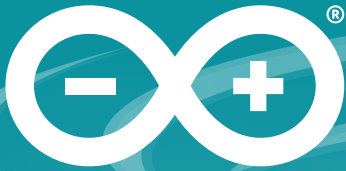


Édition spéciale
Rédaction invitée



ARDUINO

Édition bonus
dévoilée !

Domotique

Connectivité simplifiée

p. 4 **Rétro-gaming**
avec Arduino

p. 14 **Contrôleur**
pour Spotify

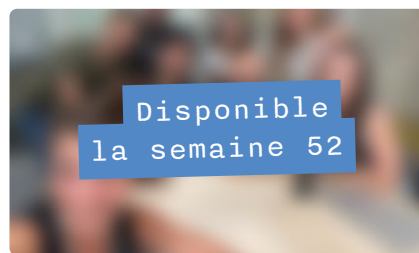
Du prototypage à la production

De nouveaux
articles bonus
gratuits chaque
semaine !



La carte d'apprentissage
Arduino d'Elektor

p. 10



Disponible
la semaine 52

App... in
au servi... on

p. 24



Disponible
la semaine 01
2023

S... es qui
réa... is

p. 32

De la part d'Arduino et d'Elektor par Amour de l'Ingénierie

En vente maintenant

Édition spéciale du magazine Elektor !
Rédaction invitée : Arduino !

Des projets électroniques

« maison », des idées pour les électroniciens, et plus encore, proposés par les experts d'Arduino et d'Elektor

De nombreux projets et tutoriels

Explorez des sujets d'actualité : **MicroPython**, **TinyML** et domotique avec Arduino

Découvrez l'Arduino : Les points de vue de **Fabio**, **Massimo** et **David**

Découvrez la **Portenta X8**

Achetez-le dès maintenant, dans votre **kiosque préféré** ou dans l'e-choppe d'Elektor.

Les liens dans les articles vous offrent un **accès facile** aux produits et solutions Arduino.



En savoir plus

www.elektor.fr/arduino-magazine



45^{ème} année
Édition bonus dévoilée

ISSN 0181-7450
Dépôt légal : décembre 2022
CPPAP 1125 T 83713
Directeur de la publication :
Donatus Akkermans

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

Pour toutes vos questions :
service@elektor.fr

www.elektor.fr | www.elektormagazine.fr

Publicité :
Raoul Morreau
Tél. : +31 (0)6 4403 9907
Courriel : raoul.morreau@elektor.com

DROITS D'AUTEUR :
© 2022 Elektor International Media B.V.

Elektor ne publie que du contenu (texte et images) produit par lui-même ou avec l'autorisation de son créateur. Avant publication, les droits d'auteur du contenu fourni par des tiers sont vérifiés. Si l'ayant droit est inconnu, Elektor s'efforce de le retrouver pour le rémunérer. Il n'est pas toujours possible de retrouver le détenteur des droits d'auteur. Si vous êtes ou connaissez le titulaire inconnu des droits d'auteur d'une publication, veuillez nous contacter à l'adresse redaction@elektor.fr.

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

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

Imprimé aux Pays-Bas par
Senefelder Misset - Doetinchem
Distribué en France par M.L.P. et en Belgique par A.M.P.

Édition bonus dévoilée

L'édition du magazine *Elektor*, publiée début décembre 2022 avec Arduino en tant que rédaction invitée, n'était pas la fin de la collaboration créative entre Elektor et Arduino. Nous avons encore plus de projets, des idées et des articles informatifs pour vous inspirer pendant les mois à venir. Sur une période de quatre semaines, nous dévoilons le contenu de cette édition jusqu'à ce que vous disposiez du magazine bonus complet début janvier 2023. Quelle belle façon de démarrer la nouvelle année ! Que vous soyez un ingénieur professionnel travaillant sur un nouveau produit industriel ou un bricoleur à la recherche d'un projet amusant basé sur

Arduino pour le week-end, vous trouverez cette édition supplémentaire du magazine Elektor informative et motivante. Nous vous proposons des articles sur un large éventail de sujets et de projets liés à Arduino, notamment le rétrogaming avec Arduino, une carte d'apprentissage Arduino d'Elektor et un contrôleur portable basé sur Arduino pour Spotify. En lisant les projets et les articles de ce numéro, n'hésitez pas à partager vos idées avec nous sur elektormagazine.fr, arduino.cc et sur les réseaux sociaux. Nous avons hâte de recevoir vos commentaires. Bonne lecture !

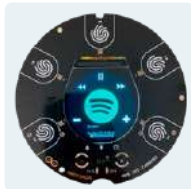
C. J. Abate (Directeur du contenu, Elektor)

DANS CE NUMÉRO



- ◀ 4 **Doom sur Portenta**
Rétro-gaming avec Arduino

- 10 **Introduction rapide au monde de l'Arduino**
Carte de développement pour l'Arduino Nano



- ◀ 14 **Contrôleur pour Spotify**
Le kit Oplà IoT contient (presque) tout ce dont vous avez besoin

- 20 **Créez, déployez et maintenez des applications évolutives et sécurisées**
avec Arduino Portenta X8 équipé du mini processeur d'applications i.MX 8M de NXP et de l'élément de sécurité EdgeLock SE050



- ◀ 24 **Appliquez les principes de la programmation de la semaine 52**
Renforcez vos compétences, la semaine 52 de la semaine 52 chez Arduino

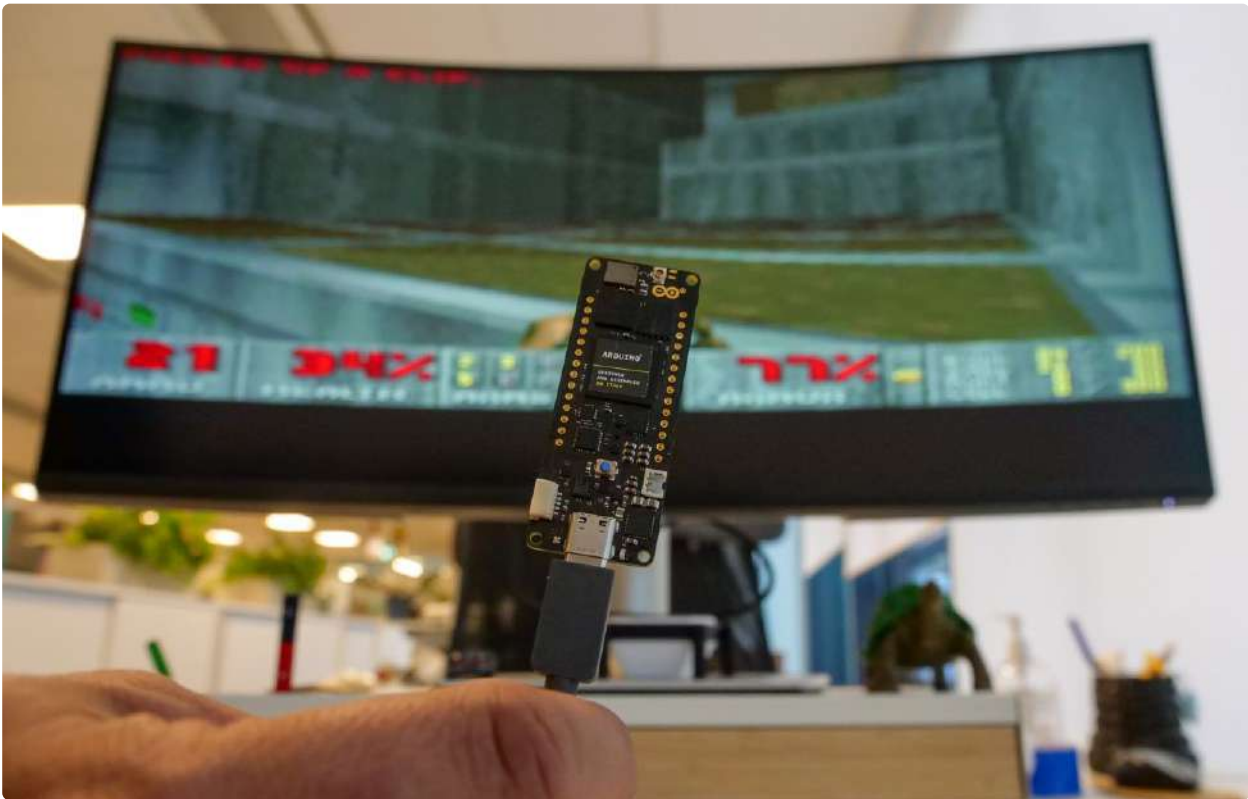


- 28 **Les principes de la semaine 52**
Parce que la semaine 52 est la semaine 52



- ◀ 32 **Art de la semaine 01**
Scénario de la semaine 01 2023

- ◀ 38 **Cartes de la semaine 01**
Mettez à jour la semaine 01 no ! 2023



Vous pouvez faire tourner Doom sur une carte Portenta !

Doom sur Portenta

rétro-gaming avec Arduino

David Cuartielles (Arduino)

Saviez-vous que *Doom* pouvait être exécuté sur une carte *Portenta H7* ? Comment et pourquoi ce jeu ? Martino Facchin, chef de l'équipe en charge des micrologiciels d'Arduino, satisfait notre curiosité.

Sorti en 1993, *Doom* est probablement le jeu le plus populaire de l'histoire du jeu vidéo. Avec plus de 3,5 millions de titres vendus 50 \$ l'unité, ses développeurs sont devenus millionnaires du jour au lendemain. *Doom* est une création d'*id Software*, à l'époque un petit éditeur de jeux vidéo connu pour avoir publié l'année précédente le titre à succès *Wolfenstein 3D*. *Doom* est un jeu de tir en vue subjective,

un genre de jeu dans lequel le joueur combat des ennemis au moyen d'armes qu'il peut aussi se procurer sur le champ de bataille. *Doom* a été porté sur tous les systèmes d'exploitation, mais peut aussi être exécuté sur du matériel dépourvu d'OS. Le code est ouvert, actuellement distribué sous licence GPL. Sa compilation n'est pas triviale, mais certains sont parvenus à l'exécuter sur de très petits ordinateurs et au sein d'autres programmes. Microsoft lui a rendu hommage sous la forme d'un « œuf de Pâques » caché dans la version Excel 95.

Certains utilisent le jeu pour mesurer les performances de petits ordinateurs ou faire étalage de leurs talents de codeur. Lors du DEF CON 22 de Las Vegas, le codeur @sickcodes (son pseudo sur Twitter et GitHub) a ainsi lancé sur l'afficheur d'un tracteur John Deere une version de *Doom* comportant des scènes agricoles. Arduino ne fait pas exception. Nous aussi avons utilisé *Doom* pour tester les capacités du premier prototype de la Portenta H7 – la plus puissante de nos cartes en 2018. J'ai invité Martino Facchin, responsable de l'équipe micrologiciel d'Arduino, à nous en dire plus sur cet épisode.



David Cuartielles : Parlons du rapport entre Doom et la carte Portenta H7. J'ai déjà esquissé à grands traits l'histoire du jeu et parlé de son succès phénoménal, un succès qui a rendu ses concepteurs millionnaires.

Martino Facchin : Des concepteurs qui par la suite ont publié le code source – c'est le point le plus important.

Cuartielles : Absolument ! À ce propos, sous quelle licence avaient-ils publié le code ?

Facchin : C'est à vérifier, mais je crois qu'il s'agit d'une licence compatible avec la GPL, la *Doom Source Code License*. Seul le moteur de jeu est open source, les éléments graphiques ne le sont pas. De fait on ne peut jouer qu'à la version *shareware* de Doom, pas au jeu complet. À moins de l'acheter bien sûr.

Note de David

J'ai vérifié, et effectivement *id Software* a publié Doom en 1997 sous la licence mentionnée par Martino – une ouverture du code à des fins éducatives. C'est après un incident de disque dur survenu chez les mainteneurs de *glDoom* – qui laissa le monde sans copie de ce portage openGL en raison de la clause de non-distribution de la licence Doom – qu'*id Software* accepta d'adopter pour sa licence les termes de la licence GPL.

Cuartielles : Le truc cool avec ce jeu, c'est que les joueurs pouvaient créer leurs propres « mods ». Je me rappelle d'une version de Wolfenstein 3D avec des personnages de Star Wars.

Facchin : Ça ne me dit rien.

Cuartielles : Eh oui, je suis vieux à ce point. Je me souviens pour nous deux.

Facchin : Ma foi, je me souviens bien avoir joué à Wolfenstein 3D quand j'étais enfant, mais comme nous n'avions pas de connexion internet, nous ne pouvions sans doute pas récupérer ce genre de mods.

Cuartielles : Avant d'en venir à Doom, pouvez-vous vous présenter ?

Facchin : [Rires] Martino Facchin, ingénieur micrologiciel chez Arduino.

Cuartielles : Quelle est votre fonction ?

Facchin : Je suis responsable des micrologiciels – le type qu'on appelle quand on a besoin d'aide sur un micrologiciel. J'étais seul au départ, mais aujourd'hui je suis entouré d'une équipe formidable, qui d'ailleurs s'agrandit encore. Nous essayons aussi d'y inclure la communauté en expliquant ce que nous faisons.

Cuartielles : À part le portage de Doom sur une Portenta (dont nous reparlerons), de quelle réalisation êtes-vous le plus fier ?

Facchin : Je dirais que c'est le *framework PluggableUSB*. Six mois après mon arrivée chez Arduino, nous avons dû répondre aux demandes des utilisateurs souhaitant exploiter le port USB de la carte *UNO Due* et celui de la toute nouvelle *Arduino Zero*. Chaque fois qu'un utilisateur reliait une carte *Leonardo* à son ordinateur, les pilotes de clavier, souris, etc., étaient activés, même s'ils n'étaient pas utilisés. Nous avons dû répondre à la hâte en écrivant un descripteur USB aidant l'utilisateur à voir uniquement ce qu'il voulait utiliser à ce moment-là. Nous avons également permis d'autres fonctions, par exemple l'USB MIDI réclamé par beaucoup d'utilisateurs. Pour moi c'était quelque chose d'énorme. J'étais assez jeune à l'époque, et pour trouver la meilleure stratégie j'avais dû m'adresser à la communauté avec l'aide de Matthijs Kooijman (www.stderr.nl pour en savoir plus sur son travail) et de Paul Stoffregen (créateur de Teensy). Et ça a marché. De temps en temps des personnes nous disent : « J'ai utilisé la bibliothèque MIDI pour ceci ou cela ». Je connais même un développeur qui exploite les fondations de ce code pour paramétrer les dispositions des claviers internationaux. Je suis très fier de cela.

Cuartielles : Vous dites que vous étiez « assez jeune à l'époque ». Depuis combien de temps travaillez-vous chez Arduino ?

Facchin : Depuis six ans et demi. Dans les bureaux de Turin.

Cuartielles : Une belle ancienneté déjà. Combien de personnes compte l'équipe Firmware ?

Facchin : Six personnes. Ça peut sembler beaucoup, mais nous nous occupons de tous les produits et avons d'autres activités, comme les certifications. À Turin travaille aussi l'équipe *Tooling*. Elle s'occupe de la CLI, l'interface en ligne de commande, et d'autres fonctions logicielles.



Doom s'est imposé naturellement, restait à implanter les fonctions nécessaires à son exécution.



Cuartielles : Tous les développeurs utilisent GitHub, n'est-ce pas ?

Facchin : Oui, et le code devient public une fois le développement achevé.

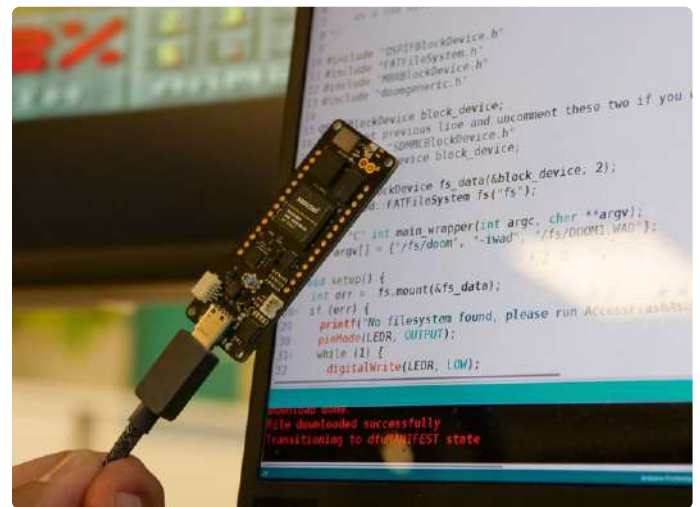
Cuartielles : Ce qui est super. Revenons à Doom. J'ai rappelé qu'il avait été vendu à plus de 3,5 millions d'unités et que ce succès, le premier dans son genre pour un jeu de tir à la première personne, avait rendu ses créateurs pleins aux as. Nous savons aussi que le code a été rendu public et porté sur toutes sortes d'appareils.

Facchin : Téléphones, calculatrices, systèmes d'exploitation...

Cuartielles : Et sur la Portenta H7, donc. Il s'agit d'une carte Arduino à double cœur conçue pour les environnements industriels. Qui a eu l'idée d'y faire tourner Doom ?

Facchin : C'est juste que nous venions d'en recevoir le premier prototype, une belle carte enrichie de nombreuses puces, et que tout restait à faire. Il y avait en particulier cette puce d'Analogix généralement utilisée pour convertir les signaux MIPI en signaux Display-Port et les diriger ensuite vers un moniteur externe. Nous n'avions aucune expérience avec les sous-systèmes de la puce chargés de cette conversion, et les exemples fournis ne nous aidaient pas. Nous sommes d'abord parvenus à afficher quelques boîtes jaunes, puis le logo Arduino entouré de formes graphiques, mais c'était loin d'être parfait. Nous ne comprenions pas pourquoi le rendu était mauvais, alors j'ai décidé d'essayer avec des graphismes et animations dont la qualité m'était connue.

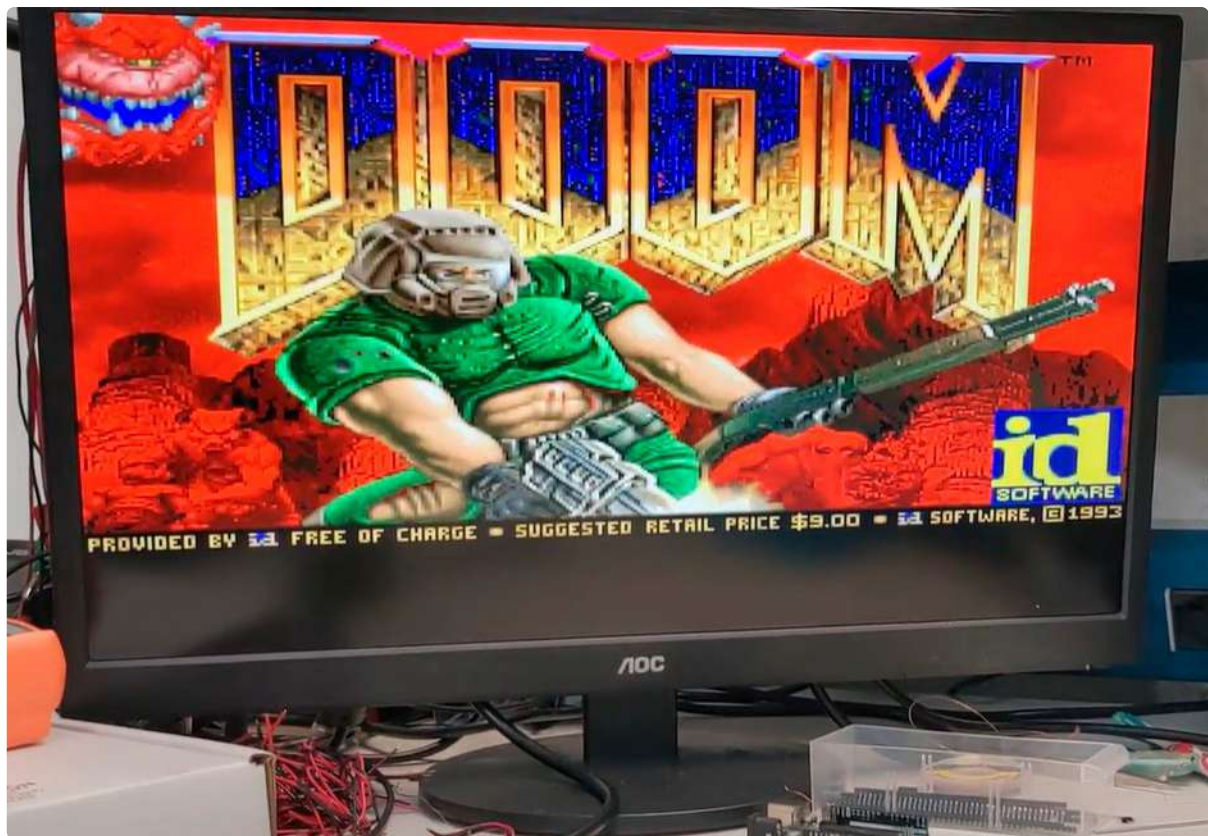
Doom s'est imposé naturellement, restait à implanter les fonctions nécessaires à son exécution. Le faire tourner sur un système sans OS demande relativement peu de travail, même si ce n'est pas vraiment la version de Doom avec laquelle vous passeriez des heures à jouer. Il suffit de modifier six ou sept fonctions pour l'adapter à votre matériel. Avant de lancer le simulateur, j'ai d'abord dû m'occuper de l'exécution des données en mémoire RAM ou externe, car nous n'avions aucun code de départ pour ça. Ensuite j'ai dû préparer un tampon de trame et procéder à un peu de magie afin d'obtenir, enfin, une sortie vidéo correcte par USB-C.



Pour jouer à Doom avec un périphérique de type manette, il vous faudra ajouter vous-même le code nécessaire.

Comment lancer Doom sur votre Portenta H7

1. Téléchargez la dernière version de l'EDI Arduino. Nous recommandons la version 2.0 ou ultérieure.
2. Téléchargez le cœur Portenta depuis le gestionnaire de cartes.
3. Sélectionnez le cœur M7 pour votre carte Portenta H7. Toutes les étapes suivantes doivent être exécutées sur ce cœur.
4. Assurez-vous que l'EDI a identifié le port auquel est relié la carte.
5. Vous trouverez sous Exemples / Doom un exemple contenant les instructions de base. Quelques exemples doivent d'abord être exécutés avant de procéder à l'installation de Doom.
6. [Optionnel] Lancez Exemples / STM32H747_System / STM32H747_manageBootloader pour mettre à jour le chargeur de démarrage de votre carte.
7. Formatez la mémoire flash externe avec Exemples / STM32H747_System / QSPIFormat. Après installation, ouvrez le terminal série et suivez les instructions données.
8. Lancez Exemples / USB As Mass Storage / AccessFlashAsUSBdisk pour transformer votre carte en périphérique de stockage USB.
9. Ouvrez le moniteur série et choisissez la façon dont la carte doit être formatée. Votre OS devrait ensuite reconnaître deux nouveaux lecteurs reliés à votre ordinateur.
10. Téléchargez DOOM1.WAD depuis doomwiki.org/wiki/DOOM1.WAD et copiez le fichier dans la partition la plus large du lecteur virtuel Portenta.
11. Retournez à l'exemple Doom.ino de l'étape 5 et flashez-le sur la carte (sur son cœur M7 !) Si vous n'aviez pas réussi à identifier le port de programmation, double-cliquez sur le bouton Reset de la carte avant de lancer le chargement du code, et assurez-vous que le port série est bien reconnu.
12. Déconnectez votre Portenta H7 et reliez-la à un hub USB-C comme s'il s'agissait d'un portable. Reliez votre hub à une alimentation externe, et utilisez un câble HDMI pour envoyer le signal vidéo à un écran.



Doom fait partie des classiques du jeu vidéo !

Cuartielles : Le processeur de la carte Portenta H7 possède deux cœurs Cortex M4 et M7. Lequel exécutait Doom ?

Facchin : À l'époque c'était le M4 parce qu'il était plus simple à mes yeux de programmeur de systèmes embarqués, mais aujourd'hui nous utilisons le M7. C'est une sorte de microcontrôleur type, sans caractéristiques spéciales. D'un autre côté le M7 possède un cache qu'il faut inspecter et invalider au bon moment lors de l'affichage des images. Il s'est montré très rapide la première fois que je l'ai utilisé, mais rien ne s'affichait correctement. A contrario la cadence qu'offrait le M4 suffisait (25 images par seconde, ou FPS), et l'affichage était parfait.

Cuartielles : 25 FPS, mon premier ordinateur était loin d'en faire autant. Résumons-nous : vous étiez parvenu à exécuter Doom sur le plus lent des deux processeurs d'une carte à modules Bluetooth et Wi-Fi. La vidéo est délivrée par USB-C. Ce port peut recevoir une souris, un clavier, un hub, que sais-je encore. Quel périphérique d'entrée utilisiez-vous pour Doom ?

Facchin : Le projet n'a hélas pas été plus loin. Après avoir réussi à afficher Doom, nous sommes passés à LVGL, une bibliothèque graphique bien plus utile aux développeurs pour l'écriture d'applications destinées à la Portenta. LVGL sait traiter les événements venant de l'hôte USB, p. ex. un clic de souris, ce qui permet d'écrire des interfaces pour toutes sortes de contextes professionnels.

Cuartielles : Ce serait cool d'avoir un automate programmable industriel à Portenta H7 sur lequel on pourrait jouer à Doom sur le M4 pendant que le M7 bosse, non ?

Facchin : Absolument !

Cuartielles : Merci Martino, c'était sympa de vous entendre narrer ce portage de Doom sur la Portenta H7. Nous ne manquons pas de partager les instructions permettant de le faire. Il restera au lecteur à ajouter au code le périphérique de son choix pour profiter du jeu. ◀

220542-04 — VF : Hervé Moreau

À propos de l'auteur

David Cuartielles est le cofondateur d'Arduino. Il est titulaire d'un doctorat en design interactif et d'un master en génie des télécommunications. Il enseigne à l'université de Malmö.

Des questions, des commentaires ?

Contactez Elektor par courriel à redaction@elektor.fr.



Produits

- **Arduino Portenta H7**
www.elektormagazine.fr/arduino-portenta-h7

20%

de réduction
sur la première année
de votre abonnement

Rejoignez la communauté Elektor

Souscrivez un
abonnement
ou



Gold Green

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



www.elektormagazine.fr/arduino-member

Utilisez le code promo :

ARDUINO22

déballage

du kit LCR-mètre
avec David Cuartielles

Réservez la date : 26 janvier 2023



Voulez-vous déballer le kit LCR-mètre d'Elektor avec moi ? Regardez l'épisode du 26 janvier 2023 (18:00 CET) d'Elektor Lab Talk, où je rejoindrai les ingénieurs d'Elektor Mathias Claussen et Jens Nickel pour discuter du kit LCR-mètre, ainsi que pour répondre à vos questions sur la technologie Arduino et sur cette édition d'Elektor. Ne manquez pas le livestream. Venez poser vos questions !

220555-04

Elektor LabTalk

Regardez David live sur Elektor Lab Talk le 26 janvier 2023 !



www.elektormagazine.com/labtalk-david

Post
discipline.

introduction rapide au monde de l'Arduino

carte de développement pour l'Arduino Nano

Wolfgang Trampert (Allemagne)

Elektor reste fidèle à sa mission éducative : nous vous présentons ici une toute nouvelle carte d'apprentissage basée sur l'Arduino Nano. Associée à une formation pratique et bien structurée, elle constitue une plateforme idéale pour améliorer vos compétences et explorer le monde des microcontrôleurs.

Note de l'éditeur

Au moment de la publication de cet article, le livre associé à ce kit n'est disponible qu'en allemand. Des traductions sont prévues très prochainement. Une fois terminée, la version française sera disponible dans l'e-choppe Elektor.

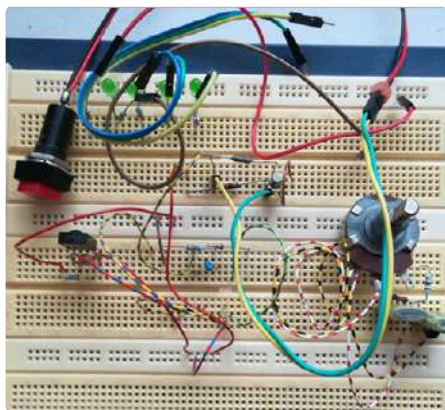


Figure 1. Montage simple pour tester un croquis Arduino.

On pourrait dire que la « philosophie Arduino » est associée à une approche de conception axée sur le matériel : dans la plupart des cas, le logiciel ou le croquis Arduino permet de commander des composants, notamment des interrupteurs, des boutons-poussoirs, des potentiomètres, des LED, des LCD, des buzzers piézoélectriques, des transistors de commande, etc. via les GPIO du microcontrôleur. D'autres types de périphériques ou de modules électroniques tels que des capteurs, des cartes d'affichage ou de pilotage utilisent diverses interfaces série telles que SPI, I²C ou le bus 1 fil pour communiquer avec le contrôleur. Afin de vous familiariser avec le monde des microcontrôleurs et des cartes Arduino, vous pourrez construire de nouveaux circuits pratiques et les contrôler avec une carte Arduino.

Cependant, la base de tout projet utilisant du matériel Arduino est le développement du logiciel (*sketch*). Le matériel n'est qu'un moyen d'arriver à ses fins. Vous pouvez bien sûr connecter tous les périphériques dont vous avez besoin pour un projet particulier en utilisant une carte de prototypage (**figure 1**). Cette approche vous donne un maximum de liberté pour installer les périphériques et acheminer les signaux selon vos besoins. Ce n'est pas toujours une bonne idée, surtout lorsque vous êtes débutant. La taille du montage peut rapidement augmenter et se compliquer en connectant un grand nombre de fils Dupont. Souvent, vous passerez plus de temps à déboguer le matériel et à résoudre les erreurs de câblage qu'à écrire le code. Cela ne fait qu'augmenter le niveau de frustration, et la correction d'erreurs ne vous apprendra pas nécessairement grand-chose d'utile.

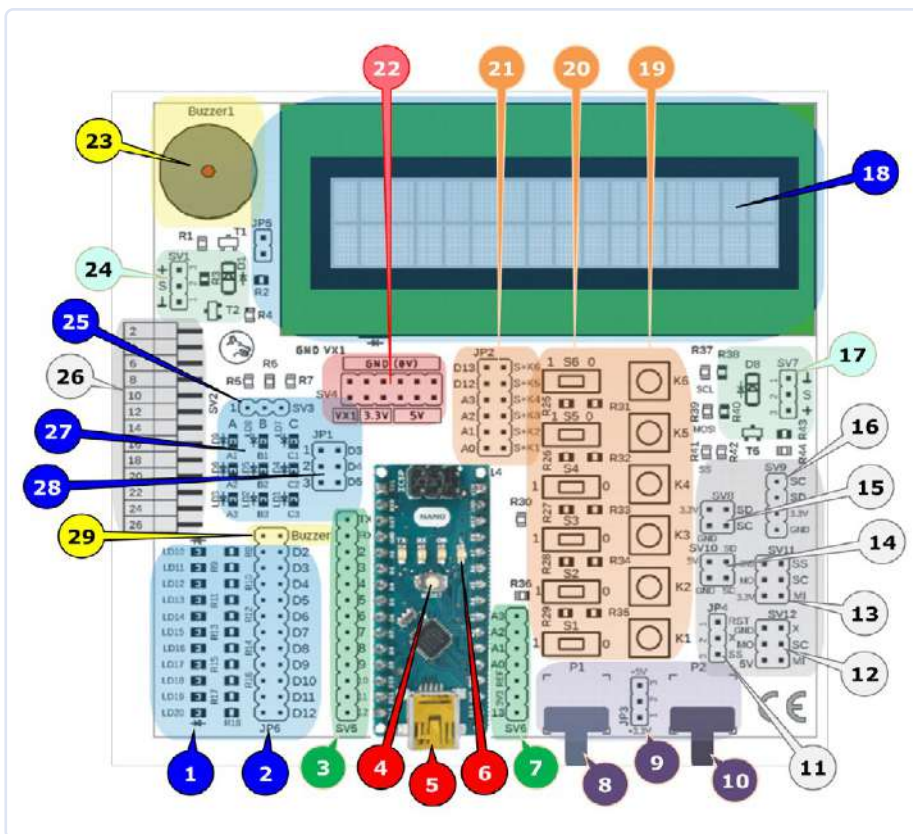


Figure 2. La carte d'apprentissage MCCAB, Rev. 3.3.

Carte d'apprentissage MCCAB

Pour contourner ces obstacles, nous avons développé la carte d'apprentissage Arduino d'Elektor, également connue sous le nom de *carte d'apprentissage MCCAB* (figure 2). Le cœur du système est une carte Arduino Nano que vous pouvez connecter sur la carte d'apprentissage MCCAB. Vous trouverez en plus de celle-ci un grand nombre de périphériques de base dont vous aurez généralement besoin pour construire de nouveaux prototypes pour de nombreuses applications telles qu'une configuration pour votre laboratoire, des circuits de test et d'expérimentation, des projets et des exercices pour faciliter vos études et votre apprentissage, ainsi que des projets de loisirs. Les E/S (GPIO) du microcontrôleur sont toutes accessibles grâce à deux connecteurs sur la carte d'apprentissage MCCAB, ce qui assure une flexibilité de déploiement maximale. Il est possible de connecter des périphériques supplémentaires ou de transmettre des signaux externes avec des fils Dupont, selon vos besoins. Vous n'aurez pas à vous soucier d'un câblage incorrect et vous passerez moins de temps à fouiller dans vos boîtes pour trouver le composant indispensable à la réalisation de votre circuit. Il est également possible de connecter facilement des circuits supplémentaires construits sur des platines à essai avec des câbles Dupont, puisque toutes les E/S du microcontrôleur de l'Arduino Nano sont connectées

aux deux connecteurs SV5 et SV6 de la carte d'apprentissage MCCAB (flèches 3 et 7 sur la figure 2). Sur le côté gauche de la carte, vous pouvez également voir la double rangée de 26 broches à angle droit du connecteur SV2 (flèche 26 sur la figure 2) où vous pouvez insérer une carte d'extension externe. Ce connecteur fournit tous les signaux des E/S importants du microcontrôleur. Les cartes externes destinées à réaliser des fonctions telles qu'un traceur de courbe de composants électroniques, une alimentation de laboratoire ou un contrôleur de feux de signalisation peuvent être connectées à la carte d'apprentissage MCCAB et commandées par celle-ci. Les informations et les résultats d'un croquis en cours d'exécution peuvent être affichés sur l'écran LCD intégré de 2 x 16 caractères (flèche 18), reliés par le bus I2C de la carte. Une matrice de LED 3 x 3 est également intégrée (flèche 27).

La carte d'apprentissage MCCAB est alimentée par une alimentation $V_{cc} = +5\text{ V}$. Cette alimentation est généralement fournie par le câble USB branché sur votre PC, dont vous avez besoin pour créer et téléverser le croquis de chaque exercice sur la MCCAB. Vous pouvez également alimenter la MCCAB avec un bloc d'alimentation externe. Dans le schéma de la carte d'apprentissage (figure 2), tous les composants associés à une fonction spécifique de la carte sont identifiés avec une couleur de fond commune.

Les commandes et indicateurs de la carte d'apprentissage MCCAB

- | | |
|----|---|
| 1 | 11 x LED (indication de l'état des entrées/sorties D2 à D12) |
| 2 | Connecteur reliant les LED LD10 à LD20 avec les E/S D2 à D12 |
| 3 | Entrées et sorties du microcontrôleur |
| 4 | Bouton <i>RESET</i> |
| 5 | Arduino NANO avec prise mini USB |
| 6 | LED <i>L</i> , associé à la broche D13 |
| 7 | E/S du microcontrôleur |
| 8 | Potentiomètre P1 |
| 9 | Tension d'alimentation de P1 et P2 |
| 10 | Potentiomètre P2 |
| 11 | Signal sur la broche X du SV12 |
| 12 | Interface SPI 5 V (le signal de la broche X est sélectionné par JP4.) |
| 13 | Interface SPI 3,3 V |
| 14 | Interface I2C 5 V |
| 15 | Interface I2C 3,3 V |
| 16 | Interface I2C 3,3 V |
| 17 | Sortie de commutation pour composants externes |
| 18 | LCD 2 x 16 |
| 19 | 6 x boutons-poussoirs K1 à K6 |
| 20 | 6 x commutateurs à glissière S1 à S6 |
| 21 | Connecteur pour relier les commutateurs aux E/S du microcontrôleur. |
| 22 | Diviseur de tension d'alimentation |
| 23 | <i>Buzzer1</i> |
| 24 | Sortie de commutation pour composants externes |
| 25 | Matrice de colonnes de LED 3 x 3 |
| 26 | 2 x 13 connecteurs pour connecter un module externe |
| 27 | 3 x 3 matrice de LED (rouge) |
| 28 | Connexions des lignes de la matrice 3 x 3 à D3, D4 et D5 |
| 29 | La position du cavalier relie <i>Buzzer1</i> à la broche D9 |

Tableau 1. Classes disponibles dans la bibliothèque MCCAB_Lib.

Classe	Utilisation
KeySwitch	État des commutateurs S1 à S6 et des boutons-poussoirs K1 à K6
Matrix	Commande de la matrice de LED 3 x 3.
LED	Commande d'allumage, d'extinction et de clignotement des 12 LED LD10 à LD20
LedBlock	Affichage d'un rythme binaire sur les 11 LED (LD10 à LD20)

La bibliothèque MCCAB_Lib à utiliser avec les cartes d'apprentissage

Le développement logiciel implique l'utilisation de l'EDI Arduino pour écrire le programme (ou *sketch*) qui commande le microcontrôleur. Le croquis est ensuite compilé et téléversé vers le microcontrôleur de l'Arduino Nano sur la carte d'apprentissage via un câble mini USB. Il est possible de configurer les E/S du microcontrôleur avec la fonction Arduino `pinMode()` et de lire ou de contrôler les données des signaux émis ou reçus par les composants connectés à la carte d'apprentissage avec `digitalRead()`, `digitalWrite()`, `analogRead()`, etc.

Cependant, une bibliothèque *MCCAB_Lib* [1] est disponible et aide les développeurs en fournissant des commandes supplémentaires pour contrôler les divers périphériques matériels de la carte d'apprentissage MCCAB. Vous pouvez télécharger gratuitement cette bibliothèque et l'intégrer dans votre propre croquis. Cette bibliothèque facilite grandement la gestion des périphériques de la carte. La bibliothèque *MCCAB_Lib* contient cinq classes permettant de commander les commutateurs, les LED et le buzzer sur la carte d'apprentissage et peut être facilement incluse dans le croquis de l'utilisateur selon les besoins. Le **tableau 1** présente une liste des classes disponibles.

Grâce à cette bibliothèque, l'utilisateur n'a pas à se soucier de définir des périodes de temps pour le débouclage des commutateurs, de générer des signaux de contrôle multiplex pour la matrice de LED 3 x 3 et de faire clignoter les LED LD10 à LD20 ou L, ou même de générer les fréquences de tonalité du buzzer. Les fonctions de la bibliothèque font cela automatiquement en tâche de fond du programme, sans que l'utilisateur s'en aperçoive.

Listage 1 est un exemple de croquis pour démontrer l'utilisation de la bibliothèque *MCCAB_Lib*.

Sur la ligne 15 du sketch, la variable objet `Led` est déclarée de la classe `LED` de la bibliothèque *MCCAB_Lib*. Le paramètre `LED_PIN` utilisé dans la déclaration de la variable objet `Led` est défini comme une constante dans la ligne 13 indiquant la broche à laquelle la LED est connectée. Cette broche est automatiquement configurée en sortie lors de l'instanciation.

La variable objet `Key` de la classe `KeySwitch` de la bibliothèque *MCCAB_Lib* (déclarée dans la ligne 22) surveille pendant l'exécution (en



Listage 1.

```
/*
 * Sketch which uses pushbutton K4 to toggle LED LD10 on and off using object variables in the
 * "KeySwitch" und "LED" classes in the MCCAB_Lib library.
 * To read the status of pushbutton K4 its necessary to insert a jumper to link position S+K4 (the switch
 * connection) with A3 (GPIO A3 of the microcontroller) on double header strip JP2 of the MCCAB.
 * Insert another jumper (in position D2 of the double header strip J6) to link LED LD10 with the
 * microcontroller GPIO of the MCCAB.
 */

11 #include <MCCAB_Lib.h> // bind the MCCAB_Lib Library to the Sketch
12
13 #define LED_PIN 2 // the LED is connected to pin D2
14
15 LED Led(LED_PIN); // Object-Variable
16
17 //function called by the object-variable "Key" when the switch is closed.
18 void switchTurnedOn() {
19     Led.toggle(); // toggle or flip the state of the LED
20 }
21
22 KeySwitch Key(SK4, ACTIVE_HIGH, switchTurnedOn, nullptr); // Object-Variable
23
24 void setup() { } // nothing to do here...
26 void loop() { } // or here
```

arrière-plan) l'état de l'entrée du commutateur sur la broche SK4, qui lui est passée en paramètre selon sa déclaration. Elle effectue le débouclage du commutateur lorsque le bouton-poussoir K4 est enfoncé ou relâché et appelle la fonction `switchTurnedOn()` lorsque le bouton est enfoncé. La méthode `toggle()` de la classe `LED` de la bibliothèque `MCCAB_Lib` est activée dans la fonction `switchTurnedOn()` de la ligne 19 pour inverser l'état courant de la diode électroluminescente LD10.

Puisque les broches de connexion du commutateur et de la LED sont automatiquement configurées comme entrée et sortie lors de la déclaration des objets, il n'y a rien à ajouter dans la fonction `setup()` de la ligne 24 de ce croquis.

La fonction `loop()` de la ligne 26 ne contient pas non plus d'instructions, car la seule tâche à effectuer dans ce sketch est de commuter l'état de la LED lorsque le bouton K4 est pressé. Cette action est déclenchée par la classe `KeySwitch` en appelant la fonction `switchTurnedOn()`.

En utilisant les classes de la bibliothèque `MCCAB_Lib`, dans des croquis plus longs, vous n'aurez pas besoin d'utiliser les deux fonctions `setup()` et `loop()` du modèle logiciel standard d'Arduino pour vérifier continuellement l'état des composants périphériques, ce qui les libérerait pour des tâches plus importantes.

12 projets, croquis, et 46 exercices

Un manuel d'instruction détaillé pour la carte d'apprentissage MCCAB est disponible et téléchargeable sur [1]. La carte d'apprentissage MCCAB et la bibliothèque `MCCAB_Lib` seront également décrites en détail dans un prochain livre qui sera bientôt disponible (voir note de l'éditeur).

Le livre explique en détail les bases matérielles et logicielles d'un système à microcontrôleur et introduit le langage de programmation C, qui est utilisé pour écrire des croquis Arduino. Le livre porte principalement sur les exercices pratiques. L'apprentissage par la pratique est le concept clé utilisé ici pour vous permettre d'acquérir les compétences dont vous aurez besoin lorsque vous vous lancerez dans la construction de vos propres projets. Dans une section pratique détaillée, vous trouverez 12 croquis de projets et 46 exercices qui vous permettent de développer vos connaissances au fur et à mesure que vous travaillez sur les nombreux exemples. Les exercices sont structurés de telle sorte que le lecteur se voit proposer une tâche à résoudre avec la carte d'apprentissage MCCAB en utilisant les connaissances acquises dans la partie théorique du livre. Pour chaque exercice, il y a ensuite une explication détaillée et un exemple de solution bien commenté qui aide à résoudre les problèmes. ◀

220450-04

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

À propos de l'auteur

Wolfgang Trampert a commencé à développer et à programmer des systèmes à microcontrôleurs après avoir terminé ses études en électronique. Son entreprise d'ingénierie a développé des solutions basées sur les microcontrôleurs pour répondre aux besoins des clients. Il est l'auteur de plusieurs ouvrages et articles spécialisés et anime des formations sur le thème des microcontrôleurs.



Produits

> **MCCAB Training Board (SKU 20295)**
www.elektor.fr/20295

> **Mikrocontroller-Praxiskurs für Arduino-Einsteiger (livre en allemand, SKU 20293)**
www.elektor.fr/20293

LIENS

[1] Bibliothèque `MCCAB_Lib` :
www.elektor.fr/20295



Arduino & Co – Measure, Control, and Hack

Avec une simple carte Arduino Pro Mini et quelques composants, il est possible de réaliser des projets, autrefois irréalisables (ou qui auraient coûté une fortune) facilement et à moindre coût grâce à ce nouveau livre (en anglais) : de simples projets d'éclairage de LED, une station de recharge et de test de batterie rechargeable, et encore plus!

www.elektor.fr/20243



contrôleur pour Spotify

le kit Oplà IoT contient
(presque) tout ce dont vous
avez besoin



Altuğ Bakan (Turquie)

Le kit Arduino Oplà IoT contient la carte pour maker MKR WiFi 1010 et une carte support qui intègre des relais, un écran OLED de forme ronde, des boutons tactiles capacitifs et quelques capteurs. Nous expliquons dans cet article comment construire un contrôleur portable pour le célèbre lecteur de musique Spotify. Certains éléments de sécurité sont bien entendu nécessaires.

Grâce à ses capacités wifi, la carte Arduino MKR WiFi 1010 constitue un cerveau parfait pour votre prochain projet IdO (Internet des objets). Mieux encore, avec le kit Arduino Oplà IoT, vous disposez à la fois d'une carte pour maker et d'une carte support (**figure 1**). Cette dernière contient des relais, un écran OLED de forme ronde et des

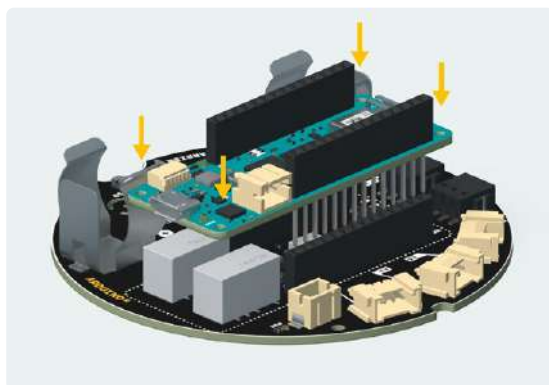


Figure 1, La carte pour maker MKR WiFi 1010 est placée sur la carte support, dotée de relais et d'autres périphériques nécessaires. .

boutons tactiles capacitifs. Le kit comprend également un capteur d'humidité et un capteur infrarouge passif (IRP) (**figure 2**). Il est ainsi facile de mettre en œuvre des dispositifs comme les alarmes de sécurité domestique et l'arrosage automatique des plantes.

La fonction wifi permet également de contrôler des programmes exécutés sur votre PC, s'ils disposent d'une interface réseau. Les boutons tactiles, l'écran, le support de batterie et le boîtier permettent de créer facilement un contrôleur portable pour différents types de logiciels PC, en complément d'une souris et d'un clavier (**figure 3**). Je suis fan du lecteur musical Spotify, et j'ai donc utilisé le kit Oplà pour construire mon propre contrôleur Spotify sans fil. Vous pouvez appuyer sur des boutons pour passer à la chanson suivante ou précédente, ou encore la lire ou la mettre en pause, et augmenter ou diminuer le volume. Pour ce faire, il va sans dire que le lecteur Spotify doit être lancé sur votre PC ou votre smartphone.

Communication sécurisée

Spotify est livré avec une interface de programmation facile à utiliser, qui permet de contrôler votre lecteur via le réseau ; vous aurez besoin pour cela de la licence Spotify Plus. Certains éléments de sécurité sont bien entendu nécessaires. Pour utiliser l'API web de Spotify, basée sur REST, vous devez d'abord vous authentifier sur le serveur de comptes Spotify avec votre nom d'utilisateur et votre mot de passe de connexion. Une fois authentifié, votre logiciel doit envoyer un identifiant (ID) client et un secret client. Le serveur Spotify vous renvoie ensuite un jeton d'accès, que vous devrez émettre à chaque appel de l'API web pour contrôler votre lecteur Spotify. Ce flux



Figure 2 . Le kit Arduino Oplà IoT.



Figure 3 . Le support de batterie permet d'obtenir un kit Oplà IoT portable.

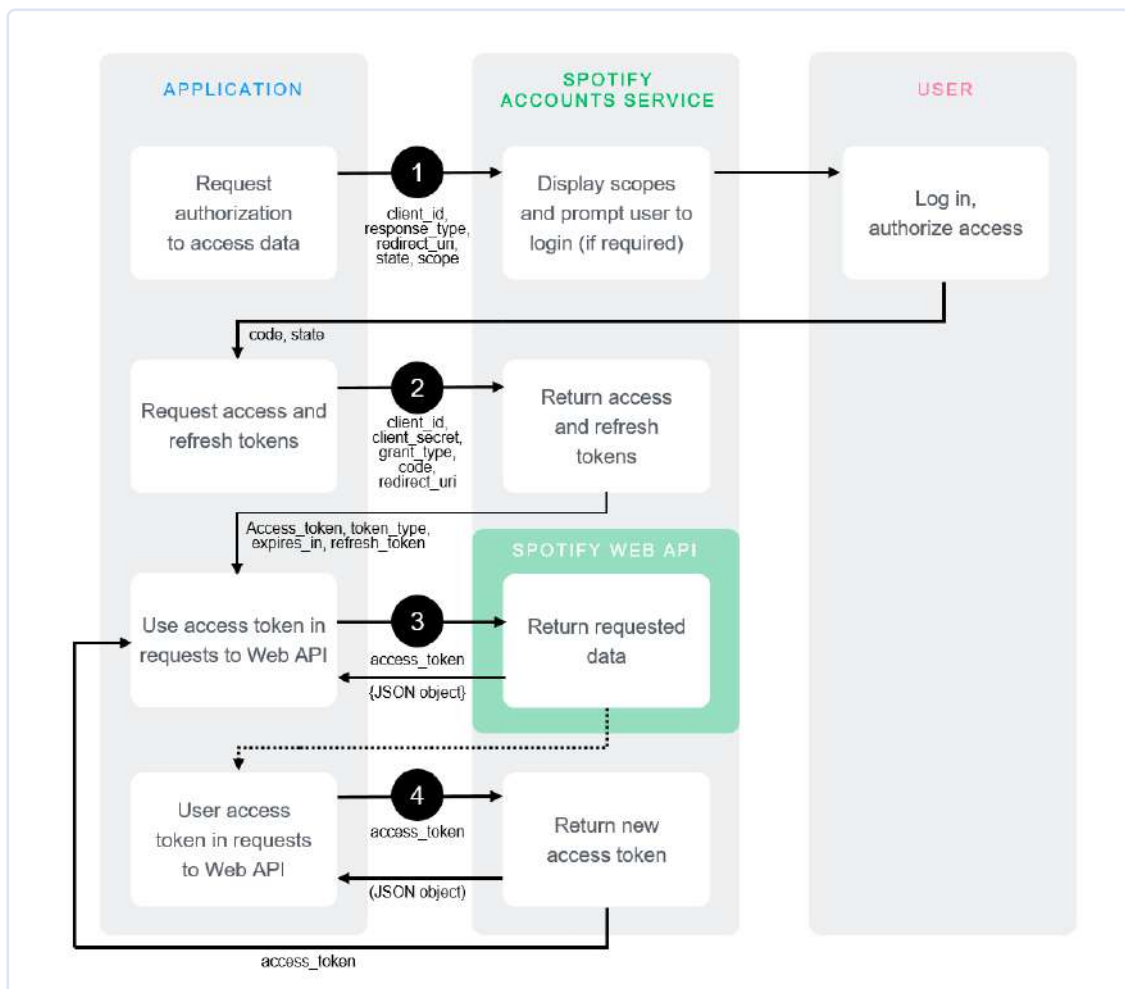


Figure 4 . Le flux d'authentification en plusieurs étapes est basé sur le processus OAuth 2.0 bien connu.

Figure 5. Vous devez créer une « appli » pour obtenir ...

Figure 6. ...votre identifiant et votre secret client.

d'authentification en deux étapes est basé sur le processus *OAuth2* bien connu (voir la **figure 4**).

Comment obtenir votre identifiant et votre secret client ? Il vous suffit d'utiliser le créateur d'applications Spotify App Builder [1] pour concevoir votre propre logiciel PC ou une application mobile servant à contrôler Spotify (**figure 5**). Cependant, nous ne le faisons pas ici ; nous voulons simplement les informations d'identification (voir la **figure 6**). L'identifiant et le secret client doivent être stockés sur notre carte Arduino MKR. Bien sûr, vous pourriez les coder « en dur » dans le croquis Arduino, mais il existe un moyen simplifié et plus sûr de le faire. L'éditeur web Arduino [2] propose un onglet *Secrets*, dans lequel vous pouvez définir des variables d'environnement utilisées ensuite dans votre code (**figure 7**). Il suffit de saisir l'identifiant client Spotify et le secret ainsi que le nom et le mot de passe de votre réseau wifi dans les champs de l'onglet. Si vous compilez et téléchargez le logiciel sur le contrôleur, vos valeurs secrètes individuelles seront également téléchargées pour être utilisées par le code du projet. Dans votre croquis, vous devez remplacer les chaînes de caractères contenant des données sensibles en écrivant une expression `SECRET_xxx` — par exemple : `SECRET_SPOTIFY_CLIENT`.

Authentification

Pour lancer le flux *OAuth2*, vous devez vous authentifier auprès de Spotify. Lorsque vous démarrez le contrôleur Spotify décrit ici, il se connecte au réseau wifi spécifié (maison) et affiche sur l'écran OLED l'adresse IP obtenue par le routeur. Je voulais donner à l'utilisateur une chance de s'authentifier facilement sur Spotify, et j'ai donc créé l'approche suivante. Le contrôleur Arduino génère une petite page web qui sera affichée dans un navigateur si

Figure 7. Saisissez toutes les valeurs privées dans l'onglet *Secrets* de l'éditeur web Arduino, avant de compiler et de télécharger le code.

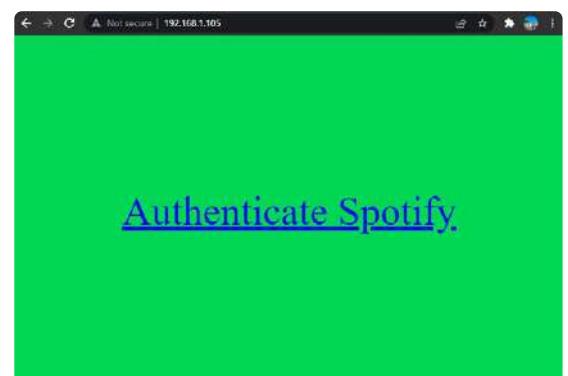


Figure 8. Page web proposée par le contrôleur, pour se connecter à Spotify.

vous entrez l'adresse IP de votre contrôleur (**figure 8**). Cette petite page web contient un lien. (Référez-vous au **listage 1** pour voir comment la page web est produite dans le code Arduino). Si vous appuyez dessus, le navigateur se rend sur la page d'authentification de Spotify, où vous pouvez facilement vous connecter. Il vous sera alors demandé si vous donnez au contrôleur l'autorisation de contrôler Spotify (**figure 9**). À noter : pour que tout cela fonctionne, vous devez

également entrer l'adresse IP du contrôleur en tant que « Redirect URI » (URI de redirection) dans l'éditeur d'application Spotify (**figure 10**). À partir de là, le contrôleur Arduino peut obtenir le jeton d'accès à l'API en envoyant l'identifiant client et le secret à Spotify (**listage 2**). Le jeton d'accès doit être régulièrement réactualisé pendant le fonctionnement. Une fonction du croquis (**listage 3**), appelée toutes les 3000 secondes, permet de réaliser cette actualisation.



Listage 1. Page web pour s'authentifier sur Spotify, mise à disposition par le contrôleur Spotify.

```
String webpage = "<!DOCTYPE html>\n";
webpage += "<html><body>";
webpage += getStyle();
webpage += "<a href=\"https://accounts.spotify.com/authorize?client_id=";
webpage += SPOTIFY_CLIENT;
webpage += "&response_type=code&redirect_uri=http://";
webpage += ip_address;
webpage += "/redirect/&scope=user-read-playback-state user-modify-playback-state\">Authenticate Spotify</a>\n";
webpage += "</body></html>";
wifiClient.print(webpage);
```

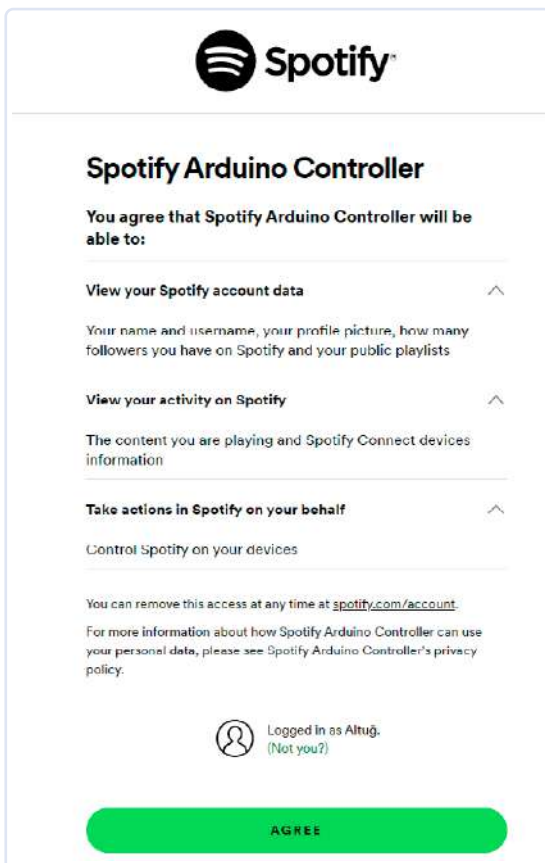
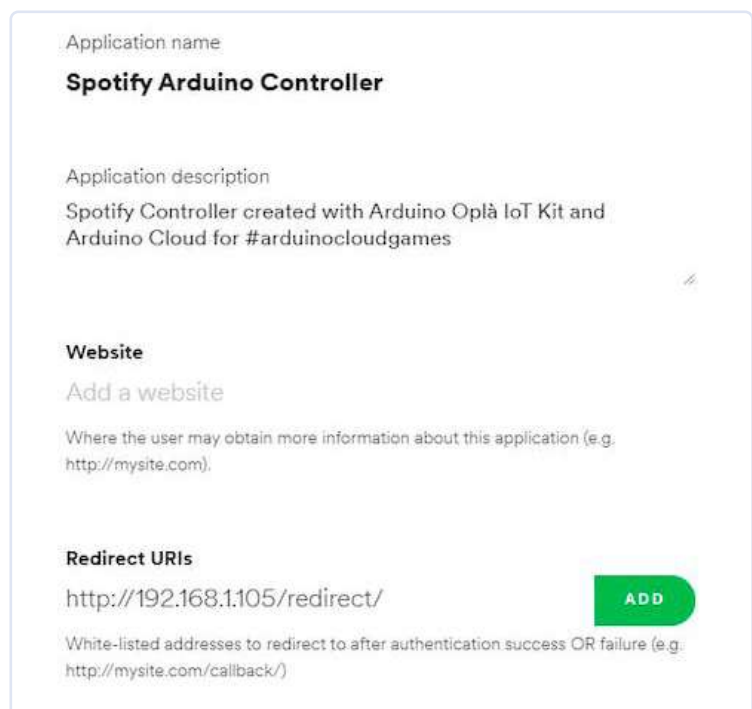


Figure 9. Si vous êtes connecté(e) à Spotify, vous devez donner au contrôleur l'autorisation d'agir en votre nom.

Figure 10. URI de redirection : l'adresse du contrôleur dans votre réseau domestique.





Listage 2. Fonction destinée à obtenir le jeton auprès de Spotify pour une utilisation ultérieure par l'API.

```
// Get the user authorization token
bool getAccessToken(String userCode) {
    String postData = "grant_type=authorization_code&code=" + userCode + "&redirect_uri="
        "http://" + ip_address + "/redirect/";
    authClient.beginRequest();
    authClient.post("/api/token");
    authClient.setHeader("Content-Type", "application/x-www-form-urlencoded");
    authClient.setHeader("Content-Length", postData.length());
    authClient.sendBasicAuth(SPOTIFY_CLIENT, SPOTIFY_SECRET);
    // send the client id and secret for authentication
    authClient.beginBody();
    authClient.print(postData);
    authClient.endRequest();

    // If successful
    if (authClient.responseStatusCode() == 200) {
        lastTokenTime = millis();
        DynamicJsonDocument json(512);
        deserializeJson(json, authClient.responseBody());
        accessToken = json["access_token"].as<String>();
        refreshToken = json["refresh_token"].as<String>();
        return true;
    }
    return false;
}
```



Listage 3. Fonction servant à actualiser le jeton.

```
// Refresh the user authentication token
void refreshAccessToken() {
    String postData = "grant_type=refresh_token&refresh_token=" + refreshToken;
    authClient.beginRequest();
    authClient.post("/api/token");
    authClient.setHeader("Content-Type", "application/x-www-form-urlencoded");
    authClient.setHeader("Content-Length", postData.length());
    authClient.sendBasicAuth(SPOTIFY_CLIENT, SPOTIFY_SECRET);
    // send the client id and secret for authentication
    authClient.beginBody();
    authClient.print(postData);
    authClient.endRequest();

    // If successful
    if (authClient.responseStatusCode() == 200) {
        lastTokenTime = millis();
        DynamicJsonDocument json(256);
        deserializeJson(json, authClient.responseBody());
        accessToken = json["access_token"].as<String>();
    }
}
```



Figure 11. Les fonctions des boutons sont affichées à l'écran.



Listage 4. Exemple d'utilisation de l'API (morceau suivant et précédent).

```
// Skip a song towards a given direction
void skipSong(String direction) {
  apiClient.beginRequest();
  apiClient.post("/v1/me/player/" + direction);
  apiClient.setHeader("Content-Length", 0);
  apiClient.setHeader("Authorization", "Bearer " + accessToken);
  apiClient.endRequest();
}
```

Fonctionnement

Le reste du code est moins complexe. L'appareil affiche le logo Spotify et la fonction des boutons sur l'écran OLED (figure 11). Si l'utilisateur touche un bouton, la fonction d'API correspondante est appelée. Référez-vous au **listage 4** pour voir comment passer d'un morceau au précédent ou au suivant.

Une autre fonction du code demande le statut du lecteur à l'API Spotify. La réponse est une chaîne JSON. J'utilise la bibliothèque *ArduinoJson.h* et certaines de mes propres fonctions pour traiter plus facilement les chaînes JSON. Pour obtenir l'état des boutons, commander les LED et afficher des graphiques sur l'écran OLED, j'utilise la bibliothèque *Arduino_MKRIoTCarrier.h*. Vous pouvez vous plonger dans mon code pour vous en inspirer et réaliser vos propres projets avec le kit Oplà. Il est possible de télécharger mon logiciel à l'adresse [3].

Connexion à la plateforme Arduino Cloud

J'ai également établi une connexion avec la plateforme Arduino Cloud et créé un tableau de bord qui affiche la chanson en cours et le nom de l'artiste à côté du volume de l'appareil (figure 12). Vous pouvez bien sûr créer votre propre tableau de bord personnel, avec les données que vous souhaitez.

Mon projet a remporté la 3^e place des *Arduino Cloud Games 2022* !

(220407-04) VF : Pascal Godart

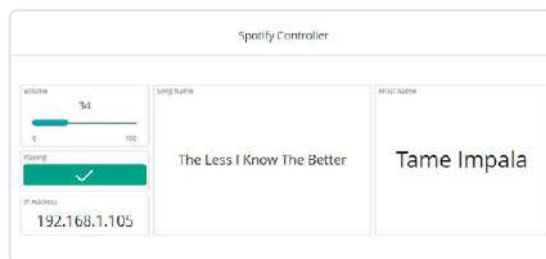


Figure 12. Le morceau en cours et le nom de l'artiste sont envoyés à la plateforme Arduino Cloud, où ils sont visibles sur votre tableau de bord personnel.

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (mail@alt.ug) ou contactez Elektor à (redaction@elektor.fr).

À propos de l'auteur

Altug Bakan a travaillé comme ingénieur électronique, principalement avec des systèmes embarqués. Il aime utiliser Arduino dans son travail pour ses qualités de prototypage rapide et de convivialité. Ses sujets préférés en électronique sont la programmation embarquée dite « bare-metal » (sans système d'exploitation associé) et l'Internet des objets (IdO)..

Produits

> **Arduino Oplà IoT Kit**
www.elektormagazine.fr/arduino-opla-iot-kit

LIENS

- [1] Spotify App Builder : <https://developer.spotify.com/dashboard/>
- [2] Éditeur web Arduino : <https://create.arduino.cc/editor>
- [3] Le projet présent sur create.arduino.cc : <https://create.arduino.cc/projecthub/Altug/opla-spotify-controller-6e7bc4>

créez, déployez et maintenez des applications évolutives et sécurisées

avec Arduino Portenta X8 équipé du mini processeur d'applications i.MX 8M de NXP et de l'élément de sécurité EdgeLock® SE050

Contribué par NXP Semiconductors

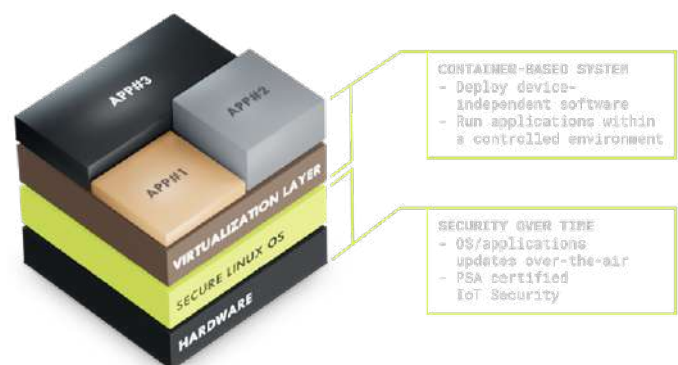
La mise sur le marché d'un dispositif **IdO** implique des efforts de conception et de développement considérables avec des problèmes d'évolutivité, des défis de sécurité et des limitations de dispositifs à chaque coin de rue. L'ajout d'intelligence complexifie encore plus. C'est pourquoi la sélection du bon matériel et du bon logiciel de développement est essentielle pour mettre plus rapidement sur le marché des produits périphériques sécurisés. Cet article présente la plateforme Arduino Portenta X8, un SoM sécurisé de qualité industrielle basé sur le microprocesseur d'applications i.MX 8M de NXP et un élément matériel de sécurité EdgeLock® SE050 intégré. Cette plateforme certifiée PSA est également Arm® SystemReady IR pour une sécurité assu.

Arduino Portenta X8 est un système puissant de qualité industrielle sur un module avec Linux® préchargé à bord, capable d'exécuter des logiciels indépendants du dispositif grâce à son architecture modulaire en conteneur. Il offre deux approches : la flexibilité d'utilisation de Linux combinée à des applications en temps réel via l'environnement Arduino. La connectivité wifi/BLE embarquée permet de mettre à jour le système d'exploitation et les applications à distance, tout en maintenant l'environnement du noyau Linux à un niveau de performance optimal.

Sécurité de pointe

Le système basé sur des conteneurs intègre différentes couches de sécurité, à commencer par la matérielle qui comprend le *Secure Element* de NXP. Il utilise la plateforme DevOps basée sur le cloud de Foundries.io [1] pour réinventer la manière dont les solutions

Linux embarquées sont construites, testées, déployées et maintenues. Le Portenta X8 comprend le système d'exploitation de microplateforme Linux open-source personnalisable, conçu selon les meilleures pratiques de l'industrie en matière de sécurité de bout en bout, de mises à jour *OTA* incrémentielles et de gestion de flotte.

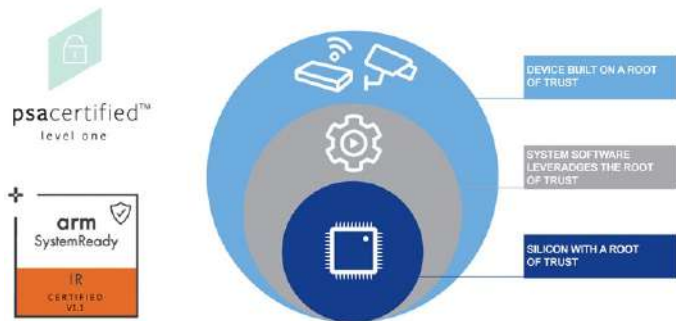


Portenta X8 Container and Security.

La couche de virtualisation permet aux utilisateurs de déployer des logiciels indépendants des appareils et fonctionnant dans un environnement contrôlé. Ils peuvent créer leurs propres conteneurs à l'aide de Docker et télécharger des images préfaites à partir de Docker Hub ou d'autres registres publics disponibles pour concevoir une application sur mesure. Si le développeur souhaite intégrer de l'embarqué, il peut le faire facilement en l'exécutant sur un conteneur, en la plaçant sur la carte et en la testant dès sa sortie de la boîte. Cela offre un large éventail de possibilités en mélangeant les capacités de Linux et d'Arduino.

Portenta X8 a obtenu la certification PSA et l'élément matériel de sécurité NXP EdgeLock SE050 assure la génération de clés, l'accélération des opérations cryptographiques et le stockage sécurisé. Il a également obtenu la certification Arm® SystemReady [2] et a intégré les services Parsec, ce qui en fait l'un des premiers produits Cassini ou dispositifs Edge natifs du cloud disponibles pour les développeurs sur le marché. Il fonctionne de manière transparente avec Fedora IoT, Fedora Server, Debian et la microplateforme Linux. En permettant la migration des charges de travail du cloud vers la périphérie, le

Portenta X8 contribue à une expérience de développeur cloud-native à travers l'écosystème IdO diversifié et sécurisé d'Arm.



Platform Security Architecture.

EdgeLock SE050 - Un ancrage de confiance pour l'IdO

L'EdgeLock SE050 [3] de NXP est un matériel de sécurité discret et inviolable destiné à protéger l'identité d'un dispositif, y compris les clés cryptographiques et les certificats. C'est un élément intégré relié au processeur principal via l'interface I²C. L'EdgeLock SE050 est certifié Critères Communs EAL 6+ pour le matériel et le système d'exploitation. Prêt à l'emploi pour les appareils IdO, il fournit une base de confiance au niveau du circuit intégré et offre une véritable sécurité de bout en bout, de la périphérie au cloud, sans qu'il soit nécessaire d'implémenter un code de sécurité ni de manipuler des clés et des informations d'identification critiques.



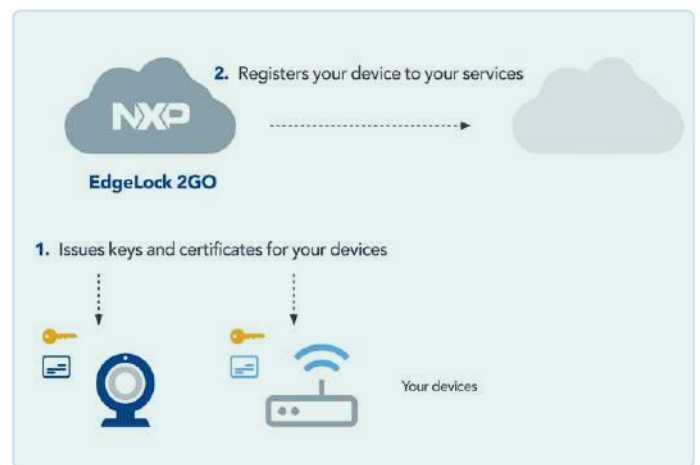
Une base de confiance fondée sur le silicium : élément de sécurité EdgeLock® SE050.

Livré sous la forme d'une solution prête à l'emploi, EdgeLock SE050 offre plusieurs algorithmes et protocoles cryptographiques pré-implémentés et est accompagné d'un paquet de support qui simplifie la conception et réduit le temps de mise sur le marché. En plus des bibliothèques pour différents microcontrôleurs (MCU) et MPU, le paquet de support offre également l'intégration avec les nombreux OS courants, y compris Linux, RTOS et Android.

Les concepteurs de dispositifs IdO sont confrontés à deux défis majeurs lors de la mise en œuvre de l'embarquement des dispositifs sur le

cloud. D'abord, le provisionnement de l'identité du dispositif faisant référence à l'installation des clés et des certificats. Le second défi est la gestion des identités des dispositifs une fois libérés sur le terrain, faisant référence à la mise à jour, l'ajout ou la révocation des clés et des certificats tout au long du cycle de vie du dispositif.

Pour aider les concepteurs à relever ces défis, NXP propose le service géré EdgeLock 2GO [4]. La plateforme est une combinaison de matériel et de services spécialement conçue qui établit une base de confiance fondée sur le silicium. EdgeLock 2GO émet les identités requises pour les dispositifs IdO et installe les références en toute sécurité dans le matériel EdgeLock SE050. Il enregistre également automatiquement le dispositif IdO directement auprès du cloud.



NXP gère les informations d'identification des dispositifs.

TCe service flexible prend en charge plusieurs types de justificatifs d'identité et applique différentes configurations en fonction du projet. Les accréditations peuvent être renouvelées ou ajoutées aux dispositifs mis en service sur le terrain. Avec la mise en place d'EdgeLock SE050 et de l'EdgeLock 2GO, les utilisateurs bénéficient d'une solution de bout en bout qui est simple, sûre et flexible.

L'IdO continue de se développer, mais les risques aussi. La combinaison EdgeLock de NXP, avec sa sécurité matérielle et son service de gestion des informations d'identification, offre aux fabricants de dispositifs un moyen plus sûr de faire des affaires. En prenant en charge le déploiement d'un appareil, NXP EdgeLock réduit les délais de mise sur le marché et les coûts quotidiens d'exploitation d'un déploiement IdO tout en ayant la certitude que les appareils sont protégés par une sécurité de haut niveau.

Libérez la puissance : plus de vitesse et une meilleure efficacité

Le SoC i.MX 8M Mini [5] est le premier processeur d'applications multi-cœur embarqué de NXP construit à l'aide de la technologie de processeur 14LPC FinFET avancée, offrant plus de vitesse et une meilleure efficacité énergétique. La famille de processeurs d'applications i.MX 8M Mini associe la haute performance informatique, l'efficacité énergétique et la sécurité embarquée nécessaires à la croissance rapide de l'informatique des nœuds de périphérie, du streaming multimédia et des applications de *machine learning*.



Le SoC i.MX 8M Mini est proposé dans des variantes à un, deux et quatre cœurs utilisant le processeur Arm® Cortex®-A53 et fonctionnant jusqu'à 1,8 GHz par cœur. Livré dans un processus avancé à faible consommation, le complexe de cœurs est optimisé pour un fonctionnement sans ventilateur, un faible coût thermique du système et une longue durée de vie de la batterie. Les cœurs Cortex-A peuvent être mis hors tension tandis que le sous-système Cortex-M4 assure la surveillance du système en temps réel et à faible consommation. Le contrôleur DRAM prend en charge les mémoires 32 bits/16 bits LPDDR4, DDR4 et DDR3L, offrant ainsi une grande souplesse de conception du système.

Les options de cœur du i.MX 8M Mini sont optimisées pour une consommation ultra-faible, voire inférieure à un watt dans certaines applications spécifiques, mais offrent la puissance de traitement nécessaire pour les applications grand public, audio, industrielles, de machine learning et d'inférence sur une gamme de fournisseurs de cloud. Le SoC i.MX 8M Mini intègre également l'accélération matérielle de la vidéo 1080p pour permettre des applications vidéo bidirectionnelles, des graphiques 2D et 3D pour offrir une expérience visuelle IHM riche, et des capacités audio avancées pour permettre des applications riches en sons. Une sélection étendue d'interfaces à haut débit permet une connectivité plus large du système et vise une qualification de niveau industriel.

Les exemples d'application comprennent :

► Automatisation industrielle

Le Portenta X8 peut alors faire office de passerelle multiprotocole, en envoyant des données au cloud ou au système ERP via wifi, LoRa, NB/IoT, LTE Cat.M1.

La disponibilité de conteneurs Linux comme ROS au sein de l'environnement Arduino fait du Portenta X8 une solution idéale pour les véhicules guidés autonomes.

► Automatisation des bâtiments

Interagissant avec des capteurs intelligents sur le plan environnemental, Portenta X8 permet la mise en œuvre de ML et de traitement d'images en temps réel sur la périphérie.

Les kiosques intelligents s'appuient généralement sur plusieurs composants (par exemple, des lecteurs de cartes, des caméras, des microphones), ce qui nécessite une sélection diversifiée d'E/S. Associé à un Max Carrier, le Portenta X8 assure la connectivité wifi et permet aux administrateurs de surveiller à distance l'utilisation des machines.

Le Portenta X8 peut simultanément contrôler les systèmes CVC, allumer/éteindre les appareils intelligents, régler de manière autonome l'éclairage et contrôler les accès en périphérie.

Commencez à développer dès aujourd'hui avec le SOM Portenta X8 [6] de qualité industrielle, sécurisé et doté d'une densité de calcul exceptionnelle. ◀

220576-04

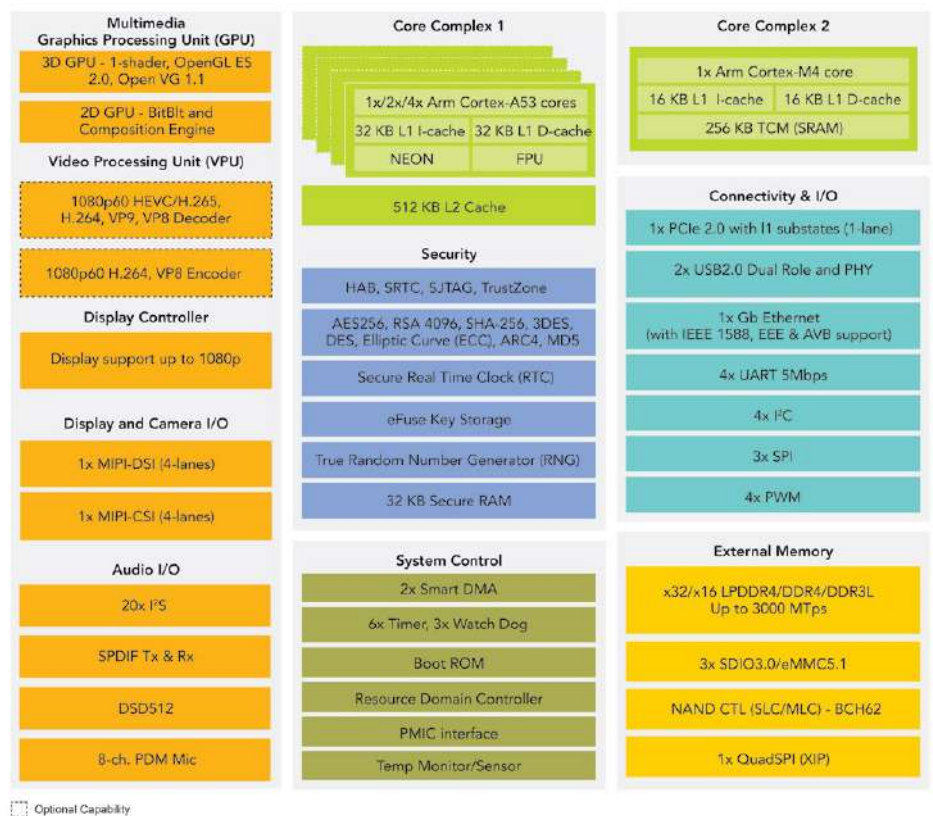


Schéma fonctionnel du processeur d'applications i.MX 8M Mini

LIENS

- [1] Foundries.io : <https://foundries.io/>
- [2] Arm SystemReady : <https://www.arm.com/architecture/system-architectures/systemready-certification-program>
- [3] EdgeLock SE050 : <https://bit.ly/EdgeLockSE050>
- [4] EdgeLock 2GO : <https://bit.ly/EdgeLock2GO>
- [5] i.MX 8M Mini : <https://bit.ly/iMX8MMini>
- [6] Portenta X8 SOM : <https://www.arduino.cc/pro/hardware/product/portenta-x8>

Nouveau projet Aduino ou électronique ?

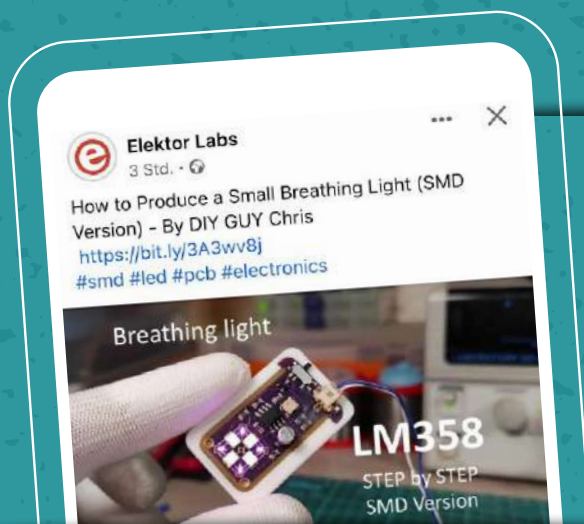
Partagez-le avec notre communauté !



Suivez-nous sur :



www.twitter.com/ElektorFR



www.instagram.com/elektorlabs



www.facebook.com/ElektorFR



Édition bonus
dévoilée !

Disponible la semaine 52

▲
Daria Baradel
ses collègues



...g...ées...ées, ce
sont les retards imprévisibles et les augmentations

Édition bonus dévoilée !

Disponible la semaine 52



maire. Tout ne se passe pas toujours de la même manière. Les changements de monde sont faits par des humains. Les changements de monde sont faits par des humains.



Édition bonus
dévoilée !

Disponible la semaine 52

locaux pour ses cartes ?



MicroPython Enters the World of Arduino

with Stuart Cording & Sebastian Romero

MicroPython has made it to the world of Arduino, providing the first significant alternative to programming in C and C++. So, what's all the fuss, how easy is it to use, and who can benefit from programming in this, for microcontrollers, relatively new language? Stuart Cording will speak with Sebastian Romero (Head of Content, Arduino) during our live webinar to find out more.

Join for free

www.elektor.com/webinar-MicroPython



Édition bonus dévoilée !

Disponible la semaine 52



circuit intégré qui comprend un processeur, de la RAM, un stockage flash ainsi que des entrées/sorties pour l'interfaçage avec le monde réel. C'est ni plus ni moins

pour désigner collectivement tout élément matériel et logiciel de nouveaux objets électroniques.

Édition bonus dévoilée !

Disponible la semaine 52



certains sont aujourd'hui des fonctionnalités équivalentes à celles de nombreux ordinateurs personnels et tablettes.

Édition bonus dévoilée !

Disponible la semaine 52



Les tests des cartes SBC sont désormais effectués
par des entreprises de conception et de fabrication de

Édition bonus dévoilée !

Disponible la semaine 52



[5] MAX-HEART-BEAT : <https://elektor.link/MouserMaxHeartBeat>

[6] MAX-ECG-MONITOR : <https://elektor.link/MouserMaxECGMonitor>

[7] Nordic Semiconductor Thingy:91™ Multisensor Prototyping Kit : <https://elektor.link/MouserThingy91>

Édition bonus dévoilée !

Disponible la semaine 01

▲
Figure 1
atelier «

l'
qu



...ant un logiciel permettant de créer des symphonie par le seul fait de jouer ensemble !



Édition bonus dévoilée !

Disponible la semaine 01



our la



Figure

Figure

Édition bonus dévoilée !

Disponible la semaine 01



une grande variété de timbres venant de la puce. La correspond à la note que nous recommandons



Édition bonus dévoilée !

Disponible la semaine 01



1 2
100

Édition bonus dévoilée !

Disponible la semaine 01

Figure 9.
des bro





Édition bonus dévoilée !

Sel
mer
pro
trou

Co

Not
par
une
ave
pro

Disponible la semaine 01



Mettez la main

Édition bonus
dévoilée !

Disponible la semaine 01

B
C

La
rob
Arc
nou
les
Bra
de c
tâch
des
suiv
pan
men
offre
évol
un n
écran
teurs
totale

www

Ardu



projets de n'importe où dans le monde.

www.elektor.fr/19351

Édition bonus dévoilée !

Disponible la semaine 01

UNO, et commander le kit de
synthé commandé par la carte UNO !

www.elektor.fr/20330

Arduino Sensor Kit Base

www.elektor.fr/19944



à la possibilité d'exécuter des app.
d'Edge Computing (IA).
www.elektor.fr/19936

Supporting **Arduino Resellers**

Cette édition spéciale Arduino du magazine Elektor a été rendue réalisable grâce au soutien de ces membres de la communauté des revendeurs Arduino.

N'hésitez pas à les consulter pour tous vos besoins liés à l'Arduino !

GOTRON
AALST GENT HASSELT



www.gotron.be

HELLAS
digital



www.hellasdigital.gr

TINYTRONICS



www.tinytronics.nl

 **Paradisetronic.com**



www.paradisetronic.com

 **Techni Science.**



www.techniscience.com

 **reichelt**
elektronik – The best part of your project



www.reichelt.com/arduino

 **WHADDA**
MADE BY VELLEMAN



www.whadda.com

 **KUBII**



www.kubii.fr

GO TRONIC
ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES



www.gotronic.fr