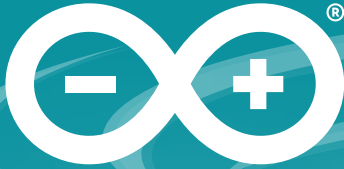


Speciale editie
gastredactie door



ARDUINO

Declassified
Bonuseditie!

Home Automation

Eenvoudige connectiviteit

p. 4 **Retro-Gaming**
met Arduino

p. 14 **Een controller**
voor Spotify

Van prototype tot productie

Elke week
nieuwe gratis
bonusartikelen!



Elektor's Arduino-
trainingskaart

p. 10



De menselijke factor
bij productie

p. 24



Maak kin...ren

p. 32

Van **Arduino** en **Elektor** met **Engineering Love**

NU VERKRIJGBAAR

Een unieke editie van Elektor Magazine met gastredactie door Arduino!

DIY elektronica projecten, tutorials en meer van Arduino en Elektor engineers

Vol met projecten, tips en tutorials

Duik in populaire onderwerpen als **MicroPython**, **TinyML** en home automation met Arduino

Leer het Arduino team kennen: Inzichten van **Fabio**, **Massimo** en **David**



Aan de slag met de **Portenta X8**

Koop uw exemplaar in uw favoriete kiosk of bestel het online in de Elektor Store!

Directe links in artikelen geven u gemakkelijk toegang tot Arduino producten en oplossingen

Meer informatie!

www.elektor.nl/arduino-magazine



Bonusinhoud voor de Arduino-editie met gastredactie van Elektor Mag december 2022.

Elektor is een uitgave van
Elektor International Media B.V.
Postbus 11, 6114 ZG Susteren, Nederland
Tel.: +31 (0)46- 4389444

Nieuwe abonnementen & bestellingen
service@elektor.nl – Tel. 046-4389444

Elektor International Media B.V. legt gegevens vast voor de uitvoering van de (abonnements) overeenkomst. De door u verstrekte gegevens kunnen gebruikt worden om u te informeren over relevante diensten en producten. Stelt u daar geen prijs op, dan kunt u dit schriftelijk doorgeven aan:

Elektor International Media B.V.,
Afdeling Customer Service
Postbus 11, 6114 ZG Susteren.
Of per email: service@elektor.nl

In overeenstemming met de Wet bescherming persoonsgegevens zijn de verwerkingen van persoonsgegevens aangemeld bij de toezichthouder, Autoriteit Persoonsgegevens te Den Haag.

Druk: Senefelder Misset, Doetinchem
Distributie: Betapress, Gilze

Advertenties Benelux

Raoul Morreau: tel. +31 (0)6 4403 9907
E-mail: raoul.morreau@elektor.com

Advertentietarieven, nationaal en internationaal, op aanvraag. Alle advertentiecontracten worden afgesloten conform de Regelen voor het Advertentiewezen gedeponereerd bij de rechtbanken in Nederland. Een exemplaar van de Regelen voor het Advertentiewezen is op aanvraag kosteloos verkrijgbaar.

Elektor gebruikt in haar publicaties uitsluitend eigen content (tekst en beeld) of met toestemming van de maker. Door derden aangeleverde content wordt voor publicatie altijd gecontroleerd op copyright. In geval de auteursrechthebbende onbekend is, doen wij ons uiterste best deze te traceren en marktconform te vergoeden. Helaas is het niet altijd mogelijk de uiteindelijke auteursrechthebbende te traceren. Stuit u hierop en bent of kent u de 'onbekende auteursrechthebbende', neem dan contact op met redactie@elektor.nl (.de/.fr/.com).

Auteursrecht

Niets uit deze uitgave mag vervaelvoudigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm of op welke wijze dan ook, zonder voorafgaande schriftelijke toestemming van de uitgever. De auteursrechtelijke bescherming van Elektor strekt zich mede uit tot de illustraties met inbegrip van de printed circuits, evenals de ontwerpen daarvoor. In verband met artikel 30 van de Rijsoctrooiwet mogen de in Elektor opgenomen schakelingen slechts voor particuliere of wetenschappelijke doeleinden vervaardigd worden en niet in of voor een bedrijf. Het toepassen van de schakelingen geschiedt buiten de verantwoordelijkheid van de uitgever. De uitgever is niet verplicht ongevraagd ingezonden bijdragen, die hij niet voor publicatie aanvaardt, terug te zenden. Indien de uitgever een ingezonden bijdrage voor publicatie aanvaardt, is hij gerechtigd deze op zijn kosten te (doen) bewerken. De uitgever is tevens gerechtigd een bijdrage te (doen) vertalen en voor haar andere uitgaven en activiteiten te gebruiken tegen de daarvoor bij de uitgever gebruikelijke vergoeding.

© Elektor International Media B.V. - 2022

Declassified Bonuseditie

De creatieve samenwerking tussen Elektor en Arduino hield niet op bij de editie van *Elektor Mag* met gastredactie die we begin december publiceerden, we hebben meer projecten, technische inzichten en informatieve artikelen om u de komende maanden te blijven inspireren. In de loop van vier weken geven we inhoud uit deze editie vrij, totdat u de complete bonuseditie begin januari in handen hebt. Wat een geweldige start voor het nieuwe jaar!

Of u nu als professional werkt aan een nieuw industrieel product of als hobbyist een leuk weekendproject met Arduino zoekt, u zult deze

extra editie van *Elektor Mag* informatief en inspirerend vinden. We bieden u artikelen over een breed scala aan Arduino-onderwerpen en -projecten, zoals retro-games met Arduino, een educatief Elektor Arduino-board, en een draagbare Arduino-controller voor Spotify.

Als u bij het lezen in dit tijdschrift behoefte voelt om te reageren, aarzel dan niet om uw gedachten met ons te delen via elektormagazine.com, arduino.cc of de sociale media. We verheugen ons op uw feedback. Veel plezier!

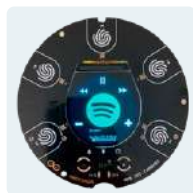
C. J. Abate (Content Director, Elektor)

IN DIT NUMMER



- ◀ **4 Doom spelen op een Portenta**
Retro-Gaming met Arduino

- 10 Speedcursus in de wereld van Arduino**
Uitbreidingskaart voor de Arduino Nano



- ◀ **14 Een controller voor Spotify**
De Oplà IoT-kit heeft (bijna) alles wat je nodig hebt

- 20 Bouw, installeer en onderhoud van schaalbare, veilige toepassingen**
Met Arduino Portenta X8 met NXP's i.MX 8M Mini Applications Processor en EdgeLock SE050 Secure Element



- ◀ **24 De menselijke factor bij productie**
Maak kennis met Daria Baradel, verantwoordelijk voor de productie bij Arduino

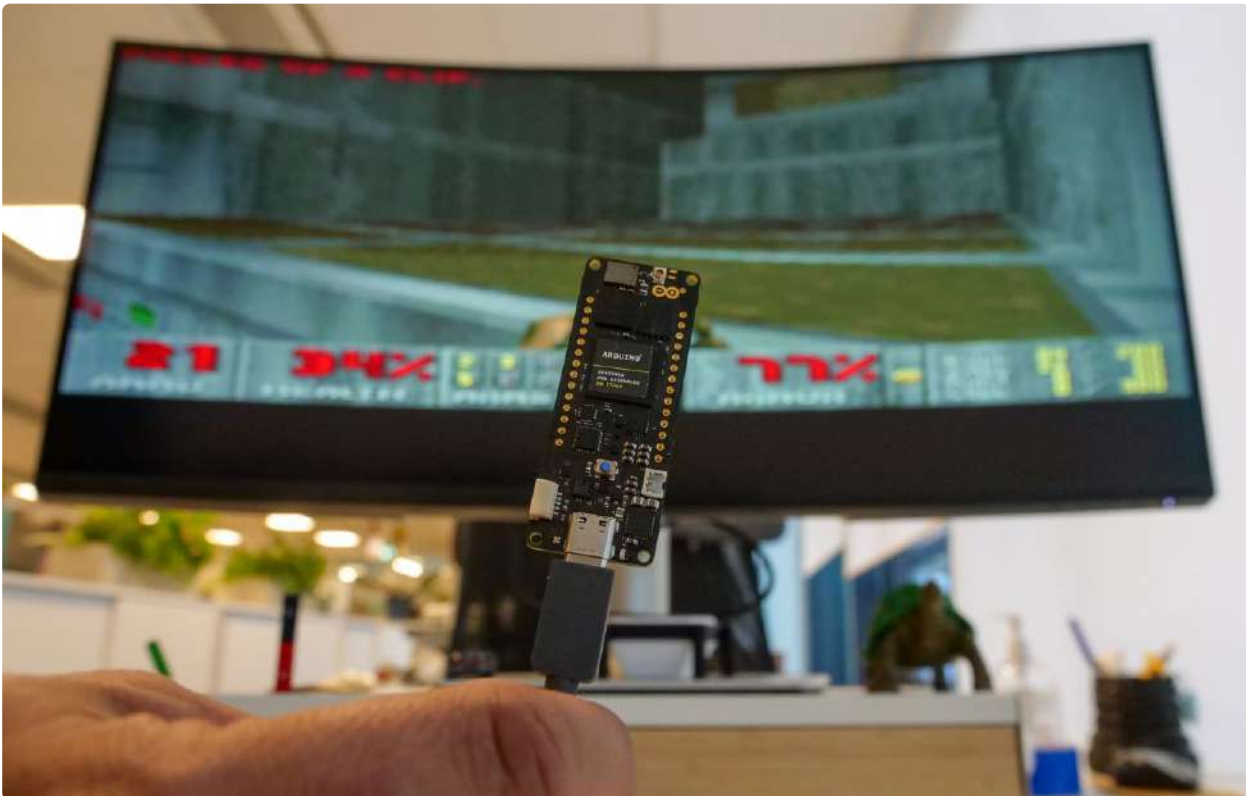


- 28 Ontwikkelborden**
Verleden, heden en toekomst

- ◀ **32 Bloed** **res**
ki **Beschikbaar in week 01** **uid**
2023

- ◀ **38 Pro** **s**
Nie **Beschikbaar in week 01**
2023





Je kunt Doom draaien op een Portenta!

Doom spelen op een Portenta

retrogaming met Arduino

David Cuartielles (Arduino)

Je kunt Doom spelen op een Arduino Portenta H7. Benieuwd hoe dat gaat? Wil je weten waarom de Arduino-technici het spel er in eerste instantie op lieten draaien? Martino Facchin, hoofd van Arduino's firmware-team, heeft de antwoorden.

Doom is waarschijnlijk het meest verkochte spel in de geschiedenis, met een verkoop van meer dan 3,5 miljoen exemplaren. Met een verkoopprijs van \$ 50 werden de ontwikkelaars van Doom, die een klein bedrijfje hadden met de naam id Software, van de ene op de andere dag miljonair. Toen Doom in 1993 uitkwam, was id Software al ongeveer een jaar bezig met een ander bekend spel, Wolfenstein 3D. Doom is een 'first-person shooter' waarin de speler door een 3D-ruimte moet navigeren en het moet opnemen tegen verschil-

lende vijanden met verschillende wapens en munitie die ook overal op het slagveld van het spel te vinden zijn. Doom is geport naar alle besturingssystemen, maar draait ook 'bare metal' op meerdere systemen. De broncode is open, momenteel gelicenseerd onder GPL. Hoewel het niet triviaal is om het te compileren, hebben we versies van Doom zien draaien op zeer kleine computers en ook binnen andere programma's. Op een gegeven moment maakte Microsoft Office's Excel 95 een hommage aan Doom in de vorm van een speelbare paasei, te vinden op de credits-pagina van de software.

Het spel is een manier geworden om zowel de prestaties van kleine computers te testen als om met hackerskunsten te pronken. Een paar maanden geleden, tijdens DEF CON 22 in Las Vegas, demonstreerde de hacker die op Twitter en GitHub bekend staat als @sickcodes Doom op een John Deere-tractor die was aangepast om landbouwgerelateerde afbeeldingen weer te geven. Arduino is geen uitzondering. En toen het eerste prototype van de Portenta H7 – ons krachtigste board tot dan toe in 2018 – uitkwam, gebruikten we Doom om de technische mogelijkheden van het board te testen. Onlangs nodigde ik Martino Facchin, hoofd van Arduino's firmware-team, uit om ons hier meer te vertellen en hoe het in zijn werk ging.



David Cuartielles: Praten we over de Arduino Portenta H7 waarop Doom draait. Ik heb een kleine samenvatting gemaakt van de geschiedenis van Doom, hoe de jongens van id Software het spel maakten, hoe het als een gek verkocht werd, en hoe de makers miljonair werden.

Martino Facchin: En toen publiceerden ze de broncode - het belangrijkste.

Cuartielles: Precies! Ze hebben de broncode gepubliceerd, maar onder welke licentie hebben ze dat gedaan?

Facchin: Ik moet het controleren, maar ik denk dat het een licentie is die compatibel is met de GPL, de Doom Source Code Licence. Het belangrijkste is dat alleen de engine open source is. De assets zijn gesloten, en in feite kun je alleen de shareware versie van Doom spelen, dus je kunt niet echt het volledige spel spelen, tenzij je het gekocht hebt.

Een opmerking van David

Ik ben dit gaan uitzoeken omdat ik er zeker van wilde zijn. In 1997 publiceerde id Software Doom onder bovengenoemde licentie, die de bron openstelde voor educatieve doeleinden. Na een voorval met de beheerders van glDoom, waardoor de wereld zonder kopie van de OpenGL port Doom kwam te zitten vanwege de non-distributie clausule in de Doom-licentie, stemde id Software ermee in de licentie te wijzigen in GPL.

Cuartielles: Het leuke van dit spel is dat mensen hun eigen mods maakten. Ik herinner me een versie van Wolfenstein 3D met personages uit Star Wars.

Facchin: Dat kan ik me niet herinneren.

Cuartielles: Zo oud ben ik. Ik herinner me voor ons beiden.

Facchin: Ik herinner me dat ik als kind Wolfenstein 3D speelde. Maar we hadden geen internetverbinding, dus konden we niet al die goodies van de modders krijgen.

Cuartielles: Voordat we verder gaan, wil ik je eerst iets vragen. Zou je jezelf even willen voorstellen?

Facchin: [lacht] Martino Facchin, firmware engineer bij Arduino.

Cuartielles: Wat is jouw rol bij Arduino?

Facchin: Ik ben hoofd firmware - de belangrijkste man als je hulp nodig hebt bij firmware. Nu heb ik een team met geweldige collega's die me daarbij helpen; destijds ben ik dit team alleen begonnen. Het team blijft groeien. We proberen ook de gemeenschap mee te

laten groeien door iedereen bewuster te maken van wat we doen.

Cuartielles: Wat is - naast het draaien van Doom op een Portenta (iets waar we het later nog over zullen hebben) - waar je bij Arduino het meest trots op bent?

Facchin: Volgens mij is dat het het Pluggable USB-framework. Toen ik net een half jaar bij Arduino was, hadden we een groot probleem met mensen die meerdere functies wilden toevoegen aan de USB-poort van de Arduino Due en de - toen nog - aanstaande Arduino Zero. Telkens als mensen een Arduino Leonardo in een computer prikten, werden de stuurprogramma's voor het toetsenbord, de muis enzovoort opgeroepen, zelfs als je die niet gebruikte. We moesten dit on the fly bouwen, een USB-descriptor die de gebruikers zou helpen alleen die dingen te zien die ze op dat moment wilden gebruiken. Tegelijkertijd lieten we ook andere dingen toe, dingen die mensen wilden gebruiken zoals USB MIDI en dergelijke. Voor mij was dit een enorme ontwikkeling. Ik was vrij jong in die tijd en moest met de hulp van Matthijs Kooijman (lees meer over zijn werk op www.stder.nl) en Paul Stoffregen (de bedenker van Teensy) in contact komen met de gemeenschap om de best mogelijke strategie uit te zoeken. En het werkte. Zo nu en dan komen mensen zeggen: "Ik heb de MIDI-bibliotheek gebruikt voor dit of dat," of er is zelfs een ontwikkelaar die nu alle layouts voor alle internationale toetsenborden maakt, gebouwd bovenop de fundamenteën van die code. Ik ben daar erg trots op.

Cuartielles: Je zei dat je 'vrij jong was in die tijd'. Hoe lang werk je al bij Arduino?

Facchin: Zes en een half jaar, nu. Vanuit het kantoor in Turijn.

Cuartielles: Dat is best wel lang. En hoeveel mensen zijn betrokken bij het firmware-team?

Facchin: Zes mensen. Dat lijkt misschien veel, maar bij het firmware-team onderhouden we alle producten, terwijl we ook andere activiteiten zoals certificering uitvoeren. Aan de andere kant hebben we Firmware gescheiden van Tooling, wat een ander team is dat zich bezig houdt met de Arduino CLI en andere delen van de hogere software.



De voor de hand liggende keuze was om Doom te porten en te proberen alle functies te instrumenteren die nodig waren om het te laten werken.



Cuarterlles: En alle ontwikkelaars werken tegen Github, klopt dat?

Facchin: Ja, aan het eind van het ontwikkelingsproces is alles open source.

Cuarterlles: Geweldig. Laten we het nu weer over Doom hebben. We weten dat het een zeer succesvol spel was waarvan meer dan 3,5 miljoen exemplaren verkocht zijn en dat de ontwikkelaars ervan stinkend rijk maakte. Het was het eerste bestseller first-person shooter (FPS) spel. De code was open source en werd overgezet naar allerlei apparaten.

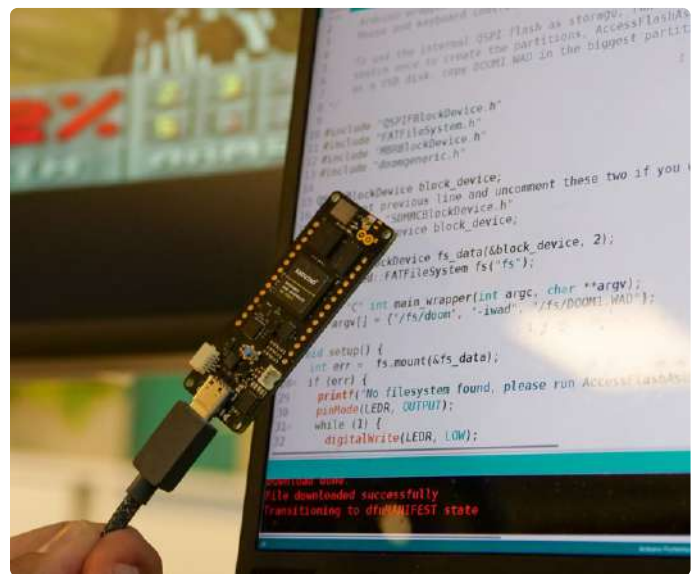
Facchin: Telefoons, rekenmachines, elk besturingssysteem.

Cuarterlles: En Arduino's Portenta H7, een dual core processorboard gericht op industriële toepassingen. Wie kwam op het idee om Doom op de Portenta H7 te laten draaien?

Facchin: Het was meer dat we net het eerste prototype van de Portenta H7 hadden binnengekregen, een mooi board met veel chips, en alles moest nog worden gedaan. We hadden deze chip van Analogix die je gewoonlijk in andere apparaten aantreft en die MIPI-signalen omzet in DisplayPort, waardoor hij externe monitoren zou kunnen aansturen. We hadden geen ervaring met de subsystemen in de chip die dit zouden moeten afhandelen, en de bestaande voorbeelden hielpen ook niet. Eerst lukte het ons om wat gele vakjes op het scherm weer te geven, daarna het Arduino logo met een paar artefacten, maar het was verre van perfect. We begrepen niet echt waarom de dingen niet werkten zoals verwacht, dus ging ik voor iets met bewegende beelden die ik gemakkelijk kon herkennen.

De voor de hand liggende keuze was om Doom te porten en te

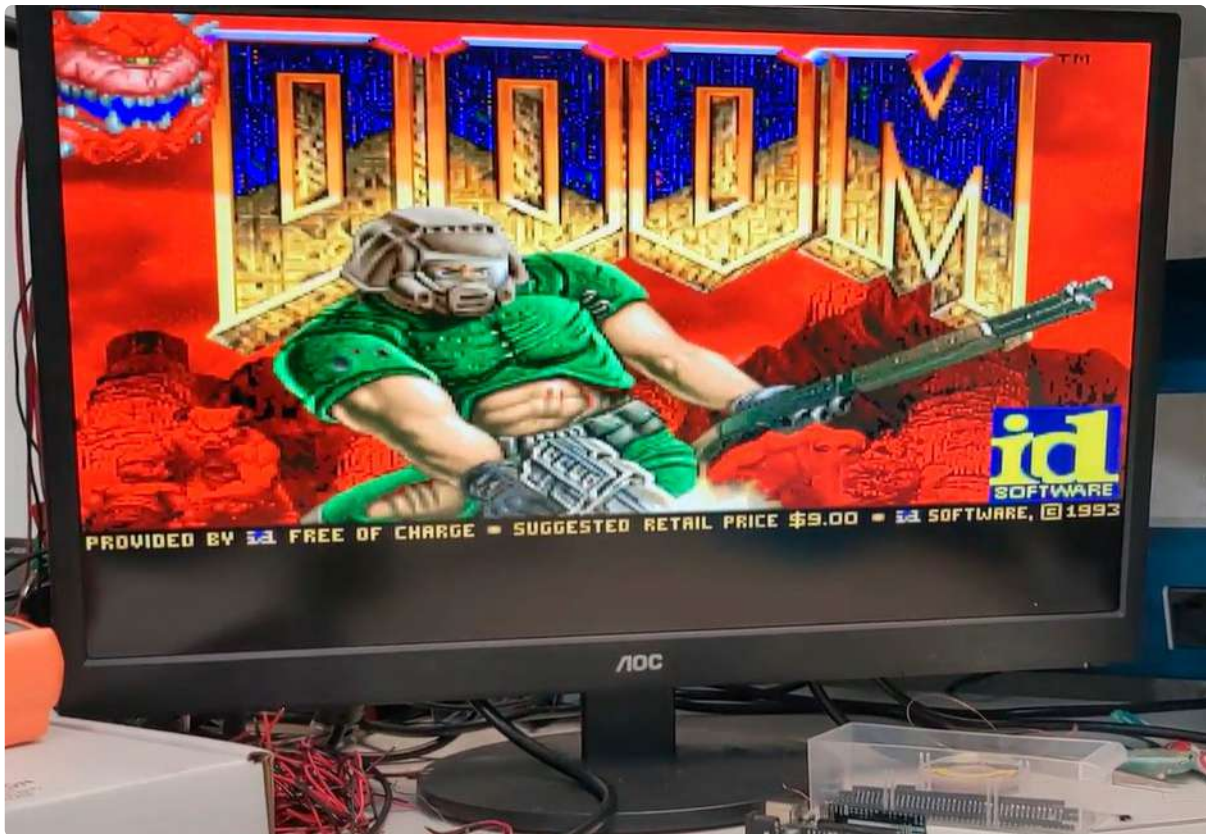
proberen alle functies te instrumenteren die nodig waren om het te laten werken. Het kostte weinig moeite om het te laten werken op iets zonder besturingssysteem. Het is niet echt de fork van Doom die je zou spelen, maar het is heel gemakkelijk te porten. Je hoeft maar zes of zeven functies te veranderen om het aan te passen aan je hardware en je bent klaar. Om te beginnen hadden we geen code om het interne RAM of het externe geheugen te laten werken. Die moest ik eerst aan de praat krijgen en daarna de simulator opstarten. Vanaf het begin kregen we video niet aan de praat, ik moest een framebuffer voorbereiden, wat toveren, en goede uitvoer over USB-C krijgen.



Het is aan de lezer om het project over te nemen en besturingselementen toe te voegen om het spel speelbaar te maken.

Hoe kun je Doom draaien op je Portenta H7?

1. Download de nieuwste versie van de Arduino IDE. Wij raden Arduino 2.0 of recenter aan.
2. Download de Portenta-core uit de board manager.
3. Selecteer de M7-core voor je Portenta H7-board. Alle volgende stappen moeten op die core worden uitgevoerd.
4. Zorg ervoor dat de IDE de poort heeft geïdentificeerd waarop het board is aangesloten.
5. Er is een voorbeeld bij "Examples/Doom" waar je de basisinstructies kunt zien. Voordat je dit installeert, moet je een paar voorbeelden op je board uitvoeren.
6. [Optioneel] Upgrade je Portenta H7 bootloader naar de laatste versie met met "Examples/STM32H747_System/STM32H747_manageBootloader".
7. Formateer de externe flash met het voorbeeld "Examples/STM32H747_System/QSPIFormat". Na de installatie moet je de seriële terminal openen en de daar gegeven instructies volgen.
8. Maak van je board een opslagmedium alsof het een USB-drive is met "Examples/USB As Mass Storage/AccessFlashAsUSBdisk".
9. Open de seriële monitor en kies hoe het formatteren van het board moet verlopen. Zodra dit is gebeurd, zou je computer twee nieuwe externe schijven moeten registreren die erop zijn aangesloten.
10. Download DOOM1.WAD van de DoomWiki site op: <https://doomwiki.org/wiki/DOOM1.WAD> en kopieer het in de grootste partitie van de virtuele Portenta-schijf.
11. Ga terug naar het Doom.ino voorbeeld dat we bij stap 5 zagen en flash het op je bord. Denk eraan: altijd op de M7-kern. Als je problemen had met het zien van de programmeerpoort, dubbelklik dan vóór het uploaden op de resetknop van je Portenta en controleer of de seriële poort goed wordt herkend.
12. Koppel de Portenta H7 los van je computer en sluit hem aan op een USB-C hub alsof het een laptop is. Op de hub zal de externe voeding aangesloten moeten zijn en een HDMI-kabel om het videosignaal naar een computermonitor te sturen.



Doom is een klassieker!

Cuartielles: De Portenta H7 is een dual-core processor. Hij heeft een Arm Cortex M4 en een M7 in de processor. Op welke draaide Doom?

Facchin: Tegenwoordig gebruiken we de M7, maar destijds draaiden we hem op de M4, omdat die veel eenvoudiger was vanuit het oogpunt van embedded programming. Het is meer een typische microcontroller, zonder bijzondere eigenschappen. Aan de andere kant heeft de M7 deze cache die je moet bekijken en op het juiste moment ongeldig moet maken bij het genereren van video. De eerste keer dat ik dingen probeerde te draaien op de M7 was hij echt te snel, en ik kon niets zien op het scherm. De M4 was snel genoeg (25 beelden per seconde) en pixelperfect.

Cuartielles: 25 fps is een stuk sneller dan mijn eerste computer. Maar laten we dit samenvatten voor de lezer: je had Doom draaien op de traagste van de twee processoren, op een board dat Bluetooth, WiFi als connectiviteit heeft. Video wordt verzonden via USB-C. Daar kun je een hub hebben, muis, toetsenbord, wat dan ook. Welke besturing had je daar geïmplementeerd?

Facchin: Helaas hield het project daar voor ons op, want toen we eenmaal zagen dat we video aan de praat konden krijgen, gingen we aan de slag met LVGL, wat een veel nuttigere bibliotheek is voor andere ontwikkelaars om toepassingen bovenop Portenta te bouwen. LVGL is volledig geïntegreerd met de USB-hub, toetsenbord en muis, zodat je alle benodigde interfaces kunt bouwen voor de professionele context die veel eindgebruikers van Arduino nodig hebben.

Cuartielles: Zou het niet cool zijn om een professionele PLC te hebben op basis van de Portenta H7 waarbij je Doom kunt spelen op de Cortex M4-core terwijl je het serieuze werk doet op de M7?

Facchin: Zeker weten!

Cuartielles: Bedankt, Martino. Het was geweldig om het verhaal van Doom op Arduino Portenta H7 te horen. We zullen met onze community de basisinstructies delen over hoe je het aan de praat krijgt. Het is aan de lezer om het project over te nemen en besturingselementen toe te voegen om het spel speelbaar te maken. ◀

220542-03 (vertaling: Hans Adams)

Over de auteur

David Cuartielles is mede-oprichter van Arduino. Hij heeft een PhD in Interaction Design en een MSc in Telecommunications Engineering, en hij doceert aan de Universiteit van Malmö.

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de redactie van Elektor via redactie@elektor.com.



Gerelateerde producten

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

20%

korting
op het eerste jaar van
uw lidmaatschap

Word lid van de Elektor Community

Neem een



of



lidmaatschap!



Gold Green

- ✓ ✓ Een compleet web-archief t/m 1980!
- ✓ 8x Elektor Magazine (Print)
- ✓ ✓ 8x Elektor digitaal (PDF)
- ✓ ✓ 10% korting in onze webshop, en exclusieve aanbiedingen
- ✓ ✓ Toegang tot meer dan 5000 Gerberfiles
- ✓ ✓ Gratis bezorging binnen Nederland en België



www.elektormagazine.nl/arduino-member

Gebruik kortingscode

ARDUINO22



Uitgepakt: Elektor's LCR METER met David Cuartielles

Noteer de datum: 26 januari 2023



Ben je geïnteresseerd in de Elektor LCR Meter Kit en wil je een blik onder de motorkap werpen? Kijk dan op 26 januari 2023 (18:00 CET) naar Elektor Lab Talk, waar ik samen met de Elektor-engineers Mathias Claussen en Jens Nickel de LCR Meter Kit zal bespreken, en je vragen over de Arduino-technologie en dit nummer van Elektor (waar wij als gastredactie aan meewerkten) beantwoord. Mis de livestream niet. En bedenk alvast je vragen. ◀

220555-03 (vertaling: Evelien Snel)

Elektor LabTalk

Elektor Lab Talk,
26 januari 2023 -
livestream met David!



[www.elektormagazine.com/
labtalk-david](http://www.elektormagazine.com/labtalk-david)

Stap in de wereld van Arduino

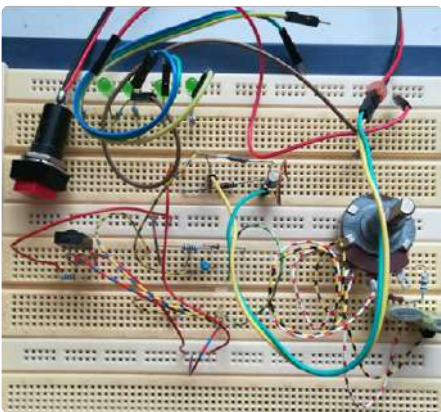
Uitbreidingsboard voor de Arduino Nano

Wolfgang Trampert (Duitsland)

Elektor blijft trouw aan zijn educatieve missie: hier presenteren we een gloednieuw trainingsboard met een Arduino Nano in het middelpunt. Samen met een goed opgebouwde, hands-on training biedt het een ideaal platform om je vaardigheden te verbeteren en de wereld van microcontrollers te verkennen.

Noot van de redactie

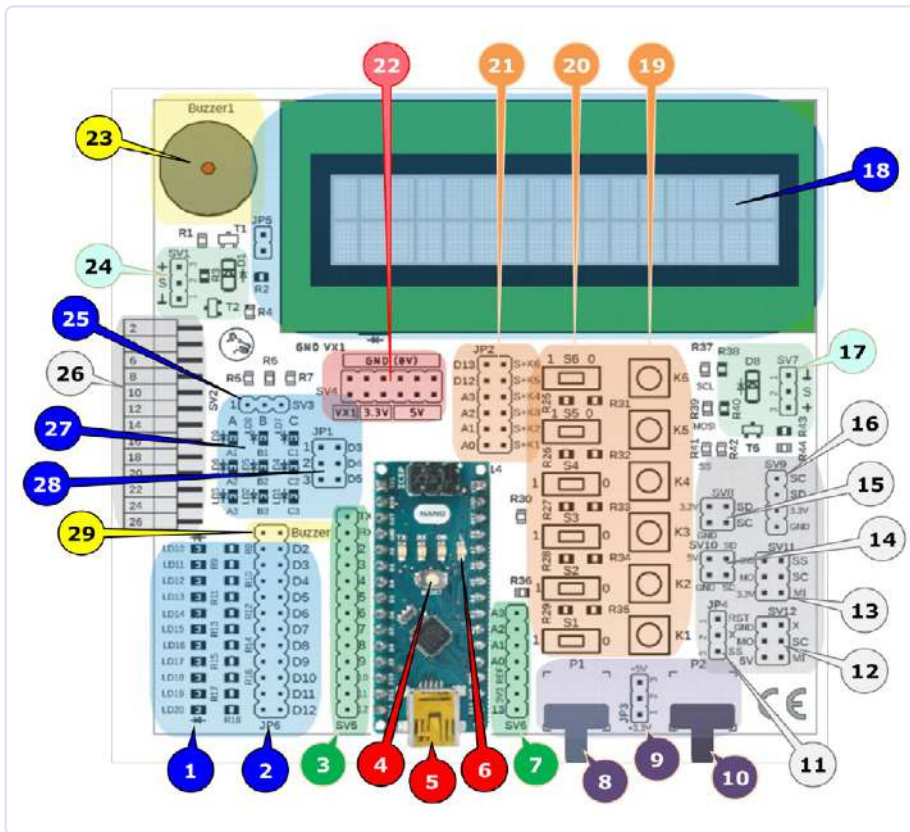
Op het moment van publicatie is het boek bij deze kit alleen in het Duits beschikbaar. Voor de nabije toekomst zijn vertalingen gepland. Zodra deze klaar zijn, zal de vertaalde versie beschikbaar zijn in de Elektor Store



Figuur 1: Typisch steekboard voor een Arduino sketch.

Je zou kunnen zeggen dat de "Arduino-filosofie" wordt geassocieerd met een dicht-op-de-hardware ontwerpbenadering: in de meeste gevallen heeft de Arduino-software of -sketch toegang tot componenten zoals schakelaars, drukknoppen, potentiometers, LED's, LCD-displays, piezozoemers, drivertransistors enz. via de GPIO's van de microcontroller. Andere soorten elektronische modules zoals sensoren, displays of driver boards gebruiken verschillende seriële interfaces zoals SPI, I²C of 1-draads bus om te communiceren met de controller. Om vertrouwd te raken met de wereld van microcontrollers en Arduino-boards, moet je oefenschakelingen bouwen en deze besturen met een Arduino-board.

De essentie van elk project met Arduino-hardware is echter de ontwikkeling van de software (sketch). De hardware is slechts een middel. Je kunt natuurlijk alle periferie die je voor een bepaald project nodig hebt aansluiten met behulp van een bread board (figuur 1). Deze aanpak geeft je maximale vrijheid om onderdelen te plaatsen en signalen aan te sluiten waar je maar wilt. Dat is niet altijd een goed idee, vooral niet als je net begint. De opbouw kan snel in omvang toenemen, zodat je eindigt met een spaghetti van draden. Vaak ben je meer tijd kwijt aan het debuggen van de hardware en het oplossen van bedradingsfouten dan aan het schrijven van code. Dit verhoogt alleen maar de frustratie, en van het corrigeren van domme fouten leer je niet altijd iets nuttigs.



Figuur 2: Het MCCAB-training board, Rev. 3.3.

Het MCCAB Training Board

Om deze hindernissen te omzeilen, hebben we het Elektor Arduino-trainingsboard ontwikkeld, ook bekend als het *MCCAB-training board* (figuur 2). De kern is een Arduino Nano-board dat op het MCCAB-training board wordt geprikt. Daarnaast zijn er de voornaamste zaken die je over het algemeen nodig hebt om een nieuw prototype te bouwen voor vele toepassingen zoals een lab set-up, test en experimentele schakelingen, projecten en oefeningen ter ondersteuning van je studie net als opleiding en ook hobby projecten. De microcontroller GPIO's zijn allemaal beschikbaar op twee pin header strips op het *MCCAB-training board* wat het board maximale mogelijkheden biedt. Extra onderdelen of externe signalen kunnen naar wens worden aangesloten met behulp van verbindingsdraden. Je hoeft je geen zorgen te maken over verkeerde bedrading van de aanwezige periferie en je hoeft minder tijd te besteden aan het doorzoeken van dozen met reserveonderdelen om dat ongrijpbare onderdeel te vinden dat je nodig hebt om een schakeling te completeren.

Extra schakelingen die op breadboards zijn gebouwd, kunnen ook gemakkelijk worden aangesloten met verbindingsdraden, omdat alle GPIO's van de microcontroller op de

Arduino Nano zijn verbonden met de twee headerstrips SV5 en SV6 op het MCCAB-training board (pijlen 3 en 7 in figuur 2). Langs de linkerrand van de print zie je ook de dubbele rij 26-pins rechthoekige header connector SV2 (pijl 26 in figuur 2) waarop een externe uitbreidingsprint kan worden aangesloten. Deze connector levert alle belangrijke GPIO-signalen van de microcontroller. Externe printen om functies te realiseren zoals een elektronische curve-tracer, een lab voeding of een verkeerslichtcontroller kunnen hier op het MCCAB-Trainingsboard worden aangesloten en door het board worden aangestuurd. Informatie en resultaten van een lopende sketch kunnen via de I²C bus naar de 2 x 16 karakters LCD (pijl 18) worden gestuurd. Er is ook een 3 x 3 LED-matrix aanwezig (pijl 27).

Het MCCAB-Training board wordt gevoed met een spanning van $V_{cc} = +5\text{ V}$. Deze wordt gewoonlijk geleverd door de USB-kabel die op je PC is aangesloten en die je nodig hebt voor het maken en uploaden van de sketch naar de MCCAB. De MCCAB kan ook gevoed worden door een externe voeding.

In figuur 2 worden alle onderdelen die verband houden met een specifieke functie op de print geïdentificeerd met een gemeenschappelijke achtergrondkleur.

De bedien- en display-elementen op het MCCAB-Training Board

- 1 11 x LED (statusindicatie van in-/uitgangen D2 tot D12)
- 2 Connector die LED's LD10 tot LD20 verbindt met GPIO's D2 tot D12
- 3 Microcontroller in- en uitgangen
- 4 Resetknop
- 5 Arduino NANO met mini-USB-aansluiting
- 6 LED L, verbonden met GPIO D13
- 7 Microcontroller GPIO's
- 8 Potentiometer P1
- 9 Voedingsspanning naar P1 en P2
- 10 Potentiometer P2
- 11 Signaal op pin X van SV12
- 12 SPI-Interface 5 V (Het signaal op pin X wordt geselecteerd door JP4)
- 13 SPI-Interface 3.3 V
- 14 I²C-Interface 5 V
- 15 I²C-Interface 3.3 V
- 16 I²C-Interface 3.3 V
- 17 Schakeluitgang voor externe devices
- 18 2 x 16 karakter LCD
- 19 6 x drukknoppen K1 tot K6
- 20 6 x schuifschakelaars S1 tot S6
- 21 Pin header om schakelaars te koppelen aan de microcontroller GPIO's
- 22 Voedingsspanningsverdeler
- 23 *Buzzer1*
- 24 Schakeluitgang voor externe devices
- 25 3 x 3 LED-matrixkolommen
- 26 2 x 13-pins headerstrip om een externe module aan te sluiten
- 27 3 x 3 LED matrix (rood)
- 28 Aansluitingen van 3 x 3 matrixrijen op D3, D4 en D5
- 29 Jumper koppelt Buzzer1 aan GPIO D9

Tabel 1: Beschikbare classes in de MCCAB_Lib-bibliotheek.

Class	Usage
KeySwitch	Debounced status van schakelaars S1 tot S6 en drukknoppen K1 tot K6
Matrix	Aansturing van de 3 x 3 LED-matrix.
LED	Aan/uit/knipperbesturing van de 12 LED's LD10 t/m LD20 en LED
LedBlock	Een bitpatroon naar alle 11 LED's (LD10 tot LD2) sturen
Sound	Controle van Buzzer1 en blokvolggenerator.

De MCCAB_Lib-bibliotheek voor gebruik met het trainingsboard

Bij de software-ontwikkeling wordt de Arduino IDE gebruikt om het programma (of 'Sketch' in Arduino-taal) te schrijven dat de microcontroller vertelt hoe hij zich moet gedragen. De sketch wordt vervolgens gecompileerd en via een mini-USB-kabel geüpload naar de microcontroller van de Arduino Nano op het trainingsboard.

De GPIO's van de microcontroller kunnen zoals gebruikelijk worden geconfigureerd met de Arduino-functie `pinMode()` en de waarde van signalen naar en van componenten op het oefenboard kan worden gelezen of gecontroleerd met `digitalRead()`, `digitalWrite()`,

`analogRead()` enz.

Er is echter een bibliotheek genaamd *MCCAB_Lib* [1] beschikbaar die de ontwikkelaar ondersteunt met extra commando's om de devices op het MCCAB-Training board aan te sturen. De bibliotheek kan gratis worden gedownload en geïntegreerd in je eigen sketch. Deze bibliotheek maakt de bediening van de aanwezige periferie veel gemakkelijker.

De bibliotheek *MCCAB_Lib* bevat vijf classes voor het aansturen van de schakelaars, de LED's en de zoemer op het trainingsboard en kan naar behoefte gemakkelijk in de sketch van de gebruiker worden opgenomen. Tabel 1 bevat een lijst van de beschikbare classes. Met deze bibliotheek hoeft de gebruiker zich

geen zorgen te maken over het definiëren van tijdsperioden voor het debouncen van schakelaars, het genereren van multiplexbesturingssignalen voor de 3 x 3 LED-matrix en het knippen van de LEDs *LD10* tot *LD20* of *L*, of zelfs het genereren van de zoemer frequenties. De bibliotheekfuncties doen dit automatisch op de achtergrond, zonder dat de gebruiker het merkt.

Listing 1 is een kleine voorbeeldsketch om het gebruik van de *MCCAB_Lib*-bibliotheek te demonstreren.

Op regel van de sketch wordt de objectvariabele `Led` gedeclareerd van de class `LED` uit de bibliotheek *MCCAB_Lib*. De parameter `LED_PIN` die in de declaratie van de objectvariabele `Led` wordt doorgegeven, wordt in regel 13 gedefinieerd als een constante die de pin aangeeft waarop de LED is aangesloten. Deze pin wordt automatisch geconfigureerd als een uitgang tijdens de instantiëring.

De objectvariabele `Key` van class `KeySwitch` uit de *MCCAB_Lib*-bibliotheek (gedecclareerd in regel 22) bewaakt tijdens de uitvoering (op de achtergrond) de toestand van de schakelin-



Listing 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...
25 void loop() { } // or here
```

gang op pin SK4, die hem volgens zijn declaratie als parameter wordt doorgegeven. Ze voert de debouncing van de schakelaar uit wanneer de drukknop K4 wordt ingedrukt of losgelaten en roept de functie `switchTurnedOn()` op wanneer de knop wordt ingedrukt. De `toggle()` methode van class `LED` uit de `MCCAB_Lib` bibliotheek wordt geactiveerd in de functie `switchTurnedOn()` in regel 19 om de huidige toestand van de LD10 LED om te keren.

Omdat de aansluitpinnen voor de schakelaar en de LED automatisch worden geconfigureerd als in- en uitgang wanneer de objecten worden gedeclareerd, hoeft verder niets te worden gedaan in de `setup()` functie op regel 24 in deze sketch.

De `loop()` functie op regel 26 bevat ook geen instructies omdat de enige actie die in deze sketch moet worden uitgevoerd het schakelen van de LED-status is wanneer de K4-toets wordt ingedrukt. Deze actie wordt gestuurd door de `KeySwitch` class door de call van de functie `switchTurnedOn()`.

Door deze classes in de bibliotheek `MCCAB_Lib` te gebruiken in uitgebreidere sketches zijn de twee functies `setup()` en `loop()` van het standaard Arduino-softwaremodel niet nodig om voortdurend de toestand van perifere componenten af te vragen, waardoor ze vrij komen voor belangrijkere taken.

12 Project-sketches en 46 oefeningen

Een gedetailleerde handleiding voor het MCCAB-training board is beschikbaar en kan worden gedownload van de website. [1] Het MCCAB-training board en de `MCCAB_Lib` bibliotheek zullen ook in detail beschreven worden in een boek dat binnenkort beschikbaar zal zijn (zie noot van de uitgever).

Het boek bespreekt in detail de hardware- en softwarebasisprincipes van een microcontrollersysteem en introduceert de programmeertaal C/C++, die wordt gebruikt om Arduino-sketches te schrijven.

Het boek is vooral gericht op praktische oefeningen, zodat "leren door doen" het sleutelbegrip is dat hier wordt gebruikt om de vaardigheden te krijgen die je nodig hebt wanneer je je eigen projecten gaat bouwen. In een uitgebreid praktisch gedeelte zijn er 12 project-sketches en 46 oefeningen, zodat je kennis toeneemt naarmate je de vele voorbeelden doorwerkt. De oefeningen zijn zo opgebouwd dat de lezer een taak krijgt die met het MCCAB-training board moet worden opgelost met behulp van de kennis die is opgedaan in het theoriegedeelte van het boek. Voor elke oefening is er dan een gedetailleerde uitleg en goed toegelichte voorbeeldoplossingen die helpen bij het oplossen van problemen. ◀

Vragen of opmerkingen?

Als je technische vragen hebt, kun je de Elektor-redactie mailen op editor@elektor.com.

Over de auteur

Wolfgang Trampert ontwikkelt en programmeert microcontrollersystemen sinds hij zijn studie elektronica heeft afgerond. Zijn ingenieursbureau ontwikkelt op microcontrollers gebaseerde oplossingen om aan de eisen van de klant te voldoen. Hij is auteur van enkele gespecialiseerde boeken en artikelen en geeft cursussen over microcontrollers.



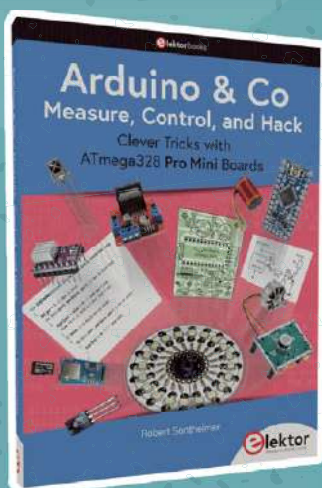
Verwante producten

- > **MCCAB Training Board (SKU 20295)**
www.elektor.nl/20295
- > **Mikrocontroller-Praxiskurs für Arduino-Einsteiger (boek in het Duits, SKU 20293)**
www.elektor.de/20293

WEBLINK

[1] The MCCAB_Lib Library:
<http://www.elektor.nl/20295>

220450-03



Arduino & Co – Measure, Control, and Hack

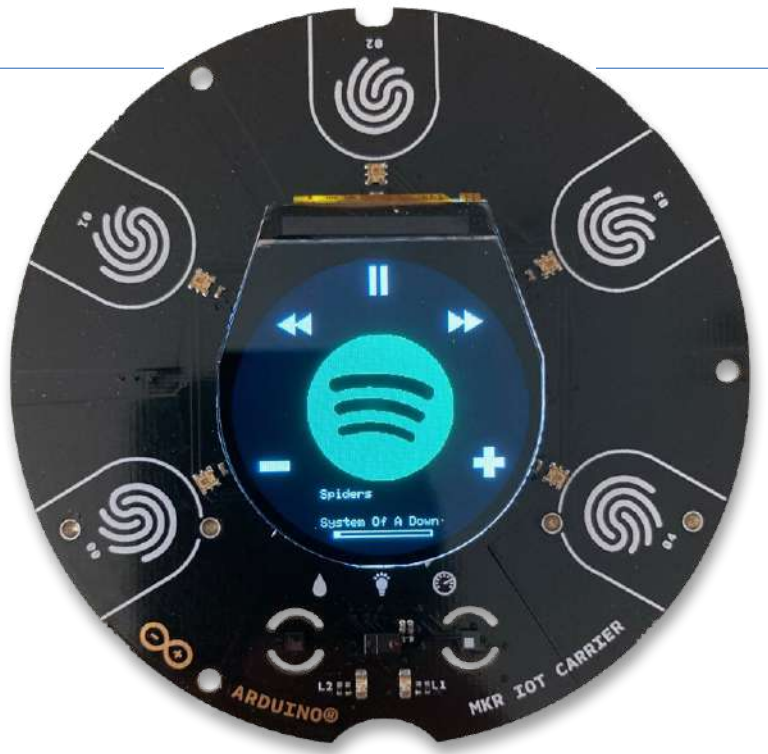
Met een eenvoudig Pro Mini board en een paar andere componenten worden projecten die 20 of 30 jaar geleden ondenkbaar waren (of een klein fortuin zouden hebben gekost) in dit boek eenvoudig en betaalbaar gerealiseerd: Van eenvoudige LED-effecten tot een volledig oplaad- en teststation dat een oplaadbare batterij op de proef stelt, er is voor elk wat wils.

www.elektor.nl/20243



Een controller VOOR Spotify

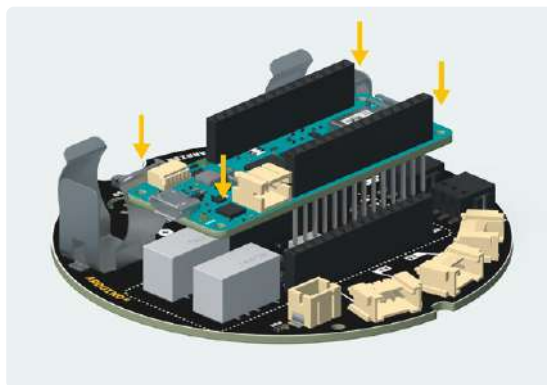
de Oplà IoT-kit heeft (bijna)
alles wat je nodig hebt



Altuğ Bakan (Turkije)

De Arduino Oplà IoT-kit bevat het MKR WiFi 1010 Maker Board en een carrier board met relais, een rond OLED-display, capacitieve aanraakknoppen en enkele sensoren. We beschrijven hier hoe je een draagbare controller voor de populaire Spotify-player kunt maken. Natuurlijk is ook een beveiliging nodig.

Het Arduino MKR WiFi 1010 Maker Board is, dankzij zijn WiFi-mogelijkheden, een perfect brein voor je volgende IoT-project. Je bent nog beter uitgerust met de Arduino Oplà IoT-kit, die naast dit Maker Board ook een carrier board bevat (figuur 1). Dat is voorzien van relais, een rond OLED-display en capacitieve aanraakknoppen. De kit



Figuur 1. Het MKR WiFi 1010 Maker Board wordt op het carrier board met relais en andere nuttige periferie geprikt.

bevat ook een vochtsensor en een PIR-sensor (figuur 2). Projecten zoals een inbraakalarm en automatisch planten water geven zijn dus gemakkelijk te implementeren. De WiFi-functie maakt het ook mogelijk om programma's op je PC te bedienen, als ze een netwerkinterface hebben. De aanraakknoppen, het display, de batterijhouder en een behuizing maken het gemakkelijk om een draagbare controller voor PC-software te maken, als aanvulling van de muis en het toetsenbord (figuur 3).

Ik ben een fan van de muzikspeler Spotify, dus ik heb de Oplà-kit gebruikt om mijn eigen draadloze Spotify-controller te bouwen. Er zijn knoppen om vooruit en achteruit door de songlijst te springen, te starten/pauzeren en om het volume te regelen. Het spreekt vanzelf dat dan de Spotify-player moet draaien op je PC of smartphone.

Veilige communicatie

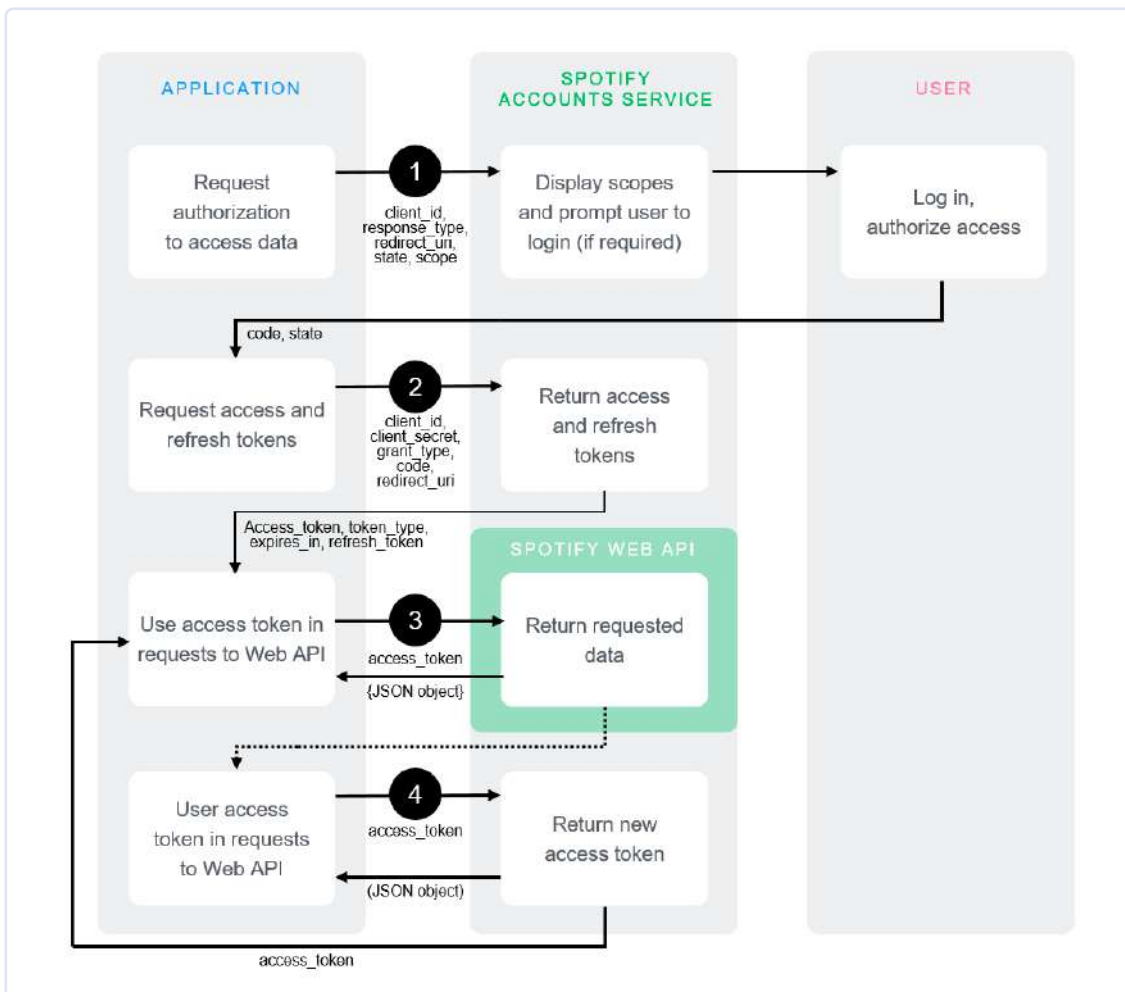
Spotify heeft een handige programmeerinterface om je Spotify-speler via het netwerk te besturen, maar die werkt alleen als je een Spotify Plus-licentie hebt. Natuurlijk is een beveiliging nodig. Om de Spotify web-API, die met REST werkt, te gebruiken moet je eerst inloggen bij de Spotify Accounts Server met je Spotify-gebruikersnaam en -wachtwoord. Na het inloggen moet je software een Client ID en een Client Secret sturen. De Spotify-server antwoordt dan met een toegangstoken, dat je moet meegeven bij elke call naar de web-API om je Spotify-speler te besturen. Deze authenticatie in twee stappen is gebaseerd op het populaire OAuth2-proces (zie figuur 4).



Figuur 2. De Arduino Oplà IoT-kit.



Figuur 3. De batterijhouder maakt de Oplà IoT-kit draagbaar.



Figuur 4. Het authenticatieproces is gebaseerd op het populaire OAuth 2.0-proces.

Figuur 5. Je moet een "App" maken om...

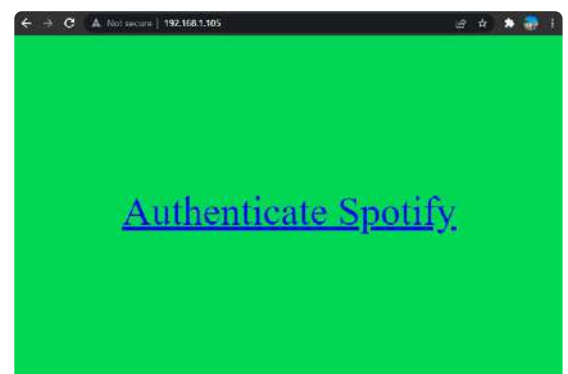
Figuur 6. ...je Client-ID en -Secret te krijgen.

Hoe krijg je een Client-ID en -Secret? Dat gaat met de Spotify App Builder [1], die je kunt gebruiken om je eigen PC-software of mobiele App voor Spotify te maken (figuur 5). Maar dat doen we hier niet; we willen alleen de credentials (zie figuur 6). Client-ID en -Secret moeten worden opgeslagen op ons Arduino MKR-board. Natuurlijk zou je dit hard kunnen coderen in je Arduino-sketch, maar er is een manier die gemakkelijker en veiliger is. De Arduino Web Editor [2] heeft een tab Secrets, waar je omgevingsvariabelen kunt instellen voor later gebruik in je code (figuur 7). Voer je Spotify Client-ID en -Secret en de naam en het wachtwoord van je WiFi-netwerk in de velden van die tab in. Als je de software compileert en uploadt naar de controller, worden je eigen geheime gegevens meteen meegenomen om te worden gebruikt in de code van het project. In je sketch vervang je de strings met de geheime data door een expressie `SECRET_xxx`, dus bijvoorbeeld: `SECRET_SPOTIFY_CLIENT`.

Authenticatie

Om het OAuth2-proces te starten, moet je je authentifieren bij Spotify. Als onze Spotify-controller wordt gestart, logt hij in op het gespecificeerde WiFi-netwerk en geeft hij het van de router verkregen IP-adres weer op het OLED-display. Ik wilde het voor de gebruiker gemakkelijk maken om in te loggen bij Spotify, daarom heb ik de volgende interface gemaakt. De Arduino Controller

Figuur 7. Voer alle geheime gegevens in de tab Secrets van de Arduino Web Editor in, voordat je de code compileert en laadt.



Figuur 8. De webpagina van de controller voert je naar de login van Spotify.

genereert een kleine internetpagina die je kunt zien in een webbrowser, als je het IP-adres van je controller invoert (**figuur 8**). Die pagina bevat een link. (In **listing 1** zie je hoe die internetpagina wordt gegeneerd in de Arduino-code.) Als je de link aanklikt, gaat de browser naar de authenticatie-pagina bij Spotify, waar je gemakkelijk kunt inloggen. Daarna krijg je de vraag of je de controller toestemming geeft om Spotify te besturen (**figuur 9**).

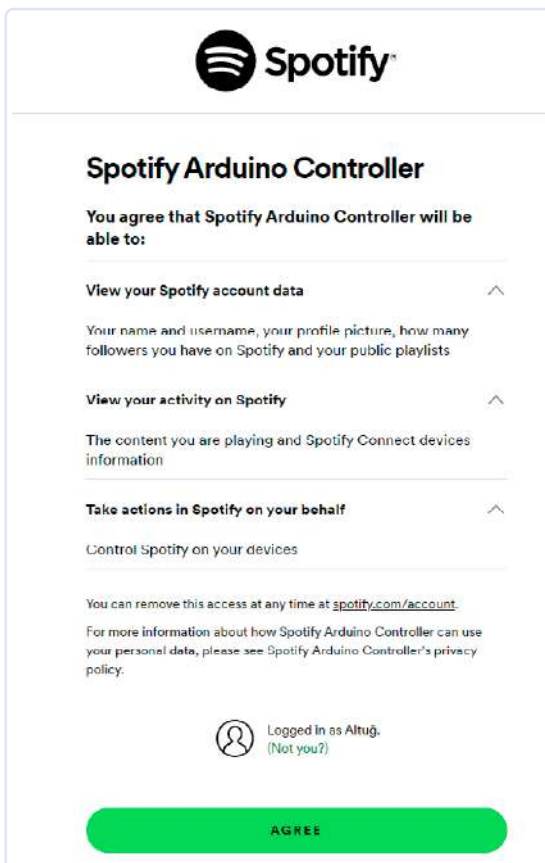
Let op: om dit te laten werken, moet je ook het IP-adres van de controller invoeren als een "Redirect URI" in de Spotify App-editor (**figuur 10**).

Voortaan kan de Arduino Controller nu het API-toegangstoken krijgen door het Client-ID en -Secret naar Spotify te sturen (**listing 2**). Het toegangstoken moet tijdens het gebruik regelmatig worden ververs. Ook dat wordt gedaan door een functie in de sketch (**listing 3**), die elke 3000 seconden wordt aangeroepen.

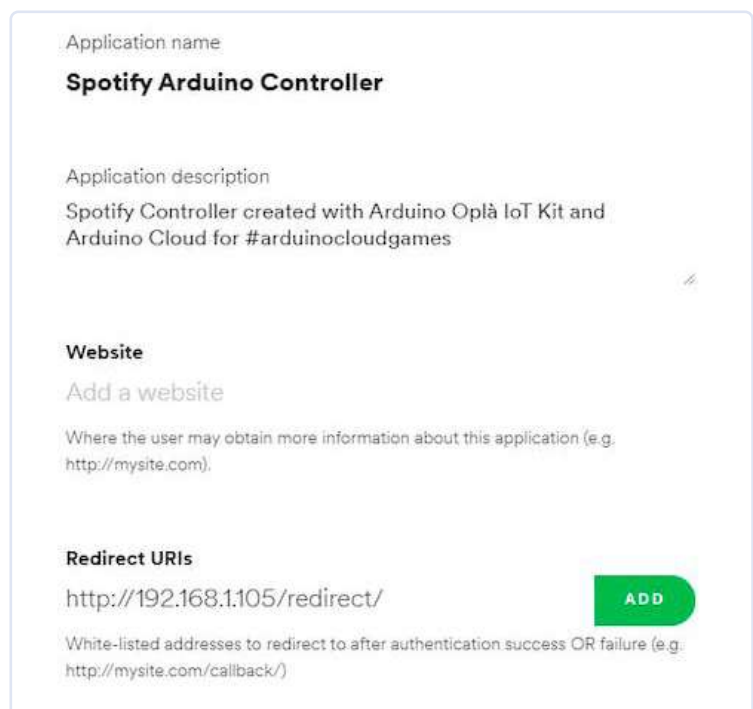


Listing 1. Webpagina voor authenticatie bij Spotify weergeven.

```
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);
```



Figuur 9. Als je bent ingelogd bij Spotify, geef je de controller toestemming om namens jou te handelen.



Figuur 10. Redirect URI: het adres van de controller in je thuisnetwerk.



Listing 2. Functie om het token van Spotify te krijgen voor verder gebruik in de 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;
}
```



Listing 3. Functie om het token te verversen.

```
// 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>();
    }
}
```



Figuur 11. De functie van de knoppen wordt op het display getoond.



Listing 4: Voorbeeld voor gebruik van de API (volgende/vorige nummer).

```
// 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();
}
```

Gebruik

De rest van de code is niet zo ingewikkeld. Op het OLED-display verschijnen het Spotify-logo en de functie van de drukknoppen (figuur 11). Als de gebruiker een knop aaraakt, wordt de bijbehorende API-functie aangeroepen. In listing 4 zie je hoe dat gaat voor het springen naar het vorige of volgende nummer.

Er is ook een functie in de code die de status van de speler opvraagt van de Spotify-API. Het antwoord is een JSON-string. Ik gebruik de library *ArduinoJson.h* en enkele eigen functies voor het verwerken van de JSON-strings. Om de status van de buttons op te roepen, de LED's aan te sturen en figuren op het OLED te zetten, gebruik ik de library *Arduino_MKRIoTCarrier.h*. Je kunt in mijn code duiken voor inspiratie voor je eigen projecten met de Oplà-kit. Mijn software kan worden gedownload van [3].

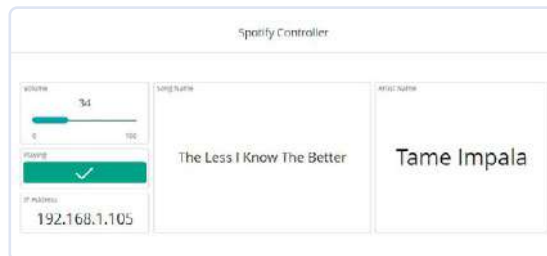
Verbinding met de cloud

Ik heb ook een verbinding met de Arduino Cloud gemaakt voor een dashboard dat het huidige nummer en de naam van de artiest laat zien plus het ingestelde volume (figuur 12). Natuurlijk kun je je eigen persoonlijke dashboard maken, met de gegevens die je wilt zien. Mijn project heeft de derde prijs gewonnen bij de Arduino Cloud Games 2022! ◀

220407-03 (vertaling: Evelien Snel)

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via mail@alt.ug of naar de redactie van Elektor via redactie@elektor.com.



Figuur 12. De huidige titel en naam van de artiest worden verzonden naar de Arduino Cloud, waar ze zichtbaar zijn op je persoonlijke dashboard.

Over de auteur

Altug Bakan heeft als electronicus gewerkt, vooral aan embedded systemen. Hij werkt graag met Arduino vanwege het gebruikersgemak en de snelle resultaten. Zijn favoriete elektronica-onderwerpen zijn bare-metal embedded programmeren en het Internet of Things (IoT).



Gerelateerde producten

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

WEBLINKS

- [1] Spotify App Builder: <https://developer.spotify.com/dashboard/>
- [2] Arduino Web Editor: <https://create.arduino.cc/editor>
- [3] Dit project op create.arduino.cc: <https://create.arduino.cc/projecthub/Altug/opla-spotify-controller-6e7bc4>

Bouw, uitrol en onderhoud van schaalbare, veilige applicaties

op basis van Arduino Portenta X8 – met de i.MX 8M Mini Applications Processor en EdgeLock® SE050 Secure Element van NXP

Een bijdrage van NXP Semiconductors

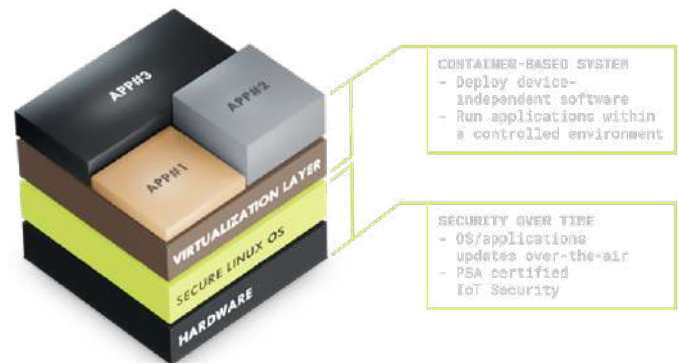
Een IoT-apparaat op de markt brengen vergt aanzienlijke ontwerp- en ontwikkelingsinspanningen – met in elk stadium schaalbaarheids- en beveiligingsproblemen en apparaatbeperkingen. Met de toevoeging van intelligentie wordt het nog ingewikkelder. Dat maakt de selectie van de juiste ontwikkelhardware en -software van cruciaal belang om veilige edge-producten sneller op de markt te brengen. Dit artikel introduceert het Arduino Portenta X8-platform, een industrieel en veilig SOM gebaseerd op de i.MX 8M Mini-applicatieprocessor en een EdgeLock® SE050 hardware secure element van NXP. Dit PSA-gecertificeerde platform is ook Arm® SystemReady IR voor gegarandeerde veiligheid.

Arduino Portenta X8 is een krachtig, industrieel system-on-module met voorgeïnstalleerd Linux OS aan boord en is geschikt om apparaat-onafhankelijke software te draaien dankzij de modulaire containerarchitectuur. Het biedt twee benaderingen: flexibel gebruik van Linux gecombineerd met realtime-toepassingen via de Arduino-omgeving. De ingebouwde WiFi/Bluetooth Low Energy-connectiviteit maakt updates van besturingssystemen/applicaties op afstand mogelijk, waarbij de Linux-kernelomgeving altijd op het hoogste prestatieniveau blijft.

State-of-the-art beveiliging

Het op containers gebaseerde systeem integreert verschillende beveiligingslagen, te beginnen met de hardwarelaag die het Secure Element van NXP huisvest. Het maakt gebruik van het cloudgebaseerde DevOps-platform van Foundries.io [1] waarmee de bouw, test, uitrol en het onderhoud van embedded Linux-oplossingen haast opnieuw is

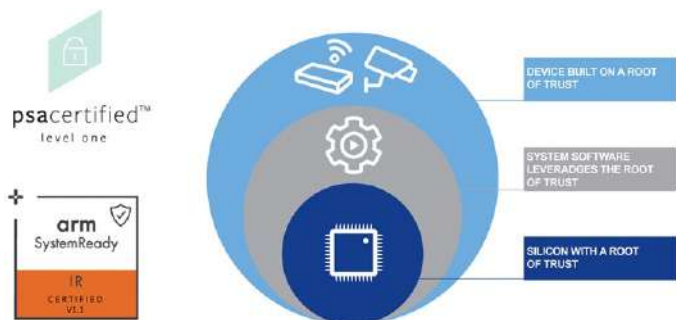
uitgevonden. De Portenta X8 bevat het aanpasbare open-source Linux microPlatform-OS, gebouwd volgens de beste industriële praktijken voor end-to-end beveiliging, incrementele OTA-updates en vlootbeheer.



Portenta X8-container en beveiliging.

Met de virtualisatielaag kunnen gebruikers apparaatonafhankelijke software implementeren die binnen een gecontroleerde omgeving draait. Ze kunnen hun eigen containers maken met behulp van Docker en vooraf gemaakte images downloaden van Docker Hub of andere beschikbare openbare locaties om een op maat gemaakte toepassing te bouwen. Als de ontwikkelaar de embedded wereld wil betreden, kan hij dat gemakkelijk doen door zijn toepassing te bouwen, binnen een container te draaien, naar het board te laden en uit te proberen. Dit biedt een breed scala aan mogelijkheden door de capaciteiten van Linux te mixen met de Arduino-standaardervaring. Portenta X8 behaalde PSA-certificering; het NXP EdgeLock SE050 hardware secure element zorgt voor het genereren van sleutels, versnelde cryptobewerkingen en veilige opslag. X8 behaalde ook Arm SystemReady [2] certificering en geïntegreerde Parsec-services, waardoor het een van de eerste Cassini-producten of Cloud Native Edge-apparaten is die voor ontwikkelaars op de markt zijn. Het draait naadloos Fedora IoT, Fedora Server, Debian en Linux microPlatform. De Portenta X8, die de migratie van cloud-native workloads van de

cloud naar de edge mogelijk maakt, draagt bij aan een cloud-native ontwikkelervaring in het diverse en veilige IoT-ecosysteem van Arm.



Platform-beveiligingsarchitectuur.

EdgeLock SE050 – een veilig anker voor IoT

De EdgeLock SE050 [3] van NXP is discrete en manipulatiebestendige beveiligingshardware voor de bescherming van de identiteit van een apparaat, inclusief cryptografische sleutels en certificaten. Het is een standalone embedded secure element dat via de I²C-interface aan de hoofdp processor wordt gekoppeld. De EdgeLock SE050 is Common Criteria EAL 6+ gecertificeerd voor de hardware en het besturings-systeem. Dit kant-en-klare beveiligde element voor IoT-apparaten biedt een vertrouwensbasis op IC-niveau en levert echte end-to-end beveiliging – van edge tot cloud – zonder de noodzaak om beveiligings-code te implementeren of kritieke sleutels en referenties te verwerken.



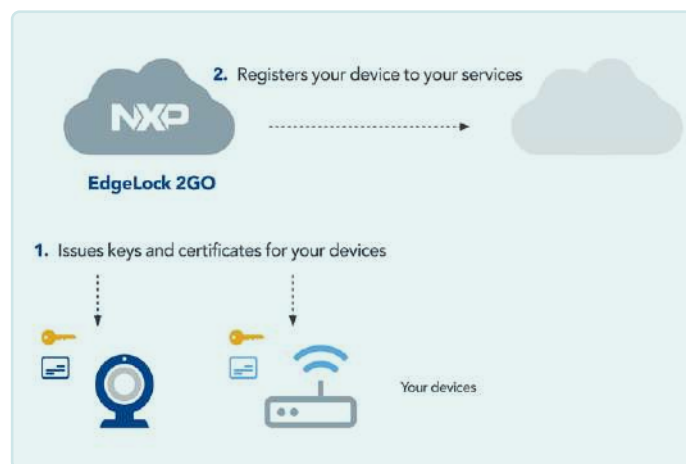
Root of Trust op silicium: het EdgeLock SE050 Secure Element.

EdgeLock SE050 wordt geleverd als een kant-en-klare oplossing met meerdere vooraf geïmplementeerde cryptografische algoritmen en protocollen en een compleet product-ondersteuningspakket dat het ontwerp vereenvoudigt en de time-to-market verkort. Naast bibliotheken voor verschillende MCU's en MPU's biedt het ondersteuningspakket ook integratie met de vele gangbare besturingsystemen, waaronder Linux, RTOS en Android.

Ontwerpers van IoT-apparaten worden geconfronteerd met twee grote uitdagingen bij de implementatie van onboarding naar de cloud: de

provisionering van de apparaatidentiteit en het beheer daarvan zodra deze is vrijgegeven voor het veld. De provisionering van het apparaat verwijst naar de installatie van sleutels en certificaten. Het beheren van apparaatidentiteiten heeft betrekking op het bijwerken, toevoegen of intrekken van sleutels en certificaten gedurende de gehele levens-cyclus van het apparaat.

Om ontwerpers te helpen deze uitdagingen op te lossen, biedt NXP de EdgeLock 2GO [4] managed service. Het platform is een speciaal gebouwde hardware- en servicecombinatie die een op silicium gebaseerde vertrouwensbasis creëert. EdgeLock 2GO verstrekt de voor IoT-apparaten de vereiste identiteiten en installeert de 'credentials' veilig in de EdgeLock SE050-hardware. Ook wordt het IoT-apparaat automatisch rechtstreeks bij de cloudservice geregistreerd.



NXP beheert de device credentials.

Deze flexibele service ondersteunt meerdere soorten referenties en past verschillende configuraties toe, afhankelijk van het project. Credentials kunnen worden vernieuwd of toegevoegd aan apparaten die in het veld worden vrijgegeven. Met de ingebruikname van EdgeLock SE050 en EdgeLock 2GO krijgen gebruikers een end-to-end oplossing die eenvoudig, veilig en flexibel is.

Naarmate IoT blijft groeien, nemen ook de risico's toe. De EdgeLock-combinatie van NXP, met zijn op hardware gebaseerde beveiliging en service voor credential management, geeft fabrikanten van apparaten een veiligere manier om zaken te doen. Als NXP EdgeLock de implementatie van een apparaat ondersteunt, verkort het de time-to-market en verlaagt het de dagelijkse kosten van een IoT-implementatie, terwijl men erop kan vertrouwen dat apparaten worden beschermd door high-level beveiliging.

Ontketend: meer snelheid en efficiëntie

De i.MX 8M Mini [5] SoC is NXP's eerste embedded multicore applicatieprocessor, gebouwd met behulp van geavanceerde 14LPC FinFET-procestechnologie, die meer snelheid en een verbeterde energie-efficiëntie biedt. De i.MX 8M Mini-familie van applicatieprocessors combineert high-performance computing, efficiëntie en embedded beveiliging – nodig voor snelgroeiende edge node computing-applicaties, streaming multimedia en machine learning-toepassingen.



De i.MX 8M Mini SoC wordt aangeboden in varianten met één, twee en vier kernen met Arm Cortex-A53 op maximaal 1,8 GHz per kern. Dankzij een geavanceerd low-power proces is het core complex geoptimaliseerd voor gebruik zonder ventilator, geringe thermische systeemkosten en een lange batterij-levensduur. De Cortex-A kernen kunnen worden uitgeschakeld terwijl het Cortex-M4 subsysteem energiezuinig, realtime systeembewaking uitvoert. De DRAM-controller ondersteunt 32-bit/16-bit LPDDR4-, DDR4- en DDR3L-geheugen, wat een grote flexibiliteit in het systeemontwerp garandeert.

De i.MX 8M Mini core-opties zijn geoptimaliseerd voor ultra-low-power, zelfs sub-watt in specifieke toepassingen, maar bieden de ruime verwerkingskracht die nodig is voor consumenten-, audio-, industriële, machine learning training- en inferencing-toepassingen in een reeks van cloud providers. De i.MX 8M Mini SoC bevat ook hardwarematige 1080p video-versnelling om tweeweg videotoe toepassingen mogelijk te maken, 2D en 3D graphics om een rijke visuele HMI-ervaring te bieden, en geavanceerde audiomogelijkheden. Een uitgebreide selectie van high-speed interfaces maakt brede systeemconnectiviteit mogelijk en is gericht op kwalificatie op industrieel niveau.


Toepassingsvoorbeelden omvatten:

Industriële automatisering

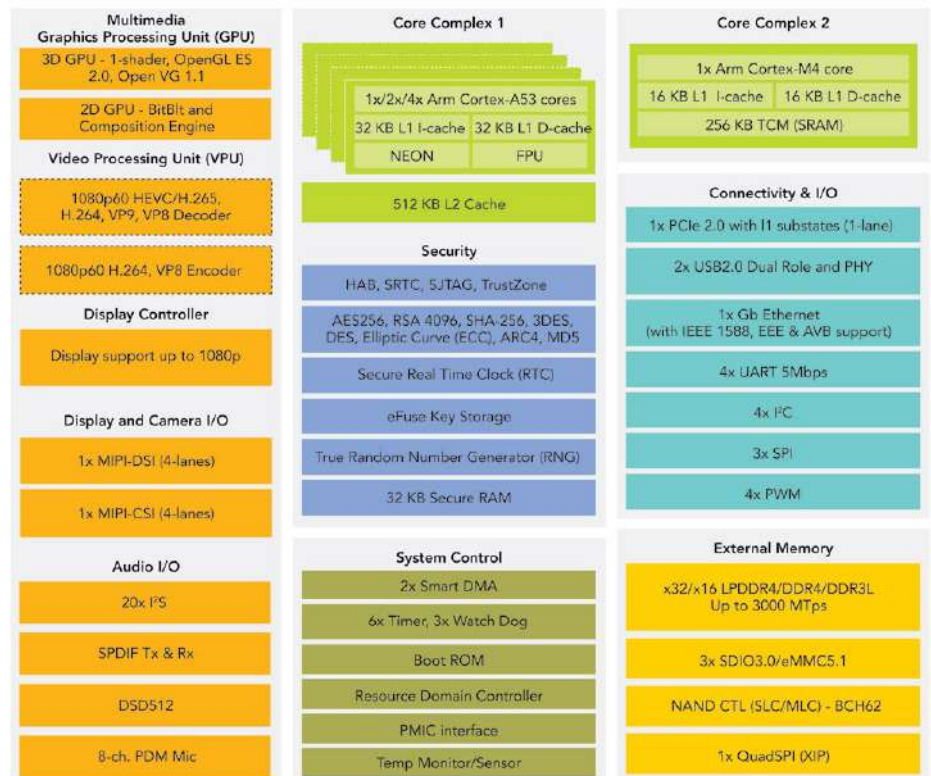
- De Portenta X8 kan hier fungeren als een multi-protocol gateway, waarbij gegevens via WiFi, LoRa, NB/IoT, LTE Cat.M1 naar de Cloud of het ERP-systeem worden gestuurd.
- De beschikbaarheid van Linux-containers zoals ROS binnen de Arduino-omgeving maakt de Portenta X8 zeer geschikt voor autonoom geleide voertuigen.

Gebouwwautomatisering

- De Portenta X8 werkt samen met slimme omgevingsensoren en maakt de implementatie van real-time ML en edge-beeldverwerking mogelijk.
- Slimme kiosken maken meestal gebruik van verschillende componenten (zoals kaartlezers, camera's, microfoons), wat diverse soorten I/O's vereist. In combinatie met een Max Carrier zorgt de Portenta X8 voor WiFi-connectiviteit en kunnen beheerders het machinegebruik op afstand controleren.
- De Portenta X8 kan tegelijkertijd HVAC-systemen bedienen, slimme apparaten in- en uitschakelen, verlichting autonoom aanpassen en edge-toegang regelen.

Ga vandaag nog aan de slag met de industriële, veilige Portenta X8 SOM [6] voor uitgelezen rekenkracht op een klein oppervlak. 

220576-03



 Optional Capability

i.MX 8M Mini Applications Processor: blokschema

WEBLINKS

- [1] Foundries.io: <https://foundries.io/>
- [2] Arm SystemReady: 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: www.arduino.cc/pro/hardware/product/portenta

Nieuw Arduino- of elektronica project?

Deel het met onze community

Volg ons op:



www.twitter.com/ElektorNL



www.instagram.com/elektorlabs



www.facebook.com/ElektorNL





De menselijke factor bij productie

▲
Daria Baradel (vooraan)
en haar Arduino-
collega's.

maak kennis met Daria Baradel,
verantwoordelijk voor de productie bij Arduino

Keith Jackson (Arduino)

Hoe heeft een van 's werelds populairste elektronicafabrikanten met succes de problemen met de toeleveringsketen van de afgelopen drie jaar overwonnen? Door een niet aflatende passie voor innovatie, nauwe samenwerking met lokale partners en de menselijke factor.

Keith Jackson: Hallo Daria. Met alles wat we in het nieuws horen over het wereldwijde tekort aan elektronische componenten, die de productie in fabrieken over de hele wereld afremt, moet het een uitdagend jaar zijn geweest om de productie bij Arduino te leiden?

Daria Baradel: Dat klopt. Iedereen het heeft over tekorten in de elektronica-industrie die alles beïnvloeden, van nieuwe auto's tot gameconsoles, maar het is niet alleen het gebrek aan voorraden dat de grootste

problemen veroorzaakt, het zijn de onvoorspelbare vertragingen en constante kostenstijgingen die het vrijwel onmogelijk maken om de boel onder controle te houden.

Voor ons hele team was een andere aanpak nodig, en meer flexibiliteit. Productie onder controle houden vereist voorspelbaarheid en risicomangement, maar wanneer die voorspelbaarheid er niet meer is, dan wordt het op peil houden van de productie elke dag weer een uitdaging. Op elke willekeurige dag kan een levering waar je op rekt worden geannuleerd, waardoor de productie in de fabriek stopt. Het zijn echter niet altijd annuleringen die wijzigingen op korte termijn in de productieplanning veroorzaken. Soms is juist het tegendeel het geval, wanneer we opeens de een willekeurige levering van componenten kunnen verwerken die eigenlijk pas in 2023 werd verwacht. Wanneer dat de enige ontbrekende component in een onderdelenlijst betreft, geeft ons dat plotseling de mogelijkheid om een model weer in productie te krijgen, dus we passen voortdurend onze planning aan aan wat beschikbaar is en aan wat onze klanten willen. We moesten dus alle standaardprocedures herzien en aanpassen aan deze instabiele en onvoorspelbare situatie, waardoor ze minder gestandaardiseerd en meer flexibel werden, zodat we elk probleem dat zich zou kunnen voordoen afzonderlijk kunnen aanpakken.

Keith: Kun je wat dieper ingaan op hoe jij en het team jullie aanpak hebben aangepast om met deze uitdagingen om te gaan?

Daria: Aanvankelijk probeerde ik te voorspellen wat er zou kunnen gebeuren, maar dat bleek onmogelijk. Dus moest ik mijn instelling veranderen en meer proactief te werk gaan. Als team waren we het erover eens dat we meer controle moesten krijgen over de leveringen, daarom vragen we elke vrijdag bij al onze leveranciers na wat er de volgende week geleverd zal worden. Voor de belangrijkste componenten checken we dit daarna dan nog elke dag.

Het is echter niet alleen een kwestie van hoe we met onze leveranciers samenwerken, het werd ook erg belangrijk om veel nauwer samen te werken met het Arduino-hardwareteam, verantwoordelijk voor het ontwerpen en testen van de producten. Alvorens een onderdelenlijst voor een nieuw product af te ronden, en zelfs nog voordat bepaald wordt welke componenten er voor de prototypes van een ontwerp worden gebruikt, komen beide teams nu bijeen om de beschikbaarheid en betrouwbaarheid van de leverancier te beoordelen, als onderdeel van het ontwerpproces.

De beschikbaarheid van een component is nu net als

de eigenschappen ervan een aspect bij het definitieve ontwerp. Dat is echter niet per se negatief, want het team heeft nu een veel beter inzicht in de ontwerpen en de componenten omdat ze veel eerder bij het ontwerpproces zijn betrokken. We hebben nu dus allemaal al een goed idee van de initiële opzet van elk nieuw product.

In 2022 heeft het toeleveringsteam samen met het hardwareteam meer dan 100 alternatieve componenten tegen het licht gehouden en het ontwerp van 16 producten aangepast, om de onderdelenlijst zo te veranderen dat de productie kon doorgaan.

Keith: Hoe reageerden de leveranciers – Arduino is immers slechts een van de vele afnemers die schreeuwen om componenten?

Daria: Ironisch genoeg heb ik de afgelopen jaren veel tijd geïnvesteerd in het automatiseren van zoveel mogelijk inkoopprocedures voor de componenten. Dit jaar is het menselijke aspect echter weer op de voorgrond gekomen. Het is essentieel om te communiceren en samen te werken met andere mensen. Alleen door hen voortdurend te bellen zijn we in staat om ons aan te passen aan veranderingen op het laatste moment en de domino-gevolgen daarvan. Waarschijnlijk vinden ze me nu een soort stalker, met al die berichten die ik ze elke dag heb gestuurd!

Veel van onze leveranciers hebben overigens veel belangstelling voor het Arduino-concept, want heel wat daarvan zijn ook Arduino-liefhebbers. En daarom ook kregen we veel steun van sommige leveranciers, en soms minder van andere. In het algemeen zijn degenen die ons flink ondersteunden nu ook vaste Arduino-leveranciers, die nauw met ons samenwerken als onderdeel van één grote gemeenschap.

Keith: Hoe loopt de productie nu en wat zijn je vooruitzichten voor de toekomst?

Daria: Ondanks alles wat er dit jaar is gebeurd is het productievolume nog altijd meer dan 20% hoger dan in dezelfde periode in 2021. Dit hebben we bereikt doordat elk team zijn deel heeft bijgedragen. Het hardwareteam heeft een enorme bijdrage geleverd door letterlijk honderden alternatieve componenten te controleren en goed te keuren, in veel gevallen deze dezelfde dag nog te monteren en te testen voor het definitieve groene licht, zodat we de beschikbare voorraden konden inkopen. Hoewel er tot eind 2023 geen licht aan het einde van de tunnel lijkt wat betreft componentenschaarste en lange levertijden, ben ik er nog steeds van overtuigd dat we kunnen blijven groeien, omdat bedrijven in laatste instantie door mensen worden geleid. De veranderingen in instel-





ling en procedures die we dit jaar hebben moeten doorvoeren, zullen ons in de toekomst alleen maar sterker en efficiënter maken.

Keith: Je noemde het testen en goedkeuren van alternatieve componenten op een en dezelfde dag. Hoe krijgen jullie dat voor elkaar?

Daria: Dat onze fabrikanten dicht bij de Arduino R&D-vestigingen zitten, was een enorm voordeel tijdens de Covid-pandemie, en in deze tijd van tekorten. Arduino is altijd trots geweest op productie in Italië, waarbij alle boards worden vervaardigd in twee fabrieken in de regio Piemonte in Italië. In de praktijk konden de technici bij problemen in een auto sprongen en onmiddellijk naar de fabriek bij Turijn gaan om de componenten te controleren en indien nodig te vervangen. Wanneer we horen dat een alternatief onderdeel beschikbaar is, moet de bestelling dezelfde dag nog worden bevestigd, anders wordt de voorraad door anderen ingepikt. Je moet er heel snel bij zijn. Dus de technici gaan naar de fabriek, testen het alternatieve onderdeel en geven ons groen licht, zodat de bestelling diezelfde dag nog kan worden geplaatst. Dit zou overduidelijk niet mogelijk zijn geweest als onze productie ver weg zou plaatsvinden.

Keith: Arduino is kennelijk een van de weinige elektronicabedrijven zijn die nog steeds lokale fabrikanten gebruiken om zijn boards te produceren?

Daria: Inderdaad. Lokaal produceren was een bewuste keuze omdat Arduino altijd een positief beleid heeft gevoerd betreffende duurzaamheid, milieu-impact en de lokale gemeenschap. Onze duurzaamheidsdoelstellingen zijn meer dan een stuk papier: we hebben altijd gestreefd naar een zo gering mogelijke CO₂-uitstoot en het reduceren van afval. We geven bijvoorbeeld oude voorraad weg aan plaatselijke scholen, en de productie en verpakking van onze producten vindt plaats binnen een straal van 50 km van ons vestiging in Turijn. De lokale gemeenschap krijgt de voorkeur bij het kiezen van onze partners en bij het lokaal vervaardigen van de producten. Het samenstellen en verpakken wordt uitgevoerd door een lokaal bedrijf, dat actief mensen met een beperking in dienst neemt en hen de mogelijkheid biedt om onafhankelijk te worden.

Keith: Het klinkt alsof je erg trots bent om bij Arduino te werken. Maar wat inspireert je nog meer dan alleen het aanpakken van dagelijkse tekorten aan componenten?

Daria: Ik ben van huis uit ingenieur, met twee masterdiploma's in production engineering en engineering management. In 2014, toen ik bij Arduino als projectmanager aan de slag ging, produceerden we tussen de 3000 en 5000 boards per maand. Dat zijn er nu meer dan 50 keer zo veel, dus heb ik nu een team van acht mensen om dit te managen.

Afgezien van mijn werk reis ik ook graag en doe ik veel aan sport, vooral klimmen, hardlopen, windsurfen en beachvolleybal. Ik vind sport de beste uitlaatklep voor de dagelijkse stress. Maar als ik al een droom voor de toekomst had dan zou het zijn om ooit een boerderij te hebben. Want ik ben afkomstig uit een familie die koeien, schapen en geiten hield en hou van dieren. En ik weet zeker dat ik zelfs dan overal Arduino's zou hebben om weer zo efficiënt mogelijk bezig te zijn. **◀**

220426-03 (vertaling: Marc Gauw)

Over de auteur

Keith Jackson doet marketing voor Arduino en heeft een passie voor alles wat Arduino betreft, omdat het meer is dan een bedrijf of een merk: het is een echte gevarieerde community.

Vragen of opmerkingen?

Hebt u vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via k.jackson@arduino.cc, of naar de redactie van Elektor via redactie@elektor.com.

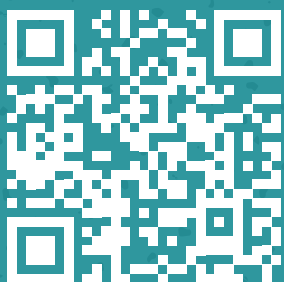
MicroPython Enters the World of Arduino

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



Ontwikkelborden

verleden, heden en toekomst

Mark Patrick (Mouser Electronics)

De term „ontwikkelbord“ wordt de laatste jaren gebruikt om verschillende “hardwareboards” voor ontwikkelingsdoeleinden te beschrijven, zoals „demonstratieborden“, „evaluatiekits“ en „referentie-ontwerpen“.

In dit artikel gaan we dieper in op de verschillende betekenissen van „ontwikkelbord“ en laten we zien hoe deze verschillen van hun nauwe verwant - de SBC (single board computer). Wij brengen hun evolutie van verleden tot heden in kaart en onderzoeken enkele trends m.b.t. hun eventuele ontwikkeling in de toekomst.

Wat is een ontwikkelbord?

Om te beginnen moeten we eerst duidelijk vaststellen wat bedoeld wordt met een ontwikkelbord en hoe het verschilt van een SBC (single-board computer). Een ontwikkelbord wordt meestal gemaakt door een fabrikant van een microcontroller om de functies te benadrukken (hoewel de term nu ook vaak wordt gebruikt voor andere soorten componenten). Een microcontroller is een geïntegreerd circuit dat een processor, wat RAM, flash-opslag en IO-functionaliteit bevat, zodat hij ook met de “echte wereld” kan communiceren. Hij functioneert in feite als een miniaturcomputer in één enkel pakket, met als doel ontwikkelaars een handige manier te bieden om ermee te werken en externe componenten aan te sturen,

zoals lampen, kleine motoren, enz. Een SBC biedt deze functionaliteit ook, met als belangrijkste verschil dat de CPU, het RAM en de opslag elk in afzonderlijke IC's op het bord zijn ondergebracht, en dat er interfaces zijn voor aansluiting op een toetsenbord en/of beeldscherm. De microprocessor op een SBC vereist een besturingssysteem, terwijl een microcontroller wordt beheerd met behulp van een geïntegreerde ontwikkelomgeving (IDE) van de fabrikant. De hedendaagse ontwikkelborden bevatten vaak een microcontroller die niet zozeer als doel heeft om de functies van de microcontroller zelf te demonstreren, maar die van gekoppelde sensoren of andere geïntegreerde circuits. Deze worden “demonstratieborden”, “evaluatiekits” of, indien ze zijn samengesteld om onderdelen te verzamelen voor een tastbaar doel, “referentie-ontwerpen” genoemd.

Sommige borden zijn niet primair bedoeld om hardware te ontwikkelen, maar om softwareontwikkelaars toegang te verschaffen tot reële gegevens voor het creëren en het verfijnen van de algoritmen die nodig zijn voor toepassingen op het gebied van AI en machine learning. Hoewel ze misschien niet voldoen aan de oorspronkelijke definitie en het doel van een “ontwikkelbord”, worden ze nu collectief gezien als een verwijzing naar elk stuk hardware dat kan worden gebruikt bij de ontwikkeling van hardware en/of software van nieuwe elektronische producten.

Verleden

Het eerste microcontroller-ontwikkelbord dat de aandacht trok van de engineering-gemeenschap verscheen in 2006. Dit prototypingplatform, dat later bekend werd als Arduino [1] (figuur 1), werd positief ontvangen door een nieuwe categorie ontwerpers op het gebied van elektronica,

waaronder liefhebbers, hobbyisten en doe-het-zelvers. Arduino legde de basis voor het commerciële succes van latere SBC's en op microcontrollers gebaseerde platforms en werd in 2008 opgevolgd door Beagleboard [2], dat technici een goedkoop en door de gemeenschap ondersteund open-source ontwikkelplatform bood. In 2012 kwam de Raspberry PI [3] op de markt, de eerste single board computer. Deze SBC werd, net als Beagleboard, speciaal ontworpen als een educatief platform bedoeld om studenten een goedkope manier te bieden om programmacodes te leren schrijven. Dankzij zijn grote aantrekkingskracht trok de Raspberry PI een veel breder publiek dan alleen studenten en werd snel omarmd door zowel amateurhobbyisten als professionele technici.

Heden

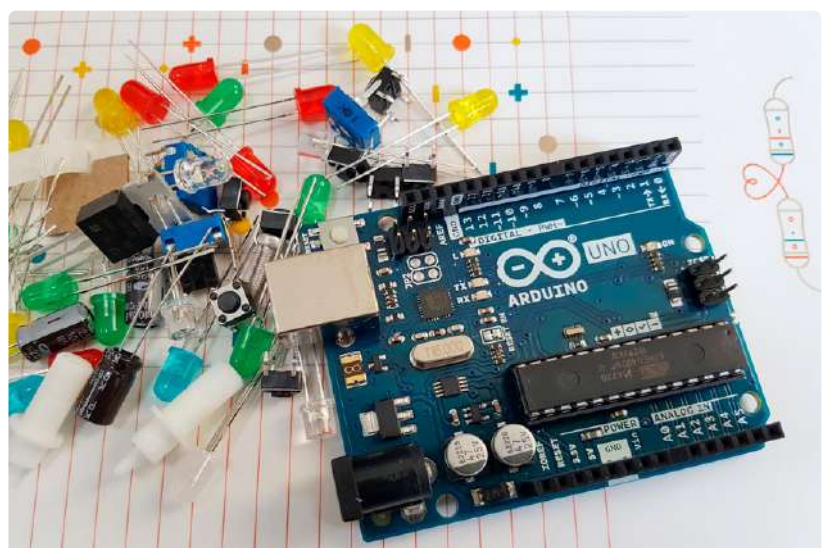
Vandaag de dag zijn er twee hoofdcategorieën SBC's - merkgebonden en open-source. Merkgebonden SBC's zijn doorgaans ontworpen voor gebruik in eindtoepassingen en worden onderworpen aan dezelfde tests en kwaliteitsborging als andere eindproducten. Ze zijn ofwel geïntegreerd in elektronische apparatuur of geïnstalleerd in een kast. Open-source SBC's bieden gebruikers toegang tot het hardware-ontwerp, de lay-out en alle benodigde broncodes. Op deze manier kunnen ze snel en gemakkelijk leren hoe de hard-/software werkt en het ontwerp vervolgens aan hun eisen aanpassen. De hedendaagse ontwikkelborden en SBC's worden geleverd met een grote verscheidenheid aan processors. Deze variëren van op X86 gebaseerde types in de traditionele pc-sferen (AMD en Intel) tot ARM-processors voor industriële en mobiele toepassingen. Linux en zijn afgeleiden (Ubuntu, Fedora, Debian, enz.), Android en Windows CE zijn de meest gebruikte besturingssystemen op SBC's. Microcontroller-ontwikkelborden hebben geen besturingssysteem nodig en worden geprogrammeerd via een door de fabrikant geleverde IDE. Microcontroller-ontwikkelborden en SBC's zijn beide uitgerust voor draadloze connectiviteit (wifi, bluetooth) en de meest recente audio- en video-interfaces. Dit betekent dat sommige SBC's nu functies hebben die gelijkwaardig zijn aan die van veel pc's en tablets.

Toekomst: Ontwikkelborden worden het eindproduct

Van oudsher creëerden fabrikanten ontwikkeltools met de bedoeling deze te gebruiken als marketingsteun om de kans op verkoop van hun microcontrollers aan potentiële klanten te vergroten (in de industrie vaak "Design-in" genoemd). Zij hoopten dat door de tijd die een ontwerper besteedt in een laboratorium aan de vervolmaking van een onderdeel te minimaliseren en door de toegang tot en het onderzoek naar de functionaliteiten te vereenvoudigen, de kans groter zou zijn dat hij voor zijn product-

prototyping voor hun microcontroller en bijkomende onderdelen zou kiezen, wat uiteindelijk zou leiden tot meer bestellingen in het geval van massaproductie. Voor producten waarbij het verschil in technische specificaties tussen onderdelen van verschillende leveranciers verwaarloosbaar is, is dit een voorzichtige aanpak. Deze strategie kent echter ook een keerzijde. Sommige fabrikanten werden in bepaald opzicht het slachtoffer van hun eigen succes. Zij realiseerden zich dat ze de hoeveelheid werk die een ontwerper aan zijn product besteedt moeten blijven verminderen. Het ontwikkelbord is hierbij de belangrijkste onderscheidende factor geworden. Dit geldt vooral voor producten die in grote lijnen vergelijkbaar zijn met die van hun concurrenten. De verwachtingen van ontwerpers zijn zodanig gestegen dat zij, zelfs voor onderdelen met een duidelijk aanwijsbaar concurrentievoordeel (bijvoorbeeld qua vermogen of snelheid), nog steeds veronderstellen dat de bijbehorende ontwikkelborden plug-and-play toegankelijk zijn. Fabrikanten hebben hun waardevoorstel verder verbeterd door referentie-ontwerpen aan te bieden, bestaande uit een microcontroller en andere geïntegreerde circuits (meestal sensoren). Aanvankelijk waren deze bedoeld als leidraad voor de manier waarop apparaten met elkaar konden worden verbonden om de elektrische functionaliteit van een eindproduct te emuleren, waarbij weinig aandacht werd besteed aan de vormfactor, de grootte van het ontwerp of de produceerbaarheid. Sommige fabrikanten hebben hun referentie-ontwerpen echter naar een hoger niveau getild om volwaardige prototypen en zelfs volledig levensvatbare producten te creëren. De HSP (health sensor platform) referentie-ontwerpen [4] van Maxim Integrated (nu onderdeel van Analog Devices) kunnen worden gebruikt als voorbeeld om deze evolutie in kaart te brengen. De eerste versie van deze

Figuur 1: Arduino microcontroller-ontwikkelbord. (Source: Shutterstock)





▲
 Figuur 2: HSP3.0 van Maxim Integrated.

referentie-ontwerpen was een klein ontwikkelbord met een assortiment sensoren (temperatuur, druk, versnellingsmeter, biopotential, enz.). Het bord was uitermate geschikt voor gezondheids- en fitness toepassingen en configureerbaar met een microcontroller. Zijn opvolgers, HSP2.0 en HSP3.0, hadden een vormfactor die het mogelijk maakte om ze om de pols te dragen en lijken sterk op andere, op de markt verkrijgbare draagbare apparatuur (figuur 2).

Dit stelde ontwikkelaars in staat om de functionaliteit van hun sensoren in reële scenario's te evalueren. Belangrijk hierbij was dat deze ontwerpen softwareontwikkelaars vrije toegang verschaften tot sensormetingen (informatie die niet gemakkelijk toegankelijk is bij andere draagbare gezondheids- en fitnessapparatuur). Het doel van deze aanpak was om algoritmen voor AI en machine learning te ontwikkelen die waarde zouden toevoegen aan de toepassing.

▼
 Figuur 3:
 Programmeerbare Logische Controller.
 (Source: Shutterstock)



Door te laten zien hoe hun hardware de toegang tot gegevens vergemakkelijkt, hoopte Maxim dat producentontwikkelaars m.b.t. hun sensoroplossing zouden kiezen voor enkele (of alle) IC's. Maxim breidde deze aanpak zelfs uit tot de ontwikkeling van de MAX HEALTH BAND [5] (pols) en MAX ECG MONITOR [6] (borstband), beide speciaal ontworpen en samengesteld als draagbare gezondheids- en fitnessapparatuur. Hoewel het niet de bedoeling was om ze op de consumentenmarkt aan te bieden, konden bedrijven een overeenkomst met Maxim sluiten om deze producten onder hun eigen merknaam uit te brengen in ruil voor een royaltybetaling. Het op deze manier aanbieden van een volledig functionerend product waarbij al het ontwikkelingswerk al is gedaan, heeft een potentiële aantrekkingskracht op een nieuwe en bredere niet-technische zakelijke klantenkring. Nordic Semiconductor's Thingy:91 [7]. Ze moeten software en algoritmen ontwikkelen waarmee de intrinsieke waarde van de hardware kan worden gerealiseerd (ze maken zo de voor de hand liggende keuze voor nieuwe productontwerpen die deze algoritmen benutten). Waarschijnlijk zullen in de toekomst nog meer fabrikanten voor deze aanpak kiezen.

Toenemend gebruik van ontwikkelborden in industriële producten

Het aanpassen van ontwikkelborden en SBC's voor gebruik in commerciële producten komt steeds meer voor, maar een andere opkomende trend is hun gebruik in toepassingen met een lager volume, maar een hogere waarde - industriële eindproducten, zoals PLC's (Programmeerbare Logische Controllers) (figuur 3), waarvoor strengere normen gelden dan voor hun commerciële tegenhangers.

Testborden voor industriële toepassingen

Veel van de huidige SBC's zijn per definitie volledig geverifieerde ontwerpen geworden omdat de onderdelen die ze bevatten oorspronkelijk werden ontwikkeld voor gebruik in eindproducten. Ze zijn derhalve getest en gecontroleerd op kwaliteit. Dit komt ook doordat open-source ontwerpen voortdurend worden beoordeeld door een leger van competente ontwerpers en programmeurs die de borden en de software die ze gebruiken updaten en evalueren. Het testen van SBC-borden wordt nu uitgevoerd door hoogwaardige ontwerp bureaus en productiebedrijven. De borden worden onderworpen aan dezelfde strenge kwaliteitscontrole als alle andere eindproducten, waardoor ze zelfs in aanmerking komen voor een CE- of FCC-certificering. Deze teststroom kan eenvoudig worden uitgebreid om aan de eisen van industriële producten te voldoen.

Anderzijds worden door fabrikanten of derden geleverde microcontroller-ontwikkelborden, die gewoonlijk ook geschikt zijn voor gebruik in commerciële producten, doorgaans niet aan dezelfde strenge testnormen onderworpen. Dit betekent dat fabrikanten ze momenteel (in hun huidige vorm) niet aanbevelen voor onmiddellijk gebruik in deze toepassingen.

Hoewel sommige borden zijn uitgerust met hoogwaardige industriële componenten, zijn ze meestal slechts van commerciële kwaliteit. Ze zijn specifiek ontworpen om te functioneren bij kamertemperatuur. Prototypes van ontwikkelborden worden gewoonlijk gedurende enkele dagen of weken getest bij kamertemperatuur, maar dit verschilt per fabrikant, omdat er geen vaste normen zijn. De belangrijkste kwaliteitseis voor fabrikanten is dat hun borden betrouwbaar functioneren bij een gangbare omgevingstemperatuur. Daarom moet de consument zich ervan bewust zijn dat het aangekochte product waarschijnlijk niet is getest bij extreme temperaturen of vochtigheid. De borden worden normaliter ook niet getest op de belastingen die gepaard gaan met intense trillingen of schokken.

Dus het belangrijkste punt om op te letten bij het bepalen van welk ontwikkelbord te gebruiken in een industriële toepassing, is het risico. Hoe lager, hoe beter. Bij het selecteren van een bord voor gebruik in een industriële toepassing moeten de componenten van het bord van de juiste temperatuurklasse zijn. Het is ook zinvol om gelijktijdig verschillende borden bij een hoge temperatuur gedurende enkele dagen te onderwerpen aan een stresst-

est. Dit geldt ook voor ontwikkelborden in producten die blootstaan aan een hoge vochtigheidsgraad. Ook deze dienen onder vergelijkbare condities te worden geëvalueerd. Borden bedoeld voor gebruik in een toepassing met hoge trillingen moeten eerst in een testframe op trillingen worden getest.

Conclusie

SBC's en microcontroller-ontwikkelborden bieden kleine bedrijven een handige manier om hun ontwerpen snel op de markt te brengen zonder de ontwikkelingskosten van nieuwe hardware. Hierdoor kunnen zij zich concentreren op software-innovatie en in toenemende mate op de ontwikkeling van algoritmen voor AI en machine learning. SBC's en ontwikkelborden zijn nu veel breder inzetbaar dan oorspronkelijk was voorzien en hebben een grote impact gehad op de recente geschiedenis van de elektronica-industrie. Ze worden steeds krachtiger, slimmer en responsiever, terwijl ze toch gemakkelijk toegankelijk blijven voor zowel professionele technici als elektronicaliefhebbers.

220597-03



Over de auteur

Als Technical Marketing Manager voor EMEA bij Mouser Electronics is Mark Patrick verantwoordelijk voor het maken en verspreiden van technische content in die regio – content die essentieel is voor de strategie van Mouser om zijn technische publiek te ondersteunen, te informeren en te inspireren.

Voordat hij leiding gaf aan het Technical Marketing-team, maakte Patrick deel uit van het EMEA Supplier Marketing-team en speelde hij een cruciale rol bij het aangaan en ontwikkelen van relaties

met belangrijke productiepartners. Naast een verscheidenheid aan technische en marketingfuncties, heeft Patrick onder andere acht jaar bij Texas Instruments in Applications Support en Technical Sales gewerkt.

Als 'hands-on' engineer in hart en nieren, met een passie voor vintage synthesizers en motorfietsen, draait hij zijn hand er niet voor om die te repareren. Patrick heeft een eerstegraads Honours Degree in Electronics Engineering van Coventry University.

WEBLINKS

- [1] Arduino Boards Distributor: <https://elektor.link/MouserArduino>
- [2] BeagleBoard Distributor: <https://elektor.link/MouserBeagleBoard>
- [3] Raspberry Pi Distributor: <https://elektor.link/MouserRaspberryPi>
- [4] Maxim Integrated Distributor: <https://elektor.link/MouserMaxim>
- [5] MAX HEALTH BAND: <https://elektor.link/MouserMaxHealthBand>
- [6] MAX-ECG-MONITOR : <https://elektor.link/MouserMaxECGMonitor>
- [7] Nordic Semiconductor Thingy:91™ Multisensor Prototyping Kit : <https://elektor.link/MouserThingy91>

Declassified Bonuseditie!

Beschikbaar in week 01

▲
Figuur 1
van een
"Creatin



...yseten van papier en andere dunne materia-
len ondersteunt. Onlangs heb ik geëxperimenteerd met

De deelnemers aan de workshop hadden zeer uiteen-
lopende achtergronden en kennis van elektronica.



Doo
vert
spe
slag
well
Enk
beh
Fou
frec
sele
Elk
wa
per
LEI
ma
de
Ee
vo
La
de
ga

Declassified Bonuseditie!

Beschikbaar in week 01



an de

Declassified Bonuseditie!

Beschikbaar in week 01



hieronder één voor één bespreken.

de microfoon in een array te zetten, om de

Figuur

Figuur



Declassified Bonuseditie!

Beschikbaar in week 01



net
vare
re
re
va
w
na
w
or
ee
El
is
du
bl
ma
als
On
mc
bl
na
12
mA
16,

1 2
AR
Declassified
Bonuseditie!

Beschikbaar in week 01

Figuur 9:
SWD-pe





Declassified Bonuseditie!

We
pers
sch
om
prin
Afh
waa
len
eer

C

We
de
is o
pri
op
we

Beschikbaar in week 01



Aan de slag met

n

Declassified Bonuseditie!

Beschikbaar in week 01



ter wereld.

www.elektor.nl/19351

Declassified Bonuseditie!

Beschikbaar in week 01



...powered synthesizer te maken!

www.elektor.nl/20330

Arduino Sensor Kit Base

www.elektor.nl/19944

heid om (AI) Edge Computing-toepassingen te draaien.

www.elektor.nl/19936

Met steun van deze Arduino-resellers

Dit nummer van Elektor, onder gastredactie van Arduino, is mogelijk gemaakt door de ondersteuning door deze leden van de Arduino reseller-community.

Hier vind je alles wat je van Arduino nodig hebt!



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