

IoT-simulatie vereenvoudigd met Wokwi

vragen van Roberto Armani (Elektor)

Wil je processoren en printen simuleren? Kijk dan eens naar Wokwi, een innovatieve open-source simulator voor embedded systemen en IoT-apparaten. De maker ervan, Uri Shaked, vertelt over de oplossing en over zijn achtergrond en interesses.

Figuur 1. Mijn Chanoeka Menora-project, dat ik in de vierde klas in elkaar heb gezet en dat nog steeds functioneert!

Elektor: Allereerst bedankt voor je tijd, Uri. Kun je ons iets vertellen over jezelf, je achtergrond? Waar ben je momenteel gevestigd?

Uri Shaked: Ik ben software-ingenieur en woon in Israël. Als kleuter haalde ik altijd al mijn speelgoed uit elkaar, omdat ik nieuwsgierig was hoe het werkte. In

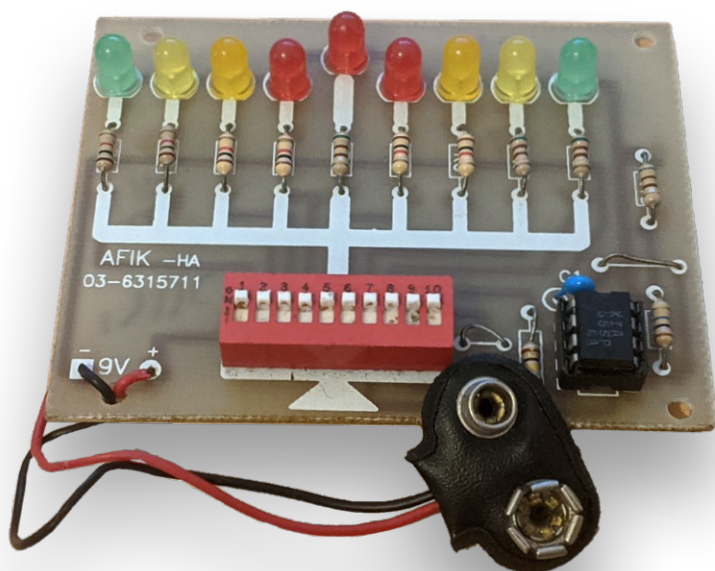
de derde klas maakte ik mijn PC/XT-machine kapot. Ik wilde graag leren hoe ik hem moest repareren, dus kocht mijn tante een boek over MS-DOS voor me. De laatste twee hoofdstukken waren een korte introductie tot QBasic. Ik was verkocht. Vanaf dat moment bleef ik programmeren, elke dag.

Elektor: Waardoor werd je interesse in elektronica gewekt? Was het een specifiek project, een bepaalde cursus, de passie voor elektronica van een familielid of misschien een leraar?

Uri Shaked: Op de basisschool hadden we wekelijks elektronica-les. Het was heel praktijkgericht: we oefenden solderen en assembleerden printen die de leraar ons gaf, terwijl we de basisbeginselen leerden. Ik weet nog hoe leuk ik deze lessen vond en het gevoel van voldoening als ik een voltooid project in handen had (figuur 1). Deze labs zorgden inderdaad voor de fundamentele kennis en ervaring en wekten mijn passie.

Elektor: Niemand vergeet zijn 'eerste liefde'. Vertel eens over je eerste 'echte' elektronica-project. Waar werkte je aan en waarom?

Uri Shaked: Toen ik in de achtste klas zat, hing ik veel rond op IRC-chatservers. Op een dag deelde een vriend een link waarin werd uitgelegd hoe je een luidsprekersysteem kon bouwen van oude harde schijven door de actuatormotor van de drives rechtstreeks op een versterker aan te sluiten. Ik was gefascineerd door het idee en besloot er zelf een te bouwen. De geluidskwaliteit was niet perfect, maar met wat vallen en opstaan ontdekte ik dat een zware pot bovenop de drive de geluidskwaliteit en het volume enorm verbeterde (de pot werkte als een soort resonator). Ik eindigde met een stapel potten in de woonkamer, waar ik de oude Pink Floyd-LP van mijn moeder afspeelde. Onnodig te zeggen dat mijn moeder hier niet blij mee was. In de loop der jaren bleven harde schijven terugkomen als thema in mijn projecten, bijvoorbeeld in mijn eerste



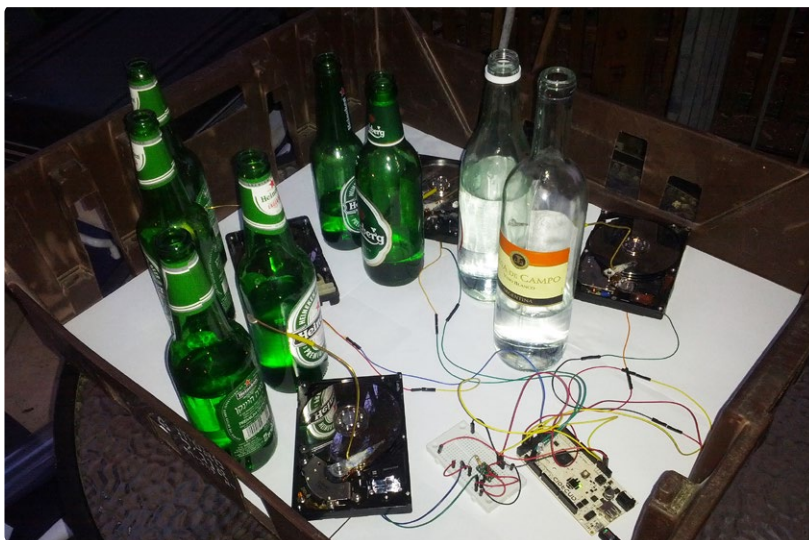
MCU-project, een PIC18 persistence-of-vision display met harde schijf of, zoals je kunt zien in **figuur 2**, een van mijn eerste Arduino projecten, een xylofoon [1] met harde schijf die liedjes afspeelde op gerecyclede bierflesjes.

Elektor: Ingenieurs en programmeurs leren veel van hun fouten. Zijn er projecten die op een ramp zijn uitgelopen? Kun je iets vertellen over dat catastrofale project?

Uri Shaked: Dat is beslist mijn GeekCon-project. GeekCon is een weekendevenement waar een groep nerds samenkomt om nutteloze maar coole dingen te bouwen. Het moet misgaan – het motto is immers “als je project niet is mislukt, heb je niet hoog genoeg gemikt.” In 2018 probeerden we een robot te bouwen die trompet speelt [2]. We gebruikten een met water gevulde latex handschoen om kunstmatige lippen te maken, een pasta-pot als luchtstroomregelaar en hoog belastbare servomotoren met metalen tandwielen om de kunstvingers te bewegen (**figuur 3**). Een minuut voor de eindpresentatie van het project was er een lek en twee van de servomotoren begaven het. Een epische mislukking. Ik houd niet van mislukkingen, dus ik ging terug naar huis en besteedde de volgende maand aan het repareren van de robot en beloofde hem mee te nemen naar Chrome's jaarlijkse evenement voor ontwikkelaars, Chrome Dev Summit. Uiteindelijk kwam ik erachter hoe ik een robuust vingermechanisme kon maken, maar het lipmechanisme was zo lastig dat ik het niet aan de praat kreeg. Uiteindelijk ‘speelde ik vals’ door een Raspberry Pi te gebruiken die vooraf opgenomen trompetvoorbeelden afspeelde via het mondstuk. Maar het eindresultaat klonk tenminste goed en mijn robot amuseerde de deelnemers aan Chrome Dev Summit [3].

Elektor: Iemand zei ooit: “Een opgeruimde werkbank is een teken van een zieke geest.” Durf je ons een foto van je werkplek te laten zien?

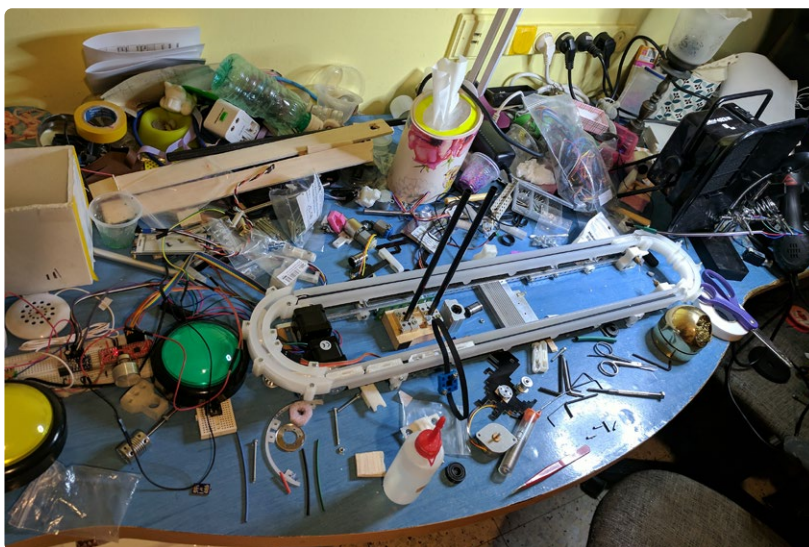
Uri Shaked: **Figuur 4** geeft een indruk van mijn werkplek toen ik probeerde een levensechte versie van het spel ‘Offline Dinosaur’ [4] van Google Chrome te bouwen. Mijn grootste probleem was waar ik alle onderdelen moest bewaren. Ik had meer dan 4.600 dingen, bewaard in 40 verschillende opbergdozen en bijgehouden in verschillende Excel-spreadsheets, die kostbare ruimte in beslag namen in mijn kleine appartement. Vorig jaar verhuisden we en besloot ik om bijna alles weg te geven aan een lokale maker space. Tegenwoordig gebruik ik voornamelijk virtuele onderdelen. Ik heb nog steeds een paar Raspberry Pi's, een Saleae Logic Analyzer en een kleine verzameling MCU-board (ESP32, STM32, Pi Pico en Arduino) waarmee ik het werkelijke gedrag van een chip kan vergelijken met het gesimuleerde gedrag.



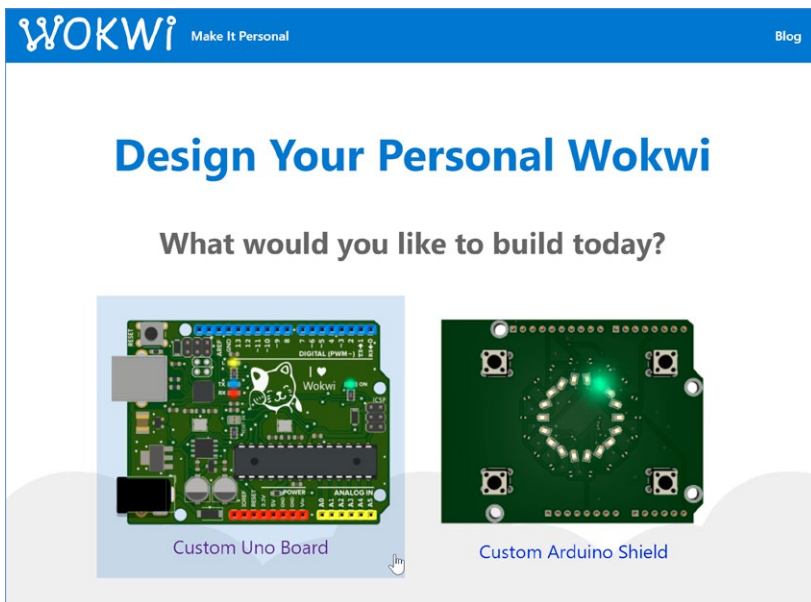
Figuur 2: Hard-disk-xylofoon.



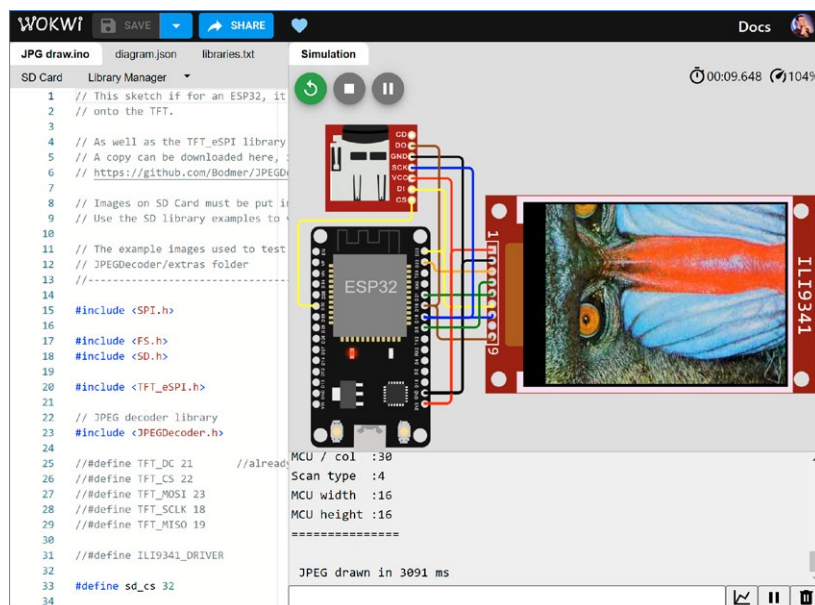
Figuur 3. Mijn vriend Avi met onze trompetrobot, een paar seconden voordat hij explodeerde...



Figuur 4. Mijn opgeruimde en nette werkplek...



Figuur 5. De oude homepage van Wokwi.



Figuur 6. Wokwi op dit moment.

Elektor: Als je beginners een paar ‘tips en trucs’ zou moeten geven voor het inrichten van een goede werkplek, wat zou je dan zeggen?

Uri Shaked: Zorg voor een goede voeding!

Elektor: Oké, op naar Wokwi. Wat is het? Waar komt de naam vandaan? Wat bracht je ertoe om het te maken?

Uri Shaked: Wokwi is een simulator voor embedded systemen en IoT-apparaten. Er is een gratis online-versie en ook een Visual Studio Code-plugin. Wokwi kan ESP32, Raspberry Pi Pico, STM32, Arduino en een heleboel andere microcontrollers simuleren, evenals een grote verscheidenheid aan invoer- en uitvoerapparaten: LC-displays, sensoren, motoren, LED's, knoppen, luidsprekers, potentiometers en meer. Zoals je kunt

zien in figuur 5, is Wokwi begonnen als een service waarmee je iegen Arduino-boards [5] kunt maken met een eenvoudige drag-and-drop interface. Begin met een boardsjabloon, voeg periferie toe (LED's, sensoren enzovoort), plaats ze op het board en wij zetten je ontwerp om in een print, produceren en assembleren die en verzenden het. Ik begon te bloggen over Arduino om de boodschap te verspreiden en ik wilde echt interactieve, live demo's in mijn artikelen opnemen. Ik kon geen goede oplossing vinden voor het gemakkelijk delen van Arduino-projecten en het simuleren ervan op het web, dus begon ik te werken aan een open-source JavaScript-bibliotheek die de ATmega328P-microcontroller kon simuleren. En de rest is geschiedenis. In **figuur 6** kun je zien hoe het er nu uit ziet. Wat de naam Wokwi betreft, was ik op zoek naar een kort, makkelijk uit te spreken woord dat nog geen betekenis had.

Elektor: Wie gebruikt het? Is het helemaal gratis?

Uri Shaked: Ongeveer 55% van de gebruikers zijn hobbyisten, 35% zijn studenten en leraren (zowel universiteiten als middelbare scholen), en 10% zijn professionele firmware-engineers. Enkele populaire use cases zijn het maken van prototypes van IoT-systemen (MQTT, Blynk, Thingsboard, IBM Cloud), online cursussen en workshops (bijvoorbeeld de cursus *Making Embedded Systems*), en we zien een groeiende interesse vanuit de Embedded Rust-gemeenschap. Wokwi heeft een gratis versie en een betaalde versie (*The Club*). De gratis versie is zeer krachtig en heeft dezelfde simulatiemogelijkheden als de betaalde versie. De betaalde versie voegt het uploaden van bestanden en bibliotheken, lokale netwerken voor de ESP32 en de mogelijkheid om privéprojecten (unlisted) op te slaan toe.

Elektor: Welke onderdelen of aspecten van het ontwikkelen en leiding geven aan het Wokwi-project vormden de grootste uitdaging?

Uri Shaked: Wokwi wil complexe systemen vereenvoudigen en een goede gebruikerservaring bieden. In plaats van compilers te moeten downloaden en installeren, drivers voor besturingssystemen te moeten installeren, te moeten vechten met rechten, defecte USB-kabels en gammele breadboardverbindingen, klik je gewoon op een knop. Je code compileert op ‘magische’ wijze en je kunt het resultaat bekijken in je webbrowser. Om zo’n ‘het werkt gewoon’-ervaring te creëren, moeten we veel complexiteit zelf oplossen. We stellen de compiler-toolchain in de cloud in voor meerdere platformen (bijvoorbeeld Rust, Arduino, esp-idf, verilog), zorgen ervoor dat het lekker snel werkt (we draaien ongeveer 2,3 miljoen compilaties per maand) en jagen op mysterieuze bugs. Vorige

maand ben ik bijvoorbeeld een dag bezig geweest om uit te zoeken waarom het gebruik van een specifieke bibliotheek in een Arduino-project ervoor zorgde dat de server bleef hangen. Het bleek dat er een probleem was met de Arduino CLI [6] die ervoor zorgde dat de server bleef hangen wanneer hij die bibliotheek probeerde te installeren. Het bieden van een goede gebruikerservaring betekent ook dat we gebruikers moeten helpen uit te zoeken waarom de code niet werkt zoals ze verwachten. Ik heb het gevoel dat we nog een lange weg te gaan hebben, maar functies zoals de virtuele Logic Analyzer en de onlangs toegevoegde Pin Function Dump-functie zijn zeker een stap vooruit.

Elektor: Kun je ons iets vertellen over je favoriete 'Featured Simulation Project'?

Uri Shaked: Het is lastig een keuze te maken, maar als ik er één zou moeten kiezen zou dat 32 *Servos Dancing* [7] zijn, zoals te zien in **figuur 7** – een Arduino Mega die 32 servo's bestuurt en hun armen animeert. Het project is meer kunst dan nuttig. En dat vind ik zo leuk aan dit project. Om het in het echt te bouwen zou je een stevige voeding nodig hebben en een paar honderd euro voor de onderdelen moeten uitgeven, maar in de simulator is het een fluitje van een cent. Klik gewoon op *Play* en je ziet het in actie. Om de boel wat op te leuken is zelfs het schema voor dit project gegenereerd door de Arduino-code. Kijk eens naar de `GenerateDiagram()`-functie, die trigonometrische functies gebruikt om de servo's te plaatsen. Er wordt een JSON-bestand afgedrukt, dat je vervolgens in *diagram.json* in je Wokwi-project kunt plakken, waardoor je zowel de schakeling als de visuele weergave ervan krijgt. Dat noem ik nog eens innovatieve engineering! Over creativiteit van gebruikers gesproken: Wokwi biedt een *Custom Chips API* [8], waarmee je nieuwe simulatiemodellen kunt coderen voor onderdelen die nog niet bestaan in de simulatie. De API biedt een framebuffer-interface, waarmee je je eigen displays kunt maken (LCD, e-papier enzovoort). Wat ik niet had verwacht is dat gebruikers hun eigen oscilloscoop zouden maken met behulp van de framebuffer-interface, zoals te zien in **figuur 8**.

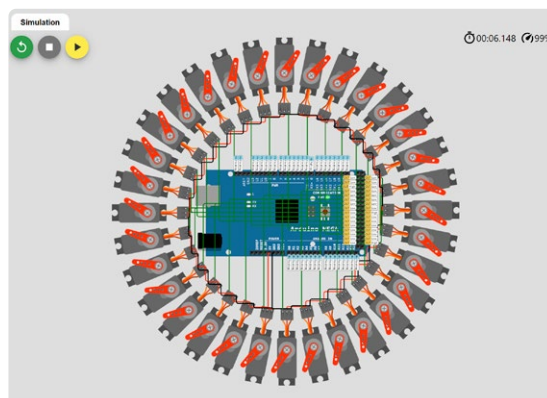
Elektor: De ESP32 is op zichzelf al een behoorlijk complexe module. Vertel ons eens hoe je de ESP32 in je project hebt geïntegreerd. Was dat moeilijk?

Uri Shaked: Wow, dat was een moeilijke. De ESP32 gebruikt de Xtensa-instructieset, die meer dan 250 instructies bevat. De eerste stap was het schrijven van code om elk van die instructies te decoderen en te simuleren, wat uiteindelijk in ongeveer 4500 regels code resulteerde. Daarna maakte ik een GDB-bridge (zodat ik de code kon debuggen die in de simulator draaide) en begon ik te werken aan de

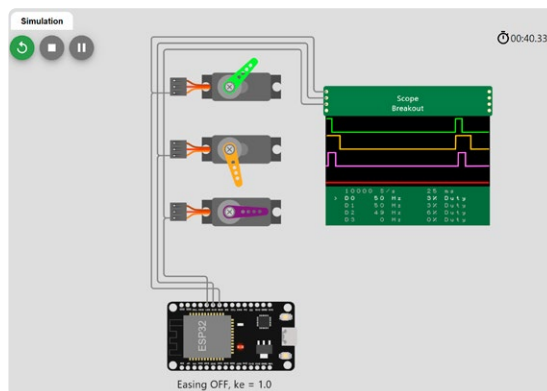
periferie van de chip. Mijn eerste doel was om voorbij de ROM en bootloader te komen en de *Hello World*-applicatie te draaien vanaf ESP-IDF. Na het implementeren van UART, SPI, de timers (TIMG) en wat andere obscure periferie (DPORT, EFUSE), was ik eindelijk in staat om de eerste bruikbare uitvoer van de seriële console van de simulator te krijgen. De grootste uitdaging was echter het simuleren van WiFi. In tegenstelling tot andere delen van de chip, is de WiFi-periferie helemaal niet gedocumenteerd en de broncode van de drivers is niet beschikbaar. Het kostte weken van intensieve reverse engineering [9] om erachter te komen welke registers de WiFi-radio aansturen en hoe deze correct te simuleren. Het harde werk heeft echter zijn vruchten afgeworpen – de ESP32 is de populairste chip op Wokwi aan het worden en Espressif, het bedrijf achter de ESP32, steunt het project van harte. Hun ingenieurs zetten vaart achter de ondersteuning van Rust, gebruiken Wokwi in hun trainingen en workshops, en bevelen het zelfs aan op conferenties voor professionals.

Elektor: Vertel ons eens over de reactie van gebruikers op Wokwi. Wat doe je met hun feedback?

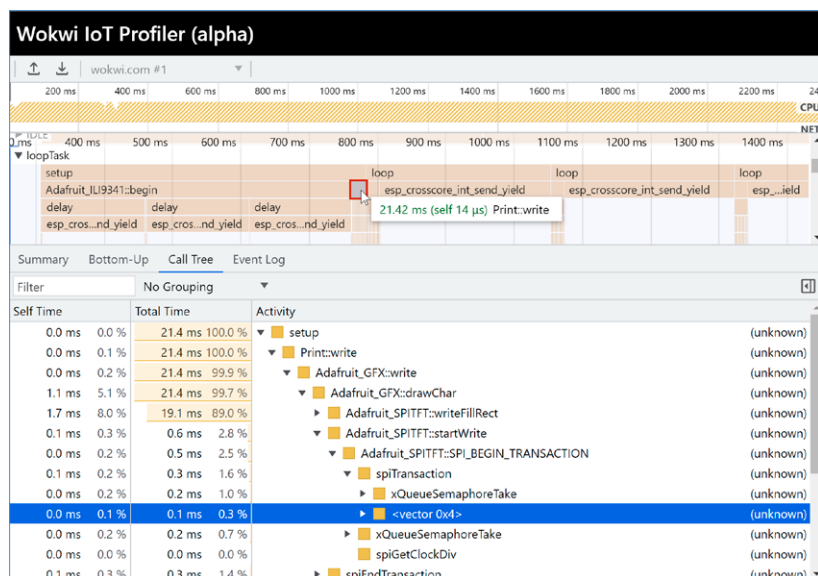
Uri Shaked: Onze gebruikers zijn over het algemeen blij met de simulator, die ze veel tijd bespaart. Maar ze vragen ook altijd om meer en meer functies, veel meer



Figuur 7. 32 dansende servo's, aangestuurd door een Arduino Mega-board.



Figuur 8. De Scope-chip toont servosignalen in real time.



Figuur 9. Wokwi Embedded Profiler.



Figuur 10. De Floppy Disks, met een close-up van de elektronica (twee EEPROM's, write-lock soldeerjumpers en een indicatie-LED).

dan ons kleine team aankan. Om te helpen focussen op wat echt belangrijk is voor gebruikers, heb ik de 'Vote for new features' -pagina [10] geopend. Iedereen kan nieuwe functies voorstellen en mensen kunnen met hun geld stemmen op functies die voor hen belangrijk zijn. Het helpt ons prioriteiten te stellen en het is ook een geweldige manier voor gebruikers om te zien wat er aan zit te komen.

Elektor: Waar wil je Wokwi hebben in de komende zes tot twaalf maanden?

Uri Shaked: Ik zou graag zien dat Wokwi meer aandacht krijgt van professionele gebruikers. Eerder dit jaar heb ik Wokwi uitgebracht voor Visual Studio Code. Hiermee kun je de simulatie direct in je IDE uitvoeren en je code veel sneller itereren. Ik zie nu al dat Rust-ontwikkelaars het gebruiken om hun productiviteit te verhogen. Wokwi kan ook integreren met de debugger in VS Code en, in tegenstelling tot echte hardware, kun je met Wokwi een willekeurig aantal breakpoints instellen. Ik werk nu aan een nieuwe embedded profiler – zie **figuur 9**. Wokwi traceert

je code terwijl die draait in de simulator. Je kunt zien welke functies worden aangeroepen, in welke volgorde, en hoeveel tijd de microcontroller in elke functie doorbrengt. Ik ben ook van plan om de ESP32-lijn uit te breiden (met ESP32-H2, ESP32-P4), nieuwe STM32-chips toe te voegen en misschien contact op te nemen met andere chipmakers en ook hun microcontrollers toe te voegen. Tot slot zou ik graag zien dat Wokwi wordt gebruikt in CI-omgevingen (continuous integration). Het idee is om je embedded code bij elke commit te testen. Dit is veel eenvoudiger op te zetten en op te schalen met Wokwi, vergeleken met het onderhouden van een fysieke hardware-setup. Ik gebruik intern al een CI-setup om er zeker van te zijn dat kritieke componenten, zoals de ESP32 WiFi, niet defect raken bij updates van de code en nieuwe ESP-IDF releases, en werk nu aan een CLI (command-line interface) voor Wokwi, om de simulator gemakkelijker te kunnen gebruiken binnen GitHub-acties.

Elektor: Je lijkt zowel een kunstenaar als een ingenieur te zijn, zoals velen kunnen zien bij het lezen van je post 'A Practical Guide to Designing PCB Art' [11]. Vertel ons eens over je creatieve benadering van kunst en elektronica.

Uri Shaked: Samen met vrienden planden we een slimme naambadge voor een conferentie. Na het bespreken van hardware add-ons voor de badge, zei iemand dat het ook cool zou zijn als de deelnemers kleine spelletjes voor de badge zouden kunnen schrijven en een soort app-store zouden hebben om hun creaties te delen. Op dat moment kwam er een idee in me op. We hebben geen online app-store nodig – in plaats daarvan kunnen we een hardware add-on maken met een kleine EEPROM-chip. De deelnemers schrijven hun apps en games op de add-on en geven deze vervolgens weg aan hun vrienden, zodat zij de code kunnen kopiëren naar hun badge. Zo werd de Floppy Disk add-on [12] geboren. Toen ik eenmaal het idee had, was het een kwestie van de afmetingen van een floppy disk opzoeken, een schets maken met Inkscape en beslissen welke PCB-layer te gebruiken voor elk element: zeefdruk voor de sticker, vertind koper (HASL) voor de metalen schuif en koper met een soldeermasker voor het HD-symbool. Ik heb het schema en de layout in KiCad gedaan, en het resultaat – wel, **figuur 10** spreekt voor zich.

Elektor: Staan er nog andere belangrijke projecten op stapel?

Uri Shaked: Tiny Tapeout. Ik werk hier samen met Matt Venn aan en ons doel is om silicium op maat betaalbaarder en toegankelijker te maken. Je kunt een digitaal ontwerp in Verilog of Wokwi gebruiken om een ontwerp te maken van individuele logische poorten, vervolgens je ontwerp 'hard maken' met

OpenLane (een open -source digitale ontwerp-workflow) en je ontwerp indienen voor opname in onze chip. Wij nemen je ontwerp, samen met honderden ontwerpen van anderen, en combineren alles tot een enkel chipontwerp. Vervolgens produceren we de chips, zetten ze op een PCB en sturen je een fysieke versie van je ontwerp. De print die je ziet in **figuur 11** heeft een set DIP-schakelaars om een ontwerp te selecteren, zodat je ook kunt spelen met de andere ontwerpen in de chip. We moedigen dat zelfs aan – alle ontwerpen zijn open source en we stellen een datasheet samen met alle documentatie. We beginnen nu samen te werken met universiteiten om Tiny Tapeout naar hun studenten te brengen en lesmateriaal te maken om hobbyisten en silicium-enthousiastelingen te helpen leren over halfgeleiders en digitaal ontwerpen. Eerder dit jaar hebben we Siliwiz uitgebracht, een open source-app die een visuele layout-editor, DRC-engine en SPICE-simula-

tie integreert, allemaal in een browser. Er is nog veel meer te vertellen over Tiny Tapeout, de gemeenschap en de creatieve ontwerpen die onze gebruikers samenstellen, maar dat is een verhaal voor een andere keer!

vertaling: Jelle Aarnoudse – 230366-03



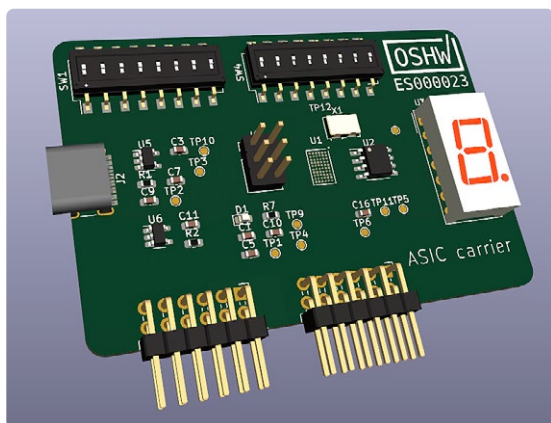
Over Uri Shaked

Uri Shaked is al heel lang een maker. Hij werkt momenteel aan Wokwi, een online simulatieplatform voor IoT en embedded systemen, en aan Tiny Tapeout [13], dat 'eigen' ASIC's betaalbaar en toegankelijk maakt. Zijn projecten en blog zijn te vinden op [14], met veel tech talks en video-interviews [15].



Over de auteur

Roberto Armani is een elektrotechnisch ingenieur met meer dan vijfendertig jaar ervaring in verschillende sectoren. Voordat hij als Senior Editor het Elektor-team kwam versterken, deed hij kennis en ervaring op in de computerindustrie, elektronische beeldvorming, telecommunicatie, materiaaltestapparatuur en webpublicatie. Naast elektronica hoort (en zingt) hij graag klassieke muziek en maakt hij 'hoge' wandelingen in de bergen.



Figuur 11. Tiny Tapeout breakout-print (voorlopige 3D-weergave).

WEBLINKS

- [1] Hard-Drive Xylophone Arduino-project: <https://youtu.be/dw9U0WxtK9c>
- [2] Experimenten met een gerobotiseerde trompet: <https://bit.ly/3N2FfCz>
- [3] Trompet-demo op de Chrome Dev Summit: <https://youtu.be/PEVAczB9uUQ>
- [4] Elektronica met een persoonlijk tintje: <https://blog.wokwi.com/wokwi-electronics-with-a-personal-touch/>
- [5] Real-life versie van de "Offline Dinosaur Game" van Google Chrome: <https://bit.ly/3qgsifz>
- [6] CLI-problemen met Arduino: <https://github.com/arduino/arduino-cli/issues/2135>
- [7] Dancing Servos Project: <https://wokwi.com/projects/305336312628511297>
- [8] Custom Chips API: <https://docs.wokwi.com/chips-api/getting-started>
- [9] "Reversing the ESP32 WiFi" video: <https://youtu.be/XmaT8bMssyQ>
- [10] Wowki's Feature Voting Page: <https://wokwi.com/features>
- [11] A Practical Guide to Designing PCB Art: <https://blog.wokwi.com/a-practical-guide-to-designing-pcb-art/>
- [12] Floppy Disk add-on: <https://github.com/urish/floppy-disk-sao>
- [13] Tiny Tapeout: <https://tinytapeout.com>
- [14] Webpagina met Uri's projecten: <https://urish.org/#projects>
- [15] Uri's video's en interviews: <https://urish.org/#talks>