

Aan de slag met de Portenta X8

software veilig beheren
met containers



Figuur 1. De Arduino Portenta X8.

Benjamin Dannegård (Arduino)

De Arduino Portenta X8 is een krachtig, industrieel systeem-op-module (SoM) waarop een Yocto Linux-distributie van Foundries.io draait. Dankzij de integratie in de Arduino Cloud is het eenvoudig om een FoundriesFactory te maken, Docker-containers samen te stellen en deze veilig naar je board(s) te laden.

De baanbrekende Arduino Portenta X8 (figuur 1) is een krachtig, industrieel systeem-op-module (SoM) met Linux OS voorgeïnstalleerd, waardoor het een plug&play-oplossing is die device-onafhankelijke software kan draaien dankzij de modulaire container-architectuur. Vanwege de ingebouwde WiFi en Bluetooth Low Energy- (BLE) connectiviteit kunnen OS- en app-updates op afstand worden uitgevoerd, waarbij de Linux-kernel omgeving altijd op het hoogste prestatieniveau blijft. De combinatie van microprocessor en microcontroller van de X8 biedt ongekende flexibiliteit om Linux-apps te draaien en realtime-taken tegelijk en veilig uit te voeren. Dankzij de uitgebreide beschikbare bibliotheken van Arduino, samen met een op containers gebaseerde Linux-distributie, kunnen IT-professionals, systeemintegratoren en adviesbureaus een grote verscheidenheid aan oplossingen voor industriële

context bouwen en stimuleren. Het leent zich ook voor gebouwautomatisering en slimme landbouwtoepassingen. Enkele toepassingen zouden kunnen zijn:

- edge computing voor productie;
- autonoom geleide voertuigen (AGV);
- interactieve full-HD beveiligde kiosken en digital signage;
- kantoor- en huisbesturingssystemen;
- navigatie en besturing voor slimme landbouw;
- gedragsanalyse voor kantoren en fabrieken.

Kernbegrippen

De Arduino Portenta X8 gebruikt een embedded Linux-omgeving, wat betekent dat het een basisdistributie moet hebben – een mechanisme om het te actualiseren

en enkele toepassingen die op het board kunnen draaien. Als basis gebruikt de X8 een Linux-distributie gebouwd met het Yocto Project met apps die zijn geïnstalleerd en verpakt als afgesloten containers (figuur 2). Het Yocto Project is speciaal gemaakt voor embedded systemen om ervoor te zorgen dat het nooit ontbreekt aan de flexibiliteit die nodig is om de gewenste footprint en functionaliteit te realiseren.

Samen met Foundries.io [1] hebben we een eenvoudige oplossing ontwikkeld om de Arduino Portenta X8 aan te sluiten op hun FoundriesFactory-distributie. Door de Portenta X8 aan te sluiten op een computer, krijgen we direct toegang tot de setup-webpagina (figuur 3). Dit zorgt voor een soepel en snel installatieproces. Hier kan de WiFi van het board worden ingesteld, en vervolgens eenvoudig worden gekoppeld aan een FoundriesFactory naar keuze (zie figuur 4). Door Foundries.io's integratie met Arduino Cloud kun je gemakkelijk je FoundriesFactory opzetten direct vanaf de Arduino Cloud-pagina.

Op de FoundriesFactory-pagina kun je leden toevoegen, zodat je eenvoudig kunt bijhouden welke leden van een team toegang moeten hebben tot de Portenta X8's die aan de Factory zijn gekoppeld. Ook is het mogelijk om teams op te zetten voor een beter beheer. Op de pagina vind je ook een lijst van alle devices die aan de Factory zijn gekoppeld, samen met hun namen en de versies van de containers die momenteel op de computer zijn geladen. Er is ook een pagina met een overzicht van

alle verschillende containerversies die naar de Factory zijn overgedragen. Dit alles zorgt voor een eenvoudig overzicht en beheer van de leden in het team, evenals de devices en de goals. Dit maakt gemakkelijk vlootbeheer mogelijk voor iemands devices, maar de tool voor vlootbeheer van Foundries.io, Waves, zorgt voor nog meer controle. Voor een beter begrip van hoe deze tool werkt met de Portenta X8, bekijk de FoundriesFactory Waves Fleet Management tutorial voor de X8 [2].

Foundries.io heeft zijn basisdistributie gebaseerd op Yocto met een minimum aan geïnstalleerde software. Het implementeert standaard cyberbeveiligingsfuncties van het hoogste niveau, zoals OP-TEE, een systeem waarmee het besturingssysteem kan worden beveiligd, eventueel met ondersteuning van een beveiligd element, en OSTREE, een systeem om elk bestand en elke map in het systeem te controleren. Dit maakt hun oplossing ideaal voor professionele toepassingen. Een aangepast OTA-mechanisme voor systeemupdates is gebaseerd op een client die draait op een target en een robuuste cloud-server. Foundries.io koppelde docker-compose als een manier om een softwareoplossing te implementeren op een target. Dit is alsof we een app-store hebben voor een bepaald device – met het verschil dat we geen app installeren maar een container die een hele distributie kan bevatten of een minimale distributie waarop alleen onze app of onze set apps draait. Daarnaast hebben ze ook de cloud side ontwikkeld. Dit betekent dat je FoundriesFactory, een cloud DevSecOps abonnementsdienst, kunt gebruiken om veilige, actualiseerbare IoT-en Edge-producten te bouwen, te testen, te implementeren en te onderhouden. Dit biedt een unieke ID en automatische builds van het basissysteem en containers voor dit systeem op één plek.

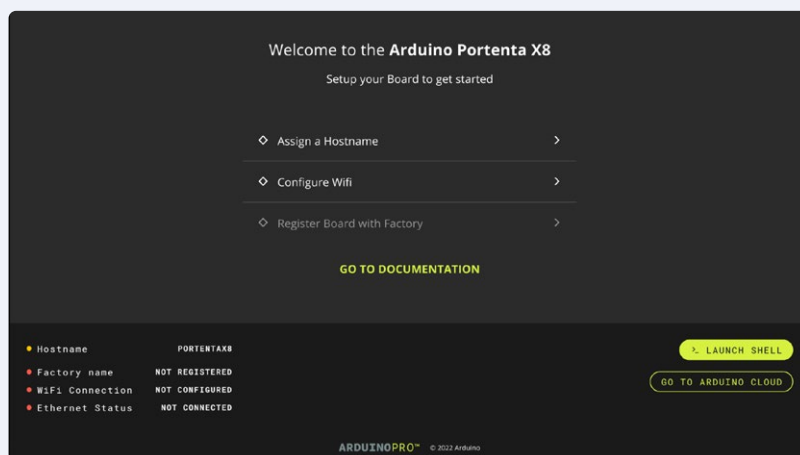
Containers

Zoals eerder vermeld, vormen containers een belangrijk onderdeel van de functionaliteit van de X8. Een container is een standaard software-eenheid die code en al zijn afhankelijkheden verpakt, zodat de app snel en betrouwbaar van het ene device naar het andere kan worden uitgevoerd. Een Docker container-image is een lichtgewicht, op zichzelf staand, uitvoerbaar softwarepakket dat alles bevat wat nodig is om een app uit te voeren. Met het geïntegreerde gebruik van Docker op de Portenta X8, is het eenvoudig

The screenshot shows the Foundries.io dashboard for a factory named 'arduino'. It displays summary statistics: 38 devices, 6 tags, and 40 members. Below this is a table of tags with columns for Name, Latest Target, Devices, Orphaned, Online, and On Latest. The 'Info' section shows the factory name 'arduino' and ID 'LA1K1G0aWes7', created on Aug 21, 2020.

Factories → arduino			FOUNDRIES.IO		RELEASES	DOCUMENTATION
Overview	Targets	Devices	Waves	Members	Teams	Source
DEVICES		TAGS		MEMBERS		
38		6		40		
Tags						
NAME	LATEST TARGET	DEVICES	ORPHANED	ONLINE	ON LATEST	
+ devel	274	1	0	0	0	
+ experimental	367	19	0	5	4	
+ next	265	3	0	1	1	
+ unknown	367	2	1	0	0	
+ experimental	0	2	2	1	0	
+ master	0	4	4	0	0	
Info						
NAME	CREATED					
arduino	Aug 21, 2020					
ID	LA1K1G0aWes7					

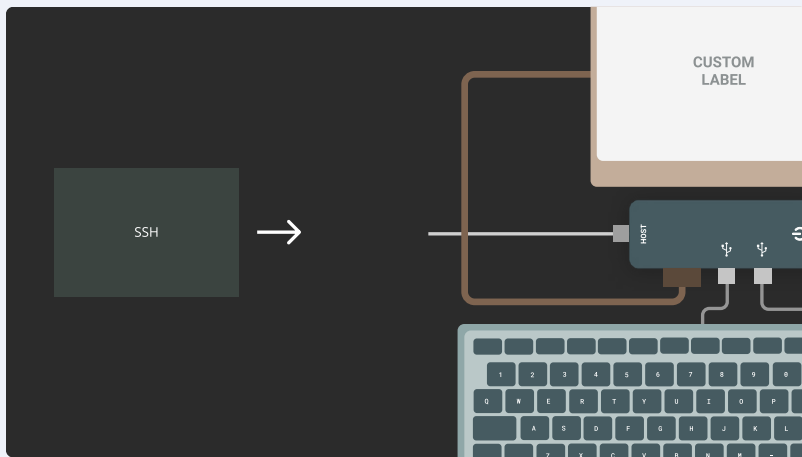
Figuur 2. De opbouw van het systeem.



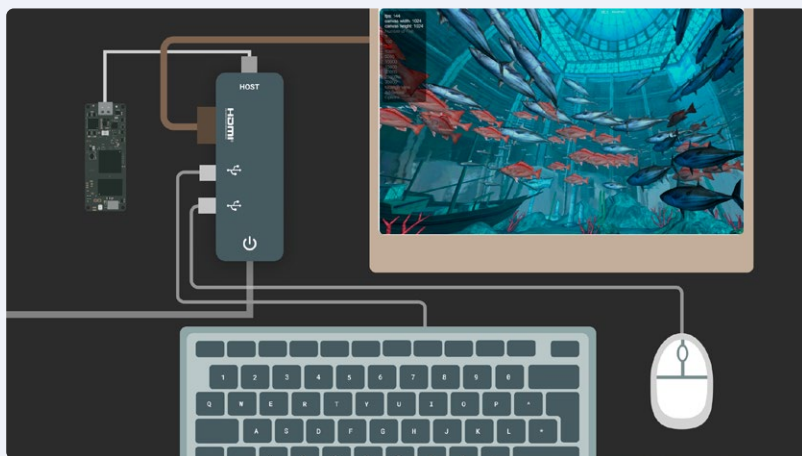
Figuur 3. Portenta X8 configuratie-webpagina.

This screenshot is identical to Figure 2, showing the Foundries.io dashboard for the 'arduino' factory with 38 devices, 6 tags, and 40 members.

Figuur 4. Overzichtspagina van FoundriesFactory (bron: Foundries.io).



Figuur 5. Voorbeeld van de aansluiting van de Portenta X8.



Figuur 6. Demo op Portenta X8.

om al bestaande containers te downloaden of om je eigen, zeer aanpasbare container te maken. Door een site zoals Docker Hub [3] te gebruiken, die een overvloed aan kant-en-klare containers bevat, kunnen we de Portenta X8 op veel verschillende gebieden inzetten. Wanneer een container uit een image wordt gecreëerd, moet dat image het volledige bestandssysteem van de container bevatten. Dit betekent dat het alles moet bevatten wat nodig is om een applicatie uit te voeren, alle afhankelijkheden, configuraties, scripts, binaire bestanden, enzovoort. Het image bevat ook andere configuraties voor de container, zoals omgevingsvariabelen, een standaardopdracht om uit te voeren en andere metadata. Een container is echter een enkele app, terwijl je als je meerdere containerapplicaties wilt gebruiken, zoals bij de X8, docker-compose

gebruikt dat ons helpt deze te beheren. De bestandsstructuur van een docker-compose-app kan er ongeveer zo uitzien:

- > **docker-build.conf** bevat de minimale 'unit test'-opdracht die op de container moet worden uitgevoerd om te bewijzen dat deze werkt.
- > **docker-compose.yml** definieert de naam van de app via de Factory, zijn machtigingen en instellingen voor de betrokken containers.
- > **Dockerfile** is een bestand dat een Docker image bouwt. Het image bevat alle afhankelijkheden die de Python-app nodig heeft.
- > **requirements.txt** bevat de Python-afhankelijkheden die je app zal kopiëren en downloaden.

- > de **src**-directory bevat de broncode van de app die je in de container wilt uitvoeren, of een opstartscript.

Dankzij Foundries.io wordt deze werkwijze eenvoudiger, omdat er een container build-script draait op de CI build-server die gekoppeld is aan de FoundriesFactory, die geactiveerd wordt door een push naar de Factory-repository. Hiermee kunnen we een nieuwe docker-compose app maken, deze toevoegen aan de repository en vastleggen of pushen. Na een moment zal de pagina van de Factory laten zien dat het container build-script is geactiveerd en het nu de nieuwe docker-compose app bouwt. Dit alles helpt ons een embedded Linux-platform te verkrijgen waarop een aangepaste Yocto-distributie draait die veilig is en bestand is tegen strenge cyberbeveiligings-checks, en waar de gebruiker kan doen wat hij in gedachten heeft door apps en aangepaste code te implementeren met behulp van containers.

RPC-functie

De container-infrastructuur van Arduino stelt ons in staat om realtime-verwerking uit te voeren aan de Arduino-kant terwijl we aan de Linux-kant een volwaardig besturingssysteem draaien. Dankzij de architectuur van de X8 kan de microcontroller bepaalde periferie afhandelen en de vereiste gegevens uitwisselen tussen de microcontroller en een Python-applicatie (het is ook mogelijk om elke programmeer- of scripttaal voor de applicatie te gebruiken) die aan de Linux-kant draait. Het communicatiemechanisme dat hiervoor wordt gebruikt, wordt RPC (Remote Procedure Call) genoemd. Met behulp van de docker-infrastructuur is het mogelijk een python-applicatie te maken die het RPC-mechanisme gebruikt om data uit te wisselen tussen de microcontroller en de iMX8, waarop het Linux-besturingssysteem draait. Als je geïnteresseerd bent om dieper op deze functie in te gaan, bekijk dan de tutorial *Data Exchange Between Python on Linux and an Arduino Sketch* [4].

Portenta X8 in actie

Laten we eens kijken hoe het proces van het gebruik van de Portenta X8 met FoundriesFactory en containers eruitziet met een korte demo. De NXP i.MX 8M Miniprocessor van de Arduino Portenta X8 kan worden gebruikt voor 3D rendering. Hiermee kunnen we 3D-content weergeven op een scherm of een video-uitgang. Het device gebruikt OpenGL om de 3D-gerelateerde berekeningen te verwerken. In deze demo zullen we webinhoud van het

internet renderen met WebGL en weergeven op een scherm via een USB-hub met een HDMI-uitgang. Er zijn veel manieren om met het board te communiceren, draadloos of bekabeld, en vervolgens ADB of SSH te gebruiken om met de X8 te communiceren via een commandoregel-interface. Omdat de Portenta X8 een Linux-device is, kun je normale Linux-commando's gebruiken om bestanden aan te maken, mappen te wijzigen, enzovoort.

Na het doorlopen van het installatieproces wordt de nieuwste firmware voor de X8 geïnstalleerd. Dit zal ook de rest van het proces veel gemakkelijker maken. Nu het board gekoppeld is aan een FoundriesFactory, kunnen we eenvoudig de nodige containers downloaden naar de Portenta X8 en deze met slechts enkele regels uitvoeren via SSH of ADB. Met FoundriesFactory kunnen we eenvoudig een target met dit voorbeeld uploaden naar een gekoppeld device. Als een lokale methode de voorkeur heeft, dan is het net zo eenvoudig om de repository te downloaden, naar het device te pushen met de ssh-of adb-shell, en het met een paar commando's te installeren en uit te voeren. Als alles is gedownload en draait, kan de Portenta X8 via een USB-hub op een scherm worden aangesloten (**figuur 5**). Het Aquarium 3D-voorbeeld van WebGL zal nu afgespeeld worden (**figuur 6**). Een muis kan ook worden aangesloten op de USB-hub om te interageren met de demo op het scherm. Op deze manier kunnen de 3D renderingmogelijkheden van de X8 met WebGL worden getest, waardoor het mogelijk wordt om verschillende parameters te testen of andere relevante informatie op te vragen. Een andere optie is om de container aan te passen met een Linux-tekst-editor om de URL te veranderen om iets anders op het scherm te laten zien.

Als je dit proces wilt herhalen of in meer detail wilt onderzoeken, bekijk dan onze tutorial *Output WebGL Content on a Screen* [5]. Op de Arduino docs-site kunt je ook meer tutorials vinden die de mogelijkheden van de X8 en het gebruik van docker containers laten zien. Als je wilt zien hoe je een database-container op de Portenta X8 kunt draaien die een WordPress-site host, ga dan naar onze *Running Wordpress and Database Containers on the Portenta X8* tutorial [6]. Misschien ben je meer geïnteresseerd in hoe de Portenta X8 een oplossing kan bieden voor MQTT-data-logging? Kijk dan eens naar de *Data logging met MQTT, Node-RED, InfluxDB en Grafana* tutorial [7]. Als je de mogelijkheden van de X8 nog meer wilt uitbreiden, plug hem dan in een Arduino Portenta Max Carrierboard. Dan kun je bijvoorbeeld een multi-protocol gateway bouwen. Bekijk daarvoor onze *Build a Multi-Protocol Gateway with Portenta X8 & Max Carrier* tutorial [8].

Nu je een idee hebt gekregen van wat de Arduino Portenta X8 kan, wat zijn kernbegrippen zijn en voor welke toepassingen hij kan worden gebruikt, is het aan jou om hem in actie te zetten en er het maximale uit te halen. 

220546-03 (vertaling: Willem den Hollander)

Over de auteur



Benjamin Dannegård is een Interaction Designer uit Malmö (Zweden) die geïnteresseerd is in alles wat met techniek te maken heeft en nerdy is.

Hij werkt al twee jaar bij Arduino als maker en Pro content creator.



Gerelateerde producten

Op zoek naar de belangrijkste producten die in dit artikel worden genoemd? Arduino en Elektor hebben ze voor je klaargezet!

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

WEBLINKS

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