

débuter avec le Portenta X8

gestion sécurisée des logiciels
avec les conteneurs



Figure 1. L'Arduino Portenta X8.

Benjamin Dannegård (Arduino)

L'Arduino Portenta X8 est un puissant système sur module (SoM) de qualité industrielle qui fonctionne avec une distribution Yocto Linux de Foundries.io. Grâce à l'intégration de l'Arduino Cloud, il est facile de créer une FoundriesFactory, de composer des conteneurs Docker et de les télécharger en toute sécurité sur votre ou vos cartes.

Le révolutionnaire Arduino Portenta X8 (**figure 1**) est un puissant système sur module (SoM) de qualité industrielle préchargé avec l'OS Linux, ce qui en fait une solution clés en main apte à exécuter des logiciels indépendants du dispositif, grâce à son architecture modulaire en conteneur. La connectivité wifi et *Bluetooth Low Energy* (BLE) intégrée permet d'effectuer à distance des mises à jour de l'OS et des applications, en maintenant toujours l'environnement du noyau Linux à un niveau de performance optimal. La combinaison du microprocesseur et du microcontrôleur du X8 offre une flexibilité sans précédent pour exécuter des applications Linux et effectuer des tâches en temps réel simultanément et en toute sécurité. Les nombreuses bibliothèques Arduino disponibles, associées à une distribution Linux basée sur des conteneurs, permettent aux professionnels de l'informatique, aux intégrateurs de systèmes et aux sociétés de conseil de construire et de

dynamiser une grande variété de solutions pour les contextes industriels. Il se prête également aux applications d'automatisation des bâtiments et d'agriculture intelligente. Quelques applications possibles :

- Ordinateur de périphérie connecté pour la production
- Véhicules guidés autonomes (AGV)
- Bornes interactives sécurisées Full-HD et affichage numérique
- Systèmes de contrôle pour bureaux et habitations
- Navigation et contrôle pour l'agriculture intelligente
- Analyse comportementale pour les bureaux et les usines

Concepts de base

L'Arduino Portenta X8 utilise un environnement Linux embarqué, ce qui signifie qu'il nécessite une distribution de base, un

mécanisme pour la mettre à jour et quelques applications qui peuvent fonctionner sur la carte. Comme base, le X8 utilise une distribution Linux construite avec le *Yocto Project*, avec des applications installées et empaquetées sous forme de conteneurs confinés (**figure 2**). Le projet Yocto a été créé spécifiquement pour les systèmes embarqués afin de garantir la flexibilité nécessaire pour atteindre les tailles d'encombrement des cibles et ajuster les fonctionnalités.

En collaboration avec Foundries.io [1], nous avons développé une solution simple pour connecter l'Arduino Portenta X8 à leur distribution *FoundriesFactory*. En connectant le Portenta X8 à un ordinateur, on peut immédiatement accéder à la page web de configuration (**figure 3**). Cela permet un processus d'installation fluide et rapide. On peut y configurer le wifi de la carte, puis la connecter facilement à une FoundriesFactory de votre choix (voir **figure 4**). Grâce à l'intégration de Foundries.io avec Arduino Cloud, vous pouvez facilement créer votre FoundriesFactory directement à partir de la page Arduino Cloud. La page *FoundriesFactory* vous permet d'ajouter des membres, ce qui facilite le suivi d'une équipe qui doit avoir accès aux Portenta X8 liés à la Factory. On peut également constituer des équipes pour une meilleure gestion. La page présente aussi une liste de tous les appareils liés à la Factory, avec leurs noms et les versions des conteneurs actuellement téléchargés sur les cartes. Il existe enfin une page qui répertorie toutes les différentes versions de conteneurs téléchargées vers l'usine. Tout cela permet d'avoir une

vue d'ensemble et de gérer facilement les personnes de l'équipe, ainsi que les appareils et les cibles. Ceci permet déjà de gérer facilement la flotte de dispositifs, mais Waves, l'outil de gestion de flotte de Foundries.io, permet un contrôle encore plus poussé. Pour mieux comprendre comment cet outil fonctionne avec le Portenta X8, consultez le tutoriel *FoundriesFactory Waves Fleet Management* pour le X8 [2].

Initialement, Foundries.io a créé sa distribution générique (mais pas trop) à partir de Yocto avec un minimum de logiciels installés. Par défaut, elle met en œuvre des fonctionnalités de cybersécurité de haut niveau telles que OP-TEE, un système qui permet de sécuriser l'OS, en prenant en charge de manière facultative un secure element et OSTREE, un système permettant de vérifier chaque fichier et répertoire du système. Cela rend leur solution idéale pour les applications professionnelles. Un mécanisme personnalisé de mise à jour OTA du système est basé sur un client fonctionnant sur une cible et un robuste serveur du nuage. Foundries.io a connecté docker-compose comme moyen de déployer une solution logicielle sur une cible. Cela revient à avoir un app-store pour un appareil particulier - à la différence qu'on n'installe pas une application mais un conteneur qui peut contenir une distribution complète ou une distribution minimale exécutant uniquement notre application ou notre ensemble d'applications. En plus de cela, ils ont également développé l'aspect cloud. Cela signifie que vous pouvez utiliser FoundriesFactory, un service d'abonnement DevSecOps en nuage, pour construire, tester, déployer et maintenir des produits IoT et de périphérie sécurisés et actualisables. On obtient au même endroit un identifiant unique et les générations automatiques du système de base et de ses conteneurs.

Conteneurs

Comme brièvement mentionné précédemment, les conteneurs constituent une partie importante de la fonctionnalité du X8. Un conteneur est une unité logicielle standard qui regroupe le code et toutes ses dépendances, afin que l'application s'exécute rapidement et de manière fiable d'un appareil à un autre. Une image de conteneur Docker est un paquet de logiciels léger, autonome et exécutable qui comporte tout ce qui est nécessaire pour exécuter une application. Avec l'utilisation intégrée de Docker sur le Portenta X8, il est facile de télécharger des conteneurs déjà existants ou de créer votre propre conteneur hautement personnalisable.

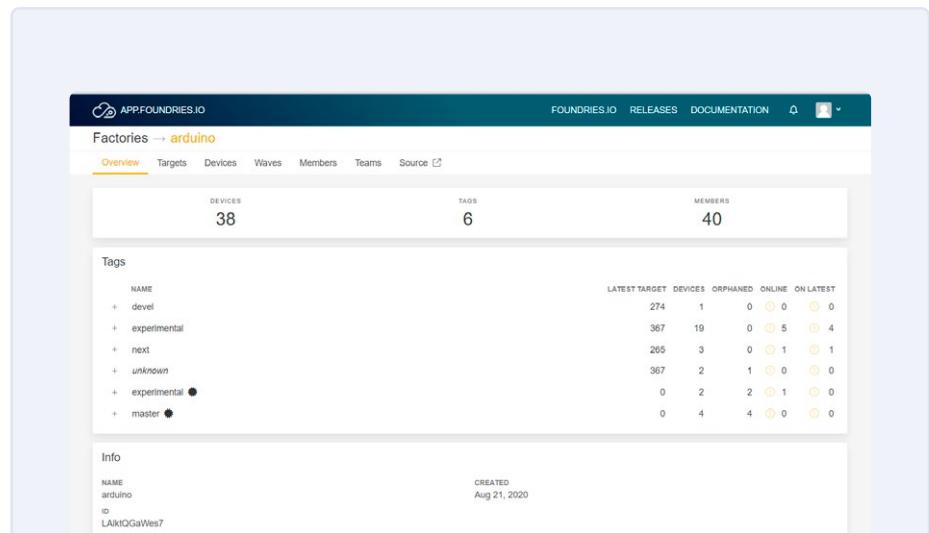


Figure 2. Infographie du système.

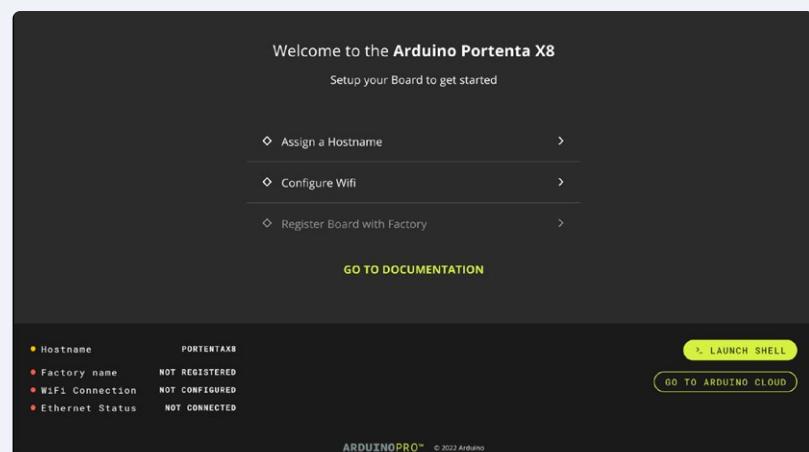


Figure 3. Page web de configuration du Portenta X8.

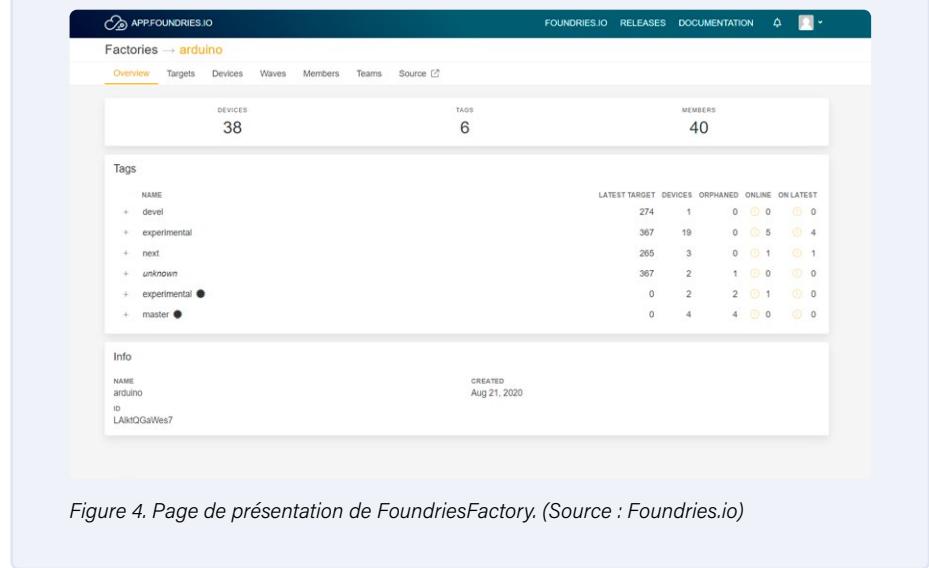


Figure 4. Page de présentation de FoundriesFactory. (Source : Foundries.io)

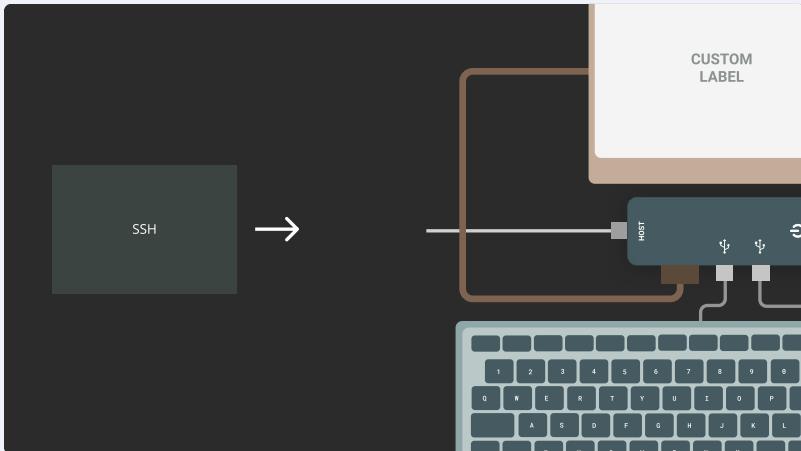


Figure 5. Illustration de la connexion du Portenta X8.



Figure 6. Démo fonctionnant sur Portenta X8.

L'utilisation d'un site tel que Docker Hub [3], qui contient une pléthore de conteneurs prêts à l'emploi, permet l'emploi du Portenta X8 dans de nombreux domaines différents. Lors de la création d'un conteneur à partir d'une image, celle-ci doit contenir l'ensemble du système de fichiers du conteneur. Cela signifie qu'elle doit contenir tout ce qui est nécessaire pour exécuter une application, toutes les dépendances, les configurations, les scripts, les binaires, etc. L'image contient également d'autres configurations pour le conteneur, comme des variables d'environnement, une commande par défaut à exécuter et d'autres métadonnées. En revanche, un conteneur est une application unique, alors que vous voulez en utiliser plusieurs, donc, avec le X8, nous utilisons docker-compose, qui nous aide à les gérer. La structure de fichiers d'une application docker-compose peut ressembler à ceci :

- `docker-build.conf` contient la commande minimale de « test d'unité » à exécuter sur le conteneur pour prouver qu'il fonctionne.
- `docker-compose.yml` définit le nom de l'application au sein de la Factory, ses permissions et les paramètres des conteneurs impliqués
- `Dockerfile` est un fichier qui construit une image Docker. L'image contient toutes les dépendances dont l'application Python a besoin.
- `requirements.txt` contient les dépendances Python que votre application devra copier et télécharger
- Le répertoire `src` contient le code source de l'application que vous voulez exécuter dans le conteneur ou un script de démarrage.

Cette procédure est facilitée grâce à Foundries.io, avec un script de construction de conteneur qui s'exécute sur le serveur de construction CI associé à FoundriesFactory, et qui est déclenché lors du dépôt (*push*) dans le répertoire de la Factory. Nous pouvons ainsi créer une nouvelle application docker-compose, l'ajouter au dépôt et l'archiver ou la déposer. Après quelques instants, la page de la Factory indiquera que le script de construction du conteneur a été déclenché et qu'il est en train de construire la nouvelle application docker-compose. Tout cela nous aide à constituer une plateforme Linux embarquée qui exécute une distribution Yocto personnalisée, sécurisée et capable de résister à un audit sévère de cybersécurité, et où l'utilisateur peut faire tout ce qu'il a en tête en déployant des applications et du code personnalisé avec des conteneurs.

Fonctionnalité RPC

L'infrastructure de conteneurs fournie par Arduino nous permet d'exécuter un traitement en temps réel du côté Arduino tout en faisant tourner un système d'exploitation à part entière du côté Linux. Grâce à l'architecture du X8, le microcontrôleur peut prendre en charge la gestion de certains périphériques et échanger les données requises entre le microcontrôleur et une application Python (on peut aussi utiliser n'importe quel langage de programmation ou de script pour l'application) qui fonctionne du côté Linux. Le mécanisme de communication utilisé à cette fin est appelé RPC (*Remote Procedure Call*). En utilisant l'infrastructure Docker, il est possible de créer une application Python qui utilise le mécanisme RPC pour échanger des données entre le microcontrôleur et l'iMX8, qui exécute le système d'exploitation Linux. Si vous souhaitez approfondir cette fonctionnalité, consultez le tutoriel *Exchange Between Python on Linux and an Arduino Sketch tutorial* [4].

Portenta X8 en action

Voyons à quoi ressemble le processus d'utilisation du Portenta X8 avec FoundriesFactory et les conteneurs avec une démonstration rapide. Le mini processeur NXP i.MX 8M de l'Arduino Portenta X8 peut être utilisé pour le rendu 3D. Cela nous permettra d'afficher du contenu 3D sur un écran ou une sortie vidéo. Le dispositif utilise OpenGL pour traiter les calculs liés à la 3D. Dans cette démo, nous allons effectuer le rendu d'un contenu web à partir d'Internet en utilisant WebGL et l'afficher sur un écran au moyen d'un hub USB pourvu

d'une sortie HDMI. Il existe de nombreux moyens de communiquer avec votre carte, que ce soit avec ou sans fil, puis d'utiliser ADB ou SSH pour communiquer avec le X8 via une interface en ligne de commande. Comme le Portenta X8 est un dispositif Linux, on peut utiliser des commandes Linux normales pour créer des fichiers, changer des répertoires, etc. Après avoir suivi le processus de configuration, le dernier micrologiciel du X8 sera installé. Cela facilitera aussi beaucoup le reste du processus. Maintenant que la carte est liée à une FoundriesFactory, on peut facilement télécharger le conteneur nécessaire sur le Portenta X8 et l'exécuter en seulement quelques lignes avec SSH ou ADB. Avec FoundriesFactory, une cible contenant cet exemple peut facilement être téléchargée sur un dispositif lié. Si vous préférez une méthode locale, il est tout aussi simple de télécharger le référentiel, de le pousser vers le dispositif avec le shell ssh ou adb, et de l'installer et de l'exécuter en quelques commandes. Une fois que tout est téléchargé et fonctionne, le Portenta X8 peut être connecté à un écran en utilisant un hub USB (**figure 5**). L'exemple Aquarium 3D de WebGL sera alors lu (**figure 6**). On peut aussi connecter une souris au hub USB pour interagir avec la démo sur l'écran. Cela permet d'évaluer les capacités de rendu 3D du X8 avec WebGL, de tester différents réglages ou d'accéder à d'autres informations pertinentes si nécessaire. Une autre option consiste à modifier le conteneur à l'aide d'un éditeur de texte Linux afin de changer l'URL et afficher autre chose à l'écran. Si vous souhaitez reproduire ce processus ou l'examiner plus en détail, consultez notre tutoriel, *Output WebGL Content on a Screen* [5]. Sur le site documentaire d'Arduino, vous pouvez également trouver d'autres

tutoriels qui montrent les possibilités du X8 et son utilisation des conteneurs docker. Pour voir comment faire fonctionner un conteneur de base de données sur le Portenta X8 hébergeant un site WordPress, consultez notre tutoriel *Running Wordpress and Database Containers on the Portenta X8* [6]. Peut-être êtes-vous plus intéressé à voir comment le Portenta X8 peut fournir une solution pour l'enregistrement de données MQTT ? Dans ce cas, vous devriez consulter le tutoriel *Data logging with MQTT, Node-RED, InfluxDB and Grafana* [7]. Si vous souhaitez étendre encore plus les capacités du X8, branchez-le sur une carte Arduino Portenta Max Carrier. Vous pourrez alors, par exemple, construire une passerelle multiprotocole. Pour cela, consultez notre tutoriel *Build a Multi-Protocol Gateway with Portenta X8 & Max Carrier* [8].

Maintenant que vous avez une idée de ce

dont l'Arduino Portenta X8 est capable, de ses concepts de base et des types d'applications pour lesquelles il peut être utilisé, c'est à vous de le mettre en action et d'en tirer le meilleur parti. 

(220546-04) — VF : Denis Lafourcade

À propos de l'auteur



Benjamin Dannegård est un concepteur d'interactions originaire de Malmö, en Suède. Il s'intéresse à tout ce qui est technique et intello. Il travaille avec Arduino depuis deux ans comme maker et créateur de contenu pro.



Produits

Vous recherchez les principaux éléments mentionnés dans cet article ? Arduino et Elektor s'occupent de vous !

- **Arduino Portenta X8**
www.elektrormagazine.fr/arduino-portenta-x8
- **Arduino Portenta Max Carrier**
www.elektrormagazine.fr/arduino-portenta-max-carrier
- **Arduino Portenta H7**
www.elektrormagazine.fr/arduino-portenta-h7
- **Arduino Portenta Max Carrier**
<https://elektor.fr/arduino-portenta-max-carrier>

LIENS

- [1] Foundries.io : <https://foundries.io>
- [2] Tutoriel : « Using FoundriesFactory Waves Fleet Management for the X8 » :
<https://docs.arduino.cc/tutorials/portenta-x8/waves-fleet-managment>
- [3] Bibliothèque d'images conteneur Docker Hub : <https://hub.docker.com>
- [4] Tutoriel : « Data Exchange Between Python on Linux and an Arduino Sketch » :
<https://docs.arduino.cc/tutorials/portenta-x8/python-arduino-data-exchange>
- [5] Tutoriel : « Output WebGL Content on a Screen » : <https://docs.arduino.cc/tutorials/portenta-x8/display-output-webgl>
- [6] Tutoriel : « Running Wordpress and Database Containers on the Portenta X8 » :
<https://docs.arduino.cc/tutorials/portenta-x8/wordpress-webserver>
- [7] Tutoriel : « Data Logging with MQTT, Node-RED, InfluxDB and Grafana » :
<https://docs.arduino.cc/tutorials/portenta-x8/datalogging-iot>
- [8] Tutoriel : « Build Multi-Protocol Gateway with Portenta X8 and Max Carrier » :
<https://docs.arduino.cc/tutorials/portenta-x8/multi-protocol-gateway>