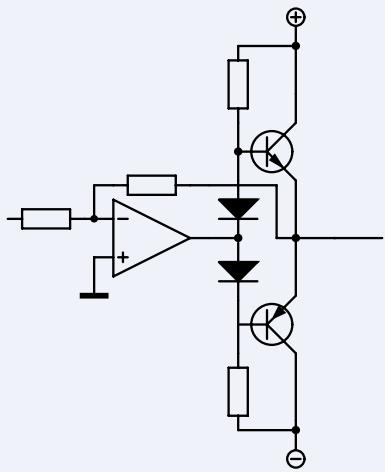


Figure 3. Réduction de la distorsion par l'ajout d'un étage de sortie en classe B.

courant via une résistance) fournit la tension de polarisation du transistor correspondant dans chaque cas.

Lorsque la qualité supérieure est une exigence, le circuit conforme à la configuration illustrée dans la **figure 4** peut être utilisé. Ce circuit est issu du lecteur de CD A725 CD de Studer, qui intègre un transformateur de sortie. Ce lecteur de CD professionnel, lancé pour la première fois en 1986, est toujours disponible aujourd'hui à des tarifs élevés sur des plateformes de vente en ligne comme eBay.

Dans ce circuit, la tension de polarisation des transistors de puissance n'est pas fournie par une diode, mais par un transis-

tor identique. Ce transistor provient du même wafer (c'est-à-dire du même disque de silicium à partir duquel les transistors individuels sont découpés) et est couplé thermiquement avec le transistor de sortie - à cette fin, les transistors sont pressés l'un contre l'autre et fixés par du cuivre.

De cette manière, la tension de polarisation a toujours la valeur optimale. Pour supprimer les interférences éventuelles, la tension de consigne est tamponnée à l'aide d'un condensateur électrolytique.

Deux autres détails de ce circuit méritent une attention particulière : premièrement, le transformateur de sortie est contrôlé par un pont. Cela permet d'éviter que des courants élevés ne circulent à travers la connexion à la masse, réduisant ainsi les tensions d'interférence à travers la résistance du câblage/des pistes. Pour obtenir la meilleure qualité de signal possible, il est crucial de minimiser autant que possible le courant dans votre ligne de masse !

Deuxièmement, la tension d'alimentation des transistors de sortie est connectée via des résistances CTP. Cela offre une protection efficace contre les surcharges, car si l'impédance de la charge est trop faible, la tension d'alimentation est automatiquement coupée.

Pensez aux situations dans lesquelles un haut-parleur est simplement connecté au socket de sortie XLR à des fins de test. Si la protection contre les surcharges était assurée par des relais qui se contentent de couper la charge, on n'entendrait rapidement plus rien ; et si une impédance de charge plus élevée était connectée, la qualité audio en fonctionnement normal aurait déjà subi une dégradation.

Nous tenons à remercier la société Studer pour nous avoir autorisé à publier ce schéma ici. De plus, ils nous ont fourni un conseil précieux :

Pour les amateurs désireux de reconstruire ce circuit, il est important de noter que son fonctionnement optimal est assuré uniquement si les paires de transistors sont issues de la même plaquette de silicium (wafer). Cependant, pour la plupart des amateurs, se procurer de telles paires de transistors peut s'avérer extrêmement difficile. De plus, ces transistors doivent être couplés thermiquement par paires. Si ces conditions ne sont pas remplies, il est conseillé d'augmenter la valeur des résistances de $3,3\ \Omega$ à environ $10\ \Omega$ pour garantir une sécurité accrue.

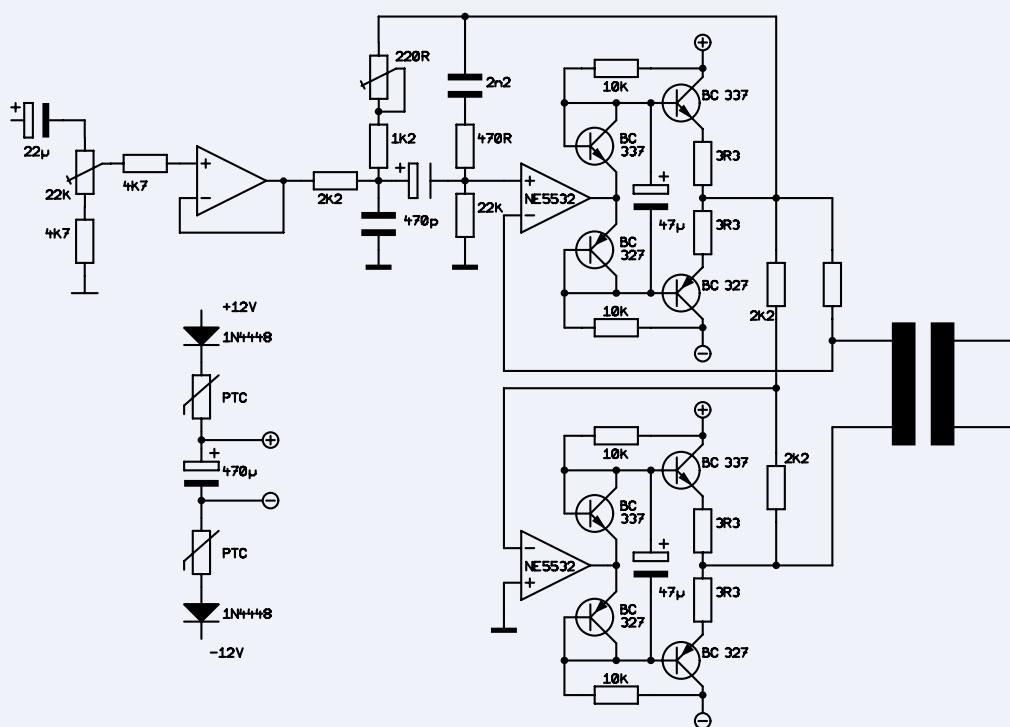


Figure 4. Amplificateur de sortie Studer A725.

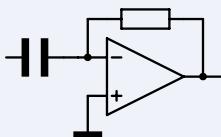


Figure 5. Différenciateur

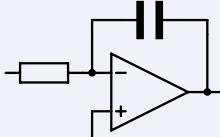


Figure 6. Intégrateur.

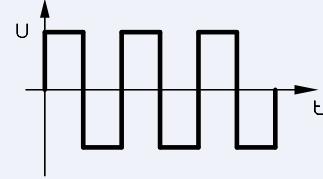


Figure 7. Signal carré.

Différenciateur

Les amplificateurs opérationnels peuvent également être utilisés pour effectuer des opérations mathématiques complexes, telles que la différentiation et l'intégration. La **figure 5** illustre le schéma d'un différentiateur. La tension de sortie peut être calculée à l'aide de la formule suivante :

$$U_{\text{out}} = -R \cdot C \cdot \frac{\Delta U_{\text{in}}}{\Delta t}$$

Intégrateur

L'intégrateur, illustré dans la **figure 6**, est l'inverse du différentiateur. La tension de sortie de l'intégrateur est :

$$U_{\text{out}} = \frac{1}{R \cdot C} \cdot \int U_{\text{in}} \cdot dt + C$$

Le terme « $+ C$ » pose souvent des difficultés aux électroniciens expérimentateurs, car il tend à devenir si élevé que le résultat final excède la plage de tension de sortie de l'ampli-op. De plus, les tensions et courants d'offset ont un impact négatif notable, car ils sont intégrés très rapidement, ce qui entraîne un écrêtage de la tension de sortie par rapport aux tensions d'alimentation positive ou négative.

Générateur d'ondes carrées

Le générateur d'ondes carrées est un générateur de signaux qui produit un signal carré (**figure 7**) (ce que vous aviez probablement déjà deviné). Dans un épisode précédent, nous avons utilisé un multivibrateur astable pour générer un tel signal. Le circuit à

ampli-op présenté dans la **figure 8** a l'avantage de ne nécessiter qu'un seul condensateur.

Nous supposons que la sortie est écrêtée par rapport à la tension d'alimentation positive. Le diviseur de tension $R2/R3$ réduit la tension de sortie de moitié (par exemple). $C1$ est alors chargé via $R1$. Dès que la tension aux bornes de $C1$ devient aussi élevée que la tension à l'entrée non inverseuse de l'ampli-op, la sortie bascule et la tension devient négative, tout comme l'entrée non inverseuse. $C1$ est ensuite déchargé via $R1$. Dès que la tension aux bornes du condensateur descend au niveau de l'entrée non inverseuse, la sortie de l'ampli-op bascule à nouveau et redévient positive. Ce processus se répète jusqu'à ce que l'appareil soit débranché.

Générateur d'onde triangulaire

Il est également possible d'utiliser le circuit de la figure 8 comme générateur d'ondes triangulaires. Si la tension à l'entrée non-inverseuse est réduite à sa valeur minimale, le condensateur se charge avec un courant presque constant. En conséquence, un signal triangulaire apparaît sur l'entrée inverseuse.

Il est également possible de générer une tension triangulaire en utilisant un intégrateur (voir **figure 9**). L'ampli-op de gauche est un générateur de signaux carrés, comme nous l'avons exploré dans la section précédente. L'amplificateur opérationnel de droite intègre le signal carré, produisant ainsi une forme d'onde triangulaire. Sans ajustements, la tension de sortie s'écrêterait

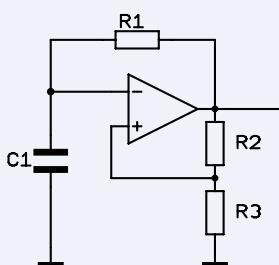


Figure 8. Générateur de signaux carrés.

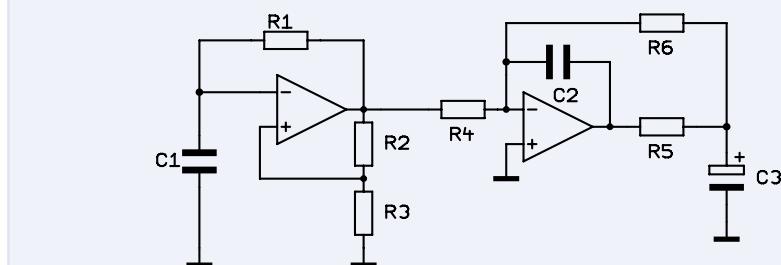


Figure 9. Générateur de signaux triangulaires.

contre la tension d'alimentation positive ou négative, causant une distorsion significative. C'est pourquoi le condensateur $C3$ est chargé via $R5$ - et la tension aux bornes de ce condensateur est transmise via $R6$ à l'entrée inverseuse de l'ampli-op.

Ainsi, nous évitons que la tension de sortie ne s'écrète par rapport à la tension d'alimentation, mais cela a pour conséquence de rendre le signal de sortie légèrement asymétrique. Si nécessaire, ce déséquilibre peut être corrigé en filtrant la tension de sortie avec un filtre passe-haut, par exemple. Nous en restons là pour l'instant ; dans le prochain épisode, nous explorerons d'autres petits circuits intéressants utilisant des ampli-op. ↗

240310-04

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

► **B. Kainka, Basic Electronics for Beginners** (Elektor, 2020)

version papier :

www.elektor.fr/19212

version numérique :

www.elektor.fr/19213

VOUS AVEZ DE LA CHANCE !

TELECHARGEZ
GRATUITEMENT



Un abonné à l'e-zine ne manque jamais
le mensuel «reverse project»

Vous n'êtes pas encore abonné ? Inscrivez-vous à
notre e-zine gratuit à elektormagazine.fr/ezine-24



Open Vario

variomètre multifonction open source pour vol en parapente

Cédric Jimenez (France)

Parfois, les meilleures créations résultent d'une opportunité ! Cédric Jimenez, lauréat du premier prix du concours STM32 Wireless Innovation Design Contest, avait été informé de ce challenge par un ami et développa un projet innovant pour la mesure de l'altitude de vol par un variomètre. En voici le récit complet.

Tout a commencé lorsqu'un ami m'a transmis le lien vers la page du concours " STM32 Wireless Innovation Design Contest " d'Elektor magazine [1] : "Hey ! as-tu vu ce concours ? Il est basé sur l'utilisation d'un chip sans-fil de ST Microelectronics, et ils offrent une carte d'évaluation !" Il savait comment susciter mon attention, car j'adore pouvoir expérimenter les nouveaux microcontrôleurs, et je suis un grand amateur de ces cartes, qui permettent de mettre en œuvre rapidement des projets pour découvrir leurs fonctionnalités.

Choix du matériel

J'ai pris connaissance des cartes d'évaluation proposées : deux cartes Nucleos (NUCLEO-WBA52CG, NUCLEO-WL55JC) et un kit d'expérimentation (STM32WB5MM-DK). Ce dernier était, selon moi, le plus attractif comme base d'un projet, car il intègre de nombreuses possibilités : affichage, mémoire, capteurs, USB, connecteurs d'extensions... Parfait ! j'avais choisi une carte, il me restait à trouver une idée de projet.

Je suis pilote de parapente, et je réfléchissais depuis longtemps à développer mes propres instruments de vol. J'avais même imaginé une conception matérielle, il y a cinq ans, mais je n'ai pas persisté, principalement par manque

de temps. J'ai alors considéré ce concours comme une opportunité, pour transformer cette idée en un appareil que je pourrais utiliser durant mes vols. En vol parapente, le variomètre, également connu sous le nom d'indicateur de vitesse d'ascension et de descente (rate of climb and descent indicator : RCDI) – est l'instrument

le plus précieux ! La vitesse ascensionnelle, exprimé en mètres (ou pieds) par seconde, aide à la recherche des courants ascendants, qui permettent au parapentiste de prendre de l'altitude, et ainsi rester en vol durant plusieurs heures. En plus de la fonction principale d'un variomètre, les instruments commerciaux possèdent des fonctions additionnelles comme le système de navigation par satellite GNSS (Global Navigation Satellite System) contrôlé par un smartphone.

Spécifications exigées

En me basant sur les possibilités offertes par la carte STM32WB5MM-DK, j'ai commencé par dresser une liste des fonctions que je souhaitais implémenter (voir **figure 1**) :

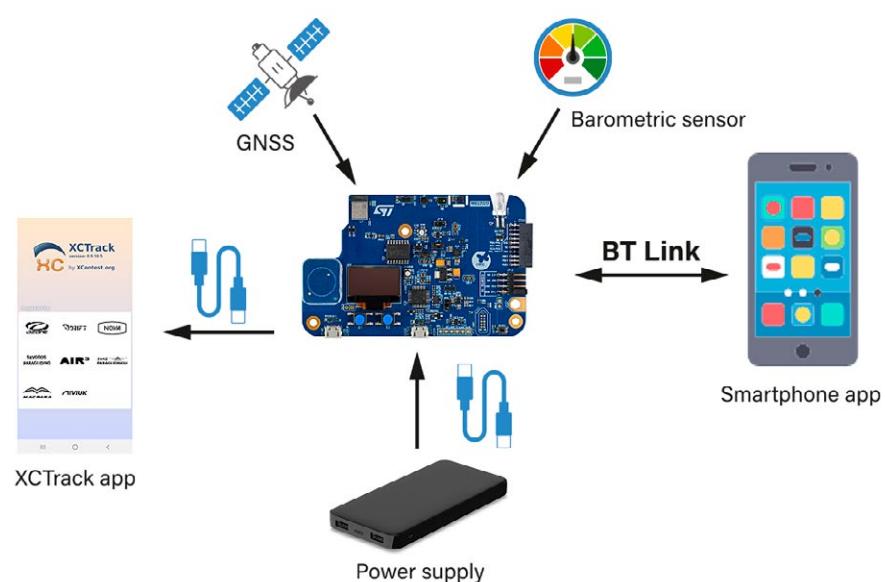


Figure 1. La carte de développement STM32WB5MM-DK est au cœur de ce projet.

- Variomètre
- Positionnement GNSS
- Accéléromètre
- Température
- Calcul du facteur de glissement (i.e., la distance horizontale pouvant être couverte par la perte d'altitude de 1mètre)
- Enregistrement des données de vol (altitude, vitesse, position...)
- Interface de contrôle (IHM) et affichage des données des capteurs, et du vol)
- Lien Bluetooth à faible consommation (BLE), permettant la configuration des paramètres et la lecture des données de vol en temps-réel
- Lien USB OTG permettant de se connecter à l'application smartphone XCTrack (application réputée pouvant être utilisée pour le déport de l'affichage)
- Interface de liaison USB pour se connecter à des utilitaires Python de ma conception, sur PC, permettant l'export des données de vol.

Matériel additionnel nécessaire

Bien que la carte STM32WB5MM-DK puisse remplir la plupart des fonctionnalités requises, il manquait deux capteurs : un récepteur GNSS permettant de recevoir la position de l'appareil et un capteur de pression permettant de calculer la vitesse ascensionnelle.

Le calcul de la vitesse ascensionnelle est réalisé en mesurant la différence d'altitude plusieurs fois par seconde. Un dispositif GNSS ne fournit pas une information d'altitude suffisamment précise, c'est pourquoi un capteur de pression barométrique de haute précision est nécessaire, afin d'obtenir une précision de 15 à 30 cm lors de la mesure de l'attitude. La nécessité de réaliser une intégration électrique robuste avec la carte STM32WB5MM-DK, résistante aux vibrations et aux chocs durant le décollage, ou produites par les turbulences et lors de l'atterrissement, introduit des contraintes supplémentaires. Le format de l'ensemble des dispositifs intégrés doit également être aussi compact que possible et facilement transportable. C'est pourquoi j'ai choisi de mettre à profit le connecteur compatible Arduino UNO de la carte STM32 et l'utilisation d'une carte enfichable 'MikroE insérée sur un module (shield) dédié [3], ainsi, aucune soudure ou fil de liaison n'est nécessaire. MikroElektronika propose une grande diversité de modules capteurs (plus de 1 000), j'en ai souvent utilisé en prototypage rapide. En ce qui concerne le GNSS ? Je possédais

un module enfichable MikroE GNSS 4 [4] comprenant le module U-blox AM-M8Q, qui intègre une antenne patch GNSS omnidirectionnelle, et une interface UART.

Pour le capteur de pression barométrique, MikroE propose trois capteurs de haute-précision pouvant convenir à mon projet. J'ai consulté les forums de vol parapente et parcouru la documentation fournie. J'ai finalement décidé d'utiliser le module Altitude 2 Clic [5] basé sur un capteur de pression MS5607 de TE Connectivity, utilisé dans d'autres projets de variomètres. Il est facile à mettre en œuvre, comporte une documentation de qualité, et peut être connecté par un bus I²C ou SPI.

J'ai ensuite relié l'ensemble, afin de voir comment cela se présentait (vous trouverez un schéma synoptique sur la **figure 2**) et... mauvaise surprise : ni le bus I²C, ni le bus SPI n'étaient présents sur les broches de liaison nécessaires au module de pression barométrique ! Bon, ce n'est pas un point bloquant, le protocole du bus I²C n'est pas complexe, j'ai pu contourner le problème en créant un pilote logiciel I²C.

Conception logicielle

Maintenant que j'avais réuni et interconnecté le matériel nécessaire, je pouvais commencer à réfléchir à la partie logicielle. J'aime partager mes codes source avec d'autres concepteurs, j'aime également pouvoir réutiliser le travail des autres personnes. C'est pour cela que je partage la plupart de mes programmes sous licence open source (logiciel libre), et c'est pourquoi ce projet se devait d'être en open source [6]. Pour ce projet, j'ai choisi la licence MIT afin d'offrir la flexibilité maximum pour l'usage par de futurs utilisateurs ou contributeurs, car ce dispositif n'est pas destiné à devenir un produit commercialisé.

Pour la programmation, j'ai utilisé mon langage favori : C++. Il offre toutes les possibilités bas niveau du langage C et offre en plus toutes les caractéristiques d'un langage moderne orienté objet : sécurité des types, héritage, templates... La plupart des gens pensent "qu'utiliser C++ signifie utiliser l'allocation dynamique de mémoire". Est-ce que l'usage de `malloc()` et `free()` est obligatoire dans un programme C ? Non, pas plus que `new` et `delete` en C++. Cela ne dépend que de la

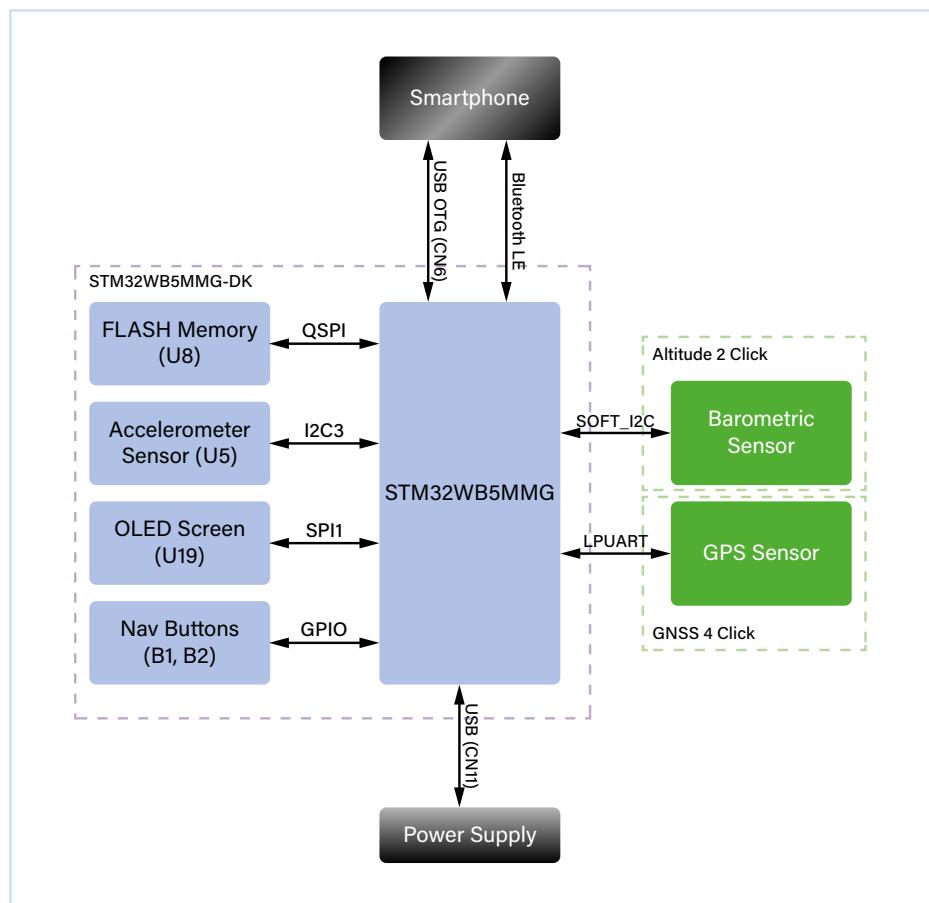


Figure 2. Schéma fonctionnel du système montrant les interfaces physiques aux modules externes.

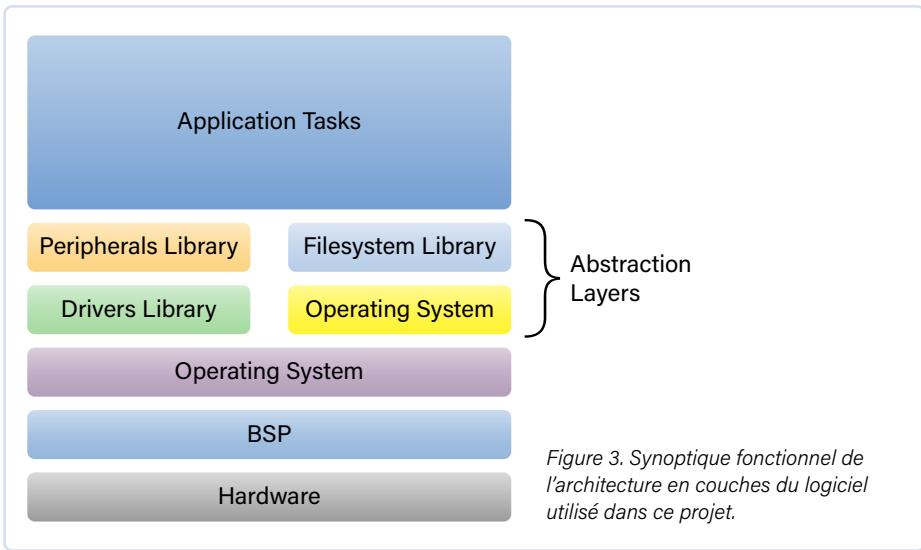


Figure 3. Synoptique fonctionnel de l'architecture en couches du logiciel utilisé dans ce projet.

partie du langage mise en œuvre, de multiples options existent sans utiliser l'allocation dynamique de mémoire.

ST offre une multitude d'outils logiciels pour ses microcontrôleurs (STM32CubeIDE, STM32CubeMX ...) et de nombreux exemples dans la bibliothèque de pilotes ST HAL, ou dans les packages MCU STM32Cube. Bien que j'aime réellement ces outils, qui m'aident à créer rapidement des petits prototypes, en générant une partie du code et en me dirigeant vers les bibliothèques adéquates à utiliser, j'ai préféré utiliser l'environnement de développement intégré (EDI) "Visual Studio Code IDE" avec les extensions C/C++ pour ses possibilités de mise en évidence du code et de navigation, avec en complément, les extensions Cortex-Debug pour la mise au point des périphériques intégrés utilisant les débogueurs ST-Links ou J-Links.

Il offre les mêmes possibilités que STM32CubeIDE, mais il est plus facile à configurer et paramétrier par ses modules attachés, et les fichiers de configuration JSON. J'ai également préféré écrire mes propres systèmes de génération de code exécutable CMake/Make, étant ainsi totalement indépendant de l'EDI utilisé et me permettant d'avoir le contrôle intégral de la configuration du code source et des fichiers de code (aucun code n'est généré par les outils ST).

En plus des bibliothèques *STM32 HAL*, *STM32 USB*, et bien entendu des couches Bluetooth *STM32 WPAN*, j'ai décidé d'utiliser les bibliothèques tierces suivantes (toutes disponibles sur GitHub) :

➤ FreeRTOS 10.6.2 : Système d'exploitation en temps réel (je l'ai trouvé facile à utiliser par rapport à la conception à boucle unique lorsque les ressources ne sont pas trop contraintes).

- Little-FS 2.81: d'utilisation facile, peu encombrant et insensible aux interruptions d'alimentation, en tant que système de fichier pour la mémoire Flash NOR.
- YACSGL/YACSWL 0.0.1: Bibliothèque graphique peu encombrante adaptée aux afficheurs OLED monochromes et/ou aux écrans E-Ink.

J'ai commencé par créer un module C++ encapsulant l'ensemble de ces bibliothèques afin de faciliter leur intégration au code C++. Par ailleurs, pour certaines parties, comme la classe `mutex`, j'ai simulé le comportement C++ standard de `std::mutex` afin de faciliter l'écriture du code C++. J'ai également décidé d'implémenter une architecture logicielle en couche, "classique" mais efficiente (voir la **figure 3**). Enfin, j'ai utilisé ce que j'appelle un "schéma d'abstraction de carte" qui définit une interface me permettant de retrouver tous mes objets traitant des périphériques et les pilotes, par l'intermédiaire, bien entendu, d'une interface abstraite. Après cette abondance de théorie, regardons maintenant le code du **listage 1**, également disponible sur GitHub [6]. En utilisant cette interface, je peux, dans mes tâches applicatives, accéder facilement aux périphériques matériels et aux pilotes sans me préoccuper de leur implémentation. Ceci me permet, par exemple, de facilement prélever les sorties de mes capteurs pour accélérer le développement des algorithmes des tâches applicatives. Dans un autre projet, j'avais utilisé la même approche pour abstraire le matériel et permettre à un même logiciel, de fonctionner sur une multitude de cartes différentes.

Test

La phase ultime nécessitait de tester mon projet dans les conditions de vol réel. À partir

de données simulées, mes algorithmes produisaient des valeurs valides, et se comportaient correctement quand je changeais les paramètres de leurs filtres. Mais en vol parapente, tout est question de ressenti ; j'ai ainsi dû ajuster les paramètres lors de vol réels, afin que le dispositif s'accorde avec ma façon de voler. J'ai également été amené à fournir un moyen de les configurer aisément, afin qu'un autre pilote puisse les ajuster.

Je ne pouvais pas exposer l'appareil à l'air libre en raison de l'impact de l'exposition au soleil (et à l'écoulement d'air) sur le capteur très précis de pression atmosphérique. Au lieu de cela, j'ai utilisé mon smartphone pour visualiser les données et contrôler les paramètres. Il y a deux options pour connecter la carte STM32 à un smartphone, Bluetooth ou la connexion USB OTG ; j'ai expérimenté les deux.

La liaison Bluetooth permet l'accès à tous les paramètres des différents services et caractéristiques (voir le document présent sur la page Elektor Labs de ce projet [7]). Malheureusement, je n'ai pas eu la possibilité de développer une application smartphone dédiée, offrant une interface utilisateur agréable, mais cela reste facilement modifiable en utilisant des

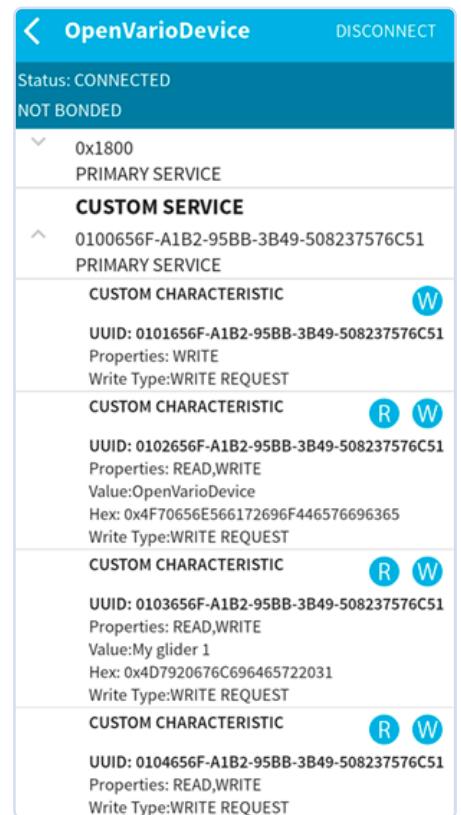


Figure 4. Capture d'écran de l'application BLE Scanner, utilisée pour personnaliser les services d'exécution.



Figure 5. Variomètre commercial utilisé pour comparer les mesures faites par mon Open Vario.

applications BLE génériques gratuites, telle que BLE Scanner (voir la **figure 4**).

Une autre option consiste à utiliser l'affichage sur smartphone durant le vol : l'application XCTrack est communément utilisée par les parapentistes ; elle utilise la liaison OTG du "Discovery Kit". Cela m'a permis de visualiser, sur mon smartphone, les données de vol calculées par mon appareil. J'ai également ajouté à mon installation le variomètre "commercial" que j'utilise habituellement, afin

de comparer les valeurs obtenues, comme le montre la **figure 5**.

Durant les vols, j'enregistre périodiquement (le défaut est 1 s.), toutes les sorties de mes capteurs et des algorithmes qui les accompagnent, en plus de ma position GNSS. Chaque vol est enregistré dans un fichier séparé de la mémoire Flash NOR QSPI. Avec les paramètres par défaut, je peux enregistrer environ 24 heures de vol ce qui est largement suffisant. J'ai développé des scripts Python pour récupérer ces fichiers sur un PC utilisant l'interface USB du "Discovery Kit" et les sauvegarder dans un fichier CSV pouvant être facilement analysé à l'aide d'EXCEL (voir **figure 6**), et également visualisables en format KML dans l'application Google Earth pour observer mon vol en 3D, comme illustré sur la **figure 7**.

NOMBREUSES OPPORTUNITÉS

Je pense que vous avez maintenant réalisé que j'ai réellement pris plaisir à participer à ce concours, même si cela était chronophage (5...10 h par semaine) ! Cette occasion m'a offert de multiples opportunités : découverte d'un nouveau microcontrôleur, réalisation d'un projet auquel je n'avais jamais consacré le temps nécessaire à son développement, tout en mélangeant deux de mes passions

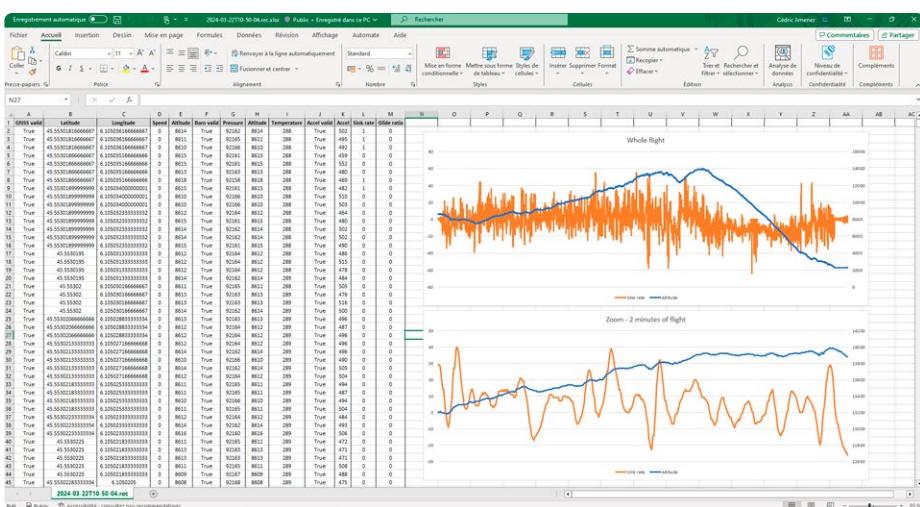


Figure 6. Données acquises, stockées dans un fichier .csv, analysées et visualisées à l'aide d'Excel.

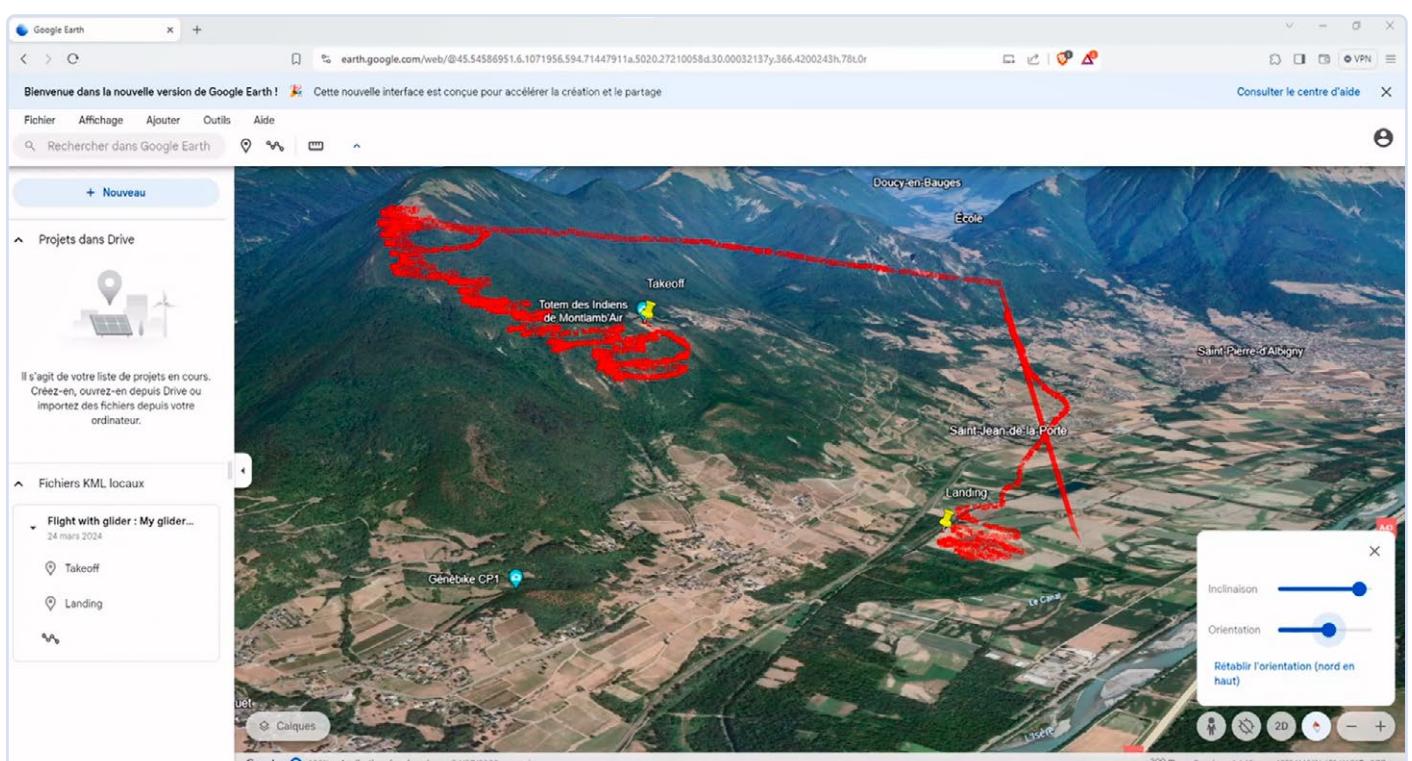


Figure 7. Visualisation 3-D des données de vol (fichier KML) par Google Earth.



Listage 1. Classe d'abstraction de carte

```
/*
 * Copyright (c) 2023 open-vario
 * SPDX-License-Identifier: MIT
 */
#ifndef OV_I_BOARD_H
#define OV_I_BOARD_H
#include "i_accelerometer_sensor.h"
#include "i_barometric_altimeter.h"
#include "i_ble_stack.h"
#include "i_button.h"
#include "i_display.h"
#include "i_gnss.h"
#include "i_serial.h"
#include "i_storage_memory.h"
#include "i_usb_cdc.h"

namespace ov
{

/** @brief Interface for boards implementations */
class i_board
{

public:
    /** @brief Destructor */
    virtual ~i_board() { }

    /** @brief Reset the board */
    virtual void reset() = 0;

    /** @brief Get the debug serial port */
    virtual i_serial& get_debug_port() = 0;

    /** @brief Get the USB CDC port */
    virtual i_usb_cdc& get_usb_cdc() = 0;

    /** @brief Get the storage memory */
    virtual i_storage_memory& get_storage_memory() = 0;

    /** @brief Get the display */
    virtual i_display& get_display() = 0;

    /** @brief Get the 'Previous' button */
    virtual i_button& get_previous_button() = 0;

    /** @brief Get the 'Next' button */
    virtual i_button& get_next_button() = 0;

    /** @brief Get the 'Select' button */
    virtual i_button& get_select_button() = 0;

    /** @brief Get the BLE stack */
    virtual i_ble_stack& get_ble_stack() = 0;

    /** @brief Get the GNSS */
    virtual i_gnss& get_gnss() = 0;
};
```

favorites, ainsi que la découverte d'autres projets excitants, utilisant le même matériel. La limite de temps imposée pour ce concours (merci à l'extension du délai !) était un véritable challenge : chaque projet comporte de nombreuses difficultés à résoudre, mais doit, en finalité, comporter un nombre important de caractéristiques fonctionnelles. Je pense que ma profession d'Ingénieur logiciel, m'a beaucoup aidé sur cet aspect car ce sont les contraintes que je rencontre de façon habituelle.

Merci au magazine Elektor et à ST Microelectronics d'avoir organisé ce concours. J'espère que d'autres compétitions seront organisées dans le futur !

VF : Jean Boyer — 240325-04



À propos de l'auteur

Cédric Jimenez est âgé de 40 ans, il est l'heureux père d'une fillette de 8 ans. Il vit dans les Alpes françaises, où il se livre à de multiples activités de montagne (ascensions, parapente). Cédric est un grand amateur de développement de logiciel pour les systèmes embarqués. Il a débuté la programmation en langage Basic à l'âge de 9 ans sur un TO7/70 Thomson et s'est orienté vers le langage C++ quelques années plus tard.

Bien qu'il ait appris de nombreux langages de programmation (VB, C#, C, Pascal, FORTRAN, Python, Ruby...), C++ demeure son langage de programmation favori.

Il est titulaire d'un Master en Sciences Informatiques, avec une spécialisation en logiciel temps-réel sur systèmes embarqués. Il a exercé pendant plus de 15 ans comme sous-traitant dans des domaines variés (ferroviaire/métropolitain, grues, services médicaux...). Il est actuellement architecte en logiciel intégré chez Schneider Electric.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (cjz.73fr@gmail.com), ou contactez Elektor (redaction@elektor.fr).

```

/** @brief Get the barometric altimeter */
virtual i_barometric_altimeter& get_barometric_altimeter() = 0;

/** @brief Get the accelerometer */
virtual i_accelerometer_sensor& get_accelerometer() = 0;
};

} // namespace ov

#endif // OV_I_BOARD_H

```

LIENS

- [1] Page web du concours STM32 : <https://tinyurl.com/bdenyy24>
- [2] Page web des outils d'évaluation STM32WB5MM-DK : <https://tinyurl.com/4dp83f8h>
- [3] MikroElektronika Arduino Uno Click Shield : <https://www.mikroe.com/arduino-uno-click-shield>
- [4] MikroElektronika GNSS 4 Click Board : <https://www.mikroe.com/gnss-4-click>
- [5] MikroElektronika Altitude 2 Click Board : <https://www.mikroe.com/altitude-2-click>
- [6] Page Github de l'auteur pour le projet Vario : <https://github.com/open-vario/open-vario>
- [7] Ce projet sur Elektor Labs : <https://tinyurl.com/4rmn5j28>



Produits

➤ Majid Pakdel, *Advanced Programming with STM32 Microcontrollers*, Elektor, 2020
www.elektor.fr/19527



Ils nous font confiance, n'est-ce pas ?

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**

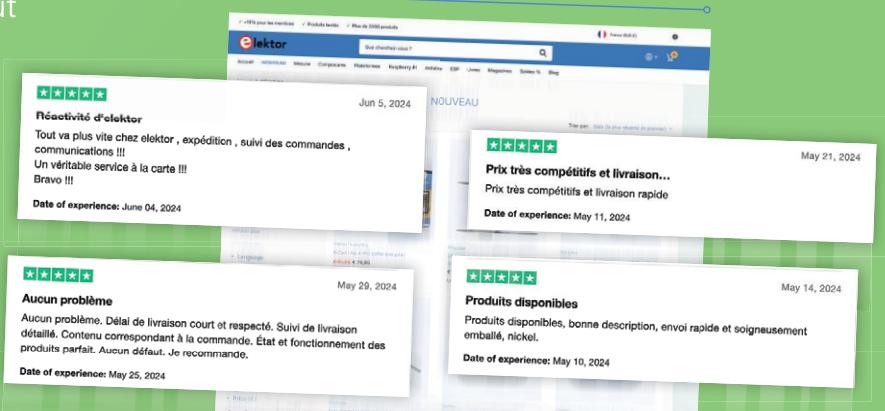
Elektor Store

Reviews 563 • Excellent



4.6

VERIFIED COMPANY



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





sur le vif

À propos de prendre les choses pour acquises

Ilse Joostens (Belgique)

De temps à autre, j'ai besoin d'une bonne conversation sur la technologie et la science, ou une soirée de réflexion avec des gens partageant les mêmes idées, sur de nouvelles idées et de nouveaux projets. Dans la vie quotidienne, cependant, la plupart des conversations sont — malheureusement — orientées vers les maladies effrayantes et leurs traitements médicaux, le temps qu'il fait, les problèmes personnels, le sport, les programmes TV, les festivals et aujourd'hui, de plus en plus sur des choses impalpables comme l'astrologie, le shamanisme ou les soi-disant dangers de la radiation 5G [1]. Pourtant, ces gens utilisent tous les jours de l'électronique avancée sans y penser, qu'ils soient néophytes au niveau technique ou farfelu.

De nombreuses (r)évolutions technologiques sont devenues si naturelles que nous ne pensons plus à la vie d'avant lorsque cette technologie n'était pas présente.

Figure 1. Frustration au volant (Source : Adobe Stock / EdNurg).

Lecture d'une carte

Beaucoup d'entre nous utilisent un système de navigation par satellite, couramment connu sous le nom de « Sat-Nav ». Tout au début, ces appareils étaient encombrants avec une antenne proéminente [2]. Aujourd'hui, ces systèmes sont incorporés dans les voitures modernes et vous pouvez également utiliser des applications gratuites dans votre smartphone. De plus, vous avez accès à des informations en temps réel extrêmement utiles telles que les travaux sur les routes, les accidents, les embouteillages, les limites de vitesse, les conditions météorologiques, les contrôles routiers, les radars de vitesse, etc. Ma seconde moitié a toujours été amoureuse des systèmes de navigation et j'ai perdu le compte du nombre de fois où on s'est fourvoyé dans la mauvaise direction, entraînant une perte de temps et du stress, mais je me souviens très bien comment ça se passait. À l'époque, vous aviez ces immenses cartes routières pliables et encombrantes qui étaient sympas pour établir la route à suivre sur la table de la salle à manger, mais qui étaient beaucoup plus frustrantes derrière le volant de la voiture (**figure 1**). Pour améliorer cela, il y avait aussi ces livres de cartes au format plus pratique. Et il fallait tourner les pages



Figure 2. IA Génératif (Source : Ilse Joostens).

d'avant en arrière et cette sortie importante, ce virage ou même votre destination finale se retrouvait immanquablement juste sur la transition d'une autre page — 30 pages plus loin dans le livre.

Dans le même style de réflexion, Google Maps et Google Street View [3] sont devenus des outils communs et j'aime particulièrement Street View pour explorer différents endroits dans le monde pour me changer les idées sans avoir la nécessité de voyager. C'est quand même plus amusant que ces interminables montages diapos de photos de vacances avec leurs commentaires d'un autre temps. J'utilise également Google Maps pour vérifier les adresses de mes clients et plus d'une fois, je suis passée à Street View juste par curiosité.

Le Majordome l'a fait

L'utilisation d'internet n'était pas si aisée quand on devait se connecter avec un de ces modems couinants et crachotants, mais aujourd'hui, nous sommes tout le temps et partout en ligne sans même s'en rendre compte. Envoyer des messages, des photos et des vidéos et même s'appeler en visio, est devenu la normalité. Il n'y a pas si longtemps, avant les SMS, qui veut dire messages courts, on n'avait que les cartes postales, les télégrammes [4] et ensuite les bipeurs si on était pressé.

Je suis ce qu'on appelle une « coupeuse de fil », au sens propre du mot lorsque je suis occupée avec mon massicot, et au sens figuré du mot. La télévision linéaire n'est plus mon truc de toute façon et malgré les dizaines de chaînes, il n'y a en général rien à la télé qui puisse me captiver, sans compter les pénibles coupures de pub. Donnez-moi des services de diffusion en continu pour pouvoir regarder des films et des séries, assis tranquillement dans mon fauteuil. Avant, il fallait aller au vidéo club par tous les temps, en risquant une amende si vous ramenez la cassette vidéo en retard

ou si vous aviez oublié de la rembobiner. Cela avait son charme, particulièrement dans ce coin interdit et mystérieux derrière le rideau. Dans un article précédent, je parlais d'intelligence artificielle et son évolution a été bien rapide depuis. En même temps, j'ai également commencé à utiliser l'IA générative de manière plus intensive (figure 2), y compris ChatGPT, pour lequel j'ai maintenant plus d'applications. J'estime que dans un futur proche, l'IA deviendra aussi banale que le rôle de James, votre assistant personnel, conseiller et majordome. De plus, cela peut être utile pour échapper à vos responsabilités si un jour, les choses tournent mal. Néanmoins, je crains toujours la mauvaise utilisation de l'IA par les gouvernements tels que la police suédoise qui veut déployer des caméras à reconnaissance faciale [5].

Les concepteurs en électronique sont gâtés

En électronique, beaucoup de choses ont aussi changé en mieux ces dernières décennies. La conception des circuits imprimés n'a jamais été aussi facile qu'elle ne l'est aujourd'hui avec des logiciels plutôt abordables et même parfois gratuits. J'ai travaillé pendant des années avec KiCad [6] à mon entière satisfaction. Même la fabrication des circuits imprimés est un jeu d'enfant avec des fabricants de PCB bon marché dans

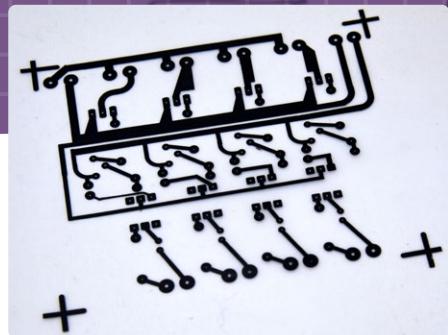


Figure 3. Symboles à gratter sur papier calque (Source : Ilse Joostens).

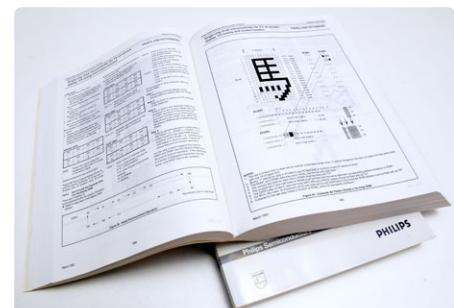


Figure 4. Data books avec une compilation de fiches techniques (Source : Ilse Joostens).

le monde entier. Autrefois, on devait perdre son temps avec des symboles à gratter sur du papier calque (figure 3) ou, si on était chanceux, avec un logiciel onéreux qui demandait un apprentissage long et difficile. Ensuite on n'avait toujours pas de circuit imprimé et on devait le graver soi-même. Je ne vous fais pas un dessin quand je vous dis qu'il y a plus de 30 ans, mes parents n'étaient pas très heureux lorsque mon bac de gravure bricolé, a soudain commencé à fuir et qu'un jet d'acide brun a aspergé le papier-peint. C'était aussi l'ère des épais « data books » (figure 4) et des tableaux d'équivalence, lorsque vous arriviez à les trouver. Aujourd'hui, il est évident de simplement télécharger les fiches techniques depuis l'internet. Les outils avancés de conception et les cartes d'évaluation avec toute une panoplie de processeurs puissants, de capteurs, de modules sans-fil et d'autres gadgets amusants sont maintenant devenus facilement accessibles. Pour moi, le temps des portes logiques et d'effacement d'EPROMs sous la lampe UV n'a pas besoin de revenir. ▶

VF : Chris Elsass — 240319-04

LIENS

- [1] SKEPP: Weird devices [en néerlandais] : <https://skepp.be/nl/rare-apparaten/alfabetisch>
- [2] A Brief History of GPS In-Car Navigation : <https://ndrive.com/brief-history-gps-car-navigation>
- [3] CNET: Crazy images caught on Google Street View : <https://cnet.com/pictures/crazy-images-caught-on-google-street-view>
- [4] VRT nws: Proximus to stop telegram service : https://vrt.be/vrtnws/en/2017/12/12/proximus_to_stoptelegramservice-1-3113214/
- [5] Swedish Government to Move Forward with Police FRT Amid Gang Violence : <https://findbiometrics.com/swedish-government-to-move-forward-with-police-frt-amid-gang-violence>
- [6] KiCad EDA – suite logicielle open-source pour l'automatisation de la conception électronique : <https://kicad.org>

relevé des compteurs d'eau basé sur l'IA

Partie 2 : 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 le remplacement de nos anciens compteurs n'est souvent pas si facile, pour des raisons techniques ou administratives.

Ce projet permet de transformer n'importe quel compteur analogique en compteur numérique à l'aide d'une plateforme ESP32-CAM, et d'un système d'intelligence artificielle (IA). Dans ce dernier épisode, nous verrons comment installer le micrologiciel et configurer l'objectif de la caméra, pour une mise au point et un positionnement parfaits sur le compteur d'eau. Et bien sûr, nous abordons l'ensemble du processus, basé sur l'IA, pour la reconnaissance et la lecture correctes des éléments du compteur.

Installation du micrologiciel

Dans un premier temps, vous devez télécharger le micrologiciel précompilé à partir du projet GitHub « AI on the Edge Device ». De nombreux développeurs contribuent à ce projet assez vaste et performant, qui est très largement et bien documenté. Plusieurs versions du firmware sont disponibles sur la page *Releases* [1], chacune étant accompagnée d'un changelog, un fichier texte indiquant les changements, les nouvelles implémentations ou les corrections de bugs effectuées. Nous allons ensuite télécharger le fichier .zip contenant le micrologiciel précompilé. A l'intérieur de ce fichier .zip, nous trouverons trois fichiers avec une extension .bin, qui représentent le firmware à installer.

Elettronica In
www.elettronicain.it

En plus, un autre fichier .zip renferme le contenu qui doit être téléchargé sur la carte SD. Commençons par préparer notre carte SD, dont la capacité ne doit pas dépasser 16 Go. En effet, des erreurs de lecture peuvent se produire avec des cartes de plus grande capacité, comme les cartes de 32 ou 64 Go. Au niveau logiciel, la carte SD ne doit pas avoir de table de partitionnement et doit être formatée au format FAT32. Si notre module ESP32-CAM rencontre des problèmes ou des dysfonctionnements au démarrage, il est conseillé de remplacer immédiatement la carte SD avant de continuer plus loin.

Une fois que nous avons extrait le fichier téléchargé, nous trouverons plusieurs dossiers à l'intérieur. Le dossier *config*, qui contient les fichiers TensorFlow, et le fichier de configuration appelé *config.ini*, nous intéressent particulièrement ici. Dans le fichier *config.ini*, nous trouverons tous les paramètres initiaux, qui peuvent être difficiles à lire ou à interpréter pour ceux qui n'ont pas directement compilé le firmware. Pourtant, parmi la longue liste de paramètres présents, nous sommes particulièrement intéressés par l'entrée *SetupMode = true*. Nous allons changer la valeur de ce paramètre de *true* à *false* pour désactiver le mode de configuration initial. Ce paramètre est responsable de ce que nous verrons dans notre navigateur, lorsque nous y accéderons pour la première fois. En gardant la valeur *true*, nous afficherons un assistant de configuration qui peut être inutile, et pouvant aussi entraîner des problèmes d'écriture sur la carte SD. En changeant la valeur du paramètre à *false*, nous accéderons directement au panneau principal, avec lequel nous devrons nous familiariser.

Il existe un deuxième fichier que nous devons éditer afin d'améliorer le démarrage initial, et c'est *wlan.ini*. Dans ce fichier, en effet, nous devons entrer le SSID et le mot de passe de notre réseau Wi-Fi. Si nous ne donnons pas ces identifiants de connexion, la puce démarera en mode AP (Access Point), montrant son réseau aux clients. À ce stade, nous pouvons procéder à l'installation du firmware proprement dit, et il y a deux manières possibles. Soit à travers l'outil *Flash Tool* d'Espressif, ou à partir de la console avec *esptool*.

Dans l'ordre, en commençant par la procédure la plus simple, nous téléchargeons le logiciel Flash Download Tools à l'adresse [2]. Il est décompressé dans un dossier et nous exécutons le programme .exe qui s'y trouve. Nous sélectionnons *ESP32* et *Develop* (Figure 1) pour

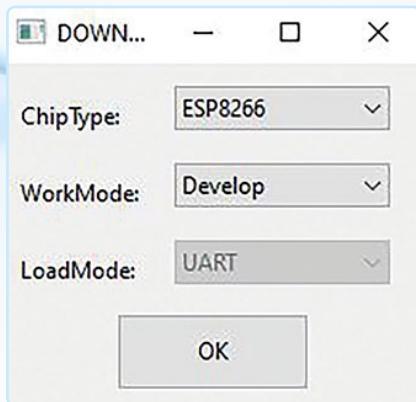


Figure 1. Sélection de ESP32 et de Develop pour charger le fichier .bin dans l'ESP32-CAM.

programmer le fichier `.bin`. Nous sélectionnons ensuite les fichiers, dans l'ordre décrit ci-dessous, et entrons leurs index respectifs à côté :

- `bootloader.bin` 0x1000
- `partitions.bin` 0x8000
- `firmware.bin` 0x10000

La vitesse du bus SPI doit être réglée sur 40 MHz et le mode (*MODE*) sur DIO (**figure 2**). Nous pouvons maintenant connecter notre programmeur, choisi d'après les configurations précédentes, et appuyer sur *START*. Après quelques instants, le micrologiciel sera correctement chargé.

Le second mode de chargement nécessite l'utilisation du terminal et d'un environnement avec Python. Par exemple, on peut utiliser Anaconda [3], mais nous recommandons cette procédure principalement pour les utilisateurs expérimentés d'Ubuntu. Dans un premier temps, il faut installer l'outil `esptool`. Pour ce faire, nous devons taper la commande suivante dans la ligne de commande :

```
pip install esptool
```

Si vous utilisez Python 3, vous devez utiliser la commande :

```
pip3 install esptool
```

Branchons maintenant notre programmeur et entrons la commande :

```
esptool erase_flash
esptool write_flash 0x01000 bootloader.bin 0x08000
partitions.bin 0x10000 firmware.bin
```

Bien entendu, en tant qu'utilisateur expérimenté, vous devez vérifier si COM est spécifiée automatiquement ou non. Si ce n'est pas le cas, des problèmes peuvent survenir dès la première instruction. Après avoir écrit le firmware, nous pouvons déconnecter le câble USB de notre PC et enlever tout le câblage fait précédemment, en laissant seulement 5 V et GND sur notre ESP32-CAM, comme indiqué.

Après avoir redémarré le module ou l'avoir relié à une alimentation externe, lors du premier démarrage du micrologiciel, nous devons vérifier si le fichier `wlan.ini` est disponible et s'il contient des informations d'identification valides. Si ces données sont manquantes ou invalides, le module démarrera en mode AP, sans mot de passe ni protection pour le réseau Wi-Fi. Et un nouveau réseau Wi-Fi appelé *AI-on-the-Edge* sera créé.

Il suffit de quelques étapes pour télécharger le fichier directement via la page web du module. Mais procérons dans l'ordre, car la première étape consiste à télécharger le fichier `remote-setup` toujours disponible

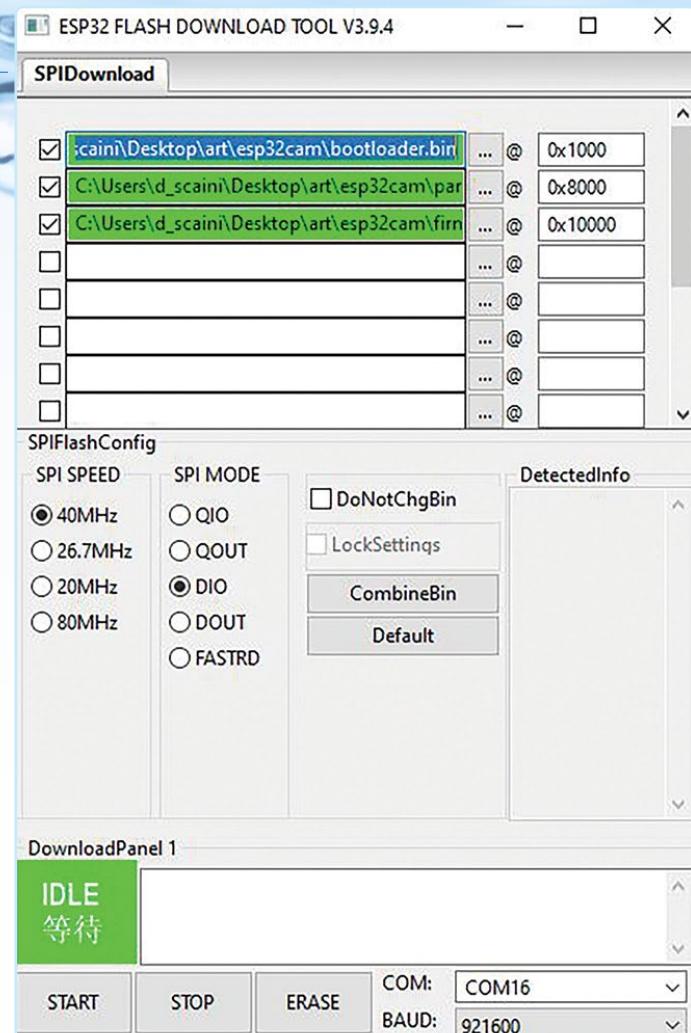


Figure 2. Le champ SPI SPEED doit être réglé sur 40 MHz, tandis que le champ SPI MODE doit être réglé sur DIO.

dans les versions du projet ici [1]. Ensuite, la connexion au réseau Wi-Fi "AI-on-the-Edge" affichera une page nous demandant de télécharger un fichier.

Nous chargeons le fichier que nous venons de télécharger et attendons, comme indiqué sur la page. Après un court téléchargement qui peut durer jusqu'à 60 secondes, l'écran de saisie des informations d'identification du réseau Wi-Fi apparaît. Nous entrons notre SSID et notre mot de passe, puis nous cliquons sur *Write wlan.ini*. Si tout s'est bien passé, le redémarrage sera demandé.

L'ESP32 est équipé d'une LED rouge, dont le clignotement a plusieurs significations :

- 5 clignotements rapides (< 1 seconde) : connexion en cours
- 3 clignotements lents (1 seconde on/off) : connexion WLAN établie

Réglage de l'optique de la caméra

Le réglage de la mise au point de l'objectif de la caméra est un point crucial, après avoir téléchargé le logiciel dans le module ESP32-CAM. Par défaut, la mise au point est réglée pour les très longues distances et non pour les distances courtes, comme cela est nécessaire dans notre cas. Lors de cette étape, il est important de procéder très délicatement, en essayant de faire tourner l'extérieur de l'objectif avec des pincelettes et, si nécessaire, en enlevant la colle placée à la base de l'objectif à l'aide d'un cutter. Ces opérations de réglage doivent être effectuées avec précaution, comme le montre la **figure 3**.

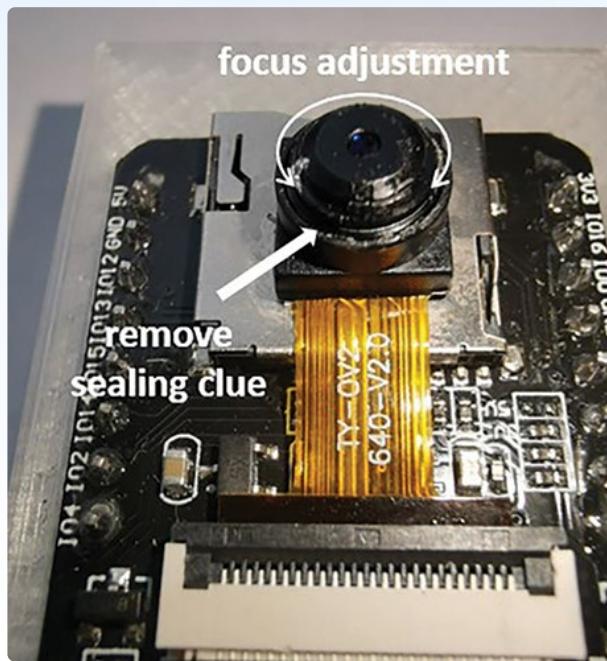


Figure 3. Réglage de l'optique pour la prise de vue rapprochée.

Positionnement de l'ensemble

Après avoir installé le micrologiciel et réglé la mise au point de la caméra, nous pouvons placer notre montage sur le compteur que nous voulons surveiller. Nous devons essayer de placer la caméra, pour rester à une distance suffisante pour lire correctement les données, et pour qu'elle puisse rester immobile pendant toutes les opérations. Pour surmonter cette difficulté, il existe plusieurs solutions. La meilleure serait sans aucun doute d'imprimer en 3D un support en plastique, qui est fermement fixé au compteur, pour abriter notre montage et le maintenir protégé et immobile à l'intérieur d'un petit boîtier. Un exemple tiré de Thingiverse peut être téléchargé à l'adresse [4]. Sur cette page, vous trouverez le fichier STL pour le boîtier du module (figure 4), ainsi

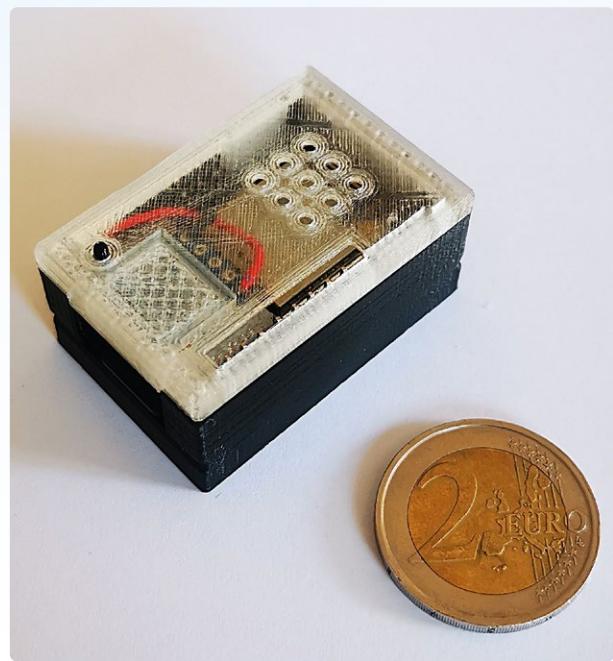


Figure 4. Le module ESP32-CAM dans sa boîte imprimée en 3D.

que l'entretoise qui vous permet de fixer correctement l'électronique sur le compteur, comme illustré dans la figure 5.

Une alternative plus simple et plus rapide consiste à utiliser un cube en nylon ou en polystyrène, couramment utilisé pour l'emballage. Nous plaçons le module à l'extérieur du cube de polystyrène, face vers le bas, et nous créons un trou pour permettre à l'objectif de l'appareil photo et au flash de passer à travers. Le trou central est dimensionné exactement à la taille de l'indicateur. Si l'indicateur possède également un couvercle, comme le montre la photo, nous pouvons modeler davantage le support pour qu'il s'adapte parfaitement.

Premiers réglages et résultats

Avant d'allumer notre système, nous allons retirer la carte SD et l'insérer dans le PC. Vérifions l'intérieur de la carte SD, dans le dossier *config*, le fichier *config.ini*, pour vérifier que nous avons bien suivi toutes les étapes, et que l'installation du firmware s'est déroulée avec succès. À la fin de ce fichier, nous trouverons le paramètre *SetupMode*, en rappelant qu'il doit être réglé sur *false*. Cette étape nous permettra d'accéder directement à l'écran principal et non à l'écran de l'assistant d'installation.

Ensuite, réinsérons la carte SD dans le module et allumons-le. Avec notre PC, connectons-nous à l'adresse que notre routeur aura donnée à l'ESP32, dans notre cas 192.168.43.102. Nous obtenons l'écran de la figure 6. Évidemment, laissons de côté les mesures et l'identification de rectangles ou de photos pour l'instant, et concentrons-nous sur la partie menu située dans la barre noire en haut. Nous allons dans *Settings* et un menu déroulant s'ouvre, affichant quatre éléments : *Set Previous Value*, *Configuration*, *Alignment*, *Region of Interest*.

Commençons par le troisième élément, où à l'intérieur de celui-ci, nous trouvons deux autres rubriques, à savoir *Reference Image* et *Alignment Marker*. En cliquant sur *Reference Image*, cette rubrique nous permet de capturer la première image de notre compteur, et de la conserver comme référence pour toutes les vérifications ultérieures. Pour ce faire, nous cliquons sur *Create New Reference Image* et ensuite



Figure 5. Un adaptateur a été fabriqué avec l'imprimante 3D, pour installer l'ESP32-CAM sur le dessus du compteur.

Digitizer - AI on the edge

An ESP32 all inclusive neural network recognition system for meter digitalization

The screenshot shows a water meter with a digital display showing '00038'. To the right, a table lists the following data:

Raw Value:	038.5975
Corrected Value:	38.5975
Checked Value:	38.5975
Start Time:	20201118-075416
Last Page Refresh:	06:57:39

Figure 6. La page web interne à partir de laquelle le système peut être configuré.

sur *Take Reference*, ce qui enverra l'instruction de prendre la photo. En haut de ce menu, nous pouvons modifier l'intensité de la LED, l'angle de rotation de l'image, le contraste, la saturation et bien d'autres valeurs afin d'optimiser l'affichage de l'échantillon choisi. Une fois la première acquisition terminée, nous cliquons sur *Save* puis, après avoir reçu la notification de sauvegarde, sur *Restart* pour activer la référence sauvegardée. À ce stade, il est important que les numéros du compteur à l'image soient correctement alignés les uns par rapport aux autres. Une fois que le formulaire a redémarré, nous passons à une autre section qui se trouve également dans *Settings/Alignment*, qui est *Alignment Marker* (figure 7). Dans cette section, nous devons faire en sorte que le système reconnaissasse la position exacte de deux marques de référence que nous pouvons choisir dans le menu déroulant *Select Reference*. Le « 0 » contiendra la balise CRI, qui n'est pas toujours présente sur les compteurs (Figure 8). Le second, le « 1 », est l'unité de mesure, c'est-à-dire le mètre cube. Après avoir correctement déplacé la boîte pour centrer les balises, le cas échéant, cliquez sur *Update*

Non sicuro | 192.168.43.102

Digitizer - AI on the edge - watermeter

An ESP32 all inclusive neural network recognition system for meter digitalization

The screenshot shows a dropdown menu under 'Alignment' with the following options:

- Reference Image
- Alignment Marks

Figure 7. Dans le menu *Alignment Marks*, nous pouvons sélectionner les marqueurs de référence.

Reference Image, puis sur *Save* sans redémarrer l'appareil pour l'instant, car nous devons d'abord définir toutes les références.

Dans le menu déroulant, nous sélectionnons maintenant *Settings* puis *Region Of Interest*, et deux autres sous-titres apparaissent, avec *Digital ROIs* et *Analog ROIs*. Nous sélectionnons le premier, ce qui nous amène à l'écran de la figure 9. Ici, nous devons marquer tous les chiffres de l'indicateur à l'aide des outils qui se trouvent au bas de l'écran. Certains des numéros sont déjà identifiés dans le menu déroulant situé sous *New ROI*, mais nous devons saisir X et Y pour chaque numéro. Ne modifiez pas les cases à cocher sur le côté, si la photo a été prise correctement, car les zones qui nous intéressent seront toutes alignées et de taille égale.

Il y a deux cadres de couleur rouge pour identifier le numéro, et le cadre intérieur doit être de la même taille que le chiffre. Une fois cette étape terminée, nous cliquons sur *Save* et passons à la dernière étape. Dans la section *Analog ROIs*, nous devons répéter une opération similaire aux précédentes mais avec les indicateurs analogiques (Figure 10). Le bord rond et rouge du masque d'identification doit correspondre exactement au bord rond de l'indicateur. Là encore, les différentes cases à cocher peuvent être conservées si la photo, c'est-à-dire la référence de départ, est bonne, puisque tous les indicateurs auront la même taille. Après avoir terminé, nous cliquons à nouveau sur *Save* et enfin sur *Reboot* pour valider les changements.

Configure Watermeter

Main Page CONFIG.INI direct Alignment Regions Of Interest (ROI) Check

The screenshot shows a water meter with a digital display showing '00016'. Below the meter, a configuration interface allows defining ROI for the CRI marker:

Reference 0 Storage path/name: /config/ref0.jpg
X: 119 dx: 57
Y: 273 dy: 31
Update Reference
Original Image Reference Image
CRI CRI Enhance Contrast
Save to Config.ini

Figure 8. Le marquage CRI n'est pas toujours présent sur les compteurs d'eau.

Digitizer - AI on the edge - watermeter

An ESP32 all inclusive neural network recognition system for meter digitalization

The screenshot shows a water meter with ROI boxes drawn on it. The interface includes a table for managing ROIs:

Number:	main	Rename	New	Remove
New ROI (after current)	Delete ROI			

Figure 9. Marquage des chiffres du compteur.

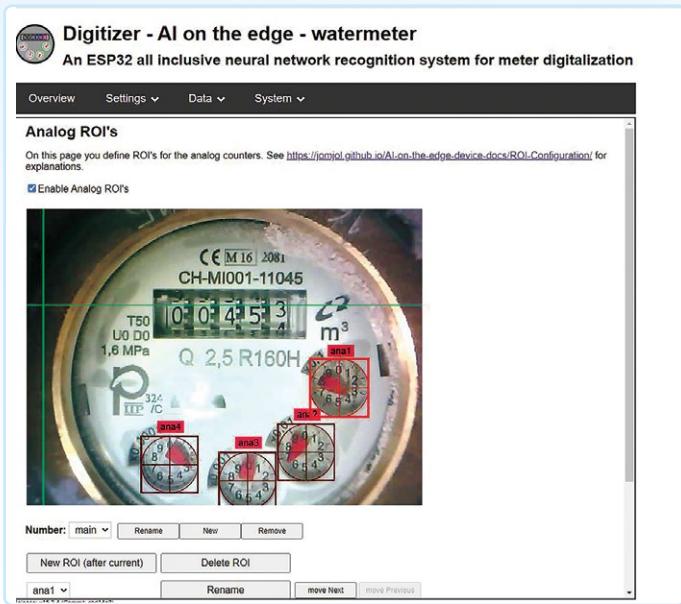


Figure 10. Marquage des aiguilles du compteur.

Configuration REST, MQTT et intégration de Home Assistant

Maintenant que notre module ESP32-CAM peut lire notre compteur, nous avons besoin d'un moyen pratique pour afficher les données. Heureusement, le code est bien écrit, et nous permet d'accéder aux données de plusieurs manières. Dans le firmware précompilé se trouve un serveur http, que nous avons déjà exploité lors de la mise au point des premiers paramètres.

Ce serveur nous fournit non seulement des pages web mais aussi des API REST, utilisables entièrement par la méthode GET. La parti-

cularité de cette méthode réside dans sa simplicité d'utilisation, en effet toutes les informations se trouvent dans l'URL et en texte clair. La partie fixe de ces appels est donnée par le point d'entrée `http://IP_DEVICE`, auquel il faut ajouter l'une des parties variables suivantes, en fonction de la finalité recherchée.

Le **tableau 1** présente les commandes qui peuvent être utilisées ainsi que leur signification. Toutes ces commandes sont très pratiques lorsque nous nous trouvons à l'intérieur du même réseau. Mais qu'en est-il si nous nous trouvons sur un réseau externe ?

L'intégration du service MQTT répond à cette question. **Message Queue Telemetry Transport** (MQTT) est un protocole de messagerie très léger, de type publication-abonnement, utilisé lorsque la bande passante est limitée ou que l'on souhaite créer un impact minime au niveau du réseau. Si l'on se réfère à la littérature, ce protocole, créé par Andy Stanford-Clark et Arlen Nipper en 1999, a d'abord été adapté à la surveillance des oléoducs.

L'objectif était de créer un protocole vraiment léger, efficace en termes de bande passante et consommant peu d'énergie électrique, un sujet épique à l'époque comme aujourd'hui. Cependant, ce protocole a connu une nouvelle vie en 2013 lorsque la version 3.1 a été publiée par IBM et a commencé à s'imposer, en partie grâce à une forte poussée du marché vers l'Internet des objets. L'un des exemples les plus marquants de l'utilisation de MQTT est certainement Facebook Messenger, car en effet, Facebook a déclaré avoir adopté ce protocole pour améliorer la messagerie et limiter l'utilisation du réseau et de la consommation électrique.

Le client est un système quelconque, allant d'un simple microcontrôleur à un serveur complet, qui exécute une bibliothèque MQTT et est connecté à un courtier (broker). Cet acteur envoie des informations au courtier, appelées *payloads*, classées en fonction de leur sujet. L'information, en fait, est organisée en fonction du sujet, appelé *topic*. Lorsqu'un client envoie un nouveau *payload* à un courtier, celui-ci relaie l'information à tous les clients qui lui sont connectés sur un

Tableau 1. Commandes pouvant être envoyées à partir du point d'entrée `http://IP_DEVICE`.

Chemin	Fonctionnalité	Report/Réponse
PATH_FISSO/GPIO	Fournit une interface pour activer ou connaître l'état d'une broche de notre ESP	Pour définir un statut <code>?GPIO=&Status=high or low</code> Pour lire un statut <code>?GPIO=</code>
PATH_FISSO/reboot	Pour redémarrer le système	
PATH_FISSO/json	Pour obtenir les résultats au format JSON de la dernière lecture effectuée	Exemple de réponse : <pre>{ "main": { "value": "521.17108", "raw": "521.17108", "pre": "521.17108", "error": "no error", "rate": "0.023780", "timestamp": "2023-01-13T16:00:42+0100" } }</pre>

Chemin	Fonctionnalité	Report/Réponse
PATH_FISSO/value	Pour obtenir les résultats sous la forme d'une valeur unique et non au format JSON	Pour extraire une valeur autre que la valeur par défaut nous pouvons utiliser les paramètres GET suivants : <code>?all=true&type=value</code> <code>?all=true&type=raw</code> <code>?all=true&type=error</code>
PATH_FISSO/img_tmp/raw.jpg	Pour obtenir une image brute du compteur	
PATH_FISSO/img_tmp/alg_roi.jpg	Pour obtenir une image brute avec les différentes zones d'intérêt (ROI) en superposition	
PATH_FISSO/statusflow	Il indique exactement quel est le processus actif sur l'appareil à ce moment-là, y compris un horodatage	Un exemple de réponse pourrait être : <code>Take Image (15:56:34)</code>
PATH_FISSO/rssi	Indique la puissance du signal WiFi reçu (Unité : dBm)	Voici un exemple de réponse : <code>-51</code>
PATH_FISSO/cpu_temperature	Affiche la température de la CPU (Unités : °C)	
PATH_FISSO/sysinfo	Affiche toutes les informations du au format JSON	Un exemple de réponse peut être : <code>[{"firmware": "", "buildtime": "2023-01-25 12:41", "gitbranch": "HEAD", "gittag": "", "gitrevision": "af13c68+", "html": "development-branch: HEAD (Commit: af13c68+)", "cputemp": "64", "hostname": "WaterMeterTest", "IPv4": "192.168.43.102", "freeHeapMem": "2818330"}]</code>
PATH_FISSO/startime	Indique la « date de mise en service »	Exemple de réponse : <code>20230113-154634</code>
PATH_FISSO/uptime	Indique depuis combien de temps il est actif	Voici un exemple de réponse : <code>0d 00h 15m 50s</code>
PATH_FISSO/lighton	Modifie l'état du flash de l'appareil photo en le réglant sur ON	
PATH_FISSO/lightoff	Modifie l'état du flash de l'appareil photo en le réglant sur OFF	
PATH_FISSO/capture	Capture une nouvelle image (sans flash)	
PATH_FISSO/capture_with_flashlight	Capture une nouvelle image (avec flash)	
PATH_FISSO/save	Enregistre une nouvelle image sur la carte SD	Vous avez en plus les paramètres suivants : <code>?filename=test.jpg&delay=1000</code>
PATH_FISSO/log	N'affiche que la dernière partie des données de la journée (seulement les derniers 80 kB)	
PATH_FISSO/logfileact	Affiche tous les relevés de la journée	

sujet donné. Le client n'a aucune idée du nombre ou de la position des abonnés (*subscribers*) connectés à ce sujet (*topic*).

Dans notre cas, le client est la puce elle-même, tandis que le courtier peut être une installation Hassio/Home Assistant. Pour activer ce service et son intégration, nous allons dans *Settings*, *Configuration*. En bas de la page, nous trouvons la section sur la configuration de MQTT, que nous pouvons éditer comme dans la **Figure 11**. Dans les URI (identifiant de ressource de réseau), nous devons entrer `mqtt://BROKER ADDRESS:PORT` (habituellement 1883) et, au cas où nous

en aurions besoin, le nom d'utilisateur et le mot de passe.

Après avoir saisi ces deux dernières valeurs, allons dans le tableau de bord de Home Assistant et là, sous *Settings* → *People*, ajoutons un nouvel utilisateur qui peut se connecter avec les mêmes informations d'identification. Dans notre tableau de bord Home Assistant, allons dans *File Editor*, un plugin hautement recommandé qui permet d'éditer des fichiers de configuration directement depuis l'interface web. Nous trouvons ici le fichier `/config/configuration.yaml`, et nous ajoutons les lignes suivante :

MQTT

<input checked="" type="checkbox"/> URI	mqtt://	
<input checked="" type="checkbox"/> Main Topic	watermeter	
<input checked="" type="checkbox"/> Client ID	watermeter	
<input checked="" type="checkbox"/> Username		
<input checked="" type="checkbox"/> Password		
MQTT Retain Flag	false	

Homeassistant Discovery (using MQTT)
The discovery topics and the static topics (IP, MAC, Hostname, Interval, ...) only get sent on startup. To send them again, you can call the following URL: http://<IP>/mqtt_publish_discovery

Homeassistant Discovery	false	
--------------------------------	-------	--

Figure 11. Menu de configuration MQTT.

```
sensor:
- platform: mqtt
  name: "waterMeterMQTT"
  state_topic: "watermeter/main/raw"
  unique_id: watermeter_value
  unit_of_measurement: 'm³'
  state_class: total_increasing
  device_class: water # Needs HA 2022.11!
  icon: 'mdi:water-pump'
  availability_topic: watermeter/connection
  payload_available: connected
  payload_not_available: connection lost
```

Comme indiqué, si vous utilisez une ancienne version de Home Assistant, il se peut que vous n'ayez pas l'entité « water » et vous pouvez facilement utiliser « power » à la place. De même, au lieu d'utiliser « raw », vous pouvez utiliser « value » qui fournit l'information déjà traitée, au lieu de la forme brute.

Vous devez maintenant redémarrer le serveur. Allez dans *Settings*, *Server Controls* et sélectionnez *restart* pour afficher notre nouvelle

entité. Une fois la connexion avec le serveur rétablie, cliquez sur *Edit dashboard* et ajoutez notre capteur comme le montre la **Figure 12**. Pour plus de détails, nous listons toutes les rubriques à utiliser dans le **Tableau 2**. Bien entendu, prenons comme exemple le fait que vous avez défini le sujet principal (main topic) comme étant le compteur d'eau (watermeter) pendant la phase de configuration.

Développement et modifications

Vous pouvez également modifier les sources afin de changer, par exemple, le comportement du serveur web. Dans un premier temps, vous devrez installer un client GIT. Pour ceux qui ne savent pas de quoi nous parlons, GIT est un système de gestion de versions extrêmement répandu. Pour cloner le projet à partir de la source distante :

```
git clone https://github.com/jomjol/AI-on-the-edge-device.git
cd AI-on-the-edge-device
git checkout rolling
git submodule update --init
```

Sensore Configurazione Scheda

Singola Entità (richiesto)	sensor.watermeter	
Nome (facoltativo)		
Icona (facoltativo)	mdi:water-pump	Tipo di grafico (facoltativo)
		line
Unità (facoltativo)	<input checked="" type="radio"/>	Mostra ulteriori dettagli
Tema (facoltativo)	Ore da mostrare (facoltativo)	
	24	
MOSTRA EDITOR DI CODICE ANNULLA SALVA		

Figure 12. Fenêtre de configuration du capteur.

Tableau 2. Thèmes à utiliser avec MQTT.

Topic	Description
<code>watermeter/MAC</code>	Pour obtenir l'adresse MAC de l'ESP
<code>watermeter/IP</code>	Pour obtenir l'adresse IP de l'ESP
<code>watermeter/Hostname</code>	Pour obtenir le nom d'hôte
<code>watermeter/Interval</code>	Pour déterminer l'intervalle de temps entre deux prises de vues
<code>watermeter/Connection</code>	Pour connaître l'état de la connexion
<code>watermeter/Uptime</code>	Pour savoir combien de temps notre CPU a été active depuis le dernier redémarrage
<code>watermeter/FreeMem</code>	Pour connaître l'espace libre restant (en Ko)
<code>watermeter/wifiRSSI</code>	Pour connaître la qualité du signal
<code>watermeter/CPUTemp</code>	Pour connaître la température de la CPU
<code>watermeter/main/error</code>	Informe de l'état des erreurs de flux. En cas d'absence, il répondra « Pas d'erreur » (la réponse principale dépend de ce que nous avons défini).
<code>watermeter/main/raw</code>	Donne la valeur de la lecture avant le processus de post-acquisition
<code>watermeter/main/rate</code>	Donne la valeur de la consommation par minute
<code>watermeter/main/rate_per_time_unit</code>	Donne la valeur de la consommation par heure
<code>watermeter/main/changeabsolut</code>	Différence entre la valeur lue précédente et la valeur actuelle
<code>watermeter/main/timestamp</code>	Pour connaître l'horodatage de l'ESP
<code>watermeter/main>Status</code>	Pour connaître l'état de notre programme
<code>watermeter/main/json</code>	Connaître un ensemble de valeurs au format json comprenant les valeurs, valeur brute, l'erreur, le débit, l'horodatage, etc.
<code>watermeter/GPIO/GPIO</code>	Pour attester et connaître l'état d'une broche GPIO
<code>watermeter/main/value</code>	Fournit la valeur de la lecture après le processus de post-acquisition

Une fois que vous avez installé le client pour extraire le référentiel, nous vous recommandons également d'installer un environnement de développement intégré (IDE) pour obtenir une vue complète du projet. Comme IDE, nous recommandons d'utiliser Visual Studio Code, qui offre également PlatformIO, un plugin essentiel. Nous commençons par installer Visual Studio Code, en le téléchargeant sur le site web dédié [5]. Ensuite, nous ouvrons les extensions (*Extensions*) dans Visual Studio Code et recherchons *platformio*. Nous installons l'extension via le bouton approprié.

Dans notre environnement de développement, nous ouvrons le code source qui se trouve dans le dossier *AI-on-the-edge-device/code* folder. Nous ouvrons ensuite un terminal PlatformIO en cliquant sur l'icône d'extension, qui représente un extraterrestre, et en sélectionnant *New terminal*. Dans le terminal PlatformIO, pour compiler le code, nous tapons `platformio run --environment esp32cam`. Dans le dossier *AI-on-the-edge-device/code/pio/build/esp32cam/* se trouvent les trois fichiers bin compilés : *bootloader.bin*, *partitions.bin* et *firmware.bin*. Tout le code est écrit en langage C++.

Évolution future

Le code est bien documenté et structuré, et fait l'objet d'une maintenance constante de la part des développeurs. Un développement futur possible serait d'envoyer la valeur lue à son téléphone portable à la fin de chaque mois, par exemple par le moyen d'un courrier électronique ou des réseaux sociaux. Une autre idée pour améliorer le projet pourrait être l'intégration du BLE (*Bluetooth Low Energy*). Cela permettrait de créer une plateforme où la lecture peut se faire via le smartphone, sans avoir à utiliser des périphériques externes supplémentaires. 

VF : Laurent Rauber — 240213-B-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).



Produits

➤ **Carte de développement ESP32-Cam-CH340**
www.elektor.fr/products/esp32-cam-ch340-development-board

➤ **Câble USB série TTL RS232 FTDI**
www.elektor.fr/products/ftdi-serial-ttl-rs232-usb-cable

LIENS

- [1] Projet GitHub, "AI on the Edge Device", page des publications : <https://tinyurl.com/2n8da7jj>
- [2] Espressif Flash tools : <https://espressif.com/en/support/download/other-tools>
- [3] Anaconda page de téléchargement : <https://anaconda.com/download>
- [4] Boîtiers imprimables en 3D sur Thingiverse : <https://thingiverse.com/thing:4573481>
- [5] Site de téléchargement de Visual Studio Code : <https://code.visualstudio.com/Download>
- [6] Première partie de cet article :
www.elektormagazine.fr/articles/releve-des-compteurs-d-eau-base-sur-l-ia-partie-1-integrez-votre-ancien-compteur-dans-l-ido

agriculture intelligente

détection des nuisibles basée sur l'apprentissage machine avec connectivité IdO



Rafik Mitry (Mouser) et Shashank Hedge (Würth Elektronik eiSos)

La combinaison de systèmes de vision basés sur l'apprentissage machine (machine learning, ML) en périphérie de réseau et d'une connectivité sans fil ouvre des perspectives entièrement nouvelles en agriculture. Cet article montre comment l'Internet des Objets (IdO, anglais IoT) peut être utilisé pour réaliser un système de détection de nuisibles.

Dans le domaine de l'agriculture intelligente, la détection des nuisibles était traditionnellement basée sur des méthodes grosses consommatrices de main d'œuvre, laissant les cultures vulnérables aux dégâts et les agriculteurs face à des rendements imprévisibles. Avec l'émergence de l'IdO cellulaire et de l'informatique en périphérie de réseau (*edge computing*), un changement important devient possible. Semblable à la maintenance prédictive dans les environnements industriels, l'IdO cellulaire permet de surveiller en temps réel l'état des cultures à l'aide de capteurs. De plus, grâce à la vision

industrielle sur les appareils en périphérie, il est désormais possible d'analyser les données localement, donc de doter les appareils alimentés par batterie de capacités de surveillance en temps réel. Cela permet d'intervenir à temps pour prévenir les infestations de ravageurs, maximiser la productivité agricole et minimiser les pertes. Cet article explore comment l'IdO cellulaire et l'informatique en périphérie de réseau transforment la détection des nuisibles, depuis la collecte de données jusqu'à l'analyse prédictive, ouvrant ainsi la voie à une nouvelle ère d'agriculture de précision.



Figure 1. Carte de développement Wireless FeatherWing Thyone-I de Würth Elektronik dotée d'un module RF propriétaire de 2,4 GHz.
(Source : Würth Elektronik eiSos)

Éléments de prototypage

Les éléments suivants sont utilisés pour réaliser un prototype de détection de nuisibles :

- Arduino Nicla Vision [1]
- Würth Elektronik Thyone-I FeatherWing [2]
- Würth Elektronik Adrastea-I FeatherWing [3]
- Adafruit Feather M0 Express [4]

Logiciels et outils :

- L'EDI Arduino
- OpenMV [5]
- Edge Impulse [6]
- Visual Studio Code avec extension PlatformIO [7]



Figure 2. Le Würth Elektronik Adrastea-I FeatherWing doté de la capacité LTE-M/NB-IoT, d'antennes LTE et GNSS externes.
(Source : Würth Elektronik eiSos)

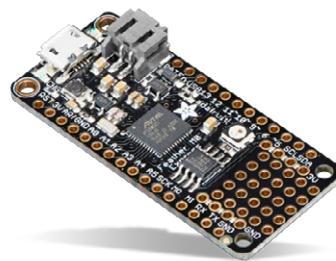


Figure 3. La carte MCU Adafruit Feather M0 Express basée sur le processeur ATSAMD21G18 Arm Cortex-M0.
(Source : Adafruit)

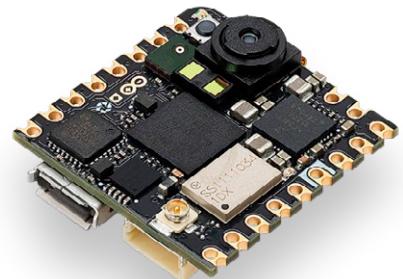


Figure 4. L'Arduino Nicla Vision est équipé d'un processeur robuste STM32H747AI16 Dual Arm Cortex M7/M4 et d'une caméra couleur 2 MP, permettant la prise en charge de TinyML.
(Source : Mouser Electronics)

Le Würth Elektronik Thyone-I Wireless FeatherWing, doté d'un module RF propriétaire de 2,4 GHz, offre une connectivité sans fil transparente avec une portée impressionnante allant jusqu'à 300 mètres (Figure 1). Grâce au chiffrement AES128 intégré et aux capacités de réseau maillé, il garantit une sécurité robuste et une communication fiable. De plus, lorsqu'il est associé à un autre appareil Thyone, il facilite le remplacement progressif et sécurisé du réseau filaire lorsqu'il est configuré en mode transparent.

La carte Adrastea-I FeatherWing est dotée de la capacité LTE-M/NB-IoT, d'antennes LTE et GNSS externes et d'un port USB (Figure 2). Ses fonctionnalités polyvalentes incluent une sélection souple de mode entre LTE-M et NB-IoT, un GNSS intégré prenant en charge GPS et GLONASS, un MCU ARM Cortex-M4 et une compatibilité avec divers protocoles et bandes, le tout dans le format pratique Adafruit Feather, garantissant une extension facile grâce aux modules Feather existants.

L'Adafruit Feather M0 Express est une carte MCU Feather basée sur le processeur Arm Cortex -M0 ATSAMD21G18, cadencé à 48 MHz et fonctionnant sous 3,3 V, le même que dans l'Arduino Zero (Figure 3). Cette puce dispose de 256 Ko de mémoire Flash (8 fois plus que l'Atmega328 ou 32u4) et de 32 Ko de RAM (16 fois plus). Son port USB intégré, converti en série, sert à la programmation et au débogage.

L'Arduino Nicla Vision est équipé d'un processeur robuste STM32H747AI16 Dual Arm Cortex M7/M4 et d'une caméra couleur 2 MP, permettant la prise en charge

de TinyML (Figure 4). De plus, il dispose d'un capteur de mouvement intelligent à 6 axes, d'un microphone intégré et d'un capteur de distance. Conçue pour une compatibilité transparente avec tous les produits Arduino Portenta et MKR, la carte s'intègre également entièrement à OpenMV, prend en charge MicroPython et offre une connectivité WiFi et Bluetooth Low Energy.

Edge Impulse est un environnement de développement pour l'apprentissage machine sur les appareils embarqués. Il permet de collecter des données de capteurs en environnement réel, de développer un algorithme d'apprentissage automatique et de déployer facilement le modèle sur n'importe quel appareil de périphérie de réseau.

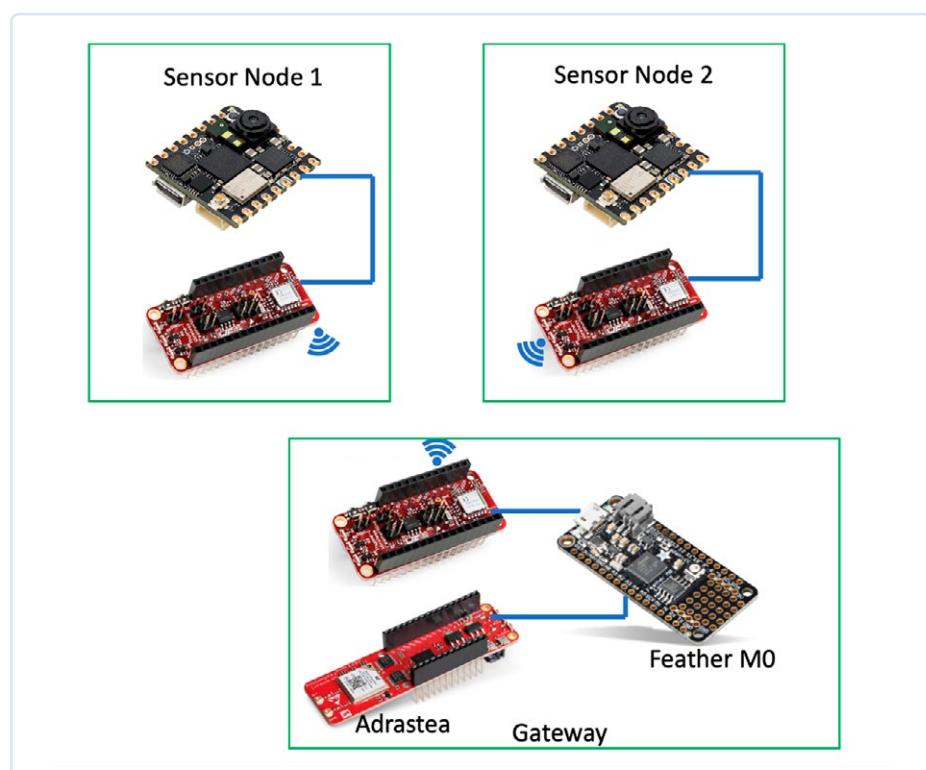


Figure 5. Architecture du prototype de détection de nuisances avec plusieurs nœuds de capteurs et une passerelle. (Source : Mouser Electronics)

Configuration du prototypage

La **figure 5** montre l'architecture du prototype. Le Nicla Vision est l'organe de vision du système de détection des nuisibles, chargé d'exécuter le modèle ML. Il est connecté à une carte Thyone, facilitant la transmission des données vers le nœud principal (la passerelle). Ce nœud principal comprend une carte Thyone, Adafruit M0 express et une carte Adrastea-I, chargée de transmettre les données collectées vers le cloud via la connectivité IoT cellulaire. Sur les nœuds capteurs, les broches doivent être connectées comme illustré dans le **tableau 1**. Pour commuter le FeatherWing Thyone-I en mode transparent, la broche 3 du cavalier JP1 doit être connectée au 3V3. Pour la passerelle, empilez simplement le Thyone-I FeatherWing en mode transparent avec l'Adrastea FeatherWing et l'Adafruit Feather M0 express.

Pour configurer le matériel dans Edge Impulse, les composants logiciels suivants doivent être installés :

1. le CLI Edge Impulse
2. le CLI Arduino

Suivez les étapes ci-dessous pour programmer le nœud principal :

1. Installez Visual Studio Code et le plug-in Platform IO.
2. Clonez/téléchargez le dépôt *FeatherWings*.
3. Ouvrez l'exemple *Thyone-Adrastea-Bridge*. Générez et téléchargez le code sur l'Adafruit Feather M0 express.



Figure 6. Étiquetage d'échantillons de parasites. (Source : Mouser Electronics)

Apprentissage des modèles de vision industrielle

Edge Impulse facilite grandement la génération d'un modèle de vision industrielle. Il vous suffit de prendre quelques images pour entraîner le modèle sur la plateforme Edge Impulse.

1. Créez un compte sur le site web Edge Impulse.
2. Créez un nouveau projet.
3. Cliquez sur Acquisition de données puis sur *Connecter un nouvel appareil*.
4. Vous pouvez utiliser votre téléphone mobile pour prendre des photos du nuisible et celles-ci seront automatiquement synchronisées avec votre compte Edge Impulse. Pour cela, vous devez scanner le code QR affiché sur la page.

5. Une fois le code QR scanné, commencez à prendre des photos sous différents angles et différents éclairages.
6. Après avoir collecté les images, vous devrez étiqueter le ravageur. Cliquez sur *Labeling queue* (File d'attente d'étiquetage).
7. Étiquetez le nuisible à l'aide de la souris en créant un cadre autour de lui (**Figure 6**). C'est tout pour la collecte des données, et passez à l'étape suivante de l'apprentissage du modèle.

Créer l'impulsion

Pour créer l'impulsion, des données brutes sont acquises en entrée, puis le traitement du signal extrait les caractéristiques et enfin un bloc d'apprentissage procède à la classification des nouvelles données (**Figure 7**).

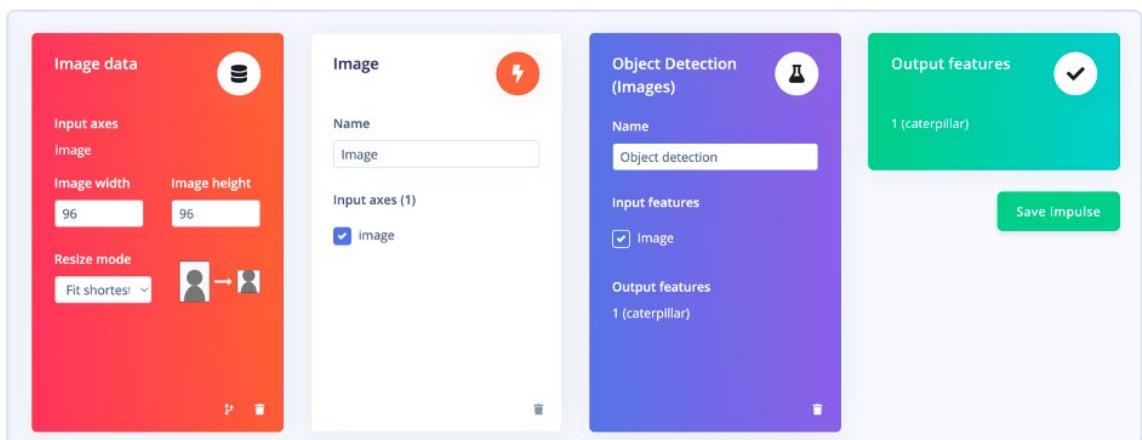


Figure 7. Création de l'impulsion. (Source : Mouser Electronics)

1. En fonction de l'application, ajustez les données d'image.
2. Pour le bloc de traitement, choisissez *Image*.
3. Pour le bloc d'apprentissage, choisissez *Détection d'objets*.
4. Enregistrez l'impulsion.

Classificateur de réseau neuronal

Le classificateur de réseau neuronal prend quelques données d'entrée et génère une valeur indiquant la probabilité que ces données appartiennent à une classe particulière. Pour ce projet, le nombre de cycles d'apprentissage a été fixé à 60.

Exportez l'impulsion en tant que micrologiciel OpenMV et accédez à l'EDI OpenMV. Vous pouvez également utiliser notre modèle déjà entraîné en téléchargeant les fichiers à partir de la page Mouser GitHub [8].

Connectez votre appareil à votre ordinateur. Cliquez sur le bouton *Connect* dans l'EDI OpenMV. L'appareil devrait être automatiquement détecté. Enfin, cliquez sur le bouton *Flash* pour charger le micrologiciel.

Connectivité sans fil

Capteur vers passerelle : le Thyone-I Wireless FeatherWing fonctionnant en mode transparent, agit comme une liaison sans fil 2,4 GHz à la place du filaire. Dans ce mode de fonctionnement, toutes les données provenant du Nicla Vision via l'interface UART sont transmises sans fil. De même, les informations reçues sur la liaison sans fil sont transmises à l'UART.

Passerelle vers le cloud : la passerelle se compose du poste Thyone-I FeatherWing pour recevoir les messages des nœuds. Les données radio sont transmises de manière transparente au Feather M0 express via l'UART. Au démarrage, le MCU M0 configure l'Adrastea-I FeatherWing pour se connecter au courtier cloud MQTT. Dès réception des messages des nœuds capteurs, le MCU M0 publie les données directement dans le cloud. Le courtier cloud MQTT met les messages à disposition de tous les clients abonnés. L'utilisateur peut les recevoir à l'aide de n'importe quelle application ouverte de téléphonie mobile MQTT. Le code de cette application est disponible sur la page GitHub de Würth Elektronik [9].

Combinaison puissante de technologies

Les progrès dans le domaine de l'électronique ainsi que du ML permettent d'effectuer des tâches complexes telles que la détection d'objets avec des ressources matérielles limitées. En y ajoutant la connectivité sans fil, on obtient un mécanisme très performant pour surveiller et contrôler l'environnement à distance. Dans cet exemple, ces technologies sont exploitées pour construire un prototype de système de détection des nuisibles, base d'une agriculture intelligente et connectée. 

Vf : Helmut Müller — 240389-04



À propos de Rafik Mitry

Rafik Mitry a rejoint Mouser Electronics en 2019 après avoir obtenu son master en génie électrique à l'Université technique de Munich, où il a également été chercheur dans le domaine de la récupération d'énergie pendant trois ans. En tant qu'ingénieur marketing technique chez Mouser, Rafik élabore un contenu technique unique qui reflète les tendances technologiques actuelles et futures de l'industrie électronique. En plus de se tenir à niveau des dernières tendances technologiques, Rafik est un passionné d'aviation et de tennis.



À propos de Shashank Hedge

Shashank Hedge est étudiant en master à l'Université technique de Munich. Il travaille chez Würth Elektronik eiSos depuis 2022 en tant que stagiaire/étudiant dans la division connectivité sans fil et capteurs. Son domaine de recherche inclut les systèmes embarqués, l'Internet des Objets et les communications entre véhicules.

LIENS

- [1] Arduino Nicla Vision: <https://tinyurl.com/arduino-nicla-vision>
- [2] Würth Elektronik Thyone-I FeatherWing: <https://tinyurl.com/thyone-i>
- [3] Würth Elektronik Adrastea-I FeatherWing: <https://tinyurl.com/adrastea-i>
- [4] Adafruit Feather M0 Express: <https://tinyurl.com/mo-express>
- [5] OpenMV: <https://openmv.io/pages/download>
- [6] Edge Impulse: <https://edgeimpulse.com>
- [7] Visual Studio Code with PlatformIO extension: <https://platformio.org/install/ide?install=vscode>
- [8] GitHub Page Mouser: <https://github.com/rmitry/pest-detection>
- [9] GitHub page Würth Elektronik eiSos: <https://github.com/WurthElektronik/FeatherWings>

Anybus CompactCom est le choix idéal pour la communication industrielle embarquée – voici pourquoi

contribué par **HMS Networks**

La liberté d'offrir des connexions avec tous les protocoles industriels majeurs devient une caractéristique des dispositifs intelligents de plus en plus essentielle. Cependant, le développement des matériels et logiciels requis est un processus complexe et chronophage. Il existe heureusement une alternative plus facile. Nous expliquons dans cet article comment l'intégration de l'Anybus CompactCom, signé HMS Networks, dans des dispositifs industriels simplifie considérablement la connectivité.

Absence de protocole industriel dominant

Au fil du temps, différents protocoles ont acquis de la popularité dans les différentes régions et secteurs industriels. PROFINET par exemple est courant en Europe, EtherNet/IP détient des parts de marché importantes en Amérique du Nord, EtherCAT et CC-Link sont populaires en Asie et particulièrement au Japon. Pour assurer le succès de votre dispositif sur le marché mondial, vous devez pouvoir vous connecter à de multiples protocoles.

Chaque année, HMS Networks réalise une analyse complète du marché des réseaux industriels visant à estimer la répartition des nœuds nouvellement connectés, par type et par protocole dans le cadre d'une automatisation d'usine. L'étude intégrale accompagnée d'une analyse est publiée sur le site Web de HMS Networks.

Le challenge du développement compatible avec plusieurs protocoles

La prise en charge de protocoles multiples requiert une connaissance approfondie des architectures aussi bien matérielles que logicielles. Elles sont intrinsèquement complexes et différentes pour chaque protocole, bien qu'elles paraissent communes à première vue.

L'Anybus CompactCom relève ces défis en fournissant une solution de communication complète. À la différence de la plupart des alternatives, il intègre le matériel et le logiciel en un seul module.

Fonctionnalités et avantages clés

Module de communication complet : l'Anybus CompactCom prend en charge tous les principaux protocoles industriels et du Cloud. Avec un seul cycle de développement, vous pouvez facilement connecter vos produits à l'ensemble des protocoles que vous avez choisis.

Maintenance au cours du cycle de vie : HMS assure la maintenance aussi bien du matériel que du logiciel tout au long du cycle de vie du produit, y compris des mises à jour régulières conformes aux spécifications des protocoles.

Disponible en deux formats :

➤ **Module** : comprend un connecteur intégré pour la flexibilité et la commodité d'utilisation, idéal pour les options du marché des pièces détachées.

➤ **Brick** : prend en charge les connecteurs personnalisés, est adapté aux applications requérant une protection spécifique dont l'étanchéité à l'eau.

Précertifié pour une intégration facile : ce qui simplifie la transition vers une certification intégrale et assure la conformité aux normes industrielles.

Interface matérielle flexible : prend en charge les communications parallèles, SPI et série, ce qui rend possible une intégration polyvalente avec différents microcontrôleurs hôtes.

Puissante interface logicielle : un pilote en langage C, gratuit et non asservi à un système d'exploitation, qui peut être compilé avec votre logiciel d'application.

Mode de fonctionnement – architecture hôte et maître

Le principe de conception d'Anybus CompactCom est la clé pour comprendre l'avantage de sa conception. L'Anybus CompactCom fonctionne, dans des réseaux d'automatisation, comme un nœud de communication qui connecte le dispositif hôte à un automate programmable et échange des variables et données du processus industriel (**figure 1**). Cette architecture est importante car elle découpe les spécificités de la gestion de protocole à partir de la logique d'application centrale, assurant ainsi une intégration fluide des différents protocoles.

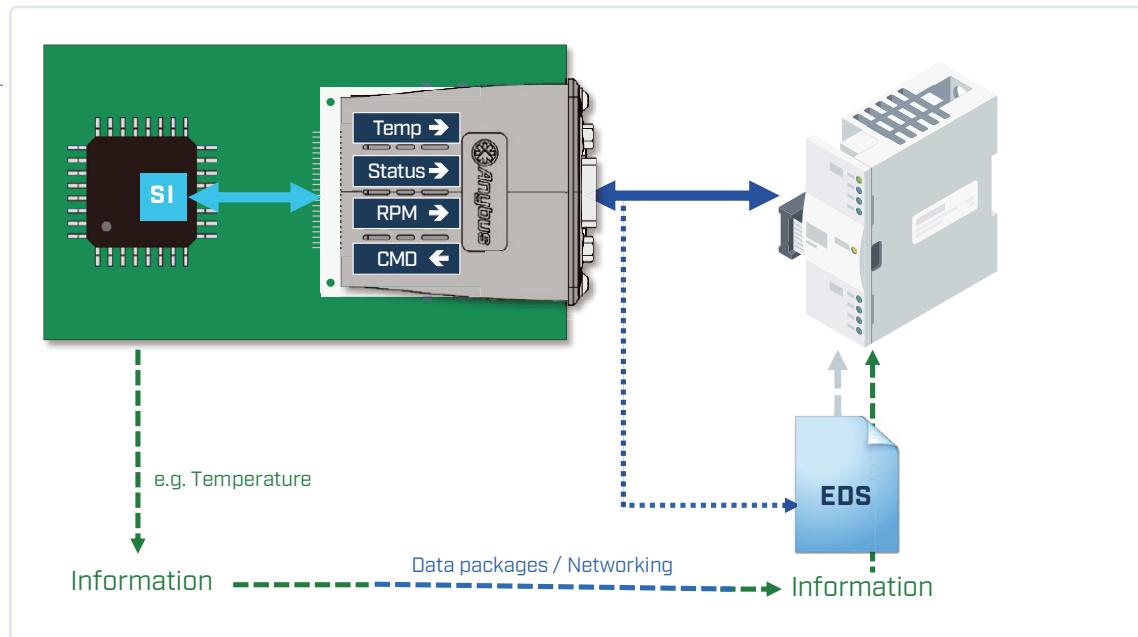


Figure 1. L'Anybus CompactCom connecte l'hôte et le maître.

Pour les fabricants de dispositifs d'automatisation, cette configuration simplifie la communication entre leurs applications avec microcontrôleurs et les dispositifs de commande, ce qui permet aux développeurs de se concentrer sur leur application plutôt que sur les méandres de chaque protocole de communication.

Intégrer le matériel

Au cœur de l'Anybus CompactCom se trouve notre NP40, une puce conçue sur mesure combinant un matériel et un logiciel programmables (figure 2) :

Microcontrôleur et FPGA : le FPGA traite la logique temps réelle des protocoles (p. ex. pour les interfaces PROFINET ou EtherCAT), tandis que le microcontrôleur exécute un système d'exploitation en temps réel pour gérer les piles de protocoles.

API standard : l'API commune apporte une interface logicielle cohérente entre la NP40 et votre microcontrôleur hôte, ce qui assure un interfaçage logiciel fluide malgré les différences internes dues au protocole.

Matériel et interfaçage avec l'application

L'Anybus CompactCom fournit des options d'interfaçage configurables sur mesure avec le matériel et l'application pour répondre à vos besoins (figure 3). Composants et considérations clés :

Modes données :

- **Parallèle** : utilise 8 ou 16 bits en parallèle pour l'échange de données à haut débit.
- **Interface périphérique série (SPI)** : échange de données rapide, jusqu'à 20 MHz, simplifiant les conceptions complexes.
- **Série** : débit de données plus lent, pris en charge pour les systèmes anciens.

Commande du débit et RAZ :

- **Ligne de RAZ** : elle établit la connexion avec un port de sortie du microcontrôleur, ce qui permet la RAZ de la NP40 depuis le logiciel.
- **Ligne d'interruption** : elle commute le flux du programme pour des événements spécifiques.

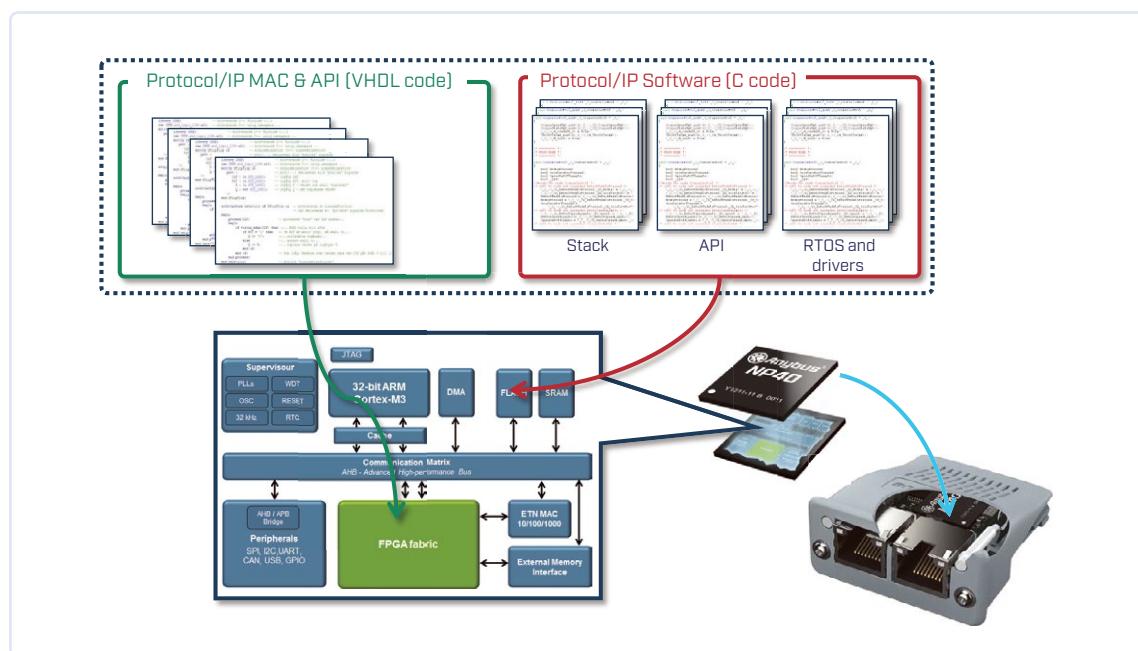


Figure 2. L'Anybus CompactCom fournit un matériel et un logiciel programmable.

Common Application Interface

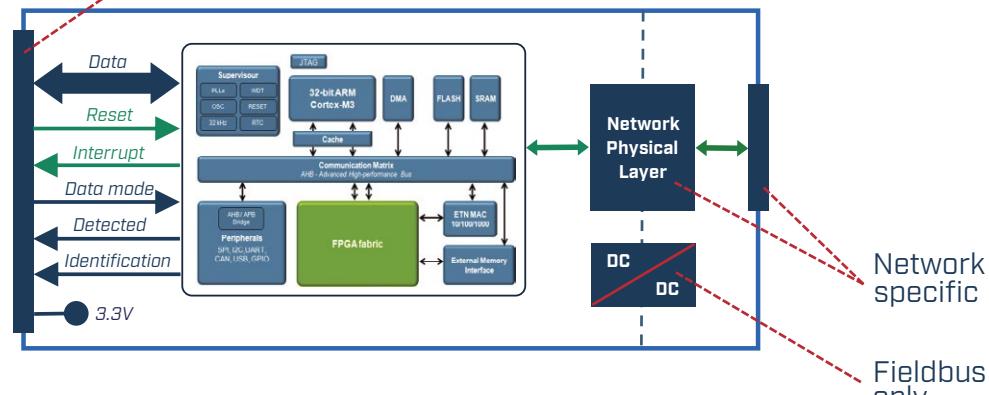


Figure 3. Matériel de l'Anybus CompactCom et interfaçage avec l'application.

Détection et identification du matériel :

- L'application peut tester si un module Anybus CompactCom est monté, et de quel modèle il s'agit, guidant la procédure de démarrage.

Alimentation électrique :

- Requiert une alimentation électrique en 3,3 V pour fonctionner.

Intégrer le logiciel

L'API de l'Anybus Compact Com est commandé par l'interface logicielle et fournit plusieurs options configurables sur mesure pour répondre à vos besoins. Composants et considérations clés :

Gestion variable :

- **Instances de données d'application (ADI)** : définissez des variables du processus (ADI) dans votre logiciel, en spécifiant des noms, types de données, nombres d'éléments et plages de valeurs. Utilisez des pointeurs pour une gestion fluide des données entre l'Anybus CompactCom et votre application. Une mise en œuvre statique ou dynamique est possible.
- **Traitements des données acyclique et en temps réel** : gérez efficacement les demandes acycliques (non urgentes) et l'échange de données en temps réel via les couches logicielles fournies.

Administration de réseau :

- **Objets de réseau** : paramétrez des tâches d'administration spécifiques au réseau, dont les ID fournisseurs, pour assurer un fonctionnement fluide selon différents protocoles.

Appels de fonction :

- **Maintien de connexion et interruption** : appels basiques pour faciliter le fonctionnement du logiciel d'interface.

Fonctionnalités additionnelles :

- **Bonus Ethernet** : appels et objets pour gérer le serveur Web, l'envoi d'e-mails, l'échange FTP, JSON, et de nombreuses autres fonctionnalités.

Communication IIoT

HMS fournit également l'Anybus CompactCom IIoT Secure, une variante qui offre une communication sécurisée avec des applications IIoT en utilisant les protocoles d'échange de données OPC UA ou MQTT. Pour assurer la sécurité de l'information, l'Anybus CompactCom IIoT Secure comprend ce qui suit :

Gestion de certificat : garantit que les communications sont toutes authentifiées et sûres.

Chiffrement : protège les données pendant la transmission pour empêcher un accès non autorisé.

Sécurité Hardware : garantit que le dispositif lui-même est protégé contre les manipulations et les cybermenaces.

Démarrage sécurisé : garantit que le dispositif n'exécute qu'un logiciel authentique, le protégeant ainsi contre des codes malveillants.

Au moment d'intégrer le logiciel, vous pouvez spécifier quelles variables doivent être échangées avec le contrôleur local et quelles autres seront envoyées aux applications digitales. Une fois configuré, l'Anybus CompactCom gère de manière sécurisé la communication des données.

Conclusion

L'Anybus CompactCom se démarque en tant que solution complète pour la communication industrielle embarquée, et offre ainsi une flexibilité et une facilité d'intégration inégalées. Combinant un matériel robuste avec une compatibilité logicielle sophistiquée, il élimine les complexités et coûts cachés associés au développement multi-protocoles, ce qui en fait le choix idéal pour les fabricants désireux de rester compétitifs sur le marché mondial.

Pour plus d'informations sur l'Anybus CompactCom, voir [1].

240392-04

LIEN

[1] Anybus CompactCom : <https://hms-networks.com/network-connectivity-solutions>

norme de communication IQRF

fiabilité des réseaux maillés sans fil à faible débit avec perte

contribué par IQRF Alliance

Les réseaux maillés sans fil sont toujours d'actualité plus de vingt ans après le premier signe de succès sur le marché selon Gartner Hype Cycles. À ce jour, il existe trop de compromis et de limitations en matière de latence, de limites de sauts, de déterminisme et de fiabilité pour toutes les technologies de réseau sans fil prises en charge à l'échelle mondiale. La raison est simple : les concepts techniques se concentraient principalement sur l'optimisation du routage plutôt que sur la fiabilité. Contrairement à cette approche, IQRF s'est principalement concentrée sur la fiabilité de la livraison des messages. Avec plus de 250 sauts de routage et une livraison fiable des messages, IQRF excelle aujourd'hui, en particulier dans les applications d'éclairage public où une longue portée doit être obtenue et une fiabilité

Après deux décennies sur le marché, IQRF se standardise et permet à chacun d'utiliser et de mettre en œuvre toutes les réalisations techniques et protocoles fiables protégés par des dizaines de brevets sous une seule licence « libre de droits ».

Principales caractéristiques et avantages de la norme IQRF

IQRF est une technologie complète, une norme et un écosystème contenant du matériel informatique (émetteurs-récepteurs, passerelles, répéteurs, appareils électroniques complets, accessoires, outils de développement), des logiciels, des protocoles, du support, des outils et des services pour la mise en service.

Depuis 2004, les sociétés MICRORISC et IQRF Tech ont développé IQRF une technologie de communication réseau sans fil bidirectionnelle fiable à faible consommation d'énergie. IQRF avec le protocole IQMESH permet une topologie maillée extrêmement robuste et sophistiquée. Des milliers de systèmes sans fil dans le monde en sont la preuve.

Propriétés

Fiabilité industrielle

Plus de nœuds de réseau signifie une plus grande fiabilité, grâce au protocole IQMESH.

Il est déterministe et fiable même dans des environnements difficiles.

Il ajoute robustesse et fiabilité industrielle à l'Internet

des objets (IoT) sans fil.

Dans un réseau IQRF utilisant le protocole de routage IQMESH, le protocole FRC (Fast Response Commands) peut être utilisé, ce qui permet une agrégation rapide des données et la confirmation des transmissions ou des multidiffusions.

Avec plus de 200 nœuds répétitifs dans un seul réseau travaillant ensemble dans une topologie maillée, il est possible de couvrir d'énormes installations, des halls industriels et des rues et de transmettre un message de manière fiable malgré les mauvaises conditions radio dans la région.

Intégration facile

Une architecture modulaire, un protocole DPA léger mais efficace, des outils de développement matériel et logiciel, de nombreux tutoriels, exemples et supports d'apprentissage, ainsi qu'un excellent support technique, facilitent le développement.

Sécurité ultime

L'approche compréhensive, de l'association des appareils au réseau jusqu'à la communication automatiquement chiffrée, s'appuie sur des normes de sécurité.

La couche de services de sécurité est chargée de garantir les objectifs de sécurité suivants : intégrité des trames, authenticité des trames réseau, confidentialité du pied de page et du contenu utile, protection

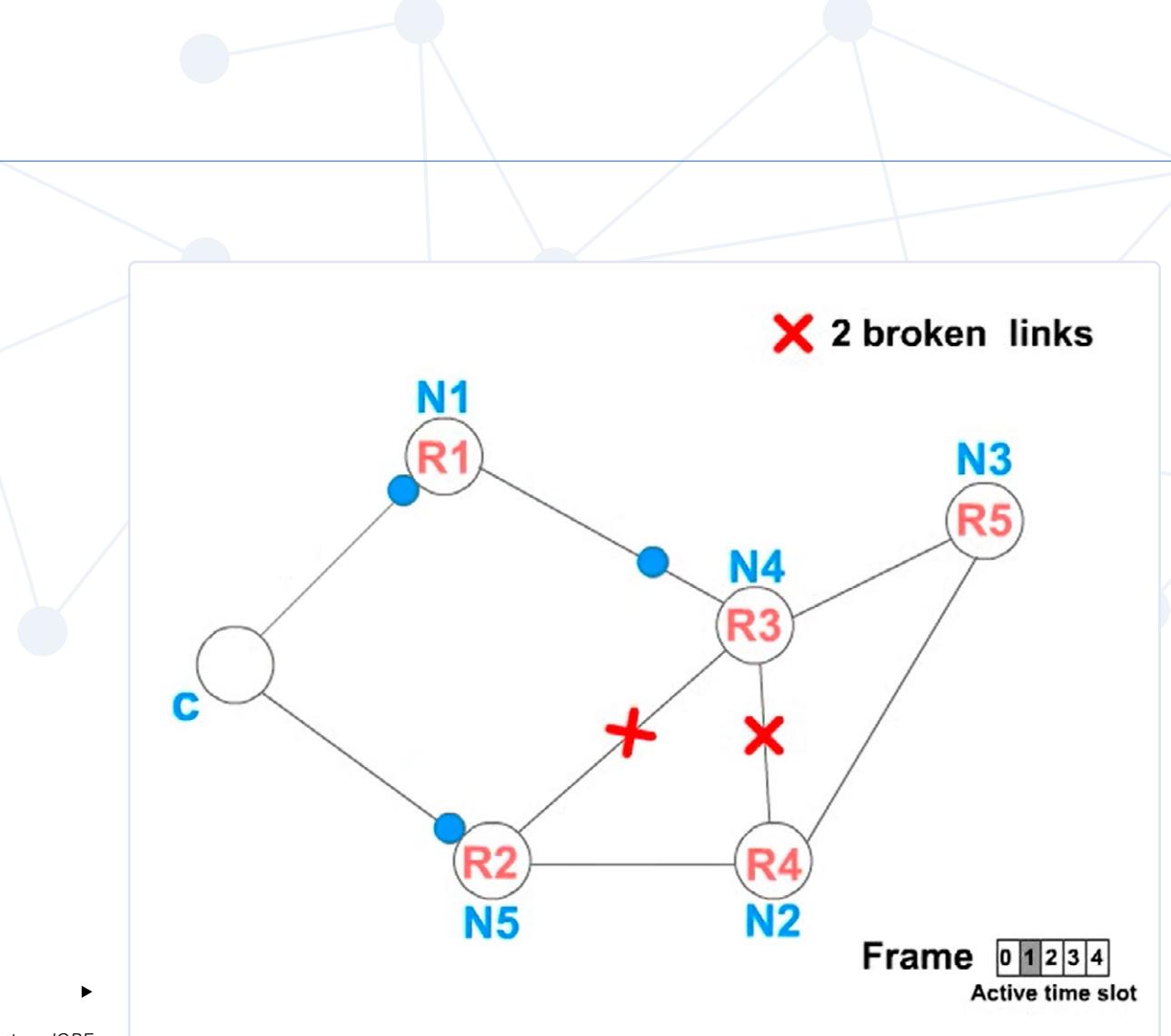


Figure 1. Routage IQRF.

contre le renvoi.

Interopérabilité et écosystème étendu

L'interopérabilité des appareils de différents fabricants est basée sur les normes d'interopérabilité IQRF. Les produits certifiés pour l'interopérabilité IQRF sont répertoriés sur le site Web IQRF Alliance sur le marché et peuvent être facilement utilisés dans des projets IdO complexes.

IQRF True Low Power

IQRF est conçu pour fonctionner efficacement avec une consommation d'énergie minimale, ce qui le rend adapté aux appareils alimentés par batterie et aux applications où l'efficacité énergétique est cruciale. La faible consommation d'énergie de la communication sans fil repousse les limites de la batterie vers de nouvelles limites, définies uniquement par sa propre durée de vie.

Réseau IQRF avec protocole IQMESH

Les réseaux IQRF sont gérés et organisés [1]. L'élément de contrôle pour d'autres périphériques réseau appelés Nœuds est le *Coordinateur*. Les Nœuds de routage sont appelés *Routeurs*. La communication réseau est toujours cryptée et authentifiée selon les dernières normes de sécurité.

Processus d'association

Les Nœuds sont associés au réseau par le Coordinateur. Lors d'une association, le Coordinateur transmet les données de l'association aux Nœuds de manière sécurisée via un message crypté. IQRF MAC est un identificateur unique du périphérique IQRF utilisé pour l'authentification. Le Coordinateur partage en toute sécurité des informations telles que les clés de communication et les paramètres réseau, et attribue à chaque Nœud une adresse logique unique utilisée pour l'adressage.

Processus de découverte de la topologie

Au cours de ce processus, le Coordinateur découvre la topologie des Nœuds de routage (routeurs) et leur attribue un numéro de routage virtuel (VRN), un numéro unique reflétant la distance par sauts par rapport au Coordinateur et définissant la période de routage du Nœud. L'envahissement directionnel basé sur TDMA garantit un routage déterministe et sans collision. Pour une meilleure idée de la façon dont l'envoi d'un message dans le réseau IQRF est effectué à l'aide du protocole de routage IQMESH, reportez-vous à [2] (figure 1).



Protocole FRC (Fast Response Commands)

Le protocole FRC vous permet d'envoyer rapidement une commande à plusieurs Nœuds du réseau et de collecter de petites données à partir de ceux-ci. Il s'agit d'une méthode beaucoup plus rapide que l'interrogation de nœuds individuels un par un à l'aide d'unicast. FRC, dans sa mise en œuvre actuelle, permet d'envoyer jusqu'à 30 B de données utilisateur à tous les Nœuds ou à certains Nœuds et de collecter :

- 2 B de jusqu'à 239 Nœuds
- 1 B de jusqu'à 63 Nœuds
- 2 B de jusqu'à 31 Nœuds
- 4 B de jusqu'à 15 Nœuds

Le temps d'exécution de la commande est strictement déterministe et indépendant de la topologie. Regardez l'animation de l'ensemble du processus sur le [3] (**Figure 2**).

Certification d'interopérabilité IQRF

Les entreprises qui fabriquent des produits pour l'écosystème IQRF sont associées à IQRF Alliance [4]. IQRF Alliance teste l'interopérabilité des produits au niveau de l'application.

Le processus de certification de l'interopérabilité IQRF comprend l'obtention du HWID pour le produit certifié, la vérification des documents et du produit requis, la vérification de sa conformité avec la norme d'interopérabilité IQRF, la résolution d'éventuels problèmes et enfin l'inscription du produit dans la liste des produits (référentiel) et sur le site web de la place de marché. Lorsque les fabricants souhaitent mettre en œuvre la technologie IQRF directement dans leur produit ou produire leurs propres émetteurs-récepteurs, ils doivent suivre les instructions de la spécification de la norme de communication IQRF, qui est définie par IQRF Standards Association [5].

IQRF Standards Association est une organisation à but non lucratif responsable de la spécification de la norme de communication IQRF, de la licence « libre de droits » de la norme de communication ouverte IQRF et de la gestion des adresses IQRF MAC. Les fabricants peuvent mettre en œuvre la norme de communication IQRF dans leurs appareils sous une seule licence « libre de droits ». L'IQRF Standards Association certifie l'interopérabilité au niveau de la communication de base.

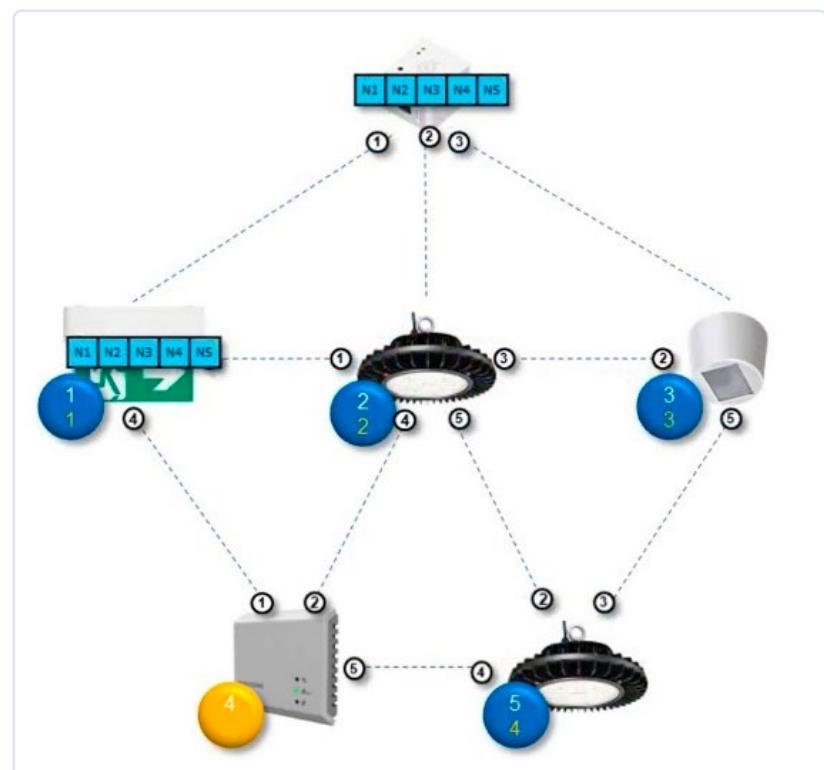
de droits ». L'IQRF Standards Association certifie l'interopérabilité au niveau de la communication de base.

Conclusion

Grâce au routage, IQRF peut communiquer de manière fiable sur de courtes distances allant jusqu'à plusieurs kilomètres. Il est parfaitement adapté à de nombreuses applications à faible consommation d'énergie issues de la surveillance, de l'automatisation ou du contrôle. Il est idéal pour des applications allant de la sécurité à l'automatisation des bâtiments en passant par la ville intelligente. L'avantage de la technologie IQRF réside dans sa combinaison d'un fonctionnement à faible consommation, de la capacité de fonctionner dans une topologie de réseau avec une fiabilité et une portée élevées, de l'interopérabilité des produits de l'écosystème IQRF, des fonctions de sécurité et d'une intégration facile, ce qui en fait une solution polyvalente pour un large éventail d'applications IdO utilisant la transmission sans fil. ■

Figure 2. Protocole FRC.

240418-04



LIENS

- [1] IQRF - Technologie pour le sans fil : <https://www.iqrf.org/>
- [2] Le réseau IQRF est effectué à l'aide du protocole de routage IQMESH : <http://www.iqrf.org/technology/iqmesh/unicast>
- [3] L'animation de l'ensemble du processus : <https://www.iqrf.org/technology/iqmesh/frc>
- [4] IQRF Alliance (2024) : <https://www.iqrfalliance.org/>
- [5] IQRF Standards Association (2024) : <https://standard.iqrf.org/>

comment construire un robot agricole intelligent

Considérations et défis techniques essentiels

Mark Patrick (Mouser Electronics)

Dans cet article, nous verrons quels sont les principaux défis liés à la conception de robots agricoles et quels éléments doivent impérativement être pris en compte afin d'en garantir la mise en œuvre efficace dans un environnement aussi difficile que celui de l'agriculture. Ces robots doivent tout d'abord allier durabilité et précision, mais aussi être capables de gérer efficacement des ressources énergétiques limitées et intégrer des composants robustes afin de résister aux assauts de la poussière, de l'humidité et de conditions météorologiques extrêmes. Le traitement de données en temps réel et une forme d'intelligence embarquée sont essentiels afin de limiter autant que possible la dépendance de ces robots à l'égard de réseaux de communication peu fiables.

Une combinaison de facteurs tels que la croissance démographique, la réduction des surfaces agricoles et la rareté des ressources, pour ne citer que les principaux, remettent en question la pérennité des méthodes agricoles conventionnelles. Il s'avère donc nécessaire de prendre ces problèmes à bras-le-corps si l'on souhaite assurer une production alimentaire durable et continue. Pour ce faire, les pratiques agricoles doivent commencer à intégrer des solutions intelligentes tirant parti de technologies de pointe telles que la robotique agricole.

Cependant, tout comme le secteur qu'elle s'efforce de soutenir, la robotique agricole doit faire face à de nombreux défis. La poussière, l'humidité et la pluie sont autant de facteurs qui font des exploitations agricoles des environnements difficiles pour les équipements électroniques sensibles. De plus, la faible disponibilité de ressources en énergie, mais aussi de moyens financiers, fait souvent obstacle au déploiement de robots agricoles. Ce sont justement ces obstacles que



Gros plan d'un robot de récolte automatisée
(Source : Sunday Cat Studio - stock.adobe.com.)

nous allons examiner dans le présent article en mettant en évidence les principales considérations techniques et les principaux défis dont les concepteurs doivent tenir compte afin de concrétiser leur projet.

Principales considérations en matière de conception

Par rapport aux robots utilisés dans le secteur du BTP, dont le maître-mot est la durabilité, et aux robots médicaux, qui se distinguent par leur précision d'exécution, les robots agricoles doivent trouver le juste équilibre entre ces deux qualités, notamment pour manipuler des produits sans causer de dommages. Ces robots doivent en outre être intelligents et sûrs, de façon à pouvoir exécuter des tâches complexes à proximité des humains et des animaux, et être optimisés sur le plan de la consommation d'énergie.

Efficacité énergétique

L'efficacité énergétique est un aspect essentiel des robots agricoles. Ils doivent en effet être capables de fonctionner longuement de manière autonome avec des batteries ou des réserves de carburant limitées. Étant donné les contraintes financières qui pèsent sur de nombreuses exploitations agricoles, pour que des équipements tels que les robots cueilleurs (voir **figure 1**) puissent se généraliser, il est impératif que la robotique agricole réponde à ce critère d'efficacité énergétique, mais aussi à celui de la viabilité économique.

Les économies d'énergie représentent un aspect fondamental de la conception de robots agricoles. Des fonctionnalités telles que la sortie de veille intelligente de l'appareil lorsqu'il est mis en mouvement ou un mode éco limitant les fonctionnalités afin de réduire la consommation d'énergie sont de bonnes pistes pour augmenter la durée de vie de la batterie ou réduire la consommation de carburant. Pour créer des robots efficaces sur le plan énergétique, les ingénieurs doivent donc privilégier le développement de ces stratégies ou l'ajout de composants intégrant ces stratégies.

Durabilité

Pour concevoir des robots capables de résister à un environnement aussi difficile que celui d'une exploitation agricole, il est indispen-

sable de disposer de composants et de matériaux robustes. Les connecteurs, par exemple, doivent être résistants aux intempéries et dotés de joints pour empêcher la pénétration de l'humidité. Les carrosseries doivent être fabriquées dans des matériaux durables et résistants aux UV afin de protéger l'équipement des chocs et des intempéries. Les antennes doivent être hautement fiables et performantes, y compris et surtout dans des conditions météorologiques difficiles.

Pour s'assurer de la fiabilité de leur robot, les concepteurs peuvent au choix utiliser des composants intrinsèquement robustes, avec des indices de protection (IP) élevés, ou opter pour des composants standard enfermés dans des boîtiers de façon à renforcer leur durabilité. Il est par exemple possible de recouvrir les PCB d'un revêtement enrobant afin de les protéger de l'humidité et de la poussière. On peut aussi ajouter des dissipateurs thermiques en vue de dissiper l'excès de chaleur généré par les composants électroniques ou causé par les conditions climatiques (un trop fort ensoleillement, par exemple). Quant aux boîtiers, ils peuvent constituer une protection à la fois contre la chaleur et les infiltrations d'eau. Une conception robuste permet ainsi d'obtenir des robots agricoles suffisamment fiables et durables pour être affectés à des applications en extérieur.

Traitements en temps réel des données et intelligence embarquée

Dans le domaine de la robotique agricole, s'appuyer sur l'intelligence ou le traitement de données dans le cloud peut s'avérer problématique à cause des éventuelles pannes de communication. Dans des zones agricoles étendues, la couverture cellulaire est bien souvent limitée et problématique. Par ailleurs, tenter de pallier ce problème en déployant des réseaux Wi-Fi locaux et des services cloud est une entreprise extrêmement coûteuse. De plus, les systèmes de communication reposant sur le cloud nuisent considérablement à la durée de vie des batteries des robots basse consommation comme les robots cueilleurs (voir **figure 2**) ou les robots désherbeurs.

Une solution consiste alors à analyser les données directement sur l'appareil plutôt que dans le cloud, et ce, grâce au traitement en périphérie (edge computing) des données à l'aide de l'IA ou d'algorithmes de ML. L'intelligence embarquée permet aux robots agricoles de réaliser des analyses en temps réel et de prendre des décisions immédiates sans dépendre d'un réseau externe. Cette approche réduit à la fois la consommation d'énergie et la dépendance aux



Figure 1. Un robot cueilleur automatisé avec chariot collecteur
(Source : kinwun - stock.adobe.com)

réseaux de communication. Parce qu'il garantit un délai de traitement minimal par rapport aux solutions basées sur le cloud, le traitement en périphérie (edge processing) améliore grandement l'efficacité et la sécurité de tâches aussi variées que la détection d'obstacles, la plantation de précision ou le suivi en temps réel de la santé des cultures.

Identifier les capteurs appropriés

La robotique agricole ne pourrait exister sans capteurs. Si les capteurs LIDAR, les caméras ou encore les détecteurs d'humidité améliorent indéniablement l'efficacité et les performances des robots, ces dispositifs ont aussi un coût. Les ingénieurs doivent choisir les technologies à mettre en œuvre tout en cherchant à équilibrer au mieux les contraintes en matière de coût, d'énergie et de conditionnement afin de produire des conceptions qui répondent aux attentes du marché.

Créer un robot agricole fiable et performant

La réussite de la conception d'un robot agricole dépend en grande partie des performances des composants sous-jacents.

Technologies basse consommation

Les modes veille ou les modes éco dans les composants comme les capteurs, l'électronique de contrôle et les modules de communication contribuent à optimiser la durée de vie de la batterie ou la consommation de carburant d'un robot. Par ailleurs, les capteurs d'images sont des éléments essentiels à toute bonne conception. L'AR0830 Hyperlux LP d'onsemi [1] est un exemple de ce type de capteurs. Il a été conçu dans le but d'offrir d'excellentes performances tout en consommant un minimum d'énergie. Avec une fréquence d'images de 60 ips, il se destine idéalement aux applications de vision industrielle en milieu agricole, par exemple l'identification de fruits mûrs ou la détection de parasites et d'adventices. Outre sa faible consommation d'énergie, ce capteur présente d'autres fonctionnalités qui en améliorent l'efficacité, notamment la sortie de veille lors de la mise en mouvement et des modes de sous-échantillonnage pour réduire davantage la bande passante requise pour la transmission des données. Ces modes optimisent les performances dans diverses applications agricoles en équilibrant la résolution, la sensibilité, la fréquence d'images et la consommation d'énergie.

L'électronique de contrôle est un élément primordial de la conception de l'alimentation dans le domaine de la robotique agricole. Son rôle est de superviser les opérations essentielles et de piloter les fonction-



Figure 2. Les robots utilisés pour la cueillette des tomates s'appuient généralement sur une technologie de vision industrielle basée sur l'apprentissage automatique (ML) afin de déterminer si les tomates sont mûres (Source : kritsana - stock.adobe.com)



Figure 3. Les connecteurs de raccordement à cliquet de la série M de Lemo sont conçus pour résister à des conditions environnementales difficiles (Source : Mouser Electronics)

nalités et l'efficacité énergétique des robots. La série STM32U5 de STMicroelectronics [2] est une famille de microcontrôleurs (MCU) à très faible consommation d'énergie basés sur la plateforme Arm Cortex-M33. Ces MCU conviennent parfaitement pour intégrer des sous-systèmes de robots agricoles comme l'interface homme-machine (IHM) ou le contrôle des capteurs.

Les MCU de la série STM32U5 résistent à des températures pouvant atteindre 125 °C et intègrent notamment des fonctions de sécurité matérielle, des périphériques de communication et une mémoire flash. Le mode autonome basse consommation en arrière-plan (LPBAM) permet aux périphériques de rester actifs lorsque l'appareil est à l'arrêt et ainsi de réaliser de belles économies d'énergie. Ce niveau élevé de fonctionnalités intégrées et de modes d'alimentation intelligents participe à une efficacité accrue du système par rapport aux solutions traditionnelles.

Des composants robustes

Des composants robustes sont une condition sine qua non pour concevoir des robots agricoles capables de résister au mieux aux aléas de l'environnement difficile dans lequel ils évoluent. Parmi ces composants, les antennes extérieures de la série IPW de TE Connectivity/Linx Technologies [3] fonctionnent sur une plage de fréquences de 617 MHz à 7,1 GHz aussi bien adaptée aux communications cellulaires qu'au Wi-Fi et au LPWA/ISM. Elles affichent en outre un gain pouvant atteindre les 8,7 dBi. L'indice de protection IP67 garantit une connexion solide dans de vastes zones rurales et une protection durable contre la pénétration de poussière et d'eau.

Dans des applications de robotique agricole, ces antennes peuvent être associées à des modules de communication LoRaWAN comme les solutions LoRa de Microchip Technology [4]. Par rapport à d'autres protocoles sans fil, ces solutions LoRaWAN proposent une connectivité basse consommation sur des bandes de fréquences radio longue portée inférieures au gigahertz, ce qui permet d'établir des communications sur des distances supérieures à 15 kilomètres en zone rurale ou périurbaine.

La fiabilité des connecteurs représente un aspect essentiel de la robotique agricole, car les connecteurs sont des points d'entrée potentiels pour des éléments extérieurs comme l'eau et la poussière. Il est donc important de choisir des connecteurs adaptés pour le câblage interne, les communications externes et les ports de charge si l'on souhaite éviter que les appareils tombent en panne sur le terrain.

Les connecteurs de la série M de Lemo [5] sont réputés pour leur durabilité dans les secteurs de la robotique industrielle et de l'automobile et sont par conséquent parfaitement adaptés aux applications agricoles.

Ces connecteurs disposent de 2 à 114 contacts et d'une excellente résistance aux vibrations. Conformes à la norme IP68, ils peuvent résister à 15 heures d'immersion à 2 mètres de profondeur.

Exploiter l'apprentissage automatique en périphérie

La précision, ainsi qu'une capacité de traitement rapide et fiable des données, sont des caractéristiques primordiales de la robotique agricole. L'intégration de solutions avancées d'IA ou de ML est désormais essentielle pour améliorer les fonctionnalités et optimiser les performances tout en garantissant l'adaptabilité à différentes tâches et à différents environnements.

Edge Impulse [6] est une plateforme polyvalente conçue pour développer et déployer des solutions d'IA de périphérie (edge) adaptées à la robotique agricole. Grâce à cette plateforme, les appareils ne sont plus totalement dépendants d'une communication constante avec le cloud. En plus de préserver la durée de vie des batteries, Edge Impulse permet une prise de décision en temps réel dans des environnements difficiles sur le plan de la connectivité et de la sécurité.

Comme elle est indépendante du matériel, la plateforme Edge Impulse permet d'utiliser une gamme presque infinie de matériel. Elle prend également en charge la boîte à outils Train Adapt Optimize (TAO) développée par NVIDIA. NVIDIA TAO simplifie la création de modèles d'IA personnalisés grâce à l'apprentissage par transfert, un mode d'apprentissage qui permet aux développeurs d'affiner des modèles pré-entraînés pour des tâches spécifiques avec un minimum de données d'entrée. Ce mode s'avère idéal dans le cadre d'applications agricoles, où les modèles d'IA doivent pouvoir être adaptés à des tâches aussi diverses que la cueillette des fruits, la surveillance des sols ou la détection des parasites. Ils peuvent ainsi contribuer à créer des solutions plus puissantes et plus efficaces.

Résumé

Bien que les robots agricoles partagent de nombreuses caractéristiques et exigences avec leurs homologues industriels, la sélection des composants est une étape décisive pour garantir qu'ils répondent aux rigueurs des environnements agricoles. L'intégration de technologies robustes, efficaces et innovantes participe à créer des produits capables d'augmenter l'efficacité de la production, d'assurer la durabilité et d'atténuer les problèmes propres à l'agriculture de façon à pérenniser la production alimentaire dans un monde où les ressources sont limitées. ▶

240380-04

LIENS

- [1] AR0830 Hyperlux LP d'onsemi : <https://eu.mouser.com/new/onsemi/onsemi-ar0830-image-sensors>
- [2] Série STM32U5 de STMicroelectronics : <https://eu.mouser.com/new/stmicroelectronics/stmicro-stm32u5>
- [3] antennes extérieures de la série IPW de TE Connectivity/Linx Technologies: <https://eu.mouser.com/new/te-connectivity/linx-outdoor-antennas>
- [4] Solutions LoRa de Microchip Technology : <https://eu.mouser.com/new/microchip/microchip-lora-solutions>
- [5] Série M de Lemo : <https://eu.mouser.com/new/lemo/lemo-robotics-industrial-control>
- [6] Edge Impulse : <https://mouser.co.uk/manufacturer/edge-impulse>

filtre coupe-bande audio à fréquence réglable

solution universelle de suppression de fréquences dans le domaine audio

Alfred Rosenkränzer (Allemagne)

Lors de la mesure de la distorsion dans les appareils audio, il est judicieux d'éliminer le signal de test sinusoïdal avant l'analyseur de spectre. Cette opération devient particulièrement aisée avec ce filtre coupe-bande dont la fréquence est réglable. Les mesures ne sont donc pas limitées à la fréquence typique de 1 kHz, mais peuvent s'étendre aux domaines des graves et des aigus.

Les avantages de l'insertion d'un filtre coupe-bande lors de la mesure de distorsions telles que la distorsion harmonique totale (THD) dans le domaine de l'audio ont déjà été discutés en détail dans mon article « Filtre coupe-bande de Fliege pour les mesures audio » [1]. En voici un résumé des points les plus importants.

Mesures avec un filtre coupe-bande

Lorsqu'on mesure la distorsion d'un signal de test à la sortie d'un circuit, il n'est pas facile de distinguer la part de cette distorsion apportée par le signal de test de celle propre à l'appareil évalué et, enfin, de celle ajoutée par l'appareil de mesure. La procédure suivante permet toutefois d'effectuer une mesure raisonnable. Tout d'abord, mesurez et notez le niveau du signal de test en l'absence de filtre coupe-bande à l'entrée de l'appareil de mesure. Puis, au lieu d'envoyer simplement la sortie (après réglage du niveau) à un analyseur de spectre, insérez un filtre coupe-bande accordé précisément sur la fréquence du signal de test sinusoïdal entre la sortie de l'appareil testé et l'entrée de l'appareil de mesure. Le signal de test est ainsi fortement atténué et ne peut

donc pas être la cause de distorsions dans l'étage d'entrée de l'appareil de mesure ou dans le numériseur d'un analyseur de spectre. Dans la plupart des cas, une atténuation de 50 dB suffit. Les distorsions mesurées maintenant sont presque certainement déjà contenues dans le signal à mesurer et proviennent donc soit du signal de test lui-même, soit, si ce signal est de haute qualité, presque entièrement de l'appareil testé.

Bien entendu, pour cette application, la qualité du filtre coupe-bande lui-même doit être excellente. Il doit produire le moins de distorsion possible et, surtout, son étage d'entrée ne doit jamais être saturé. En outre, les fréquences du filtre coupe-bande et du signal de test doivent être très précisément égales ce qui requiert des réglages très fins, en particulier pour cette dernière, car les filtres coupe-bande les plus simples, comme ceux utilisés ici, sont à bande très étroite et de nombreux filtres de ce type (comme celui décrit dans [1]) sont à fréquence fixe. L'un des inconvénients de ce genre de solutions est qu'il faut un filtre coupe-bande approprié pour chaque fréquence du signal de test si l'on veut aller au-delà de la mesure habituelle à 1 kHz.

Filtre coupe-bande réglable

Ce sont surtout les limitations concernant les fréquences de test qui m'ont motivé pour développer un nouveau filtre à fréquence

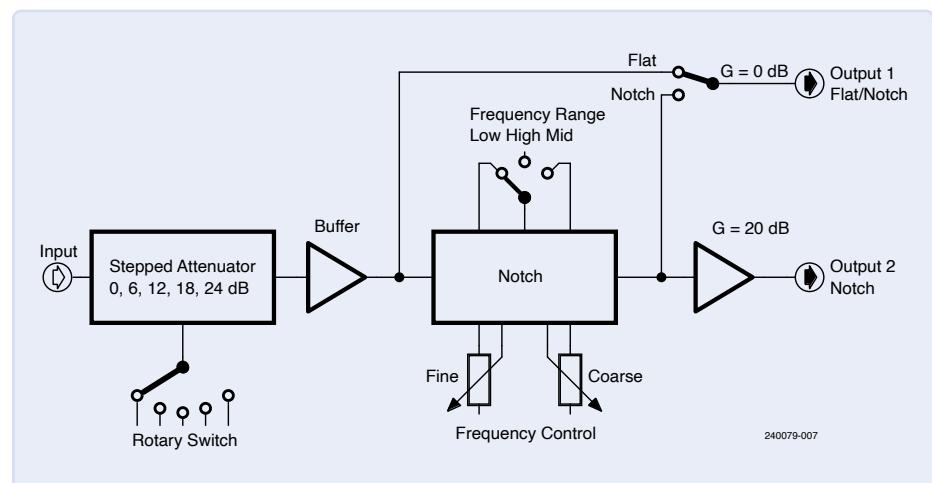


Figure 1. Schéma de principe du filtre coupe-bande.

Caractéristiques techniques

- Le signal d'entrée peut être atténué de 0, 6, 12, 18 ou 24 dB à l'aide d'un atténuateur commutable.
- La fréquence du filtre peut être réglée à l'aide de deux potentiomètres (fin et grossier) dans un rapport de 1:7.
- Il existe trois bandes de fréquences (basse, medium et haute), qui peuvent être sélectionnées à l'aide d'un commutateur. Ces bandes couvrent la majeure partie du spectre audio.
- Le filtre coupe-bande peut être contourné à l'aide d'un commutateur pour alterner entre le signal filtré et le signal d'entrée.
- Une deuxième sortie fournit un signal filtré amplifié d'un facteur 10 afin de pouvoir mesurer raisonnablement même des petits signaux de test et/ou d'étendre la plage de mesure des distorsions en les amplifiant de +20 dB.

réglable de haute qualité. Il couvre la majeure partie du spectre audio. Les performances de ce filtre sont présentées dans l'encadré « **Caractéristiques techniques** ».

Examinons tout d'abord le circuit de la **figure 1**. Le signal d'entrée passe d'abord par un atténuateur commutable de 0, 6, 12, 18 ou 24 dB afin d'être réduit au niveau optimal pour le filtre coupe-bande d'environ -10 dB (environ 0,9 V_{CC}). Ce niveau est un compromis optimal entre le bruit et la distorsion qui naissent dans le filtre coupe-bande. Le signal est envoyé à l'entrée du filtre coupe-bande via un tampon, tandis que le signal non filtré est envoyé à la sortie 1 via le commutateur (*Flat/Notch*). Lorsque le commutateur est sur la position *Flat*, le niveau peut être mesuré sur cette sortie en y connectant un analyseur audio, un voltmètre numérique approprié ou un oscilloscope.

Avec un appareil tel que le QuantAsylum QA403, il est possible de saisir la valeur d'atténuation sélectionnée pour obtenir des niveaux corrigés. Avec un appareil de mesure moins perfectionné, vous devrez finalement corriger vous-même les niveaux mesurés.

Le sélecteur de plage à trois positions permet de choisir une plage de fréquence de coupure. Les trois plages couvrent les bandes de

Liste des composants

Résistances

Sauf indication contraire :
MELF 0204 ou CMS 1206 à couche mince, 1%
R1, R4, R13, R16, R24 = 100 Ω
R2 = 100 kΩ
R3, R5, R6, R9 = 2,8 kΩ
R7, R20 = non installées
R8, R21 = 1,8 kΩ
R10 = 1,5 kΩ
R11 = 0 Ω
R12 = 4,22 kΩ, CMS 0603
R14 = 2,15 kΩ
R15, R17 = 8,2 kΩ
R18, R19, R25, R26 = 4,7 kΩ, CMS 0603
R22, R23, R27, R28 = 10 kΩ, CMS 0603
R29, R30 = 1 kΩ, CMS 0603
R31, R32 = 47 Ω, CMS 1206
R33 = 3,3 kΩ, CMS 0603
R36 = 1,1 kΩ
R37 = 10 kΩ
R38 = 220 Ω
R15, R17 = 8,2 kΩ
P1 = 10 kΩ lin., potentiomètre stéréo, 4 mm
P2 = 1 kΩ lin., potentiomètre stéréo, 4 mm

Condensateurs

C1, C7 = 82 nF, C0G, CMS 0805
C2, C8 = 100 nF, C0G, CMS 1206
C3, C9 = 5,6 n, C0G, CMS 0603
C4, C10 = 22 nF, C0G, CMS 0805
C5, C11 = 1,5 n, C0G, CMS 0603
C6, C12 = 3,9 n, C0G, CMS 0603
C13, C14 = 2,200 μF / 16 V, électrolytique, pas 2/10"
C15...C18, C22...C28 = 100 nF, X7R, CMS 0603
C19, C20, C29...C36 = 22 μF / 20 V, tantale, SMC-B
C21 = 220 μF / 16 V, électrolytique, pas 2/10"

Semi-conducteurs

D1...D5 = 1N4148, DO214AC
D6...D8 = SK34, DO214AC
B1 = B40C1000, pont redresseur, rond
LED1, LED2 = LED, CMS 0805
T1...T4 = 2N3904, SOT23
IC1...IC3 = OPA2210, SOIC8
IC4 = 7812, TO220
IC5 = 7912, TO220
IC6 = 7805, TO220

Divers

X1, X4 = Embase BNC, AMP 227161
X2 = bornier à vis à 2 bornes, pas 5 mm, AK300/2
X3 = bornier à vis à 3 bornes, pas 5 mm, AK300/3
K1...K3, K5, K6, K8 = barrette à 2 broches, pas 1/10"
K4 = barrette à 4 broches, pas 1/10"
K7 = barrette à 3 broches, pas 1/10"
K1 = relais CMS, 2 x inverseur, TQ2-SA-5V (Forme C)
K2...K5 = relais CMS, 5 V, 9802-05-00
F1 = fusible, 250 mA, porte-fusible 22.5 mm
TR1 = transformateur, 3.6 VA, 2 x 12 V sec., Gerth 387.24.2
Atténuateur = circuit imprimé assemblé (voir texte)
Commutateur 1 x 5 (or 2 x 5)
LED, 5 mm avec embase
Commutateur sur K4, 1 x inverseur à position centrale (voir texte)
Interrupteur sur K2
3 embases BNC isolées, pour montage sur panneau avant
2 x bouton pour potentiomètre, axe 4-mm
1 x bouton pour commutateur rotatif, axe 6-mm

fréquences suivantes : *basse* = 80...530 Hz, *medium* = 485...3280 Hz, et *haute* = 2,99...20 kHz. La fréquence dans ces plages peut être modifiée dans un rapport de 1 à 7 à l'aide des deux potentiomètres de réglage de la fréquence (*grossier* et *fin*). Ces plages peuvent être adaptées à vos propres besoins en utilisant d'autres valeurs de composants, moyennant toutefois un certain effort de calcul. Le signal de sortie du filtre est envoyé soit directement à la sortie 1 via le commutateur,

soit à la sortie 2 amplifié de 20 dB supplémentaires, ce qui est utile pour mesurer des signaux plus petits ou pour élargir le domaine de mesure des niveaux de distorsion faibles si l'analyseur audio ne dispose pas d'un réglage d'amplitude calibré.

Circuit

La **figure 2** montre le circuit détaillé, et la **figure 3**, la disposition du circuit imprimé que j'ai conçu. L'atténuateur d'entrée n'est pas

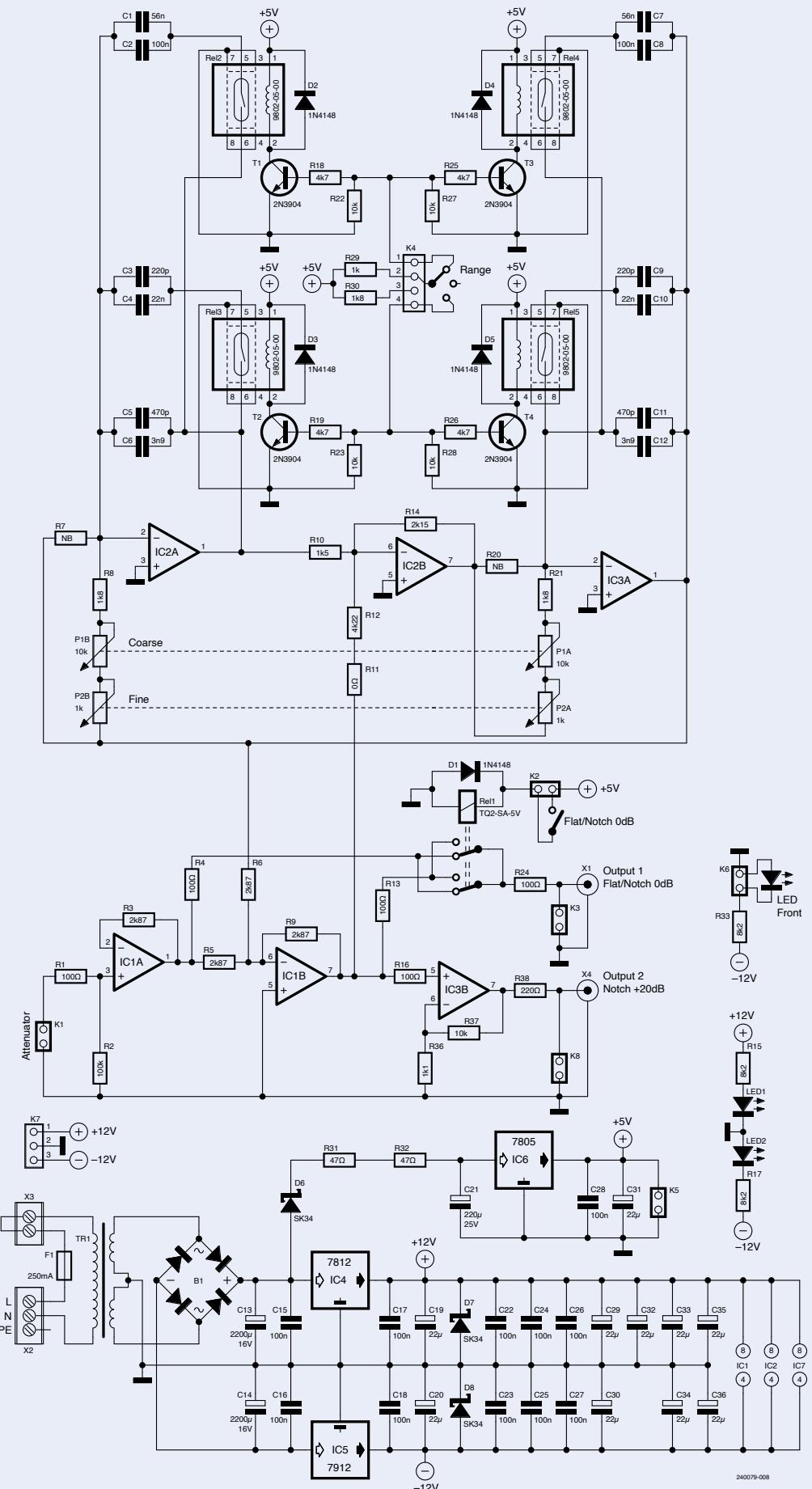


Figure 2. L'atténuateur ne fait pas partie du circuit du filtre. Il est connecté à K1.

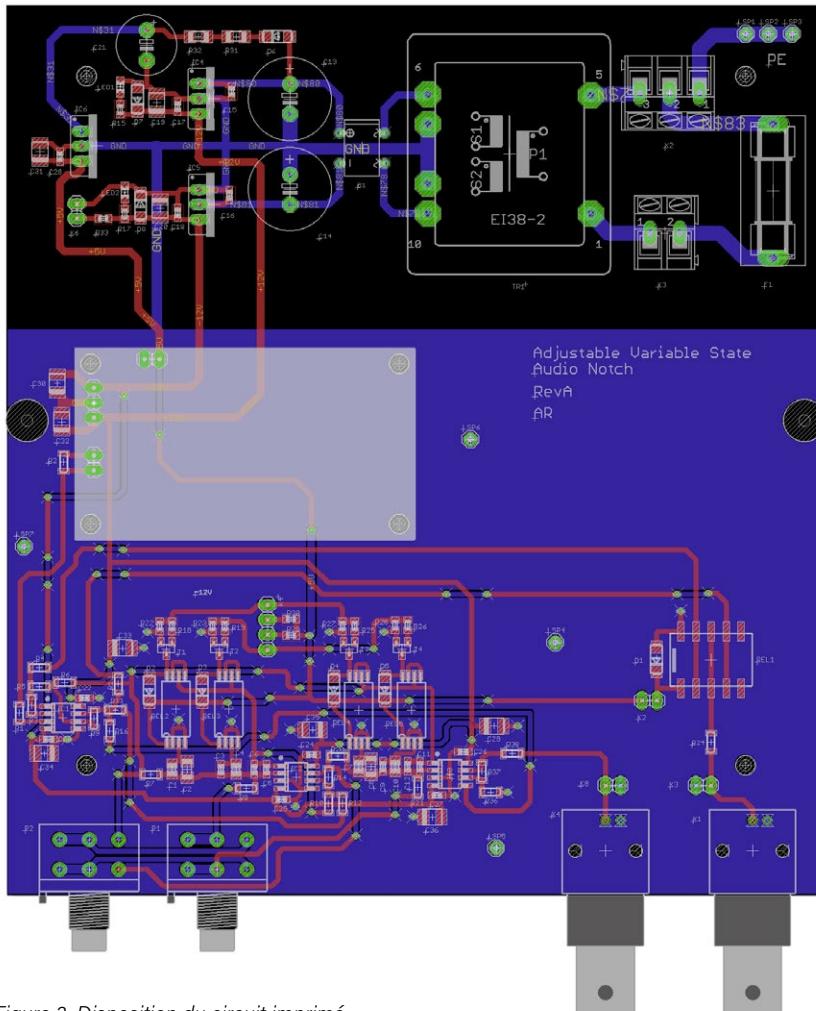


Figure 3. Disposition du circuit imprimé.

inclus dans la figure 2. Il s'agit d'un circuit imprimé séparé qui est implanté sur le circuit imprimé du filtre. Je l'ai décrit précédemment dans mon article « Atténuateurs pour signaux audio (2) » [2] dans Elektor. La version asymétrique est normalement suffisante si aucune entrée différentielle n'est requise. Les différentes atténuations sont sélectionnées à l'aide d'un commutateur rotatif.

La sortie de l'atténuateur est connectée à K1 et atteint ainsi le tampon IC1A, dont le signal de sortie est envoyé à la sortie 1 (X1, K3) via R4 et les contacts du relais Rel1 en position *Flat*. Le filtre coupe-bande est commandé par R5. Il s'agit d'un « filtre à variable d'état » construit autour de IC1B, IC2A, IC2B et IC3A. Il est important que les composants déterminant la fréquence pour IC2A et IC3A soient identiques. Ceci s'applique aux condensateurs dans les branches de rétroaction négative de IC2A et IC3A ainsi qu'aux résistances devant

les entrées inverseuses de ces op-amps, c'est-à-dire les potentiomètres 1-kΩ et 10-kΩ ainsi que R8 et R21 (puisque R7 et R20 ne sont pas montés).

Les potentiomètres doivent être des versions stéréo, car les intégrateurs autour de IC2A et IC3A doivent toujours être réglés de la même manière. Si l'on ne tient pas compte de R7 et R20 (qui ne sont pas montés), la résistance minimale pour les deux intégrateurs avec les potentiomètres tournés à zéro est simplement $R8$ ou $R21 = 1,8$ kΩ. Si les potentiomètres sont complètement tournés vers le maximum, 11 kΩ supplémentaires sont ajoutés dans chaque cas. La plage de variation est donc de 12,8 kΩ / 1,8 kΩ ≈ 7,1.

C5/C6 et C11/C12 sont des condensateurs installés en fixe pour IC2A et IC3A, respectivement, et déterminent la bande de fréquences haute. Les relais Rel3 et Rel5 sont utilisés pour commuter les condensateurs C3/C4

(ou C9/C10) connectés en parallèle avec C5/C6 (ou C11/C12) pour la bande de fréquence moyenne. De la même manière, les relais Rel2 et Rel4 commutent les C1/C2 (ou C7/C8) connectés en parallèle avec C5/C6 (ou C11/C12) pour la bande de fréquence basse. Tous les relais sont des versions 5 V. Les relais pour la commutation de bande sont contrôlés par T1 à T4. Un commutateur à position centrale (comme dans mon prototype) ou un commutateur rotatif à trois positions peut être connecté à K4 pour le réglage. Le relais Rel1, qui sert à commuter entre le signal d'entrée (atténué et tamponné) et la sortie du filtre coupe-bande, est activé directement par un interrupteur connecté à K2.

Pour la deuxième sortie avec un niveau plus élevé, le signal du filtre coupe-bande est amplifié d'un facteur 10, soit 20 dB, à l'aide de IC3B. D'autres facteurs d'amplification sont possibles en modifiant R36 et/ou R37. Si vous ne souhaitez pas utiliser de prises BNC, vous pouvez également connecter des prises RCA à K3 et K8, par exemple.

L'alimentation ±12 V des amplificateurs opérationnels est fournie par le transformateur TR1, suivi d'un pont redresseur, de condensateurs



Figure 4. Circuit imprimé assemblé du prototype, installé dans son boîtier.



Figure 5. La face avant du filtre coupe-bande.

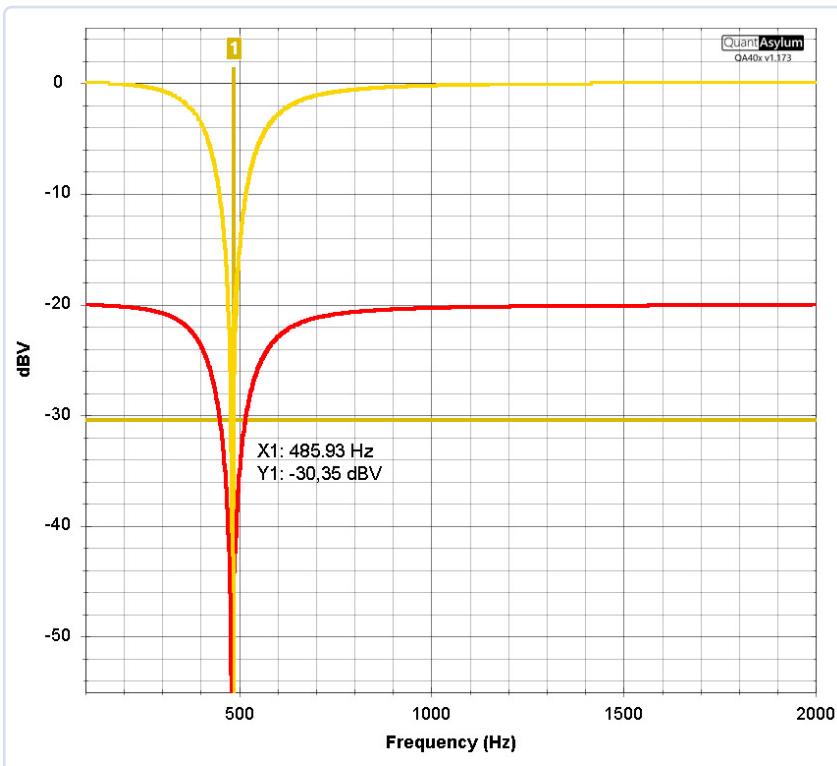


Figure 6. Courbes de réponse en fréquence, sortie 1 (rouge) et sortie 2 (jaune).

de filtrage et de régulateurs de tension. Les LED1 et LED2 indiquent la présence des tensions d'alimentation symétriques. Une LED supplémentaire peut être connectée à K6 comme indicateur de fonctionnement pour le panneau avant. Le circuit imprimé de l'atténuateur commutable est également alimenté en ± 12 V via K7.

Des condensateurs de 100 nF sont placés directement sur les broches d'alimentation des op-amps. Entre les deux se trouvent des condensateurs au tantalum de 22 μ F, destinés à garantir des tensions d'alimentation propres. Un régulateur de tension séparé, IC6, est dédié à l'alimentation en 5 V des relais. Les relais de l'atténuateur sont également alimentés par cette tension via K5.

Construction, composants, et qualité

Un circuit sophistiqué ne suffit pas à lui seul pour obtenir un bon filtre. La qualité globale dépend essentiellement de la qualité des composants sur le trajet du signal. Les résistances concernées sont donc du type Mini

MELF 204, qui peuvent rivaliser avec les résistances à film métallique en termes de qualité. Bien entendu, les résistances conventionnelles (et donc moins chères) à couche mince au format CMS 1206 peuvent également être utilisées. Cependant, les condensateurs doivent toujours être de type COG, car même un seul condensateur X7R sur le trajet du signal gâcherait tout le circuit. Le type d'amplificateur opérationnel que j'utilise est le OPA2210. La **figure 4** montre le circuit imprimé assemblé et installé dans un boîtier Teko, et la **figure 5** montre l'aspect (pas encore parfait) du panneau avant. Il a été conçu à l'aide d'un logiciel de dessin et imprimé sur un film autocollant à l'aide d'une imprimante laser. Les trous ont été percés à l'aide d'emporte-pièces. La **figure 6** montre la réponse en fréquence de la sortie 1 (*Flat/Notch*) en position *Notch* en rouge et de la sortie 2 amplifiée de 20 dB en jaune, enregistrée avec l'analyseur audio QuantAsylum QA403. Le filtre est vraiment à bande très étroite : A deux fois la fréquence de coupure, l'atténuation est déjà presque à zéro. 

Vf: Helmut Müller — 240079-04

LIENS

- [1] A. Rosenkränzer, « filtre coupe-bande de Fliege pour les mesures audio », Elektor 9-10/2022 : <https://www.elektormagazine.fr/magazine/elektor-275/61027>
- [2] A. Rosenkränzer, « atténuateurs pour signaux audio », Elektor Circuit Special 2024 : <https://www.elektormagazine.fr/240176-B-04>



Produits

- QuantAsylum QA403 Analyseur audio 24-Bit www.elektor.fr/20530
- JOY-iT JDS6600 Générateur de signaux & Fréquencemètre www.elektor.fr/18714



À propos de l'auteur

Alfred Rosenkränzer a travaillé comme ingénieur de développement pendant de nombreuses années, 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 alfred_rosenkraenzer@gmx.de, ou contactez Elektor (redaction@elektor.fr)

le système LeoINAGPS

surveillez votre véhicule électrique

Giovanni Carrera (Italie)

Il est essentiel de suivre la consommation électrique d'un véhicule au fil du temps. Toutefois, il est encore plus pertinent de pouvoir analyser cette consommation en fonction des caractéristiques spécifiques de l'itinéraire suivi. Ce projet permet de détecter les paramètres électriques de base du moteur de traction en les corrélant avec le parcours, les pentes et les altitudes, à l'aide d'un récepteur GPS et de Google Maps.

Un de mes amis, à mobilité réduite, utilise un scooter électrique pour gagner en autonomie. Il réside aussi plusieurs mois par an dans un petit village des Apennins ligures, non loin de chez moi. Notre région montagneuse présente de nombreuses montées et descentes, certaines avec des pentes particulièrement raides. C'est pour cette raison que les deux batteries au plomb de grande capacité, spécialement conçues pour la traction électrique, nécessitent des remplacements fréquents. Je me demandais quelle était la consommation du scooter dans différentes conditions et envisageais même d'adopter des batteries plus performantes, telles que celles au lithium-ion. Contrairement aux vélos et scooters électriques équipés de moteurs sans balais, compacts et légers, et de batteries au lithium, ce scooter utilise de robustes moteurs CC à aimant permanent et des batteries au plomb. Le scooter sur lequel les tests ont été effectués est équipé de deux batteries 12 V/36 Ah, récemment remplacées par des batteries de plus grande capacité (45 Ah).

Pour la fonctionnalité GPS, j'ai développé mes propres fonctions pour lire la vitesse, la position et l'heure UTC. J'ai cependant décidé de ne pas utiliser la lecture de l'altitude par le GPS, car sa précision est insuffisante pour évaluer correctement les pentes des montées. Lors des premières compilations du nouveau programme, je me suis rendu compte que la RAM (2 kB) du microcontrôleur ATMega328 de l'Arduino Nano était insuffisante pour un fonctionnement stable. J'ai donc envisagé de le remplacer par un microcontrôleur ATMega32U4 comme celui utilisé par l'Arduino Leonardo qui dispose de 2,5 kB de RAM et d'un UART libre.

Le nom de ce projet est inspiré en partie par Leonardo (la carte est compatible Arduino Leonardo), et aussi par le module INA226 qui mesure le courant et la tension de la charge, ainsi que par le récepteur GPS qui complète le système.



À partir de ces tests, j'ai pu déduire la consommation du scooter, utilisé par mon ami, dans les différentes conditions du trajet, facilement identifiable à partir de la carte Google. Malheureusement, je n'ai pas pu calculer la pente à partir de la coordonnée verticale du GPS en raison de sa précision et de sa résolution limitées. Il existe des capteurs de pression capables de fournir l'altitude avec une résolution de quelques dizaines de centimètres, mais ils n'ont pas été utilisés dans les tests.

Description du système

Le schéma fonctionnel des divers composants du système est illustré dans la **figure 1**.

Le module INA226

Pour la mesure du courant avec un shunt, deux méthodes de connexion sont possibles :

- Vers la masse (côté bas) : Le shunt est connecté entre la charge et la masse.
- Vers la source (côté haut) : Le shunt est connecté entre la source d'alimentation et la charge.

Le circuit intégré INA226, de Texas Instruments, est un dispositif numérique capable de mesurer le courant avec un shunt côté haut ou côté bas. Il mesure également la tension, calcule la puissance et fournit une alarme multifonctionnelle [1]. Le schéma fonctionnel de l'INA226 est représenté dans la **figure 2**.

Le CA/N (ADC) interne est basé sur un convertisseur delta-sigma 16 bits ($\Delta\Sigma$) avec un temps de conversion typique de quelques millisecondes, adapté également aux courants variants rapidement dans le temps. Il permet la mesure des courants négatifs - les nombres négatifs sont représentés au format complément à deux.

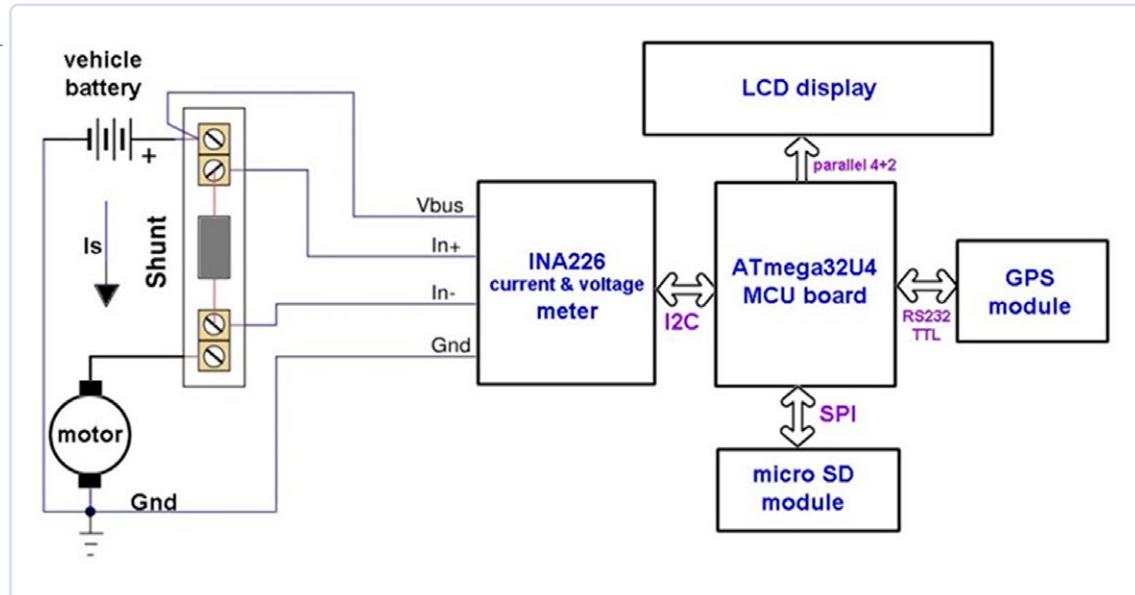


Figure 1. Schéma fonctionnel du système..

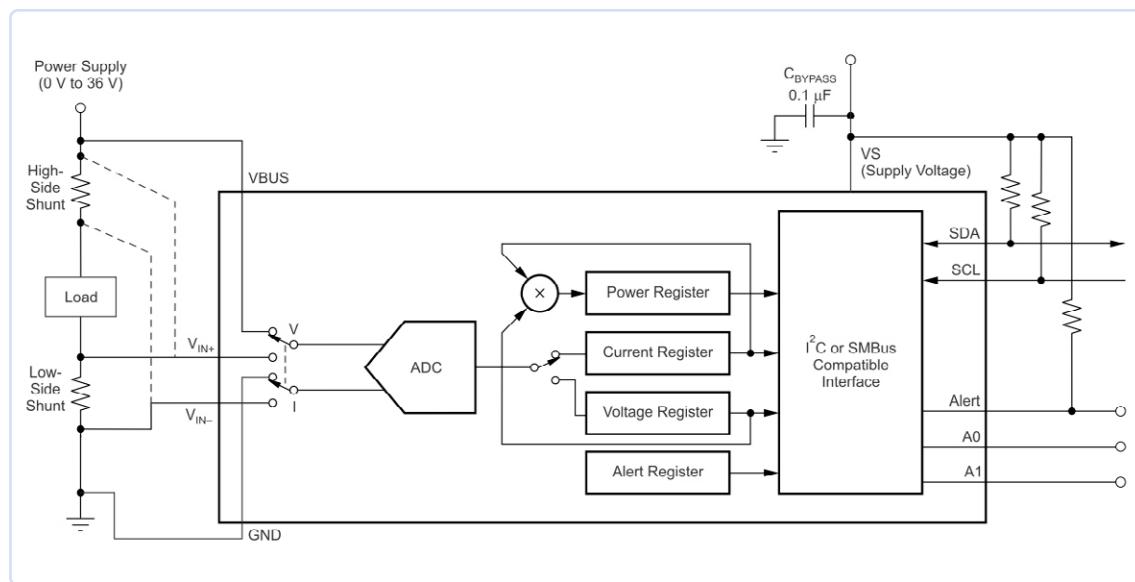


Figure 2. Schéma fonctionnel du INA226.
(Source : Texas Instruments [1])

La résolution de la tension de shunt est de $2,5 \mu\text{V}$ avec une pleine échelle de $32\,768 \times 2,5 \mu\text{V} = 81,92 \text{ mV}$. Pour la tension VBUS, la résolution est de $1,25 \text{ mV}$ avec une pleine échelle théorique de $40,96 \text{ V}$ (ne devant pas dépasser 36 V). La résolution de la puissance est 25 fois celle du courant, la pleine échelle dépendant du shunt utilisé. Le système a donc une précision remarquable.

L'interface entre la puce et le microcontrôleur se fait via le bus I²C. De petits modules (carte breakout) comme celui que j'ai utilisé, visible sur la **figure 3**, sont disponibles sur le marché. Le module comporte deux broches d'adresse, A0 et A1, que l'on peut connecter à V_{CC}, GND, SDA, ou SCL pour définir l'adresse I²C de l'INA226. Le module que j'ai utilisé intègre deux résistances *pull-down*, l'adresse est donc 0x40. Des pastilles à l'arrière permettent de modifier cette adresse, offrant jusqu'à 16 adresses différentes, utile pour surveiller une batterie avec de nombreuses cellules.

Ce module intègre un shunt de $0,1 \Omega$ (R100) qui permet de mesurer un courant maximal de $0,8192 \text{ A}$. Mais j'ai préféré l'enlever, en soudant à sa place un condensateur de $1 \mu\text{F}$ pour filtrer le bruit, et utiliser mon propre shunt (nous y reviendrons plus loin).

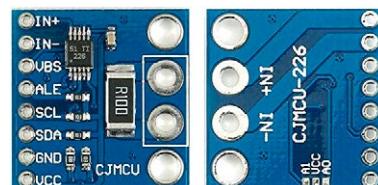


Figure 3. Module INA226.

Le module ATmega32U4 Pro Micro

Le module Pro Micro utilisé est doté d'un microcontrôleur ATmega32U4 et est compatible avec Arduino Leonardo. Il peut être programmé avec l'EDI Arduino en sélectionnant la carte Leonardo. Il est compact : environ $33 \times 18 \text{ mm}$, encore plus petit que l'Arduino Nano, ce qui est similaire à celui d'une vieille EPROM DIL à 24 broches. Si la taille n'est pas une contrainte, une carte Arduino Leonardo peut être utilisée à la place.

La carte utilisée dans ce projet est représentée dans la **figure 4**. Elle a un régulateur de 5 V et peut être alimentée par USB ou par l'entrée RAW (6...12 V). L'alimentation via la broche RAW à un minimum de 6 V fournit une tension V_{CC} de $+5 \text{ V}$ grâce au régulateur. L'alimentation via USB fournit environ $4,8 \text{ V}$ sur RAW, une valeur réduite à cause de

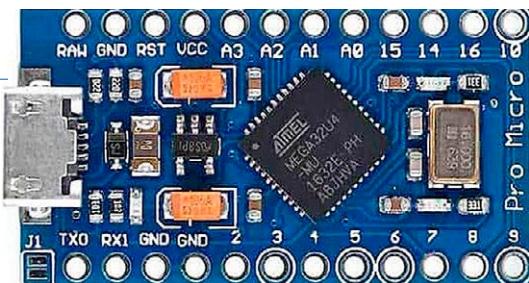


Figure 4. Microcontrôleur ATmega32U4.

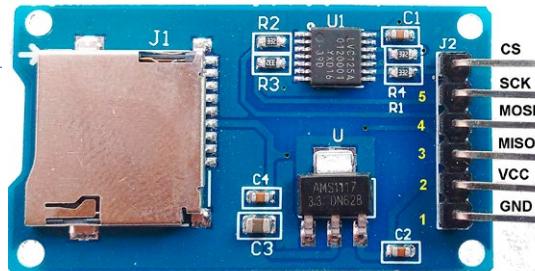


Figure 5. Module microSD.



Figure 6. Afficheur LCD.

la chute de tension dans une diode Schottky qui sert de protection, ainsi qu'un fusible. Si vous souhaitez utiliser le régulateur interne, qui n'a pas de dissipateur thermique, je recommande d'utiliser des tensions basses de 6...7.2 V et d'éviter de connecter des charges supérieures à 100...150 mA. Si vous utilisez un régulateur externe, comme dans notre cas, vous pouvez alimenter la carte avec +5 V sur la broche V_{CC} en excluant le régulateur interne.

Le module MicroSD

J'ai choisi un module SD adapté à Arduino, c'est-à-dire avec des niveaux logiques de 5 V et une alimentation de 5 V (voir **figure 5**). Il est équipé d'un régulateur de 3,3 V et d'un adaptateur de niveau. Les modules dépourvus d'adaptateur de niveau et de régulateur 3,3 V ne doivent pas être utilisés pour ce projet. Pour gérer la carte SD, j'ai utilisé la bibliothèque SD d'Arduino.

L'afficheur LCD

J'ai utilisé un afficheur LCD standard 2x16 caractères équipé d'un contrôleur compatible Hitachi HD44780 et d'un rétroéclairage à LED à haute efficacité, qui consommait environ 20 mA, réduit à 10 mA en ajoutant une résistance externe en série avec la résistance interne. La **figure 6** illustre l'aspect de l'afficheur connecté au microcontrôleur via une interface parallèle avec 4 données et 2 bits de contrôle.

Le module GPS

Sur le marché, on trouve une vaste gamme de récepteurs GPS/GNSS, y compris des modèles abordables. L'important est que le module puisse être alimenté en 5 V. Il est crucial que ces modules puissent être alimentés en 5 V et de connaître leur vitesse de transmission du port série (généralement de 4 800 ou 9 600 bit/s). Certains modules sont équipés d'une antenne intégrée, tandis que d'autres ont une antenne discrète. J'ai utilisé un ancien récepteur 4,800 baud, doté d'une antenne intégrée et d'une base magnétique, que j'ai connecté au système via un connecteur JST à 4 broches.

Schéma du circuit

La **figure 7** (voir page suivante) illustre le schéma du système. En plus du connecteur pour l'interface série, j'ai également monté un interrupteur pour sélectionner les mesures de puissance, avec ou

sans GPS. Le bouton SS sert à démarrer ou arrêter l'acquisition sur la microSD. Le régulateur LM7805 pourrait également être omis si nous utilisions celui intégré à la carte, mais l'utilisation d'un régulateur externe, nécessitant une tension minimale de 7 V et pouvant supporter jusqu'à 24 V avec un dissipateur thermique, offre une plus grande fiabilité. La consommation de courant de l'appareil (avec rétroéclairage) est d'environ 47...50 mA.

La diode Schottky D1 est utilisée pour éviter un conflit potentiel en cas de présence simultanée de l'alimentation externe (5Ve) et de l'alimentation USB. Les résistances R1 et R2, ainsi que le condensateur C5, constituent un filtre passe-bas de premier ordre pour réduire le bruit.

Calcul du shunt

Pour mesurer des courants élevés, il est recommandé d'acheter des shunts commerciaux plutôt que de les fabriquer soi-même. Des gammes de 10, 20, 50, et 100 A ou plus sont facilement disponibles sur le marché et offrent généralement une précision de 0,5% (voir **figure 8**). Une tension de sortie de 75 mV est couramment utilisée pour tous les courants à pleine échelle, très proche de la pleine échelle de l'INA226, qui est de 81,92 mV. Pour mes besoins, j'ai choisi un shunt de 75 mV pour mesurer des courants jusqu'à 50 A, donnant une résistance de :

$$R_{se} = 75 \text{ mV} / 50 \text{ A} = 1.5 \text{ m}\Omega$$

Le courant à pleine échelle devient alors :

$$I_{fs} = 81,92 \text{ mV} / R_{se} = 54.613 \text{ A}$$

La résolution du courant sera :

La puissance maximale, dissipée par le shunt, sera :

$$P = V^* I = V^2 / R = 4.474 \text{ W}$$

Données GPS

Pour analyser les messages NMEA, transmis par le récepteur GPS, j'ai développé des fonctions spécifiques. Le programme lit les phrases NMEA RMC (Recommended Minimum Specific Data) car elles fournissent non seulement les coordonnées géographiques mais



Figure 8. Shunt de 50 A.

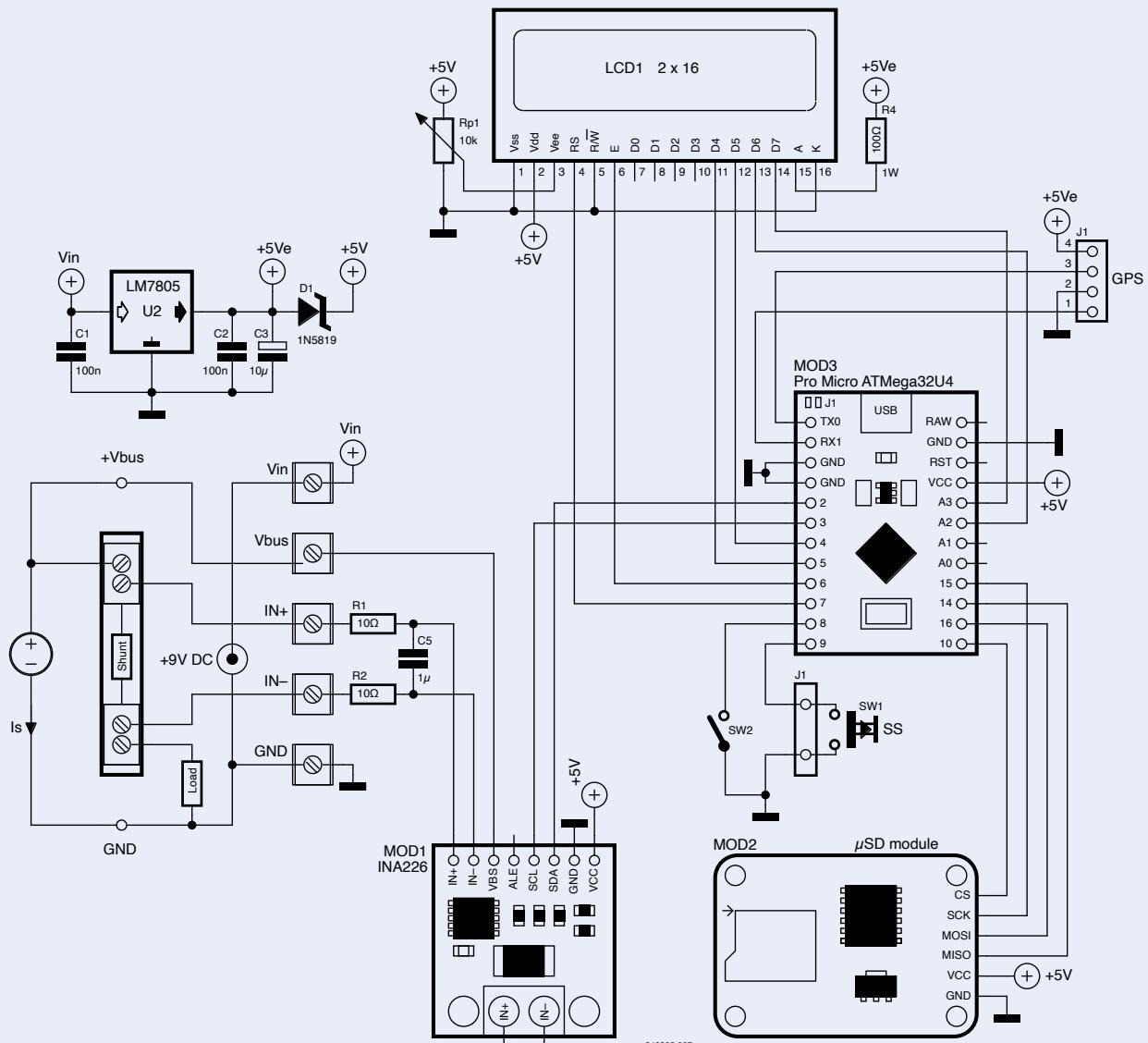


Figure 7. Schéma de câblage du système

Table 1 : The NMEA RMC Fields.

aussi la vitesse en noeuds, ainsi que la date et l'heure. Par exemple :

\$GPRMC,141507.870,A,4429.2796,N,00907.4338,E,0.00,,180522,,*1B

Cela correspond à une heure UTC de 14:15:07.870, des données valides, une latitude de 44°29.2796' N, une longitude de 9°07.4338' E, une vitesse de 0 noeuds, sans direction spécifique, en date du 18/05/2022, sans indicateur de mode. Ces paramètres sont détaillés dans le **tableau 1**.

Les coordonnées sont converties des degrés et minutes en degrés décimaux. La chaîne de données **outdata**, enregistrée sur la microSD, est structurée comme suit :

i	Champs	Nom	Description
1	hhmmss.sss	UTC time	in hhmmss.sss format (000000.000 ~ 235959.999)
2	x	Status	"V" = Navigation receiver warning "A" = Data Valid
3	1111.1111	Latitude	In ddmm.mmmmmm format, with initial zeros included
4	A	N/S indicator	"N"=North, "S"=South
5	yyyyy.yyyy	Longitude	In ddmm.mmmmmm format, with initial zeros included
6	A	E/W indicator	"E"=East, "W"=West
7	x.x	Speed on Earth	Expressed in knots (000.0 ~ 999.9)
8	u.u	Course on earth	Expressed in degrees (000.0 ~ 359.9)
9	xxxxxx	UTC data	In ddmmyy format
10	v	Mode indicator	"N" = Data not valid "A" = Autonomous mode "D" = Differential mode
11	hh	Checksum	Error check, preceded by "*"

,time,latitude,longitude,speed

Pour l'exemple cité, cela se présente comme suit :

,141507,4429.2796,00907.4338,0.00

Logiciel de gestion INA226

J'ai utilisé une version modifiée de la bibliothèque *INA226.h* en raison de quelques imprécisions dans la version originale. J'ai modifié les fonctions

`calibrate(float rShunt, float iMaxExpected) et setShuntVoltageLimit(float voltage).`

Il se peut que l'auteur ait déjà effectué des modifications, mais je recommande d'utiliser la version disponible sur la page web d'Elektor Labs pour cet article [2], car elle a été testée avec ce système.

La fonction `calibrate()` prend en paramètres d'entrée la résistance du shunt et le courant maximum, dans notre cas :

`rShunt = 0.0015 Ω et iMaxExpected = 54.613 A`

Cette fonction calcule plusieurs variables :

➤ `currentLSB = iMaxExpected / 32768 = 54.613 / 32,768 = 1.666 [mA]`
➤ `calibrationValue = 0.00512 / currentLSB / rShunt = 0.00512 / 0.00166 / 0.0015 = 2,048`
➤ `powerLSB = currentLSB * 25 = 0.000625 * 25 = 15.625 [mW]`

Formats des données de l'écran et de la microSD

Compte tenu de la résolution des variables à afficher sur l'écran, voici leurs formats :

➤ Bus voltage: `V=xx.xx` (7 caractères)
➤ Shunt current: `I=xx.xxx` (8 caractères)
➤ Bus power: `W=xxx.xx` (8 caractères affichés)

Ainsi, sur la première ligne, j'entre la tension et le courant, séparés par un espace, et sur la deuxième ligne, j'entre la puissance :

`V=xx.xx I=xx.xxx`

`W=xx.xx`

En cas d'utilisation du GPS, la vitesse en km/h et l'heure UTC sont indiquées sur la deuxième ligne.

Mesures de la consommation d'énergie

La chaîne de caractères `dataString`, qui sera enregistrée sur la SD, est structurée comme suit :

`Vbus,Ishu,power,Vshu`

Où `Vbus` est la tension de la batterie, `Ishu` est le courant circulant à travers le shunt, `power` est leur produit et `Vshu` est la tension aux extrémités du shunt. Si le GPS n'est pas utilisé à bord, ce sont les seules mesures pour chaque échantillonnage ; si le GPS est utilisé, les données relatives à l'heure, à la position et à la vitesse seront également ajoutées.



Figure 9. Scooter électrique équipé du matériel de test, avec mon ami Roberto visiblement préoccupé par les modifications apportées à son véhicule !

Tests du système LeoINAGPS

Pour ces tests, j'ai utilisé le shunt de 50 A mentionné précédemment, qui est de classe 0,5%. Les essais ont été effectués dans un petit village des Apennins ligures, et le système a été monté sur un scooter électrique appartenant à mon ami Roberto (figure 9). Afin de faire de la place à l'appareil, le boîtier situé sous le siège du conducteur a été enlevé. Le trajet débutait dans le jardin de la maison de mon ami et comprenait initialement plusieurs descentes de divers degrés de pente, tandis que le chemin du retour empruntait une autre route marquée par des montées significativement raides. Le parcours, bien que relativement court, comportait des pentes assez raides, testant ainsi les capacités du moteur de ces scooters, majoritairement de fabrication chinoise. Ces scooters sont équipés d'un moteur DC conventionnel d'environ 1 000 W, alimenté par deux batteries au plomb de 12 V conçues pour la traction électrique et connectées en série. La protection est assurée par un fusible de 40 A et un interrupteur thermique de 30 A. Le contrôleur électronique intégré, à base de MOSFET, commande le moteur en mode PWM. Les batteries d'origine avaient une capacité de 35 Ah, mais ont été remplacées par des batteries de 45 Ah pour augmenter l'autonomie sur des terrains accidentés comme celui du test. Sur un terrain aussi vallonné, les batteries ont une durée de vie relativement courte et doivent être changées presque chaque année. En effet, la capacité de 45 Ah est calculée sur une base de décharge de 20 heures ($I = 2,25 \text{ A}$), et pour une décharge de 5 heures, plus conforme à l'usage réel, cette capacité se réduit à 35 Ah ($I = 7 \text{ A}$).

À l'exception du GPS, le système a été monté sous le siège du scooter et au-dessus des batteries, comme le montrent la figure 8 et la Figure 10. Le système LeoINAGPS était alimenté par deux batteries au lithium en série (7,2 V).

Le récepteur GPS a été monté sur un support spécial pour éviter toute obstruction par le conducteur, comme le montre la figure 9. Il intègre l'antenne et le récepteur. Les récepteurs GPS fournissent généralement une mesure de position toutes les secondes, le système doit donc être synchronisé avec la réception des messages NMEA. Chaque mesure, enregistrée sur la carte microSD, se compose de huit



Figure 10. Placement du système

paramètres : tension de la batterie [V], courant délivré [A], puissance [W], tension shunt [V], heure UTC, latitude, longitude et vitesse [km/h]. Les coordonnées sont exprimées en degrés et en dixièmes de degrés. Avant de démarrer l'acquisition en appuyant sur le bouton situé à l'avant de l'enregistreur, il est nécessaire d'attendre que le GPS reçoive et traite les signaux d'au moins trois satellites, ce qui peut prendre quelques minutes. Une fois le GPS est opérationnel, la LED commence à clignoter et l'heure UTC apparaît également sur l'écran. L'heure locale italienne est en avance d'une heure sur l'heure UTC pendant l'été (CEST) ou de deux heures pendant l'heure normale (CET). Lorsque les tests sont terminés, il faut appuyer à nouveau sur le bouton pour terminer l'acquisition. Le test s'est très bien déroulé et a duré 963 s, soit environ 16 minutes.

Mesures

La **figure 11** présente les courbes temporelles de la tension de la batterie, du courant délivré, de la puissance et de la vitesse du véhicule, tous exprimés en secondes. La période où la vitesse est nulle correspond à un arrêt du véhicule pour la prise de la photo visible dans la figure 9. Dans les descentes, le moteur à balais agit comme une dynamo, inversant le courant et provoquant des surtensions de plus de 32 V à plusieurs occasions, ce qui n'est pas idéal pour les batteries. Le graphique de l'historique du courant révèle des informations précieuses, permettant d'identifier clairement les différentes descentes, marquées par des valeurs négatives de courant : la première descente (entre les secondes 89 et 112), suivie d'une légère remontée et le début de la seconde descente (à la seconde 136), puis un arrêt (entre les secondes 148 et 186), et d'autres descentes variées (entre les secondes 189 et 300), avec la troisième (entre les secondes 660 et 683) et la quatrième (entre les secondes 759 et 786) descentes étant les plus longues et les plus intenses.

Comme on pouvait s'y attendre, c'est dans les montées que la consommation est la plus élevée, avec des pics dépassant 800 W. On peut constater les pics de courant au démarrage du moteur ou à l'arrêt du scooter (freins).

Durant le test, la vitesse typique était d'environ 4 à 5 km/h avec des pointes à 6 km/h, comparable à un rythme de marche. ce qui s'explique par l'absence de longs tronçons plats. Le trajet aller s'est achevé à la seconde 372 ; le trajet retour, suivant un itinéraire différent, a été plus longue. Le moment du demi-tour est identifiable sur le diagramme de la vitesse, marqué par une chute de 5,2 à 2 km/h. Cette manœuvre est encore plus visible sur le diagramme de longitude, présenté à la Figure 12. Le scooter a effectué une montée relativement raide, suivi d'une route qu'il a parcouru sur une centaine de mètres avant de faire demi-tour. Cette inversion, qui s'est produite à la seconde 704, est visible sur le tracé de latitude. D'après ces courbes, on constate que les coordonnées du point d'arrivée sont similaires à celles du point de départ.

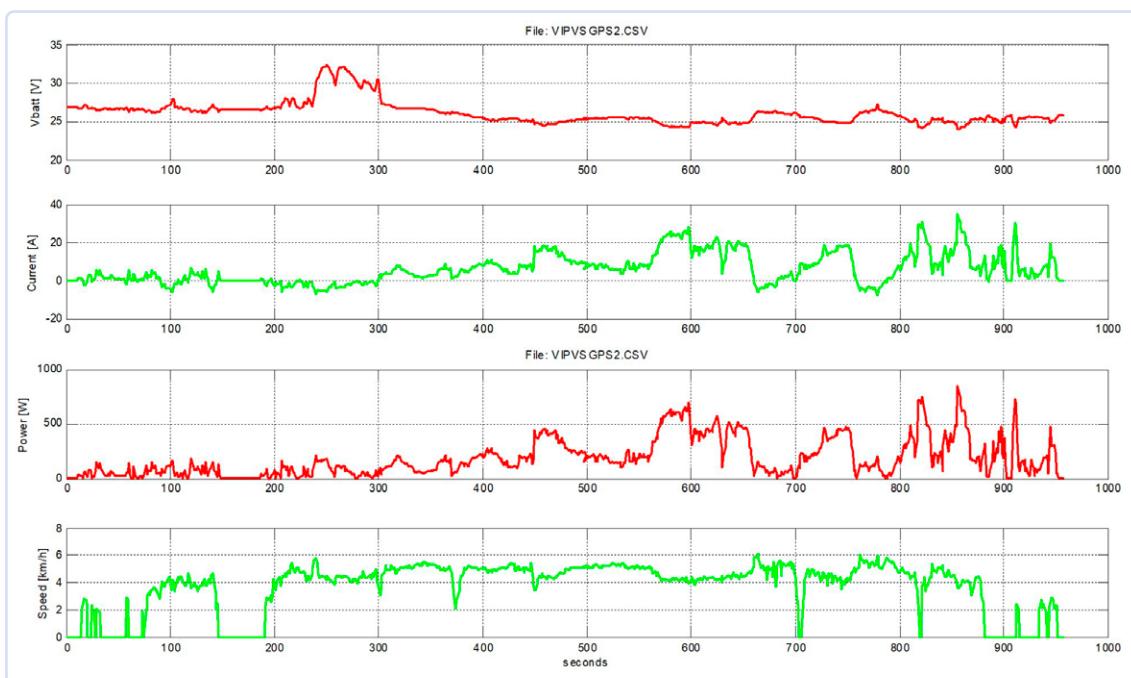


Figure 11. Graphique historique des principaux paramètres.

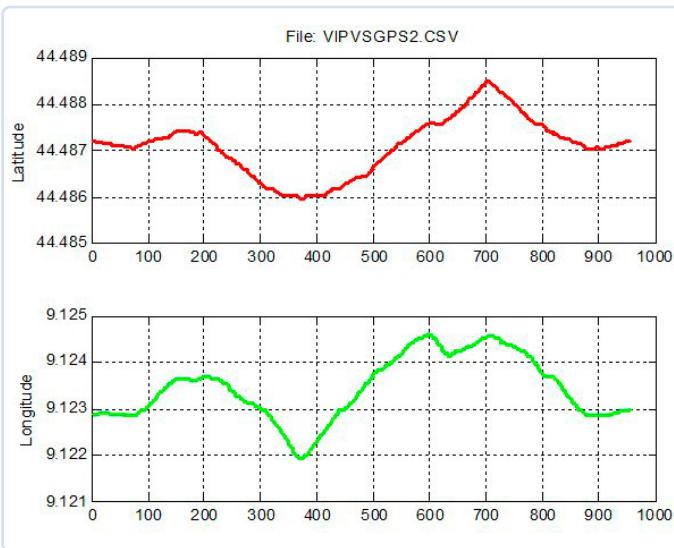


Figure 12. Graphique des coordonnées.



Figure 13. Itinéraire complet de l'aller-retour (Source : Google Maps)

Importer l'itinéraire dans My Maps

Pour visualiser l'itinéraire superposé aux images satellites, on peut utiliser My Maps [3] et Google Maps, qui sont disponibles gratuitement en ligne. Vous devez posséder un compte Google et suivre les instructions ci-dessous :

- Connectez-vous à My Maps sur votre ordinateur.
- Cliquez sur **+CREATE A NEW MAP**.
- Dans la légende de la carte, cliquez sur *Add layer*
- Nommez la nouvelle couche si vous le souhaitez.
- Dans cette couche, cliquez sur *Import*.
- Sélectionnez ou téléchargez le fichier **VIPVSGPS.csv** puis cliquez sur *Select*.
- Choisissez les colonnes pour les latitudes (première colonne), les longitudes (deuxième colonne) et les étiquettes pour chaque point (par exemple, la puissance).

Si vous souhaitez visualiser l'itinéraire avec Google Maps et superposer la photo satellite, il suffit d'importer la couche avec des opérations simples. La **figure 13** montre le résultat, et la **figure 14** montre uniquement le détail de là où l'itinéraire a commencé et s'est terminé. Ces programmes permettent de placer une valeur spécifique à côté de chaque point. J'ai donc choisi d'indiquer la puissance mesurée sur le moteur à chaque instant.

Calcul de l'énergie

Pour évaluer l'autonomie, il est crucial de savoir combien d'énergie a été consommée par les batteries. La quantité d'énergie électrique ΔE fournie par les batteries au moteur dépend de la puissance développée et de l'intervalle de temps $\Delta t = t_2 - t_1$; si la puissance P était constante durant cet intervalle, l'énergie se calculerait facilement comme suit :

$$\Delta E = P \cdot \Delta t \text{ [J]} \quad (1)$$

L'énergie est exprimée en joules (1 J = 1 W/s) dans le système international. Pour les systèmes électriques, on utilise fréquemment les wattheures (Wh), obtenus en divisant les joules par 3 600. Dans notre exemple, la puissance n'est pas constante et varie dans le temps, ce qui nécessite une intégration continue plutôt que discrète, et la formule (1) doit donc être ajustée :



Figure 14. Détail des points de départ et d'arrivée du test. (Source : Google Maps)

$$E = \int_{t_1}^{t_2} P(t) dt$$

Une méthode numérique de base pour effectuer une intégration est celle des trapèzes :

$$E = \int_{t_1}^{t_2} P(t) dt \approx (t_2 - t_1) \frac{P(t_1) + P(t_2)}{2}$$

Ce système est très approxamatif. Pour calculer l'intégrale définie d'une fonction - c'est-à-dire l'aire sous-tendue - on approxime $P(t)$ par de nombreux trapèzes dont l'aire est facile à calculer, comme le montre la **figure 15**. Cette méthode est d'autant plus précise que l'intervalle Δt diminue. Dans notre cas, la période d'échantillonnage, Δt , est d'une seconde, donc :

$$E_2 = (P_1 + P_2) \cdot 1s / 2 + E_1 \text{ [J]}$$

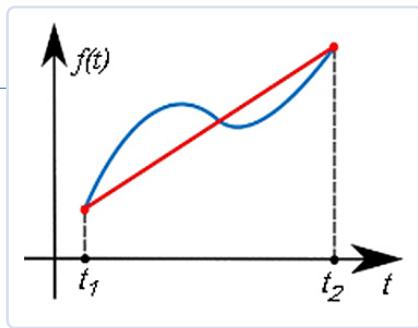


Figure 15. Intégration par la méthode des trapèzes.

Avec un tableur, il est facile de calculer l'énergie en J et en divisant par 3 600, en Wh. Pour estimer l'autonomie approximative du scooter, il faut calculer la perte de charge C des batteries, exprimée en Ah, il faut intégrer le courant, de façon similaire à ce que nous avons fait pour l'énergie, donc :

$$C_2 = (I_1 + I_2) \cdot 1s / 2 + C_1 \quad [\text{Ah}]$$

La charge est liée à l'énergie par la tension ; en effet, l'énergie en [Wh] est obtenue en multipliant la charge par la tension moyenne.

Les résultats des données d'essai, traitées dans Matlab, sont les suivants :

Traitement des mesures du système LeoINAGPS.

Nom du fichier traité = VIPVSGPS2.CSV

Nombre d'échantillons acquis = 963

Période d'échantillonnage = 1.000 [s]

Énergie fournie au moteur = 48.943 [Wh]

Capacité perdue (P&N) = 1.650 [Ah]

Capacité perdue (P) = 1.793 [Ah]

Tension moyenne de la batterie = 26,103 [V]

Courant de décharge moyen = 6,206 [A]

Capacité (E/V moyenne) = 1,875 [Ah]

Dans ce cas, la capacité perdue (positive & négative) est sous-estimée, en partie à cause du Δt trop élevé et en partie à cause des zones où le courant est négatif (pentes), qui ne contribuent pas efficacement à la recharge de la batterie. En fait, elles sont plus nuisibles qu'utiles. La capacité calculée uniquement sur le courant positif (P), la tension moyenne de la batterie et le courant moyen ont également été calculés.

Le programme

Au démarrage, le programme initialise les E/S, l'écran LCD, l'INA226, vérifie si la carte microSD est présente et lit la position du commutateur GPS.

Dans `setup()`, le port série est initialisé à un débit de 4 800 bauds :

```
Serial1.begin(4800); // initialize UART with GPS baud rate
```



Si vous utilisez un module GPS plus moderne, susceptible d'opérer à 9 600 bauds, vous devrez ajuster ce paramètre en conséquence. Lorsque le commutateur est réglé sur GPS, le programme affiche « *Wait for GPS...* » sur l'écran et attend un signal valide du GPS. La fonction `GetData()` extrait l'heure, la position et la vitesse à partir des phrases NMEA RMC.

Lorsque les données sont valides (`Statut` = « A »), la vitesse et l'heure (hhmmss) sont affichées sur la seconde ligne de l'écran. Cette fonction est aussi chargée de convertir les coordonnées, qui sont initialement exprimées en degrés et minutes, en degrés décimaux. ↗

240305-04



À propos de l'auteur

Giovanni Carrera, diplômé en ingénierie électronique, est professeur à la faculté d'ingénierie navale de Gênes, en Italie. Il a enseigné des cours tels que l'automatisation navale et la simulation des systèmes de propulsion des navires. Ayant débuté sa carrière à la fin des années 1970 avec le CPU 6502, il s'est ensuite orienté vers d'autres processeurs. Actuellement, il se passionne pour la conception et le développement de circuits électroniques analogiques et numériques, qu'il partage sur ses blogs (ArduPicLab et GnssRtkLab) et dans divers magazines.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (giov.carrera@gmail.com), ou contactez Elektor (redaction@elektor.fr).



Produits

➤ OPEN-SMART GPS - Module GPS série pour Arduino

www.elektor.fr/18733

➤ PeakTech 4350 Pince ampèremétrique

www.elektor.fr/18161

LIENS

[1] INA226 High-Side or Low-Side Measurement, TI, SBOS547A, August 2015 : <https://www.ti.com/lit/ds/symlink/ina226.pdf>

[2] Ce projet sur Elektor Labs : <http://www.elektor.fr/labs/the-Leo-INA226GPS-system>

[3] Page web de My Maps de Google : <https://mymaps.google.com/>

noeud LoRa alimenté par énergie solaire

une solution IdO modulaire, compacte et polyvalente

Dr. Claus Kühnel (Suisse)

Les nœuds de capteurs fonctionnant de manière autonome requièrent une communication sans fil, et une source d'énergie autonome. Dans cet article nous vous présentons un nœud IdO alimenté par une batterie, tamponné par une cellule solaire, et doté d'une connectivité LoRaWAN.

Étant donné la disponibilité de nombreux modules de capteurs et l'existence d'un support logiciel robuste, il est possible de modifier ce projet selon vos besoins.

Je m'appuie sur l'écosystème WisBlock développé par RAKwireless pour développer un nœud LoRa alimenté par énergie solaire et adapté à la connexion de capteurs. La carte de base RAK19007 WisBlock 2^{ème} génération constitue une excellente base pour intégrer la carte contrôleur WisBlock Core ainsi que différents modules de capteurs. Le RAK4631-R est un WisBlock Core basse consommation basé sur un microcontrôleur nRF52840 de Nordic Semiconductor et une puce LoRa SX1262 de Semtech, offrant une connectivité Bluetooth

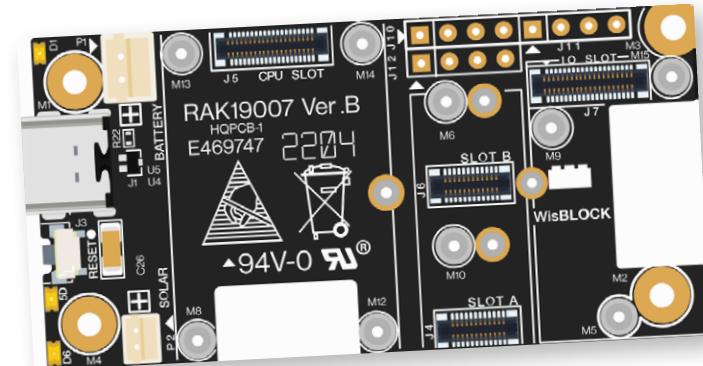


Figure 1. Carte RAK19007 WisBlock Base Board 2nd Gen. (Source : RAKwireless)

LoRa et LoRaWAN

LoRa, pour « Long Range », est une technologie de communication sans fil conçue pour transmettre des données sur de longues distances avec une faible consommation d'énergie, ce qui la rend idéale pour les applications IdO. LoRaWAN, un protocole basé sur LoRa, offre un cadre standardisé et évolutif de gérer la connectivité des appareils LoRa sur de longues distances. Ensemble, LoRa et LoRaWAN permettent de faciliter le déploiement de réseaux économies en énergie à grande échelle adaptés à une multitude d'applications. Cet article explore en détail cette technologie prometteuse.

Low Energy (BLE) et LoRaWAN. (Voir l'encadré « LoRa et LoRaWAN »).

Le RAK4631-R intègre l'interface unifiée RAKwireless V3 (RUI3), ce qui simplifie considérablement le développement de logiciels pour les nœuds IdO. J'utilise l'API RUI3 pour programmer les applications. En outre, je démontre la configuration du dispositif IdO avec l'application WisToolBox. Pour obtenir des informations plus détaillées sur l'écosystème WisBlock, je vous invite à consulter le site web de RAKwireless [1] et mon livre électronique, *IoT-Projects for Makers* [2], qui offre une introduction et des exemples pratiques d'application.

Alimentation électrique

Le RAK19007 est une carte de base WisBlock 2^{ème} génération qui permet de connecter différents modules WisBlock. Il est doté d'un connecteur USB Type-C, de connecteurs pour l'alimentation électrique, et d'une interconnexion avec les modules associés. La **figure 1** montre une vue de dessus d'un RAK19007. La section à l'extrême gauche montre les options d'alimentation disponibles avec les connecteurs JST pour l'alimentation solaire et la batterie ainsi que l'USB-C pour l'alimentation et le chargement de programme. Le module prend en charge l'alimentation par batterie de faible puissance, la charge de la

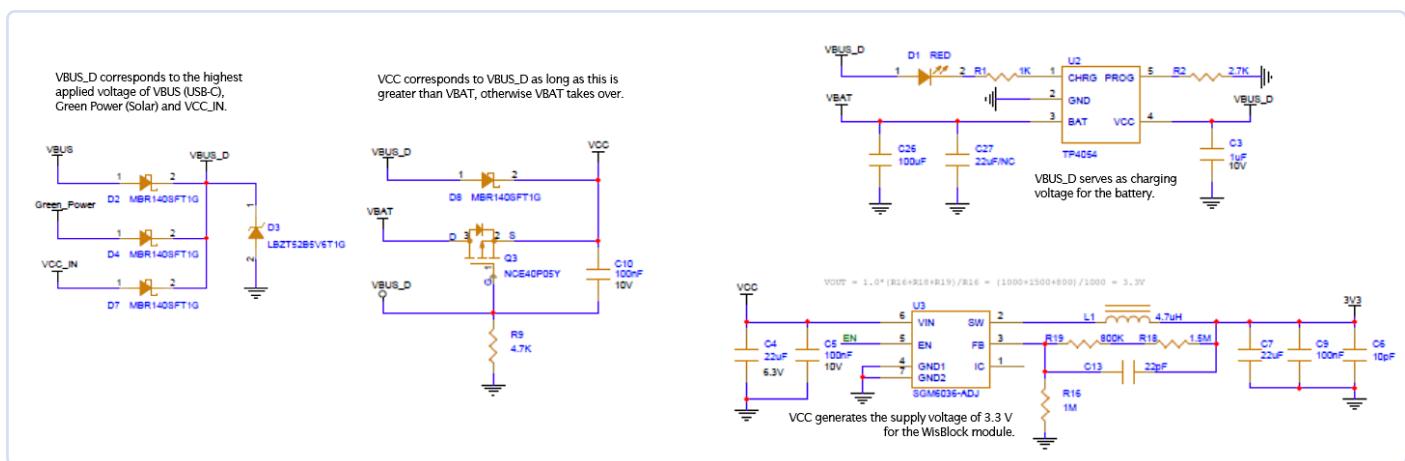


Figure 2. Alimentation du RAK19007. (Extrait du schéma de circuit de la carte de base. (Source: RAKwireless)

batterie au lithium (LiPo) et la charge par panneau solaire d'une batterie LiPo connectée. La figure 2 illustre les principaux éléments de l'alimentation d'un RAK19007. Le RAK19007 peut être alimenté de trois manières : par VBUS fournie via l'USB-C, par l'énergie fournie par une cellule solaire ou par une tension externe VCC_IN. La source ayant la tension la plus élevée est commutée sur VBUS_D. Tant que VBUS_D est supérieure à la tension de la batterie VBAT, VBUS_D détermine la tension VCC. Si VBUS_D est inférieure ou inexiste, alors VBAT détermine la tension d'entrée pour générer la tension système de 3,3 V pour les modules WisBlock, y compris le WisBlock Core, par un convertisseur abaissant de tension à très faible consommation SGM6036.

Gestion de l'énergie

Le RAK19007 ne dispose pas d'un système de gestion de l'énergie comme l'AXP192 ; il faut

donc se baser sur les tensions mesurables pour évaluer le niveau de charge de la batterie LiPo. La tension de la batterie VBAT est mesurée par un diviseur de tension relié à l'entrée analogique AIN0. L'entrée analogique AIN1 est accessible via le connecteur J11 pour mesurer la tension de l'alimentation solaire. Un diviseur de tension est également nécessaire. Dans un test pratique, je mesurerai les tensions `Vsol` et `Vbat` et utiliserai ces mesures pour calculer le rapport `State = Vsol/Vbat`. Si `State > 1`, cela indique que la batterie est chargée et que le panneau solaire fournit de l'énergie à l'appareil. Si `State < 1`, c'est la batterie qui alimente l'appareil, signifiant qu'elle se décharge. Utiliser l'interface RAKwireless Unified Interface V3 (RUI3) rend la création d'une application IoT avec un micrologiciel personnalisé particulièrement aisée. Voici le code nécessaire pour recueillir les valeurs des tensions `Vsol` et `Vbat` :

```
float Vsol = readADC(AIN1);
float Vbat = api.system.bat.get();
float State = Vsol/Vbat;
```

Configuration

Avec le programme *RAK4631-R_Sensor-Test.ino*, j'ai d'abord testé l'ADC interne ainsi que les modules RAK1903 (lumière ambiante) et RAK14001 (LED RGB) dont le schéma fonctionnel est représenté dans la figure 3. Ce programme est disponible sur mon dépôt GitHub [3]. Le module de capteur de lumière ambiante RAK1903 WisBlock surveille les variations de lumière ambiante tout au long de la journée. Avec la LED RGB, le RAK14001 offre une option de signalisation en plus des LED bleues et vertes de la carte de base. Ces deux modules sont facultatifs et non essentiels pour atteindre les objectifs décrits dans l'article.

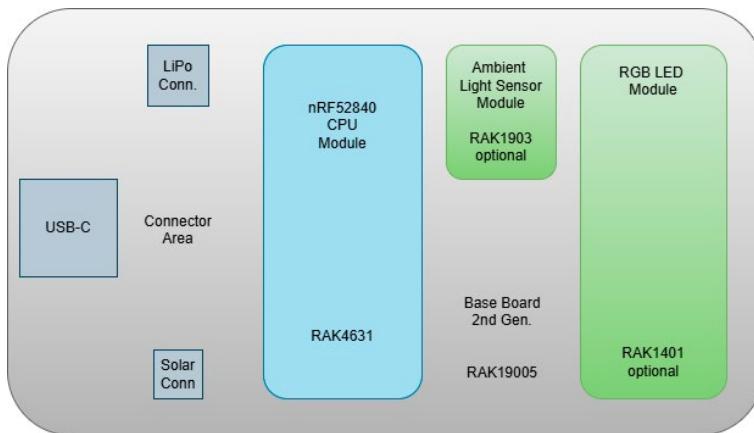


Figure 3. Appareil testé.

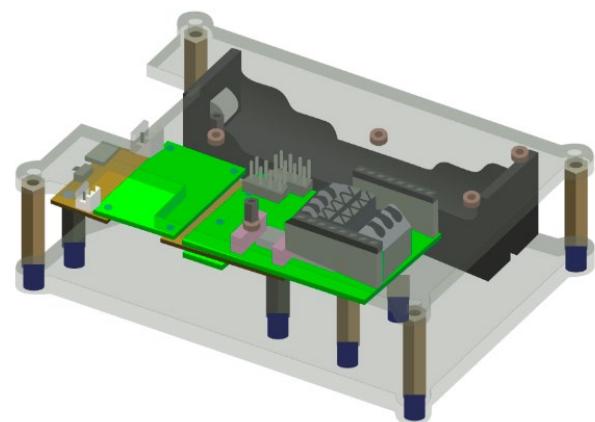


Figure 4. Boîtier acrylique transparent RAKBox-B5. (Source : RAKwireless)



Figure 5. Variantes de la batterie LiPo NCR18650 (Source : RAKwireless)

Pour assurer un fonctionnement sécurisé du dispositif de test, j'utilise un boîtier acrylique transparent RAKBox-B5, bien que celui-ci ne soit pas étanche (**figure 4**). Ce boîtier est avantageux car il intègre un support pour une batterie LiPo de type 18650. Je préfère utiliser ce format de batterie LiPo car il est plus pratique à manipuler en raison de la conception du boîtier. Les batteries LiPo 18650 sont disponibles en diverses capacités, et un aperçu des différents types et fabricants peut être trouvé sur le site [4]. Le support de batterie dans ce boîtier est spécifiquement conçu pour des batteries LiPo de taille « courte ». Référez-vous à la figure 5 pour voir les différentes variantes des batteries LiPo 18650.

Panneau solaire

Selon les tests de batterie publiés sur [4], une batterie LiPo de 3500 mAh et un courant de décharge de 1000 mA à une tension de 2,5 V offre une autonomie de 200 h. Grossièrement estimé, un courant de décharge de 1 mA pourrait permettre une durée d'utilisation estimée à 138 jours. La réduction de la consommation électrique implique que le nœud IoT demeure principalement en mode veille, ne s'activant que pour les mesures et la transmission de données. Pour assurer le fonctionnement autonome du nœud IdO sur une période donnée, il est crucial d'équilibrer la décharge de la batterie. La carte de base RAK19007 est capable de recharger la batterie lorsqu'elle est connectée à un panneau solaire sous une exposition solaire adéquate. Un panneau solaire MicroUSB revolt constitue une solution efficace pour le stockage d'énergie solaire, illustré dans la **figure 6**.

Le panneau solaire MicroUSB revolt illustré à la **figure 6** est composé de cellules solaires

monocristallines, réputées pour leur efficacité et leur longévité. Il fournit une tension de sortie de 5 V à un panneau solaire avec une puissance de charge de 5 W. Avec un câble d'une longueur de 3 mètres, son support de montage flexible et sa protection IP65, il est parfaitement adapté à une utilisation en extérieur. En raison de la tension système de 3,3 V du module de base RAK4631 utilisé, nous avons besoin d'un diviseur de tension pour mesurer la tension du panneau solaire. À cette fin, j'utilise une carte breakout microUSB, comme le montre la **figure 7**, pour ajouter le diviseur de tension. Vsol et GND sont connectés au connecteur JST de la carte de base RAK19007. La tension VAIN1 est acheminée à l'entrée analogique AIN1 sur l'en-tête J11. La **figure 8** montre l'équipement de test complet pour surveiller les tensions du panneau solaire et de la batterie LiPo.

Programmes d'application

Le programme d'application est conçu pour interroger les tensions de la cellule solaire et de la batterie LiPo et transférer les résultats sans fil. Pour réduire la consommation de courant de l'appareil, le noyau RAK4631 doit être mis en veille la plupart du temps. Grâce à l'API RUI3, l'implémentation de ce mode de veille est simplifiée. Les programmes d'application présentés dans les chapitres suivants diffèrent des programmes Arduino standard. La fonction `loop()` exécute en continu les opérations, ici elle sert uniquement à mettre le RAK4631 en veille. Un gestionnaire contrôlé par un timer exécute toutes les fonctionnalités pendant les phases actives. J'ai développé deux versions de l'application : une transmet les mesures via BLE UART et l'autre via LoRaWAN.



Figure 6. Panneau solaire MicroUSB de revolt.

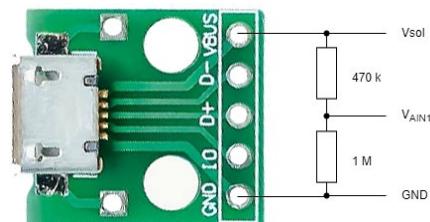


Figure 7. La carte Breakout microUSB.

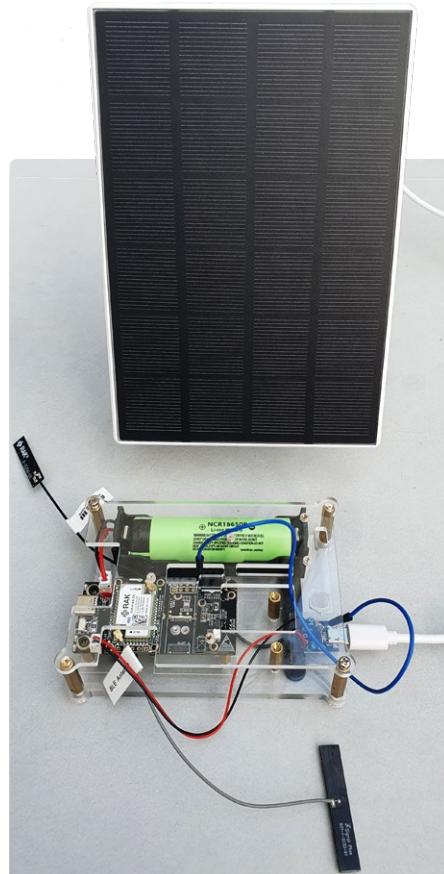


Figure 8. Équipement de test complet.

Application BLE

Après l'initialisation par `setup()`, un gestionnaire contrôlé par un timer interroge les tensions et stocke les données dans un paquet envoyé sans fil via BLE. Comme mentionné, la fonction `loop()` remet le RAK4631 en mode veille. Le programme *RAK4631-R_Sleep_Test_BLE.ino* est disponible sur GitHub [5]. Pendant la période de veille, j'ai mesuré la consommation de courant sur une tension source de 3,8 V de 270 μ A. Pour visualiser les données reçues par BLE, j'utilise l'application Android *Serial Bluetooth Terminal 1.46* [6] de Kai Morich, disponible sur Google Play Store. La **figure 9** montre l'état de la tension après la mise sous tension le soir. State < 1 signifie que la batterie alimente l'appareil et sera déchargée. Le jour suivant était ensoleillé en octobre, State > 1, et le panneau solaire a fourni suffisamment de tension pour alimenter l'appareil et recharger la batterie (**figure 10**).

Application LoRaWAN

Avant de programmer l'application LoRaWAN, j'ai utilisé l'application de bureau *WisToolBox* pour configurer le cœur RAK4631. L'application de bureau *WisToolBox* communique avec les appareils *WisBlock* via USB, et après avoir sélectionné le bon port USB et le module central RAK4631, nous pouvons voir les informations sur l'appareil sur le tableau de bord (**figure 11**). Vous pouvez régler tous les paramètres requis pour la configuration LoRaWAN via les paramètres du menu, comme le montrent les **figure 12** et **figure 13**. Les paramètres définis dans *WisToolBox* sont envoyés au noyau RAK4631 et stockés dans la mémoire Flash. Dans le programme d'application, leur réglage n'est plus nécessaire. Dans mon programme, disponible sur GitHub [7], *RAK4631-R_Sleep_Test_LoRaWAN.ino*, j'ai utilisé la méthode traditionnelle de déclaration explicite. Pendant la période de veille, j'ai mesuré la consommation de courant sur une tension source de 3,8 V de 235 μ A.

The Things Stack

J'utiliserai ici le cluster The Things Stack Sandbox (TTSS) pour l'Europe (eu1), autrefois The Things Stack (Community Edition, TTSCE), comme serveur de réseau LoRaWAN (LNS). Ce cluster, destiné aux tests et expérimentations à petite échelle, est non commercial et open-source, supporté par The Things Network [10].

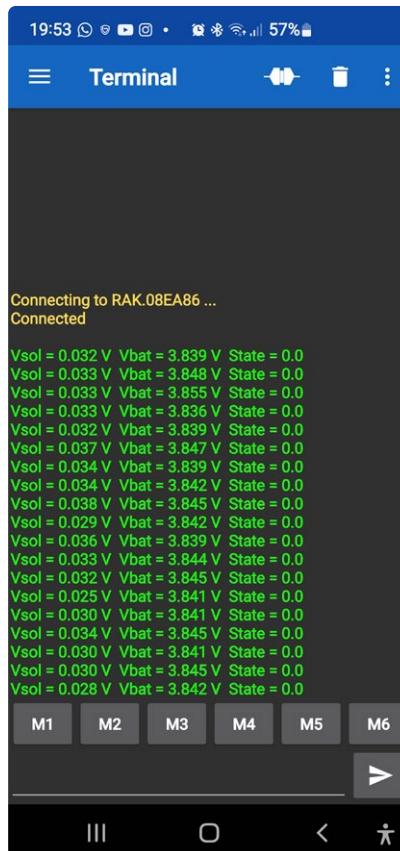


Figure 9. État de la tension après la mise sous tension pendant la soirée.

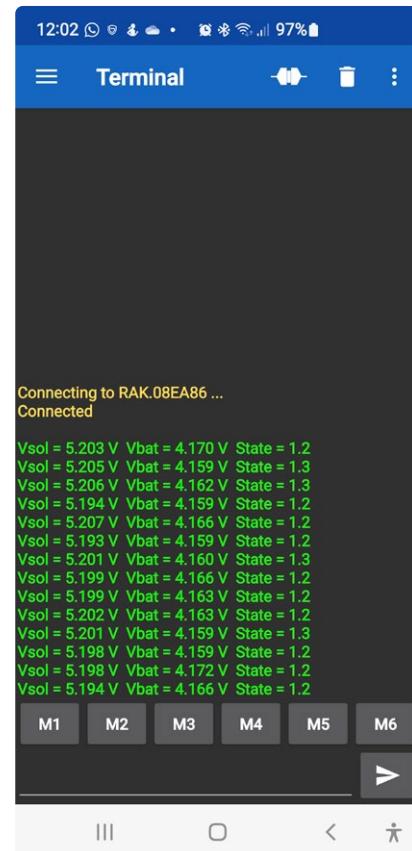


Figure 10. État de la tension lors d'une journée ensoleillée en octobre.

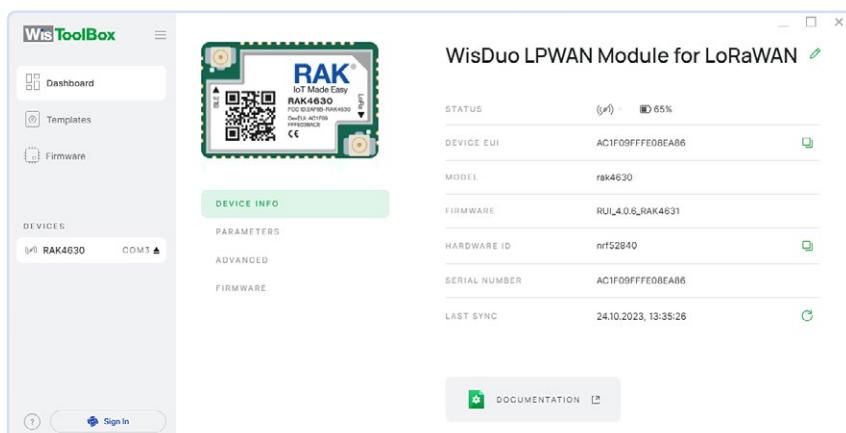


Figure 11. Tableau de bord *WisToolBox* - Informations sur l'appareil.

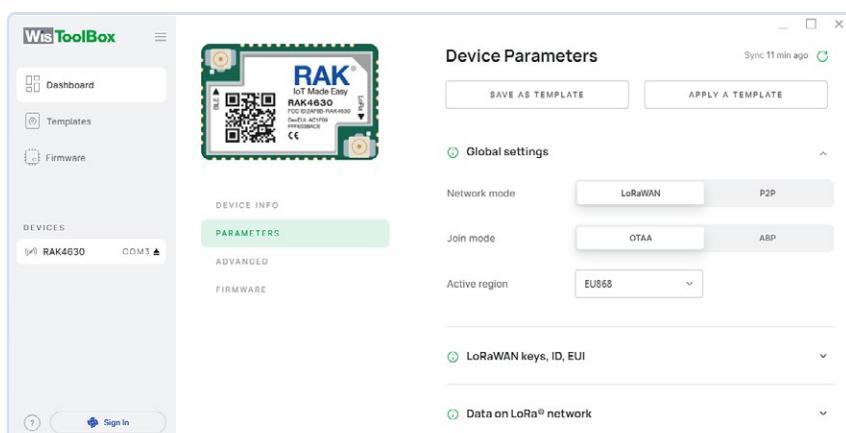


Figure 12. Tableau de bord *WisToolBox* - Paramètres généraux.

TTSS nécessite l'enregistrement de mon appareil ; les clés LoRaWAN, etc. montrées dans la **figure 12** sont nécessaires pour l'enregistrement. Vous pouvez également trouver des détails sur TTSS en tant que serveur de réseau LoRaWAN, et Datacake [8] en tant qu'outil de visualisation, dans mon livre intitulé *Develop and Operate Your LoRaWAN IoT Nodes* [9]. Sur TTSS, une *politique d'utilisation équitable* s'applique, qui limite le temps d'antenne en liaison montante à 30 secondes par jour (24 heures) par nœud et les messages en liaison descendante à 10 messages par jour par nœud. La charge utile et la fréquence du téléchargement déterminent le temps d'antenne sur la liaison montante. Vous pouvez obtenir des informations détaillées sur la fréquence de téléchargement possible à l'aide d'un calculateur de temps d'antenne [11]. Avec une charge utile de cinq octets et un débit de données DR5 (SF7BW125), le temps d'antenne résultant est de 51,5 ms. Entre deux téléchargements, il doit y avoir un temps de 148,2 secondes minimum pour respecter la politique d'utilisation équitable. Le temps de téléchargement dans le programme d'application est de 5 min. Après avoir enregistré et démarré l'application avec succès, la console TTSS affiche les messages de liaison montante reçus (**figure 14**). Le décodeur de charge utile utilisé est également disponible sur GitHub.

Résultats

Les détails de l'événement de cette sortie de console sont listés dans le **listing 1**, qui confirme que les spécifications sont respectées. La bande passante, le facteur d'étalement et le temps d'antenne résultant sont conformes aux attentes et marqués en gras. Pour visualiser les données reçues par le serveur réseau LoRaWAN, j'utilise Datacake, qui peut facilement s'intégrer à TTSS. Étudiez la documentation en ligne [12] ou les exemples dans le livre mentionné ci-dessus. Après cette intégration, vous pouvez créer un tableau de bord Datacake similaire à la **figure 15**. Vous pouvez visualiser les tensions réelles du panneau solaire et de la batterie LiPo et l'évolution de ces tensions sur une journée. Au

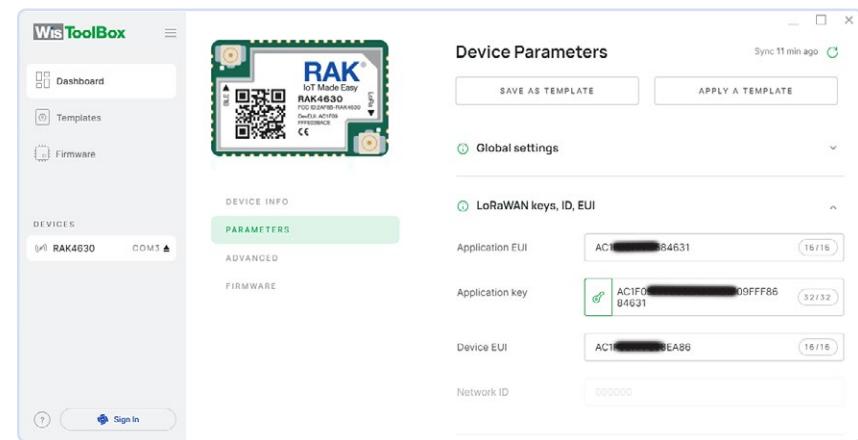


Figure 13. Tableau de bord WisToolBox - Clés LoRaWAN, ID, EUI.



Listing 1. Sortie de la console.

```

"settings": {
  "data_rate": {
    "lora": {
      "bandwidth": 125000,
      "spreading_factor": 7,
      "coding_rate": "4/5"
    }
  },
  "frequency": "867100000",
  "timestamp": 4014400019,
  "time": "2023-10-25T13:28:28.098315Z"
},
"received_at": "2023-10-25T13:28:28.116904692Z",
"consumed_airtime": "0.051456s",
"version_ids": {
  "brand_id": "rakwireless",
  "model_id": "wisblock-4631",
  "hardware_version": "1.0",
  "firmware_version": "1.2.0",
  "band_id": "EU_863_870"
},

```

cours de la journée, le panneau solaire fournit du courant pour charger la batterie LiPo. En début de soirée, la tension du panneau tombe en dessous de celle de la batterie LiPo. La batterie alimente alors le nœud et se décharge

naturellement. Le test s'est déroulé sur plusieurs semaines et a montré que le nœud IdO fonctionnait de manière fiable, même dans des conditions météorologiques défavorables en automne et en hiver. ↗

230668-04



Figure 14. Sortie de la console TTSS (CE).



Figure 15. Tableau de bord Datacake pour le nœud de capteur RAK.



À propos de l'auteur

Claus Kühnel a étudié les technologies de l'information à l'université technique de Dresde. Depuis plusieurs années, il se consacre au développement de systèmes embarqués, notamment pour les équipements de diagnostic en laboratoire. C'est à travers ce secteur interdisciplinaire qu'il a découvert la communauté des « makers ». Au fil de sa carrière, il a rédigé de nombreux articles et publié plusieurs ouvrages consacrés aux matériels et logiciels de microcontrôleurs, tant en Allemagne qu'à l'international. Claus est particulièrement passionné par les innovations en matière de microcontrôleurs.



Produits

- C. Kühnel, *Develop and Operate Your LoRaWAN IoT Nodes* (Elektor, 2022) www.elektor.fr/20147
- LR1302 LoRaWAN HAT pour Raspberry Pi (EU868) www.elektor.fr/20744
- Carte de développement RA-08H LoRaWAN avec RP2040 intégré et écran LCD 1,8" (EU868) www.elektor.fr/20541

Questions ou commentaires ?

Envoyez un courriel à l'auteur (claus@ckuehnel.ch) ou contactez Elektor (redaction@elektor.fr).



SUJET À LA UNE

Visitez notre page **sans-fil & communication** pour des articles, des projets, des actualités et des vidéos. www.elektormagazine.fr/sans-fil-communication



LIENS

- [1] RAKwireless WisBlock system: <https://www.rakwireless.com/en-us/products/wisblock>
- [2] C. Kühnel, IoT-Projects for Makers (Elektor, 2023): <https://www.amazon.com/dp/B0C8VCF4DF>
- [3] RAK4631-R_Sensor-Test.ino sur GitHub : https://github.com/ckuehnel/WisBlock/tree/main/RAK4631-R_Sensor-Test
- [4] 18650 batteries at Akkuline: <https://www.akkuline.de/test/18650-lithium-ion-zelle-vergleich?projektLinId=11>
- [5] RAK4631-R_Sleep_Test_BLE.ino sur GitHub : https://github.com/ckuehnel/WisBlock/tree/main/RAK4631-R_Sleep_Test_BLE
- [6] Serial Bluetooth Terminal app: <https://www.kai-morich.de/android>
- [7] RAK4631-R_Sleep_Test_LoRaWAN.ino on GitHub: https://github.com/ckuehnel/WisBlock/tree/main/RAK4631-R_Sleep_Test_LoRaWAN
- [8] Datacake: <https://datacake.co/>
- [9] C. Kühnel, Develop and Operate Your LoRaWAN IoT Nodes (Elektor, 2022) : <http://www.elektor.fr/20147>
- [10] The Things Network: <https://www.thethingsnetwork.org/>
- [11] Airtime calculator: <https://avbentem.github.io/airtime-calculator/ttn/eu868>
- [12] Datacake online documentation: <https://docs.datacake.de/device/configuration>

AWS

pour Arduino et cie. (2)

transmission de données avec AWS IoT ExpressLink

Tam Hanna (Hongrie)

Dans le premier volet de cette série, nous avons exploré AWS IoT ExpressLink, qui simplifie l'envoi de données vers Amazon Web Services (AWS) à partir de microcontrôleurs. Un module intermédiaire gère les communications, permettant aux microcontrôleurs et aux développeurs de micrologiciels de manipuler ce « module de connectivité basé sur AWS IoT ExpressLink » via des commandes AT, sans se soucier des détails techniques. Nous avons déjà vu comment configurer un module ExpressLink basé sur l'ESP32, créer une « Thing » sur AWS et obtenir les certificats nécessaires. Cet article se concentre sur l'envoi de données depuis un Arduino vers AWS, exploitant les fonctionnalités avancées de la plateforme pour gérer et visualiser les données.

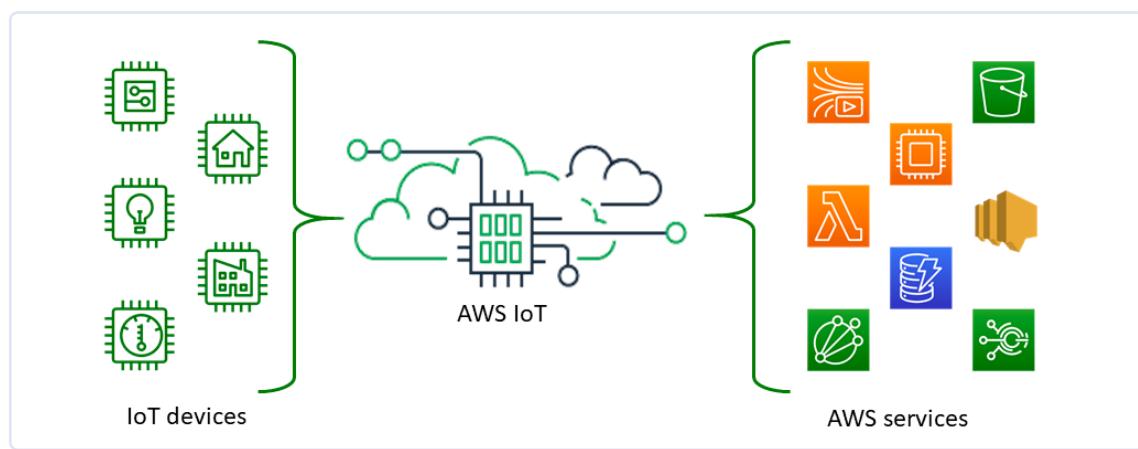
Tout d'abord, il est essentiel de souligner que l'utilisation d'AWS n'est pas une fin en soi. L'intégration d'un module AWS IoT ExpressLink dans votre projet doit viser à exploiter les nombreux services offerts par le portefeuille AWS IoT.

AWS propose une multitude de fonctionnalités (**figure 1**), tandis que la référence [1] offre un aperçu complet de ses capacités. Amazon fournit des détails sur les coûts associés à l'utilisation d'AWS IoT Core dans la référence [2].

Envoi de données

Pour tirer parti de ces fonctionnalités, il est nécessaire de transférer des données vers le cloud AWS. La méthode choisie pour acheminer ces données, souvent désignée sous le terme d'*ingestion* dans le domaine de la science des données, repose sur l'utilisation du protocole de messagerie MQTT.

Dans la première partie de cet article [3], nous avons utilisé un croquis de test d'Espressif et d'AWS pour configurer le module ExpressLink avec des commandes AT, ce qui permet de le rendre accessible par AWS (provisionnement). Il est essentiel que le module soit configuré avec le SSID du routeur pour établir une connexion. Comme mentionné précédemment, vous pouvez facilement saisir les détails de la connexion depuis un smartphone qui communique avec le module via Bluetooth.



Si tout fonctionne correctement, le croquis de test enverra périodiquement des données au serveur AWS :

```
case STATE_CONNECTED:
  if (event == EVENT_CONLOST) {
    state = STATE_PROVISIONED;
    break;
  }
  static unsigned long last_send_time;
  if (millis() - last_send_time >= 10000) {
    response =
execute_command("AT+SEND1 Hello World", 5000);
    last_send_time = millis();
  }
  break;
```

La commande `execute_command("AT+SEND1...")` envoie le message au hub AWS IoT. Toute personne familière avec le protocole MQTT pourrait se demander comment le sujet MQTT utilisé par chaque appareil est déterminé.

La réponse se trouve un peu plus haut dans le listage du code :

```
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);
```

Étant donné que la définition d'un sujet MQTT est sans coût supplémentaire, il est recommandé de modifier le programme d'exemple pour inclure cette commande avant chaque transmission de données au module.

```
if (millis() - last_send_time >= 10000) {
  response = execute_command
  ("AT+CONF Topic1=TEST", 3000);
  response = execute_command
  ("AT+SEND1 Hello World", 5000);
  last_send_time = millis();
}
```

Après avoir apporté ces modifications, le programme sera opérationnel. Il est important de noter qu'une fois alimenté, le micrologiciel de l'Arduino enverra continuellement des messages au serveur, ce qui pourra engendrer des coûts supplémentaires. Par conséquent, il est crucial de ne jamais laisser l'Arduino sans surveillance lorsqu'il est alimenté dans cette configuration !

Pour vérifier la fonctionnalité, nous pouvons utiliser le moniteur série de l'EDI Arduino. Dans la première partie de ce projet, nous avons décrit comment modifier le programme de test pour utiliser le second UART matériel de l'Arduino UNO R4. Ainsi, la console (**figure 2**) vous informera de la réussite de la transmission des paquets à envoyer au cloud AWS.

```
EXC : AT+CONF Endpoint=a3g... iot.eu-west-1.amazonaws.com
EXC : AT+CONF Topic1=TEST
EXC : AT+CONF? SSID
EXC : AT+CONNECT
EXC : AT+CONF Topic1=TEST
EXC : AT+SEND1 Hello World
EXC : AT+CONF Topic1=TEST
EXC : AT+SEND1 Hello World
EXC : AT+CONF Topic1=TEST
EXC : AT+SEND1 Hello World
```

Figure 2. Des messages apparaissent dans le moniteur série de l'EDI Arduino.

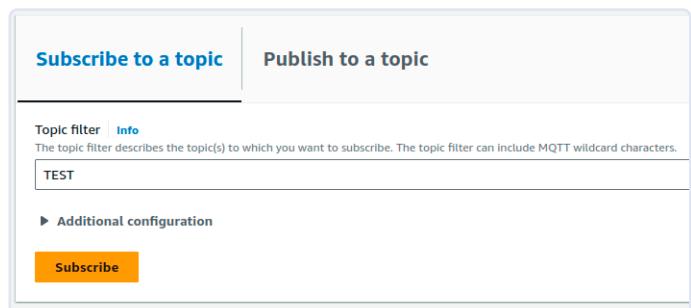


Figure 3. Utilisez le filtre de topic pour vous inscrire

Il est également important de confirmer que les messages sont bien reçus par AWS. Pour ce faire, sélectionnez **Test** → **MQTT Test Client** dans le menu du backend AWS, ce qui vous redirigera vers le client de test AWS. Comme mentionné dans l'article précédent, l'URL dépend de la localisation géographique de vos ressources AWS. Dans ma configuration, URL est la suivante <https://eu-west-1.console.aws.amazon.com/iot/home?region=eu-west-1#/test>.

Après le chargement de la page, il est important que la section **Connection Details** affiche un état **connected** — il arrive que le client fourni par Amazon ait des problèmes de démarrage.

Pour établir la connexion MQTT, l'étape suivante consiste à la configurer en utilisant **Subscribe to a Topic** et le formulaire présenté dans la **figure 3**. Une fois que vous avez cliqué sur **Subscribe**, vous devriez maintenant voir les messages entrants comme le montre la **figure 4**.

Transmission de commandes avancées

Un des avantages majeurs des courtiers MQTT opérés depuis le cloud est leur capacité à répondre directement aux informations reçues, sans nécessité de développer extensivement la logique applicative. Pour tirer profit de cette fonctionnalité utile, les développeurs doivent veiller à ce que les données envoyées soient dans un format compréhensible par le backend. Généralement, cela implique l'utilisation d'une structure similaire à JSON. En plus de la charge utile, telles que les valeurs de mesure, ces messages peuvent contenir des attributs spécifiques, comme l'indication de la température critique d'un équipement dans un format interprétable par AWS, déclenchant une alarme dès que le seuil est dépassé.

Pour implémenter de telles fonctionnalités, il est généralement nécessaire d'utiliser les fonctions avancées du module ExpressLink. La documentation [4] fournit un aperçu du jeu de commandes.

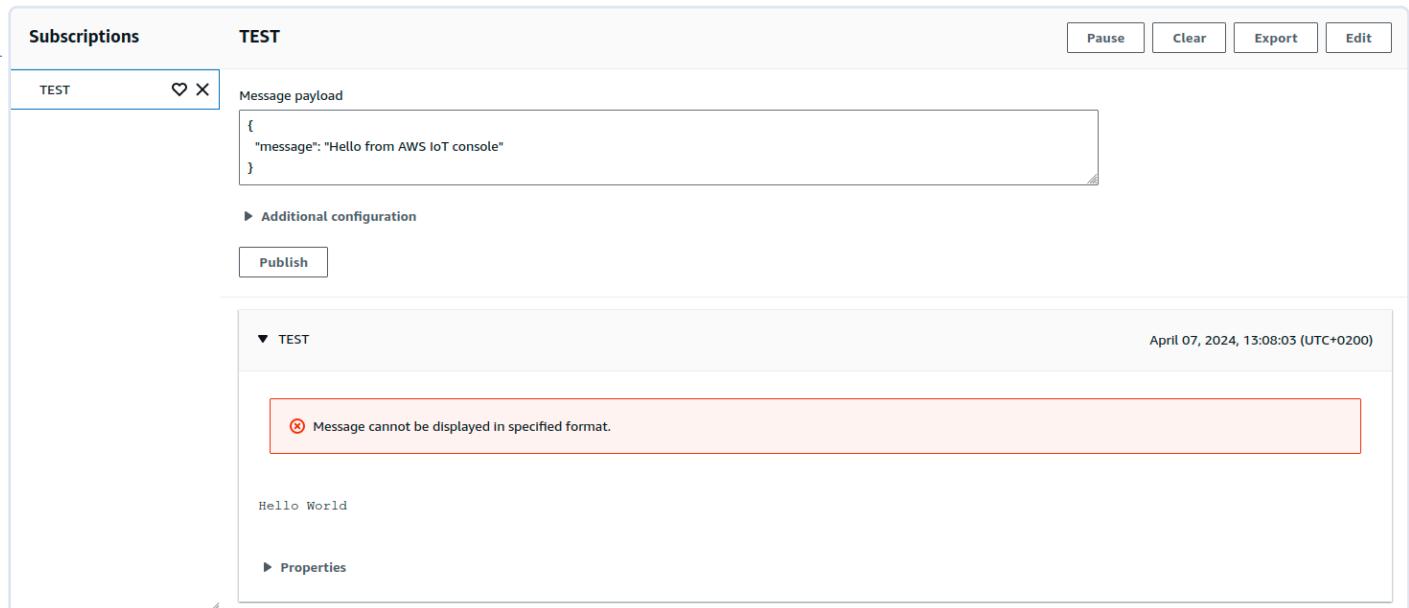


Figure 4. Les messages envoyés par l'Arduino apparaissent dans la console AWS.

Table 3 - Configuration dictionary non-persistent keys				
Configuration Parameter	Type	Initial Value	Buff Size	Description
QoS	R/W	0	1	QoS level selected for SEND commands
Topic1	R/W	{Empty}	≥128	Custom defined topic 1
Topic2	R/W	{Empty}		Custom defined topic 2
...				
Topic<Max Topic>	R/W	{Empty}		Custom defined topic MaxTopic
EnableShadow	R/W	0	1	0 - disabled, or 1 - enabled
Shadow configuration parameters (required only by modules that support the Shadow feature, see 9.2 AWS IoT Device Shadow)				
Shadow1	R/W	{Empty}	64	Custom defined named shadow
...				
Shadow<MaxShadow>	R/W	{Empty}		Custom defined named shadow
BLE configuration parameters (required only by modules that support BLE host control, see 12 Bluetooth Low Energy)				
BLECentral1	R/W	{Empty}	≥ 128	GAP Central discovery/connect configurations.
BLECentral2	R/W	{Empty}	≥ 128	
...				
BLECentral<MaxBLECentral>	R/W	{Empty}	≥ 128	
BLEGATT1	R/W	{Empty}	≥ 128	GATT Characteristic definitions (JSON).
BLEGATT2	R/W	{Empty}	≥ 128	
...				
BLEGATT<MaxBLEGatt>	R/W	{Empty}	≥ 128	
BLEPeripheral	R/W	{Empty}	≥ 128	GAP Peripheral advertising configuration.

Figure 5. La mémoire de configuration du module ExpressLink comporte de nombreux paramètres. (Source : AWS [7])

Syntaxe SQL

À l'instar d'autres systèmes de bases de données relationnelles, AWS utilise sa propre syntaxe SQL spécifique. Pour en savoir plus sur cette variante spécifique à Amazon, veuillez consulter l'URL <https://docs.aws.amazon.com/iot/latest/developerguide/iot-sql-reference.html>.

Une analyse détaillée de la commande d'envoi révèle qu'elle suit le format `SEND[#]` (comme illustré par `SEND1` ci-dessus). Cette structure permet de stocker plusieurs canaux, chacun destiné à être utilisé, dans la mémoire de configuration du module. Chaque canal contient plusieurs dizaines d'attributs, que Amazon décrit de manière exhaustive. La figure 5 présente un extrait du dictionnaire de configuration, illustrant certaines de ces options.

Le format idéal pour la transmission de la charge utile dépend fortement de l'utilisateur. Nous nous concentrerons d'abord sur les messages « facilement analysables » et nous nous conformerons donc à la norme JSON [5].

L'environnement Arduino propose en théorie une variété de bibliothèques facilitant l'assemblage dynamique des charges utiles JSON sans nécessiter une manipulation complexe des chaînes de caractères. Pour nos tests, l'utilisation de telles bibliothèques n'est pas nécessaire. Nous allons simplement adapter la routine d'envoi de notre programme d'exemple, comme indiqué ici :

```
if (millis() - last_send_time >= 10000) {
    response = execute_command
    ("AT+CONF Topic1=TEST", 3000);
    response = execute_command
    ("AT+SEND1 { \"messdaten\":"
     { \"temperatur\": 22, \"kaefer\": 0 } }", 5000);
    last_send_time = millis();
}
```

Pour les développeurs moins familiers avec le langage C, il est important de noter que la séquence `\\" (connue sous le nom de séquence d'échappement en C) garantit que le guillemet est interprété comme faisant partie de la chaîne de caractères et non comme le terminateur de la chaîne.`

À ce stade, la première version du programme est prête à être exécutée. L'analyseur MQTT mentionné précédemment, servira à confirmer la réception des charges utiles transmises.

Règles avec SQL

Dans cette étape, nous plongeons dans un univers de fantaisie en concevant un système de surveillance qui mesure l'humidité et détecte la présence de coléoptères du tabac dans une cave à cigarettes. Ce système nous permettra d'émettre des alertes régulières pour signaler la présence de ces coléoptères du tabac, appelés « kaefer » en allemand :

```
static unsigned long last_send_time;
static int laeufer = 0;
if (millis() - last_send_time >= 10000)
{
```

```
response = execute_command
    ("AT+CONF Topic1=TEST", 3000);
if(laeufer++<10)
{
    response = execute_command
    ("AT+SEND1 { \"messdaten\":"
     { \"temperatur\": 22,
       \"kaefer\": 0 } }", 5000);
}
else
{
    laeufer=0;
    response = execute_command
    ("AT+SEND1 { \"messdaten\":"
     { \"temperatur\": 22,
       \"kaefer\": 1 } }", 5000);
}
last_send_time = millis();
}
```

L'apparition de messages indiquant divers nombres de coléoptères dans le backend AWS confirme le bon fonctionnement du système. La prochaine étape consiste à signaler au backend AWS que la présence de coléoptères dans la cave à cigarettes constitue une donnée significative. Pour cela, nous devons établir une règle spécifique. Le gestionnaire de règles est accessible dans le backend AWS à l'adresse suivante *Manage* → *Message routing Rules*.

Lors de la création d'un compte AWS, un assistant vous guide à travers le processus de création de nouvelles règles, illustré par la **figure 6**. Dans la première étape, l'assistant de configuration vous demande de saisir le nom de la règle. Pour cet exemple, j'utiliserai *TamsKaeferRegel*. Aucune description ou texte n'est nécessaire. Lors de la deuxième étape, nous introduisons une requête SQL qui initie un événement. Pour simplifier, Amazon emploie une variante de SQL spécifique pour définir la règle (comme expliqué dans l'encadré « Syntaxe SQL »). Pour notre première expérimentation, nous utiliserons la requête suivante `SELECT messdaten.kaefer FROM 'TEST' WHERE kaefer>0`. Ensuite, nous procéderons à l'étape suivante.

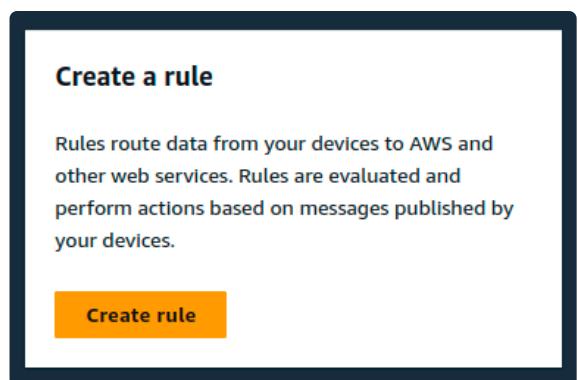


Figure 6. Cet outil peut maintenant être utilisé pour créer les règles.

Basic information Info

Blueprint name

Send email on click of an IoT button

An AWS Lambda function that sends an email on the click of an IoT button.

nodejs18.x ▾

Function name

Enter a name that describes the purpose of your function.

tamsEagKaeferFunc

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime

nodejs18.x

Architecture

x86_64

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#) .

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

 Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Role name

Enter a name for your new role.

tamsEagKaeferFunc

Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates - optional Info

Choose one or more policy templates.



AWS IoT Button permissions 

SNS

Figure 7. Les paramètres doivent être conformes au générateur de règles AWS.

 Successfully created the function tamsEagKaeferFunc. You can now change its code and configuration. To invoke your function with a test event, choose "Test". 

[Lambda](#) > [Functions](#) > tamsEagKaeferFunc

tamsEagKaeferFunc

Throttle

Copy ARN

Actions ▾

Function overview Info

Export to Application Composer

Download ▾

 Diagram  Template



+ Add trigger

+ Add destination

Description

An AWS Lambda function that sends an email on the click of an IoT button.

Last modified

14 seconds ago

Function ARN

 arn:aws:lambda:eu-west-1:326959574411:function:tamsEagKaeferFunc

Function URL Info

-

Figure 8. La bannière verte située en haut de la page nous informe que la fonction cloud AWS a été générée avec succès.

Rule actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. You can add up to 10 actions.

Action 1

Lambda
Send a message to a Lambda function

Lambda function Info
tamsEagKaeferFunc

Create a Lambda function

Lambda function version
\$LATEST

Add rule action

Figure 9. Définition de l'action de la règle sur les messages reçus.

À l'étape 3, Amazon vous invite à définir les actions à effectuer par la règle (Rule Actions), c'est-à-dire les opérations que le backend AWS doit exécuter lorsqu'un événement déclencheur est détecté. À ce stade, nous sélectionnons l'action *Lambda* : il s'agit de sections de code hébergées dans le cloud que le backend Amazon peut exécuter sans nécessiter de machine virtuelle préconfigurée par le développeur. Amazon prend en charge plus d'une douzaine de langages de programmation différents pour ces fonctions Lambda - la documentation est disponible à l'adresse [6].

Ensuite, cliquez sur le bouton *Create a Lambda Function*, ce qui lancera un assistant dans une nouvelle fenêtre de navigateur pour la création de la fonction dans le cloud. À l'étape suivante, choisissez l'option *Use a Blueprint* pour générer différents modèles de projet. Le modèle que nous utiliserons ici est *IT Automation → Send E-Mail on Click of IoT Button*.

Après avoir choisi le modèle de projet souhaité, défilez vers le bas et saisissez un nom dans le champ *Function Name*. Puisque notre fonction va jouer un rôle actif, il convient de lui attribuer les autorisations nécessaires. Dans la section *Execution Role*, sélectionnez *Create a new Role from AWS Policy Templates* et donnez un nom à ce nouveau rôle.

La **figure 7** montre la configuration prévue pour les étapes suivantes. Dans la section *AWS IoT Trigger*, sélectionnez l'option *Custom IoT Rule*. Ensuite, saisissez votre adresse e-mail dans le champ *Environment Variables*. Cliquez ensuite sur *Create Rules*. À ce stade, un problème peut survenir si le champ *Existing Rules* requiert une entrée et que celle-ci n'est pas fournie. Un message d'erreur **This field is required** s'affiche. Pour résoudre ce problème, cliquez sur le bouton *Remove* dans la section *Triggers*, puis revenez à la section *Environment Variables*. Ensuite, après avoir cliqué sur *Create Function*, le processus de création de la fonction cloud commencera. Ce processus peut prendre un peu de temps en raison des latences du backend. La **figure 8** montre la création réussie de la fonction.

Dans l'étape suivante, retournez à l'autre fenêtre du navigateur pour configurer l'action de la règle, comme illustré par la **figure 9**. Cette action est déclenchée lorsqu'une règle correspond à un message entrant. Cliquez sur *Next* pour revoir les paramètres, et enfin, cliquez sur *Create* pour configurer la nouvelle règle dans le backend AWS.

TEST June 08, 2024, 02:33:55 (UTC+0200)

```
{
  "messdaten": {
    "temperatur": 22,
    "kaefer": 0
  }
}
```

Properties

TEST June 08, 2024, 02:33:45 (UTC+0200)

```
{
  "messdaten": {
    "temperatur": 22,
    "kaefer": 0
  }
}
```

Properties

Figure 10. Les messages de test sont reçus avec succès.

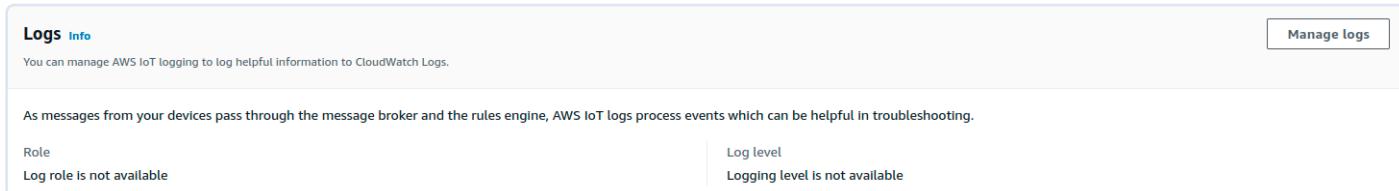


Figure 11. La possibilité d'enregistrer des événements n'est pas disponible ici.

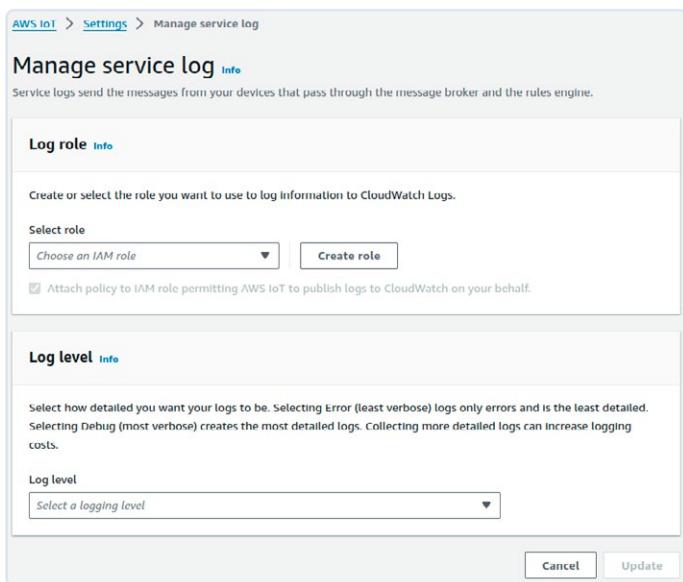


Figure 12. Ces deux paramètres permettent de définir la configuration de la journalisation.

Tout devrait désormais être en place dans le backend, en particulier la nouvelle règle dans la section Message Routing → Rules, qui décrit le « traitement ». Il est temps maintenant de tester le logiciel Arduino. La **figure 10** montre comment les messages envoyés par l'Arduino apparaissent dans le client de test MQTT pour l'évaluation.

Le problème est qu'aucun message n'est reçu à l'e-mail spécifié. Il semble qu'un dépannage soit nécessaire.

Mise en place d'un Log Watcher

Dans le domaine du cloud computing, les fournisseurs doivent trouver un équilibre entre l'enrichissement de leur offre de services et la prévention de configurations par défaut trop exigeantes, susceptibles d'engendrer des coûts supplémentaires et de décevoir les clients, résultant ainsi en une expérience utilisateur négative. C'est pourquoi de nombreuses fonctionnalités sont initialement désactivées par les fournisseurs. Par exemple, la fonction de journalisation est couramment désactivée dans AWS.

Pour activer la journalisation, retournez à la console AWS IoT, cliquez sur *Settings*, puis faites défiler vers le bas jusqu'à la section *Logs*. Lorsque vous accédez à une console AWS fraîchement configurée, il est possible de voir un avertissement semblable à celui illustré par la **figure 11**, indiquant qu'aucun rôle n'est configuré. Une fois les tâches initiales accomplies, vous accédez à l'interface principale de configuration de la journalisation, comme montré dans la **figure 12**. Ici, vous allez configurer le rôle IAM nécessaire pour accéder aux informations et définir le niveau de détail des informations enregistrées. Vous pouvez maintenant cliquer sur le bouton *Create Role* dans le champ *Log Role* pour lancer l'assistant de génération d'un nouveau rôle IAM. Vous avez une certaine liberté pour nommer ce rôle ; dans cet exemple, j'ai choisi *tamsiamlogrole*. Après avoir choisi et confirmé le nom du rôle, une bannière verte s'affiche en haut de l'écran, signalant que la création du rôle a été réussie. Le message « *Policy attached* » apparaît alors sous la combobox, confirmant que le rôle a bien été créé et que les autorisations nécessaires lui ont été correctement attribuées.

Nous pouvons à présent nous rendre dans la section *Log Level*, où nous sélectionnerons l'option *Debug* pour recueillir le maximum d'informations de journalisation. Une fois l'option *Debug* sélectionnée, cliquez sur *Update* pour appliquer les règles de journalisation. Il convient de noter que le backend peut parfois signaler une règle de journalisation manquante, comme le montre la **figure 13**. Dans cette situation, le problème pourrait être lié à la carte de crédit enregistrée. Dans ce cas, accédez à la console générale AWS et vérifiez la validité des informations de votre carte de crédit. Si les problèmes persistaient, il pourrait être nécessaire de créer un nouveau rôle. Une règle correctement configurée est présentée dans la **figure 14**.

À l'étape suivante, il est recommandé de revenir au client de test MQTT et de redémarrer l'Arduino. Prévoyez un peu de temps pour la réception d'une douzaine de messages, car les fonctions de journalisation des fournisseurs de cloud, y compris AWS, ne sont généralement pas immédiates. Avant de procéder aux étapes suivantes, il peut être nécessaire de prévoir jusqu'à une demi-heure de temps d'arrêt - la mise à jour de la liste des sujets contenus dans *Logs* → *Log Groups* se fait très lentement.

L'étape suivante consiste à ouvrir une nouvelle fenêtre de navigateur et à accéder à <https://console.aws.amazon.com/cloudwatch>. Le backend vous redirigera automatiquement vers la console où

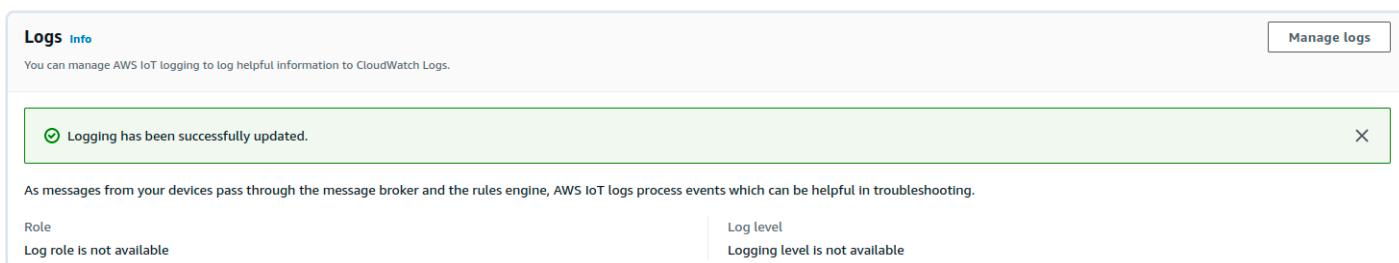


Figure 13. Message d'erreur indiquant que l'enregistrement n'est pas disponible.

Logs **Info**
You can manage AWS IoT logging to log helpful information to CloudWatch Logs.
Manage logs

Logging has been successfully updated.

As messages from your devices pass through the message broker and the rules engine, AWS IoT logs process events which can be helpful in troubleshooting.

Role: tamsiamlogrole Log level: Debug (most verbosity)

Figure 14. Aucun message d'erreur, nous pouvons donc passer à l'étape suivante.

```

2024-06-08T01:22:29.893Z {"timestamp": "2024-06-08 01:22:29.893", "logLevel": "ERROR", "logLevelRaw": "ERROR", "traceId": "e5547082-d4af-9c27-f84b-321e8ffbc5c7", "accountId": "326959574431", "status": "Failure", "eventType": "RuleExecution", "clientId": "iotconsole-55a93753-4dc7-44cf-aaaf-610f7fe1f485", "topicName": "TEST", "ruleName": "TamsKaeferRegel", "ruleAuthor": "OperatorLuiEval", "resources": [{"Operator": ">="}], "principalId": "326959574431", "details": "Undefined result"}

```

Figure 15. Les journaux de TheCloudWatch contiennent de nombreuses informations utiles.

les informations de journalisation CloudWatch de votre compte sont disponibles. Ensuite, sélectionnez le groupe de journalisation AWSIoTLogsV2 utilisé par AWS IoT Core pour stocker les données de journalisation liées à l'IoT. Cliquez sur l'un des flux de données pour afficher les informations de journalisation qu'il contient.

Si vous avez utilisé la même requête que moi, vous observerez probablement des messages d'erreur semblables à ceux présentés dans la **figure 15**, qui signalent des problèmes lors du traitement de l'opérateur. Plus précisément, vous pourriez rencontrer une erreur *OperatorEval*, indiquant qu'une expression non définie a été évaluée dans la condition de la règle. Cela entraîne logiquement l'interruption du traitement de la règle, et les avertissements supplémentaires concernant la non-satisfaction de la condition *Where* sont redondants et peuvent être ignorés.

Pour résoudre le problème, retournez aux propriétés de la règle et mettez à jour la requête SQL en utilisant `SELECT messdaten FROM 'TEST' WHERE messdaten.kaefer>0`. Après avoir enregistré la règle, attendez quelques minutes supplémentaires pour qu'elle s'exécute à nouveau, puis ouvrez les métriques disponibles dans le backend Lambda. La **figure 16** montre des invocations occasionnelles de la fonction Lambda qui deviendront visibles.

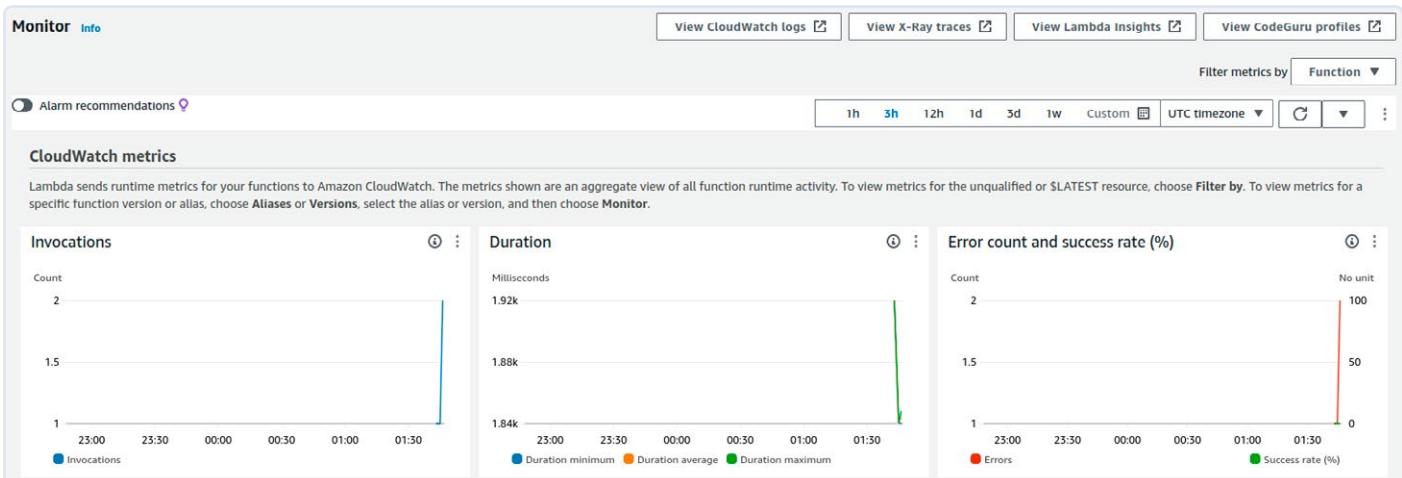


Figure 16. La fonction Lambda démarre mais échoue en cours d'exécution.

```

2024-06-08T01:46:36.786Z 2024-06-08T01:46:36.786Z undefined ERROR Uncaught Exception {"errorType": "RuntimeUserCodeSyntaxError", "errorMessage": "SyntaxError: Identifier 'SNS' has already been declared", "stack": [ "RuntimeUserCodeSyntaxError: SyntaxError: Identifier 'SNS' has already been declared", "at _loadUserApp (file:///var/runtime/index.mjs:1084:17)", "at async UserFunction.js.module.exports.load (file:///var/runtime/index.mjs:1119:21)", "at async start (file:///var/runtime/index.mjs:1282:23)", "at async file:///var/runtime/index.mjs:1288:1" ] }

```

Figure 17. Une erreur de syntaxe évidente dans la définition de la fonction.

Pour une analyse plus détaillée, cliquez sur le bouton *View Cloud-Watch Logs*. Dans le flux de données du journal, vous pourriez observer des erreurs similaires à celles présentées dans la **figure 17** qui signalent des problèmes liés au code comme cette erreur de syntaxe détectée.

L'erreur peut notamment être liée à l'intégration de l'objet SNS responsable de l'envoi des courriels :

```
import { SNS } from '@aws-sdk/client-sns';

const EMAIL = 'tamhan@tamoggemon.com';
const SNS = new SNS();
```

Pour résoudre ce problème, vous pourriez envisager de renommer `SNS` en `mySNS` — bien que ce sujet dépasse le cadre de cet article.

Important : l'utilisation d'AWS Logging peut entraîner des coûts significatifs. Il est donc vivement conseillé de désactiver les paramètres de journalisation de niveau élevé avant de terminer une session.

Une multitude de possibilités

En raison de contraintes d'espace, cet article n'a abordé qu'une fraction des capacités d'AWS IoT. Nous n'avons pas eu l'occasion d'explorer le processus d'envoi de messages depuis le cloud vers un point final, ni de nous pencher sur des visualisations enrichies ou l'intégration de charges utiles de machine learning (ML).

Il convient de noter que de nombreux modules ExpressLink supportent désormais le Bluetooth LE, facilitant ainsi le déploiement d'appareils compatibles Bluetooth. Cependant, cet article s'est concentré sur les principes fondamentaux de l'interaction avec AWS IoT. Nous espérons que cette introduction vous encouragera à expérimenter et à explorer par vous-même les autres fonctionnalités disponibles ! 

240240-B-04



Questions ou commentaires ?

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

À propos de l'auteur

Tam Hanna est un ingénieur spécialisé en électronique, informatique et logiciels depuis plus de 20 ans. Designer indépendant, auteur et journaliste (instagram.com/tam.hanna), il consacre son temps libre à l'impression 3D et à la dégustation de cigares haut de gamme.



Produits

➤ **Arduino Uno R4 WiFi**
www.elektor.fr/20528



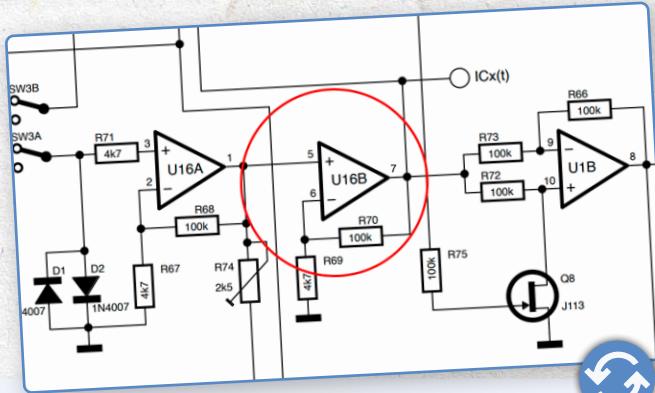
LIENS

- [1] AWS IoT Core: Developer Guide : <https://docs.aws.amazon.com/pdfs/iot/latest/developerguide/iot-dg.pdf>
- [2] AWS IoT Core pricing: <https://aws.amazon.com/iot-core/pricing/>
- [3] Tam Hanna, « AWS pour Arduino et Cie. » (1) Elektor 7-8/2024 : <https://www.elektormagazine.fr/magazine/elektor-349/62995>
- [4] AWS IoT ExpressLink - documentation : <https://docs.aws.amazon.com/iot-expresslink/latest/programmersguide/elpg.html>
- [5] JSON extensions specification : <https://docs.aws.amazon.com/iot/latest/developerguide/iot-sql-json.html>
- [6] Lambda Functions documentation : <https://docs.aws.amazon.com/lambda/>
- [7] AWS IoT ExpressLink documentation: Configuration dictionary non-persistent keys :
<https://docs.aws.amazon.com/iot-expresslink/latest/programmersguide/elpg-configuration-dictionary.html#elpg-table3>
- [8] Téléchargement du logiciel : <http://www.elektormagazine.fr/240240-B-04>

projet 2.0

corrections, mises à jour et courrier des lecteurs

Compilé par Jean-François Simon (Elektor)



LCmètre en circuit

Elektor 5-6/2024, p. 8 (230581)

Une erreur s'est glissée dans le schéma. Nous vous prions de nous en excuser ! Dans le schéma publié, les entrées de l'amplificateur opérationnel U16B sont inversées. La version corrigée est présentée ci-dessus.



Enregistreur universel de données de jardinage

Elektor 7-8/2024, p. 82 (230629)

Capteurs de station météorologiques

Elektor 7-8/2024, p. 24 (240002)

Dans l'article sur l'enregistreur universel des données de jardinage, vous construisez un capteur de pluie, destiné à détecter s'il pleut et à essayer de mesurer la quantité de pluie. Je pense que cela donne un résultat très imprécis, alors que dans le même numéro d'Elektor, à la page 26, il y a un capteur de pluie illustré qui fonctionne bien, je l'ai utilisé, et il a donné des résultats satisfaisants. Ce dernier est disponible sur AliExpress en cherchant « rain sensor ». J'espère que cela pourra aider d'autres lecteurs !

Wim Sanders (Pays-Bas)

Merci pour ce conseil ! C'est une coïncidence intéressante en effet. Nos différents articles proviennent d'auteurs différents utilisant des solutions diverses, ce qui fait aussi partie de la richesse de notre magazine ! Je n'ai jamais utilisé l'un ou l'autre capteur, je ne peux donc pas me prononcer personnellement. Comme vous le dites, il est probable que le modèle commercial ait une meilleure précision.

Jean-François Simon (Elektor)



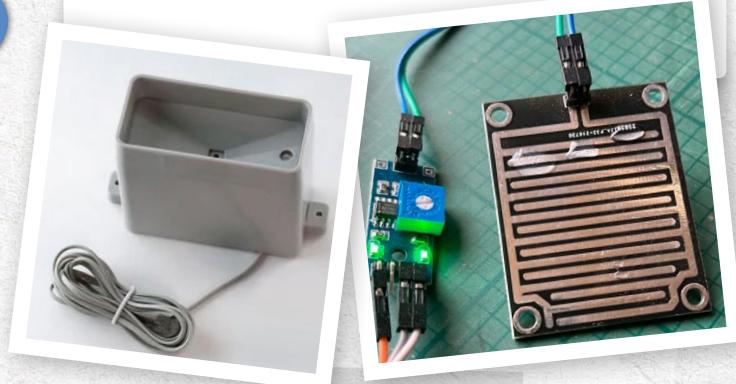
Radios DAB+

Je suis abonné à Elektor depuis plusieurs années, et j'ai vu beaucoup de projets de radio FM. Malheureusement, le service FM va être arrêté ici très bientôt. Serait-il possible d'avoir un projet de radio DAB+?

Pierre-René Beljean (Suisse)

Merci pour cette suggestion. Des projets sont en cours de réalisation, mais ils ne sont pas encore prêts à être publiés dans un numéro en particulier. En attendant, avez-vous vu le projet de récepteur FM/DAB+ [1] que nous avons publié en novembre 2022 ?

Jens Nickel and Clemens Valens (Elektor)



Vous avez une idée originale ou un retour d'information précieux pour Elektor ?
Contactez-nous à l'adresse redaction@elektor.fr. Nous sommes impatients de vous lire !



Détecteur de mouvements HB 100

Elektor 7-8/2023, p. 90 (230205)

Un article intéressant sur le thème de la mesure de la vitesse par effet Doppler. J'ai rapidement eu l'idée de mesurer la vitesse d'un train miniature à l'échelle 0, afin de pouvoir le faire circuler par rapport à la vitesse réelle. Pour ce faire, je devais d'abord me procurer du matériel et effectuer des tests. J'ai lu la fiche technique du HB100 et j'ai trouvé ce qui suit. Une fréquence d'émission de 9,35 GHz devrait être disponible en Allemagne. Cependant, je n'ai pas trouvé de HB100 avec cette fréquence. Tous les fournisseurs indiquent une fréquence de 10,525 GHz, y compris les fournisseurs basés en Allemagne. Normalement, je suppose que le HB100 avec 10,525 GHz n'est pas autorisé à être proposé ou vendu en Allemagne. Pouvez-vous m'indiquer une source d'approvisionnement et éventuellement des systèmes radar comparables avec des bandes de fréquences approuvées pour l'Allemagne ? Merci, je lis Elektor depuis 1976 !

Arno Pelzer (Allemagne)



Volet roulant intelligent

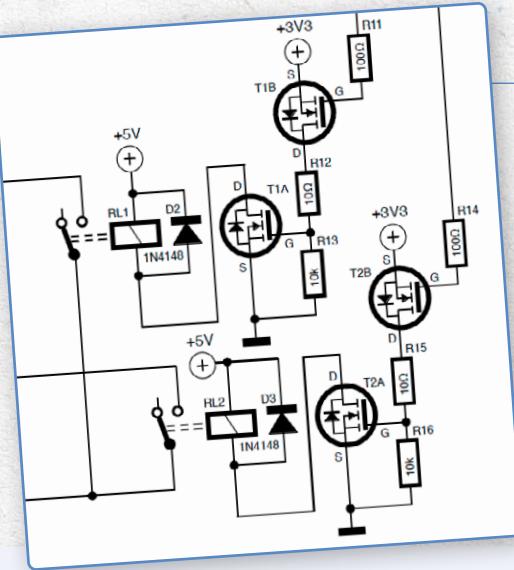
Elektor 7-8/2024 - Édition bonus, p. 6 (230276)

La commande des relais pour le contrôle des volets ne fonctionne pas. En plus, le verrouillage qui empêche d'activer les deux relais en même temps n'est réalisé que dans le logiciel. Ce problème aurait pu être résolu au niveau matériel en utilisant un contact de libre sur les relais de commutation. De plus, les sorties relais ne sont pas protégées par un fusible avec l'Allegro ACS712. On peut se demander s'il est nécessaire de faire l'effort de mesurer la durée de fonctionnement avec le contrôleur ESP. Ce problème peut également être résolu par la configuration. Merci beaucoup pour votre réponse !

Siggi Lehmann (Allemagne)

Vous avez raison ! Merci de nous avoir indiqué cette erreur, qui nous avait échappée lors de la mise en page. Veuillez voir l'extrait corrigé du schéma ci-dessus. Merci également pour vos autres commentaires techniques. Il est évident qu'il y a souvent plusieurs façons de résoudre un problème donné, et que chaque auteur a ses propres solutions privilégiées. En ce qui concerne l'absence de fusibles, vous avez raison. Mais en principe, au tableau électrique principal, le circuit d'alimentation dédié aux volets roulants électriques devrait être de toute façon protégé par son propre disjoncteur. Et cela en France du moins, et j'imagine dans d'autres pays également.

Jean-François Simon (Elektor)



Source de courant continu réglable de précision

Elektor 4/2014, p. 46 (130287)

Je cherche l'article sur le circuit imprimé 130287-1, mais je ne le trouve pas dans vos archives. Je possède maintenant les circuits imprimés terminés et j'aimerais poursuivre le montage.

Werner Ehrhardt (Allemagne)



Nous vous remercions pour votre courriel. Il est vrai que parfois, le moteur de recherche intégré au site web d'Elektor ne donne pas immédiatement ce que vous cherchez. Dans ce cas, je vous recommande d'utiliser un autre moteur de recherche, comme Google, avec les mots clés « 130287 » et « Elektor ». Dans ce cas, vous pouvez trouver l'article à l'adresse [2]. Profitez bien de votre montage !

Jean-François Simon (Elektor)



262,144 façons de jouer au jeu de la vie

Elektor 3-4/2024, p. 56 (230258)

Très beau projet. J'ai commandé une partie des pièces et je vais bientôt terminer la liste. Dans le schéma, il semble qu'il y ait des diodes sur toutes les positions des roues codeuses (ce qui est logique). Pourtant, les diodes ne sont pas répertoriées dans la liste des composants. Je vous remercie pour vos conseils.

Henry J. Becker

Bonjour Henry. Merci pour votre appréciation de ce projet et pour avoir repéré ce manque. Vous avez raison, les diodes sont essentielles dans cette configuration de schéma, mais malheureusement, elles n'ont pas été indiquées dans la liste des composants. Pour la réalisation du montage, vous pouvez vous procurer 36 diodes 1N4148.

Roberto Armani (Elektor)



Réparation d'équipements électroniques

Elektor 5-6/2024, p. 26 (240069)

Lors du dépannage de circuits directement connectés au secteur, tels que le côté « phase » d'une alimentation à découpage, ou d'un appareil directement connecté au secteur sans son propre transformateur d'isolation, un transformateur d'isolation 1:1 est essentiel. Ceci est particulièrement vrai lorsqu'on utilise un oscilloscope ou un autre instrument de test qui possède son propre cordon d'alimentation, avec un conducteur de terre relié au châssis et une fiche avec prise de terre. Ces transformateurs sont assez coûteux et encombrants, car ils doivent supporter la fréquence du réseau de 50 Hz ou 60 Hz, ainsi que la puissance consommée par l'appareil sur lequel on travaille. J'en utilise encore un qui date de quelques années (1964), lorsque de nombreuses radios, tourne-disques et télévisions avaient le châssis directement connecté au secteur, avec le chauffage des tubes à vide connecté en série sur le secteur, et les tensions de plaque redressées B+ ou même multipliées isolées du secteur.

Michael Kiley

Merci beaucoup pour votre courriel. Vous avez raison, les transformateurs d'isolation peuvent être utiles, et moi également j'en possède un et je l'utilise. Cependant, je n'ai pas voulu mettre l'accent sur ces transformateurs dans l'article, car il s'agit d'une autre « boîte de Pandore » qui mériterait son propre article. Selon les fabricants, certains transformateurs peuvent avoir des câblages différents. Parfois, une borne du secondaire peut être reliée à la terre, ce qui va à l'encontre du but recherché. Et parfois, le conducteur de terre est relié à la borne de terre de la fiche ou de la prise de sortie, parfois non, etc. En plus, leur utilisation peut être dangereuse si elle n'est pas correcte. Dans de nombreuses habitations, des disjoncteurs différentiels sont installés. Lorsque l'on teste des circuits sous tension, sans transformateur, dans de nombreux cas, si l'utilisateur touche accidentellement une zone sous tension, le disjoncteur se déclenche et permet d'éviter l'électrocution. Cependant, avec un transformateur, l'utilisateur peut toucher une zone « sous tension » d'une main, sans s'en rendre compte. Si l'utilisateur touche ensuite une autre partie du circuit avec l'autre main (ce qui est de toute façon une très mauvaise pratique, mais les accidents arrivent), il peut être tué et, en raison du transformateur d'isolation, le disjoncteur différentiel ne se déclenchera pas et ne va pas lui sauver la vie. Il s'agit d'un sujet délicat, et chaque technicien en électronique a ses propres préférences et habitudes. Restez en sécurité !

Jean-François Simon (Elektor)



Générateur de référence 10 MH

Elektor 5-6/2024, p. 96 (180385)

Bonjour, l'article de monsieur Rosenkränzer correspond une fois de plus exactement à mes attentes, avec de bonnes idées et une bonne mise en application. Elektor a toujours des projets pertinents en termes de technologie de mesure ou de technologie radio amateur. Super ! Jens Lemkamp, DL1LEP (Allemagne)

Bonjour, monsieur Lemkamp. Nous vous remercions pour votre commentaire, et sommes ravis que vous l'ayez apprécié !

Jens Nickel (Elektor)

VF : Laurent Rauber — 240397-4

LIENS

[1] Yves Bourdon, « Récepteur FM/DAB+ », Elektor 11-12/2022: <https://www.elektormagazine.fr/magazine/elektor-283/61190>

[2] Henry Morizot, « Source de courant continu réglable de précision », Elektor 5/2014 : <https://www.elektormagazine.fr/magazine/elektor-201405/26461>

2024'odyssée de l'IA

accélérateurs d'IA : une comparaison

Brian Tristam Williams (Elektor)

L'intelligence artificielle (IA) connaît une évolution rapide, rendant crucial le choix de l'accélérateur d'IA le plus adapté. Dans cet article, nous explorerons quelques-uns des principaux accélérateurs disponibles, en examinant en détail leurs caractéristiques et performances, pour vous guider vers une décision éclairée, adaptée à vos besoins spécifiques en IA.



Figure 1. Kit Raspberry Pi AI.

Pourquoi les accélérateurs d'IA sont-ils essentiels ? Bien que tout processeur ou microcontrôleur, étant Turing complet, puisse théoriquement réaliser les mêmes calculs qu'un accélérateur d'IA, cependant, ces unités polyvalents sont conçus pour gérer une grande variété de fonctions mais manquent souvent d'efficacité pour les calculs spécifiques à l'IA.

Un accélérateur d'IA est un composant matériel spécialisé conçu pour accélérer les tâches d'intelligence artificielle, notamment durant la phase d'inférence où les modèles prédisent à partir de nouvelles données. Ces accélérateurs se distinguent des processeurs classiques par leur aptitude à gérer efficacement des opérations matricielles à grande échelle et des traitements parallèles, cruciaux pour des applications telles que l'apprentissage automatique, la reconnaissance d'images ou le traitement du langage naturel. Ils sont donc nettement plus rapides et efficaces pour ces tâches que les processeurs classiques. À mesure que l'IA évolue, l'informatique en périphérie, qui consiste à traiter des données directement sur les appareils plutôt que sur des serveurs cloud centralisés, devient cruciale. Cette évolution améliore la réactivité

en temps réel, la confidentialité des données et l'efficacité générale. Pour répondre à ces besoins, divers accélérateurs d'IA en périphérie ont été conçus pour les systèmes embarqués et les PC.

Choisir l'outil adapté à vos besoins spécifiques peut s'avérer difficile, que vous recherchez des solutions puissantes pour des tâches complexes ou des dispositifs légers pour des opérations en temps réel. Nous vous proposons ici une sélection des accélérateurs d'IA les plus performants, en détaillant leurs caractéristiques, leurs applications typiques et leurs performances.

Pourquoi cette comparaison ?

Je traite régulièrement de grandes quantités de données pour des applications telles que la transcription vocale (utilisant par exemple Whisper [1]) et l'édition ainsi que la classification automatique de vidéos d'archives. Pour ces tâches, les solutions pour PC dotées de GPU puissants tels que mon NVIDIA RTX 4070 sont indispensables pour leur puissance de calcul et leur flexibilité. En revanche, pour les tâches de détection locales en temps réel ou en milieu industriel, les solutions embarquées telles que le Raspberry Pi AI Kit ou la Coral

Dev Board sont plus adaptées en raison de leur compacité et de leur faible consommation énergétique. Cette étude comparative vise à aider à choisir les outils les plus adaptés à chaque situation.

Accélérateurs d'IA intégrés

Kit Raspberry Pi AI avec Hailo-8L intégré
Récemment lancé, le kit Raspberry Pi AI [2] intègre l'accélérateur d'IA Hailo-8L au Raspberry Pi 5, offrant ainsi des capacités d'IA avancées dans un format compact et économique. Cette solution représente une option accessible pour intégrer l'intelligence artificielle à divers projets Raspberry Pi.

L'accélérateur d'IA Hailo-8L se connecte au HAT+ M.2 du Raspberry Pi. Comme illustré dans la **figure 1**, il offre une capacité de calcul 13 téra-opérations par seconde (TOPS), ce qui le rend adapté à des tâches telles que la détection d'objets en temps réel, la segmentation sémantique, l'estimation de la pose pour la reconnaissance des gestes, et le repérage facial. La consommation d'énergie réduite du kit est idéale pour les dispositifs alimentés par batterie et sa taille compacte est compatible avec accessoires officiels de la caméra Raspberry Pi. Bien qu'il n'atteigne

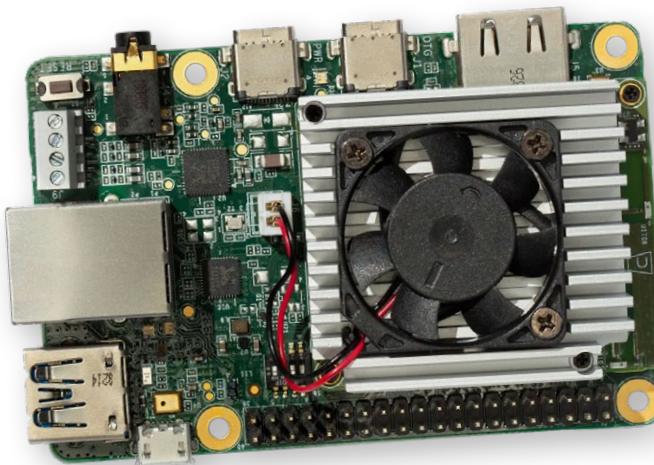


Figure 2. Carte de développement Coral. Source : coral.ai.

pas la puissance de certains NPU haut de gamme, il est parfaitement adapté aux tâches d'IA légères, aux projets éducatifs et aux applications amateurs.

Coral Dev Board

Le Coral Dev Board [3], lancé en 2019, est conçu pour exécuter efficacement les modèles TensorFlow Lite en tant que solution autonome. Présentée dans un format similaire à celui d'un Raspberry Pi (figure 2), il est doté d'un coprocesseur Edge TPU délivrant 4 TOPS. Il est efficace pour la classification d'images, la détection d'objets et la reconnaissance vocale, ce qui le rend idéal pour les applications IdO. Sa faible consommation d'énergie est également un atout majeur pour les systèmes embarqués. Bien qu'elle soit principalement conçue pour les modèles TensorFlow Lite et offre moins de capacité de calcul que d'autres accélérateurs, elle s'intègre parfaitement avec les services Google Cloud, offrant ainsi des performances solides pour des applications d'IA de base dans les systèmes embarqués.

Accélérateur Coral USB

Si vous avez déjà choisi votre plateforme et souhaitez y ajouter des capacités d'IA externe, considérez l'accélérateur USB Coral [4]. Ce dispositif compact, de la taille d'une clé USB standard (figure 3), intègre la puissance de l'Edge TPU de Google à tout ordinateur compatible via USB 3.0. Offrant 4 TOPS tout en consommant seulement 2 W, cet accélérateur est parfait pour les développeurs qui cherchent à ajouter des capacités d'IA aux systèmes existants ou à prototyper des applications d'IA avancées sans recourir à une carte de développement complète. Il excelle dans des tâches telles que la classification d'images et la détection d'objets, ce qui en fait un outil polyvalent pour l'expérimentation et les déploiements d'IA à petite échelle.

NVIDIA Jetson

La gamme Jetson de NVIDIA comprend une série de cartes de développement embarquées conçues pour les applications d'IA en périphérie. Voici une vue d'ensemble de ses principaux produits, chacun adapté à des besoins spécifiques en termes de performances et d'applications :

- Jetson Nano : L'option d'entrée de gamme, adaptée aux petits projets d'IA et au prototypage.
- Jetson Xavier NX : Une option de milieu de gamme offrant des perfor-

mances significativement supérieures à celles du Nano.

- Jetson AGX Xavier : Un module haute performance destiné aux applications d'IA exigeantes.
- Jetson AGX Orin : Le modèle le plus récent et le plus puissant, spécialement conçu pour la robotique avancée et les machines autonomes.

Ces modules ont une puissance de calcul variable, allant de 0,5 TOPS pour le Nano à plus de 200 TOPS pour l'AGX Orin.

Tous les modules Jetson sont compatibles avec le SDK JetPack de NVIDIA, qui comprend des bibliothèques dédiées à l'apprentissage profond, la vision par ordinateur, le calcul accéléré et le traitement multimédia. Cet environnement logiciel courant permet de faciliter la mise à l'échelle et le déploiement au sein de toute la famille Jetson.

La gamme Jetson est bien adaptée à une multitude d'applications, notamment la robotique, les drones, l'analyse vidéo intelligente et les appareils médicaux portables. Son évolutivité en fait un choix polyvalent pour les entreprises qui cherchent à développer et à déployer des applications d'IA à divers niveaux de performance.

Le Jetson Nano, modèle d'entrée de gamme lancé en 2019 (figure 4), a rendu les capacités de calcul de l'IA accessible dans des formats petits et abordables. Équipé d'un GPU Maxwell de 128 cœurs, d'un CPU ARM Cortex-A57 quadricœur et de 4 Go de RAM, il offre une capacité de calcul de 472 GFLOPS (0,47 TOPS). Le Jetson Nano convient particulièrement aux tâches d'IA légères mais exigeantes dans les systèmes embarqués. Capable de gérer plusieurs réseaux neuronaux en parallèle, il se présente comme une



Figure 3.
Accélérateur USB
de Coral



Figure 4. Module Jetson Nano de NVIDIA. Source : nvidia.com.



Figure 5. Intel Neural Compute Stick 2 (Intel NCS2).

solution flexible pour diverses applications d'IA. Cependant, sa consommation d'énergie est plus élevée que celle des autres modules embarquées, et son facteur de forme est plus grand.

Le Nano est un module d'IA doté d'un connecteur à 260 broches, ce qui rend son utilisation autonome impossible sans connexion externe. Pour démarrer, il est nécessaire de se procurer un kit de développement, disponible dans l'encadré "Produits".

Pour découvrir les autres cartes de la série Jetson, n'hésitez pas à consulter la page de comparaison de NVIDIA à l'adresse indiquée [6].

Intel Neural Compute Stick 2 (NCS2)

L'Intel Neural Compute Stick 2 (NCS2), lancé en 2018, est un accélérateur d'IA portable qui se connecte via USB [7]. Doté du VPU Intel Movidius Myriad X, il offre une performance d'environ 1 TOPS. Il est idéal pour le prototypage d'IA, la recherche l'augmentation des capacités d'IA des PC existants. Sa portabilité et sa simplicité d'utilisation en font une option prisée par les développeurs et les chercheurs. Cependant, il nécessite un PC hôte pour l'alimentation et le traitement des données et sa capacité de calcul reste inférieure à celle des GPU de haut de gamme.

Accélérateurs d'IA pour PC

NVIDIA GPUs

Le NVIDIA RTX 4070 [8] de la série RTX 40 lancée en 2022 et constitue une référence pour les jeux et les applications profes-

nelles d'IA. Ce GPU haut de gamme, intégré dans les cartes graphiques de divers fournisseurs, offre 29,8 TFLOPS (29 800 GFLOPS) et 12 Go de mémoire GDDR6. Il est parfait pour l'entraînement et l'inférence de grands réseaux neuronaux, les jeux à haute performance et les tâches de création de contenu telles que l'édition vidéo et le rendu 3D. Sa prise en charge logicielle étendue et la puissance de calcul exceptionnelle en font l'outil idéal pour les applications d'IA exigeantes, notamment la transcription de la parole en texte et le montage vidéo. Pour ceux disposant d'un budget plus conséquent, la RTX 4090 offre encore plus de puissance, tandis que la RTX 3080 représente une alternative économique avec des performances toujours impressionnantes. Cependant, en raison de sa consommation électrique élevée et de son coût, il est plutôt destiné à équiper des stations de travail de bureau de taille importante.

La série RTX NVIDIA promue sous le slogan "the ultimate in ray tracing and AI", représente un produit haut de gamme. Un inconvénient à considérer est que lorsque vous vous absentez de votre station de travail, vous pourriez retrouver un enfant impatient à la maison, pressé de jouer au dernier jeu de tir à la première personne (FPS) bénéficiant du taux d'images par seconde (FPS) élevé que ces machines puissantes peuvent offrir.

Puces Apple de la série M

Les puces Apple de la série M représentent une avancée significative pour les utilisateurs de l'écosystème Apple. Introduite en

2020, la puce M1 a été la première conçue spécifiquement pour le Mac, suivie par la puce M2 en 2022. Chacune intègre directement l'accélération de l'IA dans le processeur : la M1 est équipée d'un moteur neuronal à 8 cœurs capable de délivrer 11 TOPS, tandis que la M2 propose un moteur neuronal à 16 cœurs offrant jusqu'à 15,8 TOPS. Ces puces sont intégrées dans des appareils tels que le MacBook Pro et le Mac Mini, permettant un traitement IA local pour des applications comme Siri, le traitement d'images et la réalité augmentée. L'intégration transparente avec macOS et iOS rend ces puces particulièrement performantes pour les applications d'IA destinées au grand public, surtout pour ceux déjà intégrés dans l'écosystème Apple.

Les M3, M3 Pro et M3 Max ont été annoncées en 2023 avec des améliorations sur toute la ligne. Les puces M3 étaient les premières puces à 3 nanomètres et étaient dotées d'une nouvelle architecture GPU qui permettait des améliorations de vitesse allant jusqu'à 30 % et 50 % par rapport à la M2 et à la M1, respectivement. Avec des performances CPU plus rapides, un moteur neuronal plus efficace et une architecture GPU de nouvelle génération qui prend en charge des fonctions avancées telles que la mise en cache dynamique et le ray tracing accéléré au niveau matériel, la série M3 est particulièrement bien adaptée aux tâches intensives telles que le traitement graphique haut de gamme, les charges de travail AI/ML et les tâches informatiques complexes.

La peinture était à peine sèche sur l'annonce de la M3 quand Apple a révélé la puce M4 en mai 2024, disponible exclusivement dans l'iPad Pro de la marque. Il serait fascinant de voir si un développeur tiers trouve une utilité pour un iPad Pro onéreux dans un contexte commercial ou industriel, notamment pour des tâches d'apprentissage automatique.

Pour ceux qui travaillent principalement avec des produits Apple, un MacBook équipé d'une puce M2 ou M3 [9] ne manquera pas de répondre à vos besoins en applications d'IA. Le choix de la puce dépendra de votre budget et des exigences spécifiques de votre application.

Cartes intégrées ou basées sur PC : Que choisir ?

Le choix entre des solutions informatiques intégrées ou basées sur des PC dépend largement des exigences spécifiques de votre application.



IA embarquée en périphérie

- Idéale pour : les applications en temps réel à faible consommation d'énergie où la taille et la consommation d'énergie sont essentielles, telles que les appareils domotiques intelligents, les appareils portatifs et les capteurs autonomes.
- Avantages : consommation d'énergie réduite, taille compacte, rentabilité

IA en périphérie sur PC

- Idéal pour : les applications nécessitant une puissance de calcul et une flexibilité élevées, telles que les applications d'IA de bureau, l'analyse de données complexes, ainsi que le développement et le test de modèles d'IA.
- Avantages : performances élevées, plus grande flexibilité, capacité à gérer des tâches plus complexes.

Faire le choix

Les solutions IA embarquées et basées sur PC offrent des avantages uniques. Les accélérateurs intégrés conviennent idéalement aux applications en temps réel nécessitant une faible consommation d'énergie, tandis que les solutions basées sur les ordinateurs de bureau et portables, telles que les GPU NVIDIA RTX et les puces Apple de série M, assurent des performances supérieures pour les tâches plus complexes. En saisissant les atouts et les contraintes de chacune, vous pouvez choisir la solution la plus adaptée à vos exigences particulières, maximisant ainsi le potentiel de l'IA pour créer des systèmes plus intelligents et plus performants.

Cette comparaison n'est certes pas exhaustive, mais elle souligne les caractéristiques de quelques-uns des accélérateurs d'IA les plus populaires que j'ai testés. Chaque option présente des avantages et des inconvénients

distincts. L'important est de sélectionner l'accélérateur qui s'adapte le mieux à vos besoins spécifiques. Pour des informations techniques plus détaillées et des comparaisons approfondies, veuillez consulter l'encadré **Liens**.

230181-H-04



À propos de l'auteur

Brian Tristam Williams s'est passionné pour les ordinateurs et l'électronique dès l'âge de dix ans, lorsqu'il a découvert son premier micro-ordinateur. Son parcours avec Elektor Magazine a débuté à seize ans, lorsqu'il a acquis son premier numéro. Depuis, il reste immergé dans le monde de l'électronique et de l'informatique, explorant et apprenant de manière continue. Il a rejoint Elektor en 2010. Il se consacre aujourd'hui à l'étude des dernières innovations technologiques, avec un intérêt particulier pour l'intelligence artificielle et les ordinateurs à carte unique comme le Raspberry Pi.

Questions ou commentaires ?

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



Produits

- **Raspberry Pi AI Kit**
www.elektor.fr/20879
- **Google Coral USB Accelerator**
www.elektor.fr/19366
- **Waveshare Jetson Nano Development Kit Lite**
www.elektor.fr/20761



LIENS

- [1] Whisper speech-to-text from OpenAI: <https://openai.com/blog/whisper>
- [2] Raspberry Pi AI Kit: <https://raspberrypi.com/products/ai-kit>
- [3] Coral Dev Board: <https://coral.ai/products/dev-board>
- [4] Coral USB Accelerator: <https://coral.ai/products/accelerator>
- [5] Apple M3 chip series:
<https://apple.com/za/newsroom/2023/10/apple-unveils-m3-m3-pro-and-m3-max-the-most-advanced-chips-for-a-personal-computer>
- [6] Nvidia Jetson series of embedded compute: <https://developer.nvidia.com/embedded/jetson-modules>
- [7] Intel Neural Compute Stick 2 (Intel NCS2): <https://intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html>
- [8] NVIDIA GeForce RTX 40 Series GPUs: <https://nvidia.com/en-us/geforce/graphics-cards/40-series>
- [9] Apple Mac models, currently based on M2 and M3 chips: <https://apple.com/mac/compare>

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.

FNIRSI HRM-10 Testeur de résistance interne



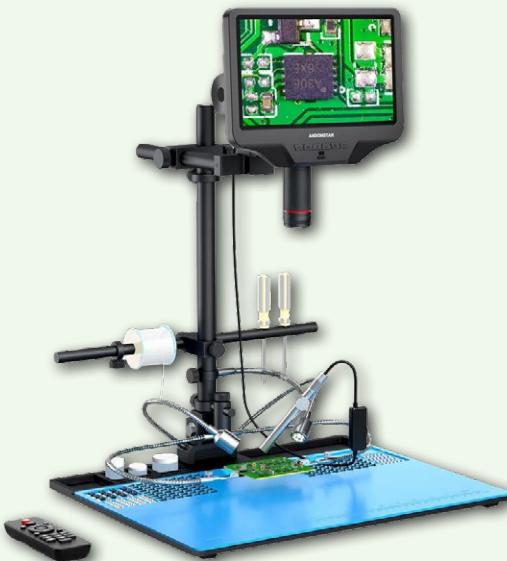
Le FNIRSI HRM-10 est un appareil de mesure de résistance interne portable de haute précision. Cet appareil offre une véritable mesure à quatre fils et est conçu pour la précision et la facilité d'utilisation. Il mesure automatiquement les valeurs de résistance interne et de tension simultanément, affichant clairement les résultats sur son écran couleur HD. Les utilisateurs ont la possibilité d'ajuster manuellement les plages de tension et de résistance en fonction de leurs besoins. L'appareil comprend également un mode de tri qui filtre automatiquement les bonnes et mauvaises batteries en fonction de seuils définis par l'utilisateur. De plus, il prend en charge le stockage des données historiques et permet d'exporter des enregistrements de mesures sous forme de tableau.

Prix : ~~59,95 €~~

Prix spécial : 49,95 €

www.elektor.fr/20947

Andonstar AD409 Max-ES 10,1" Microscope numérique (avec endoscope)



L'Andonstar AD409 Max-ES est doté d'une lentille métallique de haute qualité et d'un design de filtre UV unique. Fabriqué à partir de matériaux de qualité industrielle de premier ordre, il offre une précision et une durabilité inégalées, garantissant une expérience produit fiable. Le filtre UV placé devant la lentille métallique bloque la chaleur, la fumée et la poussière de soudure, protégeant ainsi la lentille et la rendant parfaite pour les professionnels du soudage et de la maintenance.

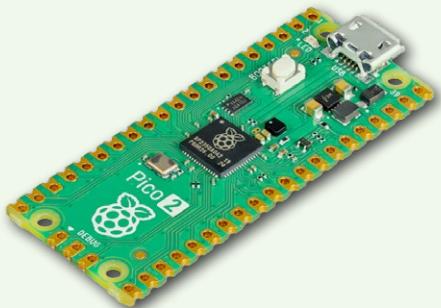
Prix : ~~499,00 €~~

Prix spécial : 349,00 €

www.elektor.fr/20852



Raspberry Pi Pico 2



Prix : 5,95 €

www.elektor.fr/20950

Raspberry Pi 5 (2 Go de RAM)



Prix : 54,95 €

www.elektor.fr/20951

KiCad Like A Pro (offre groupée)



Prix : 94,95 €

Prix spécial : 74,95 €

www.elektor.fr/20942

The Book of 555 Timer Projects



Prix : 29,95 €

Prix (membres) : 26,96 €

www.elektor.fr/20948

extension de couverture Wi-Fi sur ESP32

comment modifier simplement une antenne

Peter Neufeld (Allemagne)

Les modules ESP32 ont souvent des « antennes intégrées » (sur le circuit imprimé), ce qui entraîne parfois une faible couverture Wi-Fi, entre autres problèmes. Une solution est possible : utiliser une antenne externe ! Malheureusement, tous les modules ne disposent pas d'une prise pour une telle connexion. Ce n'est pas une raison pour abandonner, car cet article décrit une solution simple : modifier la carte !

Lors de mes expérimentations avec les modules ESP32, en particulier les ESP32-CAM, je souhaitais une couverture Wi-Fi plus performante. L'ajout d'une antenne externe aurait pu être utile, mais peu de modules sont équipés pour cela. Le coût et la complexité mécanique ne se retrouvent pas nécessairement dans les avantages obtenus.

L'antenne d'origine imprimée sur une carte ESP32 standard est en forme de F inversé avec une longueur d'onde $\lambda/4$ (voir **figure 1**). Ce type d'antenne permet de concilier une taille de dispositif, un diagramme d'antenne presque omnidirectionnel, un rendement énergétique acceptable et une couverture

Wi-Fi dans les limites de la puissance radio et de la sensibilité de réception de l'ESP32. Les liens [1] à [5] apportent des informations utiles sur ces antennes.

La modification

Mon idée était d'augmenter la couverture Wi-Fi du module en couplant l'antenne quart d'onde embarquée avec une nouvelle antenne demi-onde externe, plus grande et sans raccourcissements dus aux sinuosités, assurant ainsi une disposition optimale. En collant le fil d'antenne supplémentaire sur le circuit imprimé, aucune modification irréversible n'a été effectuée. Le résultat est illustré par la **figure 2**.

J'ai envisagé une antenne imprimée quart d'onde unipolaire à 2,4 GHz. J'ai ensuite mesuré sa longueur et je l'ai doublée, en première approximation. J'ai donc obtenu une antenne demi-onde composée d'un fil de commutation isolé ($\varnothing = 0,5$ mm) d'une longueur de 62 mm. La moitié du fil d'antenne supplémentaire suit la forme sinuuse de l'antenne F imprimée, en maintenant une distance grâce à l'isolation du fil. L'autre moitié



Figure 2. Fil d'antenne supplémentaire collé sur le circuit imprimé.

est montée sous une forme « libre ». L'une et l'autre parties assurent un fort couplage induit et capacitatif des deux antennes, formant un élément rayonnant plus grand et optimisé. Le meilleur point de colle concernant les hautes fréquences est le centre à faible impédance de la nouvelle antenne demi-onde, juste dans l'angle de droite du module ESP32. Ainsi, il est possible d'éviter que la partie à haute impédance ne soit affectée par un matériau de collage atténuant les fréquences radio.

Mesure

Pour mesurer l'effet, j'ai monté une carte ESP32 modifiée à côté d'une carte de référence inchangée, en veillant à ce que les deux modules aient approximativement les mêmes conditions de réception et de transmission par rapport au point d'accès, et ce, en minimisant leur influence mutuelle. Il a été très facile de placer cette configuration à un endroit quelconque, ce qui a permis d'explorer simultanément les conditions Wi-Fi pour les deux modules.

Mon idée a été de faire en sorte que le second module ESP32 transmette ses données au premier, et que le premier produise une page web avec une visualisation graphique des deux lignes temporelles de données, à consulter dans un navigateur web.

J'ai donc écrit du code applicable aux deux modules ESP32 (voir le **listage 1**, disponible



Figure 1. L'antenne quart d'onde en forme de F « imprimée » sur une carte ESP32.



Figure 3. Dispositif de mesure : ESP32 modifié et non modifié, et tablette affichant les valeurs de l'indicateur de puissance du signal reçu (RSSI) obtenues.

en [6]). En effaçant les signes de commentaires de quelques lignes, le code passe d'une version pour le premier module à une autre pour le second. Le code affiche l'intensité du champ Wi-Fi du point d'accès, mesurée par les deux dispositifs, sur une page web, également visible dans une fenêtre de navigateur. Les deux ESP32 doivent évidemment se connecter au même point d'accès (routeur). À noter que l'interpréteur BASIC Annex32 que j'ai utilisé possède un mécanisme intégré pour connecter une carte ESP32 à un réseau de routeurs. Un point d'accès est d'abord établi par l'ESP32 lui-même, puis une page web de configuration est proposée. L'utilisateur peut y indiquer le SSID et le mot de passe du routeur - vous ne trouverez donc pas ces données sensibles dans le listage.

La **figure 3** représente le dispositif de mesure complet. Le module 1 (avec l'antenne modifiée) enregistre sa propre valeur RSSI Wi-Fi et demande au second module, non modifié, une chaîne contenant la valeur RSSI enregistrée. Sur une tablette, une fenêtre de navigateur permet de faire défiler l'affichage de deux courbes, l'une et l'autre comportant les 100 dernières mesures - soit environ 50 secondes. La **figure 4** représente une série de mesures caractéristique. Dans le cas présent, le gain de l'antenne modifiée est d'environ +6 dB par rapport à la référence. Pas si mal, non ?

Au passage, Annex32 [7] propose un interpréteur BASIC et un environnement de développement intégré (IDE) autonome, fonctionnant intégralement sur l'ESP32. Vous pouvez ainsi facilement créer des pages web à présenter dans le navigateur d'un appareil externe, également compatible avec les fonctions

JavaScript. C'est ce que nous faisons ici, parce que des fonctions d'affichage graphique spéciales sont nécessaires. Pour Annex32, il existe également un forum actif [8] et un fabuleux manuel en ligne [9]. J'ai testé l'effet de l'antenne modifiée et enregistré la puissance du signal Wi-Fi du point d'accès dans différentes conditions. Le signal est globalement amélioré d'au moins +3 dB sur les modules modifiés, avec des fils d'antenne allongés et non coudés. Dans des conditions environnementales très réfléchissantes, l'amélioration atteint presque +10 dB. Toutefois, des mesures approximatives indiquent aussi clairement que les conditions désormais améliorées se retrouvent dans les zéros radio du diagramme d'antenne, qui sont, comme prévu, plus prononcés. Avec cette antenne, il faut garder à l'esprit que l'effet directionnel est plus important - pour le meilleur ou pour le pire. Une partie de l'amélioration du gain est certainement due à la modification du diagramme d'antenne et au couplage fort avec une antenne partiellement autonome et plus efficace.

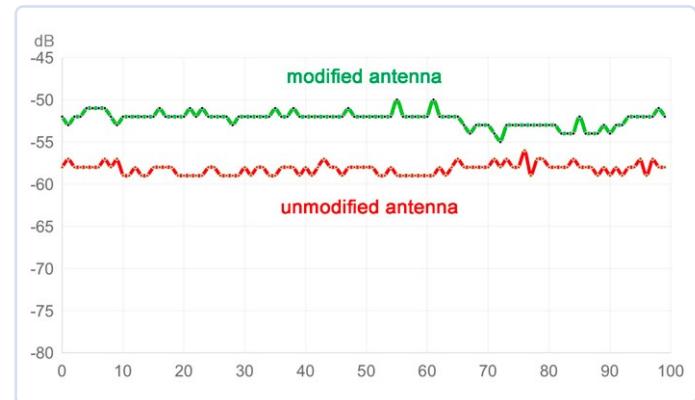


Figure 4. Résultats des mesures : le gain moyen de la modification est d'environ +6 dB.

Recherches et conclusions

J'ai effectué des expérimentations avec l'extrémité libre de l'antenne pointée vers le haut, à 90 degrés par rapport à la carte (voir **figure 5**). Cela a modifié le diagramme d'antenne résultant, qui devrait être étudié de manière plus approfondie pour des résultats optimaux. Cette variation peut être utile dans certains environnements, par exemple lorsque le module se trouve sur un circuit imprimé très dense et que l'antenne est plus libre. Dans des conditions Wi-Fi défavorables, cette simple modification de l'antenne ne fera peut-être pas de miracles, mais elle peut légèrement augmenter la couverture du module ESP32 ou améliorer les connexions instables. Cette amélioration s'accompagne inévitablement de l'inconvénient d'un effet directionnel plus prononcé. Cela fonctionne mieux si le module a un emplacement fixe et s'il est possible d'ajuster l'orientation de l'antenne par rapport au point d'accès. Cela signifie que le côté large de

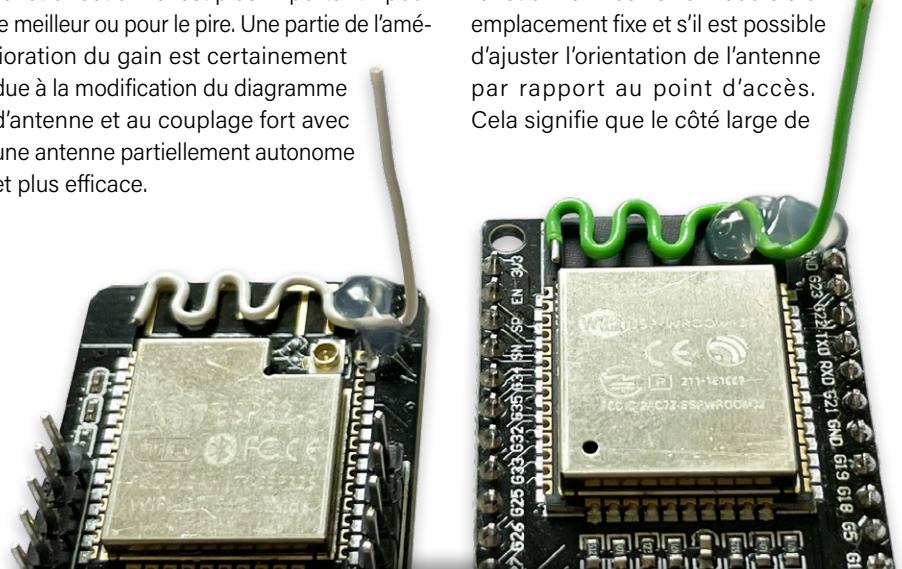


Figure 5. La partie rectiligne de l'antenne modifiée, orientée à 90° vers le haut.



Figure 6. ESP32-CAM avec antenne modifiée.

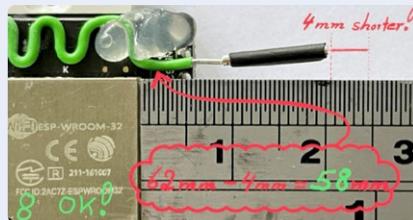
l'antenne est orienté vers le point d'accès et que la ou les antennes du point d'accès sont polarisées de la même manière, c'est-à-dire que les deux antennes sont alignées aussi parallèlement que possible l'une par rapport à l'autre.

L'encadré **Optimisation de la longueur de l'antenne** propose une méthode pour ajuster la longueur de l'antenne afin d'obtenir des résultats optimaux en termes de réception radio. Une antenne optimisée apporte une amélioration supplémentaire de +1,5 dB par rapport à l'antenne imprimée, ce qui permet d'obtenir des signaux d'au moins +6,5 dB dans des conditions normales.

La **figure 6** représente une application pratique. La modification décrite a permis d'obtenir un bien meilleur signal et une connexion vidéo très stable avec un module ESP32-CAM, à un emplacement où la couverture Wi-Fi était jusqu'ici critique. ↵

VF : Pascal Godart — 210604-04

Optimiser la longueur de l'antenne



Doubler le quart de longueur d'onde de l'antenne en F inversé fait abstraction des différentes influences physiques sur le fil partiellement libre, et j'ai donc dû déterminer la longueur optimale. Après avoir retiré l'isolant sur une longueur de 15 mm et raccourci le fil de 6 mm, j'ai ajouté une extension variable faite d'un fil argenté de 10 mm ($\varnothing = 1$ mm). Un morceau de gaine thermorétractable a servi de porte-objet provisoire.

Le système résultant (voir figure) a permis de régler facilement la longueur du fil - tout en observant mon dispositif de test - pour obtenir l'intensité de champ reçue la plus élevée possible. Cette méthode tient compte de l'influence diélectrique, inductive et capacitive sur la longueur du fil. Enfin, j'ai fixé la connexion avec un peu de soudure.

La longueur optimale déterminée expérimentalement pour l'ensemble du fil d'antenne demi-onde était de 58 mm, soit 4 mm de moins que l'ancienne valeur estimée, ce qui a entraîné une amélioration supplémentaire du signal de +1,5 dB !



À propos de l'auteur

Depuis plus de 40 ans, Peter Neufeld s'est occupé de l'informatique, des installations électriques et des systèmes de contrôle de bâtiments de différentes entités, mais aussi de technologies de communication et des médias. Aujourd'hui, il retrouve le temps de se consacrer pour son plaisir aux applications pratiques de l'électronique classique et des microcontrôleurs modernes. Il a décrit certains de ces projets sur la plateforme Elektor Labs et sur son blog.

Questions ou commentaires ?

Envoyez un courriel à l'auteur (peter.neufeld@gmx.de), ou contactez Elektor (redaction@elektor.fr).



Produit

➤ **Carte de développement NodeMCU ESP32 de JOY-iT**
www.elektor.fr/19973



Listage 1. Code BASIC ANNEX32

```
' ##### WIFI-GRAPH-LOGGER #####
' Allows to compare the two ESP-Modules which have different antennas
' (peter.neufeld@gmx.de 06/2024)
' Displays a graphical logging of two WIFI-signals in dB on module_1
' - the strength of own WIFI-connection to the WIFI access point
' - the strength of WIFI-connection of a remote ESP32 module_2 to the same AP

' RX = 1 => this ESP-Module regularly requests the WIFI-LOG-RESULT-STRING
'           from a remote module with http://REMOTE_IP/msg?x=1
' TX = 1 => this ESP-Module returns the string with its
'           WIFI-LOG-RESULT-STRING on request at http://MY_IP/msg?x=1
```

(Continues on the next page)

```

' uncomment this in code for the ESP32 module_1 as the main module
RX      = 1  'This module sends requests
TX      = 0  'This module is quiet

' uncomment this in the code for the ESP32 module_2 as the 2nd module
'RX      = 0  'This module does not send requests
'TX      = 1  'This module responds to requests

REMOTE_IP$ = "192.168.0.141" 'Set here YOUR IP-Address of module2

X_Num      = 100 'Number of WIFI-MEASURES to display in the graph
WIFI_REMOTE$ = ""
onhtmlreload WEBPAGE
gosub      WEBPAGE
if TX = 1 onurlmessage RETURN_WIFI_STRING
IF RX = 1 onwgetasync RECEIVE_REMOTE_STRING
timer0 500, LOG_MY_WIFI_CONNECTION
if RX = 1 timer1 1000, GET_REMOTE_WIFI_LOG_STRING
wait
#####
LOG_MY_WIFI_CONNECTION:
w=0
for i = 1 to 50
w = wifi.rssi + w
next i
w=W/(i-1)  ' Thanks for the hint Fernando !
WIFI_LOCAL$= trim$(WIFI_LOCAL$ + " " +str$(wifi.rssi,"%2.1f"))
c = word.count(WIFI_LOCAL$," ")
p = instr(1, WIFI_LOCAL$, " ")
P = len(WIFI_LOCAL$) - p
If c > X_NUM then WIFI_LOCAL$ = right$(WIFI_LOCAL$, p )
'wlog WIFI_LOCAL$ , c
jscall |traceme(0,"| + WIFI_LOCAL$ + |");
if WIFI_REMOTE$ <> "" jscall |traceme(1,"| + WIFI_REMOTE$ + |");
return
#####
RETURN_WIFI_STRING:
URLMSGRETURN WIFI_LOCAL$
return
#####
GET_REMOTE_WIFI_LOG_STRING:
wgetasync ("http://" + REMOTE_IP$+ "/msg?x=1")
return
#####
RECEIVE_REMOTE_STRING:
WIFI_REMOTE$ = WGETRESULT$
return
#####
WEBPAGE:
cls
jsexternal "/xy.min.js"
cnt = 0
a$ = ""
a$ = a$ + |<p>WIFI Graph for two ESP32-modules .. <br> |

```

(Continues on the next page)

```

a$ = a$ + |GREEN = WITH additional antenna <br> RED === with original antenna |
a$ = a$ + |</p><canvas id="canvas1" width="800" height="400"></canvas>|
html a$
pause 500
A$ = ""
A$ = A$ + |var datasets = []
A$ = A$ + |  {}
A$ = A$ + |    lineColor : 'rgba(20,100,100,1)',|
A$ = A$ + |    pointColor : 'rgba(20,20,20,1)',|
A$ = A$ + |    pointStrokeColor : '#fff',|
A$ = A$ + |    data : []|
A$ = A$ + |  },|
A$ = A$ + |  {}
A$ = A$ + |    lineColor : 'rgba(151,30,0,1)',|
A$ = A$ + |    pointColor : 'rgba(151,80,0,1)',|
A$ = A$ + |    pointStrokeColor : '#fff',|
A$ = A$ + |    data : []|
A$ = A$ + |  }|
A$ = A$ + |];
A$ = A$ + |var ctx2 = document.getElementById('canvas1').getContext('2d');|
A$ = A$ + ||
A$ = A$ + |var xy = new Xy(ctx2, );|
A$ = A$ + ||
A$ = A$ + |function traceme(set, data){|
A$ = A$ + |  var s = data.split(" ");|
A$ = A$ + |  for (var i=0; i<s.length; i++) {|
A$ = A$ + |    datasets[set].data[i] = [i, s[i]];|
A$ = A$ + |  }|
A$ = A$ + |  xy.draw(datasets);|
A$ = A$ + |}|
jscript a$
A$ = "" ' clean memory
return

```

LIENS

- [1] Espressif : Principes de disposition des circuits imprimés (recommandations pour la conception des antennes) :
<https://tinyurl.com/esppcblayout>
- [2] Espressif : Spécifications ESP32 (réception Wi-Fi) : <https://tinyurl.com/esp32wifirec>
- [3] Texas Instruments : Antenne miniature 2,4 GHz pour circuit imprimé - Audun Andersen : <https://ti.com/lit/an/swra117d/swra117d.pdf>
- [4] NXP : Antennes planaires compactes 2,4 GHz : <https://nxp.com/docs/en/application-note/AN2731.pdf>
- [5] Antenne en F inversé : <https://antenna-theory.com/antennas/aperture/ifa.php>
- [6] Téléchargement du code : <https://elektormagazine.fr/210604-04>
- [7] Interpréteur BASIC pour ESP32 - ciccioCB : <https://flasher.cicciocb.com>
- [8] Forum Annex RDS : <https://cicciocb.com/forum/viewforum.php?f=5>
- [9] Documentation Annex32 : <https://cicciocb.com/annex32help>

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

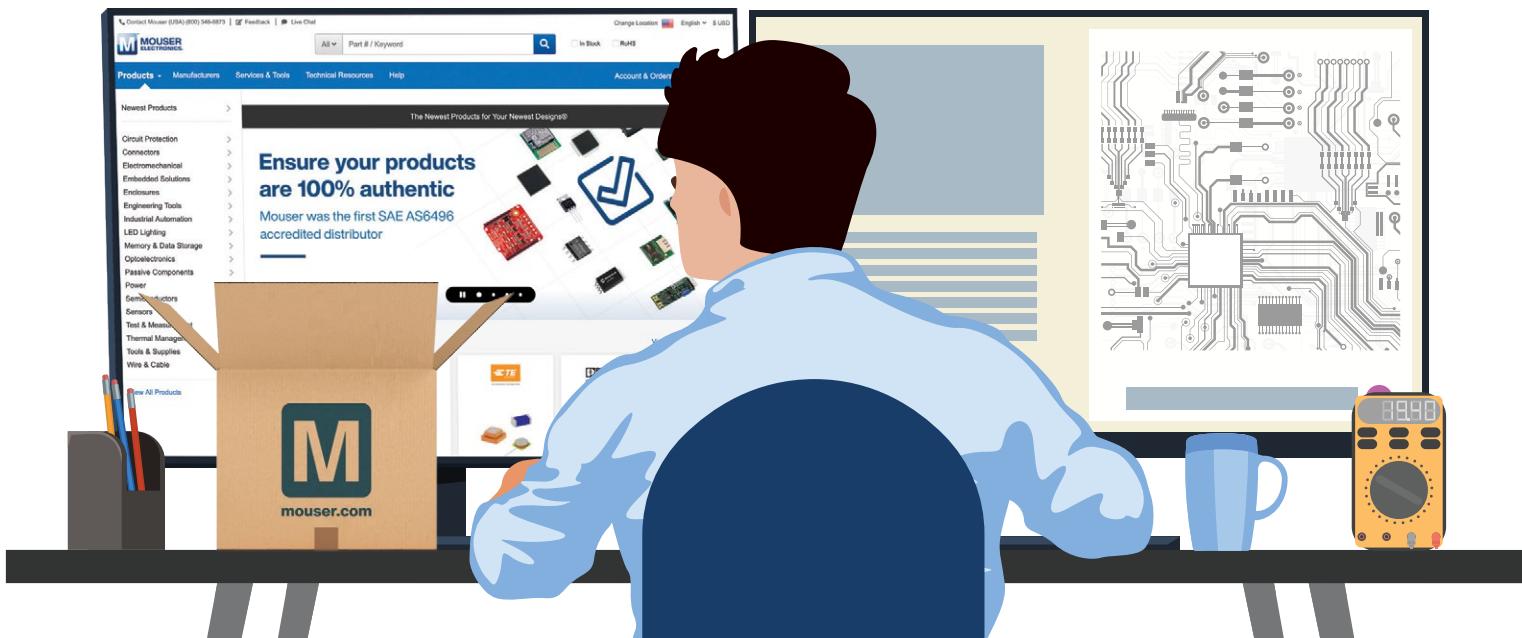


www.elektormagazine.fr/membres

elektor
design > share > earn

Vous concevez. Nous délivrons.

Les dernières nouveautés pour vos conceptions les plus récentes™



mouser.fr/new

M MOUSER
ELECTRONICS