

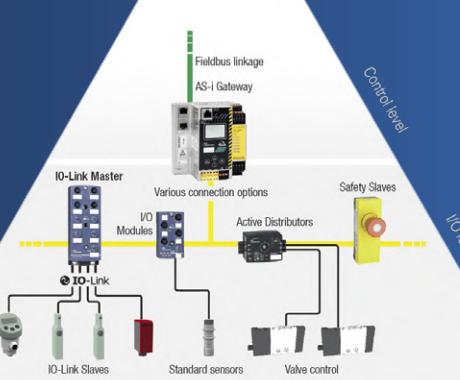
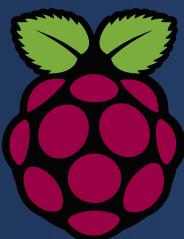
Zeskanaals temperatuurmonitor en -logger



meet van -240°C tot $+850^{\circ}\text{C}$



Snips
spraakherkenning
met de
Raspberry Pi



Basiskennis AS-Interface
zo werkt het bussysteem
voor de industriële automatisering

Zeskanaals temperatuurmonitor en -logger ▷ Spraakassistent met Snips ▷ MIDI I/O Break-out-Board ▷ De cryptocurrency IOTA – PiDiver ▷ Accumonitor ▷ Waterverbruik-monitor ▷ Felle knipper-LED ▷ Baristors – bouwstenen voor voedingsschakelingen ▷ ESP32 als tijdserver ▷ Retrotronica: van buizen naar silicium ▷ Het SCCC-project (4) ▷ Basis kennis AS-Interface ▷ Qt voor de Raspberry Pi ▷ Communicatieve stroomlussen ▷ Symbolen maken in KiCad ▷ en meer...



productronica
fast forward
the start-up platform

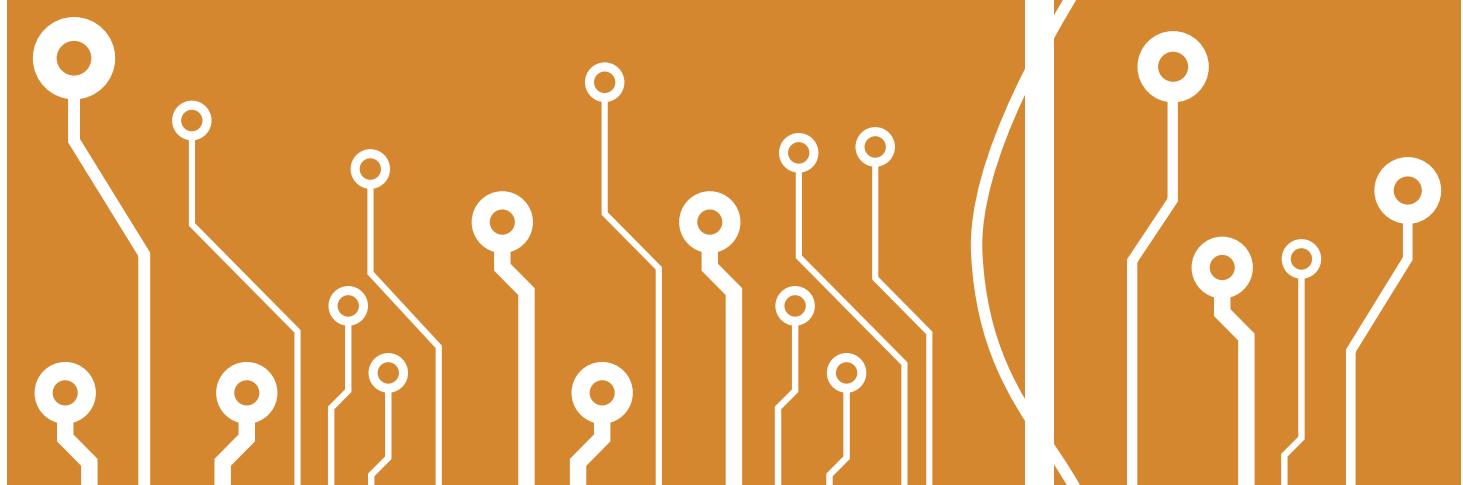
powered by Elektor

COMPETE TO
LAUNCH YOUR STARTUP
@
PRODUCTRONICA 2019

● p-ffwd 2019 – Doe mee!

12 - 15 november 2019

Messe München



Meer info:

www.elektormagazine.nl/p-ffwd

Platinum Sponsor:
DISTRELEC
Distribution with a difference

Productronica Fast Forward is brought to you by

 productronica

 Elektor
INNOVATION > STARTUP > TRADE

Colofon

59^e jaargang nr. 654,
juli/augustus 2019
ISSN 2590-0765



Elektorlabs is een uitgave van

Elektor International Media B.V.

Postbus 11, 6114 ZG Susteren, Nederland
Tel.: +31 (0)46- 4389444,
Fax: +31 (0)46-4370161

Nieuwe abonnementen & bestellingen

service@elektor.nl Tel. 046-4389444

Elektor International Media B.V. legt gegevens vast voor de uitvoering van de (abonnementen) 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: Pijper Media, Groningen
Distributie: Betapress, Gilze



...als een stekker

In de schaduw van een palmboom op een van de Griekse eilanden, genietend van een (hopelijk welverdiende) vakantie en met een lekker (Grieks) biertje onder handbereik – is er een betere plek denkbaar om het voorwoord voor het juli/augustusnummer van Elektorlabs te schrijven?

Een van de dingen die we op vakantie steevast doen is een kaartje kopen voor zo'n City-Tour hop-on hop-off bus, zoals die inmiddels in vrijwel alle grotere steden rondrijden. Duur zijn die kaartjes niet, en meestal zijn ze twee dagen onbeperkt geldig. En dat is dan een prima manier om een onbekende stad te verkennen.

Wat al die bussen gemeen hebben is gesproken tekst en uitleg over de bezienswaardigheden, in een groot aantal talen. Die kun je genieten via een koptelefoon die met een 2,5-mm TRS-stekker (jackplug, klinkstekker) in een paneeltje in de rugleuning van de stoel voor je wordt geprikt. En dan begint de pret want die stekkers zijn – na jarenlang door horden toeristen te zijn mishandeld – zo ongelooflijk gammel dat het een kwestie van puur geluk is of je überhaupt iets hoort... (Oplossing: als er niet te veel mensen in de bus zitten, kun je net zo lang zoeken tot je een zitplaats hebt gevonden die 'het doet'.)

Ik moest daaraan denken omdat we in dit nummer van Elektorlabs een MIDI-break-out-board beschrijven dat al min of meer is voorbereid op de nieuwe MIDI 2.0-standaard. En wat schrijft die standaard als aanbevolen connector voor? Inderdaad: 2,5-mm TRS-stekkers... (maar dan natuurlijk van professionele kwaliteit!)

Dat is natuurlijk niet het enige waarop u zich in dit nummer kunt verheugen: voor elk wat wils en genoeg om de zomermaanden mee door te komen. Bijzonder aanbevolen: de 6-kanaals temperatuurmonitor/logger op bladzijde 6, maar niet te versmaden zijn ook de beide artikelen over de SNIPS-spraakassistent.

Eric Bogers

Hoofdredacteur NL

Advertenties Benelux

Margriet Debeij
Tel. +49 2 41/955 09 174
Mobiel: +49 (170) 550 539 6
E-Mail: margriet.debeij@eimworld.com

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

Auteursrecht

Niets uit deze uitgave mag verveelvoudigd 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 Rijksoctrooiwet 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.

Ons team

Hoofdredacteur:

Eric Bogers (eric.bogers@elektor.com)

Internationale redactie:

Jan Buiting, Jens Nickel, Mariline Thiebaut-Brodier

Technische redactie:

Mathias Claussen, Ton Giesberts, Luc Lemmens, Denis Meyer, Clemens Valens, Jan Visser

Aan dit nummer werkten mee:

Jelle Aarnoudse, Rolf Blijlevens, Kees de Groot, Martien Jansen, Evelien Snel

Ledenmanagement:

Denise Bodrone

Vormgeving & Layout:

Giel Dols, Jack Jamar

Hoofd Online:

Daniëlle Mertens

Directeur:

Don Akkermans

in dit nummer

59^e jaargang – nr. 654

juli/augustus 2019

- 34 Studenten in de schijnwerpers!
TUE/InMotion: revolutionair accupack-ontwerp
- 42 Tips en trucs
voeding voor differentiële probes via de USB-poort
- 50 Elektor start-up challenge Parijs 2019
internationaal lanceerplatform voor start-ups
- 52 Vreemde onderdelen
programmeerbare precisiereferentie TL431
- 69 Elektor Labs Pipeline
- 74 V & A
analoge schakelingen
- 82 Project 2.0
correcties, updates en brieven van lezers
- 84 Shop-highlight
de inrichting van uw hobbylab...
- 92 HomeLab Helicopter
de wonderlijke wereld der elektronica in vogelvlucht
- 106 Retrotronica
van buizen naar silicium
- 112 Elektor Store
- 114 Hexadoku

achtergrond & info

- 22 Snips
spraakherkenning met de Raspberry Pi
- 30 Een nieuwe component (symbool) in KiCad maken
en voorbereiden voor koppeling aan een bestaande footprint
- 43 Seriële data weergeven op een webpagina
met PHP- of Python-scripts
- 48 Qt voor Raspberry Pi
de inrichting van het GUI-framework – stap voor stap
- 53 Communicatieve stroomlussen
signaaloverdracht met stroom
- 70 Basiskennis AS-Interface
zo werkt het bussysteem voor de industriële automatisering
- 76 Digitale modulatietechnieken:
coderen van gegevens
NRZ, Manchester en consorten

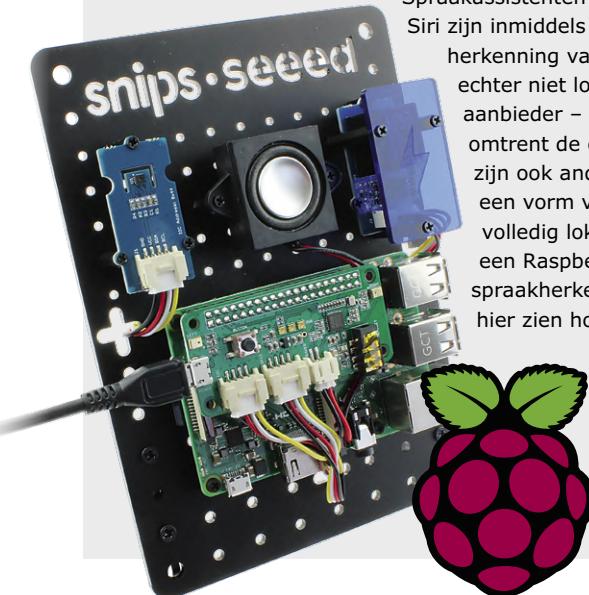
Zeskanaals temperatuur en -logger

meet van
-240 °C tot +850 °C

Snips

spraakherkenning met de Raspberry Pi

Spraakassistenten zoals Alexa, Google Home en Siri zijn inmiddels heel gewoon. De eigenlijke herkenning van gesproken woorden gebeurt echter niet lokaal, maar in de cloud van de aanbieder – wat natuurlijk vragen oproept om trent de gegevensbescherming. Maar er zijn ook andere oplossingen zoals Snips, een vorm van spraakherkenning die volledig lokaal plaatsvindt op bijvoorbeeld een Raspberry Pi. Wij hebben zo'n spraakherkennings-kit bekeken en laten hier zien hoe daarop eigen applicaties kunnen worden ontwikkeld.



22

projecten

- 6 Zeskanaals temperatuurmonitor en -logger
meet van -240 °C tot +850 °C
- 14 Bouw uw privé-spraakassistent met Snips
kunstmatige intelligentie voor de Raspberry Pi

106
Retrotronica
van buizen
naar silicium

monitor

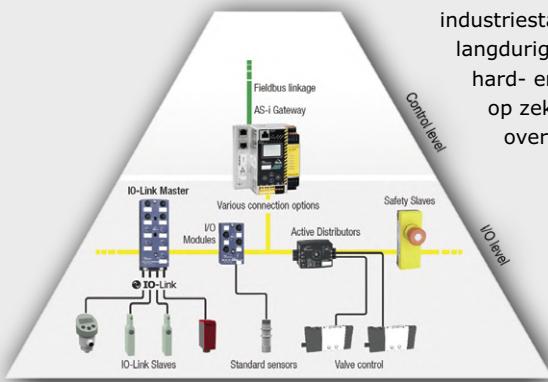


Basiskennis AS-Interface

zo werkt het bussysteem voor de industriële automatisering

Ooit werd alles aangestuurd met zelfgemaakte bedrading en merkgebonden dataformaten. Maar deze manier van werken is in de industriële automatisering al lang niet meer de gewoonte. In plaats daarvan wordt maximaal ingezet op volwaardige bussystemen. Daar is, naast de duidelijke voordelen van eenvoudig onderhoud en uitbreidbaarheid, nog een andere

reden voor: een breed geaccepteerde industrietandard garandeert de langdurige beschikbaarheid van hard- en software, in elk geval tot op zekere hoogte. We geven een overzicht van de AS-interface.



70

- 26 **MIDI I/O Break-out-Board**
met ondersteuning voor DIN- en TRS-connectoren
- 38 **Accumonitor met ESP32**
drie kanalen om te meten
- 56 **Waterverbruik monitoren met een ESP32**
met SMS-alarm bij lekkage



- 62 **De cryptocurrency IOTA (en een FPGA voor de RPi)**
deel 2: PiDiver –
FPGA-board voor snelle berekeningen
- 81 **Felle knipper-LED**
ouderwets analoog...
- 86 **AC-in, DC-uit baristors**
bouwstenen voor voedingsschakelingen
- 96 **ESP32 als tijdserver**
ESP32 + RTC + GPS + display =
eigen NTP-server in uw LAN
- 101 **Het SCCC-project (3)**
softcore-processor en C-compiler
voor zelfbouw



Elektorlabs magazine september/oktober 2019

Qi-laders • horticultuur-box • hardware-ontwerp met (V)HDL (4) • RFID reader/writer • het SCCC-project (4)
• nieuwe CPLD-print met MAX10 • 8-kanaals timer met OLED-display • softstart voor versterkers • uitbreidingen voor netuitvaldetector • tijdcorrecte basweergave
• multi-node datalogger • retrotronica • Sound Blink – een draagbare luidspreker
...en nog veel meer.

Aankondigingen onder voorbehoud.
Elektorlabs september/oktober 2019 verschijnt op 22 augustus 2019.

Elektor Industry 3/2019

Elektor Industry 3/2019 is een speciale uitgave gewijd aan Productronica 2019 in München (Duitsland) van 12 tot en met 15 november 2019. Speciale aandacht wordt besteed aan innovatieve technologieën, onderzoek, hardware en software – inclusief IoT, 5G, sensoren, CAD, SMT, PCB-productie en meer. Plus natuurlijk nieuwe afleveringen van de vaste rubrieken als Infographics en Industry Store.

Elektor Industry 3/2019 verschijnt begin november 2019 in druk voor Elektor Gold-leden en als pdf-download voor Elektor Green-leden.
Het nummer is voor belangstellenden ook te koop via www.elektormagazine.com. Aankondigingen onder voorbehoud.

Zes kanaals temperatuurmonitor en -logger

meet van -240 °C tot +850 °C

Clemens Valens (Elektor Labs) en **Sunil Malekar** (India)

Temperatuur is in allerlei industriële processen een belangrijke parameter die geregeld moet worden. Ook thuis meten de koelkast, de vriezer, de verwarming, de oven en de airconditioning allemaal de temperatuur om te doen wat ze moeten doen. Weerstations, auto's, computers, tablets en smartphones houden ook temperatuur in de gaten. Nu ik er zo over nadenk is temperatuur misschien wel de meest gemeten grootheid ter wereld.

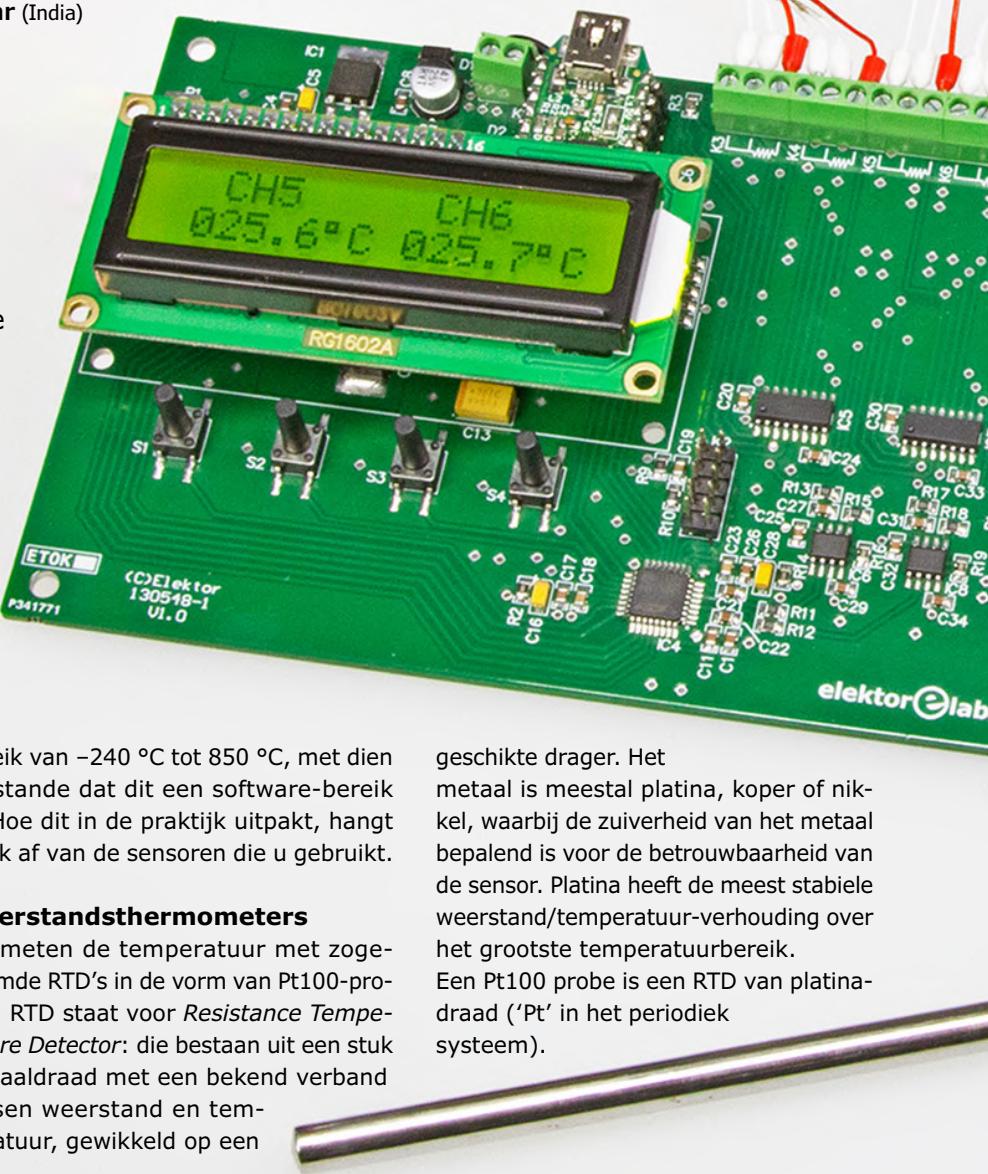
Zelfs mijn goedkope Bluetooth-speaker geeft een alarm als hij te warm wordt (daar kwam ik achter toen ik hem op de verwarming had gezet). Omdat de mensheid zo graag de temperatuur meet, besloten we een schakeling te ontwerpen die zes temperaturen kan meten. Meestal meten thermometers maar één temperatuur, sommige twee ('binnen' en 'buiten'), maar voor procesbesturing zijn er meestal meer nodig. Het meetbereik is ook belangrijk, daarom heeft de schakeling dat we hier presenteren een enorm

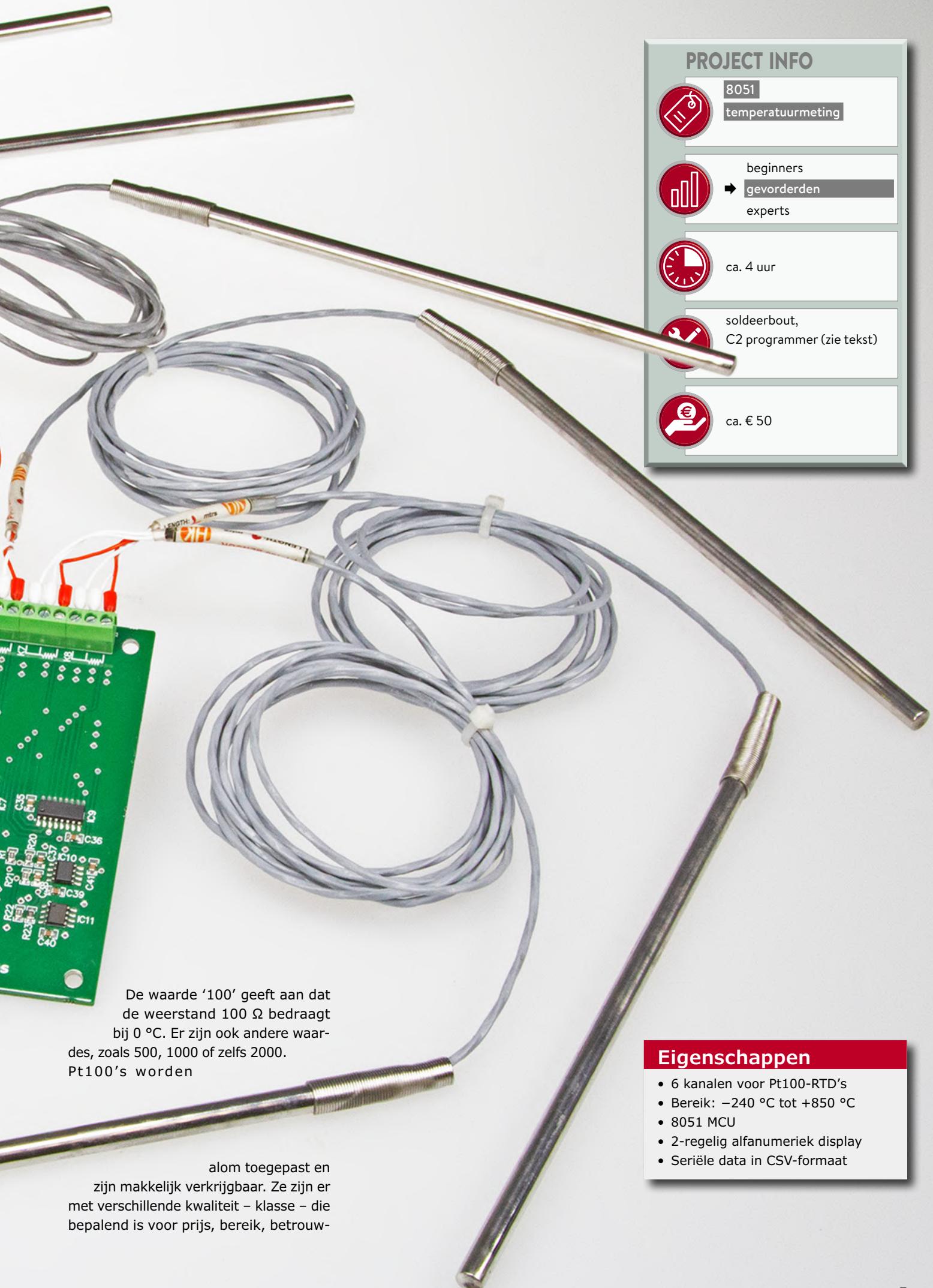
bereik van -240 °C tot 850 °C, met dien verstande dat dit een software-bereik is. Hoe dit in de praktijk uitpakt, hangt sterk af van de sensoren die u gebruikt.

Weerstandsthermometers

We meten de temperatuur met zogenaamde RTD's in de vorm van Pt100-probes. RTD staat voor *Resistance Temperature Detector*: die bestaan uit een stuk metaaldraad met een bekend verband tussen weerstand en temperatuur, gewikkeld op een

geschikte drager. Het metaal is meestal platina, koper of nikkel, waarbij de zuiverheid van het metaal bepalend is voor de betrouwbaarheid van de sensor. Platina heeft de meest stabiele weerstand/temperatuur-verhouding over het grootste temperatuurbereik. Een Pt100 probe is een RTD van platina-draad ('Pt' in het periodiek systeem).



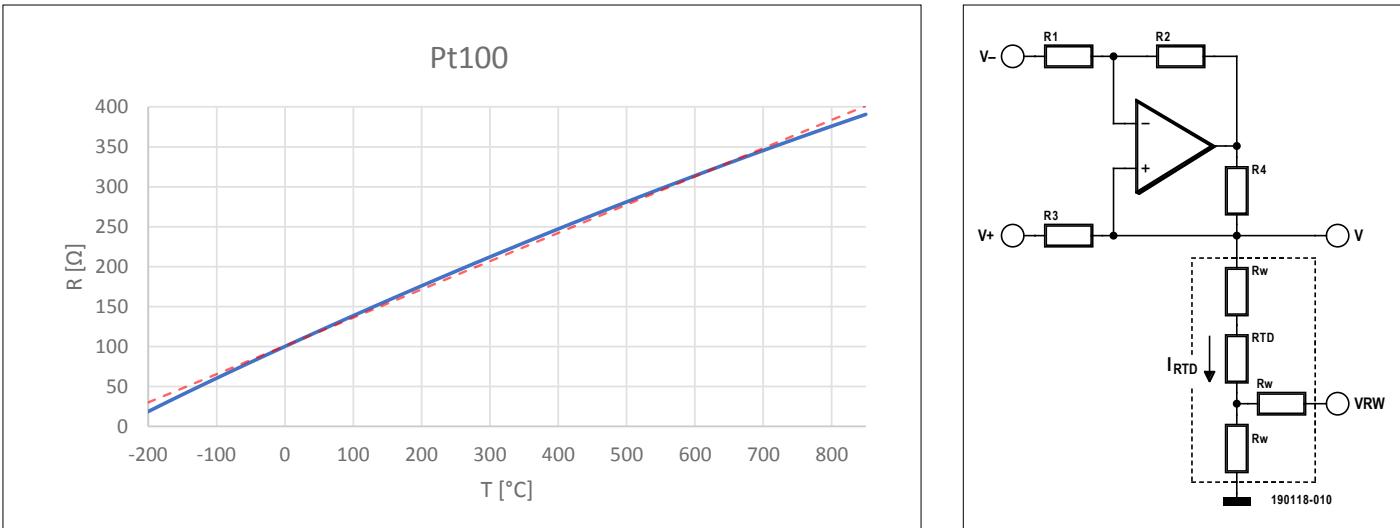


De waarde '100' geeft aan dat de weerstand 100 Ω bedraagt bij 0 °C. Er zijn ook andere waarden, zoals 500, 1000 of zelfs 2000. Pt100's worden

alom toegepast en zijn makkelijk verkrijgbaar. Ze zijn er met verschillende kwaliteit – klasse – die bepalend is voor prijs, bereik, betrouw-

Eigenschappen

- 6 kanalen voor Pt100-RTD's
- Bereik: -240 °C tot +850 °C
- 8051 MCU
- 2-regelig alfanumeriek display
- Seriële data in CSV-formaat



Figuur 1: De verhouding tussen weerstand en temperatuur van een platinum draad is niet lineair. We kunnen deze verhouding benaderen met een opzoektabel van korte 'rechte' segmenten.

Figuur 2: De Howland-stroombron heeft een heel hoge uitgangsimpedantie. We gebruiken hem om een 3-draads RTD aan te sturen.

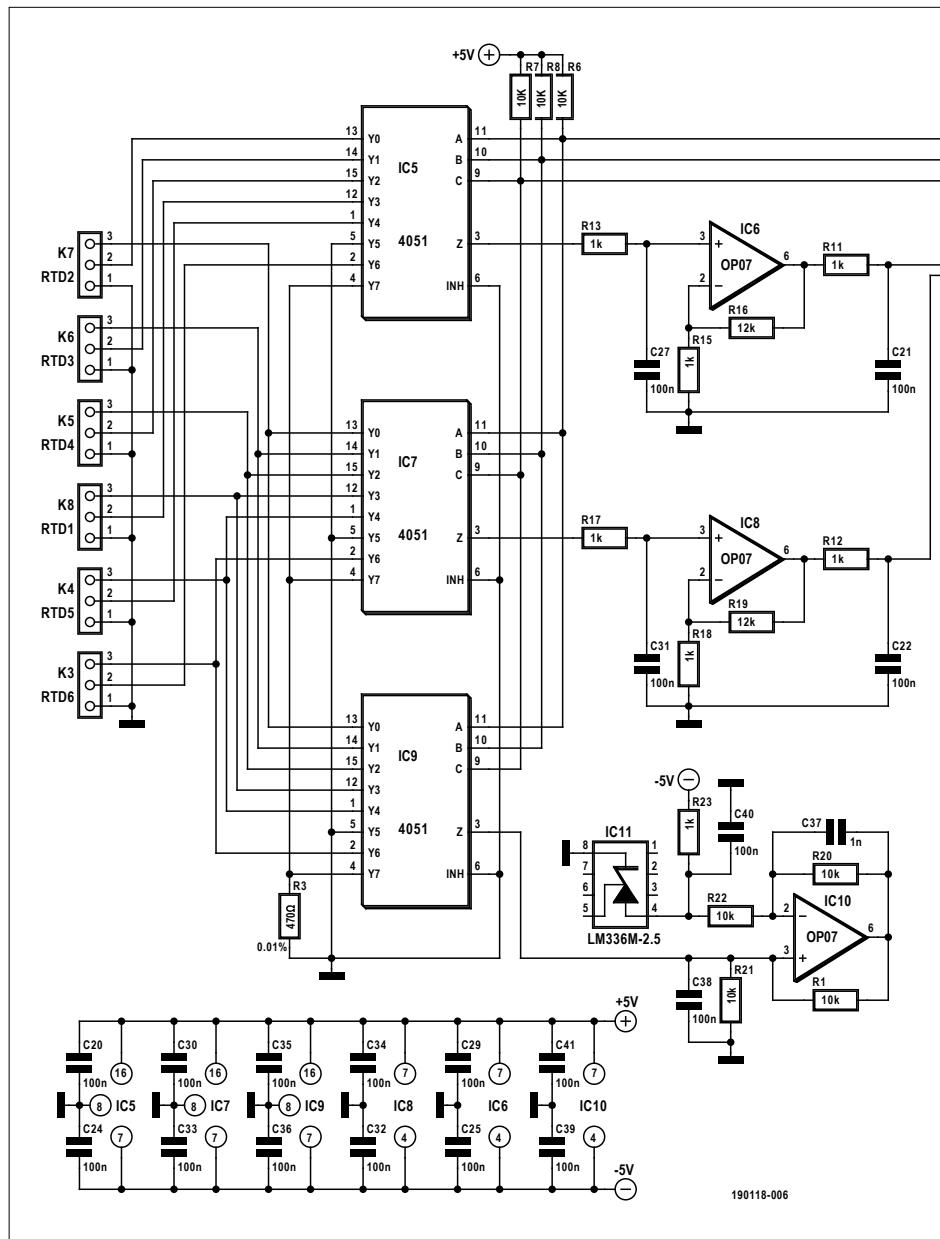
baarheid en nauwkeurigheid. Ze kunnen verschillende aantal aansluitdraden hebben: 2, 3 of 4. De extra draden dienen om de weerstand van de aansluitdraden te meten (hierover straks meer). De bedrading – die best lang kan zijn – gedraagt zich als onbetrouwbare RTD's, daarom is het belangrijk dat we, om een betrouwbare, reproduceerbare meting te krijgen, daarvoor compenseren. Om ons systeem niet te ingewikkeld te maken, hebben we gekozen voor 3-draads RTD's.

Een RTD is geen thermokoppel

Dit moet u beseffen: een thermokoppel bestaat uit een junctie van twee verschillende metalen die ten gevolge van het thermo-elektrisch (Seebeck-) effect een temperatuur-afhankelijke spanning produceert. Een thermokoppel heeft daarom geen voeding nodig om te functioneren, terwijl een RTD altijd een zekere spanning of stroom nodig heeft. Thermokoppels zijn minder nauwkeurig dan RTD's, maar zijn daarentegen heel goedkoop.

Lineariteit

Tenslotte is nog belangrijk om te vermelden dat de weerstand/temperatuur-verhouding van RTD's niet helemaal lineair is (zie **figuur 1**). Dat betekent dat we berekeningen moeten doen om de juiste temperatuur uit de gemeten weerstand af te leiden. Voor platina RTD's is lange tijd de vergelijking van Callendar-Van Dusen gebruikt, maar in 1990 heeft het Comité International des Poids et Mesures die vervangen door een 12de-orde polynoom die geldt van $-259,35^{\circ}\text{C}$ tot 0°C , en een



ander (9de-orde) polynoom voor temperaturen van 0 °C tot 961,78 °C. De uitwerking van die polynomen vergt een heleboel rekenkracht (en debugging), daarom gebruiken de meeste applicaties opzoektabellen. Dat hebben wij ook gedaan.

Howland-stroombron

Zoals we al opmerkten functioneert een RTD niet 'vanzelf'. We kunnen een constante spanningsbron of dito stroombron gebruiken; onze voorkeur ging uit naar een constante stroombron omdat die minder problemen met lange bedrading geeft. In dat geval hangt de spanning over de sensor uitsluitend af van de stroom en natuurlijk van de temperatuur, en niet ook nog van de weerstand van de aansluitdraden. Er zijn allerlei ontwerpen voor een constante stroombron, maar voor gebruik met

RTD's is met name de Howland-stroombron of stroompomp (**figuur 2**) heel populair. Dit soort stroombron heeft het voordeel van een zeer hoge uitgangsimpedantie vergeleken bij andere uitvoeringen, en hij kan fungeren als stroombron (*source*) en als (actieve) stroomafvoer (*sink*). Om hem echter te laten werken zoals bedoeld moeten de weerstanden R1 t/m R4 rond de opamp een tolerantie van 1% of beter hebben.

Is het quotiënt R1/R2 gelijk aan R3/R4, dan geldt voor de uitgangsstroom:

$$I_{\text{RTD}} = (V^+ - V^-) / R3 \text{ [A]}$$

Compensatie voor de aansluitdraden

Figuur 2 laat ook zien hoe de RTD met zijn drie draden aan de stroombron is

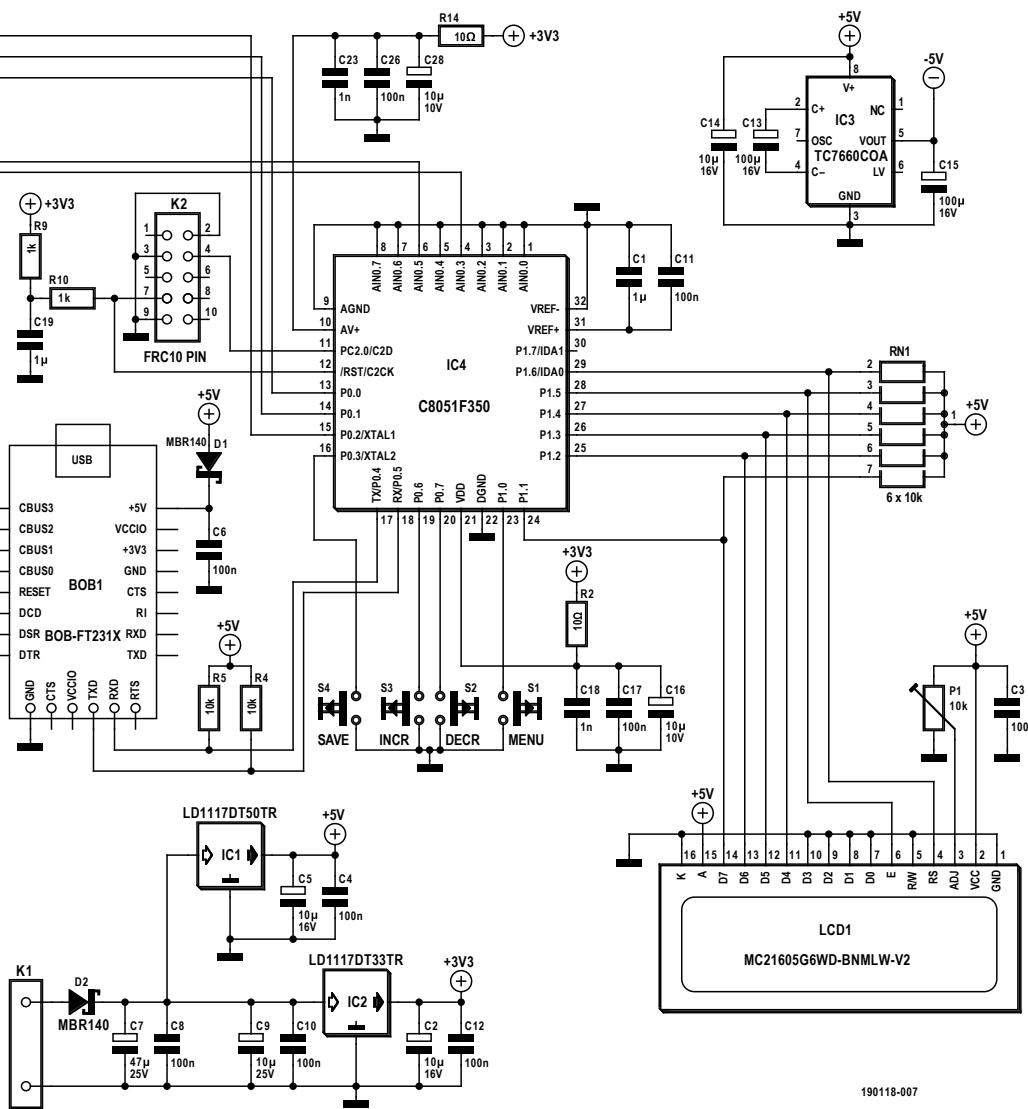
aangesloten. Aangenomen dat de drie draden identiek zijn en dat we de spanningen V en V_{RW} met een zeer hoge ingangsimpedantie meten (waarbij er geen stroom vloeit in de richting van V en V_{RW} (lees: V_{RW})), dan geldt voor de spanning over de RTD:

$$V_{\text{RTD}} = V - 2 V_{\text{RW}} \text{ [V]}$$

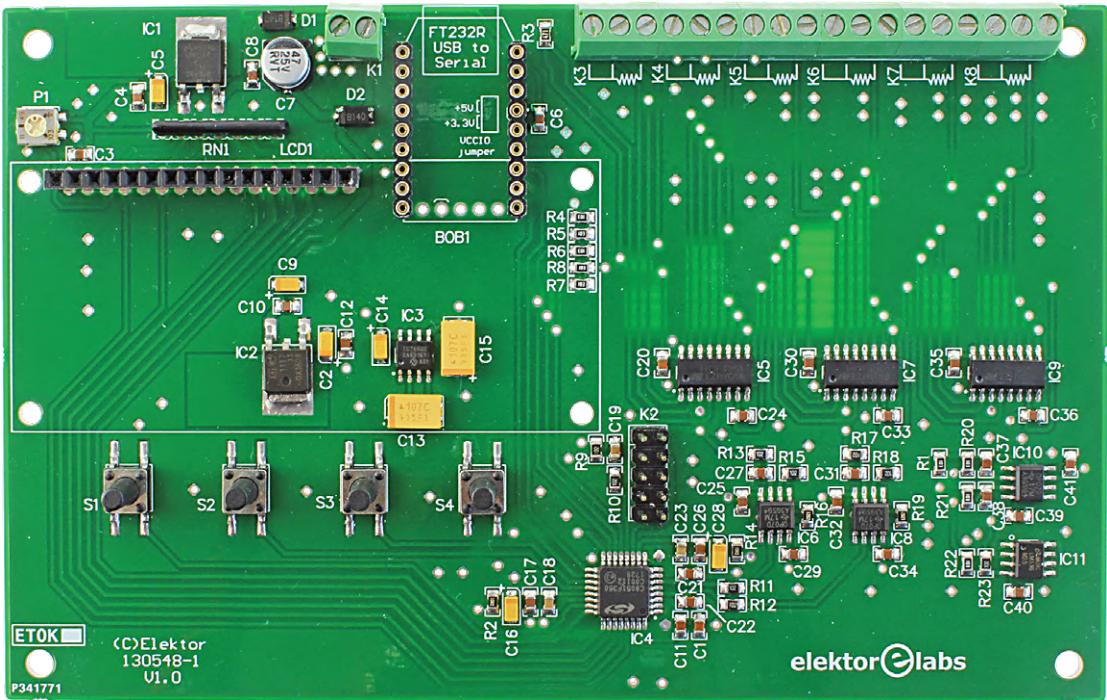
Hiermee is de theoretische inleiding voltooid en kunnen we over tot het bestuderen van de eigenlijke schakeling.

Ingangsmultiplexer

Links in **figuur 3** vinden we de connectoren K3...K8 voor de zes RTD-kanalen. Van elke RTD is één aansluiting verbonden met GND, de andere twee gaan naar een CMOS-multiplexer van het type



Figuur 3: Schema van de meerkanaals temperatuurlogger, met links de analoge ingangstrap en rechts het digitale gedeelte.



Figuur 4: De opgebouwde hoofdprint. Dit is versie 1.0 met de 'gouwe ouwe' FT232R BoB in plaats van de FT231X BoB.

4051. IC7 stuurt het signaal V in figuur 2 door naar de rest van het systeem; IC5 doet hetzelfde met VRW. De derde multiplexer, IC9, werkt andersom en stuurt de uitgang van de stroombron – rond IC10 – naar de geselecteerde RTD.

Elke multiplexer heeft acht kanalen, waarvan er twee worden gebruikt voor de referentiesignalen V_{\min} (GND) en V_{\max} (V_{R3}). Nu zult u eindelijk begrijpen waarom er zes RTD-kanalen zijn en niet zeven of vijf. Deze referentiesignalen volgen hetzelfde pad als de RTD-signalen (R3, een weerstand met een tolerantie van 0,01% is zelfs dichtbij de RTC-connectoren geplaatst om het gedrag van een RTC zo goed mogelijk na te boot-sen). Omdat de waardes van deze signalen bekend zijn, kan de software de digitaliseringstrap iken en fouten, geïntroduceerd door de multiplexer en de stroombron, elimineren.

De uitgangen van de ingangsmultiplexer worden individueel versterkt met een factor 13 door IC6 en IC8. Het signaal V van figuur 2 belandt vervolgens bij de ADC-ingang van de microcontroller (AIN0.3), terwijl VRW is verbonden met AIN0.5.

Stroombron, deel twee

De Howland-stroombron in dit project bestaat uit IC10 en weerstanden R1,

R20, R21 en R22. V^+ in de bovenstaande beschrijving ligt aan GND en is daarom 0 V. V^- is $-2,5$ V, geleverd door spanningsreferentie IC11. R21 komt overeen met R3 in het schema van figuur 2. De uitgangsstroom is gelijk aan:

$$(0 - -2,5) / 10.000 = +250 \mu A.$$

Voeding

In theorie kan de voedingsspanning voor de gehele schakeling (aangesloten op K1) maximaal 15 VDC bedragen, maar dat zorgt wel voor een onnodige belasting van de low-drop spanningsregelaars. De minimale voedingsspanning is 7,5 VDC, aanbevolen wordt 9 VDC. Het totale stroomverbruik is ongeveer 70 mA, waarvan 20 mA voor rekening komt van de achtergrondverlichting van LCD1. (Merk op dat deze laatste een ingebouwde stroombegrenzingsweerstand heeft.) IC1 verlaagt de voedingsspanning naar 5 V voor het leeuwendeel van de schakeling, behalve microcontroller IC4; IC2 maakt 3,3 V voor dit slimme onderdeel. Het analoge ingangsgedeelte heeft ook een voedingsspanning van -5 V nodig, die wordt geleverd door IC3.

De microcontroller

In de afgelopen jaren zagen we zoveel schakelingen rond de Arduino dat som-

migen zijn gaan geloven dat er maar één type 8-bit microcontroller bestaat. Dat dit niet zo is, bewijst ons circuit met zijn gepimpte, op een 8051 gebaseerde brein! En voordat u nu walgend wegkijkt, moet u weten dat de 8051 veertig jaar na introductie nog steeds een veelgebruikte 8-bit processor is. De C8051F350 van Silicon Labs die we hier gebruikten draait intern op 49 MHz en voert de meeste instructies in één klokcyclus uit, waarmee hij behoorlijk snel is.

De C8051F350 maakt deel uit van een familie met vier leden die alleen verschillend zijn in behuizing en ADC-resolutie. De 'F350 is het grootste familielid met een 24-bit sigma/delta-ADC, acht analoge ingangen en zeventien digitale I/O-pennen. Deze controller is bedoeld voor analoge toepassingen, met ingebouwde digitale filters en twee 8-bit DAC's. In dit project benutten we niet al die mogelijkheden, maar voor toekomstige projecten zou u deze MPU gerust wat nader kunnen bestuderen.

Hoewel hij gevoed wordt met 3,3 V zijn de digitale I/O-pennen van de C8051F350 bestand tegen 5 V. Daardoor werkt hij prettig samen met de schakeling eromheen, die op 5 V draait (zie de pullup-weerstanden R4 t/m R8 en het netwerkje RN1).

Een apart klokkristal is niet nodig, want de MCU heeft een gekalibreerde oscillator aan boord.

Voor de bediening worden de vier druktoetsen S1...S4 en een LCD gebruikt. Met P1 stelt u het contrast van het display in; als u vergeet om dit in te stellen wanneer u het board voor de eerste keer opstart, dan valt er wellicht niets te zien. Voor communicatie met een computer kunt u optioneel een seriële USB/TTL-converter plaatsen. Wij hebben ons eigen breakout-board gebruikt, maar u kunt ook kiezen voor een functioneel equivalente interface. Op de foto's in dit artikel ziet u een prototype uitgerust met Elektors (gouwe ouwe) FT232R BoB, nummer 110553. Die is inmiddels vervangen door onze nieuwe FT231X BoB nummer 180537 die compatibel is met de *de facto* standaard USB/serieel-kabels van FTDI.

Software-zaken

Toen we dit schreven waren de mensen bij Silicon Labs zo aardig om nog steeds gratis licenties te verstrekken voor Keil µVision5, zodat mensen die software in C ontwikkelen voor C8051-apparaten dat op gerieflijke wijze kunnen blijven doen. Als u zo'n licentie aanvraagt, lijkt die maar voor één maand geldig te zijn, maar geïnstalleerd in de IDE blijkt hij een permanent. Deze gang van zaken hebben we natuurlijk verkend, en alles ging goed. Je weet alleen nooit of men ooit besluit om hiermee te stoppen en daarom hebben we onze software geport naar SDCC, de open source-compiler voor 8051 en PIC-devices.

SDCC werkt niet met projecten noch met enige ander GCC-gebaseerd tool. Daardoor is ons project gewoon een verzameling bestanden in een map met een

batch-file om ze te compileren.

Zoals met ieder C-programma begint het zichtbare deel in de functie `main`. Als eerste schakelen we hier de watchdog-timer uit. Die kunt u later altijd weer aanzetten als u dat wilt. (Merk op dat de Keil-compiler dit in de runtime-opstartfile doet (CRT), dus daardoor merkt u dat niet. Wij zijn daar door schade en schande achter gekomen.)

Na de initialisatie van de randapparaten van de MCU – 1 ms systeemtimer, UART op 9600 baud, enzovoort – voert het programma een ijking van de ADC uit. Dit is een heel nette voorziening van de MCU, want hiermeeijken we allerlei onvolkomenheden van het analoge frontend weg. Na de ijking levert conversie van V_{max} – de spanning over de 0,01%-weerstand R3 – een waarde van `0xffffffff` op (maximale 24-bit waarde), terwijl V_{min} (GND) een waarde `0` moet geven. De ADC past de correcties hardwarematig toe, we hoeven er zelf niets aan te doen.

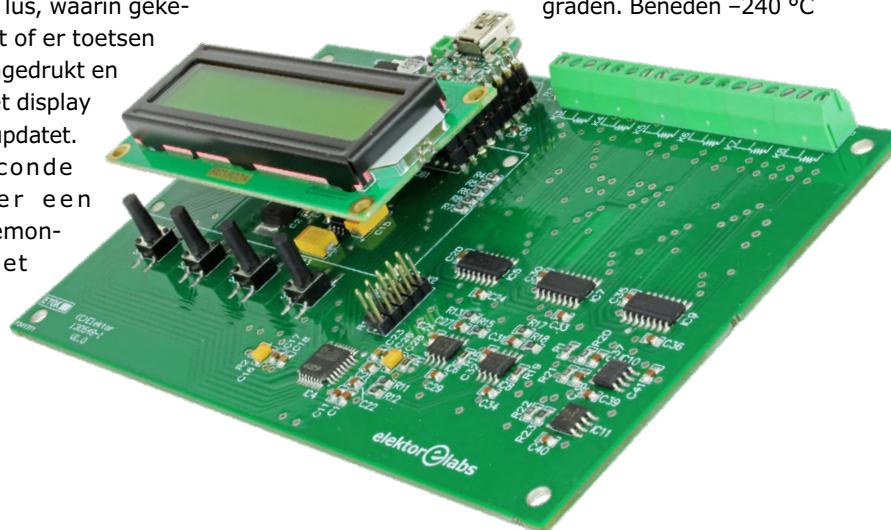
Over bemonsteren gesproken

De rest van de tijd draait `main` in een eindeloze lus, waarin gekken wordt of er toetsen worden ingedrukt en waarin het display wordt geüpdatet. Elke seconde wordt er een kanaal bemonsterd; het resul-

terende sample wordt omgezet in een waarde in graden Celsius of Fahrenheit die vervolgens op het display wordt gezet. Een seconde later is het volgende kanaal aan de beurt. De ADC draait op de achtergrond op bijna 18,7 Hz. Deze hogere snelheid heeft twee redenen. Ten eerste moeten we per RTD twee spanningen meten, V en VRW, omdat we de RTD-waarde moeten compenseren voor de aansluitdraden. Ten tweede heeft de ADC een ingebouwd laagdoorlaatfilter dat drie monsters nodig heeft om een uitgangswaarde te kunnen berekenen. Er zijn daarom tenminste zes samples per RTD-meting nodig, wat overeenkomt met 320 ms. Met een update-interval voor het display van niet meer dan 1 Hz heeft het systeem dan voldoende speling om te kunnen bemonsteren en filteren.

Opzoektafel

We moeten samples omzetten in temperatuurwaarden; dat doen we met een opzoektafel. Pt100-conversietabellen zijn te vinden op internet; wij hebben er een gecompileerd van -240 °C tot +850 °C met een stapgrootte van 10 graden. Beneden -240 °C



Advertentie

smd tec
2009 - 2019

ANNIVERSARY
10 YEARS

www.smd-tec.be

Machines voor
Efficiente
Electronica
Productie

wordt het erg onbetrouwbaar, dus we hebben het display beperkt tot -260°C (wat in feite ‘buiten bereik’ betekent). Om dezelfde reden is de bovengrens vastgelegd op $+850^{\circ}\text{C}$.

De meeste ADC-waarden zullen tussen tabelwaarden in liggen, daarom gebruiken we lineaire interpolatie om de temperatuur te schatten. Als u betere nauwkeurigheid nodig heeft, dan kunt u de stapgrootte van de opzoektafel kleiner maken, zolang die maar in het geheugen past (dat niet zo groot is). Als de tabel erg groot en onhandelbaar wordt, dan zou u een efficiënter zoek-algoritme

kunnen gebruiken dan lineair zoeken, of u zou een polynoom-benadering kunnen implementeren.

Bouw

Hoewel dit project grotendeels uit SMD’s bestaat is de print niet moeilijk op te bouwen. U begint met de laagste componenten met twee pootjes, zoals weerstanden en keramische condensatoren, en vervolgens doet u hogere onderdelen. Let goed op de polariteit waar nodig, zoals bij diodes en elco’s. Het is altijd een goed idee om eerst de voeding te bouwen en te testen (D2, IC1 en IC2 en

de condensatoren daar omheen) voordat u de overige onderdelen plaatst.

Gebruik weerstanden met een tolerantie van 1% in de uitgangstrap, behalve R3 die een tolerantie van 0,01% heeft. Voor een heel erg goede stroombron zou u voor R1, R20, R21 en R22 0,1% tolerantie kunnen kiezen.

Als u de USB/serieel-interface wilt plaatsen, dan zou u extra hoge connectoren voor het display kunnen monteren, want dat zit boven de serial BoB.

De firmware laden

Via connector K2 wordt de firmware in



ONDERDELENLIJST

Weerstanden

(0805, 1% tenzij anders vermeld):

R4,R5,R6,R7,R8 = 10 k

R1,R20,R21,R22 = 10 k 0,1%

R2,R14 = 10 Ω

R3 = 470 Ω 0,01%

R9,R10,R11,R12,R13,R15,R17,R18,R23 = 1 k

R16,R19 = 12 k

P1 = instelpotmeter 10 k

RN1 = weerstandsnetwerk, 6 \times 10 k

Condensatoren (0805):

C1,C19 = 1 μF

C2,C5,C9,C14,C16,C28 = 10 $\mu\text{/}16\text{V}$, Case-A

C3,C4,C6,C8,C10,C11,C12,C17,C20,C21,C22,

C24,C25,C26,C27,C29,C30,C31,C32,C33,

C34,C35,C36,C38,C39,C40,C41 = 0,1 μF

C7 = 47 μF 25 V, diameter 6,3 mm

C13,C15 = 100 $\mu\text{/}16\text{V}$, Case-C

C18,C23,C37 = 1 n

Halfgeleiders

D1,D2 = MBR140SFT1G

IC1 = LD1117DT50TR

IC2 = LD1117DT33TR

IC3 = TC7660EOA

IC4 = C8051F350-GQ, LQFP32

IC5,IC7,IC9 = 4051, SOIC16

IC6,IC8,IC10 = OP07CD

IC11 = LM336M-2.5

Diversen

K1 = 2-polige printkroonsteen, steek 3,5 mm

K2 = 10-polige 2-rijige pinheader,
raster 2,54 mm

K3...K8 = 3-polige printkroonsteen,
steek 3,5 mm

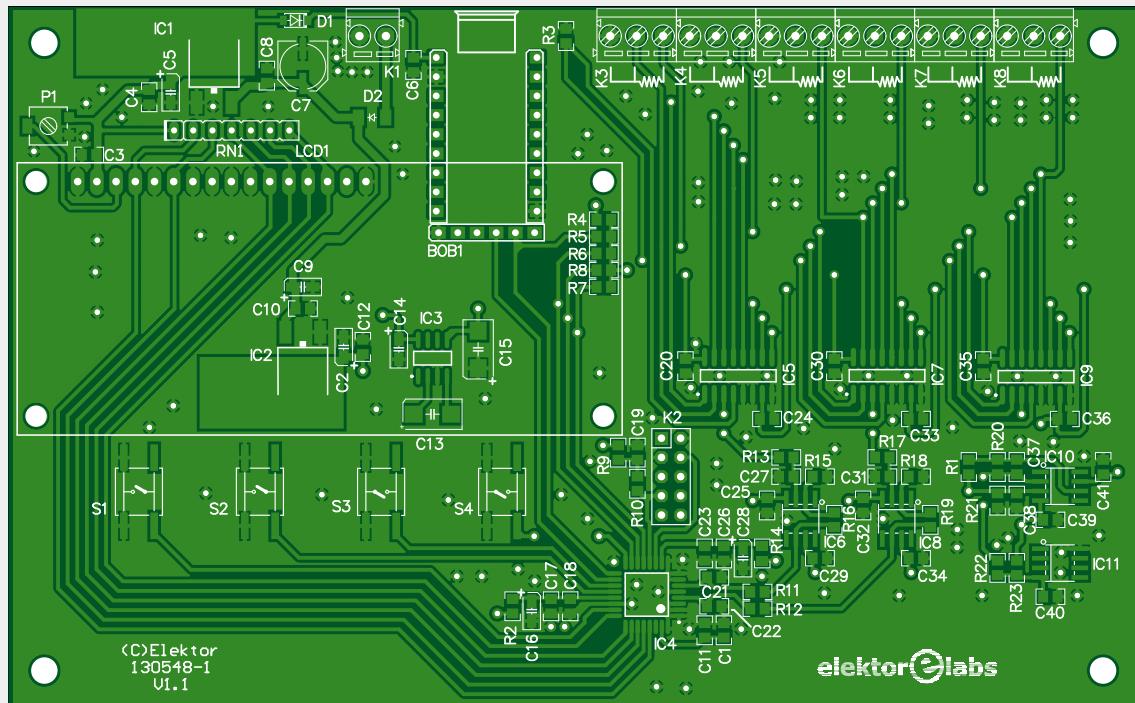
LCD1 = alfanumeriek LCD, 2 regels van
16 karakters, 5 V met backlight

S1...S4 = druktoets, 6 \times 6,2mm, SMD

BOB1 = USB/Serial Bridge/BOB met passende
connectoren

16-polige pinheader + busstrip,
steek 2,54 mm, voor LCD1

Print # 130548-1



de MCU geprogrammeerd. Deze connector voldoet aan de officiële *Silicon Labs 2-wire C2 Programming/Debugging Interface Standard*, dus verbinden met een C2-compatibele programmer mag geen problemen geven. Zulke programmers zijn echter slecht verkrijgbaar, daarom hebben we er zelf maar een gemaakt. In de gratis download bij dit artikel [3] vindt u een Arduino-sketch waarmee u van elk willekeurig Arduino-Uno-achtig board een C2 programmer maakt. De code staat ook op GitHub [2]. Doordat de MCU-poorten bestand zijn tegen 5 V kunnen we dit ongestraft doen.

Sluit Arduino pin A0 aan aan K2 pin 7 ('C2CK'), Arduino pin A1 aan K2 pin 4 ('C2D'), en Arduino GND aan K2 pin 2. Verbind de Arduino met de computer voordat u het board aan zet, en alles zou moeten werken.

Met een Python 3-script, ook in de download, kunt u Intel HEX files uploaden in uw zelfgebouwde programmer. Voor dat script hebt u pySerial nodig.

Gebruik

Er zijn twee firmware-versies voor dit project, één met debugging-opties (suffix 'd') en een met minder opties. De bediening is hetzelfde voor beide versies.

- 'Menu' (S1): met deze toets komt u in de instellingen-'pagina'. Hier kunt u Celsius of Fahrenheit instellen. Met 'Decr' (S2) en 'Incr' (S3) kunt u het display-update-interval instellen tussen één seconde en één minuut. Als u klaar bent kunt u uw instellingen niet-vluchtig opslaan met 'Save' (S4), of u kunt wachten op timeout voor deze pagina. In het laatste geval blijven uw nieuwe instellingen

Weblinks

- [1] Project op Elektor Labs: www.elektormagazine.com/labs/6-channel-temperature-monitor-logger
- [2] C2 programmer op GitHub: https://github.com/ElektorLabs/ElektorLabs_Tools
- [3] Projectpagina bij dit artikel: www.elektormagazine.nl/190118-03

IN DE STORE

- 6-kanaals temperatuur-monitor en -logger, gedeeltelijk geassembleerde module
www.elektor.nl/130548-91
- 6-kanaals temperatuur-monitor en -logger, kale print
www.elektor.nl/130548-1
- LCD 2x16 karakters met blauw achtergrondverlichting
www.elektor.nl/120061-77
- FT231X USB/Serial Bridge/BoB
www.elektor.nl/180537-91

bewaard tot het systeem wordt uit-en weer aangezet (of totdat u iets anders instelt).

- 'Save' (S4): kalibreer het systeem. De debug-optie biedt daarnaast nog:
- 'Decr' (S2): houd deze toets ingedrukt bij het opstarten totdat de splash screens voorbij zijn gekomen, daarmee slaat u de ADC-ijking bij het opstarten over. Gebruik 'Save' (S4) als u wiltijken.
- 'Decr' (S2) en 'Incr' (S3): druk deze toetsen gelijktijdig in om de debug-pagina te openen, waar u de ruwe 16-bits ADC-waarden voor zowel de signalen V (RTD) als VRW (draad) kunt zien. Met 'Incr' gaat u naar een volgend kanaal. Met 'Save' (S4) gaat u terug naar normaal.

Data logging

Prik BOB1 op het board (of gebruik een andere USB/serieel-omzetter, de aansluitingen zijn compatibel met een FTDI-kabel) en verbind hem met een vrije USB-poort op een PC. Die moet hem zien als seriële poort. Start een terminalprogramma en stel het in op 9600 baud, acht databits, geen pariteit en één stopbit (9600n81). Zodra het board is opgestart begint het waardes te sturen, gescheiden door een komma (CSV), in normaal leesbaar ASCII-formaat. De meeste seriële terminals voorzien in de mogelijkheid om deze data op te slaan in een bestand voor latere verwerking in bijvoorbeeld een spreadsheet. ▶

(190118-03)



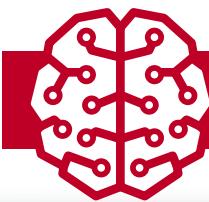
ONLINE ASSEMBLAGE VAN ELEKTRONISCHE PRINTPLATEN

www.emsproto.com

ONLINE PRIJSOPGAVE VAN UW ELEKTRONISCHE PRINTPLAAT

LEVERTIJD VAN 2 tot 12 DAGEN

HOEVEELHED VAN 1 tot 50 PRINTPLATEN



ARTIFICIAL
INTELLIGENCE

Bouw uw privé-spraakassistent met Snips



kunstmatige intelligentie voor de Raspberry Pi

Mennad Yami en Robin Guignard-Perret (Frankrijk)

Snips is een spraakgestuurd platform voor het Internet of Things, waarmee ontwikkelaars en fabrikanten een spraakassistent aan hun producten kunnen toevoegen. Wil uw Raspberry Pi nog steeds niet naar u luisteren? Snips brengt daar verandering in. In dit artikel wordt uitgelegd hoe u een Raspberry Pi van deze 'oren' kunt voorzien.

Snips, opgericht in 2013, stelt zich als doel om ieder apparaat te voorzien van een ‘kunstmatig intelligente’ assistent die de technologie zo intuïtief maakt dat deze naar de achtergrond verdwijnt.

De technologie van Snips is uniek, want alles gebeurt lokaal op het apparaat waar de gebruiker tegen praat en dit betekent dat er geen data naar de cloud worden verzonden. Het concept staat garant voor privacy en een betrouwbare werking, zelfs als het internet niet beschikbaar is, en hiermee is Snips een van de eerste spraaktechnologieën die de algemene regels voor de gegevensbescherming volgt.

Het Snips-platform is gratis voor iedereen die onze technologie voor niet-commerciële doeleinden wil uitproberen. We hebben een community gecreëerd van meer dan 25.000 personen – de grootste verzameling ontwikkelaars van stemtoepassingen naast Google en Amazon! Snips werkt onder Linux (Raspberry Pi), Android, iOS, MacOS en Debian. De tools en de documentatie om eigen spraaktoepassingen te creëren zijn voor iedereen online beschikbaar [1].

Snips is beschikbaar in het Frans, Engels, Duits, Japans, Spaans en Italiaans.

De werking van onze modellen op embedded systemen

Privacy-by-design betekent dat privacy de kern vormt van het ontwikkelproces van het systeem. Privacy is met name van groot belang als spraakassistenten in wisselende omgevingen worden toegepast, waaronder de privésfeer van de gebruikers. Het doel is om de gebruikers te beschermen tegen ieder misbruik van hun gegevens, nu of in de toekomst. Bij Snips heeft *privacy-by-design* er toe geleid dat er geen enkele overdracht van gebruikersgegevens naar de cloud plaatsvindt.

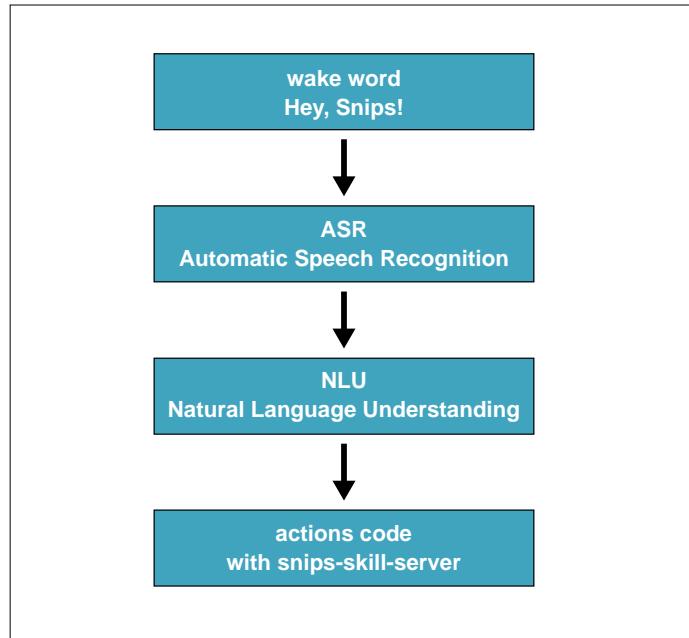
Sinds de geboorte van het Snips-spraakplatform hebben de portabiliteit en een goed beheer van resources prioriteit gekregen. Het resultaat is dat het platform op veel IoT-hardware kan werken, zelfs op de ‘lichte’ Raspberry Pi 3, een board dat populair is bij ontwikkelaars. Ook andere Linux-boards worden ondersteund. De Snips-SDK voor Android werkt met apparaten die beschikken over Android 5 en een ARM-processor, en de iOS-SDK richt zich op iOS 11 en latere versies. Om redenen van efficiëncy en portabiliteit zijn de algoritmes waar nodig opnieuw geïmplementeerd in Rust (een gecompileerde programmeertaal van Mozilla Research).

Spraakassistenten

Moderne spraakassistenten kunnen worden opgesplitst in een aantal fundamentele bouwstenen: het detecteren van een sleutelwoord (*wakeword*), het herkennen van woorden, (*Automatic Speech Recognition*, ASR), het begrijpen van spreektaal (*Natural Language Understanding*, NLU), en de actions-code (actie) die wordt uitgevoerd zodra een opdracht is gedetecteerd.

Het *wakeword* is het sleutelwoord waarmee het begin van een verzoek aan uw assistent wordt aangegeven. Een Snips-assistent begint actief te luisteren zodra hij “Hey, Snips” hoort. Vervolgens begint hij met het omzetten van de woorden van de gebruiker, en dit wordt gedaan door de tweede bouwsteen: de ASR.

De **ASR** probeert de geluidssignalen die door de microfoon worden opgevangen om te zetten naar voor mensen begrijpelijke tekst. Op het moment dat de gebruiker “Hello World” zegt wordt het opgevangen geluidssignaal langs een eerste *deep learning*-functie geleid, een akoestisch model dat het



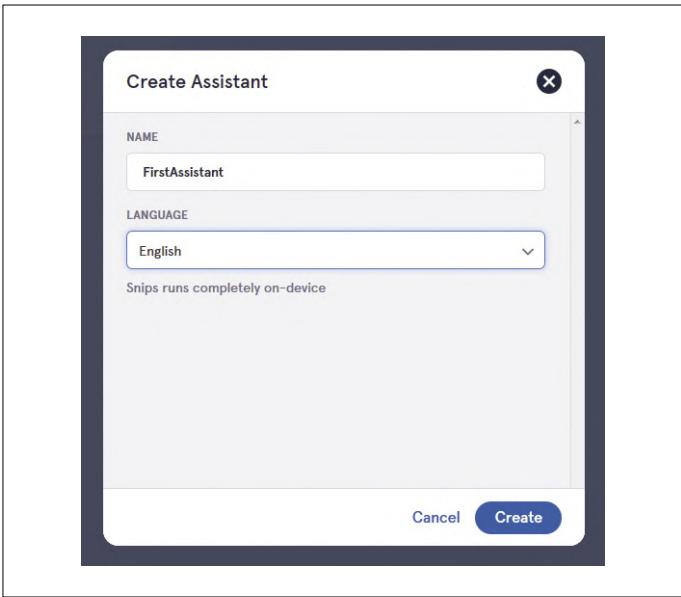
Figuur 1. De bouwstenen die nodig zijn voor spraakherkenning.

converteert naar fonemen, de elementaire geluidseenheden die door taalkundigen worden gebruikt. “Hello World” zal hierbij in eerste instantie worden voorgesteld door “hə'ləʊ w3:lд”. Vervolgens wordt een op de taal van de assistent afgestemde functie gestart die deze fonetische voorstelling transformeert naar woorden die wij gebruiken, dit is het taalmodel. Als de assistent voor de Engelse taal is geconfigureerd zal “Hello World” correct worden geretourneerd en vervolgens doorgeven aan het volgende blok: de NLU.

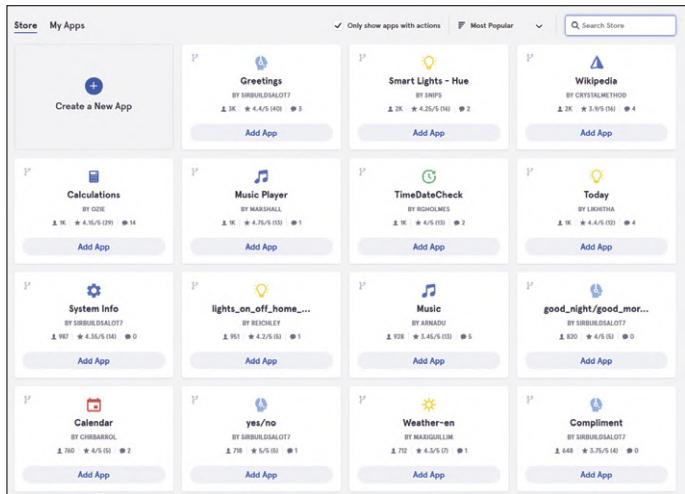
Met de **NLU** kan de betekenis van een tekst worden bepaald. De spraakassistent is ontworpen voor een beperkte reeks verzoeken, en het is de taak van de NLU om een bedoeling te koppelen aan een bedoeling. Zo’n bedoeling wordt in een zin vaak begeleid door parameters die moeten worden afgesplitst om het verzoek correct te kunnen behandelen. Als de gebruiker bijvoorbeeld heeft gezegd: “Hey, Snips, doe het rode licht aan in de kamer van Paul” en de NLU ontvangt de transcriptie van deze zin, dan zal hij de bedoeling ‘het licht aandoen’ detecteren, met als parameters de plaats ‘kamer van Paul’ en de kleur ‘rood’. Deze informatie kan vervolgens aan de actions-code worden doorgegeven.

Het laatste onderdeel is de **actions-code**, die in een oneindigelus wacht totdat een bedoeling wordt gedetecteerd en deze vervolgens, rekening houdend met de bijbehorende parameters, zal uitvoeren. Deze code is specifiek voor ieder apparaat en elke toepassing, en kan iedere gewenste in een script aangegeven actie uitvoeren. De actions-code kan bijvoorbeeld een beroep doen op een API op afstand, een aangesloten object besturen, de spraakassistent laten spreken, of al deze dingen tegelijk. Het *wakeword*, de ASR en de NLU dienen als intelligente zintuigen voor uw assistent, en de rest wordt geregeld door de actions-code (zie het schema in **figuur 1**).

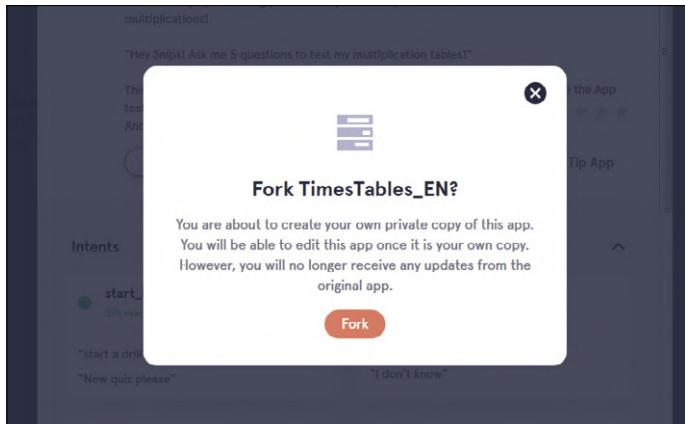
De andere spraakassistenten die op de markt zijn (Google Home, Alexa, Siri) hebben hun detector voor het *wakeword*, hun ASR en hun NLU in de *cloud* ondergebracht. Ze maken gebruik van zeer grote modellen die vanwege de gebruikte technologie erg



Figuur 2. Creëren van een spraakassistent met de online-console van Snips.



Figuur 3. Voorbeelden van Apps die via de Snips-console beschikbaar zijn.



Figuur 4. Met de functie *fork* kopieert u een bestaande App om deze naar uw eigen wensen te kunnen aanpassen (dit bericht verschijnt automatisch als u op *Edit* klikt).

veel rekenkracht vereisen. Dit is bij Snips niet het geval. De modellen worden van geval tot geval voor elke assistent en voor iedere gebruiker gecreëerd en zijn specifiek voor bepaalde taken ontworpen. Hierdoor kunnen met Snips ultralichte assistenten worden gecreëerd die op een Raspberry Pi 3 of zero draaien, en daarbij 100% *offline* zijn. Dit alles zonder enige concessie te doen aan de prestaties van de spraakherkenning.

Het Snips-platform

Het Snips-platform is de omgeving die alle benodigde onderdelen voor een spraakassistent bevat. Door het te installeren kan iedereen beschikken over zijn eigen privacy-respecterende spraakassistent voor Linux, MacOS, iOS en Android. Naast de *wakeword*-detector, de ASR en de NLU vindt u hier ook de **Dialog Manager** die verantwoordelijk is voor de coördinatie en de volgorde van de taken die moeten worden uitgevoerd. De **TTS** (*Text To Speech*) houdt zich bezig met de stemsynthese voor eventuele antwoorden. Deze ontvangt tekst en spreekt deze duidelijk uit met het accent dat past bij de taal van de assistent. Tot slot is er de **snips-skill-server**, die is belast met de uitvoering van de actions-code.

De interne communicatie op het platform berust op het lichte en snelle Hermes-protocol, dat door Snips rond het MQTT-protocol is gecreëerd. Dankzij Hermes kan de werking van het platform eenvoudig worden gepersonaliseerd, bijvoorbeeld voor de communicatie met nevenapparaten van uw hoofd-apparaat, of om de TTS een zin te laten uitspreken of om een van de onderdelen van het platform afzonderlijk te bedienen. Als het Snips-platform eenmaal op een apparaat is geïnstalleerd, kunnen we het van een assistent voorzien. Deze creëert u op de webconsole van Snips [1], zie **figuur 2**. Met de console definiert u de taal van de assistent, de begrippen die hij moet herkennen en zijn functiegebieden (*Apps*).

Zo kunt u bijvoorbeeld het platform op uw Raspberry Pi een Franse assistent geven, met een App om de verlichting te besturen, en ook een App voor het weer.

Creëer en installeer uw assistent

We gaan nu uw Raspberry Pi van een persoonlijke assistent voorzien. Eerst moet het Snips-platform op de RPi worden geïnstalleerd. Om de installatie en het manipuleren van assistenten te vereenvoudigen hebben we **SAM** gemaakt (*Snips Assistant Manager*), een implementatietool dat eenvoudig via de commandoregel werkt. SAM is een PC-programma dat via SSH met uw RPi communiceert. De twee apparaten moeten zich dus in hetzelfde netwerk bevinden. Om SAM te kunnen gebruiken moeten op uw hoofdcomputer *Node.js* (JavaScript execution environment) en *npm* (packet manager voor Node.js) zijn geïnstalleerd. Vervolgens kunt u SAM installeren met `sudo npm install -g snips-sam`. Met het commando `sam devices`, dat een lijst geeft van alle RPi's op het netwerk, kunt u uw RPi detecteren. Als u het IP-adres of de *hostname* van uw apparaat hebt verkregen, kunt u dit met SAM verbinden met het commando `sam connect <adresseIP/hostname>`. Tenslotte wordt met `sam init` de installatie van het Snips-platform gestart. Om uw microfoon en uw audio-uitgang in te schakelen gebruikt u `sam setup audio`.

Snips-console

Ga naar [1] om een account aan te maken en vervolgens een nieuwe assistent. Deze assistent kan beschikken over een of meer functies die u kunt kiezen uit de App's die beschikbaar

zijn in de *App store* (het weer, kookboek, besturing van Philips Hue lampen... zie **figuur 3**). U kunt ook uw eigen Apps maken, of met behulp van de *fork*-functie een App uit de *App store* personaliseren en zo een unieke functie creëren (**figuur 4**). Een App ontvangt bedoelingen (*intents*) van de NLU, samen met eventuele bijbehorende parameters (*slots*). Als de gebruiker van een kalender-App zegt: "Voeg een verjaardag toe op 13 november" dan wordt deze opdracht door de NLU vertaald naar de *intent* 'addEvent' met als *slots* de naam van het evenement (verjaardag) en de datum (13 november). Het Snips-team heeft vooraf al bepaalde veelgebruikte *slots*, zoals de datum, aangemaakt om de detectie ervan te vergemakkelijken. In **listing 1** ziet u de gegevens die door de NLU in JSON-formaat aan de actions-code worden doorgegeven.

Uw assistent heeft oefengegevens nodig om zijn neurale netwerk te trainen. U kunt met de console uw eigen voorbeelden creëren. Om de prestaties van de NLU te verhogen kunt u de positie en de aard van de *slots* in een zin aangeven door ze met de muis te markeren. Hoe groter het aantal voorbeelden, en hoe gevarieerder, des te beter zal uw assistent presteren. De Snips-console biedt ook een betaalde dienst aan voor de productie van trainingsdata.

Vergeet niet om de ingevoerde gegevens op te slaan. Vervolgens zal de console uw assistent updaten. U kunt de testinterface rechts op het scherm gebruiken om de reactie van uw assistent op gesproken of geschreven verzoeken te bekijken. Nu kunt u met behulp van SAM uw assistent op uw RPi installeren. Met het commando `sam login` krijgt u toegang tot het account dat u op de console hebt ingesteld. Vervolgens voert u `sam install assistant` uit en kiest u de te installeren assistent.

Actie

De actions-code bestaat uit *callbacks* die worden uitgevoerd als de bijbehorende *intent* wordt gedetecteerd. U kunt iedere gewenste programmeertaal gebruiken, maar Python en JavaScript beschikken over een Hermes-bibliotheek die de interne communicatie met het Snips-platform vereenvoudigt. Deze twee talen worden dus aanbevolen (maar zijn niet verplicht). U kunt uw code handmatig uitvoeren, maar de *snips-skill-server* kan dit ook bij het opstarten van de RPi automatisch voor u doen. Maak hiervoor een map met een naam naar keuze aan in `/var/lib/snips/skills/`, en plaats uw script in deze map. Uw script moet een naam hebben die begint met action – en moet executable zijn. De rol ervan is in een oneindige lus te wachten op *intents*. Het is mogelijk om een App te koppelen aan een map op GitHub en bij installatie van de assistent uw bestanden door SAM naar de juiste plaats te laten kopiëren. De opslag moet openbaar zijn, en u kunt deze toevoegen aan de Actions-tab van de console. Naast uw action-bestand kunt u hier ook een bestand `setup.sh` plaatsen dat tijdens de installatie wordt uitgevoerd, bijvoorbeeld om uw project te compileren.

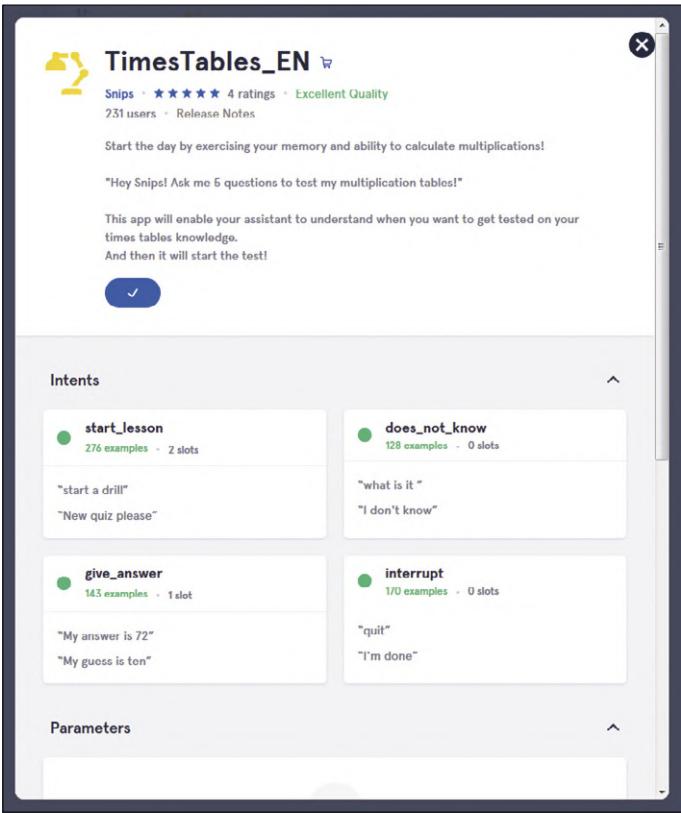
Voorbeeld-assistent

Nu volgt een praktijkvoorbeeld, waarin we een assistent maken die u in de vorm van een vraag- en antwoordspel de tafels van vermenigvuldiging overhoort. Maak vanaf de console een nieuwe assistent aan met de Engelse naam "FirstAssistant", die een App met de naam "TimesTables". bevattet. [Noot van de redactie: U vindt in de *App Store* de App "TimesTableEN" die wat uitgebreider is dan degene die hier wordt beschreven, zie **figuur 5**]. De speler moet het spel starten, vervolgens antwoord



Listing 1. Gegevens die de NLU aan de actions-code doorgeeft (JSON-formaat).

```
"input": "add an anniversary for the thirteenth
of november",
"intent": {
"intentName": "userName:addEvent",
"probability": 0.9662896
},
"slots": [
{
"rawValue": "anniversary",
"value": {
"kind": "Custom",
"value": "anniversary"
},
"range": {
"start": 10,
"end": 22
},
"entity": "common names",
"slotName": "summary"
},
{
"rawValue": "for the thirteenth of november",
"value": {
"kind": "InstantTime",
"value": "2019-11-13 00:00:00 +00:00",
"grain": "Day",
"precision": "Exact"
},
"range": {
"start": 23,
"end": 46
},
"entity": "snips/datetime",
"slotName": "start_datetime"
}
]
```



Figuur 5. De App *TimesTables* is online beschikbaar. U kunt deze selecteren en vervolgens onderaan rechts op het scherm op *Add 1 App* klikken om deze toe te voegen.

geven op een vraag (of passen als hij het antwoord niet weet). We hebben de drie *intents* daarom *start_quiz*, *gives_answer* en *doesn't know* genoemd. Let op: de actions-code ontvangt de naam van de *intent* in het formaat *userName:intentName* (bijvoorbeeld *snips:gives_answer*).

- *intent start_quiz*: beperkt zich tot het starten van het spel, en heeft dus geen *slot* nodig om het gedrag van uw assistent te bepalen. Hij moet zinnen van het type "let's start the game" of "I want to play" begrijpen. Om de *intent* op veel situaties voor te bereiden is het belangrijk om gevareerde voorbeelden van zinnen te maken. Daarom voegen we "I feel like playing the game" en "let's play" toe aan de voorbeelden voordat we ze opslaan (**figuur 6**).
- *intent gives_answer*: moet een variabel antwoord kunnen begrijpen. Daarom is een *slot* gewenst dat we *answer* noemen en waar we het voorgedefinieerde type *number* aan toekennen. Vergeet niet de getallen in de voorbeeldzinnen te markeren als dit niet al automatisch is gebeurd (**figuur 7**). Hiervoor moet u het getal met de muis selecteren, vervolgens op *Select a slot* klikken en daarna op *answer*. Er worden antwoorden verwacht zoals "forty-nine" of "I think it's twelve". De gegevens in JSON-formaat die uw actions-code ontvangt bevatten de waarde van het getal dat de gebruiker in zijn *slots* heeft uitgesproken.
- *intent doesn't know*: volgt dezelfde logica als *start_quiz* omdat deze geen parameter nodig heeft. U hoeft dus alleen maar voldoende voorbeelden te geven om de functies van uw assistent te trainen. "I don't know" "I have no

Figuur 6. Mogelijke formuleringen om het spel te starten, opgeslagen in de *intent start_quiz*. Dit is de Engelse versie, maar u kunt zelf een anderstalige versie van deze App maken.

idea" of "I can't get the answer" zijn in dit verband relevante voorbeelden.

De assistent is nu klaar om met SAM te worden geïnstalleerd, maar eerst moeten we de actions-code nog onder de tab *Actions* onderbrengen.

We hebben hiervoor een Python-script geschreven met de naam `action-timetable.py` (te downloaden van [2]). Dit moet in de map `/var/lib/snips/skills/timetables/` worden opgeslagen.

Actions-code

Bij het produceren van de actions-code maken we gebruik van de bibliotheek `hermes_python`. De eerste stap bestaat uit het importeren van deze bibliotheek en ook van de bibliotheek `random` (om de getallen voor de vragen te genereren).

Definities

```
#!/usr/bin/env python2
# coding: utf-8
```

```
from hermes_python.hermes import Hermes
import random
```

De combinatie "#!" (eerste regel) is nodig om de *snips-skill-server* de code van uw assistent te kunnen laten uitvoeren. Nu definiëren we de globale variabelen voor de actie:

```
MQTT_HOST = "localhost"
```

```
MQTT_PORT = "1883" # Standard MQTT port
MQTT_URL = "{}:{}".format(MQTT_HOST, MQTT_PORT) # ==>
"localhost:1883"

SNIPS_USER_NAME = "snips"
INTENT_STARTS_QUIZ = "{}:start_quiz".format(SNIPS_USER_NAME) # ==> "snips:start_quiz"
INTENT_GIVES_ANSWER = "{}:gives_answer".format(SNIPS_USER_NAME) # ==> "snips:gives_answer"
INTENT_DOESNT_KNOW = "{}:doesnt_know".format(SNIPS_USER_NAME) # ==> "snips:doesnt_know"
```

```
sessions_states = {}
```

De drie variabelen `MQTT_...` definiëren de toegang tot de MQTT-server, in ons geval wordt er verbinding gemaakt met `localhost`, want de code wordt uitgevoerd op de RPi die Snips bevat. De drie variabelen `INTENT_...` definiëren de namen van uw *intents*, voorafgegaan door de naam van de gebruiker die de assistent op [1] heeft gecreëerd. Deze moet worden vervangen door uw gebruikersnaam.

De variabele `sessions_states` stelt de assistent in staat om de vraag te onthouden die hij aan de gebruiker heeft gesteld.

Het spel starten

Nu definiëren we de functie die moet worden aangeroepen als de gebruiker de spraakassistent vraagt om hem de tafels van vermenigvuldiging te overhoren.

```
def starts_quiz(hermes, intent_message):
```

The screenshot shows the Snips Intent Settings interface. At the top, there's a breadcrumb navigation: Home > firstAssistant > TimesTables > gives_answer. On the right, there's a button labeled 'Intent Settings'. Below the navigation, the intent name 'gives_answer' is displayed in bold. A sub-header 'Let the user try to give the right answer' follows. The main area is divided into two sections: '1 Slot' on the left and '6 Training Examples' on the right. The 'Slot' section has a 'Create New Slot' button and a dropdown menu set to 'answer' with 'snips/number builtin' selected. The 'Training Examples' section has an 'Add' button and a list of six examples: 'the answer is 10', 'twenty-four', 'maybe 16', 'I think it's twelve', 'forty-nine', and '56'. At the bottom, there's a button labeled 'Generate'.

Figuur 7. In de *intent gives_answer* zijn de getallen voor het *slot* van het type *Answer* gemarkeerd.

```

session_id = intent_message.session_id

# Generating question
a = random.random() * 10 + 1
b = random.random() * 10 + 1
tts = "What's {} times {} ?".format(a, b)
sessions_states[session_id] = [a, b]

# Define what the assistant is able to understand
intent_filter = [INTENT_DOESNT_KNOW, INTENT_GIVES_ANSWER]

hermes.publish_continue_session(session_id, tts,
                                intent_filter)

hermes.publish_end_session(session_id, tts)

```

Iedere functie die is geassocieerd met een *intent* ontvangt twee objecten als parameters: `hermes` en `intent_message`.

De eerste stap bij het opzetten van een dialoog met Snips is het ophalen van de `session_id`. Hiermee wordt de identiteit van de dialoog op een unieke manier bepaald. Dit is belangrijk, want verschillende personen kunnen uw assistent via nevenapparaten op hetzelfde moment gebruiken. Een dialoogsessie begint als een gebruiker het *wakeword* uitspreekt en eindigt als uw code een `end_session` stuurt.

Vervolgens creëren we een willekeurige vraag over de tafels van 1 tot 10 en plaatsen deze in de `sessions_states` dictionary. Hiermee kunnen we later het antwoord van de gebruiker verifiëren.

Nu moeten we een evenement publiceren om de dialoog te vervolgen met het stellen van de vraag aan de gebruiker en vervolgens te wachten op zijn antwoord. Hiervoor voeren we `hermes.publish_continue_session(...)` uit, met als eerste parameter de identiteit van de voort te zetten sessie. De tweede parameter is de vragende zin die de assistent naar de gebruiker gaat uitspreken (`tts = text to speech`). De laatste parameter is het `intent filter` waardoor de assistent weet dat dat de gebruiker gaat antwoorden met een zin die in een van de twee geselecteerde *intents* voorkomt.

De vraag beantwoorden

Deze twee *intents* zijn in dit geval `INTENT_DOESNT_KNOW` en `INTENT_GIVES_ANSWER`. Dit betekent dat de gebruiker alleen maar antwoord kan geven **of** zeggen dat hij het niet weet.

De volgende stap is het definiëren van de functie die wordt aangeroepen als de gebruiker een antwoord geeft.

```

def gives_answer(hermes, intent_message):
    session_id = intent_message.session_id

    # get answer from user
    answer = intent_message.slot.get("answer")
    answer = None if answer is None else answer.first().value

```

```

# defines TTS
tts_good = "Good answer !"
tts_bad = "Wrong answer ..."
tts_no_answer = "I'm sorry I didn't get your answer"
tts = ""

# Check user's answer
if answer is None:
    tts = tts_no_answer
elif answer == sessions_states[session_id][0] * sessions_states[session_id][1]:
    tts = tts_good
else:
    tts = tts_bad

hermes.publish_end_session(session_id, tts)

```

De twee initialisatieregels van de variabele `answer` dienen om de waarde (in de vorm van een integer) van het antwoord van de gebruiker binnen te halen. De tweede regel verifieert of de gebruiker wel heeft geantwoord voordat de waarde wordt opgehaald. De regels `tts_...` definiëren de zinnen waarmee de assistent in de volgende drie gevallen antwoordt:

1. De gebruiker heeft het goede antwoord gegeven;
2. De gebruiker heeft een fout antwoord gegeven;
3. De assistent heeft het antwoord van de gebruiker niet begrepen.

Met de `if` wordt bepaald in welk van de drie gevallen we ons bevinden.

Tot slot publiceren we het evenement `end_session` met als parameter de `session_id` evenals de zin die de assistent gaat uitspreken voordat de dialoogsessie wordt gesloten.

De functie die wordt aangeroepen als de gebruiker het antwoord niet weet is vrij simpel. Er wordt alleen een end-session gepubliceerd om de dialoog met een vooraf gedefinieerde TTS te sluiten.

```

def doesnt_know(hermes, intent_message):
    session_id = intent_message.session_id
    tts = "Ok, at least try next time !"

hermes.publish_end_session(session_id, tts)

```

Het einde van het spel

De volgende functie wordt aangeroepen als de sessie na het aanroepen van `hermes.publish_end_session(...)` klaar is met afsluiten.

```

def session_ended(hermes, session_ended_message):
    del sessions_states[session_ended_message.session_id]

```

Weblinks

- [1] Console, tools en documentatie van Snips: <https://console.snips.ai>
- [2] Python-code van de App op de webpagina bij dit artikel: www.elektrormagazine.fr/180733-01
- [3] Review: 'Snips – spraakherkenning op de Raspberry Pi':
www.elektrormagazine.nl/news/review-snips-spraakherkenning-op-de-raspberry-pi

Hierbij wordt het element dat met de sluitende sessie was geassocieerd uit de dictionary verwijderd.

Hermes komt erbij

Nu moet alles nog met elkaar worden verbonden:

with Hermes(MQTT_URL) as h:

```
# Subscribe to intents
h.subscribe_intent(INTENT_STARTS QUIZ, starts_quiz) \
.subscribe_intent(INTENT_GIVES ANSWER, gives_answer) \
.subscribe_intent(INTENT_DOESNT KNOW, doesnt_know)
```

```
# Subscribe endSession
h.subscribe_session_ended(session_ended)
# Start Listening
h.loop_forever()
```

De eerste regel creëert een instantie van het object `Hermes` in een contextmanager. Merk op dat het argument `MQTT_URL` de aan het begin van het bestand gedefinieerde url is.

De drie regels na `# Subscribe to intents` bepalen dat met een gegeven *intent* (gedefinieerd door de globale variabelen) een gegeven functie wordt geassocieerd. Dankzij deze drie regels weet Snips welke functie voor ieder van de ontvangen *intents* moet worden aangeroepen.

Vervolgens geven we op dezelfde manier aan dat de functie `session_ended` wordt aangeroepen als een dialoogsessie klaar is. Tot slot voeren we de functie uit die wacht op de verschillende Snips-evenementen, en die de bij iedere *intent* behorende functies aanropt.

Samenvattend heeft de actions-code de volgende structuur:

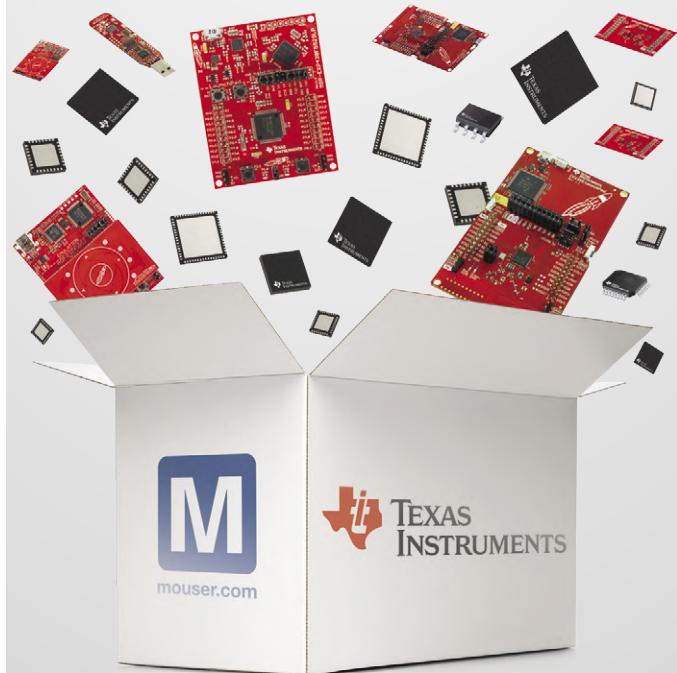
- importeren van bibliotheken
- definiëren van variabelen
- definiëren van de acties bij ieder *intent*
- initialisatie van een context (`Hermes`)

Wij hopen dat dit eerste artikel over de spraakassistent Snips u een eind op weg heeft kunnen helpen. U bent nu klaar om gesproken commando's aan uw projecten voor de Raspberry Pi toe te voegen. Vergeet voor uw eigen gemak niet om altijd even in de *App Store* te checken of er wellicht al iemand een App heeft ontworpen die sterk lijkt op wat u nodig hebt. ▶

(180733-04)

Advertentie

Grootste assortiment TI-producten op voorraad



Meer dan **46.000**
TI-producten

Meer dan **4.000**
TI-ontwikkelingstools



Mouser Electronics - uw officiële TI-distributeur met meer producten op voorraad voor uw volgende ontwerp.
nl.mouser.com/ti

IN DE STORE



→ Snips Voice Interaction Base Kit for Raspberry Pi (incl. RPi 3B+)

www.elektor.nl/snips-voice-interaction-base-kit-for-raspberry-pi-incl-rpi-3b

Zie ook het online-review op [3].

→ Raspberry Pi 3B+

www.elektor.nl/rpi-3b-plus



Snips

spraakherkenning met de Raspberry Pi

Mathias Claußen (Elektor Labs)

Spraakassistenten zoals Alexa, Google Home en Siri zijn inmiddels heel gewoon. De eigenlijke herkenning van gesproken woorden gebeurt echter niet lokaal, maar in de cloud van de aanbieder – wat natuurlijk vragen oproept omtrent de gegevensbescherming. Maar er zijn ook andere oplossingen zoals Snips, een vorm van spraakherkenning die volledig lokaal plaatsvindt op bijvoorbeeld een Raspberry Pi. Wij hebben zo'n spraakherkennings-kit bekeken en laten hier zien hoe daarop eigen applicaties kunnen worden ontwikkeld.

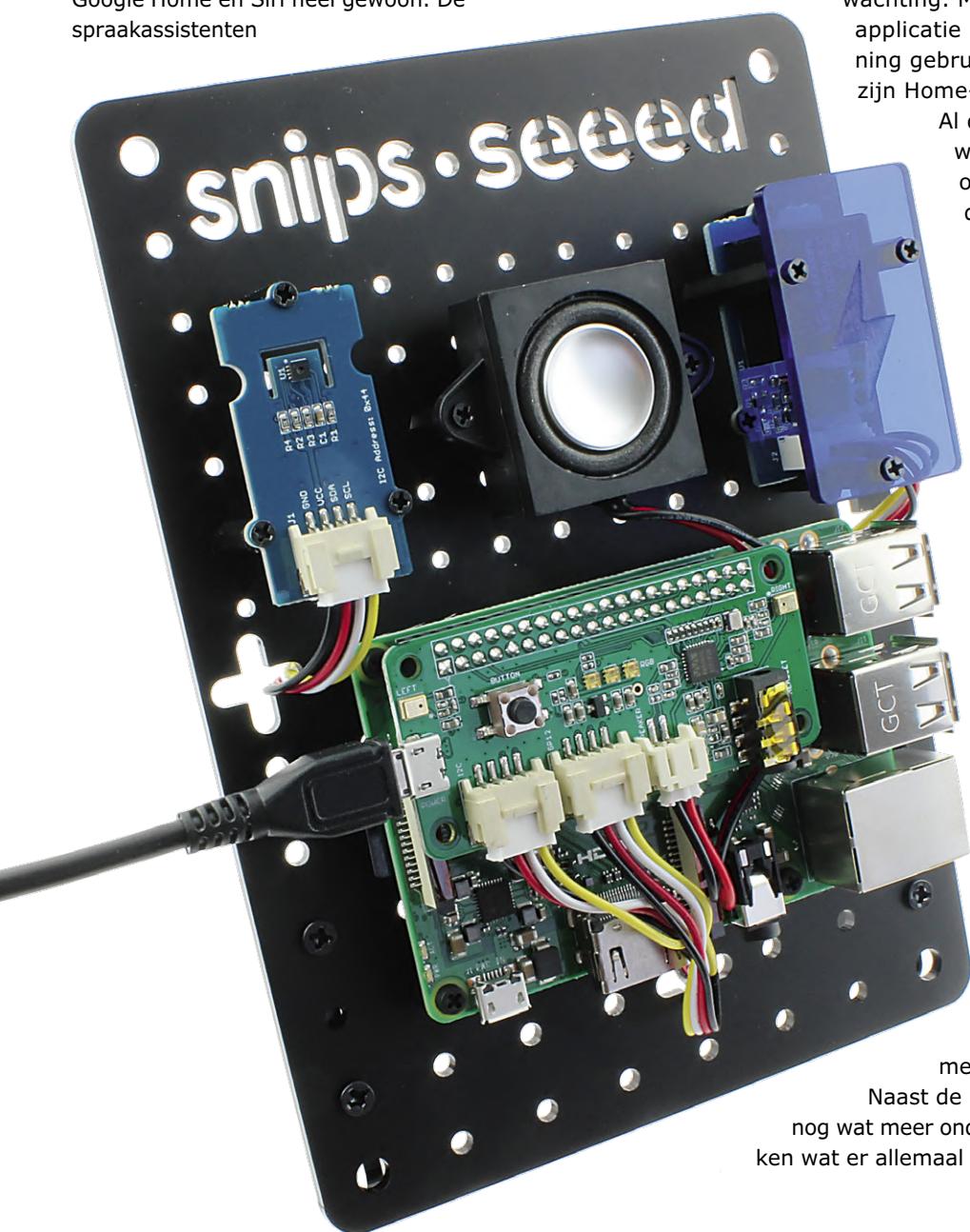
"Computer – thee, Earl Gray, heet" – wat 10 jaar geleden nog klonk als sciencefiction, is vandaag de dag dankzij Alexa, Google Home en Siri heel gewoon. De spraakassistenten

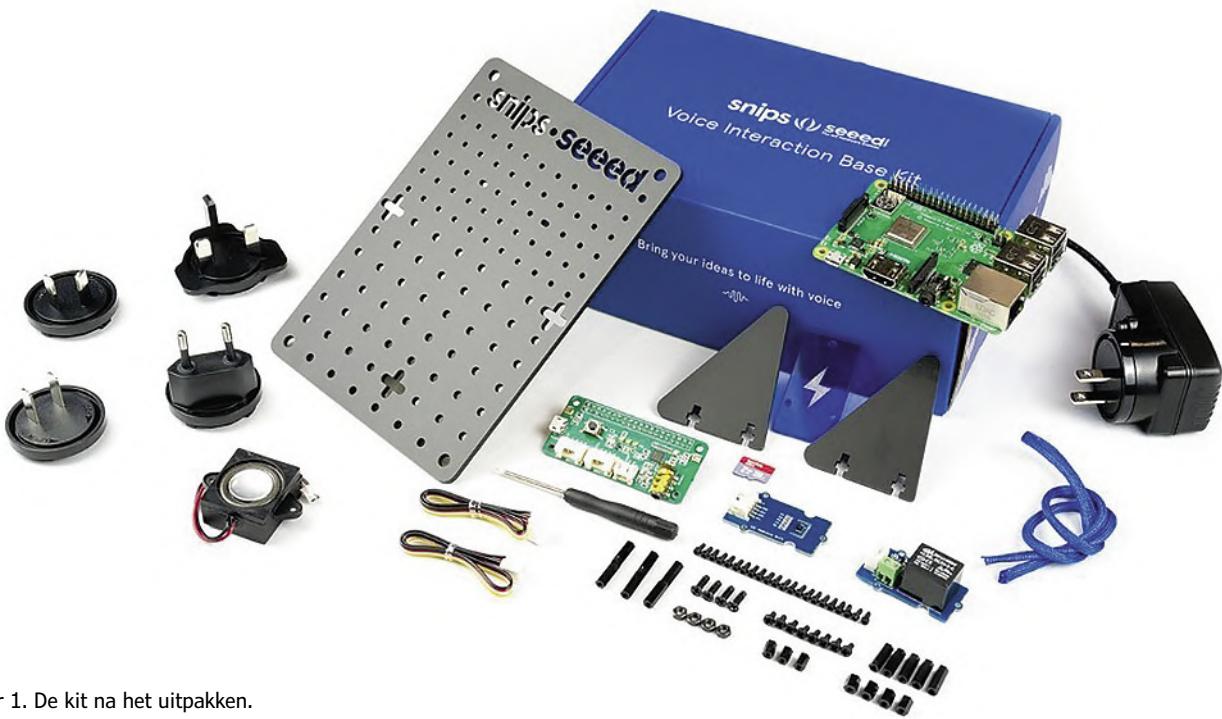
van Amazon, Google en Apple luisteren naar onze woorden en reageren daarop, zij het nog niet altijd naar verwachting. Maar wie voor deze systemen zijn eigen applicatie wil schrijven om van de spraakherkenning gebruik te maken, komt er al snel achter dat zijn Home-assistent niet blijkt te zijn wat hij lijkt.

Al deze systemen reageren op een sleutelwoord dat gewoonlijk is voorgeschreven of alleen uit een beperkte lijst kan worden geselecteerd. Nadat we de assistent met het sleutelwoord hebben geactiveerd, worden alle geluidssignalen die hij binnen krijgt rechtstreeks naar de server van de fabrikant gestuurd. De eigenlijke herkenning van de woorden gebeurt dus niet lokaal, maar in de cloud van de aanbieder. Daarbij is de cloud niets anders dan een groep computers die ergens in een rekencentrum staat. Als gebruikers en app-ontwikkelaars hebben we er geen zicht op wat met de data gebeurt. We kunnen slechts hopen dat de aanbieder zich houdt aan de hier geldende regelgeving op het gebied van gegevensbescherming. Amazons Alexa is in de VS al eens opgeroepen als 'getuige' in een moordzaak.

Maar er zijn ook andere oplossingen, zoals Snips – dat is een spraakherkenning die volledig offline werkt en de data binnen het eigen thuisnetwerk houdt. Ook is geen permanente internetverbinding of een cluster van computers nodig om de spraakherkenning uit te voeren. Om te kunnen beginnen volstaat de Snips Voice Interaction Kit, met een Raspberry Pi 3B+ als basis.

Naast de Raspberry Pi zijn voor spraakherkenning nog wat meer onderdelen nodig. Laten we eerst eens bekijken wat er allemaal in de kit zit (**figuur 1**).





Figuur 1. De kit na het uitpakken.

De kit

Het bouwpakket wordt geleverd in een simpele blauwe doos. Na het openen komen de verschillende onderdelen van de kit en een snelstart-handleiding tevoorschijn. De kit bevat de Raspberry Pi 3B+, een netadapter (die 5 V levert bij 3 A), een relais, een temperatuur- en luchtvochtigheidssensor, een luidspreker, een ReSpeaker 2-Mics Pi HAT, een bodemplaat met bevestigmateriaal en een SD-kaart met vooraf geïnstalleerde software.

Hiermee hebben we alles wat nodig is voor een eigen kleine spraakassistent. Bij het opbouwen moeten de juiste afstandsbus en schroeven worden gebruikt. Helaas vermeldt de handleiding niet welke schroeven waar moet komen. Maar met wat geduld en proberen is dat gelukkig wel uit te voegen. Verder blijkt de plaats van de RPi niet erg handig als u de SD-kaart wilt insteken.

De geperforeerde bodemplaat

Als basis voor de hardware wordt een geperforeerde bodemplaat meegeleverd waar de diverse onderdelen op vastgeschroefd kunnen worden. Dankzij het gaatjesraster kan de opbouw enigszins naar eigen inzicht worden gevareerd.

De sensoren en actuatoren

De temperatuur- en luchtvochtigheidssensor is een Sensirion SHT31 in een Grove-module. De temperatuur kan worden gemeten met een nauwkeurigheid van $\pm 0,3^\circ\text{C}$ en de relatieve luchtvochtigheid met $\pm 2\%$.

Als relais hebben we een Grove-module van Seeed Studio, gespecificeerd voor 250 VAC/30 VDC bij 5 A en met een levensduur van 100.000 schakelcycli. Vanwege de constructie van deze module kan die echter beter niet worden gebruikt voor de netspanning, omdat eventueel onder spanning staande delen niet onaantrekbaar zijn gemonteerd. Maar zolang we ons tot

veilige laagspanningen ($<25\text{ VAC} / <30\text{ VDC}$) beperken en binnen de relaisparameters blijven, kan met het relais worden geëxperimenteerd.

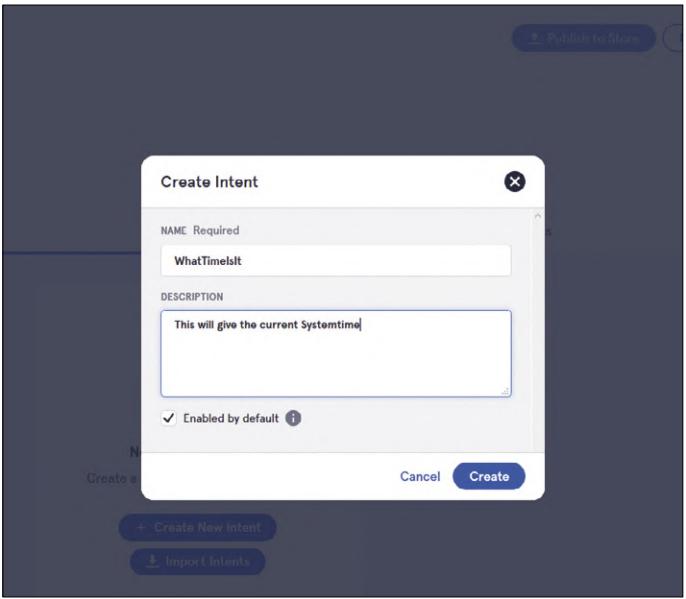
Om ervoor te zorgen dat het systeem ook zichzelf verstaanbaar kan maken, is voorzien in een kleine luidspreker. Hiervan moet geen hifi-geluid worden verwacht, hij is slechts bedoeld voor feedback vanuit de spraakassistent.

ReSpeaker 2-Mics Pi HAT

Naast de RPi vormt de opsteekprint ReSpeaker 2-Mics Pi HAT het hart van het geheel. Deze is opgebouwd rond een WM8960 audio-codec. Helaas wordt deze nog niet rechtstreeks vanuit Raspbian ondersteund en de driver bevat nog enige bugs. Daarom moet die driver, als het shield met een maagdelijk Raspbian-image wordt gebruikt, zelf voor de Raspberry Pi worden vertaald. Onder [1] is beschreven hoe de driver aan het systeem kan worden gekoppeld. De WM8960 integreert een klasse-D eindtrap met 1 W in $8\ \Omega$ per kanaal en bevat tevens de elektronica voor de microfoons. Op het board zijn twee microfoons gemonteerd en verder zijn aansluitingen voor luidspreker en hoofdtelefoon beschikbaar.

Op de print zitten ook drie RGB-LED's die via SPI met de Raspberry Pi kunnen worden verbonden. Hiermee kan worden gesigneerd dat de spraakassistent begint met luisteren, of iets niet heeft begrepen. Voor interactie met het systeem is ook een drukknop beschikbaar. Tot slot zijn er nog aansluitingen voor het relais en de temperatuur- en luchtvochtigheidssensor. Om de voedingsspanning makkelijk te kunnen aansluiten is aan een zijde van de Raspberry Pi een micro-USB connector gemonteerd. Helaas kunnen op de HAT geen andere HAT's worden geplaatst, zodat het toevoegen van bijvoorbeeld een SPI-TFT-display niet mogelijk is.

Bij het aansluiten van het relais en de temperatuur- en luchtvochtigheidssensor moet erop worden gelet de kabel met de



Figuur 2. Aanmaken van een Intent (spraakcommando).

juiste poort te verbinden. Als alles is samengebouwd en de SD-kaart aangebracht, zijn we klaar voor een eerste inbedrijf-name. Daarbij is het handig ook een HDMI-monitor en een USB-toetsenbord aan te sluiten om het bootproces van de RPi te kunnen volgen.

Na het booten is het systeem klaar voor een eerste test. Na de zin "Hey Snips – Whats the temperature?" retourneert het systeem de actuele temperatuur, met "Hey Snips – Turn Relay on" wordt het relais ingeschakeld. Dit moet werken zonder zelf iets te hoeven configureren. Maar nu wordt het echt interessant: zelf toepassingen en commando's realiseren met de spraakherkenning.

Listing 1. Weergave van status en versienummers van de diverse Snips-bestanddelen.

```
Connected to device snips-base.local
OS version ..... Raspbian GNU/Linux 9 (stretch)
Installed assistant .. MakerKitBundle_EN
Language ..... en
Hotword ..... hey_snips
ASR engine ..... snips
Status ..... Live

Service status:

snips-analytics ..... 0.60.8 (running)
snips-asr ..... 0.60.8 (running)
snips-audio-server ... 0.60.8 (running)
snips-dialogue ..... 0.60.8 (running)
snips-hotword ..... 0.60.8 (running)
snips-nlu ..... 0.60.8 (running)
snips-skill-server ... 0.60.8 (running)
snips-tts ..... 0.60.8 (running)
```

Configuratie

Nu heeft de Raspberry Pi ook toegang tot ons netwerk nodig, wat het makkelijkste gaat met een kabel. Als we WiFi willen gebruiken, moet de SD-kaart uit de Raspberry Pi worden gehaald, waarna in de drive *boot* het bestand *wpa_supplicant.conf* wordt aangemaakt. De inhoud hiervan moet als volgt luiden:

```
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev
update_config=1
country=<your_country_id>
network={
    ssid=<your_ssid>
    psk=<your_password>
}
```

De velden Country, SSID en Password moeten worden gewijzigd in overeenstemming met het eigen WiFi-LAN.

Via *snips-base.local* kan via SSH met de Raspberry Pi verbinding worden gemaakt, waarna we een beetje in het systeem kunnen rondkijken. De Snips-software is al op het systeem voorgeïnstalleerd.

Nadat we de zeer eenvoudige basisconfiguratie van de Raspberry Pi hebben voltooid, installeren we *Sam*. Sam is een commandoregel-interface waarmee de Snips-software op de Raspberry Pi kan worden bestuurd en beheerd. Om Sam te installeren, moet we een terminal openen en `sudo npm install -g snips-sam` invoeren. Nu kunnen we met `sam connect snips-base.local` verbinding maken met de Snips-software op de Raspberry Pi. Met `sam status` kunnen we nu status en versienummer van de diverse bestanddelen opvragen. Het resultaat moet eruit zien zoals in **listing 1**.

Omdat de software nog essentiële updates ontvangt, kan het zijn dat de ene of andere versie niet goed werkt. Om de meest actuele update te verkrijgen, voeren we `sam update` uit. Daarmee downloadt de Raspberry Pi de meest actuele versie en installeert hem aansluitend. Daarna moeten we nog `sam update-assistant` uitvoeren. Daarmee worden ook de applicaties geactualiseerd en moeten alle onderdelen van Snips weer online zijn. Als we nu `sam status` ingeven, moeten alle Snips-onderdelen draaien (met uitzondering van *snips-analytics*) en moet de Raspberry Pi reageren op de voorbeeldcommando's uit de handleiding.

Eigen toepassing

Na het aanmaken van een eigen account [2] kunnen we met een web-interface onze eigen applicatie bouwen en nog beter: de assistent trainen met onze eigen spraakcommando's. Als kleine demonstratie moet onze assistent de actuele tijd uitspreken. We maken eerst een nieuwe assistent aan, die we de naam *WhatTimeItIs* geven. Binnen een assistent kunnen nu apps worden aangemaakt. We maken een app aan met de naam '*currentTime*'. Deze app is verantwoordelijk voor het uitspreken van de actuele tijd.

Om de app een spraakcommando te kunnen laten verstaan, moeten we dat als *intent* (kort voor: Intention) aanleren (**figuur 2**). Er zijn een paar ingebouwde modules die het ons daarbij makkelijk kunnen maken, als we bijvoorbeeld getallen of een datum uitspreken. Omdat onze sleutelwoorden voor het

opvragen van de tijd niet als kant-en-klare module beschikbaar zijn, moeten we die zelf aanmaken. We willen dat de app luistert naar 'Time', en naar een paar synoniemen.

Als we het aanleren hebben voltooid, kijken we naar de acties die door ons spraakcommando moeten worden uitgevoerd. Daarvoor gaan we naar het punt *Actions* en kiezen voor onze demo *Code Snippets*. Op dit moment kunnen we via de web-interface Python-3- en Python-2-code schrijven (**figuur 3**) en als 'Action' laten uitvoeren. Voor verdere details verwijzen we graag naar de documentatie van Snips [3].

Voor het uitspreken van de tijd gebruiken we de Python-3-code in **listing 2**.

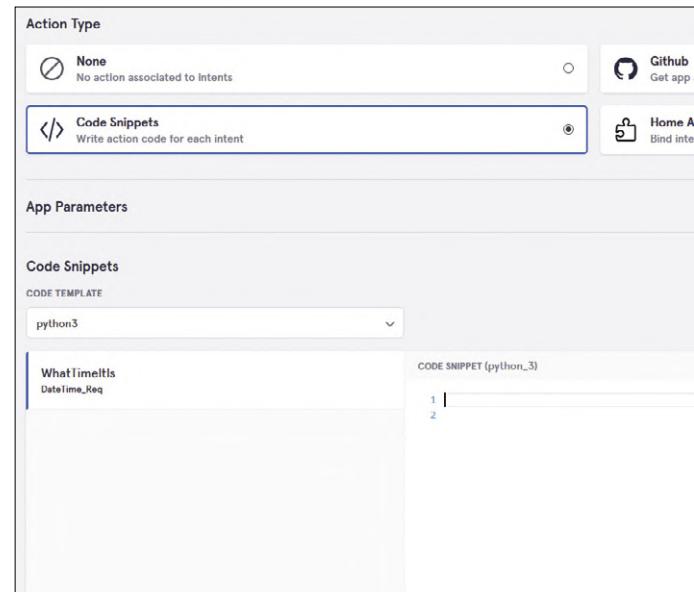
Deze code wordt uitgevoerd als het woord 'Time' of één van de synoniemen wordt herkend. Als antwoord bouwen we een string op, die dan als Tekst-To-Speech-module naar de software wordt gestuurd en ons de tijd voorleest.

Daarmee is onze eerste applicatie gebouwd en kunnen we die gaan testen. Daarvoor moeten we hem eerst naar de Raspberry Pi overzetten. In onze Pi geven we daartoe `sam install assistent` en laten de zojuist gebouwde assistent installeren. En daarna kunnen we met "Hey Snips, what's the time" naar de tijd vragen.

Samenvattend

De kit bevat alles om een eerste begin te maken en te gaan experimenteren: luidspreker, microfoons, sensor en relais. Helaas hebben de beide ingebouwde microfoons wat last van ruis. Dat stoort weliswaar de spraakherkenning niet, maar wie conform de handleiding een nieuw sleutelwoord wil aanleren om de spraakassistent te activeren, zal daarvoor een andere microfoon moeten gebruiken. Het samenbouwen is onnodig moeilijk gemaakt omdat is bezuinigs op één of twee pagina's illustraties voor het plaatsen van de afstandsbusen.

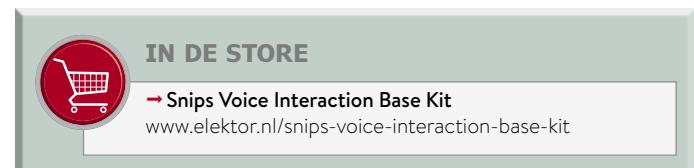
Dat de officiële Raspbian-versie op dit moment geen driver voor de soundcard bevat is niet zo fraai. Om een eigen image op basis van Raspbian te kunnen maken, moet de driver eerst zelf worden gecompileerd en gepatcht. Ook bij het actualiseren van de software zijn bepaalde stappen in acht te nemen. Als een nieuwe versie van Snips wordt geïnstalleerd, dan moet de assis-



Figuur 3. Invoeren van de code voor het uitvoeren van een actie.

tent daarna ook worden geactualiseerd en nieuw geïnstalleerd. Maar, dat gezegd zijnde, is het na enige gewenning een leuk systeem om mee te werken en de diverse mogelijkheden te gebruiken om het naar eigen wens aan te passen. Maar vooral ben ik blij dat ik baas blijf over mijn eigen data. ■

(180208-03)



Listing 2. Python-code van onze eigen applicatie.

```
import time
now = time.localtime()
if len(intentMessage.slots.DateTime_Req) > 0:
    date_or_time = intentMessage.slots.DateTime_Req.first().value # We extract the value from the slot
    result_sentence = "Current time is : {} {}".format(str(now.tm_hour), str(now.tm_min))
                                # The response that will be said out loud by the TTS engine.
else:
    result_sentence = "Time is running out"
current_session_id = intentMessage.session_id
hermes.publish_end_session(current_session_id, result_sentence)
```

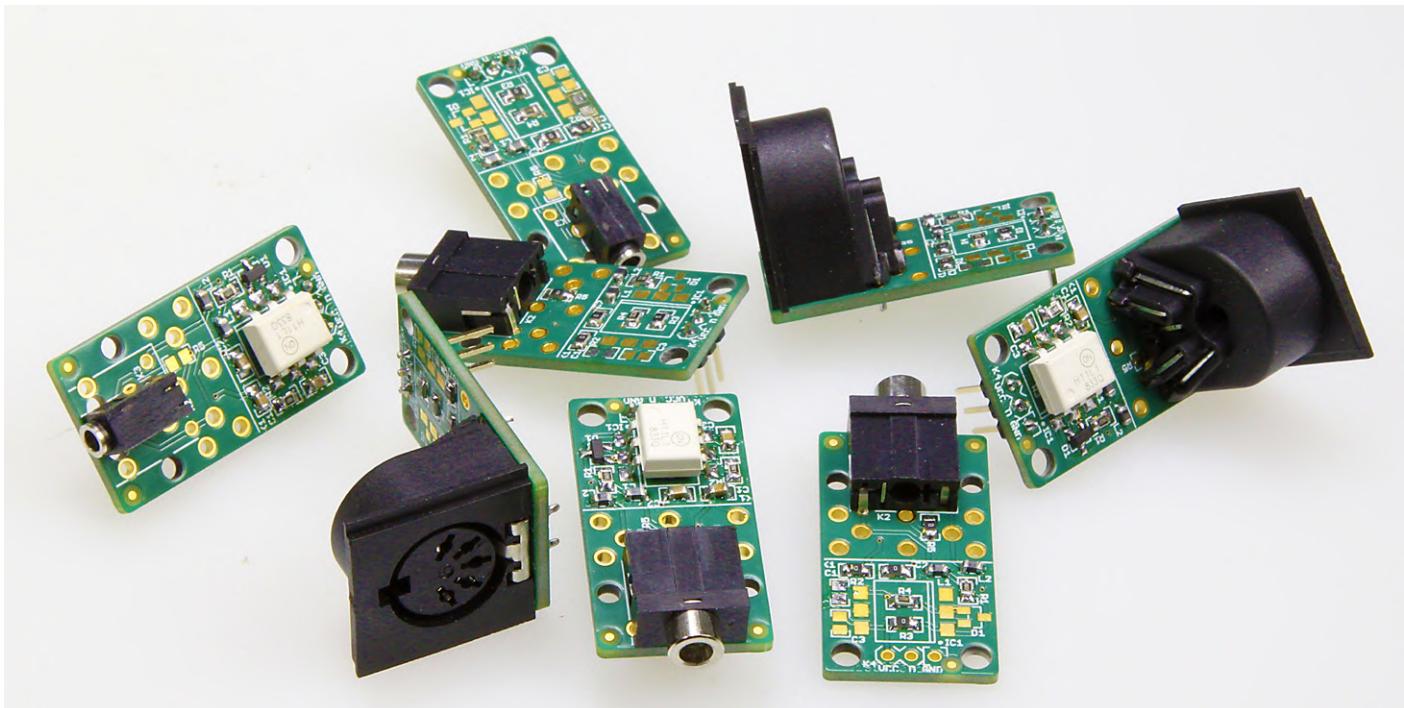
Weblinks

- [1] Installeren van de audiodriver: http://wiki.seeedstudio.com/ReSpeaker_2_Mics_Pi_HAT/
- [2] Snips-homepage: <https://snips.ai/>
- [3] Documentatie: <https://docs.snips.ai/articles/console/actions/actions/snippets>

MIDI I/O Break-out-Board

met ondersteuning voor DIN- en TRS-connectoren

Clemens Valens (Elektor Labs)



Begin 2019 is, zo'n 35 jaar na de eerste introductie, versie 2.0 van de Musical Instrument Digital Interface, beter bekend als MIDI, aangekondigd. MIDI 2.0 biedt veel nieuwe functionaliteit en ook twee nieuwe connectoren. Er is dus behoefte aan een nieuw universeel break-out-board (BoB) voor de MIDI-interface, en dat presenteren we hier.

Eigenschappen

- Eén print voor MIDI In, MIDI Out en MIDI Thru
- Configureerbaar voor 5V- en 3,3V-systemen
- Ondersteunt DIN-, 3,5mm TRS- en 2,5mm TRS-connectoren
- Voldoet aan de nieuwste MIDI-standaard

weten, is dat de nieuwe standaard 100% compatibel zal blijven met MIDI 1.0.

TRS? Is dat een "jack"?

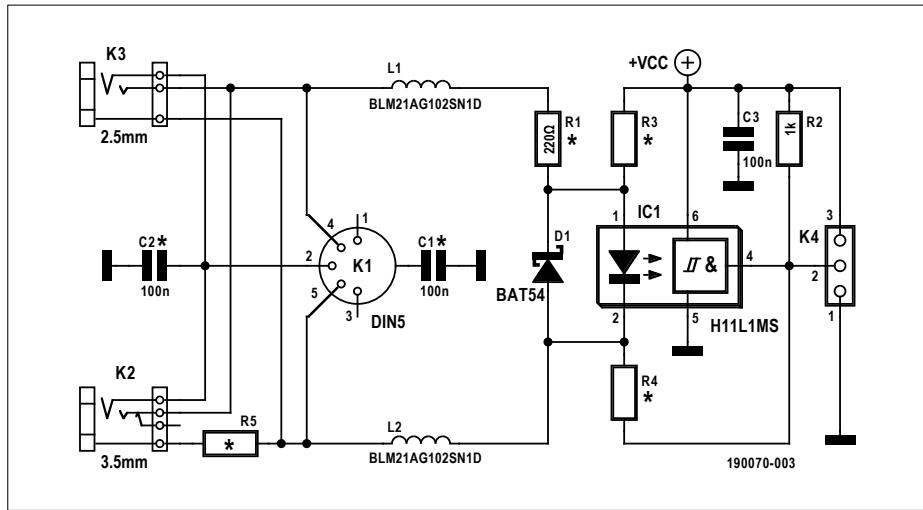
Eerlijk gezegd is specificatie van de nieuwe MIDI-connector al vrijgegeven in de zomer van 2018, hij is daarom pre-MIDI 2.0. In een *Letter of Agreement for Recommended Practice* is de Tip-Ring-Sleeve (TRS)-connector aanvaard als de MIDI-connector/plug, en de manier om die te koppelen met de traditionele 5-polige DIN-connector/plug is gestandaardiseerd. De meeste mensen kennen TRS-connectors als 'audio jacks'. Eigenlijk zijn de twee nieuwe connectoren hetzelfde, maar er zijn twee verschillende

diameters: 3,5 mm en 2,5 mm. Die laatste het aanbevolen type. De motivatie voor de toevoeging aan de standaard is heel eenvoudig, het gaat om de afmetingen: de vermaarde DIN-plug is te groot om in moderne, platte apparatuur zoals smartphones te passen.

Nieuw, nieuwer, nieuwst!

Het nieuwe MIDI-break-out-board (BoB) dat we hier presenteren kan worden voorzien van elk van de drie connectortypes: DIN, TRS-3,5 of TRS-2,5. Verder ondersteunt het BoB HF-aarding en 3,3V-signalen. De print kan worden geconfigureerd als MIDI In of MIDI Out (of MIDI Thru, dat is hetzelfde als MIDI Out).

Op het moment dat we dit artikel schrijven (april 2019) is de MIDI 2.0-specificatie nog niet vrijgegeven. We gaan daarom hier niet in op de details. Wat we al wel



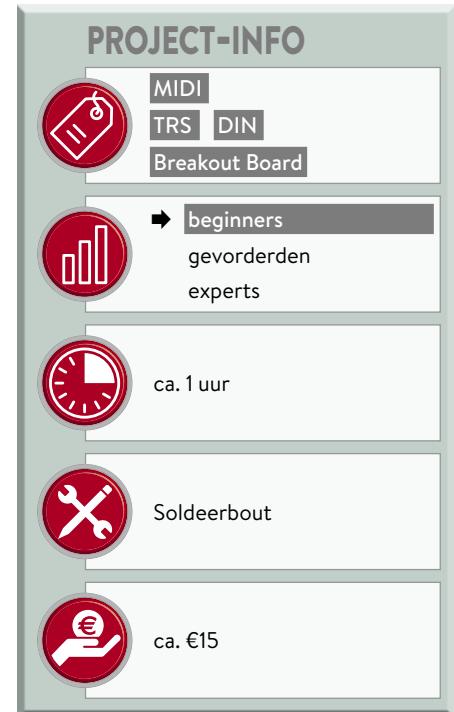
Figuur 1: Dit ietwat onoverzichtelijke schema van het MIDI I/O BoB is het resultaat van het combineren van verschillende schakelingen tot één configurerbaar geheel.

Het schema van de MIDI I/O BoB is te zien in **figuur 1**. Vanwege de configuratiemogelijkheden en de EMI/EMC- en HF-maatregelen, zijn er wat meer componenten dan in een normale MIDI-interface.

Aardlussen versus EMC

De originele MIDI 1.0-specificatie stelde dat voor het vermijden van aardlussen tussen apparaten "De afscherming van de MIDI-connectoren niet mocht worden verbonden met de massa van de schakeling of met het chassis." Maar in 2014 werd deze eis afgezwakt en werd een optionele verbinding tussen de afscherming van MIDI Out- en MIDI Thru-connectoren en de massa toegestaan om de storingsgevoeligheid te minderen. De afscherming van MIDI In-connectoren mag worden geaard via een kleine condensator, maar nooit direct. Daarom maakt ons MIDI I/O BoB deze optionele aarding mogelijk met behulp van C1 en C2. Als de kaart wordt gebruikt als MIDI In, kunnen C1 en C2 worden

vergroot of *met het chassis.*" Maar in 2014 werd deze eis afgezwakt en werd een optionele verbinding tussen de afscherming van MIDI Out- en MIDI Thru-connectoren en de massa toegestaan om de storingsgevoeligheid te minderen. De afscherming van MIDI In-connectoren mag worden geaard via een kleine condensator, maar nooit direct. Daarom maakt ons MIDI I/O BoB deze optionele aarding mogelijk met behulp van C1 en C2. Als de kaart wordt gebruikt als MIDI In, kunnen C1 en C2 worden



weggelaten, of u kunt condensatoren van 100 nF plaatsen; voor een MIDI Out (of Thru) mogen C1 en C2 0Ω-weerstanden zijn.

Vermijden van HF-storing

De ferrietkralen L1 en L2 verzwakken HF-storingen maar kunnen voor het gemak worden vervangen door 0Ω-weer-

ONDERDELENLIJST

MIDI In	
Weerstanden (SMD 0805):	
R1 = 220 Ω	
R2 = 1 kΩ	
R5* = 0 Ω	
Condensatoren (SMD 0805):	
C1,C2,C3 = 100 n	
Spoelen (SMD 0805):	
L1,L2 = ferrietkraal, 1 kΩ @ 100MHz	
Halfgeleiders:	
D1 = BAT54	
IC1 = H11L1 (of equivalent)	
Diversen:	
K1* = 5-polige DIN-socket, 180°	
K2* = 3-polige TRS-socket, 3,5 mm Ø	
K3* = 3-polige TRS-socket, 2,5 mm Ø	
K4 = 3-polige pinheader, raster 0,1"	
Print # 190070-1	

MIDI Out / MIDI Thru, 5 V

Weerstanden (SMD 0805):

R1,R4 = 220 Ω

R3,R5* = 0 Ω

Condensatoren (SMD 0805):

C1,C2 = 0Ω* (!)

C3 = 100 n

Spoelen (SMD 0805):

L1,L2 = ferrietkraal, 1 kΩ @ 100 MHz

Diversen:

K1* = 5-polige DIN-socket, 180°

K2* = 3-polige TRS-socket, 3,5 mm Ø

K3* = 3-polige TRS-socket, 2,5 mm Ø

K4 = 3-polige pinheader, raster 0,1"

Print # 190070-1

MIDI Out / MIDI Thru, 3,3 V

Weerstanden (SMD 0805):

R1 = 33 Ω, 0,5 W

R3,R5* = 0 Ω

R4 = 10 Ω, 0,25 W

Condensatoren (SMD 0805):

C1,C2 = 0 Ω* (!)

C3 = 100 n

Spoelen (SMD 0805):

L1,L2 = ferrietkraal, 1 kΩ @ 100 MHz

Diversen

K1* = 5-polige DIN-socket, 180°

K2* = 3-polige TRS-socket, 3,5 Ø

K3* = 3-polige TRS-socket, 2,5 Ø

K4 = 3-polige pinheader, raster 0,1"

Print # 190070-1



standen (afhankelijk van wat u in voorraad hebt, bijvoorbeeld).

R3 en R4 zijn configuratieweerstanden en zijn alleen nodig voor MIDI Out- (en Thru)-toepassingen. Om te voorkomen dat iemand ze per ongeluk toch plaatst op een MIDI In-module, zitten ze op dezelfde plaats als optocoupler IC1.

Voor MIDI In is een optocoupler nodig

De optocoupler zorgt voor een goede signaalqualiteit en een geringe latentietijd tussen In en Thru bij het aaneenrijgen van meerdere MIDI-apparaten. Diode D1 beschermt hem tegen signalen met een verkeerde polariteit.

R2 is de uitgangsweerstand van de open collector-uitgang van IC1. Het is mogelijk om de snelheid iets te vergroten door voor R2 een lagere waarde te kiezen, maar dat leidt wel tot een hoger stroomverbruik. IC1 heeft een voedingsspanning van minimaal 3 V nodig, dus dat is de laagste toegestane waarde voor VCC voor de MIDI In-configuratie (de maximale waarde voor IC1 is niet minder dan 16 V).

C3 zorgt voor wat ontkoppeling van de voedingsspanning. Het is een goede gewoonte om de voedingsspanning altijd te ontkoppelen, al is het in dit geval misschien niet echt nodig. Van de andere kant: wie weet wat de gebruikers voor aansluitdraden op K4 gaan gebruiken (hoe koper, hoe beter natuurlijk).

Weerstand R5, een 0Ω-weerstand, dient alleen om de afscherming van K2 te verbinden met die van K1, zonder alles onbedoeld aan GND te leggen. Hij moet alleen worden geplaatst als K2 ook geplaatst is.

Configuratie voor MIDI Out en MIDI Thru

Voor een MIDI Out- of MIDI Thru-module zijn minder onderdelen nodig, want IC1, D1 en R2 zijn niet nodig. Voor deze configuraties is R3 een 0Ω-weerstand die R1

verbint met VCC. R4 begrenst de uitgangsstroom op de MIDI-output. De waarden van R1 en R4 zijn afhankelijk van VCC. Voor een voedingsspanning van 5 V krijgen ze beide een waarde van 220 Ω; dat is de traditionele MIDI Out-poort voor 5V-systemen. Maar voor 3,3V-systemen moeten de waarden als volgt worden aangepast.

Berekening voor het werken met een lage spanning

Bij de introductie van de MIDI-standaard werd een stroom van 5 mA gespecificeerd voor de LED in de optocoupler aan de MIDI-input om er zeker van te zijn dat die goed zou oplichten. Moderne optocouplers hebben meestal niet zoveel stroom nodig (de onze is tevreden met 1,6 mA), maar omdat een moderne MIDI Out nog steeds een klassieke MIDI In moet kunnen aansturen, is 5 mA wat we moeten leveren.

Een MIDI-input ziet eruit als een weerstand van 220 Ω in serie met een LED. De MIDI-standaard gaat er van uit dat die LED een doorlaatspanning van maximaal 1,9 V kan hebben. Dat betekent dat de uitgangsspanning V_{TX} tenminste gelijk moet zijn aan:

$$V_{TX} \geq 0,005 \times 220 + 1,9 = 3,0 \text{ V}$$

Het is nodig om de stroom te begrenzen in geval van kortsluiting. Dat is de functie van R4. R1 en R4 samen (vergeet niet dat R3 = 0 Ω) veroorzaken samen een spanningsval die moet overeenkomen met V_{TX} . Dus:

$$V_{TX} - 3,0 \geq 0,005 \times (R1+R4)$$

Als $V_{TX} = 3,3 \text{ V}$, dan kunnen we berekenen:

$$R1 + R4 = 0,3 / 0,005 = 60 \Omega$$

De MIDI Manufacturers Association stelt voor: R1 = 33 Ω en R4 = 10 Ω, een totaal van 43 Ω. Dat geeft ruimte voor allerlei toleranties bij weerstandswaarden en voedings- en uitgangsspanningen.

Pas op voor kortsluitingen

Een detail dat gemakkelijk over het hoofd wordt gezien, is de stroom door R1 en R4 als de aangesloten MIDI-input defect is en een kortsluiting naar GND vormt:

$$V_{CC} / R1 = 3,3 / 33 = 0,1 \text{ A}$$

$$P_{R1} = (0,1)^2 \times 33 = 330 \text{ mW}$$

Omdat R1 is verbonden met de voeding van de zender, waarvan we kunnen aannemen dat die krachtig genoeg is om deze 100 mA te leveren, moet R1 geschikt zijn om 330 mW te dissiperen. Het moet dus een 0,5W-weerstand zijn. Een dergelijke redenering geldt voor R4 (dan zou een type van 1,2 W nodig zijn), maar in dit geval mogen we aannemen dat de weerstand wordt aangestuurd door een open collector- of open drain-output met een pullup-weerstand. Die weerstand zal (typisch) een waarde van een paar honderd ohm hebben, zodat de stroom beperkt wordt. Daarom volstaat voor R4 een weerstand van 0,25 W. Pas dus op met het aansturen van de 3,3-V MIDI Out vanuit een digitale driver. Deze beperking geldt niet voor de 5-V MIDI Out.

Tenslotte...

Uit het bovenstaande blijkt wel dat er, zelfs bij eenvoudige schakelingen, van alles mis kan gaan, vooral als we apparatuur uit verschillende bronnen en tijdsperken met elkaar proberen te verbinden. NB: een compleet overzicht van de componenten voor de verschillende configuraties van het MIDI I/O BoB is te vinden op [1]. ◀

(190070-03)

Weblinks

- [1] MIDI I/O BoB bij Elektor Labs:
www.elektormagazine.com/labs/midi-io-break-out-board



IN DE STORE

→ MIDI IO Breakout Board, geconfigureerd als input – gedeeltelijk opgebouwde module:
www.elektor.nl/190070-91

→ MIDI IO Breakout Board - kale print:
www.elektor.nl/190070-1



Wat wilde u worden toen u jong was?

Dankzij uw ambitie heeft u toegang tot de razendsnel veranderende wereld van engineering, waarin uw ideeën de toekomst vormen. Maar om te doen wat u het beste doet, heeft u tijd nodig om zich te concentreren op wat echt belangrijk is.

Wij helpen klanten al 80 jaar hun doelen te bereiken met ons uitgebreide aanbod procesautomatiserings-, onderhouds- en elektronica-producten, en op u afgestemde services.

We're here

for the **inspired**

Deskundige technische ondersteuning | Toonaangevende merken | Levering de volgende werkdag



Een nieuwe component (symbool) in KiCad maken

en voorbereiden voor koppeling aan een bestaande footprint

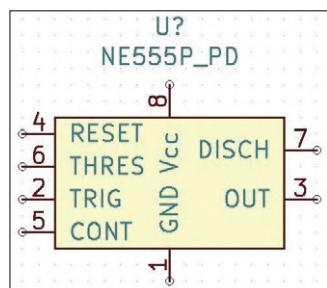
Dr. Peter Dalmaris (Australië)



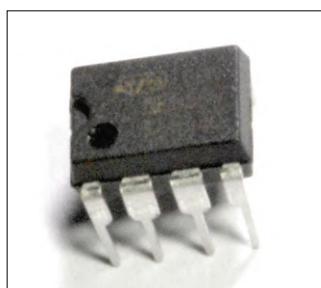
Opmerking van de redactie: dit artikel presenteert een voorbeeld van een hoofdstuk uit het boek "Kicad Like a Pro" door dr. Peter Dalmaris, omge-werkt naar de pagina-layout van ElektorLabs. Het boek is verkrijgbaar in onze shop.

Waarschijnlijk wilt u een eigen schemasymbool maken omdat u een fysiek onderdeel hebt waarvoor u geen bestaand passend symbool kunt vinden, noch in de met KiCad meegeleverde symbolen, noch op internet.

Ook is het waarschijnlijk dat u uw nieuwe symbool aan een footprint wilt koppelen. Als uw onderdeel een standaard behuizing heeft (zoals DIP), kunt u het nieuwe symbool aan een bestaande footprint koppelen. Zo niet, dan zult u ook een eigen footprint moeten maken. Het boek wijdt hier een apart hoofdstuk aan. In deze handleiding leert u een eigen symbool te ontwerpen voor het 555 timer-IC. Natuurlijk zijn er meer dan genoeg KiCad-libraries die dit symbool al bevatten, maar laten we voor ons leerproces aannemen dat u het niet hebt kunnen vinden. Wat we willen maken is een symbool zoals dat in **figuur 35.1**. Ons doel is een symbool te genereren dat aan de conventies voldoet. In geval van een IC-symbool:



Figuur 35.1: Een zelfgemaakt symbool voor het 555-IC.

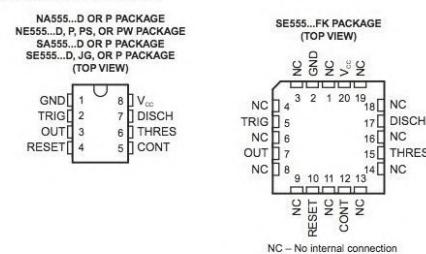


Figuur 35.2: Ons onderdeel is een 555-timer in een 8-pins DIP-behuizing.

In deze handleiding leren we u een eigen component (ook wel symbool genaamd) te creëren. In schema's worden symbolen gebruikt die in Eeschema kunnen worden aangemaakt. Veel van wat hier wordt beschreven, kan ook worden gebruikt om bestaande symbolen aan te passen. In het boek staat overigens een aparte handleiding hoe dat in zijn werk gaat.

- we plaatsen pinnen rondom een rechthoek;
- we groeperen pinnen naargelang hun functie (zoals ingangen, uitgangen, voeding etc.);
- we plaatsen de VCC-pin aan de bovenzijde van de rechthoek;
- we plaatsen de GND-pin aan de onderzijde van de rechthoek;
- we kiezen een passende naam en codering voor het symbool. Symbool-coderingen zijn gestandaardiseerd; meer

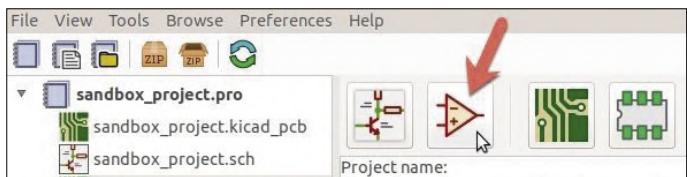
6 Pin Configuration and Functions



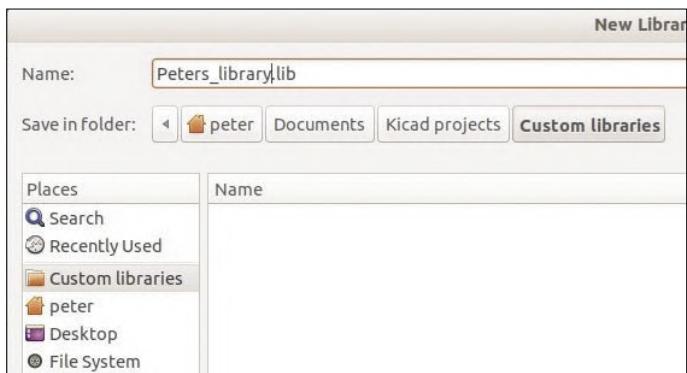
Pin Functions

NAME	PIN		I/O	DESCRIPTION
	D, P, PS, PW, JG	FK NO.		
CONT	5	12	I/O	Controls comparator thresholds. Outputs 2/3 VCC, allows bypass capacitor connection
DISCH	7	17	O	Open collector output to discharge timing capacitor
GND	1	2	-	Ground
NC	1, 3, 4, 6, 8, 9, 11, 13, 14, 16, 18, 19	-	-	No internal connection
OUT	3	7	O	High current timer output signal
RESET	4	10	I	Active low reset input forces output high and discharge low
THRES	6	15	I	End of timing input. THRES > CONT sets output low and discharge open
TRIG	2	5	I	Start of timing input. TRIG < 1/2 CONT sets output high and discharge open
Vcc	8	20	-	Input supply voltage, 4.5 V to 16 V. (SE555 maximum is 18 V)

Figuur 35.3: Pinning en pinfuncties uit de datasheet van het IC.



Figuur 35.4: Start de Symbol Editor.



Figuur 35.5a: Ik heb mijn symbool in een directory met de naam 'Custom libraries' gezet.

hierover in [1].

Het onderdeel waarvoor we een symbool willen maken is te zien in **figuur 35.2**. De component heeft een DIP-behuizing met 8 pinnen.

Omdat we een symbool gaan maken, hoeven we geen andere 'tastbare' eigenschappen van het onderdeel te kennen dan het totaal aantal pinnen van de behuizing. Voor het maken van een footprint moeten we die fysieke eigenschappen natuurlijk wel kennen.

Niettemin is het een voordeel als we de beschikking hebben over de datasheet van het onderdeel. Deze bevat informatie over naam, nummer en functie van elke pin en of het een ingang, een uitgang of een voedingsaansluiting is, of die bidirectioneel is, enzovoort. Hoe meer van dit soort nuttige informatie u hebt, hoe beter.

De datasheet voor ons voorbeeldonderdeel is beschikbaar bij de fabrikant. De benodigde informatie staat op bladzijde 6, die voor het gemak in **figuur 35.3** is gereproduceerd.

We beginnen nu met het proces om een nieuw symbool te creëren. Klik in het hoofdscherm van KiCad op de knop Symbol Library Editor (**figuur 35.4**).

Elk symbool moet worden opgeslagen in een library-bestand, dus voordat we beginnen met het creëren van het symbool maken we eerst een nieuwe library aan. Dat doen we door te klikken op de knop 'New Library', of via het File-menu (**figuur 35.5a**).

KiCad vraagt nu of deze library beschikbaar moet zijn voor alle projecten ('Global') of alleen binnen het huidige project ('Project'). Maak hier uw eigen keuze (ik koos voor 'Global'). Vanaf nu werkt u in de nieuwe library waarin u het nieuwe symbool opslaat. U kunt dit controleren bovenin het Editor-window, waar pad en naam van de zojuist gecreëerde library te zien moeten zijn.

Klik nu in de bovenste toolbar op de knop 'Create New Symbol'. Er wordt gevraagd de library te selecteren waarin u uw sym-

bool wilt opslaan. Daar moet ook de zojuist aangemaakte library bij staan. Klik hierop en vervolgens op 'OK' om verder te gaan. Nu verschijnt het Symbol Properties-venster. De belangrijkste gegevens die moeten worden ingevuld om verder te kunnen gaan zijn de symboolnaam en de codering. De naam is de typeaanduiding van het onderdeel en eventuele aanvullende gegevens om het uniek identificeerbaar te maken. Om het later te kunnen gebruiken, is het wel zo handig als het door een betekenisvolle naam makkelijk in de symbool-library is terug te vinden. Omdat ik mijn eigen 555-symbool wil kunnen onderscheiden van de al in andere libraries aanwezige 555-IC's, heb ik aan het typenummer mijn initialen 'PD' toegevoegd.

Ga voor de codering niet gokken, maar gebruik de Reference Designators-tabel op Wikipedia. Een gedeelte van deze tabel is te zien in **figuur 35.5b**. De codering voor IC's is 'U', dus typ dat in het Default Reference Designator-veld in.

figuur 35.6 toont de gegevens die ik heb ingevoerd in het Symbol Properties window. Behalve symboolnaam en -codering laten we de overige gegevens ongemoeid. Klik 'OK' om te bevestigen en ga door. KiCad plaatst symboolnaam en codering in het midden van het blad, bovenop elkaar. Gebruik de 'M'-hotkey om de twee tekstblokken te verplaatsen. Het resultaat moet er ongeveer uitzien als in het voorbeeld van **figuur 35.7**.

We tekenen nu de omtrek van het nieuwe symbool. Daarvoor kunt u zowel het veelhoek- als het rechthoek-tool gebruiken. Het resultaat moet er uitzien zoals in **figuur 35.8**.

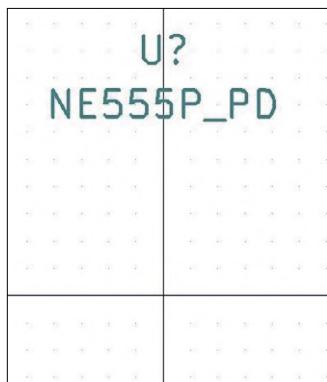
Voeg nu de achtergrondkleur

RV	Varistor
S	Switch (all types, including push-buttons)
T	Transformer
TC	Thermocouple
TP	Test point
TUN	Tuner
U	Integrated circuit (IC) 
V	Vacuum tube
VR	Variable resistor (potentiometer or rheostat)

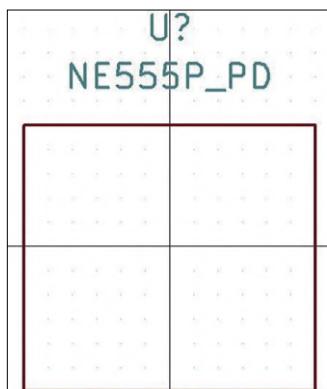
Figuur 35.5b: Een gedeelte van de Reference designators standard IEEE 200-1975/ANSI Y32.16-1975.



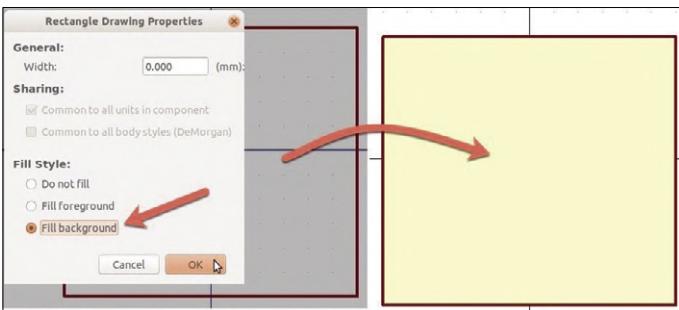
Figuur 35.6: De eigenschappen van het nieuwe symbool.



Figuur 35.7: Een leeg nieuw symbool.



Figuur 35.8: De footprint.



Figuur 35.9: Vul de rechthoek-achtergrond.

toe die ook voor andere IC's wordt gebruikt door voor de rechthoek het venster Drawing properties te openen (Plaats de muis op de rechthoek en typ 'E' in). Vink onder 'Fill Style' de optie 'Fill Background' aan (**figuur 35.9**).

Nu zijn de pinnen aan de beurt. Houd hiervoor de datasheet bij de hand, zie ook figuur 35.3. Klik op de Pin-knop in de rechter toolbalk. Plaats de pinnen rond de omtrek van het symbool, zoals in figuur 35.1. Denk aan de conventie om pinnen naar functie te groeperen en de voedingspinnen aan boven- en onderzijde te plaatsen. We beginnen met de Vcc-pin. Volgens de datasheet is dit pin 8 en volgens de conventie moet deze aan de bovenkant van de rechthoek. Klik op het Pin-tool en vervolgens op de bovenrand. Nu verschijnt het venster Pin Properties. Vul dit in zoals in **figuur 35.10**.

Nu de velden in het venster. De Pin-name moet zinvol zijn. Meestal houd ik mij aan de benaming zoals die in de datasheet wordt gehanteerd. Het Pin-number is heel belangrijk, omdat hiermee schemasymbolen en fysieke pinnummers van de footprint aan elkaar worden gekoppeld. Als u een eigen footprint ontwerpt voor dit onderdeel in de procedure 'Creating new footprint', bepaalt de waarde in het veld Pin-number welk schema-knooppunt wordt verbonden met welk soldeereilandje van de footprint. Neem het pinnummer voor de Vcc-pin uit de datasheet ('8') en zet dat in dit veld.

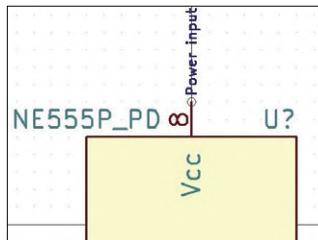
Selecteer in het drop-down-menu Orientation de optie die overeenkomt met de zijde van de rechthoek waarmee u de pin wilt koppelen. De Vcc-pin moet aan de bovenzijde en zijn ronde

connector moet van de rechthoek af wijzen. De horizontale lijn van het pin-oriëntatie-icoon represeneert de rechthoek. Als u de pin aan de linkerzijde van de rechthoek zou willen plaatsen, dan zou u het icoon moeten kiezen waarvan de ronde connector naar links wijst.

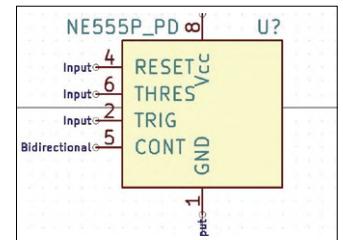
Tot slot, omdat de Vcc-pin een voedingspin is, heb ik als elektrisch type 'Power Input' gekozen. Hetzelfde geldt voor de GND-pin.

Klik op 'OK' om de wijzigingen te bevestigen. Plaats de pin in het midden van de bovenzijde van de rechthoek, zoals te zien in **figuur 35.11**. Ik heb het tekstblok verplaatst zodat het niet de pin overlapt.

Volg dezelfde procedure om de GND-pin (pin 1) aan de onder-



Figuur 35.11: De Vcc-pin op zijn plaats.



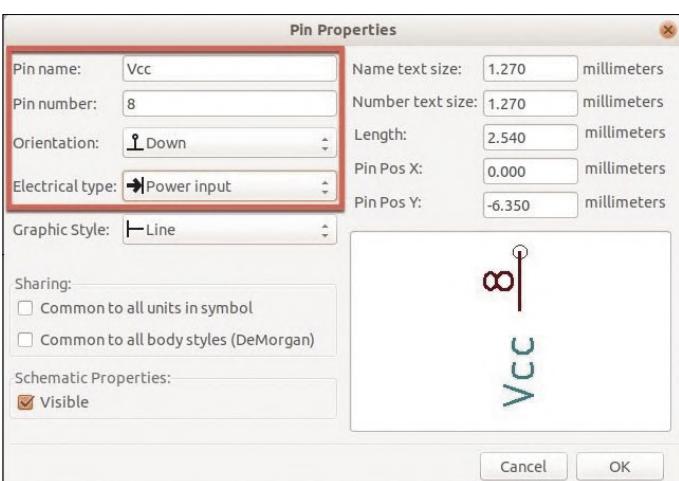
Figuur 35.12: Ingangspinnen aan de linkerzijde.

zijde toe te voegen. Kopieer pinnaam en -nummer uit de datasheet en markeer deze eveneens als voedingspin.

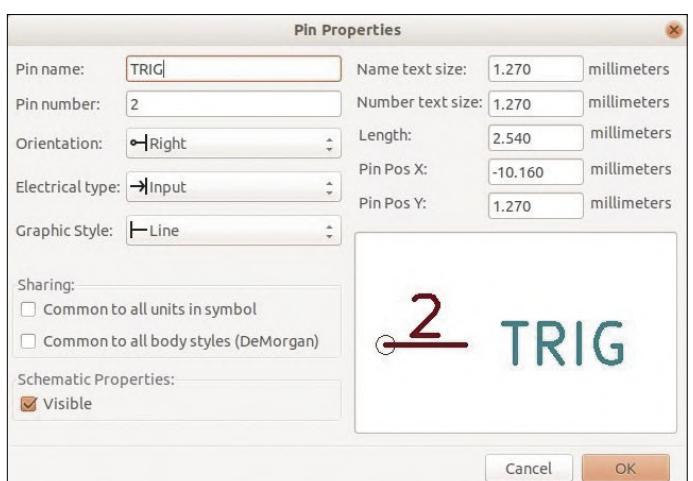
Ga verder aan de linkerzijde van de rechthoek, waar de ingangspinnen moeten komen. Volgens de datasheet zijn de ingangspinnen: RESET, THRES, TRIG. Er is één bidirectionele pin, 'CONT', die zowel aan de linker- als aan de rechterzijde kan komen. Ik heb hem links geplaatst. Uw symbol zou er nu uit moeten zien zoals in **figuur 35.12**.

De pin-attributen van de ingangspinnen aan de linkerzijde van het symbool zien eruit zoals in **figuur 35.13**. Vergeet niet dat pin 5 bidirectioneel is en het elektrisch type dus moet worden gemarkerd als 'bidirectional'.

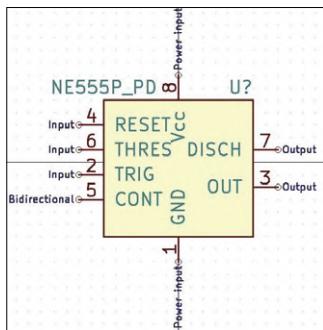
Creëer nu op dezelfde wijze de laatste twee pinnen. Volgens de datasheet zijn dit uitgangen. Gebruik de pinnamen en -nummers zoals aangegeven in figuur 35.3 voor de pinnen 7 en 3. Als dit allemaal is afgerond zou uw symbool eruit moeten zien



Figuur 35.10: De eigenschappen van de Vcc-pin.



Figuur 35.13: De ingangsattributen.

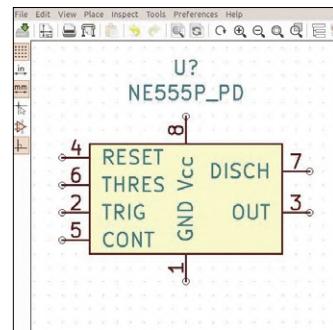


Figuur 35.14: Het voltooide eigen symbool.

de URL in het Field Value-veld (**figuur 35.15**). Klik 'OK' om de wijzigingen te bevestigen.

U bent nu klaar. Sla het symbool op in de geselecteerde library met de knop Save Current Symbol. Controleer dan of u het kunt gebruiken in Eeschema. Ga in het Preferences-menu naar Symbol Libraries en voeg de nieuwe library toe (raadpleeg de betreffende handleiding als u niet weet hoe). Plaats de cursor

zoals in **figuur 35.14**. Wat rest is het toevoegen van de URL naar de datasheet aan de symbool-eigenschappen. Dat is handig voor toekomstig gebruik. Het is zeker handig als u verder gaat met het maken van een bij het symbool behorende eigen footprint. Om de datasheet-URL toe te voegen, klik op Symbol, 'Fields...', om het window Field Properties te doen verschijnen. Klik op de datasheet-rij, selecteer het en kopieer/plak

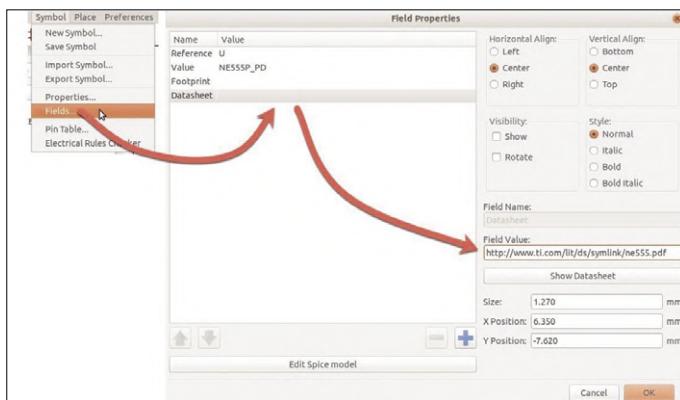


Figuur 35.17: Uw eigen nieuwe symbool in Eeschema.

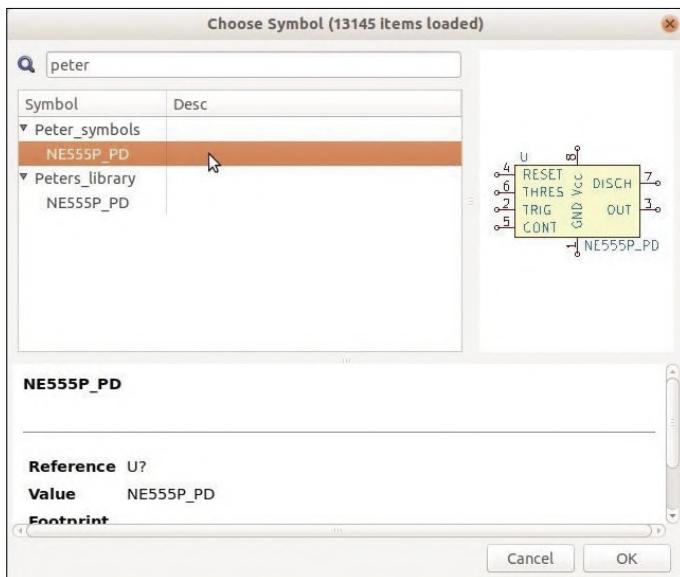
het eigen 555-symbool heeft nu zijn plaats in Eeschema en u kunt het vanaf nu net als elk ander symbool gebruiken (**figuur 35.17**).

U hebt hier geleerd een geheel nieuw symbool te creëren. Maar stel dat u een symbool hebt dat bijna gelijk is aan wat u nodig hebt, kan dit dan een heel klein beetje worden aangepast? Dat leert u in handleiding (recipe) '36. Modifying an existing component (symbol)'. ▶

(190041-03)



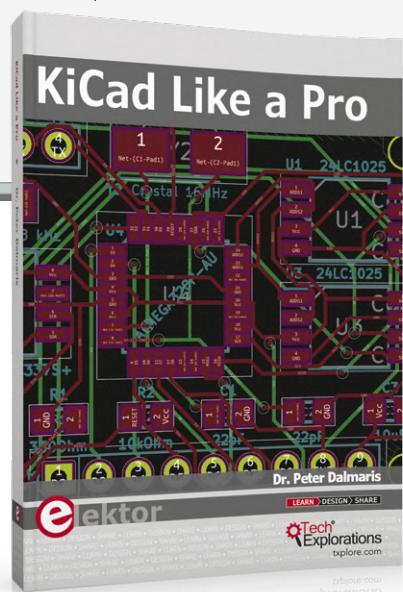
Figuur 35.15: Het is heel handig om de datasheet bij de hand te hebben.



Figuur 35.16: Vind uw nieuwe library en symbool.

@ WWW.ELEKTOR.COM
Boek (Engelstalig): "Kicad Like a Pro"
www.elektor.nl/kicad-like-a-pro

E-boek (Engelstalig): "Kicad Like a Pro"
www.elektor.nl/kicad-like-a-pro-e-book



Weblink

- [1] **Symbol designators:**
https://en.wikipedia.org/wiki/Reference_designator

Studenten in de schijnwerpers!



TUE/InMotion: revolutionair accupack-ontwerp

Foto: Johan van Uden.

Kjeld Teunissen (Nederland)

Het InMotion-studententeam van de Technische Universiteit van Eindhoven (TU/e) grijpt deze gelegenheid aan om bekendheid te geven aan hun werk aan 'Electric Refuelling', een concept dat erop is gericht om een oplossing te bieden voor het even snel 'volgooien' van elektrische auto's als van auto's met een verbrandingsmotor. Interessant? Laten we eens kijken waar ze meer bezig zijn...

Het InMotion-studententeam, waar ik deel van uitmaak, bestaat uit meer dan vijftig jonge, ambitieuze studenten van de Technische Universiteit Eindhoven en de Fontys Hogeschool voor Toegepaste Wetenschappen. Studententeams bieden een unieke mogelijkheid tot samen-

werking tussen universiteiten en industrie, zodat de studenten kunnen werken aan grensverleggende technologie. Wij werken aan een concept dat 'Electric Refuelling' wordt genoemd. Het gaat om de mogelijkheid een elektrische auto even snel op te laden als het tanken van

een wagen met verbrandingsmotor zou duren. We willen de mogelijkheden van deze techniek in 2023 demonstreren bij de **Garage 56 Class** tijdens de 24-uurs rally van Le Mans met een volledig elektrische racewagen. We moeten daarvoor een eigen op maat gemaakt accupack

Is jouw team grensverleggend bezig?

Maak je deel uit van een studententeam dat slimme of verbazende prestaties zou kunnen rapporteren op het gebied van elektronica? Stuur dan een mailtje naar Editor@elektor.com, met als onderwerp "Studenten in de schijnwerpers".



ontwikkelen en dat is mijn taak voor dit jaar. Als u zich dus ooit hebt afgevraagd

- hoe elektrische voertuigen worden gevoed,
- en hoe het mogelijk is om een voertuig te voeden met cellen die in de palm van je hand passen

dan is dit artikel iets voor u.

In de beginne...

Bij het ontwerpen van een accupack, voor welke toepassing dan ook, is de eerste stap het vastleggen van de ontwerp-eisen. De vier belangrijkste eisen zijn de (minimale, nominale en maximale) spanning, de capaciteit, het gewicht en het vermogen. Verder moet er altijd rekening worden gehouden met de veiligheid en de regelgeving.

De huidige accutechnologie stelt ons voor een dilemma. We kunnen kiezen voor een accupack met een grote capaciteit.

Daarmee kun je meer energie meenemen en kun je dus langere afstanden afleggen. Een alternatief is om te gaan voor een accupack dat eerder geschikt is voor het leveren van een groot vermogen en voor sneller opladen. Omdat de focus bij InMotion ligt op het ontwikkelen van snellaadtechnologie, hebben wij voor het laatste gekozen. We ontwikkelen dus nu een accupack met een geringere capaciteit dan momenteel wordt aangeboden in commerciële voertuigen. Niettemin stelt dit ons in staat technologie te ontwikkelen die kan worden gebruikt als de accutechnologie zover is verbeterd dat er accupacks met een groot vermogen en voldoende capaciteit voor commerciële toepassingen beschikbaar zijn.

Structuur

Een accupack bestaat uit een groot aantal kleinere accucellen. Meerdere cellen worden gecombineerd in een module, en die modules worden dan gecombineerd tot een accupack (zie ook **figuur 1**).



Het InMotion-team (foto: Bram Naus).



Figuur 1: Een voorbeeld van een accukoelpack gemaakt door Kokam [1].

Cel

De keuze van het type accucel is waarschijnlijk de belangrijkste beslissing in het proces. Het is de bepalende factor voor de prestaties van het accupack. InMotion maakt gebruik van pouch-cellen omdat die beter geschikt zijn voor snelladen dan cilindrische cellen. Ze zijn robuuster, hebben meestal een lagere

inwendige weerstand (waardoor ze minder warm worden) en ze hebben een groter oppervlak waardoor ze gemakkelijker te koelen zijn.

Module

De ontwerpvariabelen van de individuele modules worden bepaald door de vrije ruimte die nodig is rondom elke cel. Bij

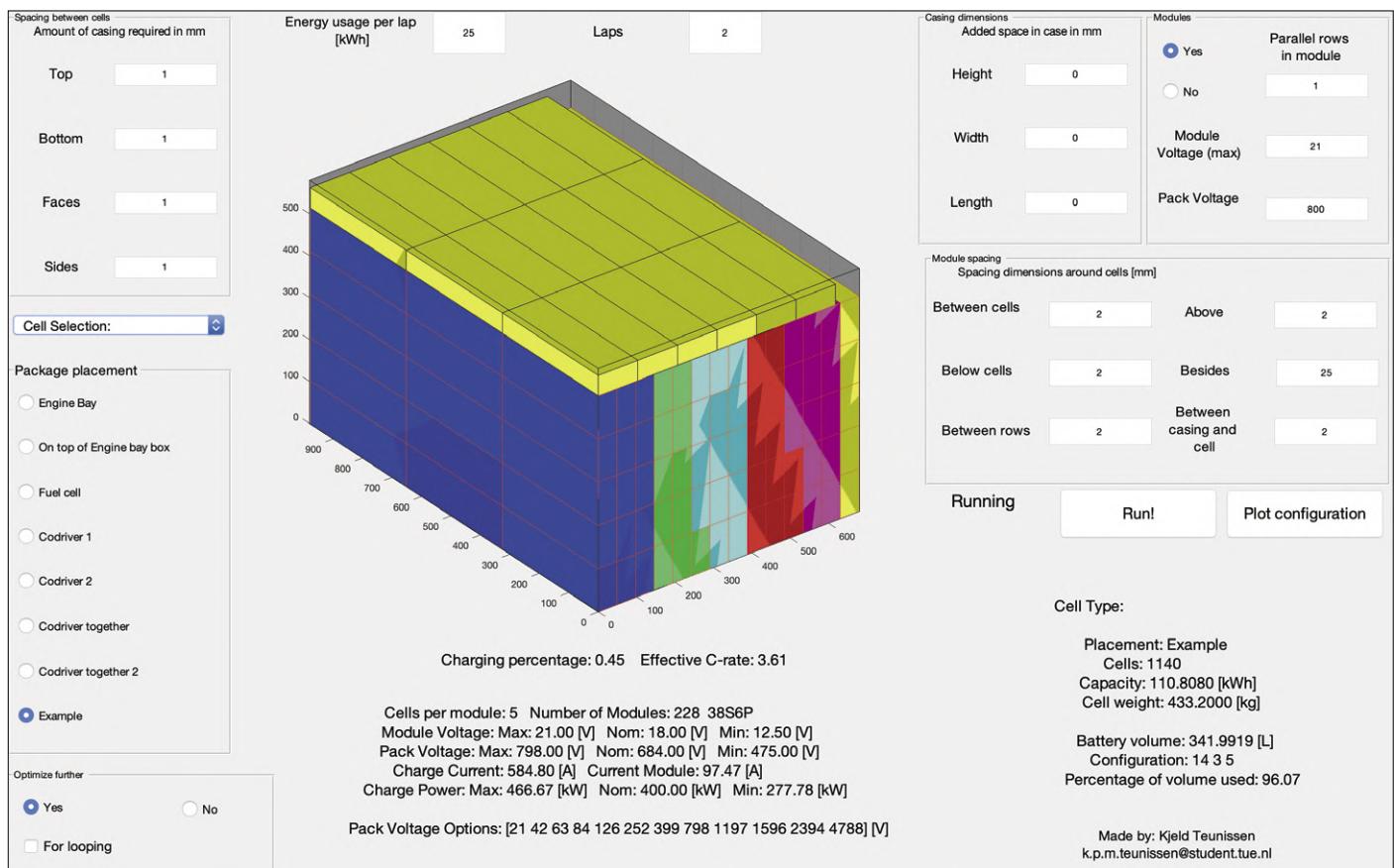
gebruik van pouch-cellen zijn de benodigde ruimtes boven, onder en naast de cellen allemaal parameters die moeten worden bepaald of onderzocht. Meestal zijn die afhankelijk van de locatie van de aansluitingen en de manier van koelen.

Pack

Als de afmetingen van de modules zijn bepaald, kan worden bepaald hoe die tot een pack moeten worden gestapeld. Dat zal bepalen waar de koelslangen moeten komen en hoe de modules elektrisch moeten worden aangesloten in het pack.

Slimme software

Het ontwikkelproces bij InMotion is erg dynamisch. Daardoor kunnen we allerlei dingen al doende veranderen. Dat betekent ook dat niet alle ontwerpeisen keihard zijn vastgelegd. We proberen een eisenpakket te vinden dat past bij onze doelen. Het accupack is daar een goed voorbeeld van, want het kan op ontelbare manieren worden geconfigureerd. Maar hoe moeten we uitvogelen welke configuratie het beste werkt? Om te voorkomen dat we telkens een nieuw model moeten genereren voor elke



Figuur 2: Een speciaal geschreven programma helpt bij het ontwerp van het accupack.

configuratie, heb ik een accupack-configurator gemaakt die het pack kan optimaliseren voor verschillende beginparameters. Om het gebruikersgemak te garanderen en de gebruiker in staat te stellen om verschillende configuraties uit te proberen zonder de onderliggende code te hoeven begrijpen, heb ik een GUI gemaakt (zie **figuur 2**). Die neemt alle ontwerpvariabelen als input en probeert dan zo veel mogelijk modules in een gegeven ruimte te plaatsen, waarbij ze allemaal in dezelfde richting worden gestapeld. In de ruimte die overblijft, probeert hij dan opnieuw zoveel mogelijk modules te stapelen, zelfs als dat betekent dat de richting van sommige modules anders is dan die van de andere.

Het script maakt gebruik van een heuristische methode in plaats van een analytische benadering om dit optimalisatieprobleem aan te pakken. Deze aanpak is gekozen, omdat die gemakkelijker te programmeren is en toch lokale minima kan vinden. Dit werkt ook bijzonder efficiënt en snel. Op een doorsnee-PC berekent het script 190 optimale configuraties, selecteert de beste voor een gegeven groep voorkeursinstellingen en plot die binnen 5 seconden. In die tijd slaat het ook alle gemaakte optimale configuraties op, zodat ze kunnen worden gefilterd voor verschillende voorkeuren. Als een andere configuratie beter blijkt te zijn, kan die weer gemakkelijk worden geplot met behulp van de GUI. De GUI berekent de verwachte prestatie van het accupack, en wel:

- de verwachte capaciteit;
- het aantal in serie en parallel geschakelde modules;
- het verwachte gewicht van de cellen;
- het spanningsverschil tussen een vol pack en een leeg pack.

Daarnaast berekent het de laadstromen en geeft het aan hoeveel warmte daarbij vrijkomt en welk laadvermogen nodig is om het pack in 7,5 minuten op te laden.

Prestaties

Ons doel met Electric Refueling is een elektrische auto op te kunnen laden in slechts een paar minuten. Als dat lukt, gaat het ongetwijfeld de energietransitie van benzine naar elektrisch in de automotive industrie versnellen. Om de wereld te laten zien hoe snel elektrisch racen kan zijn, hebben we de "Fusion" gebouwd: een elektrische racewagen die op dit moment recordhouder is met de snelste ronde tijd voor een elektrisch voertuig op de circuits in Zandvoort, Assen en Zolder (in België). Onze concept car "Vision" is de ware belichaming van de toekomstige elektrische mobiliteit. Het ontwerp is tien keer aerodynamischer dan een Formule-1-wagen. Dat is werkelijk de *endurance racer* van onze dromen. ▶

(180732-04)

Weblinks

- [1] Kokam-website: <http://kokam.com/modulepack>
- [2] InMotion-website: www.inmotion.tue.nl/

Advertentie

ALL ABOUT SUPPLY

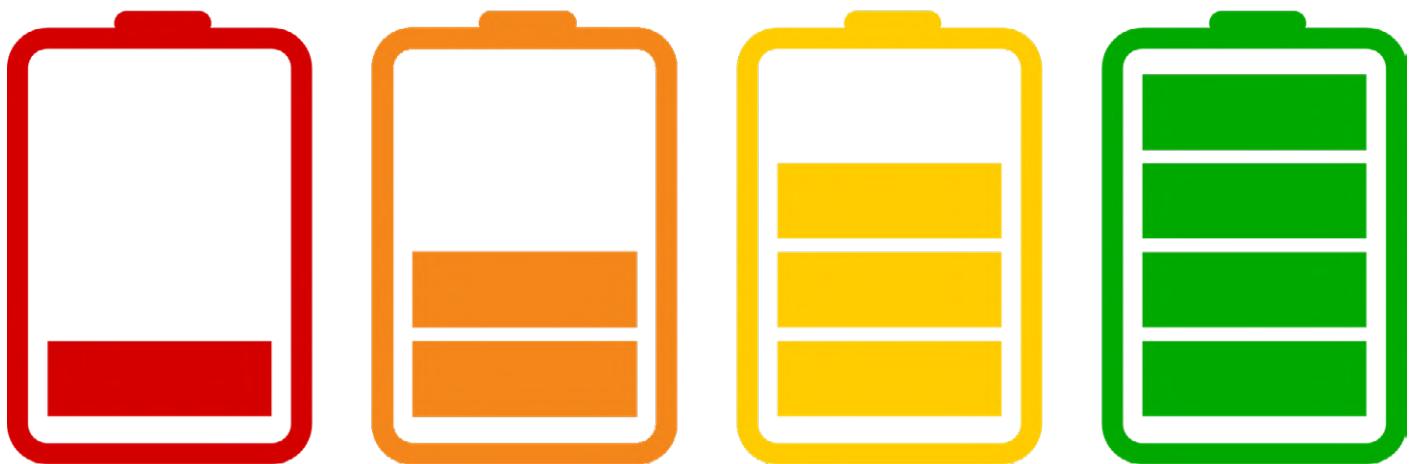


#coinnovation

IFA Contact
Nederlands-Duitse Handelskamer
Tel. +31 70 311 41 58
g.corbelli@dnhk.org

Accumonitor met ESP32

drie kanalen om te meten



Laurent Labbe (Hong Kong) en **Luc Lemmens** (Elektor Labs)

Dit project laat zien hoe eenvoudig het is om analoge waarden met een ESP32-board te meten, de meetresultaten via WiFi door te sturen naar een router, en ze vervolgens via ThingSpeak weer te geven en op te slaan.

Laurent werkt op dit moment aan verschillende projecten die met zonne-energie worden gevoed. Een van die projecten is een zonnevoeding voor zijn Nixie-horloge [1]. Omdat het niet te voorspellen is wanneer de zon schijnt, kost het soms uren of zelfs dagen om het prototype van een voeding te testen. Hij heeft drie verschillende schakelingen ontworpen waarbij drie kleine accu's van 130 mAh worden opgeladen. Het stroomverbruik van het horloge wordt hierbij met een weerstand gesimuleerd. Iedere tien minuten wordt het laadniveau van de accu's met drie verschillende modules gemeten, en worden de meetresultaten naar het gratis IoT-platform ThingSpeak gestuurd.

Metingen

De metingen worden uitgevoerd met behulp van een ESP32-PICO-Kit V4 ontwikkelboard, dat via een USB-connector met 5 V wordt gevoed. Omdat de scha-

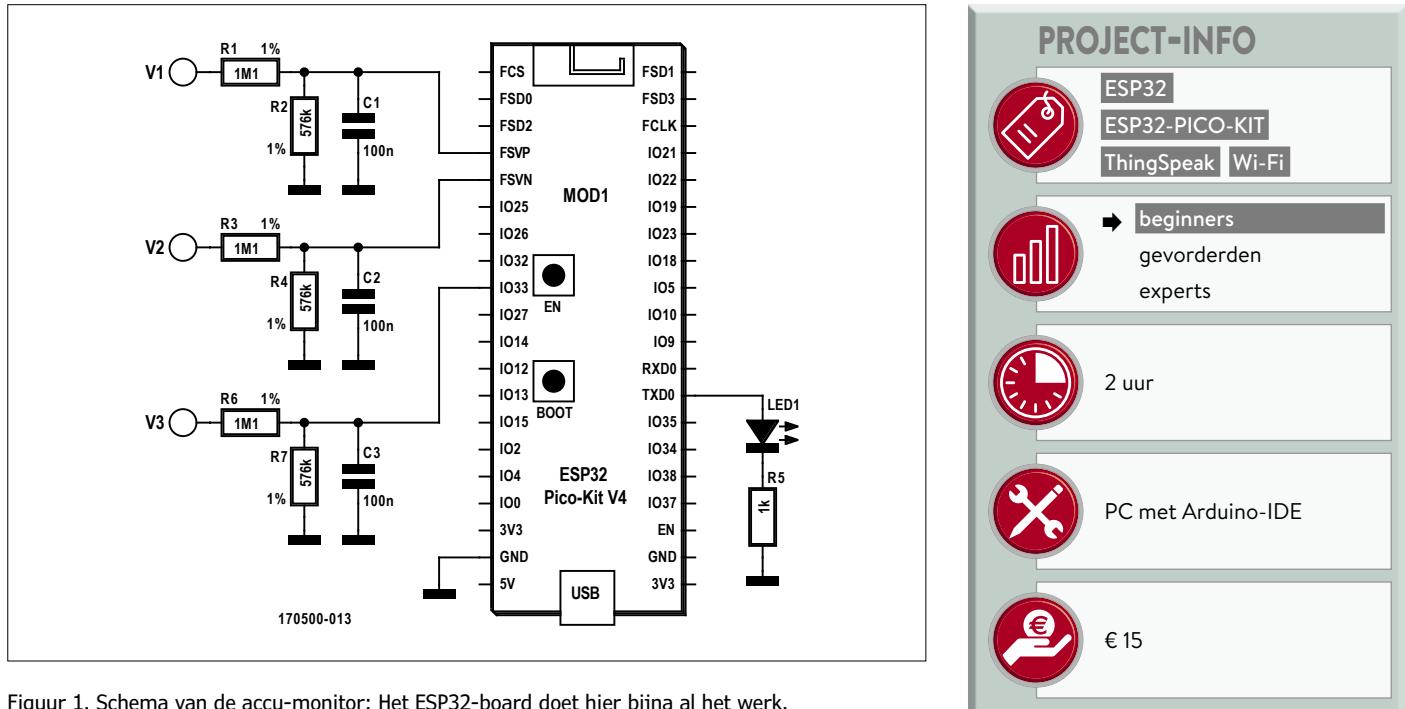
keling buiten is opgesteld op een plaats waar geen netaansluiting aanwezig is, gebruikt de auteur voor de voeding van het board twee externe accu's (*powerbank*), een kleine van 1000 mAh en een grote van 10.000 mAh. De kleine externe accu is continu in gebruik om de ESP32 te voeden en wordt zelf door de grote accu opgeladen. Deze laatste wordt vrijwel iedere dag verwisseld omdat het systeem altijd operationeel moet zijn. Als er maar één accu zou worden gebruikt, zouden de metingen tijdens het opladen moeten worden onderbroken.

Er worden drie I/O-aansluitingen van het ESP32-board gebruikt (**figuur 1**) om de metingen van de accu's via de drie kanalen uit te voeren. Ieder van deze I/O-aansluitingen beschikt over een eigen spanningsdeler en een filtercondensator van 100 nF. In de sketch wordt de verzwakking van deze spanningsdelers gepresenteerd door de variabelen `Coef_Vx`

waarvan de waarde kan worden aangepast om toleranties in de spanningsdelers te compenseren.

De analoog/digitaal-omzetter van het ESP32-board bevat een interne verzwakker die in de Arduino-sketch met de functie `analogSetAttenuation` op de volgende waarden kan worden ingesteld: 0 dB, -2,5 dB, -6 dB en -11 dB; dit komt overeen met de factoren 1, 0,75, 0,5 resp. 0,28.

Het spanningsbereik aan de ingang van de A/D-omzetter is 1,1 V. Dit is de maximale waarde die kan worden gemeten (als functie van het aantal bits dat voor ADC1 is geconfigureerd: 4095 bij 12 bit, 2047 bij 11 bit, 1023 bij 10 bit en 511 bij 9 bit). Om hogere spanningen te meten (tot de maximaal voor deze aansluiting toelaatbare spanning, in dit geval $VDD = 3,3$ V) moet voor het betreffende ADC-kanaal een verzwakking van meer dan 0 dB worden ingesteld. In **tabel 1**



Figuur 1. Schema van de accu-monitor: Het ESP32-board doet hier bijna al het werk.

PROJECT-INFO



ESP32
ESP32-PICO-KIT
ThingSpeak | Wi-Fi



beginners
gevorderden
experts



2 uur



PC met Arduino-IDE



€ 15

ziet u de instellingen van de verzwakker en het bijbehorende ingangsspanningsbereik. Bij een verzwakking van -11 dB wordt de maximale spanning beperkt door VDD_A en niet door het ingangsbereik.

De schakeling is ontworpen om LiPo-accu's van 4,2 V te monitoren, waarbij de analoge ingangen op een maximale ingangsspanning van 5 V zijn berekend. Met spanningsdelen die bestaan uit weerstanden van 1,1 MΩ en 576 kΩ voeren de ingangen van het ESP32-board een spanning van ongeveer 1720 mV bij een ingangsspanning van 5 V, en dat is het aanbevolen maximum bij een verzwakking van -6 dB. U kunt natuurlijk altijd het ingangscircuit aanpassen voor een ander ingangsspanningsbereik, al dan niet in combinatie met een andere interne verzwakking van de ADC op het ESP32-board.

Let op: op de ingangen van de schakelingen mogen alleen maar gelijkspanningen worden aangeboden die binnen het regelgebied van de ADC vallen. Bovendien is er geen beveiliging tegen overspanning of verkeerde polariteit.

Op GPIO17 van het ESP32-board is een LED aangesloten. Deze licht op tijdens de spanningsmetingen en tijdens het verzenden van de gegevens. Daarna gaat de LED uit en gaat het ESP32-board over in de slaapstand totdat het zichzelf automatisch wekt voor de volgende metingen.

Hiermee wordt energie bespaard voor de externe accu's.

De schakeling kan eenvoudig op een breadboard of een stukje gaatjesprint worden gemonteerd.

Een ThingSpeak-account aanmaken

Omdat Laurent het online-platform ThingSpeak al voor een weerstation-project gebruikte, lag het voor de hand om daar ook een pagina aan te maken om de metingen aan de horloge-accu's vast te leggen [3].

ThingSpeak is een gratis open-source webservice waarmee IoT-toepassingen kunnen worden ontwikkeld en waarmee gegevens van sensors kunnen worden verzameld en in de cloud worden opgeslagen. Sensordata kunnen naar ThingSpeak worden verzonden vanaf onder andere Arduino, Raspberry Pi en Beagle-

Beagle Black.

Als u met ThingSpeak bent verbonden is het ook mogelijk om verschillende toolboxen van MATLAB (MathWorks) te gebruiken. Hiervoor moet u bij het inloggen op ThingSpeak over een MathWorks-account met een actieve licentie beschikken.

Als u nog geen ThingSpeak- of MathWorks-account hebt en u wilt aan dit project beginnen, ga dan eerst naar [www.thingspeak.com](http://thingspeak.com), klik op [Get Started For Free](#), en maak vervolgens een account aan (**figuur 2**). Ga naar Connexion om u met uw gebruikersnaam en wachtwoord bij uw account aan te melden en creëer vervolgens onder [New Channel](#) een kanaal waar de gegevens van de accu-monitor zullen worden verzameld. Geef dit kanaal een logische naam die duidelijk herkenbaar is tussen alle gegevens van de accu-monitor. Voeg bij [Channel Set-](#)

Tabel 1. Theoretisch en praktisch ingangsbereik van de ADC, als functie van de ingestelde verzwakking bij VDD_A = 3,3 V.

Verzwakking van de ADC	Theoretisch ingangsbereik	Realiseerbaar ingangsbereik voor nauwkeurige meetresultaten
0 dB (ADC_ATTEN_DB_0)	1,1 V	100 tot 950 mV
2,5 dB (ADC_ATTEN_DB_2_5)	1,5 V	100 tot 1250 mV
6 dB (ADC_ATTEN_DB_6)	2,2 V	150 tot 1750 mV
11 dB (ADC_ATTEN_DB_11)	3,9 V	150 tot 2450 mV

(Bron : <https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/peripherals/adc.html#adc-calibration>)

Sign up for ThingSpeak

It is free to sign up for ThingSpeak. Free accounts offer a fully functional experience on ThingSpeak with limits on certain functionality. Commercial users may sign up for a time-limited free evaluation. To send data faster to ThingSpeak or to send more data, consider our [paid license options](#) for commercial, academic, home and student usage. To start using ThingSpeak you must create a new MathWorks account, or, click cancel and log in using an existing MathWorks account.

Create MathWorks Account

Email Address

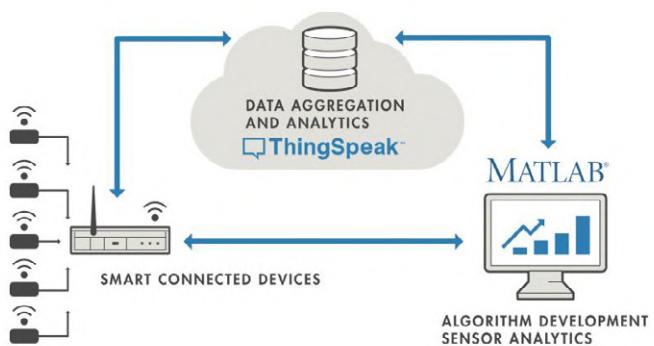
Missing required information

To access your organization's MATLAB license, use your school or work email.

Location

First Name

Last Name



Figuur 2. Een account aanmaken bij ThingSpeak.

ting drie velden toe voor de data die de Arduino-sketch zal verzenden, en geef ze een naam (in ons geval V1, V2 en V3, zie **figuur 3**). Selecteer nu de tab **API keys**, hier vindt u de *Write API key* die u naar de sketch van het ESP32-board moet kopiëren om aan te geven welk ThingSpeak-kanaal u gebruikt (**figuur 4**).

De sketch van de accumonitor

De in C geschreven sketch [Battery_monitor_elektor_3.ino](#) is niet erg ingewikkeld, maar voordat u deze compileert

en in het ESP32-board laadt moet de sketch worden aangepast zodat het board zich met uw WiFi-netwerk verbindt en de gegevens naar uw ThingSpeak-kanaal verzendt.

Voer op regel 68 van de sketch de bij uw kanaal behorende *Write API key* in, die u van de website van ThingSpeak hebt gehaald. Voer vervolgens in de regels 70 en 71 respectievelijk de SSID en het wachtwoord van uw WiFi-netwerk in. Upload de sketch naar het ESP32-board en open de seriële monitor

van de Arduino-IDE in het **tools**-menu. Zorg ervoor dat de transmissiesnelheid van de monitor is ingesteld op 115200. U zult nu het volgende bericht van de ESP32 ontvangen: [Battery monitor V1.0 Start](#), met aansluitend de eerste drie meetwaarden.

Na elke meetcyclus gaat het ESP32-board over naar de slaapmodus. Met **Boot number: x** wordt aangegeven hoeveel cycli sinds de start van de sessie zijn uitgevoerd (oftewel hoeveel metingen er tot nu toe zijn gedaan).

Figuur 3. Instellen van de kanalen op ThingSpeak.

Figuur 4. API-sleutel om in de sketch in te voegen.

Na het ontwaken verbindt het ESP32-board zich (weer) met uw WiFi-netwerk en geeft het aan dat de verbinding met het IP-adres van de ESP32 tot stand is gekomen. Als dit niet lukt krijgt u het bericht `WiFi cannot connect`.

U kunt het meetinterval instellen door in de sketch `TIME_TO_SLEEP` op regel 45 aan te passen.

Merk op dat alleen de functie `setup` wordt uitgevoerd, deze voert alle acties uit (zowel de metingen als het verzenden van de gegevens) en zet vervolgens het ESP32-board in de slaapmodus totdat een interne timer deze weer waker maakt voor de volgende meetcyclus. Daarna zal `setup` opnieuw worden uitgevoerd. De functie `loop` is leeg, deze wordt nooit uitgevoerd.

Als dit uw eerste project is met een ESP32-board in de Arduino-IDE, vergeet dan niet de ESP32-boards in deze ontwikkelomgeving te installeren. U vindt hiervoor verschillende tutorials op het internet (zie bijvoorbeeld [4]). Let op: selecteer in het menu `Tools` onder `Port` de juiste COM-poort voor de verbinding met de ESP32-PICO-Kit V4, en kies onder

 **IN DE STORE**

→ Ontwikkelboard ESP32-PICO-Kit V4, Art. nr. 18423
www.elektor.nl/esp32-pico-kit-v4

Board voor de ESP32 Pico Kit.

Omdat er geen aparte print voor hoeft te worden gemaakt is dit eenvoudige project snel klaar voor gebruik. Het opent

de weg naar verschillende soorten metingen en biedt toegang tot de cloud. ▶

(170500-03)

Weblinks

- [1] Labpagina van het Nixie-horloge: www.elektrormagazine.nl/labs/nixie-watch-1
- [2] ThingSpeak -platform: www.thingspeak.com
- [3] Openbare pagina van de accu-monitor op ThingSpeak: <http://thingspeak.com/channels/329068>
- [4] Installatie van ESP32-boards in de Arduino-IDE: <http://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions>
- [5] Labpagina van de accu-monitor: www.elektrormagazine.nl/labs/simple-wifi-multimeter
- [6] Webpagina bij dit artikel: <http://www.elektrormagazine.nl/170500-03>

Advertentie



Deel je ideeën en elektronische projecten – eender welke moeilijkheidsgraad – op www.elektor-labs.com en wordt beroemd!

Elektor Labs: www.elektor-labs.com
Design, Build, Share... Electronics!



Voeding voor differentiële probes via de USB-poort

Dr. Martin Oppermann (Duitsland)

In Elektor september/oktober 2016 [1] presenteerde Elektor-auteur Alfred Rosenkränzer een wonderbaarlijk simpele differentiële meetkop voor oscilloscopen op basis van de precisie-verschilversterker AD8479, en die heb ik direct nagebouwd. Deze probe is echt superhandig voor metingen in bijvoorbeeld het audio-bereik. Een klein detail kwam in dat bewuste artikel echter niet aan de orde, namelijk de voeding voor die meetkop (of voor meerdere). Nu hebben alle moderne oscilloscopen aan de voorkant een USB-aansluiting voor de uitvoer van meetgegevens. En aangezien het stroomverbruik van een meetprobe heel laag is, ligt het voor de hand om deze aansluiting te gebruiken voor de voeding voor de probe. In **figuur 1** ziet u de schakeling. De probevoeding sluiten we aan op de USB-poort van de oscilloscoop met behulp van een gewone USB-kabel; u zou zelfs een gewone USB-laadkabel kunnen gebruiken. C1 en C2 filteren de USB-gelijkspanning, LED1 geeft een visuele indicatie dat de voeding is ingeschakeld. Het hart van de schakeling vormt de geïsoleerde DC/DC-converter, een IH0512S van de firma XP Power [2]. Van +5 V op de ingang maakt hij een symmetrische ± 12 V bij een maximale uitgangsstroom van ± 84 mA, en dat alles galvanisch gescheiden! De isolatiespanning bedraagt 1000 V, zodat we geen massaproblemen bij het meten hoeven te verwachten. De uitgangsspanning wordt gefilterd met meerdere condensato-

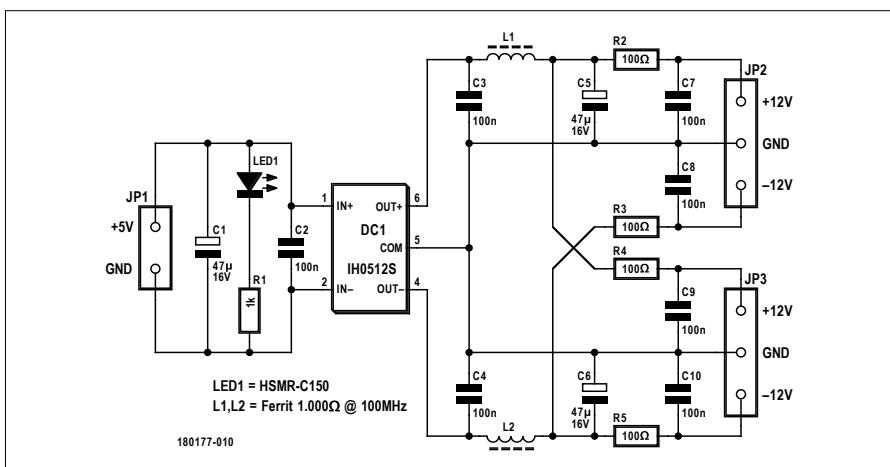
toren en ferrietkern-spoeltjes. Bij de maximale meetfrequentie van de AD8479 van 130 kHz kunnen die ook vervallen of worden vervangen door weerstanden van 0Ω of draadbruggen.

Voor de aansluiting van Rosenkränzers meetkop heb ik stereo-jacks gebruikt. Deze zijn lekker klein, maar ze hebben het nadeel dat de contacten kortstondig worden kortgesloten bij het insteken van de plug. Om te voorkomen dat daardoor schade zou kunnen ontstaan, zitten er $100\text{-}\Omega$ -weerstanden in elke uitgangskring. De spanningsval over deze weerstanden vormt geen probleem, want het stroomverbruik van de meetkop ligt in de orde van grootte van enkele mA (de AD8479 trekt volgens de datasheet 850 μ A; alleen de indicatie-LED's D5 en D6 trekken meer stroom).

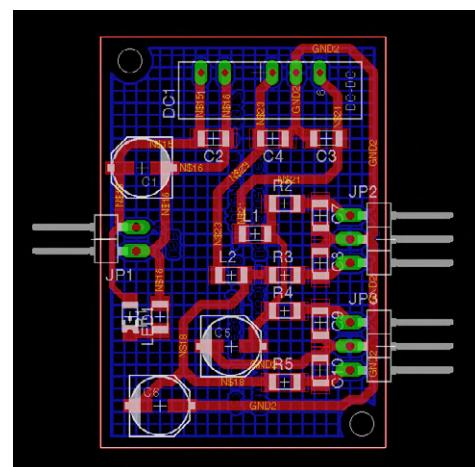
De schakeling is merendeels uitgevoerd met SMD's op een dubbelzijdige print, waarvan de onderkant een puur massavlak is. De kleinste SMD-maat is 1206, wat nog prima met de hand te solderen is. In **figuur 2** ziet u de print-layout, gemaakt in Autodesk Eagle. U kunt hem downloaden via de projectpagina bij dit artikel [3]. Het printje is ondergebracht in een kleine transparante kunststofbehuizing van Hammond Manufacturing (type 1551RTBU).

Nog een tip: in mijn versie heb ik voor IC1 en IC3 in de meetkop spanningsregelaars van het type 78L09 en 79L09 gebruikt (dus ± 9 V), om het meetbereik te vergroten naar $\pm 8,7$ V. ◀

180177-03



Figuur 1. Schakeling van de probe-voeding, hier voor twee probes.



Figuur 2. Print-layout en onderdelenopstelling.

Weblinks

- [1] Rosenkränzer, Alfred: "Stroommeetkop voor oscilloscoop", www.elektormagazine.nl/magazine/elektor-201609/39831
- [2] 2 W DC/DC converter IH0512S: www.xppower.com/Portals/0/pdfs/SF_IH.pdf
- [3] Projectpagina bij dit artikel: www.elektormagazine.nl/180177-03

Seriële data weergeven op een webpagina met PHP- of Python-scripts



Clemens Valens (Elektor Labs)

Soms kan het handig zijn om een continue stroom seriële data op een webpagina in een browser weer te geven. Hoe u dit kunt doen? Eenvoudig, met een paar scripts.

We moeten hiervoor periodiek een webpagina maken die de meest recente seriële data bevat. Deze pagina kan dan op dezelfde computer of via een netwerk in een webbrowser worden weergegeven. Er is dus een programma nodig dat de seriële data continu omzet naar bijvoorbeeld HTML- of PHP-bestanden. PHP is dat niet een programmeertaal voor internet-gerelateerde toepassingen? Kunnen we daarmee dit probleem oplossen? Ja, dat kan, maar we zullen zien dat er ook andere methoden zijn.

Automatische pagina-refresh

We beginnen met HTML, dat met een meta-tag een browser kan opdragen om regelmatig een pagina te herladen:

```
<meta http-equiv="refresh" content="10">
```

Deze tag vertelt de browser dat de pagina met die tag iedere tien seconden moet worden herladen. (Als uw browser de tag niet ondersteunt, kunt u deze vervangen door een stukje JavaScript. Zie hiervoor de download op [1].) Als we nu een webpagina met deze tag maken, dan zal deze door de browser iedere tien seconden opnieuw worden geladen (een ander

tijdsinterval is ook mogelijk). Herschrijven we de pagina nu iedere tien seconden met recente seriële data, dan zal de browser deze recente data weergeven.

Taken verdelen

Als de refresh-tag zich in een PHP-bestand in plaats van in een HTML-bestand bevindt, zal de browser dezelfde actie uitvoeren. In dit PHP-bestand zouden we een script kunnen opnemen waarmee de data van de seriële poort worden gehaald. Maar dit wordt lastig omdat PHP van huis uit geen seriële poorten ondersteunt. En zelfs als het dit wel zou doen, dan zou het script als de browser de recente versie van de pagina opvraagt de seriële poort moeten openen, de data ophalen en de poort weer sluiten. Data die buiten dit venster bij de seriële poort aankomen gaan verloren. Ook kunnen sommige Arduino-achtige systemen zich resetten als de seriële poort wordt geopend en dat maakt deze opzet onbruikbaar. Een oplossing hiervoor is het proces te splitsen in twee subprocessen:

- Proces 1: een script dat continu de seriële poort leest en de ontvangen data opslaat in een bestand dat door de

Listing 1: Een PHP-script dat data van de seriële poort leest, en deze vervolgens wegschrijft naar een bestand met de naam 'data.txt'.

```
<?php

// Linux $comPort = “/dev/ttyACM0”;
$comPort = “COM15”;

include “php_serial.class2.php”;
$serial = new phpSerial;
$serial->deviceSet($comPort);

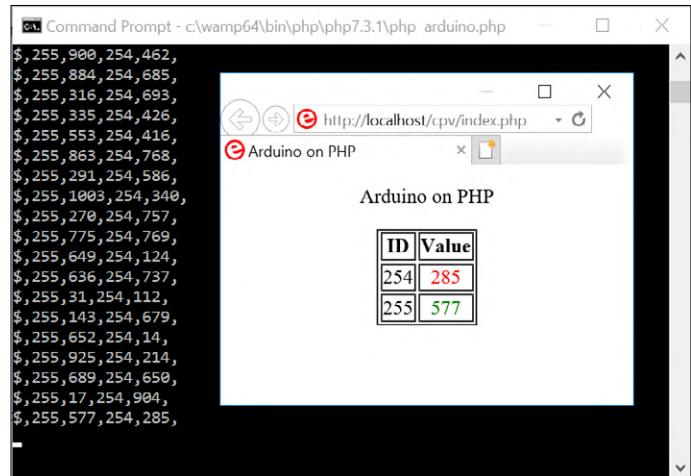
// On Windows (10 only?) all mode settings must be done in one go.
$cmd = “mode “ . $comPort . “ baud=115200 parity=n data=8 stop=1 to=off xon=off”;
$serial->_exec($cmd);
$serial->deviceOpen();

echo “Waiting for data...\n”;
sleep(2); // Wait for Arduino to finish booting.
$serial->serialflush();

while(1)
{
    $read = $serial->readPort();

    if (strlen($read) !=0)
    {
        $fp = fopen(“data.txt”,”w”);
        if ($fp!=false)
        {
            fwrite($fp,trim($read));
            fclose($fp);
        }
    }
}

?>
```



PHP-webpagina wordt geïmporteerd (**listing 1**);

- Proces 2: een browser die periodiek de PHP-webpagina herlaadt zodat deze de data kan verversen (**listing 2, figuur 1**).

Er is een webserver nodig

Het probleem met het openen en sluiten van de seriële poort en het hiermee samenhangende dataverlies is nu opgelost, maar dit vereist een script dat op de achtergrond loopt. Als dit een PHP-script is, dan moet de computer PHP-scripts kunnen uitvoeren. Ook is een webserver nodig om de PHP-pagina aan een browser aan te bieden, anders geeft de browser niet de pagina maar de onderliggende PHP-code weer. De oplossing hiervoor is de installatie van een 'AMP'- of 'WAMP'-pakket. AMP is de afkorting van Apache-MySQL-PHP, de 'W' is van Windows. Met zo'n pakket krijgt u een complete webserver met alle toeters en bellen in huis.

Gebruik geen PHP...

We hebben deze methode geprobeerd en kregen het aan de praat, maar niet zonder problemen. Naast moeilijkheden bij het installeren van de webserver, was het een probleem om PHP betrouwbaar een seriële poort te laten openen om de data te ontvangen. Uit internet-research bleek dat er maar één PHP-bibliotheek voor seriële communicatie is: PHP Serial. Alle

Figuur 1: Het door het PHP-script gegenereerde dynamische PHP-bestand wordt door een WAMP-server aan onze browser aangeboden (zie de adresregel). Het bevat een tabel met kommagescheiden waarden die van de seriële poort zijn gelezen. Waarden onder de 500 worden in rood weergegeven, de andere zijn groen. In de titel van het commandopromptvenster ziet u het commando om het script uit te voeren.

andere lijken hiervan te zijn afgeleid. Op de GitHub-pagina [2] schrijft de auteur: "*Windows: bij sommige mensen lijkt het te werken, maar voor sommige anderen niet.*" Wij hoorden bij de tweede groep... Om met PHP de seriële communicatie werkend te krijgen, moesten we eerst met een terminalprogramma (zoals TeraTerm) de poort openen en meteen weer sluiten; een andere manier was er niet. Daarom hebben we besloten om PHP te verlaten en Python te gebruiken.

...maar Python

Python 3 met pySerial [3] bleek op onze Windows-10-testcomputer prima te werken, en dus maakten we een script dat de data van de seriële poort leest en er een webpagina mee vult. Nu er geen reden meer is om PHP te gebruiken, kan met het pythonscript ook een eenvoudig HTML-bestand worden geproduceerd (**listing 3, figuur 2**). Ook de PHP-dataformatering

Listing 2: Deze PHP-webpagina formateert de inhoud van het bestand met de naam ‘data.txt’ als een tabel.

```
<?php

$page_title = "Arduino on PHP";

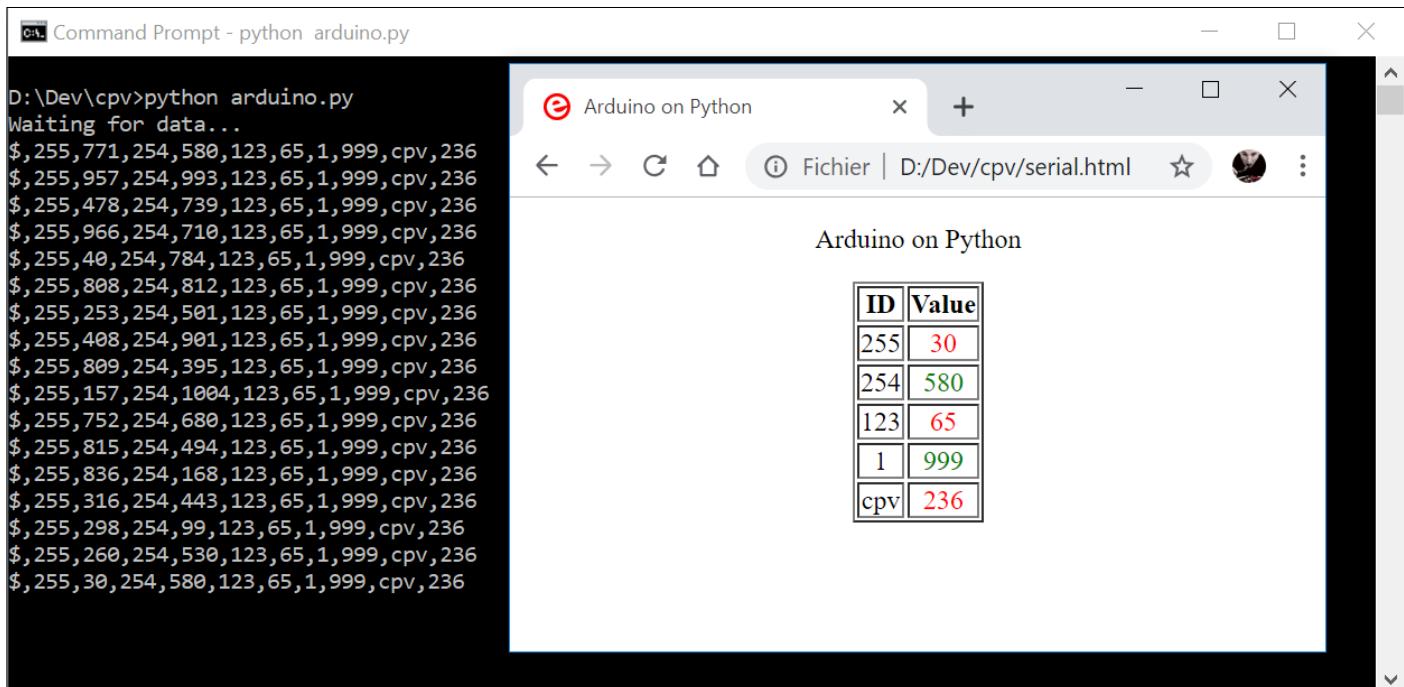
// Start of HTML page.
echo "<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.01//EN' 'http://www.w3.org/TR/html4/strict.dtd'>";
echo "<html>"; // Page begin.
echo "<head><title>$page_title,</title>"; // Head begin.
echo "<meta http-equiv='refresh' content='1'>";
echo "<meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>";
echo "<link rel='shortcut icon' href='favicon.ico' />";
echo "<link rel='icon' type='image/x-icon' href='favicon.ico' />";
echo "<link rel='icon' type='image/png' href='favicon.png' />";
echo "</head>"; // Head end.
echo "<body><center>"; // Body begin.

echo "<p>$page_title,</p>"; // Page title.

// Create a table from data file.
$handle = fopen("data.txt","r");
if ($handle!=NULL)
{
    // Read one line from the file, then close it.
    $data = fgets($handle);
    fclose($handle);

    // Synchronise to the data.
    if ($data[0]=='$')
    {
        // Remove whitespace.
        str_replace(' ','',$data);
        // Split data in fields separated by ','.
        // Expected format: "$,id1,value1,id2,value2,CRLF"
        list($startchar,$id1,$value1,$id2,$value2,$newline) = explode(",",$data);
        // Create array from list.
        $numbers = array($id1=>$value1,$id2=>$value2);
        // Sort array in ascending key order.
        ksort($numbers);

        // Table begin.
        echo "<table border='1' border-spacing='5' style='text-align:center;'>";
        echo "<tr><th>ID</th><th>Value</th></tr>";
        foreach ($numbers as $x => $x_value)
        {
            echo "<tr>"; // Table row begin.
            echo "<td>", $x, "</td>"; // Table column 1.
            echo "<td>"; // Table column 2 begin.
            if ($x_value>=500) echo "<font color='green'>";
            else echo "<font color='red'>";
            echo $x_value;
            echo "</font></td>"; // Table column 2 end.
            echo "</tr>"; // Table row end.
        }
        // Table end.
        echo "</table>";
    }
}
echo "</body>"; // Body end.
echo "</html>"; // Page end.
?>
```



Figuur 2: Hier produceert het Pythonscript een dynamisch HTML-bestand (zoals u kunt zien in de adresbalk van de browser).

The screenshot shows the Arduino IDE interface with a sketch named 'arduino_on_php'. The code is as follows:

```
int id[2];

void setup(void)
{
    Serial.begin(115200);
    Serial.println("Arduino on PHP");
    Serial.println("Data transmission started.");
    randomSeed(analogRead(0));
    id[0] = 255;
    id[1] = 254;
}

void loop(void)
{
    Serial.print('$');
    Serial.print(',');
    Serial.print(id[0]);
    Serial.print(',');
    Serial.print(random(1023));
    Serial.print(',');
    Serial.print(id[1]);
    Serial.print(',');
    Serial.print(random(1023));
    Serial.println(',');
    delay(1000);
}
```

At the bottom of the IDE, it says 'Done Saving.' and 'avrduude done. Thank you.'

kan in Python worden gedaan en de webpagina kan zonder server door een browser worden weergeven en ververst zodat er geen (W)AMP-pakket meer nodig is. Dit maakt alles veel eenvoudiger.

Tot slot

Dit artikel beschrijft een methode om seriële data in een webbrowser weer te geven. Die methode is zeker niet nieuw, exclusief of 'de beste'. Als u een andere manier weet – eenvoudiger, eleganter, of wat dan ook – deel deze dan met ons op [1]. En voor het script kunt u in plaats van Python iedere andere programmeertaal gebruiken die seriële communicatie en het schrijven van bestanden ondersteunt. Het voordeel van Python met pySerial is dat het draait op Windows-, macOS- en Linux-machines (en nog meer).

De voor dit artikel ontwikkelde code in de vorm van PHP- en Pythonscripts en een Arduinosketch kan van [1] worden gedownload. ►

(170111-03)

Weblinks

- [1] Downloads bij dit artikel: www.elektormagazine.nl/170111-03
- [2] PHP Serial: <https://github.com/Xowap/PHP-Serial>
- [3] pySerial: <https://pypi.org/project/pyserial/>

Figuur 3: Deze Arduino-sketch genereert een seriële datastream die kan worden gebruikt voor het ontwikkelen, debuggen en testen van scripts.

Listing 3: Een Python-script dat data leest van de seriële poort en er een HTML-bestand voor genereert.

```
import serial
import time

file_name = "serial.html" # Once created, open this file in a browser.

# Adapt serial port nr. & baud rate to your system.
serial_port = 'COM15'
baudrate = 115200

page_title = "Arduino on Python";

def write_page(data_list):
    fo = open(file_name,"w+")
    # Start of HTML page.
    fo.write("<!DOCTYPE HTML PUBLIC '-//W3C//DTD HTML 4.01//EN' 'http://www.w3.org/TR/html4/strict.dtd'>")
    fo.write("<html><head><title>" + page_title + "</title>") # Page & Head begin.
    fo.write("<meta http-equiv='refresh' content='1'>")
    fo.write("<meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>")
    fo.write("<link rel='shortcut icon' href='favicon.ico' />")
    fo.write("<link rel='icon' type='image/x-icon' href='favicon.ico' />")
    fo.write("<link rel='icon' type='image/png' href='favicon.png' />")
    fo.write("</head><body><center><p>" + page_title + "</p>"") # Head end, body begin.

    # Table begin.
    fo.write("<table border='1' border-spacing='5' style='text-align:center;'>")
    fo.write("<tr><th>ID</th><th>Value</th></tr>")
    for i in range(0,len(data_list),2):
        fo.write("<tr>") # Table row begin.
        fo.write("<td>" + data_list[i] + "</td>") # Table column 1.
        fo.write("<td>") # Table column 2 begin.
        fo.write("<font color='>")
        # Values >= 500 will be printed in green, smaller values will be red.
        if (int(data_list[i+1]) >= 500): fo.write("green")
        else: fo.write("red")
        fo.write("</font>")
        fo.write(data_list[i+1])
        fo.write("</td>") # Table column 2 end.
        fo.write("</tr>") # Table row end.
    fo.write("</table>") # Table end.
    fo.write("</body>") # Body end.
    fo.write("</html>") # Page end.
    # Done, close file.
    fo.close()

    s = serial.Serial(serial_port,baudrate) # Open serial port.
    s.dtr = 0 # Reset Arduino.
    s.dtr = 1
    print("Waiting for data..."); 
    time.sleep(2) # Wait for Arduino to finish booting.
    s.reset_input_buffer() # Delete any stale data.

while 1:
    data_str = s.readline().decode() # Read data & convert bytes to string type.
    # Clean up input data.
    # Expected format: "$,id1,value1,id2,value2,...,CRLF"
    data_str = data_str.replace(' ', '') # Remove whitespace.
    data_str = data_str.replace('\r', '') # Remove return.
    data_str = data_str.replace('\n', '') # Remove new line.
    data_str += '123,65,1,999,cpv,236' # Add some more data
    print(data_str)
    # Split data in fields separated by ','.
    data_list = data_str.split(",")
    del data_list[0] # Remove '$'
    # Write HTML page.
    write_page(data_list)
```

Qt voor Raspberry Pi

de inrichting van het GUI-framework – stap voor stap



Logo Qt-Frameworks (www.qt.io).

Mathias Claußen (Elektor Labs)

Onze software-ontwikkelaar Mathias Claussen moest voor het project FM-radio op de Raspberry Pi [1] een aansturing voor een aanraakscherm bouwen. Nu is het platform-onafhankelijke (GUI)-framework Qt voor dergelijke gebruikersinterfaces al zo ongeveer de industrie-standaard geworden, dus de keuze was snel gemaakt. Voor ontwikkelaars die voor een vergelijkbare taak staan, heeft Mathias een handleiding samengesteld waarmee u vlot Qt voor de Raspi kunt compileren. En uiteraard zit er ook een kleine demo bij.

Bij software-ontwikkeling krijgt men meestal een eisenpakket van de opdrachtgever: een lijst schriftelijk vastgelegde functionaliteiten van, en eisen aan, het eindproduct. In dit geval kwam het eisenpakket van de Elektor-redactie. De opdracht van de collega's was best overzichtelijk: "Zorg dat het werkt!". Met zulke 'gedifferentieerde' criteria heeft vermoedelijk ook deze en gene ontwikkelaar onder onze lezers al een keer te maken gehad. Maar na enig gefrons konden de eisen dan toch op een adequate manier worden vastgelegd.

De wens van de collega's was dat er een Linux-applicatie voor de Raspberry Pi gebouwd moest worden waarmee een FM-radio moest kunnen worden bediend. Ook DRS-informatie, als die in het signaal aanwezig was, moest worden gedecodeerd en weergegeven. En het zal waarschijnlijk geen verrassing voor u zijn, want zo gaat het vrijwel altijd met software: het project moet het liefst gisteren klaar zijn ;-)

Op dat punt besloot ik om het project met behulp van het Qt-framework [2] te verwerkelijken. Aangezien het een *open source*-project moet worden, hoeft ik niet te aarzelen tussen de GPL- en de commerciële versie van Qt. Als we de GPL-versie gebruiken, dan mogen we daarmee ook toepassingen maken die niet *open source* zijn. In dat geval is het wel aan te bevelen om de licentievoorwaarden voor Qt wat nauwkeuriger te bestuderen.

Qt

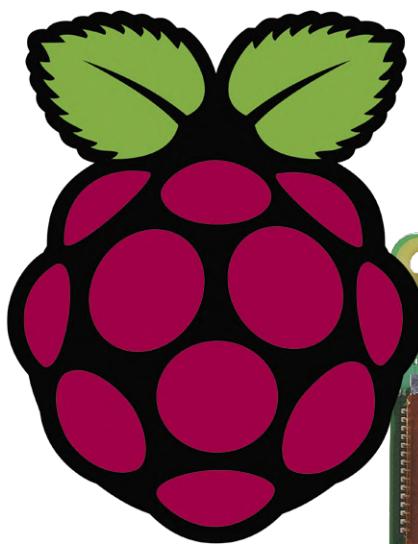
Qt [2] is een platform-onafhankelijk framework met behulp waarvan applicaties in C++/QML kan maken, die u dan in het ideale geval met gemeenschappelijke code op meerdere platforms kunt draaien. Met 'platforms' bedoelen we dan de *usual suspects* zoals Windows, Linux, Android, iOS, QNX en een paar embedded-Linux-varianten. Met de keuze voor dit

framework besparen we ons een heleboel werk en hoeven we ons niet te laten ringloren door de eigenaardigheden van de diverse besturingssystemen. Qt omvat meerdere bibliotheken, maar juist die voor de ontwikkeling van GUI's zorgt ervoor dat we snel iets aan de praat kunnen hebben. Wat in het algemeen geldt, geldt ook voor Raspberry Pi. Het nut van Qt was me bij eerdere projecten ook al duidelijk geworden (van console-toepassingen tot een klein GUI-tooltje). Het elegante van Qt is dat de ervaring die men er in andere projecten mee heeft opgedaan, niet alleen van pas komt op de RPi, maar dat men die ook kan 'porten' naar andere platforms die door het framework worden ondersteund.

Een vers framework

Met een bespreking van het Qt-framework kunnen hele boekwerken worden gevuld. Dat geldt voor allerlei aspecten zoals de bediening van de tools, de ontwikkeling van applicaties of het bouwen van grafische interfaces. Als u nu met een RPi aan

de slag wilt, dan kunt u een kant-en-klare versie van Qt inzetten. Dat is wel makkelijk, maar het heeft als nadeel dat die kant-en-klare versie op dit moment al meer dan



Logo Raspberry Pi Foundation
(www.raspberrypi.org).

twee jaar oud is. Voor software-begrippen is dat tweemaal een eeuwigheid. Het alternatief is de klassieke CIY (Compile It Yourself). Op die manier zijn de veiligheids-updates en alle patches vers en niet twee enen oud en dat is altijd een geruststellende gedachte. Bij het eigenlijke Qt-framework hoort ook de Qt Creator. Dit nuttige stuk gereedschap wordt met elke nieuwe versie alleen maar beter. In het verleden zijn er al veel bugs verholpen. Qt Creator kan ook draaien op de RPi, maar RAM en de monitor-aansluitingen op de Pi zijn maar in beperkte mate beschikbaar. Als u echter Qt zelf bouwt, dan kunt u tegelijk ook een versie voor de PC configureren. Zodoende kunt u RPi-toepassingen ook op

de PC compileren en testen, zolang ze maar geen RPi-specifieke features gebruiken. Met één klik en de juiste instellingen kunt u hetgeen u op de PC getest hebt, kopiëren naar de RPi en daar uitvoeren. En dat allemaal inclusief de mogelijkheden van code-debugging, het zetten van breakpoints en het weergeven van de variabelen-waarden.

Handleiding (voor een gelukkig leven)

Als u nu op zoek gaat naar een handleiding, dan zult u vroeger of later ergens vastlopen, want dan blijkt dat u een of andere instelling mist, of u komt erachter dat die bewuste handleiding geschreven is voor een versie die helaas problemen geeft met het huidige besturingssysteem van de RPi. Daarom heb ik een handleiding [3] geschreven, waarin de nodige stappen voor het opzetten van Qt op de Raspberry Pi en de ontwikkelomgeving op een PC worden beschreven.

Aangezien het laatste eenvoudiger is onder Linux, is voor de PC-omgeving uitgegaan van Ubuntu. Maar wees maar niet bang: u hoeft geen extra PC alleen voor RPi-doeleinden aan te schaffen (de prijs is daarbij vaak minder een probleem dan de beschikbare plaats). Wat u nodig hebt is een virtuele machine (*virtual box*) en ongeveer 60 GB vrije schijfruimte op de SSD van uw PC, die u dan inzet voor technische doeleinden.

Naast de pure setup wordt ook beschreven hoe een klein demo-programma tot stand komt. Daarnaast kunt u het

FM-radioproject in Qt laden en zien hoe de afzonderlijke delen met elkaar samenwerken. Deze handleiding is bedoeld als uitgangspunt, en ze laat zien hoe u met de diverse tools kunt omgaan zonder al bij de inrichting belangrijke details over het hoofd te zien. ▶

(180477-03)



IN DE STORE

- Kale print voor de FM-Radio-HAT voor de Raspberry Pi (160520-1 V2.0)
www.elektor.nl/rpi-receiver-160520-1
- Raspberry Pi 3 B+
www.elektor.nl/rpi-3b-plus

Weblinks

- [1] FM-Radio met RDS, Elektorlabs januari/februari 2019: www.elektormagazine.nl/magazine/elektor-72/42299
- [2] Qt-framework: www.qt.io
- [3] Qt-handleiding (PDF, 41 pagina's): <https://github.com/ElektorLabs/160520-QT-compile-and-install-manual>

elektor start-up Challenge Parijs 2019

internationaal lanceerplatform voor start-ups

Denis Meyer
Mariline Thiebaut-Brodier

Wacht niet tot Elon Musk u aan boord van de SpaceX uitnodigt!

Na Duitsland en Nederland zal Elektor dit jaar ook Parijs aandoen met een nieuwe wedstrijd voor startende ondernemers op het gebied van embedded elektronica. Deze zal plaatsvinden tijdens de nieuwe beurs 'Forum de l'électronique' die in september 2019 wordt gehouden. De wedstrijd heet de elektor start-up challenge en heeft als doel beginnende bedrijven een 'boost' te geven. De winnaars zullen worden geselecteerd op basis van de industriële haalbaarheid, de bruikbaarheid en natuurlijk de originaliteit van de projecten. De elektor start-up challenge is het resultaat van een nauwe samenwerking met Cap'tronic, het steunprogramma voor het MKB op het gebied van elektronische systemen en embedded software, en GL Events Exhibitions, de organisator van de beurs.

Wie kunnen er meedoen?

U, als u een product, software of dienst gerelateerd aan elektronica ontwikkelt. De *elektor start-up challenge* biedt u de mogelijkheid om uw project aan professionals uit de branche te presenteren. De vorige door Elektor georganiseerde wedstrijden, met name die tijdens de *electronica* in München, de grootste elektronica beurs ter wereld, hebben opmerkelijke projecten op verschillende gebieden opgeleverd.

Waarom instappen in 'onze' raket?

Als u zich inschrijft voor de *elektor start-up challenge* van Parijs 2019 en uw deelname is geaccepteerd, zult u met de andere kandidaten worden ontvangen op het podium van de *elektor start-up challenge*, een grote centrale stand op de beurs **Forum de l'électronique** (Paris Expo - Porte de Versailles, 24 tot 26 september 2019).

U kunt hier:

- achter uw eigen desk uw project presenteren;
- de competitie aangaan met de andere finalisten;
- de aandacht trekken van potentiële partners.

Voor, tijdens en na de beurs profiteert u van veel exposure:

- u en uw start-up krijgen een plaats in de ondersteunende media van het *Forum de l'électronique*: catalogus van exposanten, lijst van deelnemers, officiële website...
- u en uw start-up zullen worden gepresenteerd aan alle lezers van het tijdschrift Elektor, via al onze media (papier, web, socials, YouTube-kanaal...) in Frankrijk, in Frans-sprekende landen en in de rest van de wereld;
- u en uw start-up zullen worden gepresenteerd aan deelnemers aan het Cap'tronic-programma.

Voor u en voor uw start-up is de hoogste beloning het winnen van de hoofdprijs. De winnaar van de *elektor start-up challenge* in Parijs ontvangt een toegangs bewijs voor de volgende **electronica** in München: u zult automatisch worden geselecteerd om er deel te nemen aan de internationale wedstrijd **Fast Forward Award** waarbij u uw project kunt presenteren en profiteren van de enorme publiciteit rond dit evenement.

De **Elektor Start-up Challenge** wordt georganiseerd in samenwerking met:



Uw inschrijving

- Om deel te nemen aan de *elektor start-up challenge* tijdens het *Forum de l'Électronique*, kunt u rechtstreeks contact met ons opnemen op het adres redactie@elektor.nl. U kunt ook een (gratis) account aanmaken op de website www.elektormagazine.nl.
- U dient een zo volledig mogelijke beschrijving van uw project aan te leveren, waarmee de jury zich een goed oordeel kan vormen over de missie van de start-up, de technologie van het project, het uiteindelijke product en waar nodig de geleverde diensten. Alle voorstellen zullen door de jury van de *elektor start-up challenge* worden beoordeeld.
- Om als start-up te worden beschouwd, moet uw onderneming minder dan vijf jaar geleden zijn opgericht.
- Als uw start-up is toegelaten tot deelname aan de *elektor start-up challenge*, dient de vertegenwoordiger van de onderneming persoonlijk in Parijs aanwezig te zijn tijdens het *Forum de l'Électronique 2019*.
- Alle algemene voorwaarden voor deelname aan de *Elektor Start-Up Challenge* in Parijs zijn online beschikbaar.



Deelnemers en winnaars van de *Fast Forward Award* tijdens *electronica 2018*.



De winnaar doet volgend jaar automatisch mee met de *Fast Forward Award* tijdens *electronica* in München.

Uw start-up werkt aan zijn toekomst

Waar? Paris Expo Porte de Versailles

Wanneer? Van 24 tot 26 september 2019

Hoe? Door vóór woensdag 31 juli 2019 in te schrijven

Tot ziens op:
www.elektormagazine.nl/escparis2019



De jury aan het werk, met voorzitter Clemens Valens van Elektor Labs.



De voorbereidende vergadering van Elektor en partners.

Let op:

De inschrijving sluit op woensdag 31 juli 2019.

Het aantal beschikbare plaatsen is beperkt. Bij grote belangstelling kan dit er toe leiden dat de sluitingsdatum voor de inschrijvingen wordt vervroegd. Aarzel dus niet en ga direct naar de webpagina www.elektormagazine.nl/escparis2019 om u in te schrijven.

De algemene voorwaarden voor deelname aan de *elektor start-up challenge* in Parijs zijn online beschikbaar.

Programmeerbare precisiereferentie TL431

vreemde onderdelen

Neil Gruending (Canada)

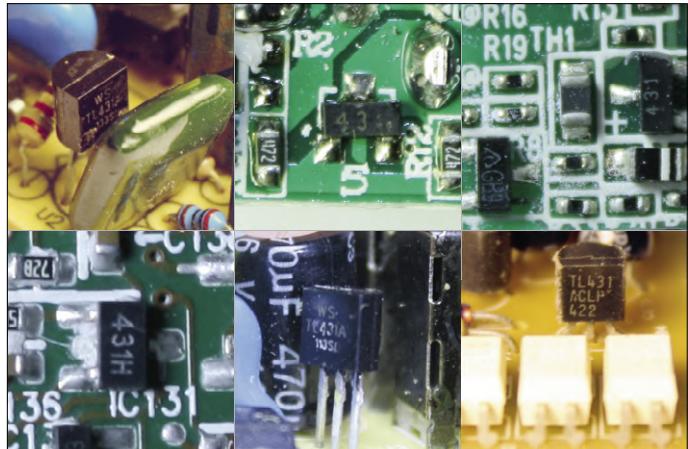
Het is goed mogelijk dat u zonder het te beseffen onlangs nog gebruik hebt gemaakt van een TL431 programmeerbare spanningsreferentie. Deze kleine onderdelen met drie pootjes vallen niet op tussen de elektronica eromheen, maar het zijn spanningsreferenties die in veel voedingen worden gebruikt. Laten we eens kijken wat ze zo populair maakt.

De TL431 van Texas Instruments werd in 1978 geïntroduceerd als een verbeterde versie van de shuntregelaar TL430. De TL431 had een betere spanningsnauwkeurigheid en -stabiliteit en werd daarom een 'instelbare precisie-shuntregelaar' genoemd. Ook beschikte hij over steilere schakelkarakteristieken dan temperatuurgecompenseerde zenerdiodes.

Een TL431 bevat een bandgap-spanningsreferentie, een opamp en een uitgangstransistor (**figuur 1**). Het schemasympool wordt meestal getekend zoals in **figuur 2**. De inverterende ingang van de opamp is verbonden met de interne 2,5-V-referentie en de niet-inverterende ingang is aangesloten op de referentie-ingang, waardoor de versterker extern kan worden geconfigureerd. De uitgangstransistor kan een stroom van 100 mA opnemen. Deze configuratie maakt de TL431 tot een veelzijdig onderdeel. Als u op de referentie-ingang een spanning aansluit, zal de TL431 zich gedragen als een comparator met een goede temperatuurgecompenseerde interne spanningsreferentie. U kunt ook de referentie-ingang met de kathode verbinden, zoals in figuur 2 getekend. Dit geeft een uitgangsspanning van 2,5 V op de kathode. Met twee weerstanden om de versterking te bepalen kunt u de uitgangsspanning op elke gewenste waarde instellen.

De TL431 wordt in talloze verschillende voedingen toegepast. Een ATX PC-voeding kan er zelfs drie bevatten, voor de standby-voeding, de hoofdvoeding en de 3,3-V-regelaar. Ze komen zo vaak voor in ATX-voedingen dat veel PWM-controller-IC's een interne TL431 hebben om het aantal onderdelen te beperken. Ze worden ook toegepast in acculaders en zelfs als versterker voor een kristalontvanger [3]. Ik hoop dat ik u heb geïnspireerd om eens een TL431 te proberen als u een precisiereferentiespanningsbron nodig hebt. Er zijn veel schakelingen voor te vinden. Kijk ook eens op Ken Shirriff's website waar hij de TL431 'ontleedt' [2] Dit leverde prachtige foto's op. ▶

180575-D-03

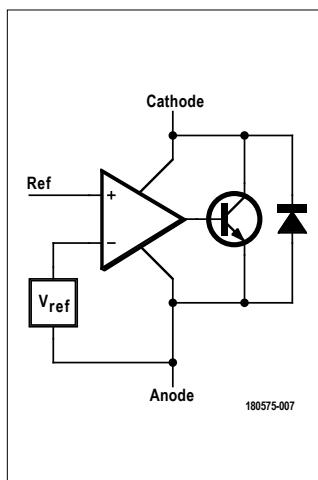


Zes voorbeelden van voedingen waarin de TL431 is toegepast.

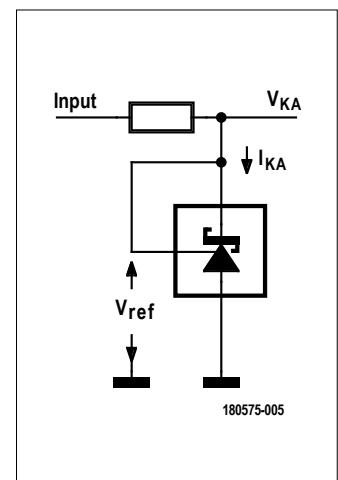
Bovenste rij: goedkope 5-V-voeding, goedkope telefoonlader, Apple iPhone-lader (met linksonder een 'GB9'-variant).

Onderste rij: MagSafe-voedingsadapter, KMS USB-lader, Dell ATX-voeding (met optocouplers op de voorgrond).

Bron: Ken Shirriff, www.righto.com/2014/05/



Figuur 1: Equivalentie schakeling van de TL431 [1].



Figuur 2: Schemasympool TL431 [1].

Weblinks

- [1] Datasheets TL431 / TL432: www.ti.com/lit/ds/symlink/tl431.pdf
- [2] TL431 'ontleed': www.righto.com/2014/05/
- [3] TL431 als versterker: www.techlib.com/electronics/crystal.html#AudioAmp

Communicatieve stroomlussen signaaloverdracht met stroom

Door Tam Hanna (Slowakije)

Waarom vinden we eigenlijk bij veel industriële sensoren een stroom- in plaats van een spanningsuitgang? In dit artikel geven we antwoord op deze vraag en gaan we nog wat nader op dit interessante onderwerp in.

Bij experimenteren in het thuislab komt het maar zelden voor dat overgangsweerstanden problemen geven. Misschien merk je er iets van als je bijvoorbeeld een goedkope insteekprint gebruikt, of een oude kabel. In industriële systemen zijn onberekenbare variabele weerstanden in leidingen echter dagelijkse kost. In dergelijke omgevingen loop je met analoge systemen met spanningsuitgangen al gauw tegen grenzen aan.

Dat probleem wordt opgelost met zogenaamde stroomlussen, waarbij de signalen niet als spanning, maar als stroom worden overgedragen. **Figuur 1** laat het principe zien. We zien dat de door de variabele spanningsval over de sensorweerstand R_{sense} veroorzaakte veranderingen van de door de leiding vloeiende stroom onafhankelijk zijn van de (in het ideale geval) oneindig hoge inwendige weerstand R_i van de stroombron en de constante belastingweerstand R_L (van het meetapparaat). Dat betekent dat de spanningsval over R_L precies evenredig is met de verandering in de stroom i_Q die door R_{sense} zijn veroorzaakt. Dergelijke stroomlussen vinden wijd en zijd toepassing in de industrie. Het stroombereik is 4...20 mA. De maximale ingangsweerstand van de meter (de meetversterker, beter gezegd) is makkelijk uit te rekenen met de wet van Ohm. Nemen we aan dat de maximale uitgangsspanning van de sensorversterker 0,5 V onder V_{cc} ligt, dan krijgen we bij een voeding van 5 V een maximale belastingsweerstand R_L van $4,5 \text{ V} / 20 \text{ mA} = 225 \Omega$. Het is geen probleem als die weerstand lager uitvalt. In de praktijk werkt men meestal met hogere spanningen (24 V is een gebruikelijke waarde).

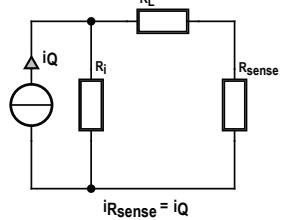
Een voor de hand liggende vraag is waarom de ondergrens 4 mA is. Waarom niet nul? Het antwoord is een korte maar waargebeurde anekdote. Er was eens in een industriegebied

een groep turbines die werd bewaakt door een procescomputer. Per ongeluk sneed een medewerker van het elektriciteitsbedrijf een leiding van een meetsonde door. De ingang voor de sonde werd daardoor logisch nul. Waarop de procescomputer vol gas gaf op de turbines, en het had maar een haar gescheeld of de turbines waren in rook opgegaan.

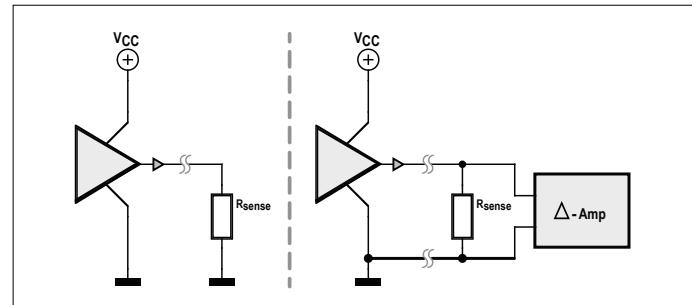
Deze bijna-ramp was het gevolg van het feit dat de procescomputer geen kabelbreuk kon herkennen. Bij het 4...20-mA-systeem kan dat niet gebeuren omdat een kleinere stroom dan 4 mA als fout wordt geïnterpreteerd en de machine wordt afgeschakeld. Technisch gezien is zo'n ondergrens geen probleem: bij het interpreteren van de stroomwaarde wordt gewoon een vaste offset met de gemeten waarde verrekend.

Realisatie met discrete componenten

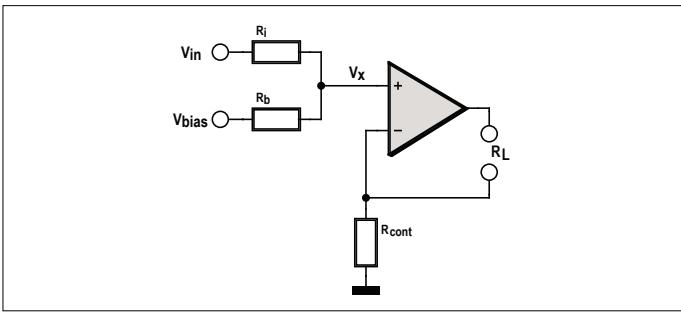
Hoewel je in de praktijk nog maar zelden stroomlussen met opamps tegenkomt, leent zo'n uitvoering zich heel goed om de werking wat nader te kunnen bekijken. De eerste en meteen belangrijkste vraag is hoe we een gemeten signaal het beste aan massa kunnen relateren. **Figuur 2** toont twee mogelijkheden. In de praktijk kan bij de configuratie links een niet onaanzienlijke fout worden geïntroduceerd. Mag je in het lab meestal wel aannemen dat de massapotentiaal overall identiek is, in een fabriekshal heb je te maken met het in- en uitschakelen van diverse actoren die allerlei storingen op de massapotentiaal geven, en die dus het signaal kunnen storen. Systeemcomponenten die met differentiële signalen overweg kunnen, verdienen daarom in elk geval de voorkeur. Merk ook op dat de stroombron zich niet per se in de sensor hoeft te bevinden, er zijn ook sensoren die een externe stroombron moduleren.



Figuur 1. Stroombronnen zijn onafhankelijk van de weerstand.



Figuur 2. De massa moet niet altijd worden doorgelust.



Figuur 3. De dimensionering van de ingangsweerstanden is niet bijzonder kritisch.

Een heel slimme schakeling in dit verband is afkomstig uit een boek van J. Michael Jacob [1], zie **figuur 3**. Deze combinatie op basis van een opteller zorgt voor een uiterst efficiënte spanning/stroom-omzetting **en** voor het toevoegen van de 4-mA-offset. Naast de eigenlijk te meten spanning is nog een instelspanning V_{bias} nodig die ervoor zorgt dat de schakeling een minimale stroom van 4 mA levert.

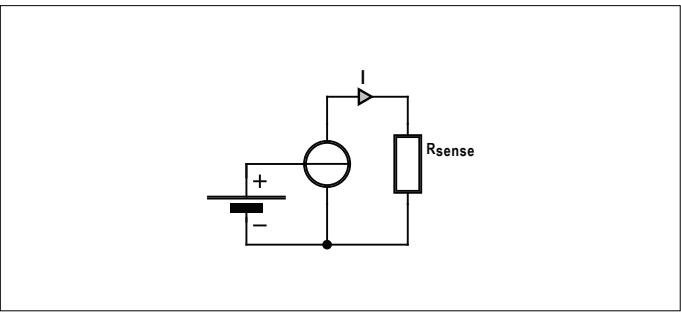
Weliswaar is de toepassing van operationele versterkers in deze tijd van procescontrollers niet meer gangbaar en kan dit ‘merkwaardig’ overkomen, toch willen we de dimensionering op zijn minst voor een deel met potlood en papier doorrekenen. Als eerste leggen we R_i en R_b min of meer willekeurig vast, voor het gemak nemen we voor beide 680 k Ω .

Vervolgens kijken we naar de uiterste waarden van het meetbereik: $V_{in} = 5$ V moet overeenkomen met $I_i = 20$ mA, en $V_{in} = 0$ V met $I_i = 4$ mA. In beide gevallen kunnen we de spanning op de inverterende ingang berekenen met $V = I_i * R_{cont}$. Aangezien een operationele versterker met tegenkoppeling altijd probeert om de zijn beide ingangen op dezelfde potentiaal te krijgen, mogen we voor beide kringen schrijven:

$$\begin{aligned}V_{in} - V_R - V_R - V_b &= 0 \\V_{in} - V_R - V_x &= 0\end{aligned}$$

De ingangen van een opamp trekken geen stroom, en dus mogen we ervan uitgaan dat de stroom door de beide weerstanden constant is.

$$\begin{aligned}V_{in} - V_b &= 2 * V_R \\V_{in} - V_b &= 2 * I_{in} * R \\I_{in} &= (V_{in} - V_b) / 2R\end{aligned}$$



Figuur 4. Primitieve sensoren krijgen hun energie van ‘buiten’ de stroomlus.

Dan hebben we voor de kortere lus:

$$\begin{aligned}V_{in} - I_{in} * R - V_x &= 0 \\V_{in} - [(V_{in} - V_b) / 2R] * R - V_x &= 0 \\(V_{in} * 2R) / 2R - [(V_{in} - V_b) / 2R] * R - V_x &= 0 \\(V_{in} + V_b) / 2 &= V_x\end{aligned}$$

En dat kunnen we invoeren in $V_x = I * R_{cont}$. Stel de beide uitersten aan elkaar gelijk om V_b en R_{cont} te bepalen. En merk vervolgens op dat de nauwkeurigheid van de schakeling in het ideale geval uitsluitend door R_{cont} wordt bepaald!

Sensorsystemen in de praktijk

Wie geïnteresseerd is in het ontwerp en de dimensionering van opamp-schakelingen voor stroomlussen, vindt in [1] ruim twintig pagina’s over dit thema. Dit boek is antiquarisch nog goed te vinden en niet duur. Voor ons is het van belang hoe ontwerpers in de praktische sensortechniek deze 4...20-mA-stroomlussen realiseren.

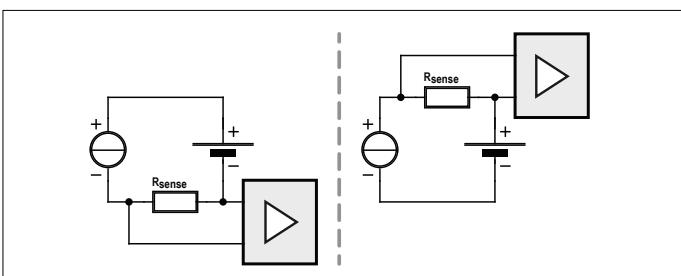
De veruit populairste configuratie ziet u in **figuur 4**. U ziet hier een extern gevoede sensor, of de uitgang van een het 4...20-mA-besturing. Als de uitgang naar massa gevoerd mag worden, dan is het makkelijk: neem een weerstand R_{sense} (met een waarde binnen het toegestane bereik) op in de stroomlus en meet de spanning erover.

In het Angelsaksische taalgebied wordt deze groep ‘primitieve’ sensoren nog verder onderverdeeld in sensoren die zichzelf voeden uit de te meten grootheid, en ‘driebenige’ sensoren die via een derde pin worden gevoed.

Shunt-sensoren zijn zoals in **figuur 5** samen met een meetweerstand in de stroomkring opgenomen, dus weer wat minder eenvoudig dan wat we eerder zagen. De niet-massagerelateerde meetwaarden worden opgenomen met een differentiële versterker. *High-side- of low-side-sensing*, dus of de shunt aan de hoge of aan de lage kant in de lus wordt opgenomen, maakt niet uit. De keuze voor een bepaalde architectuur wordt meestal gemaakt door de fabrikant. Als u een bepaalde grootheid wilt meten, dan is het verstandig om bij de keuze voor de sensor steeds de verschillende topologieën in het achterhoofd te houden.

Kant-en-klare IC's

Een opamp met een paar weerstanden en een transistor op de siliciumchip zal alleen al vanwege de kleinere *footprint* meer kosten dan de afzonderlijke componenten samen. Fabrikanten



Figuur 5. Twee leidingen van de verschilversterker volstaan voor zowel de voeding als de meetwaarde.

van opamps en sensoren voor 4...20-mA-stroomlussen doen dan ook goede zaken.

Zo is de AD5749 *Industrial Current Out Driver* van Analog Devices in wezen een analoge bouwsteen. In soortgelijke uitvoeringen wordt hij ook door andere fabrikanten geleverd. Het blokschema van de AD5749 ziet u in **figuur 6**. Hij onderscheidt zich van zijn concurrenten doordat de stroomsterkte via een SPI-interface programmeerbaar is. Bij andere fabrikanten is een bepaald stroombereik vooringesteld, waardoor geen microcontroller voor de instelling van het stroombereik nodig is.

Informatie binnenhalen van een sensor kan in zoverre lastig zijn omdat het vaak voorkomt dat een sensor sdamen met twee of drie actuatoren in een stroomlus is opgenomen. Wanneer dat niet het geval is, kan simpelweg een weerstand in serie worden geschakeld. Als het vermogen van de uitgang onvoldoende is, kan een opamp of andere versterker aan de uitgang worden opgenomen.

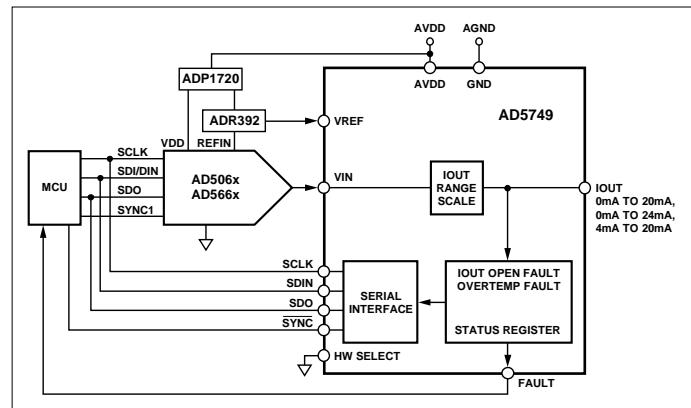
In de literatuur wordt alom verwezen naar meertraps analoge schakelingen die het 4...20-mA-stroombereik omzetten in een spanningsbereik van 0...x V. De auteur is echter van mening dat een dergelijke kunstgreep tegenwoordig nog maar zelden aanbeveling verdient. Digitaliseer die meetgegevens en voer het schalen van de meetwaarden en de foutcorrectie in de software uit! Er is relatief vrij weinig keuze in IC's die een stroomlus-meet-signalen kunnen omzetten in een spanningsbereik. Zo levert Texas Instruments de van Burr-Brown overgenomen RCV420, waarmee het 4...20-mA-singaal worden omgezet naar 0...5 V. Merkwaardig genoeg moet dit IC dubbelzijdig worden gevoed. In de datasheet zien we zelfs dat ± 15 V wordt aanbevolen, wat voor industriële toepassingen nu niet bepaald gangbaar is. Bovendien is het ding met € 7,45 per stuk (bij een Europese distributeur) tamelijk duur. We noemen hier nog een *application note* van Vishay [2], waarin een groep galvanisch volledig geïsoleerde schakelingen met de IL300-optocoupler wordt besproken. Die zijn echter wel tamelijk complex.

Met zekering en verstand

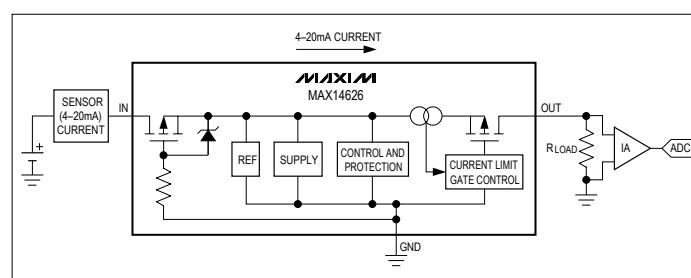
De firma Maxim/Dallas houdt zich op het gebied van sensoren voor 4...20 mA een beetje op de vlakte: één (dure) temperatuursensor en dan heb je het wel zo'n beetje gehad. In plaats daarvan richt men zich op andere componenten die zinvol kunnen zijn in stroomlus-toepassingen.

Een voorbeeld daarvan is de MAX14626 *Current Loop Protector*, speciaal voor — de naam zegt het al — het beveiligen van sensoren in stroomlussen. Grootverbruikers betalen voor deze chip slechts USD 0,61 per stuk bij afname van minstens 1000 stuks. Hij wordt opgenomen in de stroomlus en heeft verder alleen maar een verbinding met massa nodig. De beveiliging houdt in dat deze chip de lus onderbreekt zodra de stroom groter wordt dan 30 mA. Als hij bovendien thermisch gekoppeld wordt met de sensor, dan onderbreekt hij de stroomlus ook als de temperatuur boven 150 °C komt (**figuur 7**). Maxim levert de MAX14626 echter uitsluitend in een TDFN-behuizing, wat hem voor handmatig solderen dan weer wat moeilijker toepasbaar maakt.

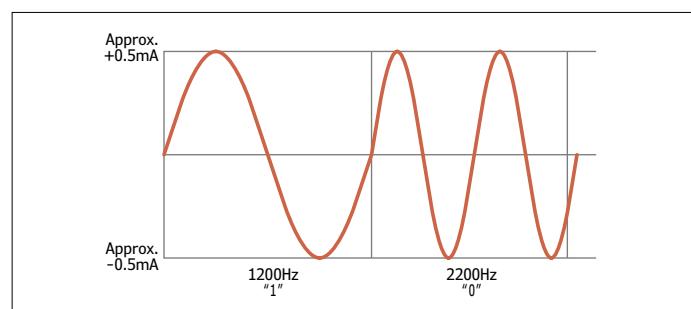
Iets dergelijks geldt voor de DS8500 van Maxim (voorheen Dallas Semiconductor), die alleen in TQFN-behuizing leverbaar is. Dit is een transceiver voor HART-technologie (*Highway Addressable Remote Transceiver*), waarmee de stroomlus met een (langzaam) digitaal modem wordt gecombineerd. HART werkt met sinusvormige FSK-signalen (zie **figuur 8**) die op het stroomsingaal worden gesuperponeerd. Het blijkt in de



Figuur 6. In combinatie met een A/D-converter ontstaat een complete controllergestuurde stroomlus (Figuur: Analog Devices).



Figuur 7. De MAX14626 beschermt sensor en ontvanger (figuur: Maxim).



Figuur 8. HART werkt met gesuperponeerde sinussen (figuur: Maxim).

praktijk dat trage bemonsteringssnelheden soms zeker aanbeveling verdienen — een samplerate van 20 kHz geeft bij HART zeker een boel ruisproblemen.

Conclusie

Een sensor-stroomlus is beslist niet altijd en overal de beste keuze, maar in de harde industriële wereld met overal stoornissen is het vaak de aangewezen oplossing. ▀

(160443)

Links

- [1] J. Michael Jacob: Industrial Control Electronics: Applications and Design, 1988, ISBN 9780134593067
- [2] www.vishay.com/docs/83710/appn54.pdf

Waterverbruik monitoren met een ESP32

met SMS-alarm bij lekkage

Denis Lafourcade (Frankrijk)

Er zijn twee soorten mensen: zij die al eens een grote waterlekkage hebben gehad en zij die er (nog) geen hebben gehad... Het risico wordt hopelijk door de verzekering gedekt, maar het is handiger om voorzorgsmaatregelen te nemen met een systeem dat waarschuwt bij de eerste tekenen van een lek. De hier beschreven schakeling is gebaseerd op een print met een ESP32-microcontroller die zorgt voor de WiFi-verbinding en een OLED-display dat de informatie ter plaatse laat zien.



Voorkomen is beter dan genezen!

Alles wat aan uw kant van de watermeter gebeurt valt onder uw verantwoordelijkheid. In het geval van een lek is het extra verbruik dus voor uw rekening, en dat kan behoorlijk in de papieren lopen:

- druppelend: 4 l/h, 35 m³ per jaar, ongeveer € 125;
- een lekkende stortbak: 25 l/h, 220 m³ per jaar, ongeveer € 800;
- een breuk in de ondergrondse leiding tussen de watermeter en het huis: meerdere m³ per uur, enkele duizenden euro's per week!

De waterleidingbedrijven zijn begonnen met het installeren van systemen om de meter op afstand uit te lezen, en deze slaan ook alarm in het geval van een

mogelijk lek. Maar de uitrol ervan gaat langzaam en in de meeste kleine dorpen zijn deze systemen nog niet geïnstalleerd. Er bestaan daarentegen ook watermeters met een impulsgever die bij iedere verbruikte liter een signaal afgeeft, en sommige meters kunnen ook achteraf van zo'n impulsgever worden voorzien. Het hier beschreven systeem maakt gebruik van deze laatste mogelijkheid en zorgt voor een continue bewaking van uw verbruik. Binnen een tijdsbestek van een paar maanden hebben drie van mijn buren bij de ontvangst van hun waterrekening ontdekt dat er een lek zat tussen de watermeter en het huis. Het lek had wekenlang bestaan zonder te zijn ontdekt, en dit had een enorm extra verbruik veroorzaakt met als gevolg een rekening van duizenden euro's! Omdat ik bang was dat dit mij ook zou

overkomen, ben ik op zoek gegaan naar methoden om mijn waterverbruik te monitoren. Gelukkig zijn veel meters, waaronder de mijne, uitgerust met een telwerk met cijferrollen. De rol voor de eenheden bevat bij het cijfer nul een magneetje, waarmee een rondgang kan worden gedetecteerd die overeenkomt met een verbruik van één liter. Er zijn in de handel impulsgevers verkrijgbaar (**figuur 1**) die op de meter worden geklikt en draadloos of via een kabel een signaal versturen als de magneet een Hall-sensor passeert. Deze impulsgevers worden gevoed uit een lithiumbatterij met een levensduur van ongeveer vijftien jaar en hebben dus geen externe voeding nodig.

Draadloos of niet?

Ik heb mij eerst beziggehouden met de overdracht van de informatie van



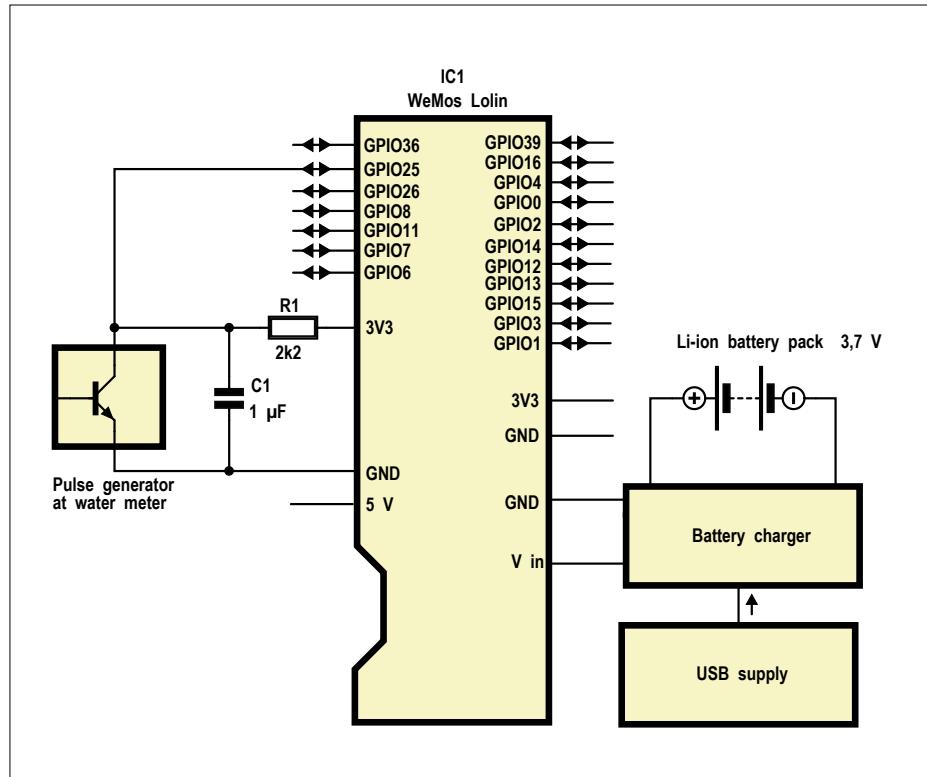
Figuur 1. De impulsgever 'in situ'.

de impulsgever naar het intelligente gedeelte van het systeem. De draadloze versie van de impulsgevers zag er aantrekkelijk uit, maar helaas is het gebruikte protocol niet openbaar, en ondanks verschillende zoekpogingen heb ik geen betrouwbare informatie over dit onderwerp kunnen vinden (als u hier meer over weet, hoor ik het graag). Omdat de inspectieput van mijn watermeter dicht bij die van mijn telefoonlijn ligt, heb ik hiervan geprofiteerd door eenvoudig via de kabelgoot van de telefoonlijn een tweedelige kabel van de sensor naar mijn huis aan te leggen. Bij andere installaties bleek dat de telecomprovider bij de aanleg van glasvezel de oorspronkelijke telefoonkabel had laten liggen, en hoefde ik geen aparte kabel aan te leggen omdat ik een van de bestaandeaderparen kon gebruiken.

Specificaties

Voordat met het ontwerpen van het systeem kon worden begonnen, moesten eerst de specificaties worden vastgelegd:

- ophalen en verwerken van de verbruiksimpulsen;
- weergave van het verbruik per uur, per dag, en het totaalverbruik;

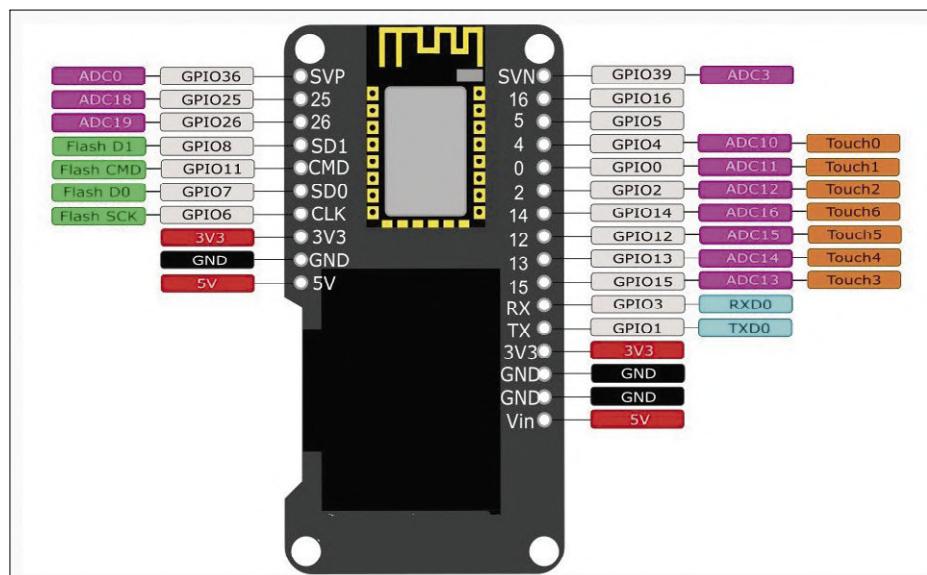


Figuur 2. Schema van de waterverbruikmonitor.

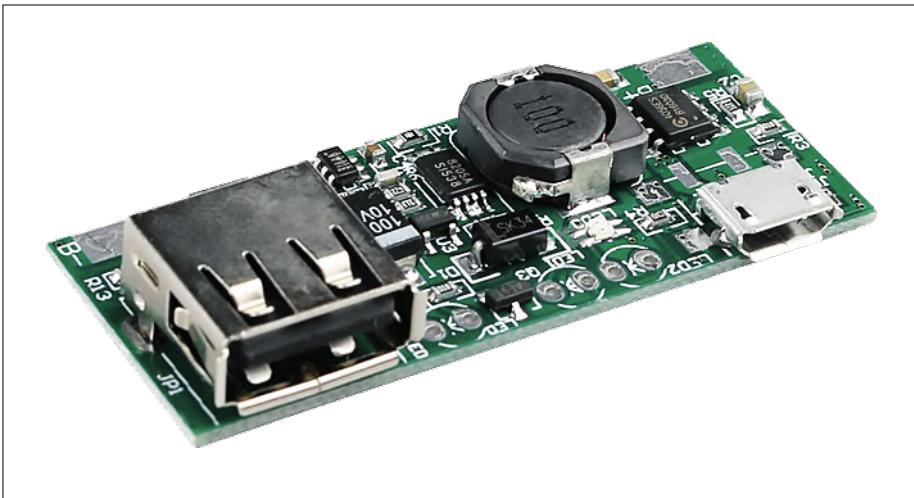
- algoritme voor de detectie van een mogelijk lek;
- bewaakte voeding;
- opslag van de gegevens in niet-vluchtig geheugen;
- netwerktoegang via WiFi en tijdsynchronisatie via NTP;
- webserver voor uitlezing en besturing op afstand;
- systeem voor alarmering via SMS.

Hardware

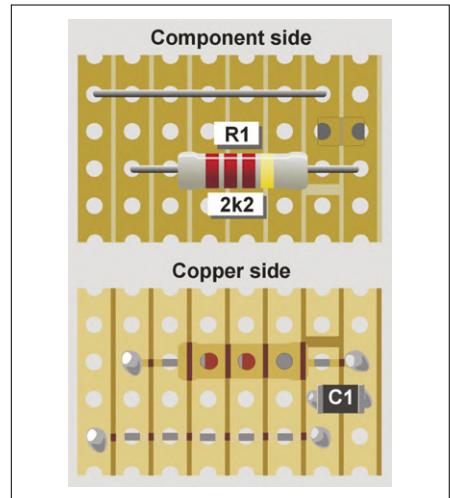
Deze specificaties hebben geleid tot de in **figuur 2** weergegeven architectuur. De schakeling is gebaseerd op een Lolin-module van WeMos (pinning in **figuur 3**), die naast een System-on-Chip (SoC) ESP-WROOM-32 ook een OLED-display van 128 × 64 pixels bevat. Dit board is in verschillende versies voor weinig geld via het internet en in de Elektor-shop ver-



Figuur 3. Pinning van de WeMos Lolin-module.



Figuur 4. Voedings-/acculadermodule.



Figuur 5. Interface-print (aangesloten op de pennen GND en GPIO25 van de Lolin-module).

krijgbaar. De ESP32 beschikt over WiFi en heeft voldoende geheugencapaciteit voor toepassingen die hiervan gebruikmaken. Bovendien is de ESP32 compatibel met de Arduino-IDE en hierdoor erg eenvoudig te gebruiken. De voeding wordt verzorgd door een Li-Ion-accu met een lader die via een micro-USB-aansluiting van 5 V wordt voorzien. Deze speciaal voor Li-Ion-accu's ontworpen lader (**figuur 4**) onderhoudt de lading van de accu en beschermt deze tegen te ver laden en

ontladen. De lader is onder verschillende namen op het internet te vinden (zoek op '3.7 V Li-ion Battery Mini USB To USB A Power Supply Module'). Als de netspanning wegvalt kan de accu het systeem minimaal 36 uur voeden.

De sensorpulsen worden aangeboden via een open collector-uitgang die bij iedere verbruikte liter naar massa wordt kortgesloten. Het ontvangen van deze pulsen gaat erg eenvoudig: een pull-up weerstand verbindt de datalijn met de 3,3V-aansluiting van het board; een condensator van 1 μ F onderdrukt eventuele storingen op de lijn. De datalijn is verbonden met GPIO 25 van de ESP32 en de massalijn met de GND-aansluiting. De interface is opgebouwd op een stukje experimenteerprint dat op de pinheader van de module wordt gesoldeerd (**figuur 5**).

Software

Het hart van het systeem wordt uiteraard gevormd door de software. Deze kan worden gedownload van de pagina bij dit artikel [1]. We zijn hierbij natuurlijk niet bezig om het wiel opnieuw uit te vinden, want veel zaken bestaan al en zijn beschikbaar gesteld door de maker-community. De meeste functies die we nodig hebben, worden door de volgende bibliotheken geleverd:

- [WiFi](#), [ESPmDNS](#), [WiFiUpd](#) en [Arduino OTA](#) voor de WiFi-toegang en het draadloos updaten van de software;
- [TimeLib](#) voor tijdsynchronisatie via NTP (*Network Time Protocol*);
- [ESP32WebServer](#) en [WiFiClient](#) voor de webserver;



Figuur 6. De drie pagina's van het OLED-scherm.

- [Wire](#), [SSD1306](#) en [OLEDDisplayUi](#) voor het OLED-display;
- en natuurlijk de functies van de [ESP32 Arduino Core](#).

Na het includen van al deze bibliotheken en het definiëren van de structuren en constanten van het programma volgt de gebruikelijke Arduino-architectuur, te weten een initialisatieroutine `setup()` en een hoofdlus `loop()`. De routine `IRAM_ATTR handleInterrupt()` zorgt voor de verwerking van de interrupt die wordt veroorzaakt door een overgang van hoog niveau naar laag niveau op de datalijn die met GPIO 25 is verbonden. Deze routine doet niets anders dan het ophogen van de waarde van de teller voor het aantal verbruikte liters. Een software-filter beschermt tegen ongewenste schakelfecten op de lijn.

De hoofdlus is verantwoordelijk voor de volgende acties:

- **Weergave op het OLED-scherm**
Met de bibliotheek [OLEDDisplayUi](#) kan de gewenste weergave eenvoudig worden gerealiseerd: er zijn drie pagina's gemaakt die elkaar elke vijf seconden opvolgen (**figuur 6**). De eerste pagina geeft de datum, het dagverbruik en de status van de lekdetectie weer (de tijd wordt op alle pagina's rechtsboven aangegeven). Op de tweede pagina ziet u de huidige meterstand, de stand bij de laatste meteropname en het verbruik sinds die meteropname. Op de derde pagina vindt u het gebruikte IP-adres en de status van de WiFi-verbinding.

• Websrvr

De server luistert op een poort die tijdens de initialisatie is gedefinieerd en waarvan u de doorgifte in uw router (*box*) hebt ingesteld. Als een gebruiker via een browser verbinding maakt, dan laat de server een pagina zien met drie tabs (**figuur 7**). Onder de tab *Visualisation* (display), waarvan de informatie iedere 5 seconden wordt ververst, vindt u naast de informatie van het OLED-scherm een tabel met het verbruik voor ieder uur van de afgelopen 24 uren. Onder de tab *Saisie* (instellingen) kunnen verschillende parameters worden gedefinieerd: de initiële meterstand, de waarde van de officiële laatste meteropname en de waarde van de alarmdrempel voor het dagelijks verbruik (zie hieronder). De toegang tot deze tab is beveiligd met een gebruikersnaam en wachtwoord. Onder de tab *A propos* (info) worden de systeemfuncties en de softwareversie weergegeven.

• Lekdetectie

Ieder uur bekijkt het programma het verbruik per uur en bepaalt aan de hand van de volgende criteria of er een lek is:

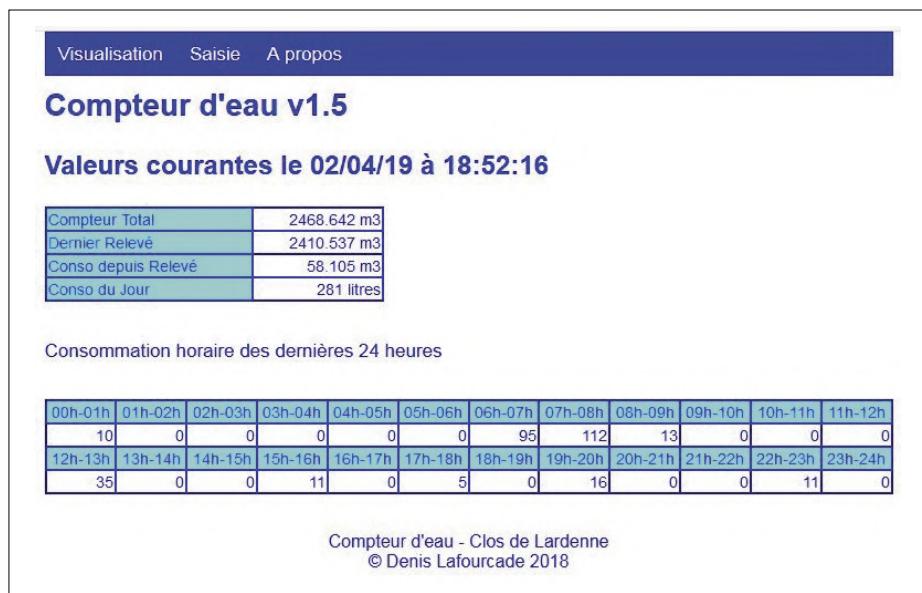
- het verbruik van de afgelopen 24 uur ligt boven een drempel;
- er zijn geen twee opeenvolgende uren zonder verbruik geweest.

Zonder lek moeten er, met name 's nachts, minstens twee opeenvolgende uren zonder verbruik verlopen. Als dit niet het geval is, dan is er een grote kans dat het ergens lekt.

Bij het vermoeden van een lek wordt het soort lek op de eerste pagina van het OLED-scherm en op de webpagina weergegeven, en wordt er via IFTTT een SMS verstuurd naar een vooraf ingesteld nummer.

• IFTTT

If This Then That is een gratis dienst die een actie uitvoert op basis van een gebeurtenis (zo'n combinatie wordt een 'applet' genoemd). In dit geval is de gebeurtenis het verzenden van een specifiek 'POST'-verzoek naar de website maker.ifttt.com, die vervolgens een SMS zal versturen naar het nummer dat bij het maken van de applet is aangegeven. Als er een lek is gedetecteerd zal de software ieder uur een SMS versturen, maar omdat IFTTT het aantal SMS-berichten per maand beperkt, verzendt het programma geen berichten meer als deze limiet is bereikt.



Figuur 7. De (Franstalige) webserver.

Bladwijzer(s) naar de webpagina

Gebruikers kunnen op hun smartphone een bladwijzer naar de webpagina aanmaken. Als uw netwerkprovider een router met loopback-functie aanbiedt hebt u maar één enkele bladwijzer nodig met het externe adres van uw router gevuld door het nummer van de gekozen poort. Zonder loopback is ook nog een tweede bladwijzer nodig met het IP-adres van de server op uw lokale netwerk (bijvoorbeeld 192.168.1.234:4321). Sommige providers leveren standaard (of op verzoek) een vast IP-adres. Andere providers veranderen dit adres periodiek, en dit vereist een regelmatige update van de bladwijzer of het gebruik van een dynamische domeinnaamserver (DynDNS), waarvan sommige gratis zijn. Hoe u hierbij te werk moet gaan valt buiten het kader van dit artikel, maar alles wat u hiervoor nodig hebt is op het internet te vinden.

• Tijdinstelling en backup

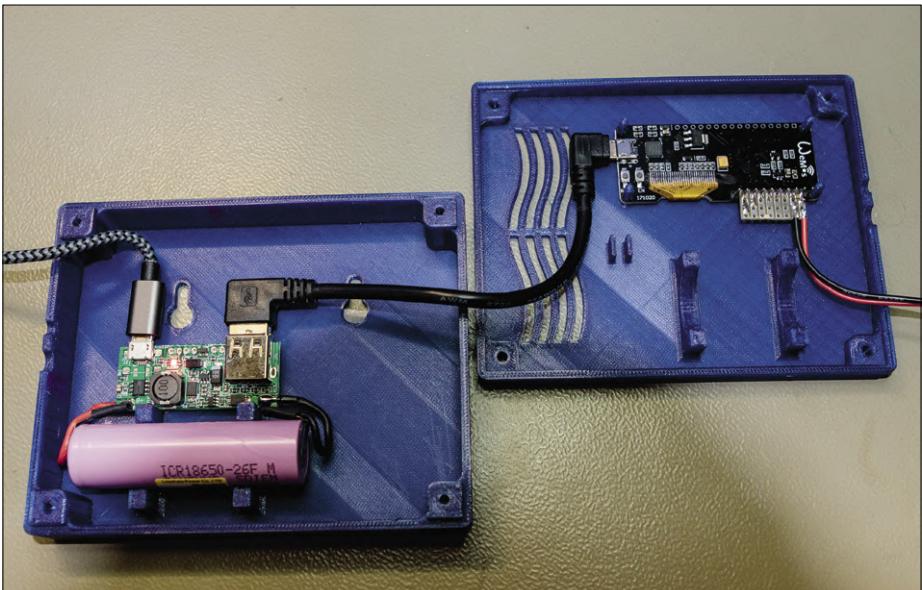
De juiste tijd wordt ieder uur automatisch ingesteld via het NTP (*Network Time Protocol*). De klok wordt bijgehouden door de realtime-klok (RTC) van de ESP32. De overgang van zomer- naar winterijd gebeurt ook automatisch.

Ieder uur maakt het programma een backup van de actuele verbruiksgegevens in niet-vluchtig geheugen (EEPROM).

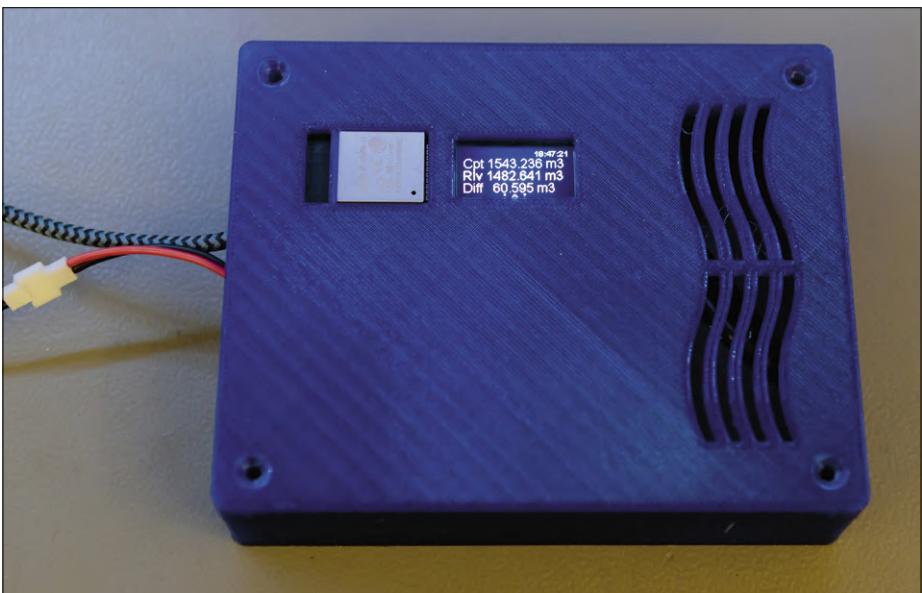
Bouw

Het systeem is ondergebracht in een speciaal hiervoor ontworpen 3D-ge-





Figuur 8. Foto van het complete systeem, ingebouwd in de behuizing.



Figuur 9. Het functionele systeem in zijn 3D-geprinte behuizing.

printehuizing (**figuur 8** en **figuur 9**). Deze behuizing heeft openingen voor het OLED-scherm, het SoC (koeling) en de ruimte rond de WiFi-antenne. Er is ook een golfmotief uitgesneden, dat er niet alleen fraai uitziet maar tevens als venti-

latie-opening dient voor de afvoer van de warmte die bij het laden van de accu vrijkomt. Aan de binnenzijde bevindt de behuizing montagesteunen voor de Lolin-module en de accu. De lader is simpelweg met dubbelzijdig plakband bevestigd.

Eerst wordt de kabel van de impulsgeber aangesloten, vervolgens een standaard netadapter met een micro-USB-kabel. Een USB/micro-USB-kabel van 10 cm verbindt de Lolin-module met de lader.

Ingebruikname

Het systeem wordt geprogrammeerd met de Arduino-IDE, die vooraf is geconfigureerd voor gebruik met ESP32-boards en met name het type WeMos Lolin32. Ook moeten alle hierboven beschreven bibliotheken zijn geladen. Voor de eerste upload moet de module op de PC worden aangesloten. De volgende uploads kunnen dankzij de OTA-functie (*Over The Air*) via WiFi plaatsvinden..

Zodra de spanning wordt ingeschakeld start het systeem, zoekt het verbinding met het ingestelde WiFi-netwerk, zet zijn klok gelijk, begint met het verwerken van de meterimpulsen en wacht op de verbinding met een gebruiker. U kunt nu de webpagina opzoeken met het adres op de derde pagina van het display, en er de actuele meterstand, de laatste opgenomen meterstand en de alarmdremel voor het dagelijks verbruik bekijken.

Conclusie

De halfjaarlijkse officiële meteropname laat bij een verbruik in de orde van grootte van 50 m³ een verschil met de door dit systeem gemeten waarde zien van slechts enkele literen, en dat is nauwkeurig genoeg. Er zijn in de afgelopen tijd een paar alarmmeldingen geweest waarmee we beginnende lekken hebben kunnen ontdekken voordat deze serieuze problemen opleverden.

Ik heb dit systeem bij een aantal van mijn buren geïnstalleerd, en deze hebben mij allemaal verteld dat ze het ook hebben gebruikt om hun verbruik te monitoren en zelfs te optimaliseren doordat ze konden zien hoeveel het verbruik is van de douche, het doen van de was, de vaatwasser, het bijvullen van het zwembad enzovoort. Ziehier een andere interessante functie die oorspronkelijk niet was voorzien! Ik ben benieuwd naar uw ervaringen! ◀

(180694-04)

Weblinks

- [1] Projectpagina op Elektor Labs:
www.elektormagazine.nl/labs/waterflow-monitor
- [2] Webpagina bij dit artikel: www.elektormagazine.nl/180694-04

EURO

CIRCUITS

- Prototypes and Small Series
 - Calculate and Order Online
 - Virtual production before ordering
 - Online DRC and DFM of the Bare and Populated PCB before ordering

Cost Effective PCB Services

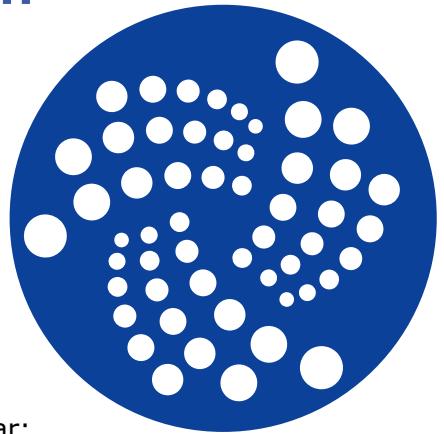
Small Orders on a Quick Turnaround - Always Available and Always On Time

www.eurocircuits.com

De cryptocurrency IOTA (en een FPGA voor de RPi)

deel 2: PiDiver – FPGA-board voor snelle berekeningen

Julia Orlitzky en Thomas Pototschnig (Duitsland)

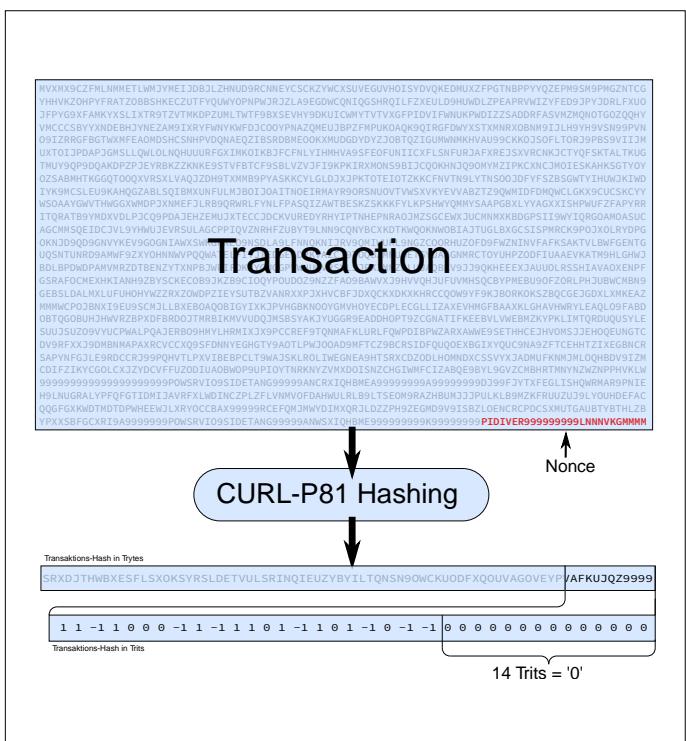


In de vorige aflevering hebben we de cryptocurrency IOTA voorgesteld, waarmee IoT-knooppunten andere deelnemers kunnen betalen – bijvoorbeeld voor het opslaan van data [1]. Om IOTA-transacties te kunnen uitvoeren, moet eerst een Proof-of-Work-rekenopgave worden uitgevoerd; hiermee wordt de transactiedatabank ('tangle') beschermd tegen spam. Bij die rekenopgave hebben zelfs x86-processoren het zwaar; daarom is er bij IOTA in voorzien dat kleinere IoT-knooppunten dit werk kunnen uitbesteden aan gespecialiseerde apparaten. In dit tweede deel van deze serie wordt een dergelijk platform voorgesteld dat is gebaseerd op een FPGA-HAT voor de Raspberry-Pi.

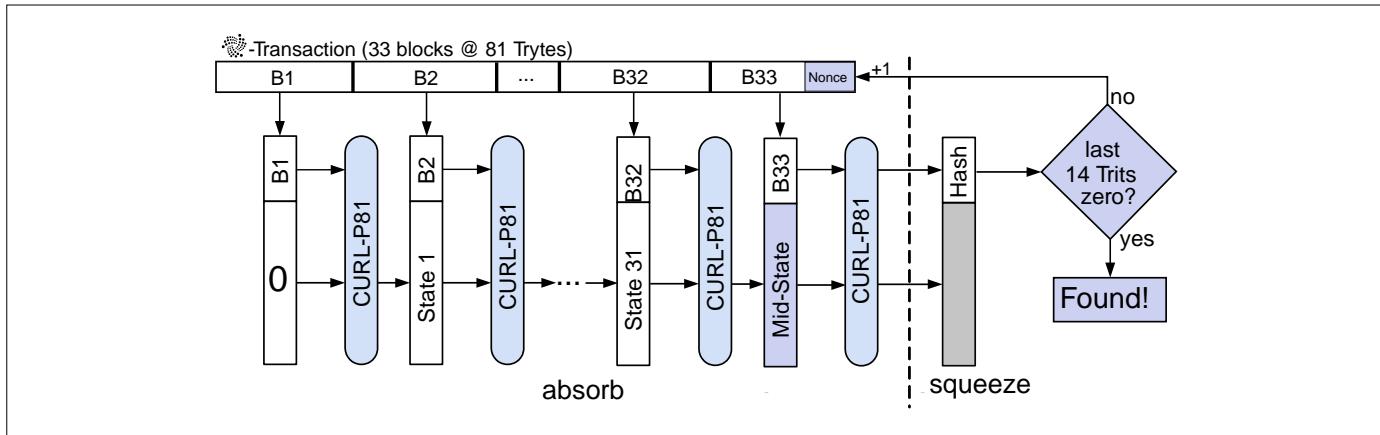
Dit artikel beschrijft een snelle en energiezuinige oplossing voor het IOTA-Proof-of-Work-probleem (PoW); de benodigde berekeningen worden door een FPGA uitgevoerd. Daardoor wordt een doorsnee-PoW-tijd van 300 ms bereikt, wat niet alleen sneller is dan een SSE-geoptimaliseerde Multi-Threaded-Library op een i5-processor (ca. 1,7 s), maar ook het vermogensopname is met 2 W aanzienlijk geringer.

TOTA's Proof-of-Work

Het idee hierachter is dat de tangle wordt beschermd tegen spam, omdat de maximaal haalbare transactiesnelheid door de computerinspanning aanzienlijk wordt verkleind. Een verificatie daarentegen kost weinig tijd, omdat alleen de hash van de transactie hoeft te worden geverifieerd. Een geldige transactiehash wordt daarom een Proof-of-Work genoemd omdat het



Figuur 1. Transactie-hash.



Figuur 2. De spons-constructie van de hash-berekening (bron: Wikipedia).

een bewijs is dat de berekeningen zijn uitgevoerd.

We maken gebruik van het hashing-algoritme CURL-P81, dat – vergelijkbaar met SHA3 – is ontworpen als een sponsconstructie. Nieuwe blokken met een vaste blokgrootte (rate) worden geabsorbeerd (absorb) in de interne toestand, die drie keer de capaciteit van een blok heeft. Vervolgens wordt de volledige toestand via een sponsfunctie omgezet in de daaropvolgende toestand (**figuur 2**). Als alle blokken zijn geabsorbeerd, kan de hash worden ‘geëxporteerd’ (squeeze).

Een IOTA-transactie bestaat uit 33 blokken van elk 81 trytes (in het diagram B1...B33). Het laatste blok bevat de teller, die wordt opgehoogd als de hash van de transactie niet aan de eis van het hoofd-net voldoet (er bestaan ook nog andere tangle-netwerken voor ontwikkeling en testen).

Omdat alleen het laatste van de 33 blokken de teller bevat, kan de toestand na het absorberen van het 32e blok (‘Mid-State’) als start-toestand voor het 33e worden gebruikt. Na het ophogen van de teller in het laatste blok hoeft zo altijd slechts een enkel blok te worden gehasht om de hash van een volledige transactie te berekenen, wat zeer veel tijd uitspaart.

PiDiver in vogelvlucht

Bij de PiDiver – de naam is samengesteld uit Raspberry Pi en het referentie-algoritme ‘Pearl-Diver’ [1] – gaat het om een HAT (Hardware Attached on Top) voor de Raspberry Pi, die IOTA’s Proof-of-Work snel en energiezuinig kan uitvoeren.

figuur 3 toont het blokschema van de PiDiver. In het centrum zien we een FPGA Cyclone 10 LP van Intel. Dit is een lid van de nieuwe Cyclone-familie (gelanceerd in 2018), waarmee Intel zich richt op goedkope en energiezuinige IoT-toepassingen. De FPGA’s, die zeer veel logica-bouwstenen bevatten, zijn niet alleen goedkoop en stroomsparend, maar zitten ook in (relatief) hobby-vriendelijke behuizingen (EQFP), waardoor ze een perfecte keus voor dit project zijn.

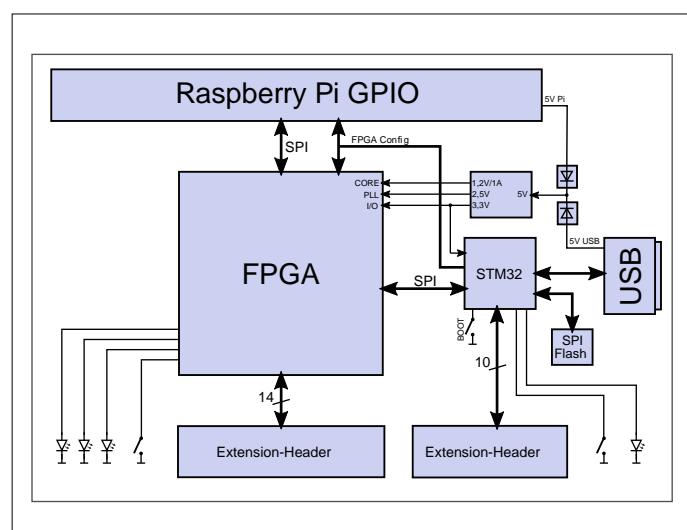
De FPGA wordt met 200 MHz geklokkt en is in staat om per klokcyclus een volledige hashing-ronde (CURL-P81 gebruikt 81 rondes) te berekenen. Bovendien berekent hij zeven hashes met verschillende tellers (noncen) tegelijkertijd. Bovenop de 81 klokcycli zijn nog twee aanvullende klokcycli nodig om het resultaat te testen, de oude toestand (Mid-State in figuur 2) te herstellen en de nieuwe tellerwaarde in te stellen. Zo komen we op ongeveer 16,8 MHashes/s. In geval van een IOTA-hoofdnet wordt (inclusief datatransfer) een gemiddelde PoW-tijd van

300 ms bereikt, ofwel 3,33 PoW/s.

Verder zien we in de schakeling een STM32-microcontroller (STM32F302), die het mogelijk maakt de FPGA ook via USB te gebruiken. De Proof-of-Work kan zo naar keuze via de SPI-interface van een Raspberry Pi, of via USB (zonder Raspberry Pi) worden doorgevoerd.

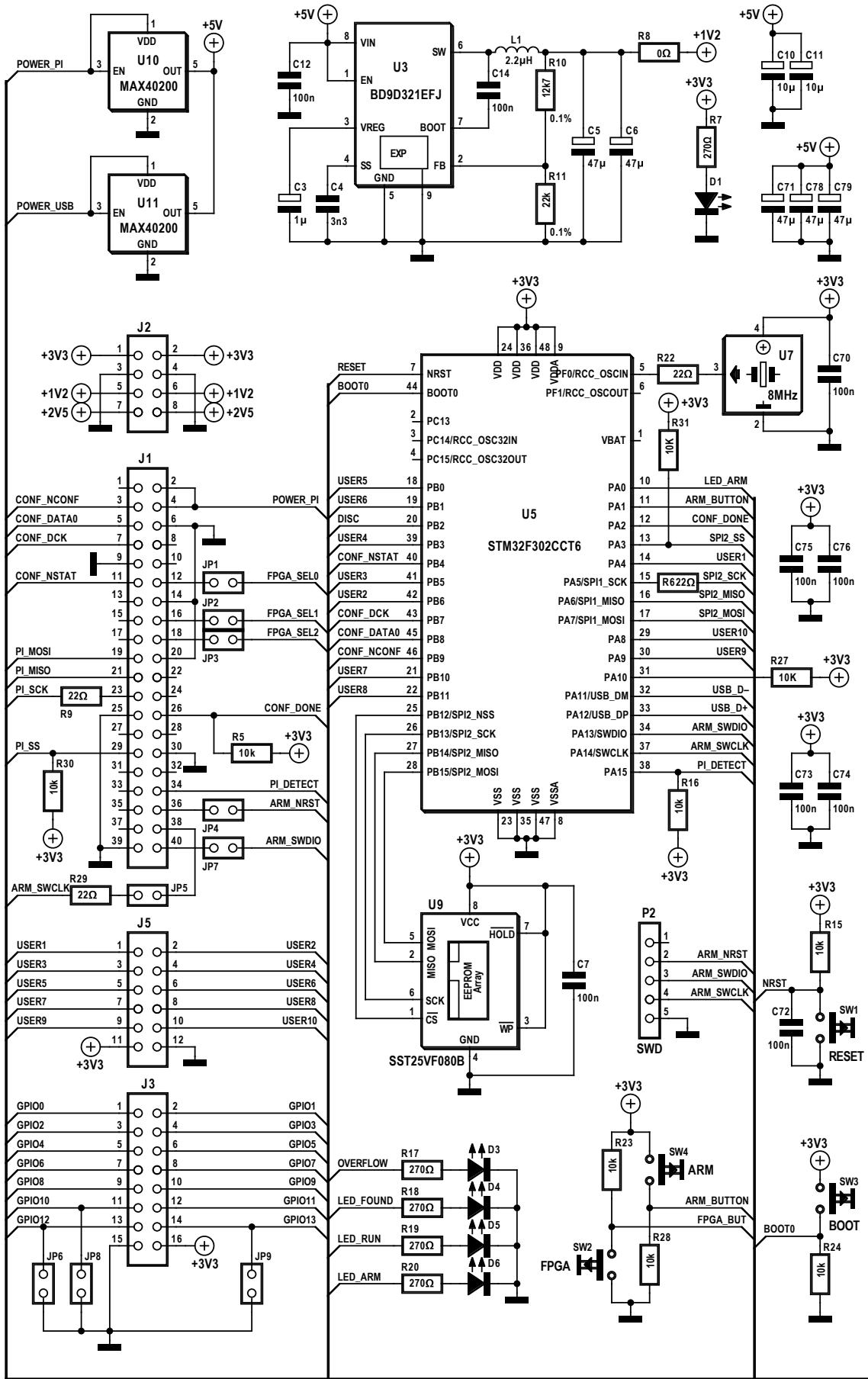
Wellicht valt op dat er geen flash-geheugen is voor de configuratie van de FPGA. Dat betekent dat de configuratie na iedere power-cyclus opnieuw in de FPGA moet worden geladen. Ook dit kan zowel via de Raspberry Pi als via USB gebeuren. De configuratie wordt door de PiDiver-library automatisch herkend en uitgevoerd. Voor toekomstige uitbreidingen is rekening gehouden met een SPI-flash-geheugen aan de STM32, dat groot genoeg is om de complete FPGA-bitstream op te slaan. Dit zou een automatische configuratie na power-up mogelijk maken, wat vooral voor embedded toepassingen interessant kan zijn (bijvoorbeeld met een ESP32-microcontroller), omdat de bitstream met ca. 800 kB voor het flash-geheugen van de meeste microcontrollers te groot is.

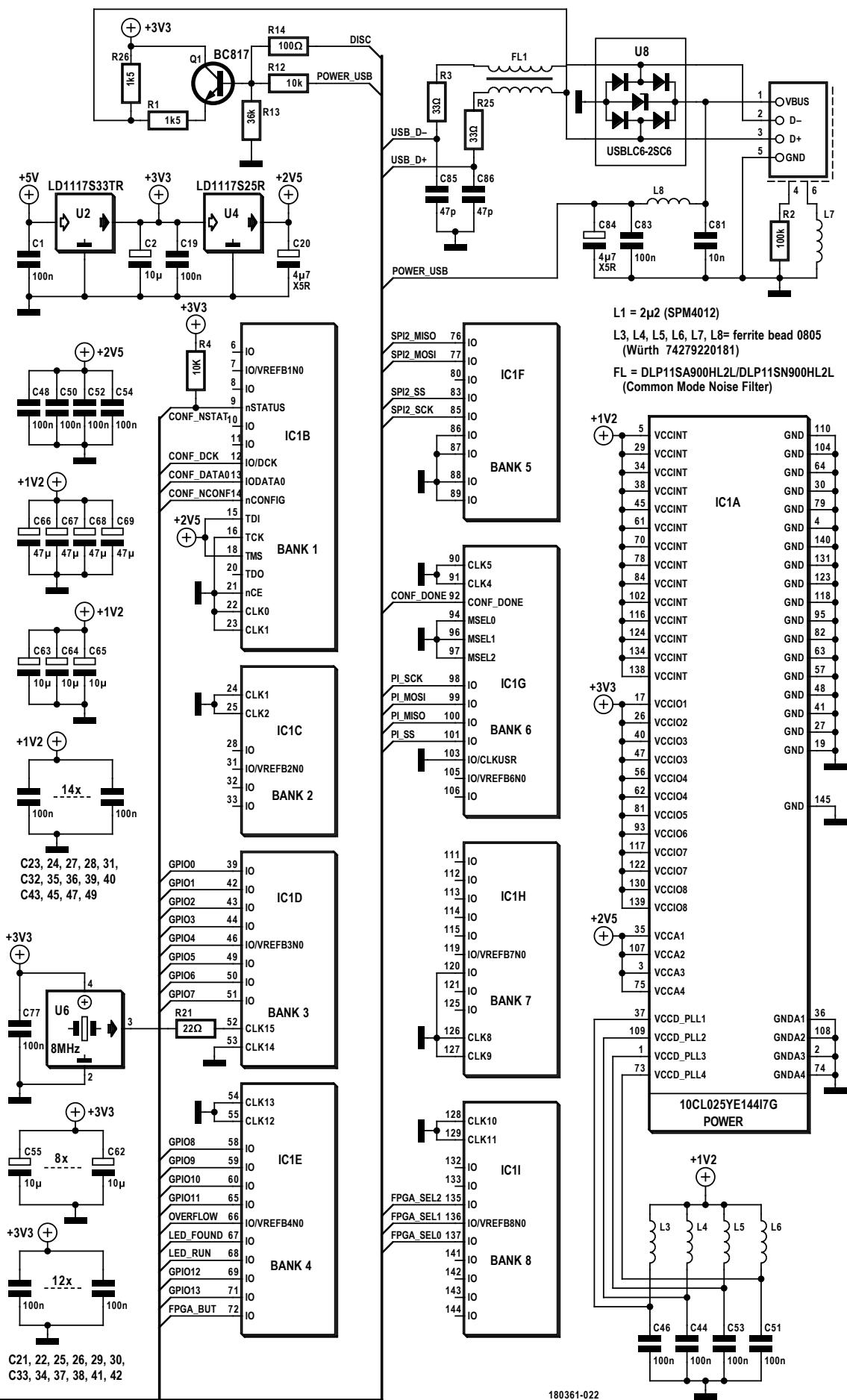
Veel meer dan dit heeft het schema in **figuur 4** niet om het lijf. Er zijn drie voedingsspanningen met elk hun eigen spanningsregelaar. Twee daarvan zijn achter elkaar geschakelde low-drop spanningsregelaars voor vaste spanningen van 3,3 V en 2,5 V,



Figuur 3. Blokschema van de PiDiver met de centrale FPGA.

Figuur 4. Schema van de PiDiver.





de derde is een instelbare schakelende regelaar, die 3 A kan leveren (waarvan hier slechts 1 A nodig is). De print kan naar keuze via de Raspberry Pi-GPIO-connector of via USB met 5 V worden gevoed. Twee volgens de fabrikant ‘ideale dioden’ (MAX40200) voeren beide 5V-spanningen samen, zodat beide potentiële spanningsbronnen niet per ongeluk met elkaar verbonden kunnen worden. In werkelijkheid is de MAX40200 geen diode, maar een logische MOSFET met een spanningsval van maximaal 85 mV bij 1 A. Van een dergelijk gering vermogensverlies wordt elke Schottky-diode jaloers!

De schakeling heeft enkele uitbreidingsmogelijkheden voor andere projecten. Hiervoor zijn twee extension-headers voorzien, waarop 14 I/O’s van de FPGA en 10 poortpinnen van de

STM32-microcontroller beschikbaar zijn voor eigen toepassingen. Bovendien heeft zowel de FPGA als de STM32 een drukknop. De schakeling is zo ontworpen, dat de interne DFU-bootloader van de STM32 kan worden benut. Hiertoe moet tijdens power-up een aparte BOOT-knop worden ingedrukt, zodat de controller in Bootloader-modus opstart. De totale stroomopname is met 350 mA zo laag dat die probleemloos uit een standaard USB-poort kan worden betrokken.

Opbouw

De print van de PiDiver [2] heeft vier lagen en is vrijwel geheel met SMD’s bestukt. Enkele daarvan (U6, U7 en FL1) zijn met de soldeerbout niet verwerkbaar, omdat de soldeerpads onder

Ontwikkeling van de PiDiver

IOTA’s Proof-of-Work moet de tangle beschermen tegen spam, maar vormt ook een hindernis voor veel valide toepassingen die data snel naar de tangle moeten sturen. Het geheel is zeer reken-intensief, zelfs voor volwassen computers, laat staan voor kleinere IoT-systemen. Zo kost het een Raspberry Pi bijvoorbeeld gemiddeld 90 seconden om het probleem op te lossen.

Daarom werd het PiDiver-project gestart. Eerst is een Proof-of-Concept op een tien jaar oud Development-Board (Altera DE1) ontwikkeld. Daarbij is de ontwikkelomgeving Altera Quartus 13.0.1 gebruikt, omdat Cyclone II FPGA’s door nieuwere versies niet meer werden ondersteund. De PiDiver-kern is geschreven in VHDL op basis van de referentiecode “Pearl-Diver” (in Java) van Sergey Ivancheglo [12]. Eerst werd een hashing-prestatie van ongeveer 4,6 MHash/s bereikt, die korte tijd later tot ongeveer 12,9 MHash/s kon worden opgevoerd. Om de FPGA met IOTA-libraries [7] te kunnen gebruiken, is de library ‘dcurl’ [8] uitgebreid met een hardware-SPI-stuurprogramma voor de Raspberry Pi en een module die de commandoset voor de FPGA implementeert.

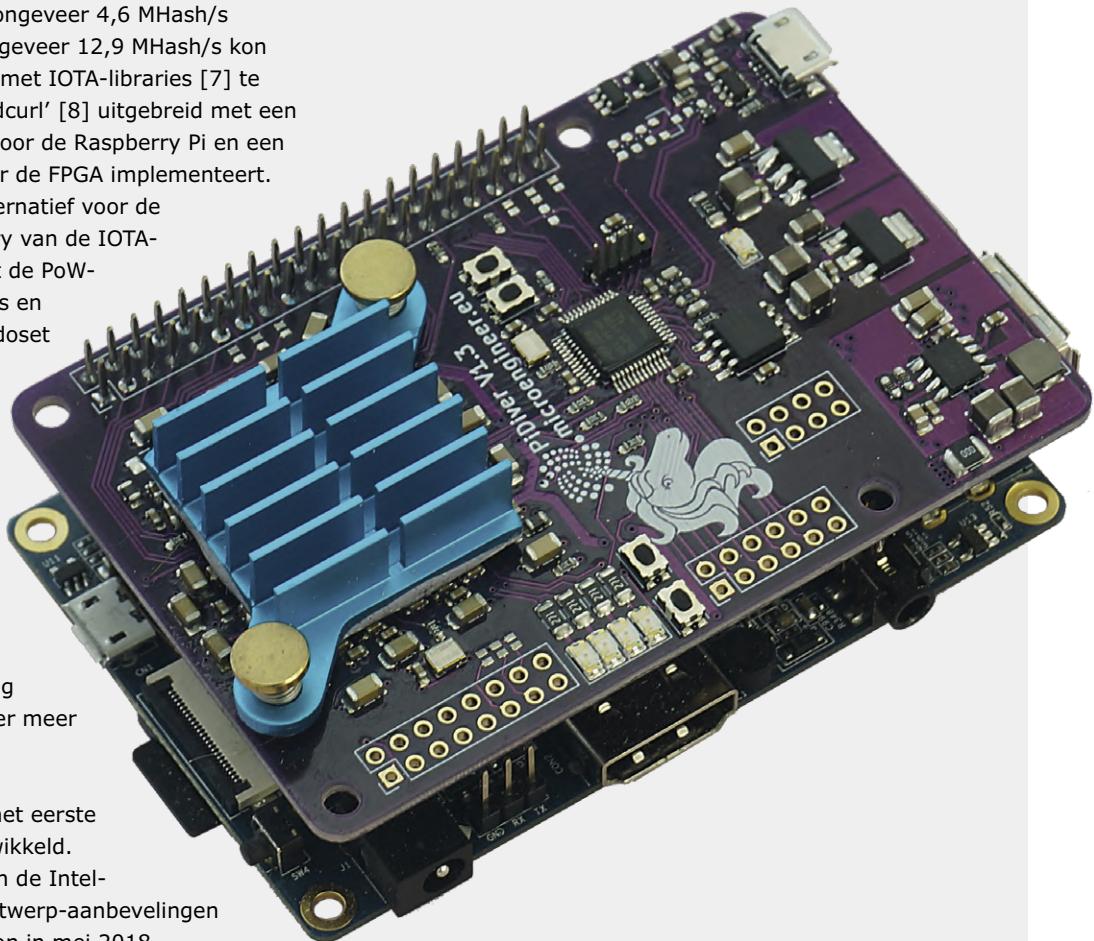
De library dcurl is een sneller alternatief voor de Proof-of-Work library ccurl (library van de IOTA-Foundation [9]), omdat die naast de PoW-implementatie in C ook nog GPU’s en CPU’s met SSE- of AVX-commandoset ondersteunt. De toenmalige wallet van de IOTA-Foundation [10] – een Electron-app – gebruikte ccurl als externe library, omdat de wallet zelf in Javascript is geschreven, wat voor PoW verhoudingsgewijs langzaam is. Deze library kon door dcurl worden vervangen. Later is een eigen PiDiver-library en een test-programma in GoLang ontwikkeld, zodat dcurl niet langer meer nodig was.

Tijdens de eerste optimalisering gedurende de PoC is met KiCad het eerste printontwerp van de PiDiver ontwikkeld.

De basisschakeling is afgeleid van de Intel-datasheet, voor de layout zijn ontwerp-aanbevelingen gebruikt. De eerste printversie kon in mei 2018

suksesvol met een hashing-vermogen van 14,8 Mhash/s worden getest. In totaal zijn er vier versies van de print. De eerste twee versies waren uitgerust met een STM32F1, die met het oog op de bootloader-functionaliteit vanaf de derde versie vervangen is door een STM32F3. Bij de tweede versie is de voeding geoptimaliseerd, omdat de schakelende regelaars van de eerste versie overgedimensioneerd waren. In versie 3 zijn ESD-bescherming en EMC-filters op de USB toegevoegd, de twee stuks MAX40200 (‘ideale dioden’) en de DFU-bootloader-functionaliteit. De vierde versie heeft montagegaten voor een koellichaam en kleinere (en dus meer) knoppen en headers voor eigen uitbreidingen.

De firmware voor de STM32-microcontroller is in C++ met Eclipse ontwikkeld.



de behuizing verstopt zitten. Voor de opbouw is dus soldeer-pasta en hete lucht aan te raden.

Weerstand R8 (een 0-Ω-weerstand aan de uitgang van schakelende regelaar U4) kunnen we beter eventjes wegleggen, want deze moet voorlopig niet worden gemonteerd. De in de componentenopstelling (zie [2]) doorgestreepte onderdelen zijn voor gebruik als PoW-versneller niet nodig. De onderdelen R12, R13, R14, Q1 en P2 zijn alleen van belang als u op de print software voor de STM32-microcontroller wilt ontwikkelen. Zij maken een via software te schakelen pullup-weerstand voor de D+ aansluiting van de USB mogelijk. Dat is vooral voor het debuggen van de STM32-software van belang, omdat daarmee de host via een microcontroller-reset wordt meegedeeld dat het USB-device opnieuw moet worden genummerd.

Helaas ondersteunt de standaard bootloader die in de STM32 vanaf de fabriek is geïmplementeerd geen schakelbare pull-up-weerstand, zodat in onze schakeling een vaste weerstand (R26) is opgenomen die deze functie voor zijn rekening neemt. Eveneens voor debuggen van belang is pinheader P2, waarop de SWD-debugging-interface naar buiten is uitgevoerd.

De meeste andere onderdelen behoeven geen verdere toelichting (behalve dan dat ze heel erg klein zijn). Wel zij erop gewezen dat de metalen vlakken onder FPGA (U1) en schakelende regelaar (U3) ook moeten worden gesoldeerd. Om handmatig bestukken mogelijk te maken zijn op deze platen grote doorvoeren gemaakt, waar doorheen de metalen vlakken kunnen worden gesoldeerd. Het is aan te bevelen de metalen vlakken door de doorvoeren heen met hete lucht te verwarmen totdat het soldeerlijm smelt. Het is af te raden om hiervoor een soldeerbout te gebruiken, omdat de daardoor toegevoerde warmte al snel door de tussenliggende lagen in de print wordt afgevoerd en de metaalvlakken moeilijk op de benodigde temperatuur kunnen worden gebracht. In het ergste geval kan de print zelfs onherstelbaar worden beschadigd!

Eerste gebruikname

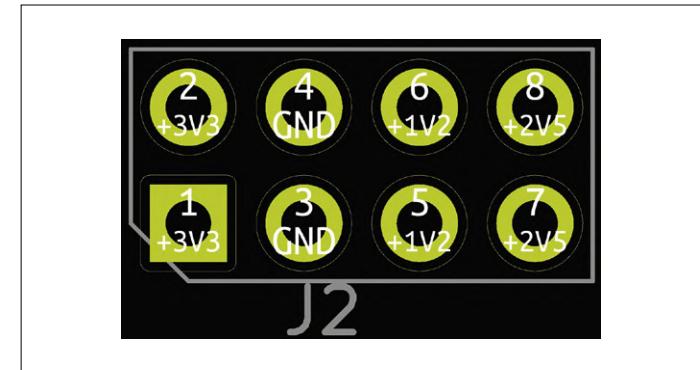
Als de print is opgebouwd is het aan te bevelen eerst de spanningen te controleren. Als eerste moet de 1,2 V aan de uitgang van de schakelende regelaar worden getest. Met deze spanning wordt (later) de kern van de FPGA gevoed. De weerstanden in de terugkoppel-spanningsdelen kunnen maar al te gemakkelijk verwisseld worden, waardoor een te hoge spanning de FPGA kapot maakt. Als de spanning in orde is, kan R8 worden aangebracht.

Vervolgens kunnen de spanningen worden gecontroleerd op de soldeervlakjes van de niet gemonteerde header J2 (**figuur 6**). De 1,2V-spanningen kunnen hier pas worden getest als R8 is gemonteerd. Als de print wordt gevoed, moet LED D1 oplichten ter indicatie dat de 3,3V-voedingsspanning aanwezig is. Aansluitend kan worden getest of de STM32-microcontroller door USB wordt herkend. Hiertoe wordt de print via een USB-kabel aangesloten op de PC, terwijl de BOOT-knop wordt ingedrukt, waarna de microcontroller zich bij de PC aanmeldt als ‘DFU Bootloader’.

Software

Flashen van de STM32-microcontroller

Om de microcontroller te flashen, moet hij eerst met de DFU-bootloader worden gestart. Hiertoe houden we de BOOT-knop ingedrukt, terwijl de print via USB wordt aangesloten. Kort daarna moet de print zich bij de PC aanmelden als



Figuur 5. Pinning van de (niet gemonteerde) connector J2 voor het testen van de voedingsspanningen.

‘DFU-Bootloader’.

Onder Linux kan het tool dfu-util [3] worden gebruikt, om de firmware te flashen:

```
dfu-util -v -d 0483:df11 -a 0 -s 0x08000000 -D  
usbdriver.bin
```

Na het succesvol flashen en een reset moet de controller zich melden als USBDiver.

GoLang-Installatie

De libraries zijn geschreven in de programmeertaal GoLang, die eerst moet worden gedownload en geïnstalleerd. GoLang is op [4] voor meerdere besturingssystemen en architecturen beschikbaar. Na installeren en instellen van de omgevingsvariabelen moet de PC opnieuw worden gestart.

Installeren van het testprogramma

Als GoLang succesvol is geïnstalleerd, kan het testprogramma worden gecompileerd. Eerst wordt (op de Raspberry Pi resp. de Linux-PC) het Git-pakket opgehaald:

```
go get gitlab.com/microengineer18/pidiver1.3/golang/  
pidiver  
cd $GOPATH/src/gitlab.com/microengineer18/pidiver1.3/  
golang/pidiver/main  
go get -d ./...  
go build
```

Nu kan de eigenlijke test beginnen. Als de PiDiver met een Raspberry Pi wordt gebruikt, dan wordt het programma als volgt gestart:

```
$ sudo ./main -t pidiver
```

(Opmerking: sudo is belangrijk, omdat anders geen rechtstreekse hardware-toegang via SPI mogelijk is).

Wanneer we de PiDiver via USB willen gebruiken, dan moet op de Linux-PC het onderstaande worden ingevoerd:

```
$ ./main -t usbdiver
```

In beide gevallen ziet de uitvoer er ongeveer hetzelfde uit:

...

```
2018/07/14 09:09:02 Found nonce: 0005e423 (mask:  
00000010)
```

```
2018/07/14 09:09:02 PoW-Time: 172ms (15.62MH/s)
```

```
2018/07/14 09:09:02Nonce-Trytes:
```

```
PIDIVER9V9999990MKVVNGMMMM
```

```
2018/07/14 09:09:02 hash: RIRQYZKVJIDQI09RBPLCJTSRTJ
```

CWFVOODAMKXFHRJMFZQLR9AJPLCNYPEW9IJCEJWRBTVTRIWXYZ
DA9999

2018/07/14 09:09:03 Found nonce: 00094b5e (mask:
000000020)

2018/07/14 09:09:03 PoW-Time: 265ms (16.03MH/s)

2018/07/14 09:09:03 Nonce-Trytes:

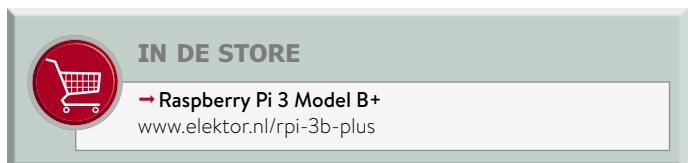
PIDIVER9T9999999UNGEKTVMMMM

2018/07/14 09:09:03 hash: EIW9MEACXYGVAQFCUGAKGMXKNC
PGMMSPQWOLGMHZWHZKUEOLNRFKJGMNSELCSLRAEVCLYCBREBW
TZ9999

...

Het testprogramma berekent noncen van toevallige transacties en geeft de transactie-hash uit, die in het geval van het IOTA-Main-net altijd op '9999' eindigt (elke 3 trits, die 0 zijn, leveren bij uitvoer als tryte een 9 op).

Er is ook een server (loopt op de Raspberry Pi, maar ook op een Linux-PC) die het commando `attachToTangle` [5] van de IOTA-API accepteert en voor echte transacties noncen berekent. Hier volstaan we echter met het verwijzen naar dieper-



gaande documentatie [6], omdat dit buiten het bestek van dit artikel valt.

Power to the Pi!

Voor een goed functioneren is het van belang kabels te gebruiken die genoeg stroom kunnen transporteren. Voor het voeden van de Raspberry Pi moet een USB-kabel worden gekozen die is bedoeld voor powerbanks. De netvoeding moet een echte 5 V en werkelijk 2 A kunnen leveren! Ook bij gebruik met een PC moet een USB-kabel worden gebruikt die tenminste 500 mA aan kan. Als een slechte kabel of netvoeding wordt gebruikt, dan verliest de FPGA mogelijk zijn configuratie op het moment dat de Proof-of-Work begint. ◀

(180361-B-02)

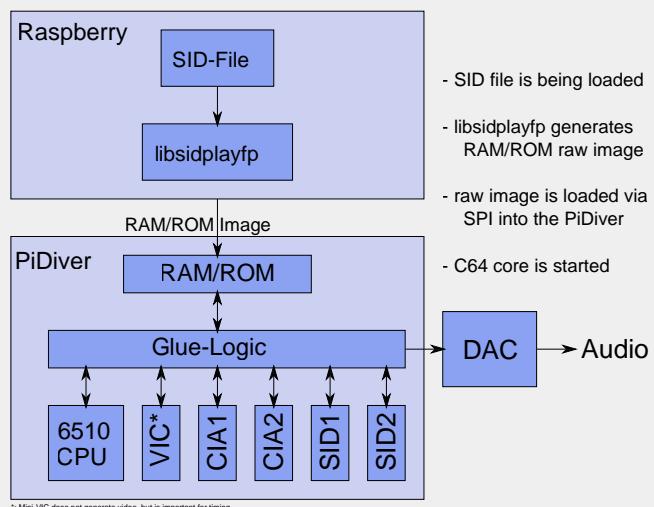
Weblinks

- [1] Eerste aflevering van deze serie: www.elektormagazine.nl/180361-04
- [2] GIT project-repository: www.gitlab.com/microengineer18/pidiver1.3
- [3] DFU-tool: <http://dfu-util.sourceforge.net/>
- [4] GoLang: <https://golang.org/>
- [5] Attach to Tangle: <https://iota.readme.io/reference#attachtotangle>
- [6] PiDiver-documentatie: <https://ecosystem.iota.org/tutorials/pidiver-usbdiver-documentation>
- [7] IOTA Ledger: <https://github.com/iotaledger>
- [8] dcurl-library: <https://github.com/DLTcollab/dcURL>
- [9] ccurl-library: <https://github.com/iotaledger/ccURL>
- [10] Wallet van de IOTA Foundation: <https://github.com/iotaledger/wallet>
- [11] PiDiver maakt muziek: www.youtube.com/watch?v=GhgDCf9oBEo
- [12] PearlDiver: <https://github.com/Come-from-Beyond/PearlDiver>

Muzikale PiDiver

De PiDiver is in principe een veelzijdig inzetbaar FPGA-board, dat via headers niet-gebruikte pinnen ter beschikking stelt voor eigen toepassingen. Als voorbeeld is een C64-emulatie getest, die via een op het board aangesloten I²S D/A-converter 8-bit muziek uit de jaren '80 in stereo kan afspelen.

Daarbij is gebruik gemaakt van de library `libsidplayfp`, die een C64-software-emulatie beschikbaar stelt, waarmee SID-muziek op een PC kan worden afgespeeld. De library genereert intern RAM- en ROM-images die op de geëmuleerde C64 worden uitgevoerd. Hoewel deze images niet bedoeld waren om op C64-hardware te worden uitgevoerd, functioneert het uitstekend. Hulde daarom aan de ontwikkelaar van deze library! Een video van de muzikale PiDiver kan op [11] worden bekijken.



Elektor Labs Pipeline



Het feit dat oude technologieën in een razend tempo door nieuwe worden opgevolgd, betekent niet dat ze uitgestorven zijn. Soms leven ze weer op, vaak om nostalgische redenen, waarbij moeiteloos delen van nieuwe technologieën worden overgenomen. Fotografie is hier een voorbeeld van.

Bouw een draagbare Bluetooth-luidspreker met lichteffecten!

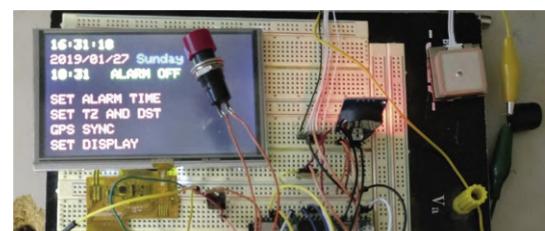
Draadloze Bluetooth-luidsprekers en -soundsystemen zijn enorm populair. Maar hoewel ze erg handig zijn, laat de geluidskwaliteit vaak te wensen over. Waarom zou u niet uw eigen systeem bouwen dat aan al uw wensen voldoet? Voor de benodigde elektronica zijn goedkope modules uit China beschikbaar die garant staan voor een goede geluidskwaliteit! En als u dan toch bezig bent, kunt u ook nog wat lichteffecten toevoegen. Hier vindt u een goed tutorial.



@ Elektor Labs: www.elektormagazine.com/labs/1795

GPS-gestuurde klok met coole grafische touchscreen-bibliotheek

Klokken en nog eens klokken. Iedereen lijkt tegenwoordig klokken te bouwen. Ze staan in de belangstelling omdat veel van deze ontwerpen iets speciaals hebben dat nadere aandacht verdient. Bij deze GPS-gestuurde klok met een Teensy-3.5-board, is het aanraakscherm met bijbehorende grafische bibliotheek erg interessant. Dit kan ook prima bij uw project passen, of dat nu een klok is of niet.



@ Elektor Labs: www.elektormagazine.com/labs/1792

Elektronische dobbelsteen zonder punten

Bent u het ook zat om de stippen op dobbelstenen te tellen? Gebruik dan deze schakeling met een 7-segment-display dat de waarde duidelijk aangeeft. Makkelijk te bouwen, gebaseerd op een ATtiny2313-microcontroller en geprogrammeerd in Bascom. Door de druktoets in te drukken gaat de dobbelsteen 'rollen', wanneer de druktoets weer wordt losgelaten verschijnt een willekeurig getal tussen 1 en 6.

Dit biedt u de mogelijkheid om 's avonds, als de kinderen in bed liggen, in alle rust te spelen.



@ Elektor Labs: www.elektormagazine.com/labs/137

Klassieke donkere kamer met moderne rode LED-lamp en timer

Bij al die digitale fotocamera's met hun miljoenen pixels en kleuren en ongelooflijk krachtige beeldbewerkingssoftware is het nauwelijks voorstellbaar dat dat er nog steeds mensen zijn die vele uren in een donkere kamer doorbrengen en hun vingers in giftige chemicaliën dopen om monochrome negatieven en positieven te ontwikkelen. Maar die zijn er nog steeds, zoals dit project laat zien dat bij Elektor Labs werd gepost. ►



@ Elektor Labs: www.elektormagazine.com/labs/1797

Basiskennis AS-Interface

zo werkt het bussysteem voor de industriële automatisering

Tam Hanna (Slowakije)



Ooit werd alles aangestuurd met zelfgemaakte bedrading en merkgebonden dataformaten. Maar deze manier van werken is in de industriële automatisering al lang niet meer de gewoonte. In plaats daarvan wordt maximaal ingezet op volwaardige bussystemen. Daar is, naast de duidelijke voordelen van eenvoudig onderhoud en uitbreidbaarheid, nog een andere reden voor: een breed geaccepteerde industrietstandaard garandeert de langdurige beschikbaarheid van hard- en software, in elk geval tot op zekere hoogte. We geven een overzicht van de AS-interface.

Als complexe fabricageprocessen moeten worden geautomatiseerd, dan oriënteert men zich met de in **figuur 1** getoonde automatiseringsboom. Op de onderste laag bevinden zich de sensoren en actuatoren. De communicatie daarmee kan, zowel vanwege de kosten als vanwege de efficiëntie, niet gemakkelijk via een ethernet-kabel lopen. Verbindingen over korte afstand met protocollen zoals SPI of I²C blijken vaak niet opgewassen tegen de vaak erg ruwe omgeving in de maakindustrie [1]. Voor het aansluiten van sensoren en dergelijke zijn twee breed geaccepteerde technieken: de als ASi bekend staande AS-interface (Actuator-Sensor-Interface) en het IO-link-protocol (genormeerd onder de aanduiding SDCI) uit de Profibus-omgeving. Het grootste voordeel van de AS-interface is dat een master een groot aantal slaves via één enkele poort kan aanspreken. Bij IO-link zijn normaal gesproken voor elke slave een poort en een kabel nodig. In de Engelstalige literatuur wordt IO-link geschikt beschouwd voor situaties waar grotere hoeveelheden data moeten worden getransporteerd.

Het komt trouwens ook voor dat ASi en IO-link ‘collectief’ worden ingezet. Het in **figuur 2** weergegeven en onder [2] in detail beschreven systeem van Bihl-Wiedemann gebruikt IO-link voor het verzamelen van informatie. De verdichte datagrammen gaan dan via ASi naar de master.

Een kwestie van versie

De geschiedenis van de AS-interface gaat terug tot 1990, toen een groep van bedrijven een samenwerkingsverband oprichtte onder de naam *AS-International Association* [3] dat zou moeten functioneren als een soort belangenbehartigingsorganisatie. Deze organisatie heeft de rechten op de naam *AS-Interface* en verkoopt producten rondom de officiële standaard. Op dit moment kost het lidmaatschap ongeveer 4.000 €/jaar (voor het geval dat u lid zou willen worden). Voor wie alleen een blik op de standaard zou willen werpen, kan het ook goedkoper: de AS-Interface wordt gespecificeerd via een groep van industrietstandaards; de meest populaire is

EN 62026-2:2015. Het bijbehorende document kost bij het officiële normeringsinstituut ‘slechts’ ongeveer 250 €. Als u de moeite wilt nemen om persoonlijk te gaan speuren, dan kunt u het ook vinden in de bibliotheken van technische universiteiten en hogescholen. En als de documenten daar niet aanwezig zijn, is er via het interbibliothechair leenverkeer wel aan te komen. Wie zich in deze materie wil gaan verdiepen, zal merken dat er op dit moment eigenlijk twee versies van ASi tegelijk op de markt zijn. Alleen versie 3.0 is momenteel werkelijk verkrijgbaar, zowel de specificatie als de hardware. Maar sinds de beurs *SPS IPC Drives 2018* wordt overal versie 5.0 aangebroken, die verschillende verbeteringen op detailniveau bevat.

Het lastige aan deze, op zich normale, ontwikkeling is dat de exacte protocolspecificatie van versie 5.0 (op het tijdstip van het schrijven van dit artikel) nog niet eens voor alle leden beschikbaar is. Volgens insiders is die alleen verkrijgbaar voor organisaties die meewerken aan de verdere ontwikkeling van het protocol.

Dat leidt tot ‘interessante’ verschijnselen: het transceiver-IC voor versie 3.0 (details verderop in dit artikel) komt van de firma ZMD, die intussen door IDT is opgekocht en daarna weer door Renesas is opgesloten. Voor versie 5.0 zal daarom met Renesas worden samengewerkt, hun slogan zou inmiddels kunnen luiden: “Wij implementeren ieder protocol”.

ASi-kabels

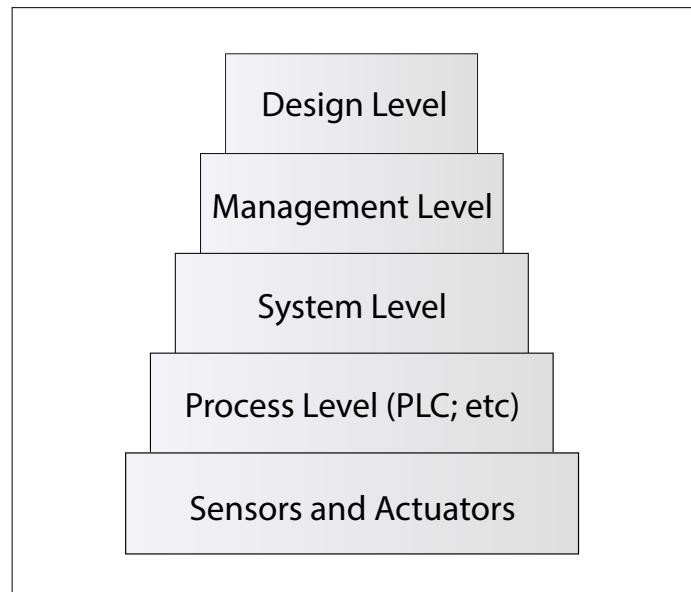
De ‘Multidrop’-mogelijkheid van ASi heeft onder meer te maken met de opbouw van de kabel. Het gaat om een tweedraadige kabel, die de signalen AS+ en AS- overdraagt. AS- is de ‘massa’, terwijl AS+ een positief signaal voert. We moeten er meteen bij zeggen, dat ‘aarden’ van AS- beslist niet is toegestaan.

De ASi-datakabels zijn bij concrete installaties vrijwel altijd in het geel uitgevoerd. Typisch is de in **figuur 3** getoonde asymmetrische kabel, die er in doorsnede uit ziet als een soort neus. Een optionele zwarte leiding met een gelijksoortige opbouw is bestemd voor het aanleveren van extra energie voor de apparaten.

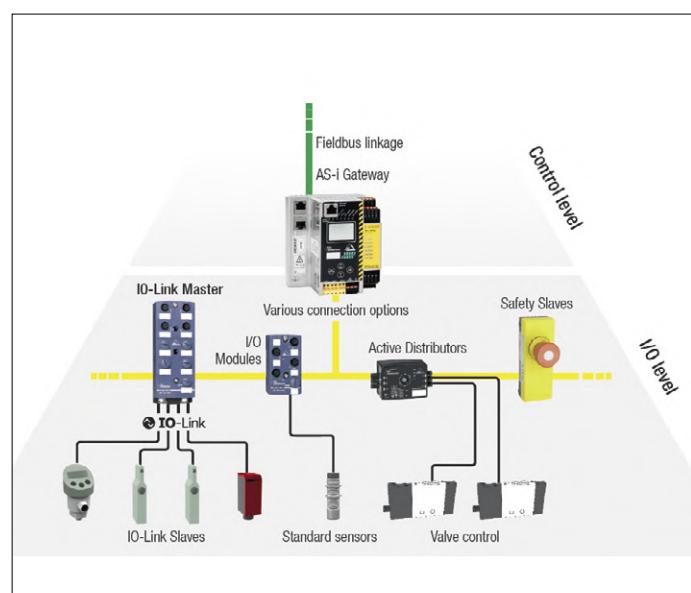
De meeste ASi-apparaten hebben connectoren voor de kabel van figuur 3, die voorzien zijn van tanden die zich door de isolatie boren en een galvanisch contact met de beide draden maken. Het toevoegen van een nieuw apparaat aan een ASi-bus is daarmee heel gemakkelijk en ‘failsafe’. Hoewel deze werkwijze gebruikelijk en praktisch is, mogen we de ASi-bus ook routeren via willekeurige andere kabels.

Net als bij de voor directe aansluiting van sensoren aan microcontrollersystemen geschikte One-Wire-bus, geldt ook bij de AS-interface dat de voeding van apparaten met een kleine energiebehoefte rechtstreeks via de buslijn loopt. De nominale spanning is daarbij 24 V. Volgens de specificatie mogen we in theorie maximaal 8 A door de bus jagen. Realistischer is de aanbeveling, dat we vanaf 2 A rekening moeten houden met de spanningsval over de kabel plus connectoren.

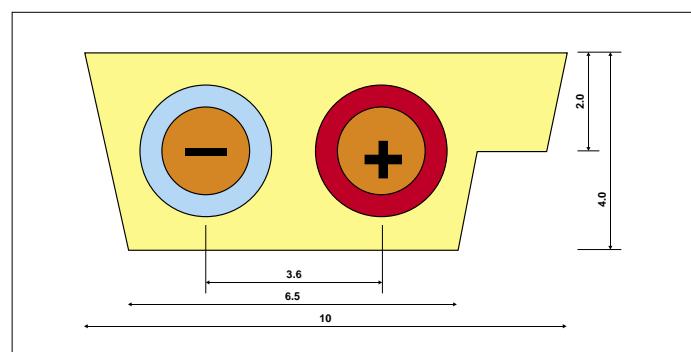
Het is belangrijk dat de AS-interface potentiaalvrij werkt en geen referentie naar aarde heeft. Het is, zoals al eerder vermeld, expliciet verboden om aardingen uit te voeren. Naast het gebruik van volwaardige AS-slaves is er ook de mogelijkheid om ASi-apparaten alleen van voeding te voorzien. Die mogen dan bij 24 V maximaal 400 mA aan stroom trekken, maar krijgen geen adres en zijn dus uitgesloten van de communicatie. Omdat een ASi-bus fysiek maximaal 100 m lang



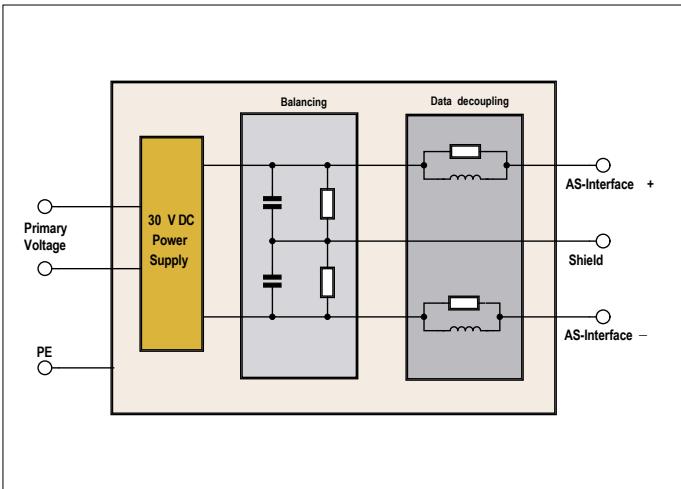
Figuur 1: De hiërarchische lagen van de automatiseringsboom.



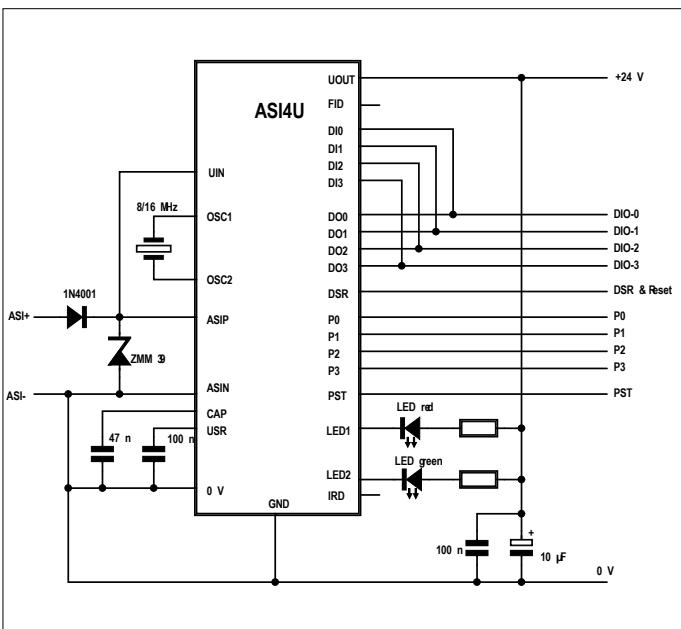
Figuur 2: Van de veldbus naar sensoren en actuatoren (bron: [2]).



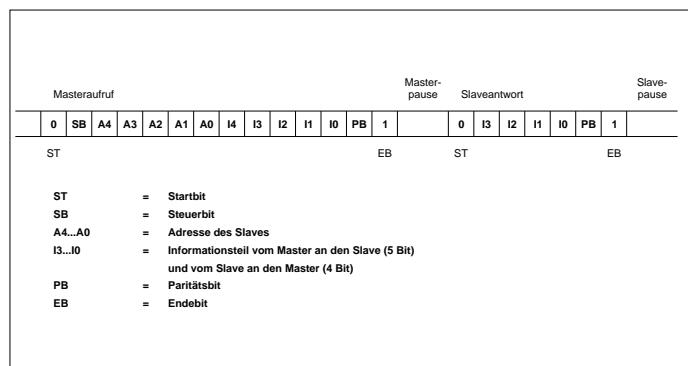
Figuur 3: Doorsnede van een ASi-kabel (doorsnede van de geleider 1,8 mm). De ‘neus’ verhindert onbedoeld ompolen (afbeelding: naar Bihl+Wiedemann).



Figuur 4: De als PELV aangeduide netvoedingen zijn tamelijk complex van opbouw (bron: [4]).



Figuur 5: De transceiver-chip heeft erg weinig externe componenten nodig (bron: [6]).



Figuur 6: Dataformaat van ASI-berichten (bron: [7]).

mag zijn (en bij goede afsluiting zelfs maximaal 300 m), is hij dus goed geschikt voor het voeden van externe systemen die weinig stroom trekken. Zo'n apparaat wordt gewoon aangesloten aan AS+ en AS- en doet dan zijn werk.

De voeding in een ASI-systeem werkt normaal gesproken met een speciale netvoeding, die is opgebouwd zoals in **figuur 4**. Zelfbouw en eigen ideeën worden bij deze professionele toepassingen ten sterkste afgeraden. Op de webpagina van de ASI-organisatie [3] zijn links te vinden naar bedrijven die gecertificeerde producten aanbieden.

Het is daarbij interessant, dat deze netvoedingen over het algemeen een gelijkspanning van 29,5 V tot 31,6 V genereren. Deze hogere spanning dan 24 V is nodig om spanningsval in kabels en connectoren te compenseren. Hoewel een ASI-systeem een grote immuniteit voor storingen heeft, is dat geen vrijbrief voor slordige of wilde bekabeling. Als u een ASI-kabel in dezelfde kabelgoot legt als netspanningslijnen waarop veel storing aanwezig is, kunt u te maken krijgen met moeilijk te determineren problemen. Een presentatie van Phoenix Contact [4] geeft verdere informatie over de opbouw van een installatie.

Communicatie

Uit de in figuur 3 geschetsste kabel blijkt duidelijk, dat er geen gescheiden lijnen voor voeding en data zijn. De informatie voor de toepassing moet dus eerst nog gedemoduleerd/omgevormd worden, voordat die kan worden gebruikt.

De data wordt verzonden via APM (alternerende pulsmodulatie). Deze modulatie werkt met een Manchester-codering, waarbij het kloksignaal en de databits worden gecombineerd. De Manchester-gecodeerde bitreks wordt door de zender omgezet in een stroom. Volgens de vrij beschikbare informatie heeft deze stroom een amplitude tussen nul en 60 mA.

Negatieve flanken worden gecodeerd met een toename van de stroom van de zender en positieve flanken met een afname. Door de zelfinductie van de leiding worden deze stroomveranderingen omgezet in spanningsimpulsen, die een niveau van ongeveer ± 2 V om de nominale spanning hebben. De start van een datagram wordt altijd aangegeven met een negatieve puls. Dat is belangrijk voor het triggeren van digitale oscilloscopen en analyzers.

Helaas is er bij de traditionele oscilloscoöpfabrikanten tot nu toe maar weinig belangstelling voor een gebruiksklare integratie van geschikte decoders in hun producten. Zelfs de apparaten van Pico Technology kunnen dat niet van huis uit. Gelukkig is het wel mogelijk dankzij een API [5] van een tool van een andere aanbieder.

Het is niet waarschijnlijk dat iemand voor een eigen ASI-module ook zelf de decoder-hardware zal willen implementeren. De al genoemde firma IDT biedt een transceiver-chip aan van het type ASI4U. Deze kan als master maar ook als slave werken. In **figuur 5** zien we de basisschakeling uit de datasheet. Wat vooral opvalt zijn de beide LED's, waarvoor IDC in het datasheet groen en rood voorschrijft. Er is namelijk gespecificeerd, dat zulke LED's op slaves beschikbaar zijn. Ze geven aanwijzingen over de actuele bedrijfstoestand; meer informatie hierover is te vinden in de bovengenoemde datasheet. De DIX- en DOx-pennen dienen als in- en uitgangen. Bevindt een via de transceiver aangesloten module zich in de slave-modus, dan staat daar de via de bus overgedragen informatie ter beschikking. De met Px aangeduide ingangen dienen voor het

vastleggen van parameters. Die spelen vooral een belangrijke rol bij de hier niet besproken master-modus.

De communicatiestandaard

Nu we de elektronische hardware van een ASi-systeem hebben besproken, kunnen we gaan kijken naar de logica van het communicatieformaat. De eigenlijke communicatie vindt volledig onder besturing van de master plaats. De master voert een cyclische polling uit, waarbij hij alle slaves die zich op de bus bevinden na elkaar afwerkt. De slaves antwoorden normaal gesproken met 4 bits aan nuttige data, waarna de cyclus weer opnieuw begint (zie **figuur 6**).

Het adresformaat legt het maximale aantal adresseerbare apparaten op de bus vast. In ‘normale’ datagrammen zijn slechts vijf adresbits beschikbaar; het adres ‘0’ is gereserveerd voor een nieuwe slave. Er blijven dus netto 31 mogelijke adressen over om slaves aan te kunnen spreken. Periferie die op de bus verschijnt met adres 0, krijgt van de master een nieuw, nog vrij adres toegewezen.

Uit de cyclische polling volgt een deterministische reactietijd in een ASi-systeem. Bij ASi 3.0 is die normaal gesproken ongeveer 5 ms, terwijl de nieuwe versie 5 bij 24 deelnemers een cyclustijd van slechts 1,2 ms belooft. Bovendien is het maximale aantal aan slaves bij ASi 5.0 een stuk groter, namelijk 96. Er kunnen zowel data als parameters worden verzonden. Een informatiebit geeft aan welk soort informatie er op een bepaald moment via de bus wordt verzonden. Het besturingsbit geeft aan of er adres- of commandogegevens volgen.

Wat te doen?

Als u te maken krijgt met een al bestaand ASi-systeem of als u een nieuw ASi-systeem wilt gaan opbouwen, moet u economisch denken. Het zelf ontwikkelen van modules met sensoren kan leuk zijn, maar het is in de praktijk niet efficiënt. Er zijn al veel bedrijven die zich daarmee bezighouden. Ze bieden vrijwel alle denkbare sensoren en actuatoren aan met een AS-interface. Het is dus veel effectiever om uit deze componenten een compleet systeem samen te stellen en dat samen met uw consulting-diensten aan klanten te verkopen.

Voor wie absoluut zelf met ASi wil experimenteren, zit er niets anders op dan zich aan te melden voor een lidmaatschap in de standaardiseringsorganisatie. De norm kan dan wel alle nodige informatie bevatten en de transceiver-chip ASI4U is volgens oemsecrets.com werkelijk verkrijgbaar op de vrije markt, maar dan komt er een grote MAAR.

Het probleem is namelijk, dat het hier om markenrecht gaat. Zodra iemand zonder expliciete vergunning gebruik maakt van de benaming ASi of het ASi-logo gebruikt, wordt het juridisch

Lees beslist de datasheet!

Dit artikel kan vanwege de lengte alleen een oppervlakkig kijkje bieden op de ASi-standaard. Als u er zelf mee aan de slag wilt, zult u werkelijk het datasheet *asi4you* van IDT [6] moeten bestuderen.

Informatie uit Kempten

Een snel overzicht van versie 3.0 van de ASi-standaard is trouwens beschikbaar op het YouTube-kanaal van de Staatlichen Berufsschule Kempten.

Onder de URL www.youtube.com/user/BS1Elektro/videos zijn vier video's over dit thema in het Duits te zien.

In het Nederlands is hierover op YouTube momenteel niets te vinden, maar als u liever naar Engelstalige video's kijkt, kunt u uw hart ophalen, zoek gewoon op YouTube naar “AS-Interface”.

erg gauw erg gevaarlijk. Als u uw systeem aanbiedt op de vrije markt, duurt het niet lang voordat u wordt aangeklaagd. Deze situatie is voor in laboratoriumwerk geïnteresseerde elektronici natuurlijk niet prettig. Maar werken met ASi-systemen kan u wel een hoop geld opleveren, want in de industriële automatisering zijn relatief hoge tarieven voor ontwikkelaars en consultants gebruikelijk. Dat kan een mooie compensatie zijn voor al het formele gedoe. ►

(190124-03)



IN DE STORE

→ MonoDAQ-U-X (50 kS/s) – Flexibel

USB-datalogsysteem

www.elektor.nl/monodaq-u-x

→ PicoScope 2205A

www.elektor.nl/picoscope-2205a

Links en literatuur

- [1] J. M. Jacob, Industrial Control Electronics, pagina 260 e.v.
- [2] ASi-informatie: www.bihl-wiedemann.de/en/applications/communication/as-interface-and-io-link.html
- [3] AS-International Association: www.as-interface.net
- [4] Phoenix Contact: www.phoenixcontact.com/assets/downloads_ed/global/web_dwl_technical_info/instrecomm10_e.pdf
- [5] Pico Technology ASi-tool: www.picotech.com/library/picoapp/as-iexpert-network-diagnosis-tool
- [6] ASi-datasheet van IDT: www.idt.com/document/dst/asi4u-datasheet
- [7] DESY: http://www-mks2.desy.de/content/e3740/e5177/e7190/e7997/e8006/e9274/e9656/index_ger.html



(Bijna) alles wat u altijd al wilde weten over... analoge schakelingen

Met antwoorden van **Ton Giesberts** (Nederland)

V Hoe kun je de ruis in een versterker zo laag mogelijk houden?

A Kies weerstandswaarden zo laag mogelijk, maar pas op dat de stroom niet groter wordt dan de versterker of deeltrappen kunnen leveren. De tegenkoppeling vormt ook al een belasting voor de versterker. Houd er dus goed rekening mee hoeveel de belasting maximaal nodig heeft, en houd een ruime reserve aan. Te veel stroom zorgt weer voor meer ruis en toename van de vervorming. Voorkom oscillaties. HF-oscillatie veroorzaakt ook veel ruis en kan gelijkstroominstellingen verstören en een hogere vervorming veroorzaken. Let er op dat je, als bij opampschakelingen de versterking éénmaal is, een *unity gain stable* type gebruikt. Er zijn ook opamps die pas stabiel zijn bij een versterking van 5 of 10 maal. Een andere truc die vaak bij phonoversterkers wordt toegepast is transistoren in de ingangstrap parallel te schakelen.

V Hoe houd je de vervorming over een belasting zo klein mogelijk?

A De massa's van ingangssignaal, tegenkoppeling en belasting moeten op hetzelfde punt samenkomen: het sterpunkt. De verbinding naar de voedingsmassa moet van dit punt afgetakt worden. In die drie verbindingen mogen verder geen vertakkingen voorkomen.

V Hoe zorg je ervoor dat gelijkspanningsoffset in een versterker zo klein mogelijk is?

A Voor de in- en uitgang is het toepassen van een ontkoppelcondensator de eenvoudigste oplossing. Maar als zeer lage frequenties of DC doorgegeven moeten worden, is dat niet mogelijk. Voor zeer lage frequenties worden die condensator(en) veel te groot. Een betere oplossing is ervoor te zorgen dat de gelijkstromen aan de ingang van een versterkertrap zo klein mogelijk zijn. Bij bipolaire opamps lopen vaak μA 's. Beter is dan JFET-typen toe te passen waar de ingangsstromen slechts nA's bedragen. Het beste is toepassing van typen met MOSFET-ingangen. JFET- en vooral MOSFET-typen hebben echter weer een grotere (vooral laagfrequente) ingangsruis. Welke van de drie typen wordt toegepast, hangt ook af van de dimensionering en het frequentiegebied van de schakeling (zie de eerste vraag). Kijk altijd goed naar de informatie en specificaties in de datasheet van de fabrikant.

V Hoe beveilig je de ingang van een versterker tegen overspanning, en waar moet je dan op letten?

A Een weerstand in serie met de ingang is een optie, maar dan nemen ruis en storingsgevoeligheid toe. Ook zal de bandbreedte nadelig beïnvloed worden. Het toepassen van

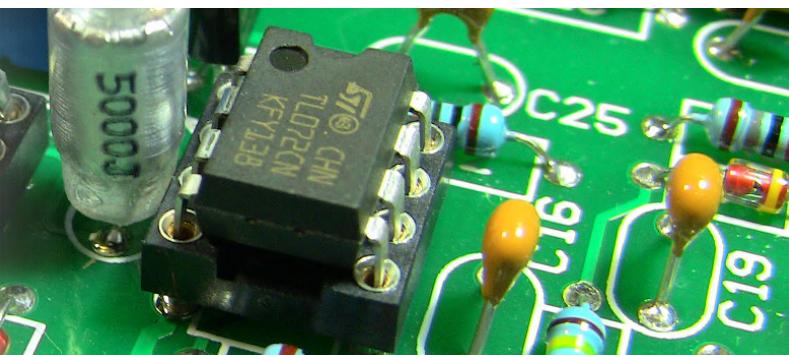
clamping-dioden naar de voedingsspanning is een betere optie. Dan kan een veel kleinere seriere weerstand worden toegepast. Bij extreem hoge ingangsimpedanties, zoals bij een actieve probe, kan de lek van de dioden echter een grote offset veroorzaken. Dan moeten speciale dioden toegepast worden. Alleen bij HF zullen de dioden de bandbreedte beïnvloeden. Let wel op dat als de stroom door de clamping-dioden te groot wordt, de voedingsspanning niet 'opgetild' kan worden. Om dat te verhinderen moet de regelaar voor de voedingsspanning daarvoor ontworpen of beveiligd zijn. Een simpele oplossing is ook twee zenerdioden in antiserie parallel aan de ingang (of een aantal dioden in serie/antiparallel) te zetten. Die kunnen echter op de ingangsimpedantie en lineariteit invloed hebben.

V Waarom is er een analoge massa?

A In een schakeling waar digitale en analoge deelschakelingen worden toegepast, wordt voor het analoge deel vaak een aparte massa gebruikt. Deze is duidelijk gescheiden op de print om storingen door het digitale circuit te voorkomen. Ook is er dan vaak een aparte voedingsspanning aanwezig voor het analoge circuit. Het is dan wel zaak om de voedingsstroomen en de deelschakelingen goed van elkaar te scheiden. Het kan zelfs bij zeer gevoelige schakelingen noodzakelijk zijn om in het analoge deel nog eens te voorzien in een aparte massa voor het ingangsdeel.

V Hoe kun je storing in en door analoge schakelingen voorkomen?

A Een goede ontkoppeling van de voedingsspanning is een goed begin. Denk dan aan ferrietkraaltjes in serie met de voedingsspanning van schakelende circuits. Afhankelijk van de bandbreedte van het stoorsignaal kan het gebruik van SMD-componenten de enige oplossing zijn, waarbij voor hogere frequenties steeds kleinere componenten nodig zijn. Die moeten dan zo dicht mogelijk bij de 'stoorbron' geplaatst worden. Denk ook aan common mode onderdrukking; de meeste storing die dit soort schakelingen produceren is juist common mode. Dat betekent in de praktijk dat de schakeling en vooral aansluitdraden als antenne werken, en dus storing kunnen oppikken. Vandaar de ferrietkernen op netsnoeren en computerkabels en dergelijke. Ook moeten de bandbreedte en slew rate van de schakeling niet groter zijn dan strikt noodzakelijk. Vermijd in de printlayout lussen: de retourstroom moet zo dicht mogelijk bij de heengaande stroom lopen. Is dit niet door een spoor of kopervlak ernaast of eronder mogelijk, dan is het ontwerpen van een vierlaags (of meer) print vrijwel onvermijdelijk. Bij HF is vaak een afscherming nodig. Let dan goed op



of ook hier common mode onderdrukking van in en uitgaande signalen nodig is.

V Wat kan de oorzaak van brom in een versterker zijn?

A Voor een deel is deze vraag hiervoor al beantwoord. We hebben het nu niet over de loszittende afscherming van een kabel die loszit of een onbedoelde verwisseling van signaalader en massa. De mogelijkheid bestaat dat het magneetveld van een voedingstrafo in de buurt van de schakeling de oorzaak is. Vaak wordt daar al rekening mee gehouden door een ringkerntrofo te gebruiken, maar ook die hebben een stoorveld, het verwachte positieve effect valt in de praktijk vaak tegen. Een oplossing kan een afscherming met mumetaal zijn. Een eenvoudiger oplossing is een grote afstand tussen trofo en schakeling, maar dat is natuurlijk vaak onmogelijk. Het voordeel van een E-kern trofo is dat de metalen zijden, waar de gelamineerde kern duidelijk zichtbaar is, een veel kleiner stoorveld hebben dan bij een ringkerntrofo het geval is. Je ziet soms om die reden dit type trofo juist toegepast en dan 45° gedraaid in een hoek van een behuizing staan, zodat de twee ergste stoorvelden buiten de behuizing terecht komen. Een ander effect is de storing die een bruggelijkrichter kan veroorzaken, vandaar dat vaak viermaal 10...47 nF over de aansluitingen worden gezet. Die storing is vooral bij gevoelige schakelingen (zoals microfoonversterkers) een mogelijke oorzaak van brom. Maar dit effect kan ook bij ontvangers en andere HF-schakelingen problemen veroorzaken. Zorg bij een printontwerp ervoor dat de laadstromen van condensatoren niet door of in de buurt van signalmassa's lopen. De stromen door de nul moeten via de kortste weg terug naar de trofo. Maak ook hier de afstand tot de gevoeligste delen van de analoge schakeling zo groot mogelijk. Ook kan oscillatie, in een versterker bijvoorbeeld, de oorzaak zijn. Er kunnen dan veel groter stromen gaan lopen waardoor de voedingsimpedantie aanzienlijk toeneemt.

V Op welke manier kunnen kleine signalen het beste doorgegeven worden?

A De eenvoudigste manier om een signaal door te geven is een signallijn met bijbehorende massa. Dit noemt men asymmetrisch of ongebalanceerd. In het geval van een kabel heeft een coax-type vaak de voorkeur, alhoewel twee getwiste draden ook al kunnen volstaan. Tussen de massa's van twee verschillende apparaten kunnen echter potentiaalverschillen voorkomen die stromen door de massaverbinding veroorzaken. Ook een kabel in een stoorveld kan dit soort storingen veroorzaken. Deze problemen kunnen ook al bij ver-

bindingen tussen deelschakelingen op een print voorkomen. Een betere manier is een symmetrisch signaal te maken, vooral als een grotere afstand overbrugd moet worden. Dit wordt wel een gebalanceerd signaal genoemd. Eigenlijk is dit de enige optie om een groter dynamisch bereik te bereiken (lees: hoe goed de schakeling in staat is heel kleine signalen nog te verwerken). Er worden dan twee gelijke signalen gebruikt die in tegenfase zijn. Een storing op beide signalen wordt dan door aftrekken van beide signalen onderdrukt. Het nuttige signaal is dan twee maal zo groot als een enkel signaal in de gebalanceerde verbinding. Dit is de reden dat de betere A/D-convertisers met hogere resoluties in de regel een symmetrische ingang hebben.

V Wat wordt nog wel eens over het hoofdgezien bij meten aan een schakeling?

A Bij gebruik van een multimeter moet men rekening houden met de inwendige weerstand bij zowel spannings- als stroommeting. In beide gevallen kan een meetfout ontstaan, maar bij gevoelige schakelingen kan door aanraken ook nog instabiliteit of zelfs oscillatie worden veroorzaakt. Die veroorzaakt dan vaak een veel grotere afwijking in de meting. Ook wordt in de schakeling een stoorsignaal geïnjecteerd – iets om rekening mee te houden. Maak bij twijfel daarom gebruik van een oscilloscoop om het signaal te bekijken. De probe van een oscilloscoop is ook niet ideaal en heeft een capaciteit (en weerstand) die weer andere problemen kunnen veroorzaken. Voor zeer gevoelige metingen en/of zeer hoogfrequente signalen kan een actieve probe een uitkomst bieden, die hebben vaak een hogere impedantie en een kleinere capaciteit.

V Welke condensatoren kan men het beste vermijden in audioschakelingen?

A Iets waar velen niet bij stilstaan is dat condensatoren, afhankelijk van de constructie maar meer nog van het toegepaste diëlektricum, verre van ideaal zijn en de kwaliteit en specificaties van een ontwerp behoorlijk kunnen aantasten. De ergste zijn wel keramische condensatoren. Afhankelijk van het materiaal van het diëlektricum is de capaciteit spanning- en temperatuurafhankelijk; ook hebben ze last van microfonie (een mechanisch trilling of belasting kan een spanning opwekken in de condensator). Daarnaast hebben ze last van veroudering. Ook voor elektrolytische condensatoren geldt dat de capaciteit niet stabiel is, en dat vaak om vervorming bij een kantelpunt laag te houden de waarde eenvoudigweg vele malen groter genomen wordt en het kantelpunt daardoor lager komt te liggen. Voor wie het interesseert: zoek eens naar *diëlektrische absorptie*. Een bekend fenomeen is dat na ontladen de spanning over de condensator weer langzaam toeneemt. Beter is het gebruik van kunststoffilm, zoals polyester, polypropyleen of polystyreen. Met een afnemende diëlektrische constante neemt vaak de kwaliteit toe. Nadeel is dat bij de betere typen zoals polystyreen het capaciteitsbereik klein is, vaak maar enkele tientallen nF maximaal. Het meest geschikte type wordt dan ook door de voorgescreven waarde bepaald en/of door de afmeting van de condensator. ▲

190099-01

Digitale modulatietechnieken: coderen van gegevens

NRZ, Manchester en consorten

Robert Lacoste (Frankrijk)

In het voorgaande artikel [1] heb ik u de belangrijkste digitale modulatietechnieken laten zien. Ik schreef in dat artikel dat ‘de data moeten worden gecodeerd in een voor verzending geschikt formaat’. Waar gaat het hier precies om, en waarom is coderen nodig? Als het antwoord hierop u niet direct duidelijk is, of als de afkortingen NRZ, AMI en MLT u niets zeggen, dan wordt het tijd om dit artikel te lezen.

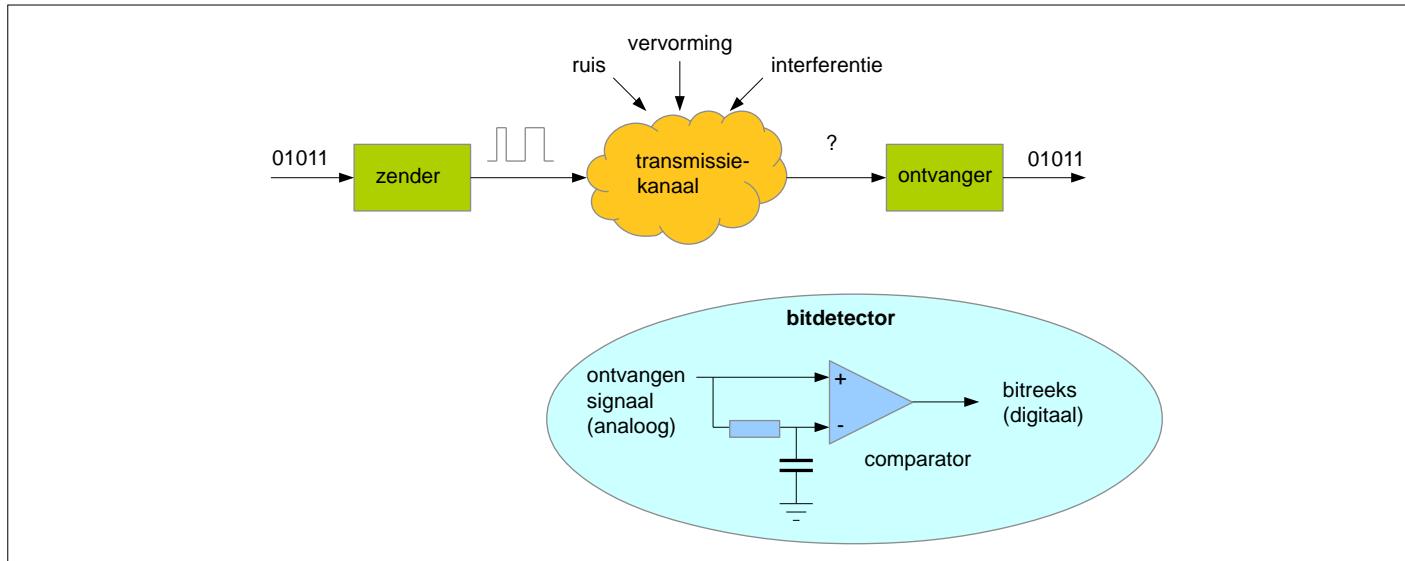
Om het eenvoudig te houden zal ik mijn voorbeelden baseren op transmissie via een kabel. Ook ga ik ervan uit dat bij deze transmissie geen gebruik wordt gemaakt van modulatie (we spreken dan van transmissie ‘in de basisband’). Onthoud dat alles wat ik u hier laat zien ook van toepassing is op draadloze communicatie of communicatie via het elektriciteitsnet. Stel dat u twee elektrische geleiders hebt die twee gebouwen met elkaar verbinden. U wilt deze draden gebruiken om gegevens zo efficiënt mogelijk te verzenden, uiteraard met zo eenvoudig mogelijke elektronica. In het jaar 1840 zou u dan gebruik kunnen maken van het werk van Samuel F.B. Morse en uw data omvormen naar punten en strepen. Morsecode heeft echter het nadeel dat het niet echt snel gaat. Bovendien is de transmissieduur per karakter verschillend. Dat is een voordeel voor een menselijke telegrafist, maar met elektronica lastig te realiseren. De eerste code met een vaste lengte is de Baudot-code die in 1874 door Jean-Maurice Émile Baudot werd uitgevonden. De eenheid ‘Baud’ (één binaire informatie-eenheid per seconde) is afgeleid van zijn achternaam. Deze code betekende voor de automatisering een grote stap vooruit, want ieder gecodeerd karakter had altijd een lengte van vijf bits. In de jaren ’60 maakte Gordon Bell een verbeterde versie voor de PDP-computer (Digital Corporation): de eerste echte UART (*Universal Asynchronous Receiver and Transmitter*) was geboren! Een eenvoudig principe dat nog steeds in alle microcontrollers wordt toegepast: de gegevens worden gecodeerd in bytes (8 bit) en de bits worden één voor één over twee draden verzonden. Er wordt een bepaalde spanning gebruikt om een ‘0’ aan te geven en een andere voor een ‘1’. (respectievelijk 0 V en 5 V bij TTL, +3...+15 V en -3...-15 V bij RS232, enzovoort). Deze (eenvoudigste) transmissiemethode wordt **NRZ** genoemd (*Non Return to Zero*).

Nadelen van NRZ

Het ziet er erg eenvoudig uit, maar niets is eenvoudig in deze wereld en al helemaal niet als het over het verzenden van gegevens gaat. Een UART die NRZ-codering gebruikt werkt prima als de afstand klein is, de snelheid niet al te groot en er niet

teveel storing optreedt, maar is in alle andere gevallen onbruikbaar. Waarom? Omdat het signaal tijdens de transmissie wordt verzwakt, wordt vervormd door onvolkomenheden in het systeem, wordt verstoord door ruis en last heeft van interferentie (**figuur 1**). De ontvanger heeft de taak om de verzonden bits met zo min mogelijk fouten te reproduceren. In een volgend artikel komt foutcorrectie aan de orde, maar op het fysieke niveau moet iedere ontvanger eerst het ontvangen analoge signaal omzetten in een reeks nullen en enen. Vanwege alle verstoringen van het signaal is deze zogenoemde detectiefase niet eenvoudig. Zoals in figuur 1 is te zien, bestaat de detector uit een comparator tussen het ingangssignaal en het gemiddelde van dit signaal in de loop van de tijd. Zo’n schakeling is in feite een hoogdoorlaatfilter. Alle gelijkstroomcomponenten (DC) worden uit het signaal verwijderd, en omdat het signaal wordt vergeleken met zijn gemiddelde vindt de detectie van de bits plaats op basis van snelle variaties rondom dit gemiddelde. Bij draadloze verbindingen gebeurt precies hetzelfde als er gebruik wordt gemaakt van een modulatievorm waarbij de frequentie wordt verschoven (FSK). Omdat de centrale frequentie niet exact bekend is, wordt er gedetecteerd ten opzichte van een gemiddelde frequentie.

Maar wat zijn dan precies de nadelen van NRZ-codering? De makkelijkste manier om deze te begrijpen is door van het tijddomein over te gaan naar het frequentiedomein, oftewel het spectrum van het signaal te analyseren. Geen paniek, het zal u zometeen duidelijk worden. Om u uit te leggen wat er gebeurt heb ik een klein simulatiescript van NRZ-modulatie gemaakt met behulp van Scilab. Dit digitale rekentool is gratis en de code van mijn voorbeeld is te vinden op [2]. Aarzel niet om hier zelf mee te experimenteren! Kort samengevat heb ik 4000 willekeurige bits met NRZ gecodeerd (met -1 en +1 voor respectievelijk de nullen en de enen). Vervolgens heb ik het frequentiespectrum van dit signaal berekend met wat een snelle Fouriertransformatie (FFT) wordt genoemd. Het resultaat ziet u in **figuur 2**. Wat zien we in deze simulatie? In de eerste plaats is het spectrum niet nul aan de linkerkant van de grafiek, bij 0 Hz. Dit betekent dat het NRZ-signaal compo-



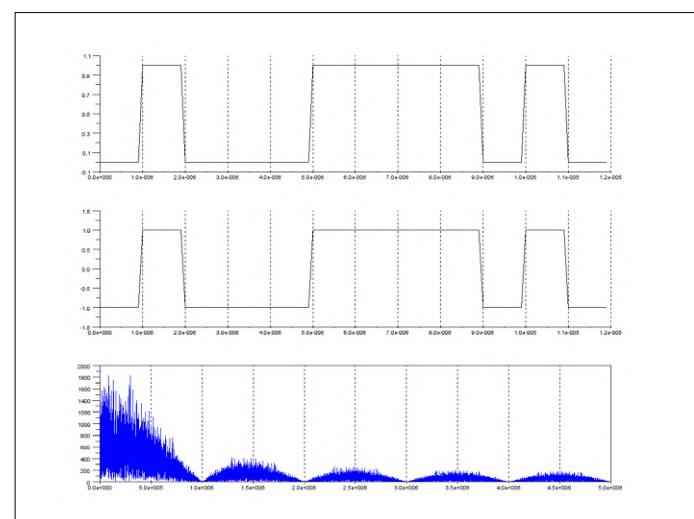
Figuur 1. Iedere ontvanger bevat een detector die moet bepalen of het ontvangen bit een '1' of een '0' is. Meestal wordt deze stap gerealiseerd door de ontvangen spanning te vergelijken met een gemiddelde spanning. Dit betekent dat een ongebalanceerd signaal problemen kan veroorzaken.

nenten bevat met zeer lage frequenties, inclusief DC, zelfs als de binaire reeks erg lang is en de bits evenveel kans hebben om nul of één te zijn. Waarom? Eenvoudig omdat er in iedere willekeurige reeks lange reeksen nullen of enen kunnen voorkomen... Deze lange reeksen vertalen zich in het frequentiedomein naar een vermogensspectrum dat reikt tot aan DC. Het probleem is dat deze lage frequenties de detector in de war brengen. U herinnert zich dat deze de signaalspanning vergelijkt met het 'lange-termijn'-gemiddelde. Bij lage frequenties zal dit gemiddelde niet erg stabiel zijn, waardoor er schommelingen in de gedetecteerde binaire datastrook ontstaan, en uiteindelijk bitfouten als er ruis op de lijn zit. Om een extreem voorbeeld te nemen: stel dat u een zeer lange reeks 'enen' verzendt. Hoe kunt u er zeker van zijn dat een bit een '0' of een '1' is als het gelijkstroomniveau op de lijn is verschoven? We kunnen dit probleem ook op een andere manier bekijken: de gemiddelde waarde van het signaal zal alleen dan precies het gemiddelde van de twee spanningen zijn die voor de 'enen' en de 'nullen' worden gebruikt als er exact evenveel enen en nullen in de verzonden bitreeks voorkomen. De kans dat dit gebeurt is helaas erg klein. Als u nog niet bent overtuigd, gooi dan honderd keer een geldstuk op en probeer exact 50 keer kop en 50 keer munt te krijgen... Daarom kan een NRZ-decoder zich 'vergissen' als er ruis op de lijn zit, en zal dit gemakkelijker gebeuren als de verzonden bitstroom niet in evenwicht is met betrekking tot de aantallen 'nullen' en 'enen'.

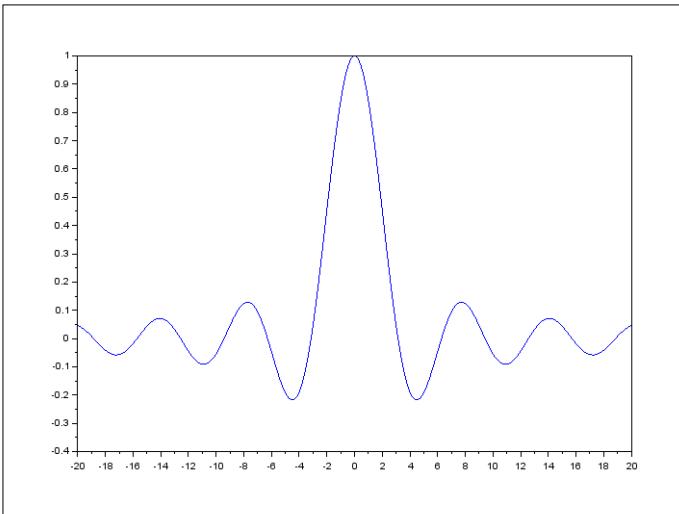
Verderop zal ik u een aantal alternatieve coderingsmethoden laten zien, maar laten we eerst nog even naar het frequentiespectrum van het NRZ-signaal kijken (figuur 2). Herinnert u zich deze vorm nog? Als u een regelmatige lezer van deze rubriek bent, zou dit het geval moeten zijn. Het betreft de functie *sinus cardinalis*, of $\sin(x)/x$, die we al verschillende malen zijn tegengekomen, en wel om een erg goede reden: we vinden de functie $\sin(x)/x$ terug in het spectrum van ieder digitaal signaal met bits van een vaste lengte, omdat deze functie de Fouriertransformatie van een rechthoekige impuls is (**figuur 3**). Nog een laatste opmerking over figuur 2: het frequentiespectrum in de vorm van $\sin(x)/x$ blijft erg hoog in het rechter

gedeelte van de grafiek, dus in het hoge frequentiegebied. Dit betekent dat de gebruikte kabel van goede kwaliteit moet zijn, want deze moet een grote doorlaatband hebben om de snelheid van de verbinding niet te beperken. Het betekent ook dat er EMC-problemen kunnen optreden.

Samenvattend zult u hebben begrepen dat een codering zoals NRZ erg eenvoudig kan worden gerealiseerd, bijvoorbeeld met een simpele UART. Het nadeel is dat het spectrum sterke componenten in het lage frequentiegebied bevat, hetgeen betekent dat de transmissie gevoelig zal zijn voor verstoringen, met name als er lange reeksen nullen en enen in het signaal voorkomen. Het spectrum strekt zich ook uit naar de hoge frequenties waardoor de gebruikte bandbreedte tamelijk groot zal zijn. Bestaan er slimmere coderingsmethoden dan NRZ, die het gebruikte frequentiespectrum zowel in de lage als in de hoge frequenties reduceren zonder dat dit de snelheid nadrukkelijk beïnvloedt? Jazeker, en deze kunnen zorgen voor een verhoging



Figuur 2. De eenvoudigste coderingsmethode is NRZ, die bestaat uit... niets doen. Helaas bevat het frequentiespectrum van zo'n signaal (onderste grafiek) zeer veel componenten van DC tot zeer hoge frequenties.

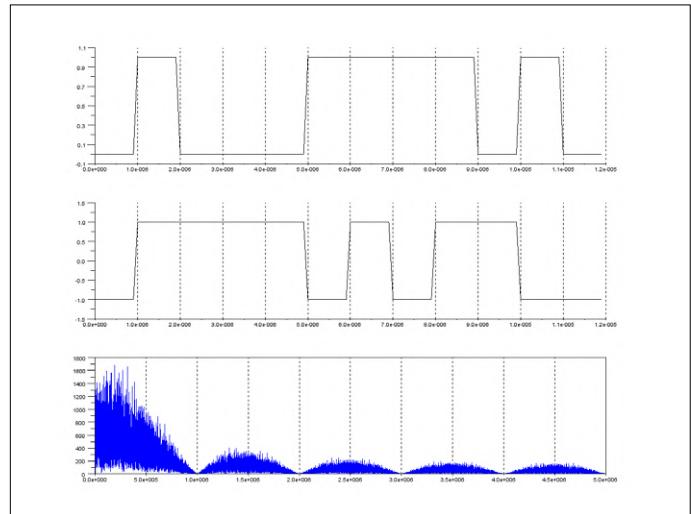


Figuur 3. De vorm van het spectrum van een NRZ-signaal wordt bepaald door de functie $\sin(x)/x$, omdat deze functie de Fouriertransformatie is van een rechthoekige puls. Vergelijk deze curve met die van figuur 2.

van de bitrate, het vergroten van de afstand, of het gebruik van een goedkopere kabel.

Andere methoden

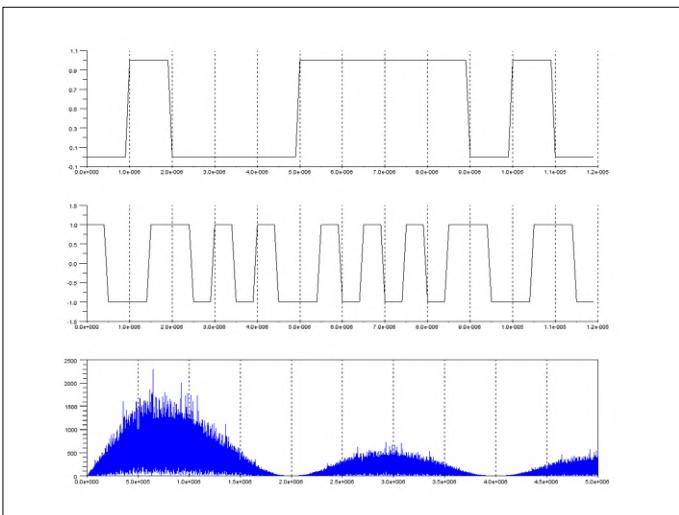
Laten we eens kijken naar betere oplossingen. Het eerste alternatief wordt NRZI genoemd (**figuur 4**). De I betekent ‘geïnverteerd’. Het is bijna dezelfde methode als NRZ, maar met een subtiel verschil. Als er een ‘1’ wordt verzonden verandert het signaalniveau, en bij een ‘0’ blijft het constant. Zoals in figuur 4 is te zien, is het spectrum van een NRZI-signaal vrijwel gelijk aan dat van een ‘standaard’ NRZ-signaal. Er is dus geen echt voordeel wat betreft het spectrumgebruik. Niettemin biedt de NRZI-codering twee voordelen. In de eerste plaats is NRZI een differentiële code, dat wil zeggen dat deze niet afhangt van de polariteit van het signaal: als u de polariteit aan de ingang omwisselt zullen de bits nog steeds goed worden gedecodeerd. Alleen het wel of niet aanwezig zijn van een



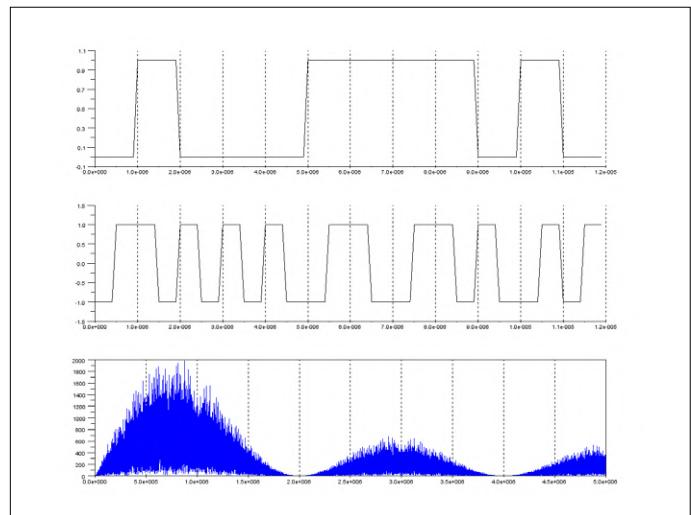
Figuur 4. Bij NRZI verandert het verzonden signaal (middelste grafiek) van niveau als het ingangsbit een ‘1’ is, en blijft anders constant. Het spectrum is hetzelfde als dat van NRZ.

niveau-overgang is van belang, niet de polariteit. Het tweede voordeel is dat lange reeksen ‘enen’ worden vermeden en dat maakt het leven van de ontvanger veel makkelijker. Het probleem van de lange reeksen ‘nullen’ blijft echter bestaan en daar zijn dus andere slimme oplossingen voor nodig. Zo wordt er bij USB een variant van NRZI gebruikt waarbij na zes op elkaar volgende ‘enen’ een extra ‘nul’ wordt ingevoegd waarmee het probleem van lange reeksen zowel voor de nullen als voor de enen wordt opgelost.

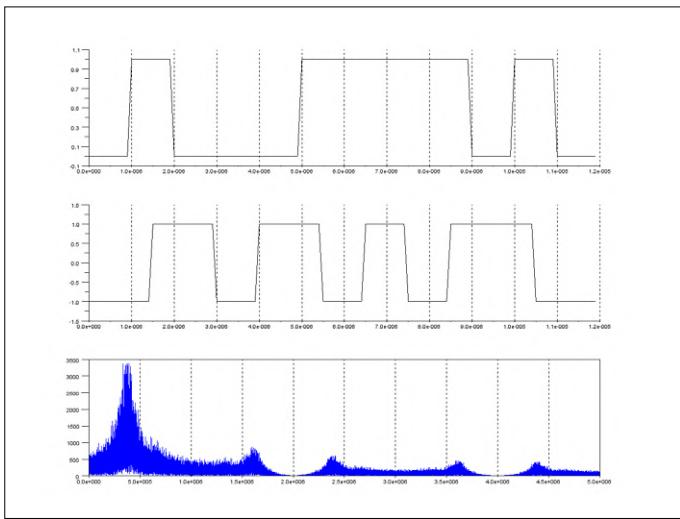
Een andere veelgebruikte methode om het evenwicht tussen de nullen en de enen te bewaren is de **Manchestercode** (**figuur 5**). Het principe is het volgende: ieder bit wordt gecodeerd in de vorm van twee opeenvolgende bits. Een ‘1’ wordt gecodeerd als ‘01’ en een ‘0’ als ‘10’. Dit verdubbelt het aantal te verzenden bits maar garandeert dat er in het verzonden signaal precies evenveel veel nullen als enen voorkomen. Bovendien zit er minstens één niveauverandering in het midden van ieder verzonden



Figuur 5. Dit is een voorbeeld van de Manchestercode. Ieder bit wordt in twee opeenvolgende bits ‘01’ of ‘10’ gecodeerd, afhankelijk van het te verzenden bit. Het spectrum bevat geen laagfrequente componenten, maar er worden meer hoge frequenties gebruikt omdat de bitrate is verdubbeld.



Figuur 6. Voorbeeld van de differentiële Manchestercode die ten opzichte van de Manchestercode het voordeel heeft dat de polariteit van het signaal niet van belang is.



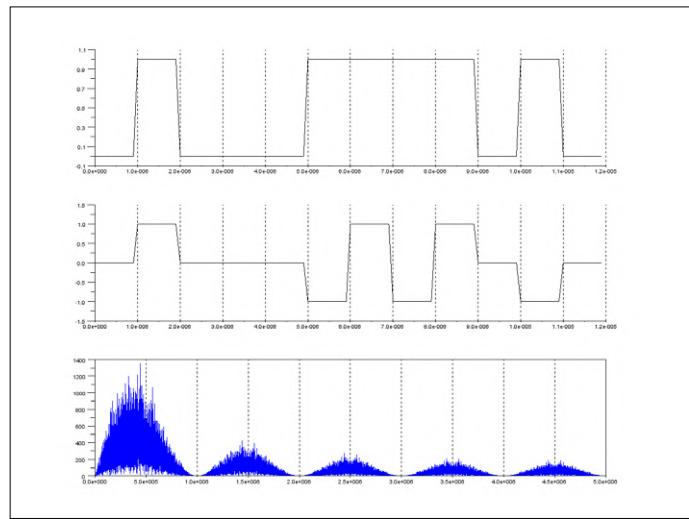
Figuur 7. De Millercode is wat het spectrum betreft veel efficiënter.

bit waardoor het terugwinnen van de klok wordt vereenvoudigd. Kijk nog eens naar figuur 5: het frequentiespectrum is duidelijk verbeterd. Bij de Manchestercode zijn er geen laagfrequente of DC-componenten meer. De Manchestercode wordt zowel voor draadloze verbindingen (van het type afstandsbediening) als voor bedrade verbindingen gebruikt. Deze code wordt bijvoorbeeld gebruikt bij Ethernet in de variant 10BT (10 Mbps).

Er bestaat ook een variant van de Manchestercode: de differentiële Manchestercode die ook wel ‘Biphase Mark Code’ (**BMC**) wordt genoemd. Dit is een kruising van NRZI en Manchester. Ieder bit wordt gecodeerd als een opeenvolging van twee logische niveaus zoals bij Manchester, zodat er twee mogelijke transities per bit zijn. Een van deze transities, bijvoorbeeld in het midden van het bit, wordt gebruikt als referentie voor de klok en is altijd aanwezig. De tweede transitie, aan het begin van het bit, is niet aanwezig als het te verzenden bit een nul is (**figuur 6**). Het spectrum is hetzelfde als dat van Manchester, maar net als bij NRZI is de transmissie onafhankelijk van de polariteit van het signaal. Bovendien vergemakkelijkt de transitie in het midden van ieder bit de synchronisatie van de ontvanger. De differentiële Manchestercode wordt bijvoorbeeld toegepast bij S/PDIF-verbindingen.

Betere prestaties?

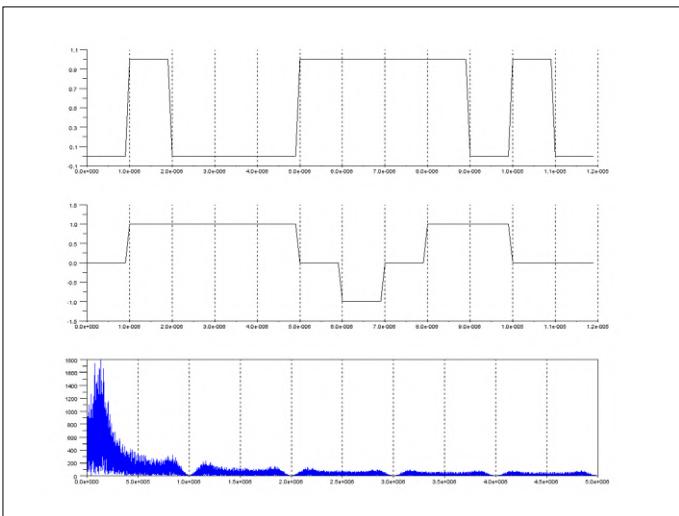
Gaan we verder. Het zou een grote verbetering zijn wanneer we dezelfde hoeveelheid data konden verzenden via een veel smaller kanaal, dus met een kleiner spectrum. Een van de mogelijkheden om dit te bereiken is de **Millercode**. Deze is eenvoudig te realiseren: ga uit van de Manchestercode en onderdruk één van elke twee niveauveranderingen. U krijgt dan een signaal zoals in **figuur 7** is weergegeven. Als u dit nauwkeurig bekijkt zult u zien dat er nog steeds in het midden van ieder ‘1’-bit een transitie zit, maar nooit in het midden van een ‘0’-bit. Er is ook een transitie aan het einde van een ‘0’-bit, maar alleen als dit bit wordt gevolgd door een andere ‘0’. Het spectrum is opnieuw verbeterd (vergelijk de figuren 5 en 7 eens met elkaar...). Een variant van de Millercode wordt bijvoorbeeld gebruikt voor RFID-systeem van het type EPC GEN2. Er is nog een andere familie van oplossingen. De ‘Alternate Mark Inversion’-codes (**AMI**) werden tot voor kort veel toegepast



Figuur 8. De AMI-code is een ternaire code, waarbij drie verschillende spanningen worden gebruikt. De nullen worden verzonden met een spanning van 0 V, en de enen met afwisselend positieve en negatieve pulsen. Het spectrum lijkt op dat van de Manchestercode, maar is twee keer zo compact.

bij telecommunicatie (T1-verbindingen en wat dies meer zij). Hierbij worden geen twee maar drie verschillende spanningen op de lijn gebruikt: een positieve spanning, nul en een negatieve spanning. Het idee is het volgende: een ‘0’ wordt altijd voorgesteld door de spanning nul, en een ‘1’ wordt afwisselend voorgesteld door een positieve of een negatieve spanningspuls (**figuur 8**). Dit levert een gemiddelde spanning nul op, evenals een fraai frequentiespectrum. Het grote nadeel is echter dat het onmogelijk is om de ontvanger te synchroniseren als het signaal alleen maar nullen bevat. Dit was geen probleem bij de transmissie van spraaksignalen, want die data bevatten altijd ruis, maar bij digitale signalen is dit wel een probleem. Een nog efficiëntere variant is de **MLT-3**-code (**figuur 9**). MLT betekent ‘Multi-Level Transmit’, want er zijn net als bij AMI drie spanningen op de lijn mogelijk (positief, nul, negatief). Het principe is echter verschillend: als het te verzenden bit een ‘0’ is, blijft de toestand van de lijn ongewijzigd. Als het bit een ‘1’ is springt het lijnniveau naar de volgende toestand in een zich herhalende reeks: +V, 0, -V, 0, +V, enzovoort. Het uitgangssignaal is dus min of meer sinusvormig en dit levert een spectrum op waarbij de nadruk op de lage frequenties ligt, zoals u in figuur 9 kunt zien. U hebt ongetwijfeld zonder het te weten al eens gebruik gemaakt van deze MLT-3-code, want hierop is de Ethernet-variant 100BT gebaseerd.

Ten slotte is er nog een andere manier om het spectrumgebruik van een signaal te optimaliseren die vaak als aanvulling op de voorgaande wordt gebruikt. Hierbij wordt niet alleen met bits gewerkt, maar ook met symbolen die groepjes van bits voorstellen. Zo kunnen bijvoorbeeld de te verzenden bits per vier stuks worden gegroepeerd, waarna ieder symbool van 4 bits volgens een voorgedefinieerde tabel wordt geconverteerd naar een reeks van vijf bits. Deze conversie met de naam **4B5B** (**figuur 10**) zorgt er door een slimme keuze van de tabel voor dat er altijd transities in het signaal voorkomen, ook als de ingangsbits altijd ‘0’ (of ‘1’) zijn. Bovendien kan het verzonden spectrum door een juiste keuze van de zestien woorden van 5 bits (één voor iedere combinatie van vier ingangsbits) worden geoptimaliseerd. Bij Ethernet 100BT wordt bijvoorbeeld vóór de MLT-3-codering een 4B5B-codering toegepast. Varianten met lange woorden zoals

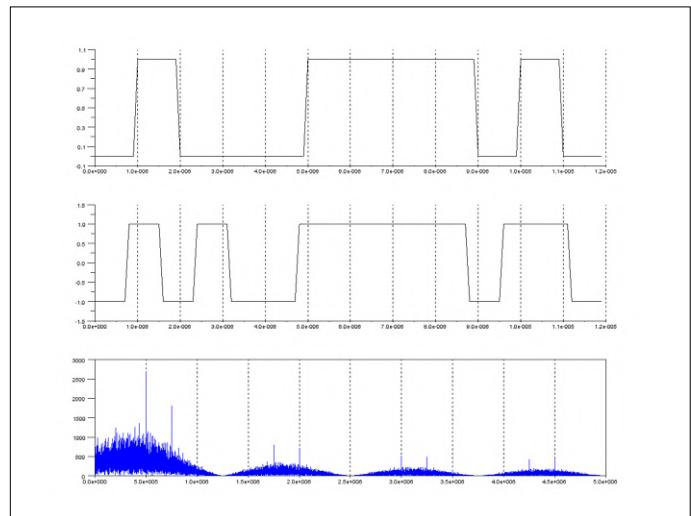


Figuur 9. De MLT-3-code, bijvoorbeeld gebruikt voor Ethernet 100 Mbps, maakt het met een intelligente codering op drie niveaus mogelijk om het spectrum aanzienlijk te verkleinen.

8B10B, die door IBM al in het begin van de jaren 1970 werden geïntroduceerd, presteren natuurlijk nog beter. 8B10B wordt tegenwoordig overal toegepast: van diskettes en de antieke Apple II tot de PCI-express-bus, en van Firewire tot SATA, DVI of HDMI, maar ook voor Gigabit-Ethernet en USB 3.0. Uiteraard kan dit nog verder worden doorgevoerd: bij Ethernet op 10 Gbps wordt gebruik gemaakt van een 64B66B-codering.

Tot slot

Zoals u hebt gezien zijn er talloze manieren om een digitaal signaal te verzenden. De eenvoudigste oplossingen, zoals de NRZ-codering, zijn bruikbaar voor korte verbindingen en lage snelheden. Maar het is altijd interessant om alternatieve, efficiëntere oplossingen te kennen... Nogmaals, alles wat ik u heb laten zien, is ook van toepassing op gemoduleerde signalen, in het bijzonder voor draadloze verbindingen. Als voorbeeld herinner ik me nog dat ik erg teleurgesteld was toen ik mijn eerste 'draadloze' project probeerde te realiseren. Ik had een



Figuur 10. Coderingsmethoden met symbolen, zoals de hier afgebeelde 4B5B-methode, worden in het algemeen als aanvulling op andere methoden gebruikt om hun efficiëntie te vergroten. Vier opeenvolgende bits worden volgens een tabel in vijf bits gecodeerd.

paar kleine zend- en ontvangstmodules gekocht en ik probeerde nullen en enen te verzenden met een microcontroller. Uiteraard gebruikte ik de standaard UART, en dus NRZ. Ik kreeg het min of meer aan de praat, maar met een belachelijk klein bereik van maar enkele meters. Na het lezen van de documentatie van de modules begreep ik dat deze alleen bitreeksen ondersteunden die een perfect gebalanceerd aantal enen en nullen bevatten. Met wat software voor de Manchestercode kon ik het bereik met een factor 20 vergroten!

In het volgende artikel ga ik op dezelfde manier verder met een besprekking van een aanvullende techniek: het filteren van het kanaal. Aarzel niet om in de tussentijd de kleine Scilab-simulaties bij dit artikel te downloaden en ermee te experimenteren. Dat is de beste manier om te begrijpen hoe het precies werkt. ◀

(180370-02)

Dit artikel is eerder gepubliceerd in het tijdschrift Circuit Cellar (nr. 255, december 2011).

Weblinks en literatuur:

- [1] Digitale modulatie, deel 1: www.elektormagazine.nl/170009
- [2] Webpagina bij dit artikel: www.elektormagazine.nl/180370-02
- [3] Emile Baudot: http://en.wikipedia.org/wiki/%C3%89mile_Baudot
- [4] 'Universal Asynchronous Receiver/Transmitter': http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter
- [5] 'Oral history of Gordon Bell', Computer history museum: www.computerhistory.org/collections/accession/102702036
- [6] 'CIS 677: Introduction to computer networking', hoofdstuk 4.2 'Digital data, digital signal', Eitan Gurari, Ohio state university: www.cse.ohio-state.edu/~gurari/course/cis677/cis677Se12.html
- [7] 'Wireless Digital Communications: Design and Theory', Tom McDermott, N5EG, Tucson Amateur Packet Radio Corporation, ISBN 0-9644707-2-1
- [8] 'Telecommunications technologies – Line coding', Maciej Ogorzałek, PolyU, EIE: www.eie.polyu.edu.hk/~maciej/pdf/week5.pdf
- [9] Delay codes: http://en.wikipedia.org/wiki/Delay_encoding
- [10] 8B10 encoding: http://en.wikipedia.org/wiki/8b/10b_encoding
- [11] Scilab: www.scilab.org

Felle knipper-LED

ouderwets analoog...

Ir. A.L. (Bram) Melse (Nederland)

Een knipper-LED? wat is daar nou zo bijzonder aan, vraagt u zich wellicht af. Een microcontroller plus een handjevol discrete onderdelen, en klaar is Kees – of niet soms? In overeenstemming met de ‘rode draad’ door dit nummer (analoge elektronica) presenteren we hier een geheel discreet opgebouwd exemplaar dat tot eigen experimenten uitnodigt!

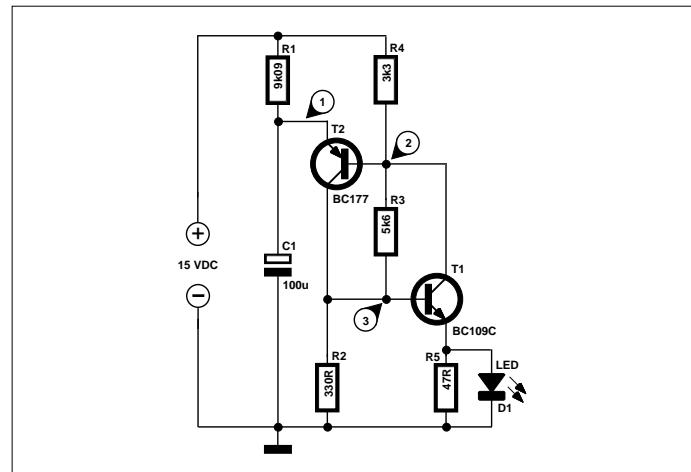
De auteur heeft zich beroepsmatig beziggehouden met het ontwerpen van (vooral) analoge chips. Maar om te bewijzen dat het niet altijd per se een IC hoeft te zijn, en natuurlijk ook omdat het gewoon leuk is een analoge schakeling uit te puzzelen, presenteert hij hier een knipper-LED met niet meer dan twee transistoren en een handjevol klein grut. Het resultaat is getekend in **figuur 1**.

De werking van de schakeling kan het beste als volgt worden beschreven. We gaan gemakshalve uit van de situatie waar de spanning op knooppunt 1 relatief laag is. Via R3 en R4 loopt er een stroom naar punt 3 van de schakeling (en dus naar R2), maar die stroom is te klein om de ongeveer 0,6 V te leveren die nodig is om T1 open te sturen.

Via weerstand R1 wordt condensator C1 steeds verder geladen; zodra R1 ook (via T2) stroom in knooppunt 3 gaat leveren, wordt de som van de stromen groot genoeg om T1 open te sturen. De spanning op punt 2 wordt dan omlaag getrokken waardoor T2 lading uit C1 naar knooppunt 3 doorsluist en de schakeling zich vergrendelt (latch)..

De stroom in knooppunt 3 via weerstand R3 stopt nu, maar daarvoor in de plaats ontladt condensator C1 zich (via T2) en de LED licht fel op. Naarmate C1 zich ontladt, neemt de stroom via T2 in knooppunt 3 af tot niet meer dan wat via R1 wordt geleverd, en dat is niet genoeg om T1 opengestuurd te houden. Punt 2 wordt dan door R4 hoog getrokken waardoor T2 gaat sperren. Op dat moment begint de hele cyclus opnieuw. Weerstand R5 heeft als bypass een belangrijke functie: zonder deze weerstand zou de LED een (te) grote drempelspanning voor het opensturen van T1 veroorzaken, waardoor de latch niet in actie zou kunnen komen. De bypass-stroom bedraagt ongeveer 10% van de LED-stroom.

Merk op dat de LED zeer helder opflitst (bij de gegeven dimensionering met een frequentie van ongeveer 0,8 Hz); een simulatie (in LTspice XVII) gaf een waarde van ongeveer 600 mA. Natuurlijk is het niet ideaal wanneer de condensator zich uitsluitend via de transistoren en de LED ontladt; men kan hier met



Figuur 1. Het schema telt slechts een paar onderdelen. De transistoren zijn niet kritisch: zo ongeveer elke TUP en TUN zullen voldoen.

een stroombegrenzende weerstand experimenteren (waardoor overigens de flits ook wat langer zal duren).

Audio-variant

Wie daar zin in heeft, kan ook een audio-variant van deze schakeling opbouwen. De schakeling blijft daarbij vrijwel hetzelfde; alleen moeten de onderdelenwaarden als volgt worden gewijzigd: **Weerstanden:** R1 = 3k9; R2 = 180 Ω; R3 = 2k7; R4 = 1k8; R5 = 15 Ω.

Condensator: C1 = 0,2 μF.

Transistoren: T1 = BC109C; T2 = BC240B (niet kritisch, elke TUP en TUN is bruikbaar; de auteur had deze toevallig nog liggen).

De luidspreker (8 Ω) wordt tussen C1 en massa aangesloten; de LED wordt weggelaten.

Deze variant produceert een (tamelijk onaangenaam) geluid met een frequentie van ongeveer 1 kHz. Voor alarmdoeleinden kan dat echter goed voldoen.

De opbouw van de schakeling is niet kritisch; een stukje gaatjesprint is meer dan voldoende.

(180542-1)



IN DE STORE

→ Boek: **Basiscursus Elektronica**
www.elektor.nl/basiscursus-elektronica

→ Boek (Engelstalig): **Electronic circuits for all**
www.elektor.nl/electronic-circuits-for-all

→ Boek: **Elektronica kunst en kunde deel 1 (analoge elektronica)**
www.elektor.nl/elektronica-kunst-en-kunde-1



Project 2.0

correcties, updates en brieven van lezers



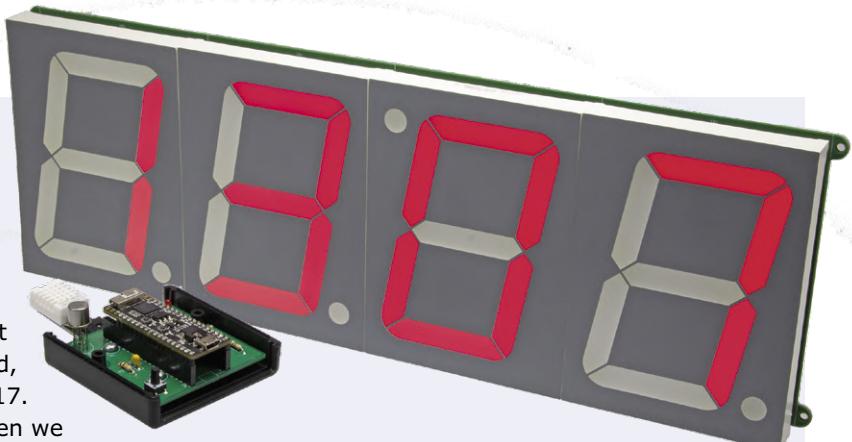
Gigantische LED-klok met WLAN en temperatuurweergave

Elektorlabs mei/juni 2019, p. 64 (180254)

Ik heb het artikel met veel plezier gelezen. Vooral de koppeling van de sensorprint met de klok via MQTT vind ik klasse. Op deze manier kan het project ook aan andere projecten worden gekoppeld, bijvoorbeeld met de IoT-gateway uit maart/april 2017.

En wanneer we dan toch een Rpi gebruiken, kunnen we meteen ook FHEM installeren en de sensordata daar ook gebruiken. De IoT-gateway zou wellicht tot een zuinige sensor voor batterijvoeding kunnen worden omgebouwd. Op de gigantische LED-klok zouden dan eventueel nog andere data kunnen worden getoond, bijvoorbeeld of er post is. Of we ontwikkelen een dotmatrix-display waarop ook teksten kunnen worden weergegeven. De klok kan op die manier de basis vormen voor een hele reeks projecten – jullie kunnen vast en zeker nog veel meer verzinnen.

Frank Klee



Dank voor uw brief. Inderdaad staan voor de klok nog enkele modificaties op het programma en willen we ook de data-uitwisseling via MQTT in meerdere producten inbouwen. We denken aan de koppeling van de klok en het weerstation, plus een modificatie om de klok ook te laten aangeven wanneer er aan de deur gebeld wordt. Ook denken we erover na de ongebruikte I/O-pinnen te gebruiken om de klok niet alleen de tijd te laten weergeven maar ook zelf data te laten produceren – bijvoorbeeld door een lichtsensor of een DHT22 voor temperatuur en luchtvochtigheid aan te sluiten. Dat zijn echter allemaal updates die voor het volgende nummer van Elektorlabs nog niet gereed zullen zijn.

Mathias Claußen (Elektor Labs)



9-kanaals relaiskaart

Elektorlabs mei/juni 2019, p. 80 (190027)

Groeten uit Oostenrijk! Komt er van deze kaart een compleet bouwpakket? Want dat is precies wat ik nodig heb – echter met een klein voorbehoud: de relais zijn geschikt voor 1200 W AC, terwijl de toelaatbare spanning niet is vermeld. Het ligt voor de hand dat dit 230 V is, maar daarvoor lijkt me de kruipafstand tussen de middencontacten van de relais en de logica aan de kleine kant.

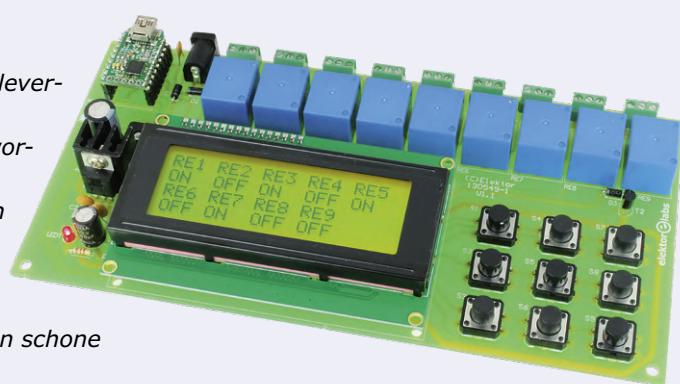
Manfred Tischler

Inderdaad: binnenkort zal dit project als onderdelenpakket leverbaar zijn (we wachten op onze leverancier).

De print is geschikt voor 230 VAC; de aansluitingen zijn ontworpen voor 250 VAC, 7 A.

De aanbevolen kruipafstand is afhankelijk van het gebruik en de normen waaraan men zich wil houden. In een schone en droge omgeving kan die kruipafstand heel klein blijven; in een vuile en vochtige omgeving moet de kruipafstand daarentegen groot zijn. Deze relaiskaart kan probleemloos in een schone en droge omgeving worden gebruikt.

Clemens Valens (Elektor Labs)





Dual-anode MOSFET-thyristor

Elektorlabs mei/juni 2019, p. 15 (190017)

Ten onrechte zijn bij dit artikel alleen Michael Shustov en Ton Giesberts als auteurs vermeld; als co-auteur heeft echter ook Andrey Shustov aan het artikel meegewerktd.



Luchtvervuilingsmonitor

Elektorlabs maart/april 2019, p. 90 (170182)

Voor dit project zou ik een verbetering willen voorstellen. Zoals in het schema is getekend (maar in de tekst niet nader is vermeld) kunnen met de bijbehorende sketch de LED's niet gelijktijdig met het OLED-display worden gebruikt – het is of-of. Maar dat is nergens voor nodig omdat de ESP32 Pico Kit over meer dan genoeg ongebruikte aansluitingen beschikt om zowel de LED's als het display simultaan aan te sturen. Ik heb deze monitor opgebouwd en een paar kleine wijzigingen doorvoerd, zodat nu LED's en display parallel gebruikt kunnen worden.

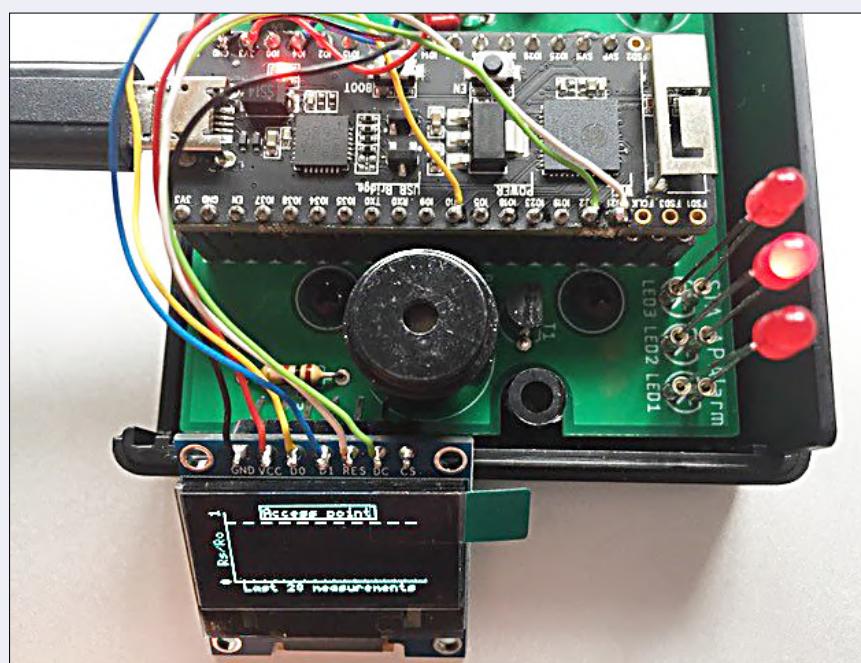
Daartoe moeten als eerste een paar aansluitingen van het display met andere uitgangen van de ESP32 worden verbonden, omdat de LED's op de print vast bedraad zijn en dat niet veranderd kan worden. De DAT-pin van het display gaat in de nieuwe opzet naar IO4 van de ESP32, DC gaat naar IO22 en RST naar IO21. De sketch moet natuurlijk worden aangepast:

```
//pin definitions
#define gasSensorPin 35
#define btn 9
#define alarmPin 19
#define dispClk 10
#define dispDat 4
#define dispD_C 22
#define dispRst 21
#define STALed 5
#define APLed 18
#define alarmLed 23
```

Als dit gedaan is, kan men het beste de complete sketch nalopen en kijken waar `STALed`, `APLed` en `alarmLed` voorkomen. Daar waar corresponderende commando's in `#ifndef USEDISPLAY ... #endif` resp. in de `#else`-tak van `#ifdef USEDISPLAY ... #endif` zijn ingesloten, worden deze daaruit bevrijd. Dit komt een paar maal voor in `tgs2600.ino` en `network.ino`. Wanneer deze veranderingen zijn doorvoerd, tonen nu ook bij gelijktijdig gebruik van een display de LED's de correcte toestand van de netwerkverbinding en het alarm. De vereiste verbindingen kunnen gewoon aan de ESP32 worden gesoldeerd.

Hans Schneider

Onze dank voor deze prima verbetering!
Ralf Schmiedel (redactie)



SHOP-HIGHLIGHT

De inrichting van uw hobbylab...

...hoeft dankzij ons niet veel te kosten!



Wie de eerste schreden op het elektronica pad heeft gezet, krijgt behoefte aan test- en meetapparatuur. En dat hoeft (om te beginnen) echt niet dat heel dure semiprofessionele 'spul' te zijn.



OWON XDS3064E vierkanaals 'scoop met touchscreen

De XDS-serie oscilloscopen van het Chinese merk OWON heeft een aantal eigenschappen die men bij vergelijkbare modellen van andere merken (nog) niet aantreft, en dat alles tegen zeer scherpe prijzen. Hier werpen we een korte blik op de uiterst betaalbare 4-kanaals XDS3064E, die voor nog geen € 500 heel wat te bieden heeft.

Bij het uitpakken van deze oscilloscoop valt meteen het grote 8-inch LC-display op, met een resolutie van 800x600 pixels plus touchscreen-functionaliteit. In deze prijsklasse komt dat niet vaak voor. Na het inschakelen wordt deze eerste goede indruk bevestigd: een mooi scherm met een goede kijkhoek. Als gezegd betreft het een 4-kanaals 'scoop met een ingangsbreedte van 60 MHz en een sample-rate van 1 GS/s. Weliswaar daalt die sample-rate bij gebruik van meerdere kanalen tot de helft of een kwart, maar dat is gebruikelijk in dit prijssegment. Het apparaat wordt geleverd met vier probes, een netsnoer, een USB-kabel en een korte start-handleiding. Een uitgebreide handleiding is als PDF beschikbaar.

De bediening van de XDS3064E went vrij snel. Enkele functies vragen wel wat toelichting, daar is het verstandig om de uitgebreide handleiding te bestuderen. De gevoeligheid van de ingangskanalen is instelbaar tot maar liefst 1 mV/div (bandbreedte-begrensd). De ingangsruis van de ingangstrappen blijft

daarbij vrij gering, anders zou zo'n grote gevoeligheid ook geen zin hebben. Met behulp van de Autoset-knop worden automatisch de juiste instellingen gezocht voor een duidelijke weergave van een ingangssignaal. Via de Math-knop kunnen niet alleen bewerkingen op de signalen worden uitgevoerd, maar het is ook mogelijk om zelf vergelijkingen in te voeren. Ook bezit de XDS3064E een FFT-functie.

Concluderend kunnen we stellen dat wie genoeg heeft aan een ingangsbreedte van 60 MHz, met dit apparaat heel wat meetplezier in huis krijgt.

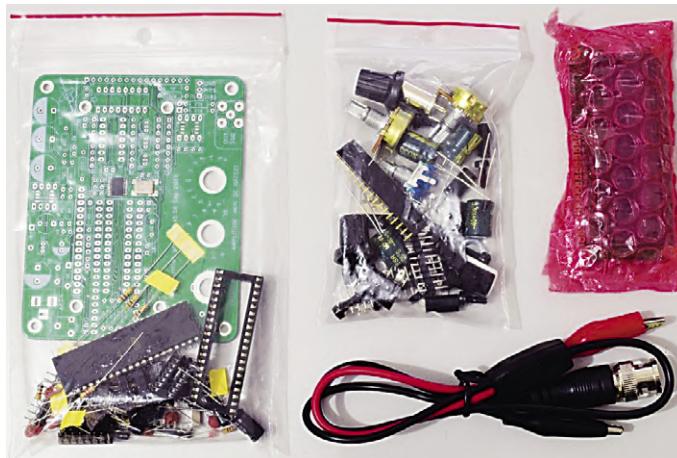
Functiegenerator-bouwpakket van Pulsar Labs

Een oscilloscoop (zoals het hiervoor besproken OWON-model) is onmisbaar om te zien wat een elektronische schakeling doet, maar zo'n instrument komt pas echt tot zijn recht wanneer we een goed gedefinieerd (dus een bekend) signaal op de ingang van een schakeling kunnen zetten. En daar doet een functiegenerator zijn intrede. Hier bekijken we een functiegenerator die we eerst zelf in elkaar moeten zetten: de open-source functiegenerator van Pulsar Labs.

Het opbouwen van de generator is (hoewel dat niet expliciet in de meegeleverde manual wordt beschreven) ook voor de minder ervaren soldeerartiest goed te doen. Het belangrijkste is dat we er de tijd voor nemen, goed kijken waar en hoe

IN DE STORE

- OWON XDS3064E vierkanaals oscilloscoop
www.elektor.nl/18829



de onderdelen moeten worden gemonteerd en bij twijfel eerst even passen en zo nodig de pootjes op maatbuigen. De enige twee SMD-onderdelen zijn al gemonteerd, zodat niemand zich daar zorgen over hoeft te maken. Als de print is volgebouwd en gecontroleerd (tip: laat er in geval van twijfel ook iemand anders een blik op werpen) kan de spanning erop worden gezet en kan de generator worden uitgeprobeerd.

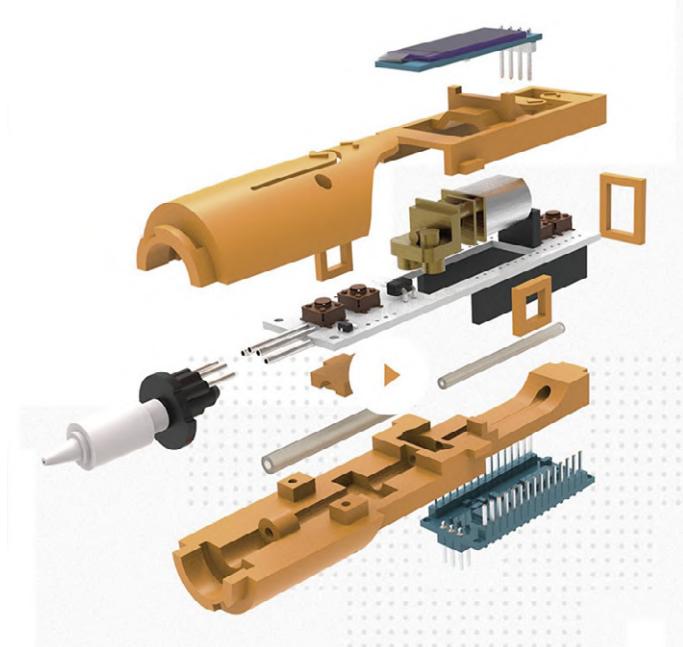
De generator kan de drie belangrijkste golfvormen (sinus, blok en driehoek) leveren met een voor deze prijs (nog net geen € 60) respectabel frequentiebereik van 1 Hz...10 MHz. Reuze praktisch is ook een sweep-modus om snel het gedrag van een schakeling over een groot frequentiebereik te bepalen.

De maximale uitgangsspanning bedraagt ongeveer 10 V; het offsetbereik bedraagt -5 V...+5 V. De vervorming van de uitgangssinus is bij 0,7 V en 1 kHz ongeveer 0,1% – helemaal niet zo beroerd dus. Alles bij elkaar genomen een heel leuke generator die in het lab van de beginnende hobbyist helemaal geen gek figuur slaat.

Handheld 3D-printer

Is hij om te knutselen of om te leren? Pijnig uw hersenen niet langer: 3D Simo-kit is beide! Als het gaat om 3D-printen, denkt u waarschijnlijk aan een vrij ingewikkeld en luidruchtig apparaat en niet aan een pen waarmee u met verschillende kleuren in drie dimensies kunt tekenen. Maar enige tijd geleden deed een 3D-gereedschap met kleine afmetingen zijn intrede in de wereld van de knutselaar. Het doet denken aan een klassiek lijmpistool. Dit elektrische gereedschap is te vergelijken met een 3D-printer als een handboor met een kolomboormachine. En veel stiller! Of anders gezegd: het is een PLA-extruder die u in de hand kunt houden als een potlood zodat u met de hand kunt 3D-printen. Hij is ontworpen als een handgereedschap, licht en gemakkelijk in het gebruik en is bedoeld voor liefhebbers van 3D-printen die graag het heft in eigen hand willen nemen, letterlijk en figuurlijk. Het is zowel een gereedschap om te retoucheren en af te werken als een gereedschap om handmatig dingen te creëren. De afmetingen zijn 138 x 25 x 21 mm, en hij weegt niet meer dan enkele tientallen gram. Hij wordt gevoed met 5 V via een USB-kabel (die 1,5 A moet kunnen leveren).

Het hier besproken model is ook een kit om zelf in elkaar te zetten. De hoofdprint is een Arduino Nano. De 3D Simo-kit is rechtenvrij: alle ontwerpgegevens zijn gratis beschikbaar op GitHub (de printbestanden, de 3D-geprinte onderdelen en zelfs de software). Met andere woorden: alles is aanpasbaar en personaliseerbaar.



Met deze kit krijgt u niet alleen de beschikking over een 3D-pen voor handmatig tekenen, het is ook een object om mee te leren en te experimenteren. Hij is dus zeker ook interessant voor gebruik in het onderwijs. ▶

180689-D-01



IN DE STORE

→ Pulsar Labs functiegenerator-bouwpakket
www.elektor.nl/18653



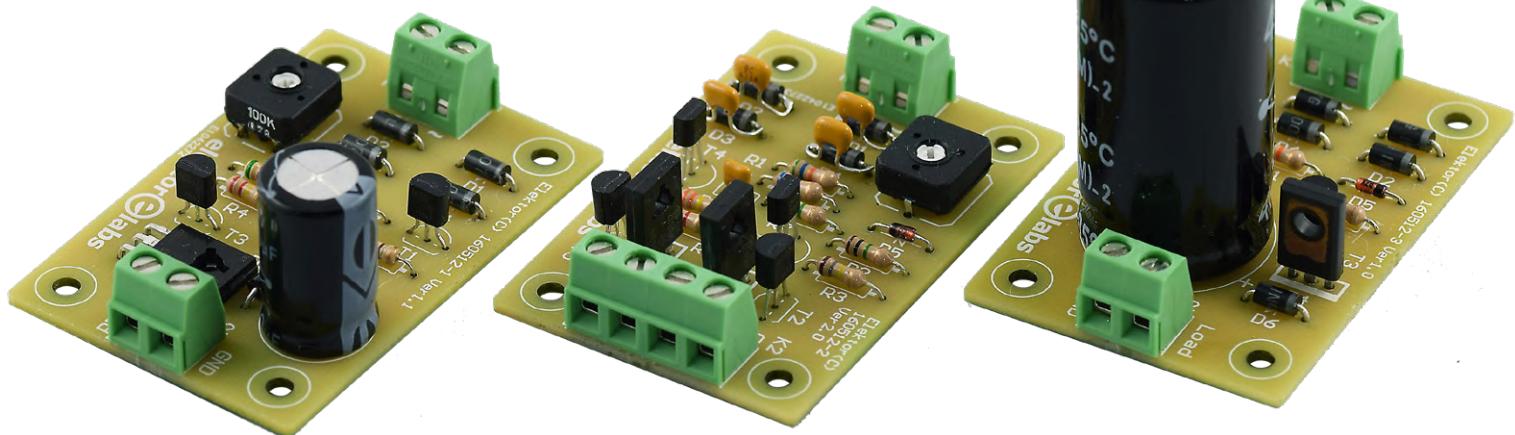
IN DE STORE

→ Simo 3D-pen kit
www.elektor.nl/18867

AC-in, DC-uit baristors

bouwstenen voor voedingsschakelingen

Michael A. Shustov (Duitsland) en Andrey M. Shustov (Rusland)



Het is bekend dat het gebruik van thyristoren en triacs in industriële en consumentenelektronica de netspanning vervuilt. Er worden harmonischen, ruis, bursts en pieken gesuperponeerd op wat toch een schone sinus uit het stopcontact zou moeten zijn. Een nieuw ontwikkelde elektronische deelschakeling met de naam baristor kan helpen om de netspanning zuiver te houden terwijl hij toch goede regeleigenschappen heeft. Dit artikel beschrijft de functionaliteit van de baristor en drie praktische AC-in, DC-uit schakelingen, compleet met print.

De barrièreweerstand (*barrier + resistor = baristor*) is een schakelend element voor vermogens- en analoge elektronica, waarvan de elektrische ‘ingang-naar-uitgang’ weerstand abrupt kan omklappen van geleidend naar niet-geleidend en andersom. Het element komt in de gewenste ‘toestand’ als een van te voren bepaalde drempelwaarde (‘barrière’) van het ingangssignaal wordt bereikt. Symbaristors zijn symmetrische baristors die bedoeld zijn voor wisselstroomtoepassingen (AC).

Baristor versus SCR

In de vermogenselektronica zijn SCR’s zoals thyristoren en triacs bekend als commutatie-elementen die kunnen schakelen van een niet-geleidende toestand naar een geleidende toestand wanneer de gate-elektrode wordt aangestuurd. Ze schakelen weer uit als momentane anodespanning naar nul gaat. Thyristoren en

triacs kunnen vrijwel elk gewenst deel van het sinussignaal van de wisselstroom afsnijden en zo het effectieve vermogen regelen dat naar de belasting gaat. Maar ze hebben enkele nadelen:

- beperkte schakelsnelheid;
- uitschakelen is onmogelijk zonder de stroom door het apparaat te onderbreken;
- vervorming van de netspanning-sinus;
- afname van de vermogensfactor;
- laag economisch rendement.

De bovenstaande nadelen kunnen grotendeels worden overwonnen door het gebruik van baristors en symbaristors die:

- het mogelijk maken om een wisselstrooms signaal op te delen in segmenten van instelbare breedte vol-

gens de eisen van de gebruiker;

- (dankzij sommeren) het vermogensverbruik van verwarming en verlichting kunnen regelen of stabiliseren;
- geschikt zijn voor gebruik in voedingseenheden voor het genereren van verschillende uitgangsspanningen.

U zou kunnen zeggen dat thyristoren en triacs schakelen op de ‘horizontale’ as (als functie van de tijd) terwijl baristors schakelen op de ‘verticale’ as (als functie van de spanning). Deze eigenschap van de baristor opent fundamenteel nieuwe mogelijkheden voor gebruik in communicatieapparatuur en vermogenselektronica.

Baristors zijn bedoeld om signalen te scheiden waarvan de amplitude boven of onder een door de gebruiker ingestelde drempelwaarde (barrière) ligt.

Baristortypen en -toepassingen

Baristors kunnen worden ingedeeld naar:

- geregeld of ongeregeld (d.w.z. met een bestuurde of onbestuurde drempelwaarde);
- AC of DC (asymmetrische en symmetrische baristors: symbaristors);
- eenkanaals of meerkanals (multi-level, multi-drempel baristors);
- schakeltype: aan / uit / wissel.

Dit zijn de namen van de leden van de baristorfamilie:

High-level baristor: er komt alleen spanning op de uitgang als het ingangsspanningsniveau een bepaalde drempelwaarde overschrijdt.

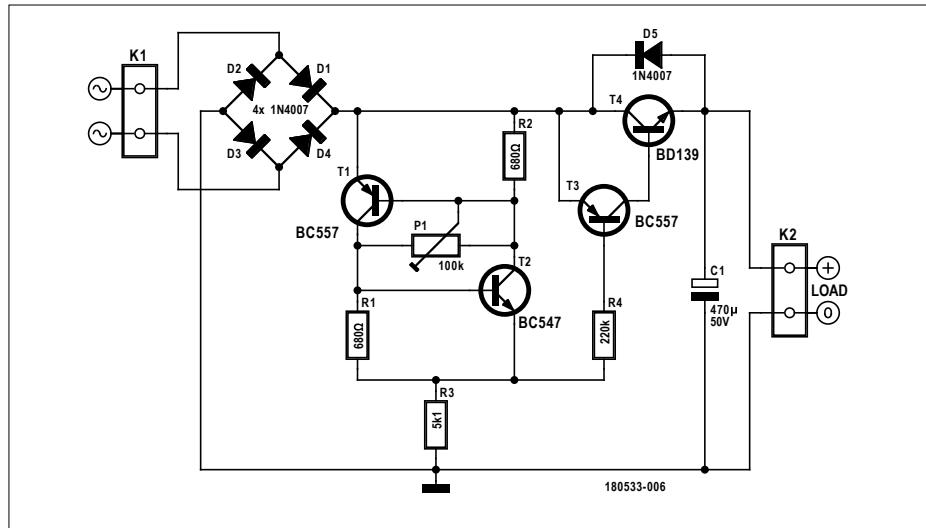
Low-level baristor: er is spanning op de uitgang, zolang het ingangsspanningsniveau beneden een bepaalde drempelwaarde blijft. Daarna verandert de toestand van de baristor en schakelt hij de belasting uit.

Switching single-threshold baristor: als de ingangsspanning langzaam toeneemt, gaat het signaal eerst naar de uitgang voor het lage signaalniveau. Nadat een bepaalde drempelwaarde is overschreden, wordt automatisch overgeschakeld naar de uitgang voor het hoge niveau.

Switching multi-threshold baristor: bij voortdurend toenemen of afnemen van de ingangsspanning schakelt de baristor over naar verschillende uitgangen. Het ingangssignaal wordt op die manier doorgegeven naar een uitgang afhankelijk van de spanning.

Met baristors en symbaristors kunnen we wisselstroomvermogen verdelen tussen verschillende gebruikers van elektrische energie, hogere harmonischen van de netspanning onderdrukken en de vermogensfactor vergroten (dus corrigeren) door handig gebruik te maken van een deel van de sinusgolf dat SCR's 'weggooien'. Al deze mogelijkheden kunnen helpen om de efficiëntie van het elektrisch energiegebruik aanzienlijk te verbeteren. Barrièrerestanden kunnen worden toegepast in de volgende gebieden:

- efficiënte, kleine voedingseenheden met geringe verliezen;
- spanningsconverters en -stabilisatoren;
- bescherming van elektronische componenten en schakelingen



Figuur 1: Basisversie van de single-threshold-baristor zoals ontwikkeld uit een idee in het boek "Electronic Circuits for All".

Prestaties van de single-threshold baristor.

Zie de schakeling in figuur 1. Enkele meetresultaten met een belasting van $150\ \Omega$ en een vaste instelling van P1:

AC in	DC uit
12 V	11,4 V
16 V	9,9 V
20 V	9 V
24 V	8,6 V

Bij een ingangsspanning van 20 VAC daalde de uitgangsspanning van 9 naar 6,7 V als de belastingsweerstand werd gehalveerd naar $75\ \Omega$. Bij bepaalde ingangsspanningen en instellingen van P1 bleek de uitgangsspanning in de tijd te variëren. De onbelaste stroom kan minder dan 1 mA zijn maar ook een paar mA, afhankelijk van de instelling van P1 en de ingangsspanning.

gen, vermogensapparatuur en communicatielijnen;

- systemen voor meerkanals afstandsbediening en communicatiesystemen, inclusief een HF-signaal gesuperponeerd op een tweedraads lijn;
- meetinstrumenten en transducers;
- apparaten voor pulstechniek;
- isolatie, scheiding of genereren van signalen;
- analoge elektronica voor signalscheiding in componenten met behulp van een door de gebruiker ingestelde drempelwaarde.

Als u meer wilt weten over de bronnen en werking van de baristor, koop dan het boek "Electronic Circuits for All", uitgegeven door Elektor; zie het kader **In de store**.

Baristor nummer 1: de basis

We gaan nu kijken naar de praktische schakelingen met de baristor. In **figuur 1** zien we het schema van een baristor met

één schakeldrempel, die is ontworpen door Ton Giesberts van Elektor Labs op basis van een versie in het boven genoemde boek.

Zolang de spanning op de uitgang van de bruggelijkerichter D1...D4 (ongeveer 16 VAC in dit geval) onder de drempel die is ingesteld met T1/T2 blijft, geleidt de 'samengestelde transistor' T3/T4. Het spanningsniveau van D1...D4 min de spanningsval over T4 wordt opgeslagen in buffercondensator C1. Als de ingangsspanning de ingestelde drempel overschrijdt, blokkeert T4 en de uitgangsspanning neemt niet verder toe. Op die manier kan de gelijkspanning op de uitgang worden ingesteld, afhankelijk van het drempelniveau van T1/T2, dat wordt ingesteld met P1. Als we P1 vervangen door een optocoupler of een JFET kunnen we de uitgangsspanning laten besturen vanuit een andere schakeling. Als de wisselspanning aan de ingang hoger is dan ca. 16 V, moeten de weerstanden R3 en R4 worden aangepast. Afhankelijk van de waarden van R3 en R4, en andere



ONDERDELENLIJST

Single-threshold baristor / print 160512-1 v1.1 / schema figuur 1.

Weerstanden:

R1,R2 = 680 Ω , 5%, 0,25 W

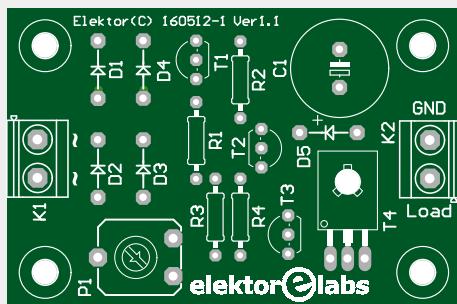
R3 = 5k1, 1%, 0,25 W

R4 = 220 k, 5%, 0,25 W

P1 = instelpotmeter, 100 k, 20%, 0,15 W, CB10LV104M Citec-TE Connectivity

Condensatoren:

C1 = 470 μ F/50 V, 20%, Ø 13mm, steek 5 mm



Halfgeleiders:

D1,D2,D3,D4,D5 = 1N4007

T1,T3 = BC557B

T2 = BC547B

T4 = BD139



Diversen:

K1,K2 = 2-polige printkroonsteen, steek 0,2" (5,08mm) bijv. Phoenix Contact MKDSN 1,5/2-5,08

Print # 160512-1 v1.1

omstandigheden, varieert de stroom/spanning-karakteristiek van de schakeling rondom T1/T2 van 'thyristor' tot 'zenerdiode'. Het probleem daarbij is... het is niet echt een zenerdiode. Als T1 en T2 niet geleiden, loopt er nog steeds stroom door R2-P1-R1-R3. Dus als de waarde van P1 te laag is, of die van R3 te hoog, dan kan T3 aan- of uitschakelen, d.w.z. er is geen echte drempel en de uitgangsspanning zal variëren met de wisselspanning op de ingang. Ook kan er een zeer laagfrequente instabiliteit ontstaan bij bepaalde instellingen van P1 in combinatie met verschillende wisselspanningen aan de ingang. Erger nog: de uitgangsimpedantie van de transformator beïnvloedt de werking en stabiliteit van de schakeling. Een condensator van 10 μ F van de basis van T4 naar massa kan de stabiliteit van de schakeling verbeteren. Een verandering van de belastingsweerstand zal de uitgangsspanning veranderen.

Enkele belangrijke meetresultaten van een proefopstelling van de schakeling zijn te lezen in het kader **Prestaties van de single-threshold-baristor**.

Verbeterde single-threshold baristor

Ook de schakeling in **figuur 2** is gebaseerd op een idee uit het boek 'Electronic Circuits for All'.

Deze schakeling is niet echt bedoeld als een volledig werkend apparaat, maar meer als een schakeling die u kunt aanpassen aan uw eigen toepassing, om u nieuwe ideeën te geven of om deel uit te maken van een schakeling voor een ander doel. De door Elektor Labs voor

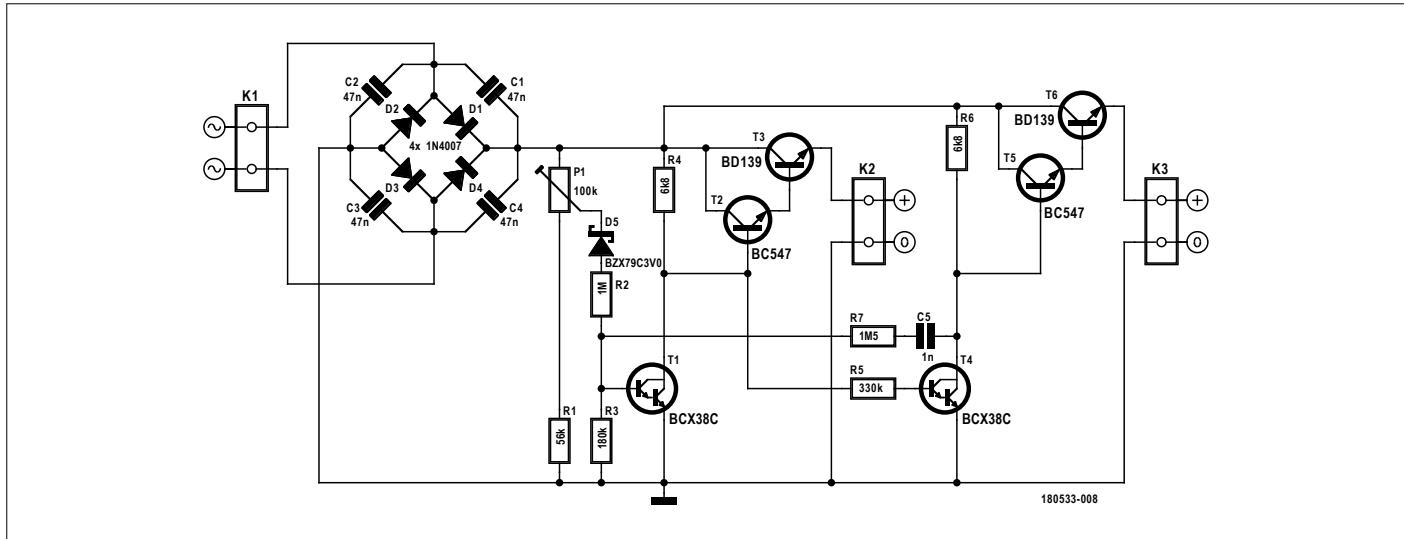
deze schakeling ontworpen print maakt het testen en experimenteren gemakkelijker, maar de schakeling is ook gemakkelijk te testen op een breadboard. Er is een positief terugkoppelnetwerk R7/C5 toegevoegd om ongewenst schakelen tegen te gaan en om te zorgen dat de uitgangstrappen beter schakelen. De basis van T1 werkt daardoor als de input van een Schmitt-trigger. Vanwege de hysterese, die afhankelijk is van de wisselspanning op K1, verandert de op P1 ingestelde fase met het niveau van de ingangsspanning. Afhankelijk van de toepassing en de gebruikte wisselspanning moeten de waarden van sommige componenten iets worden aangepast. Een simulatieprogramma, zoals LTSpice, komt daarbij goed van pas. Het gebruik van een spanningsbron in plaats van een 'echte' transformator in de simulatie geeft bedrieglijke resultaten. Elke transformator heeft een complexe uitgangsimpedantie die voornamelijk bestaat uit een combinatie van een weerstand en een inductantie. Daardoor verandert de ingangsspanning op het moment dat de baristor schakelt, en dat kan instabiliteit veroorzaken, vooral als T1 een grote versterking heeft.

In de originele schakeling in het boek werd gebruik gemaakt van hoogspannings-Darlingtons van het type BC373 die helaas moeilijk te vinden bleken. De specificaties zijn: 80 V/1 A in een TO92-behuizing. De parameter h_{FE} bij 100 mA is gespecificeerd als >20000. Gelukkig zijn er heel wat vervangende types die nog steeds verkrijgbaar zijn, zoals de BCX38C (60 V/0,8 A); KSP13B (30 V/0,5 A); MPSA14, (30 V/0,5 A);

MPSA29 (100 V/0,5 A); ZTX603 (80 V/1 A); ZTX605 (120 V/1 A); ZTX614 (100 V/0,8 A); BC517-D74Z (30 V/1,2 A maar met verwisselde collector- en emitteraansluiting!)

Maar pas op met een TO92-behuizing voor T6: omdat deze behuizing weinig vermogen kan dissiperen is het verstandiger om een grotere behuizing te kiezen. Uiteindelijk hebben we de goede oude BD139 gebruikt in combinatie met een standaard BC547C om een discrete Darlington-vervanging te maken voor de origineel gebruikte BC373. De BD139 heeft een SOT32 (TO126)-behuizing en kan een vermogen van maximaal 1,25 W dissiperen zonder extra koeling (de juctie-temperatuur is dan 150 °C bij een omgevingstemperatuur van 25 °C). Gebruik nooit een transistor bij een juctie-temperatuur van 150 °C, houd de temperatuur beneden 70 °C, anders gaat het ten koste van de levensduur. En waarom kiezen we niet voor een BD679-Darlington om het aantal componenten te reduceren? Omdat vrijwel alle power-Darlingtons interne weerstanden parallel aan de basis/emitter-overgang van beide transistoren hebben. Die beïnvloeden de versterking van de transistor, afhankelijk van de collectorstroom, terwijl bij kleine stromen de versterking veel kleiner is dan die van Darlingtons zonder interne weerstanden.

Terug naar het schema. De maximale spanningsval van K1 naar K2 of K3 is ongeveer 4 V bij 100 mA. Dit wordt veroorzaakt door D1...D4 en een kleine spanningsval over R4 en R6 en de basis-emitterspanning van T2/T3 en T5/T6. Dus zonder extra koeling van T3/T6 moet de maximale uitgangsstroom



Figuur 2: Deze versie van de baristor heeft verschillende verbeteringen vergeleken met de basisversie. Hij biedt veel mogelijkheden voor experimenteren en aanpassen aan uw eisen.

Prestaties van de verbeterde single-threshold baristor.

Zie de schakeling in figuur 2.

Een voorbeeld van maximale faselhoeken bij minimale en maximale instelling van P1 (18 V op K1):

- P1 volledig rechtsom: 29° – 148°
- P2 volledig linksom: 79° – 126°
- Onbelaste stroom: 1,71 mA (12 V op K1)
- Onbelaste stroom: 3,75 mA (24 V op K1)

Met gelijke belastingsimpedanties op K2 en K3:

- Ingangsstroom ($U_{in} - 3,2 \text{ V}$) / R_{load} , gemeten bij 300 mA (K1) en belastingen van 75 Ω.
- Ingangsstroom ($U_{in} - 3 \text{ V}$) / R_{load} , gemeten bij 120 mA (K1) en belastingen van 75 Ω.

Met twee identieke belastingen op K2 en K3 is de ingangsstroom vrijwel onafhankelijk van de instelling van P1 en varieert alleen een paar procent vanwege componenttoleranties.

worden beperkt tot enkele honderden milliampères.

Verwijderen van D5 vergroot het bereik van P1 en heeft meer effect bij lage ingangsspanningen. Om het bereik van P1 te veranderen bij grote ingangsspanningen, kunt u overwegen ook R1

en R3 te veranderen en misschien de werkspanning van D5. Maar vanwege de lage stroom door D5 is de spanningsval over die diode minder dan de gespecificeerde werkspanning.

We hebben deze schakeling weer op de proef gesteld in ons lab en de resulta-

ten met opmerkingen zijn te vinden in het kader **Prestaties, single-threshold baristor, verbeterde versie**.

In onze testopstelling hebben we een variabele autotransformator gebruikt, ook bekend onder de handelsnaam Variac. We gebruiken die om de pri-



ONDERDELENLIJST

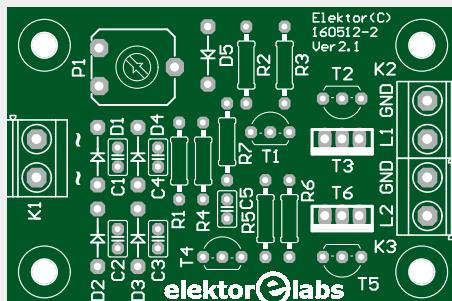
Single-threshold baristor, verbeterde versie / print 160512-2 v2.1 / schema figuur 2.

Weerstanden:

R1 = 56 k, 5%, 0,25 W
R2 = 1 M, 5%, 0,25 W
R3 = 180 k, 5%, 0,25 W
R4,R6 = 6,8 k, 5%, 0,25 W
R5 = 330 k, 5%, 0,25 W
R7 = 1M5, 5%, 0,25 W, 250 V
P1 = instelpotmeter 100 k

Condensatoren:

C1,C2,C3,C4 = 47 n, 50 V, keramisch X7R, steek 0,1"



C5 = 1 n, 100 V, keramisch X7R, steek 0,1"

Halfgeleiders:

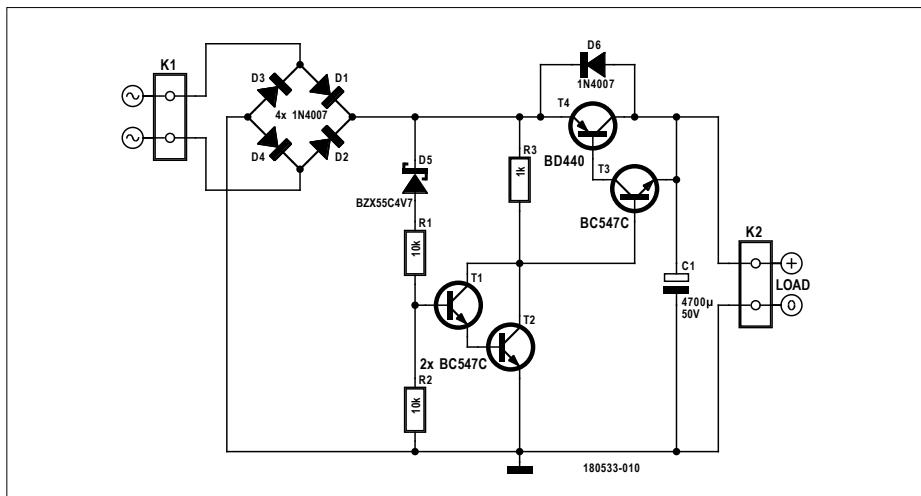
D1,D2,D3,D4 = 1N4007
D5 = BZX79-C3V0, 3 V, 500 mW
T1,T4 = BCX38C, Darlington, 60 V, 0,8 A
T2,T5 = BC547



T3,T6 = BD139

Diversen:

K1,K2,K3 = 2-polige printkroonsteen, steek 0,2" (5,08 mm) MKDSN 1,5/ 2-5,08
Print # 160512-2 v2.1



Figuur 3: Deze versie is het resultaat van een compromis tussen het aantal componenten (laag) en de stabilisatie (hoog).

maire spanning van een 'safety transformer' van type FL30/12 van Block te variëren. Deze heeft twee primaire wikkelingen van 115 V en secundaire wikkelingen van 12 V die elk 1,25 A kunnen leveren. Op die manier is een veilige

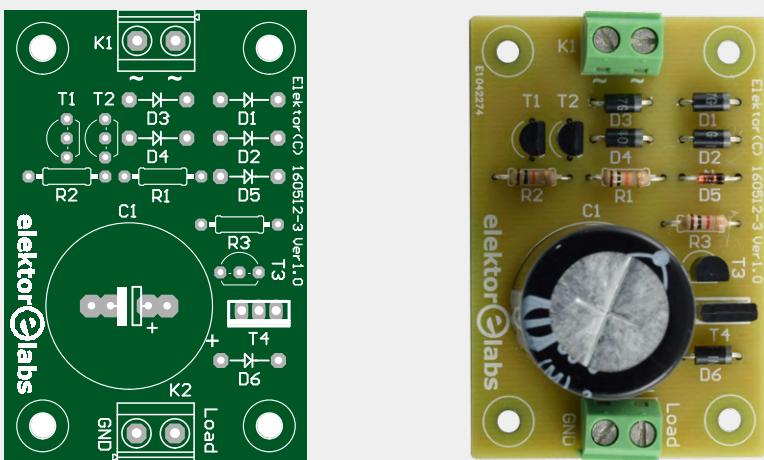
testopstelling gegarandeerd. Maar pas op met deze opstelling. De maximale uitgangsspanning van de autotransformator is typisch hoger dan de ingangsspanning. Wij hebben er een gebruikt die nog gespecificeerd is voor gebruik met

de ouderwetse netspanning van 220 V. De uitgang is gespecificeerd als max. 250 V/2 A. Omdat de netspanning tegenwoordig hoger is, op dit moment 230 V nominaal, zal de onbelaste uitgangsspanning nog hoger zijn, in de buurt van 261 V. Maar het blijkt zelfs nog erger als we echt gaan meten. We vonden een uitgangsspanning van maar liefst 274 V bij een netspanning van 239 V op onze testlocatie. Zet daarom de autotransformator nooit op het maximum in deze testopstelling. Dat zou vrijwel zeker de veiligheidstransformator die erop is aangesloten vernielen.

Bij een eerste test werd een kleine transformator gebruikt en R7/C5 waren genoeg om het foute schakelen te verhelpen. Maar toen we aan de slag gingen met de hierboven beschreven testopstelling bleek dat niet genoeg. De vier diodes van de gelijkrichter moesten worden ontkoppeld met 4 condensatoren van 47 nF.

Minder componenten, betere stabilisatie

Om u een idee te geven van eenvoudig haalbare verdere verbeteringen en optimalisaties, hebben we een kleine, slanke baristor-schakeling ontworpen die een uitgangsspanning produceert die minder afhankelijk is van de belasting en de ingangsspanning, terwijl hij is opgebouwd uit een klein aantal componenten. Het principe is eenvoudig: de uitgangstrap functioneert als een schakelaar en moet worden uitgeschakeld zodra de ingangsspanning een bepaalde drempel overschrijdt. De drempelwaarde wordt grotendeels bepaald door zenerdiode D5. R1 beperkt hier de stroom bij grotere ingangsspanningen en R2 zorgt ervoor dat er wat stroom door de zener loopt voordat de discrete Darlington T1/T2 inschakelt. T1/T2 trekt de basis van de samengestelde transistor T3/T4 naar massa, waardoor de uitgang wordt uitgeschakeld. Om de spanningsval over T4 te minimaliseren is een combinatie van een NPN-kleinsignaaltransistor en een PNP-vermogenstransistor gebruikt, dat scheelt enkele honderden millivolts. Om de spanningsval ten gevolge van de basisstroom zo klein mogelijk te maken, is de waarde van R3 slechts 1 kΩ. Dat betekent dat een weerstand van een kwart watt zijn maximum toelaatbare vermogen zal dissiperen bij een ingangsspanning van 21 V. Alle metingen met ingangsspanningen boven 21 V gaan de veilige specificaties voor R3 te



boven. Gebruik in dat geval een weerstand van 500 of 600 milliwatt. Er zijn ook weerstanden in deze behuizing die voor 1 watt zijn gespecificeerd. Montereer

Prestaties van de afgeslakte baristor.

Zie de schakeling in figuur 3. Meetresultaten met T4 = BD440 en een belasting van 150 Ω:

U_{in} [V]	I_{in} [mA]	U_{uit} [V]	U_{rimpel} [mV]
AC	AC	DC	AC _{pp}
5	44	4,28	24
6	64	4,48	56
7	74	4,46	60
8	80	4,44	60
9	85	4,43	60
10	89	4,42	60
11	92	4,42	60
12	94	4,41	60
13	96	4,41	60
14	98	4,40	60
15	101	4,39	60
16	102	4,38	60
17	104	4,37	60
18	106	4,36	60
19	109	4,36	60
20	110	4,35	60
21	110	4,34	60
22	111	4,33	60
23	112	4,33	60
24	113	4,32	60

Bij kleine belastingen wordt de meeste ingangsstroom veroorzaakt doordat R3 naar massa wordt getrokken door T1/T2. Overweeg een grotere waarde voor R3 bij gebruik met zeer kleine belastingen. De spanningsval over T4 neemt dan wel toe, maar toch neemt de totale efficiëntie toe. Iets om mee te experimenteren.

Er is nog een snelle test uitgevoerd met een veel grotere belasting van 10 Ω. Bij een ingangsspanning van 12 V was de ingangsstroom voor een belasting van 10 Ω gelijk aan 0,57 A. De uitgangsspanning was 3,99 V. Maar zonder koeling werd T4 heet! De efficiëntie is niet optimaal maar is meer nodig dan vermenigvuldigen van de spanning en de stroom omdat de stromen en spanningen in een baristor verre van sinusvormig zijn.

zulke weerstanden een beetje boven de print. De dissipatie zal het grootst zijn zonder belasting.

Omdat T3/T4 permanent uitgeschakeld zal zijn, is de spanning over R3 bijna gelijk aan de ingangsspanning, verminderd met de spanning die verloren gaat in de bruggelijkerichter. Het feit dat T1/T2 boven een bepaalde spanning inschakelt, verandert niet veel aan de situatie. Met een zener van 4,7 V (0,5 W) is de uitgangsspanning ongeveer 4,4 V bij een belasting van 150 Ω. Bij een belasting van 10 Ω neemt de uitgangsspanning af tot iets minder dan 4 V. De onbelaste spanning is ongeveer 4,67 V. Aanvankelijk gebruikten we een BD140 en bij een kleine uitgangsstroom werkte dat goed. Maar met een grotere belasting, zoals een weerstand van 10 Ω, sloot hij bijna meteen kort. Onze testopstelling kan een nominale wisselstroom van 1,25 A RMS leveren, dus de piekstroom is nog groter. Een BD140 kan piekstromen van 1,5 A verwerken, wat vooral met een dikke condensator aan de uitgang niet voldoende is. We hebben hem vervangen door een BD440. Dit type is nog steeds leverbaar (BD440S, Mouser # 512-BD440S). De BD440 is een 60 V/4 A (7 A puls) PNP-vermogenstransistor. Er zijn andere geschikte kandidaten in een TO-126 (SOT-32)-behuizing om T4 te vervangen, zoals de 2N5195 (PNP, 80 V/4 A), BD438 (45 V/4 A) of BD442G (80 V/4 A)\. De afstand tussen de aansluitpennen kan iets verschillen, afhankelijk van de fabrikant. Soms is die 2,39 mm (TO-225AA) in plaats van 2,29 mm. Ook kan de behuizing iets groter zijn, bijvoorbeeld 14,2 x 8 mm in plaats van de originele TO-126 afmetingen van 11,1 x 7,8 mm. Gelukkig passen al die varianten nog wel op de print. De ruimte voor de (grote) buffercondensator C1 is ontworpen voor condensatoren met een penafstand van 5, 7,5 en 10 mm

en een maximum diameter van 22 mm. De uitgangsspanning kan instelbaar worden gemaakt door R2 te vervangen door een potentiometer. De uitgangsspanning is dan inderdaad instelbaar, maar bij hogere spanningen wordt hij steeds meer afhankelijk van de ingangsspanning. De spanning over R1 zal meer toenemen met een lagere waarde van P1 en wordt dan meer dominant dan de zenerspanning. Om een andere uitgangsspanning in te stellen, kunt u het best de zenerspanning veranderen. Als er spanningspieken optreden bij de bruggelijkerichter, kunnen die worden gereduceerd door een diode tussen D5 en R3 in te voegen.

We hebben ook deze schakeling getest in ons lab, zie voor meetresultaten het kader **Prestaties van de afgeslakte baristor.**

Aan de slag!

Alle drie de schakelingen bleken te werken, maar ze zijn experimenteel en het zijn geen complete projecten voor gebruik in industriële toepassingen. Vergeet niet dat er geen stroombegrenzing of beveiliging tegen overbelasting is. Houd ook de temperatuur van de uitgangstransistoren in de gaten bij grote belastingen en ingangsspanningen. Neem het zekere voor het onzekere en bevestig een plaatje aluminium of iets dergelijks aan de vermogenstransistor(s) voor wat koeling. ▶

(180533-03)

@ WWW.ELEKTOR.COM



→ Electronic Circuits for All, door M.A. Shustov en A.M Shustov.
www.elektor.nl/electronic-circuits-for-all

→ Single-threshold baristor, kale print no. 160512-1 v.1
www.elektor.nl/160512-1

→ Single-threshold baristor, verbeterde versie, kale print 160512-2 v.2.1
www.elektor.nl/160512-2

→ Afgeslakte baristor, kale print no. 160512-3 v.1.0
www.elektor.nl/160512-3

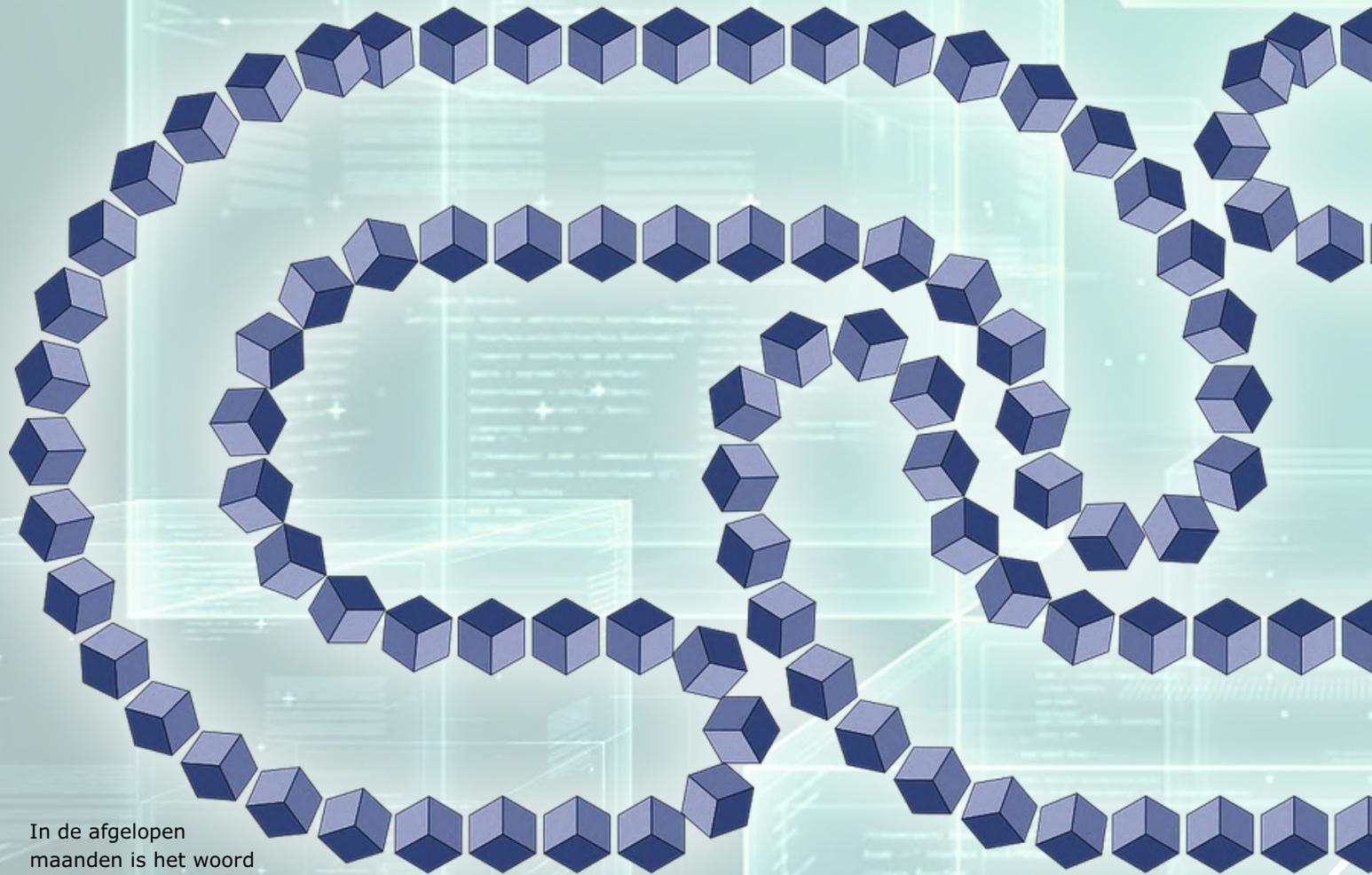


HomeLab Helicopter

de wondere wereld van de elektronica in vogelvlucht

samengesteld door Clemens Valens (Elektor Labs)

Over blockchains, het universum en tijdreizen



In de afgelopen
maanden is het woord
blockchain een veelgebruikt

buzzword geworden waarvan men beweert dat het een positieve invloed heeft op verkoopcijfers. Gebruik het woord blockchain en je bent er zeker van dat er naar je geluisterd wordt. Het feit dat u dit leest, is al een bewijs.

Maar wat is een blockchain? Volgens Wikipedia kunnen we het definiëren als een keten van blokken waarbij elk blok een handtekening (een 'hash') bevat van de (inhoud van de) blokken die eraan voorafgaan, een tijdstempel en transactiegegevens. Het berekenen van de hash van een blok is ingewikkeld, waardoor

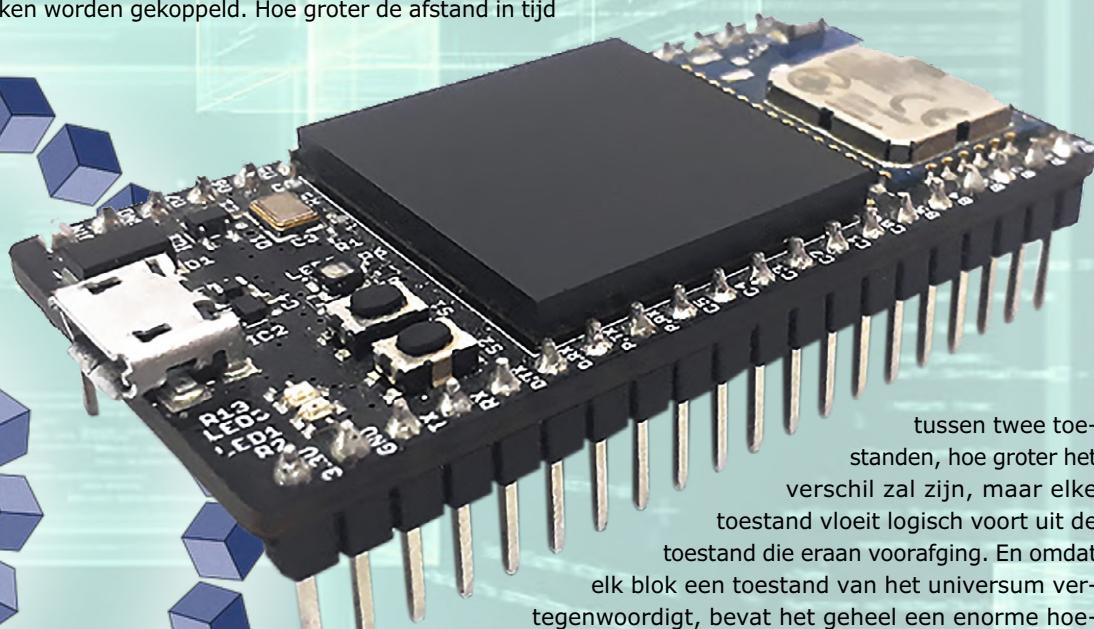
dat een dure en tijdrovende activiteit is. Bovendien moet elke nieuwe handtekening worden geaccepteerd door meerdere controleurs. Veranderen van de inhoud van een blok ergens in de keten kan alleen als u de hashes van alle blokken die volgen corrigeert, anders wordt de blockchain ongeldig. Als het updaten van één handtekening al 'lastig' is, kunt u zich wel voorstellen hoeveel moeite het kost om dat voor honderdduizenden blokken te doen! Een blockchain is daarom bijzonder resistent tegen verandering van de data die hij bevat.

Binnenkort op een breadboard bij u in de buurt

Blockchains worden meestal geassocieerd met cryptovaluta, maar het Internet of Things (IoT, *buzzword 2*) is ook een interessant domein voor op blockchains gebaseerde beveiligde datacommunicatie. Tegen de tijd dat u dit leest, heeft de startup Elkrem uit Egypte een Kickstarter-campagne voor zijn Blockchain IoT Development Board mogelijkwijs al afgerond.

In de echte wereld

Een tijdje geleden drong het tot me door, dat we het concept van virtuele blockchains kunnen vertalen naar de echte wereld. Stel dat we de toestand van het universum op een bepaald moment, laten we zeggen t_0 , kunnen vastleggen in een blok. De toestand van het universum één moment later, op t_1 , is bijna identiek aan de vorige toestand maar niet helemaal: sommige deeltjes hebben zich bewogen. Omdat de bewegingen van de deeltjes de natuurwetten volgen, is de nieuwe toestand acceptabel, de handtekening is geldig, en het blok kan aan de voorafgaande blokken worden gekoppeld. Hoe groter de afstand in tijd



tussen twee toestanden, hoe groter het verschil zal zijn, maar elke toestand vloeit logisch voort uit de toestand die eraan voorafging. En omdat elk blok een toestand van het universum vertegenwoordigt, bevat het geheel een enorme hoeveelheid informatie. (Big Data. Pats! Tien punten voor het gebruik van nog een *buzzword*.)

Na deze inleiding is het (hopelijk) gemakkelijk om over te stappen naar tijdreizen (wat misschien wel de echte reden is, waarom u dit artikel bent gaan lezen). Stel dat u terug wilt reizen in de tijd. Hoe ver precies, dat maakt niet uit. Het kan één nanoseconde zijn of een miljoen jaar of meer, de redenering is hetzelfde.

We zetten blok t_0 bij het begin van de tijd en het heden, 'nu', bij t_n . U reist terug in de tijd door uzelf toe te voegen aan de toestand blok t_{n-m} (met $m \leq n$). Om dit mogelijk te maken, moet u alle m blokken die volgen bijwerken. Het is duidelijk dat dat niet eenvoudig zal zijn. Aanpassen van de volgende toestand, t_{n-m+1} , om te zorgen dat de toevoeging van één deeltje wordt geaccepteerd is al moeilijk, maar om dat te doen voor de verplaatsing

of het verwijderen van de luchtmoleculen die ruimte moeten maken voor uw lichaam is waarschijnlijk onmogelijk. Zelfs als u dat zou kunnen, zou er heel veel energie voor nodig zijn. (Maar waar moet die vandaan komen?) En dan moet u hetzelfde nog doen voor de toestanden t_{n-m+2}, t_{n-m+3} enzovoort.

U kunt het aantal benodigde berekeningen verminderen door uw aanwezigheid in het verleden tijd zo kort mogelijk te houden, maar een paar toestanden lang, zodat de verstoringen die u hebt veroorzaakt snel wegsterven, zodat latere toestanden onveranderd blijven. Uw impact op het verleden moet zo klein mogelijk zijn.

Zelfs als we aannemen dat u deze problemen kunt oplossen, dan zult u ook nog soortgelijke dingen moeten corrigeren als u weer terugreist naar het heden. Ieder stofdeeltje en elke herinnering (een toestandsverandering in uw brein) dat u tijdens uw bezoek aan het verleden hebt opgepikt zal verantwoord moeten worden in de toestanden die volgen op uw bezoek. Om die enorme hoeveelheid werk te voorkomen, doet u er verstandig aan om niets mee te brengen uit de verleden tijd naar het heden. Maar dat maakt het reizen in de tijd natuurlijk tamelijk zinloos.

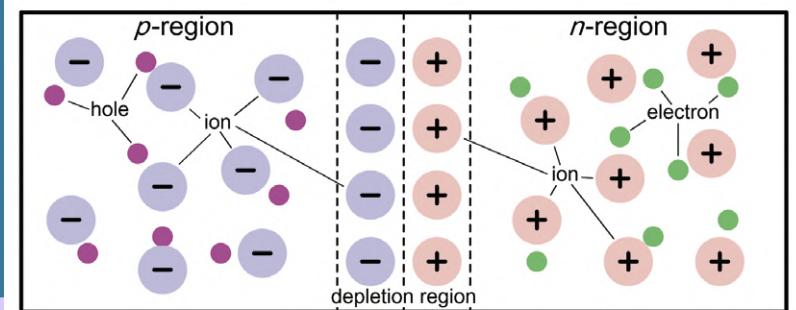
Wormgaten



Maar, hoor ik u roepen: "Hoe zit het dan met wormgaten?" Volgens sommige theorieën zouden wormgaten het mogelijk maken om door de tijd te reizen. Welnu, blockchains kunnen ook wormgaten hebben, want niets verbiedt ons om meer dan één blok aan dezelfde keten te rijgen. In blockchain-terminologie heet zo iets een vork en het resultaat is dat er twee of meer paden voorwaarts ontstaan. (Huiswerk: beantwoord de vraag "Kan het universum gevorkt worden?" in 500 woorden.) Eén van die paden zou terug kunnen voeren naar een blok eerder in de keten. Dat blok hoeft niet te linken naar het alternatieve vorige blok. Maar zelfs als u via een wormhole terug zou kunnen gaan naar een blok in het verleden, dan zouden dezelfde problemen als eerder beschreven ontstaan, wanneer u dat blok zou binnengaan.

En reizen naar de toekomst? Dat zou betekenen dat u vooruit gaat naar een toestand t_{n+1} , die nog niet bestaat. (Als het wel zou bestaan, zou dat betekenen dat wij in het verleden leven. Volgens onze blockchain-analogie zou dat impliceren dat al onze acties al zijn voorbeschikt, anders zou de toekomst ongeldig zijn. Vaarwel, vrije wil.) U zou die toestand zelf moeten aanmaken, compleet met het pad dat het verbindt met de toestand van het heden, tn. Hoe u de toestanden in dat pad inricht, mag u zelf weten, zolang ze maar een geldige blockchain vormen. Dit impliceert dat de toekomst alleen afhankelijk is van uw verbeelding. Gewoon uw ogen sluiten en wegdrömen zou een veel gemakkelijker alternatief zijn voor het reizen naar de toekomst. En zou veel minder energie kosten.

Het kristal NaSn_2As_2 kan zowel een elektron als een gat bevatten, waardoor het mogelijk zou zijn om halfgeleiders te maken die niet uit meerdere lagen hoeven te bestaan.





Homelab-hebbeding

Soms wilt u een spanning of een stroom meten op een plaats waar het het display van uw multimeter moeilijk zichtbaar is terwijl u de meetpennen op hun plaats houdt. Een Bluetooth-multimeter zoals deze kan dan uitkomst bieden. Een apparaat met een internetverbinding (zoals een smartphone) werkt als een mobiel display dat u op een goed zichtbare plaats kunt neerzetten. Als zelfs dat niet mogelijk is, kan de smartphone de geregistreerde waarden hardop voorlezen, terwijl u zich concentreert op de probes. Deze handige functies zijn op zich al voldoende om zo'n multimeter onmisbaar te maken, maar er valt nog meer over te vertellen. U kunt ook data loggen en plotten en zelfs meerdere meters weergeven op één telefoon. Niet gek toch?

www.elektor.nl/owon-ow16b-digital-multimeter-with-bluetooth

Roken is echt gevaarlijk!



Dat wist u natuurlijk al. Misschien hebt u uw ouderwetse tabakstaafjes al vervangen door de, hopelijk veiliger, e-sigaret? De gezondheidsrisico's op lange termijn van de vloeistoffen die in deze apparaten worden gebruikt zijn nog steeds niet bekend, maar waarschijnlijk wist u dat ook al. Wat u waarschijnlijk nog niet weet, is dat in het Franse plaatsje Plougoulm een man brandwonden heeft opgelopen, toen de accu van zijn e-sigaret vlam vatte bij een kortsluiting met zijn autosleutels in zijn broekzak. Het slachtoffer vertelde, dat hij een knal hoorde en dat er een 40 centimeter lange steekvlam uit zijn broek kwam. Toen hij probeerde om de brandende (!) e-sigaret uit zijn zak te halen, raakte hij gewond aan zijn hand. (Hij mag van geluk spreken dat hij er zo gemakkelijk vanaf kwam.) Soortgelijke incidenten zijn bij kortsluiting door munten voorgevallen.

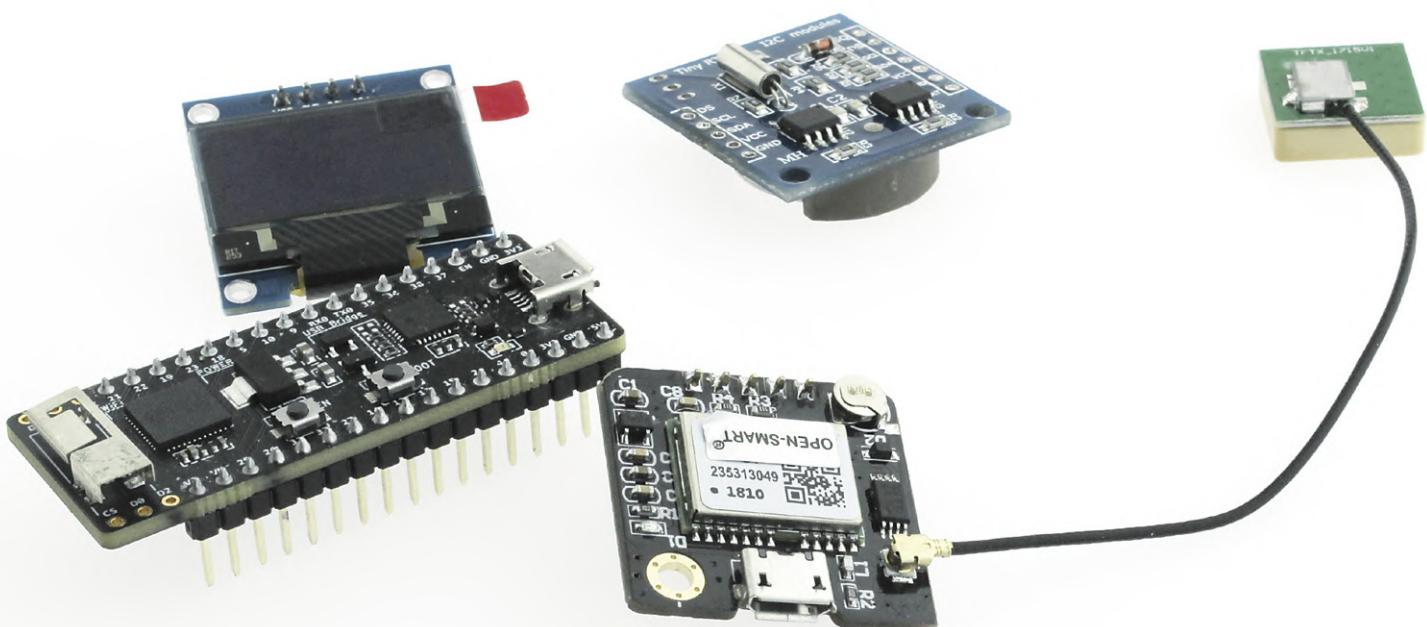
Maar het kan nog erger. In Fort Worth, in Texas, is een man gedood toen zijn e-sigaret in zijn gezicht ontplofte en een halsslagader openscheurde.

ESP32 als tijdserver

**ESP32 + RTC + GPS + display =
eigen NTP-server in uw LAN**

Mathias Claussen (Elektor Labs)

Vaak is het handig om een eigen NTP-server in uw LAN te hebben. Niet alleen in situaties met een van internet afgeschermd net en allerlei software die op een tijdsreferentie is aangewezen. Ook voor eigen projecten die hun tijd van het internet halen is het vaak prettig als u de verzonden NTP-data kunt manipuleren. Dat – en meer – is met de volgende beschrijving (relatief) eenvoudig te doen.



Figuur 1. Deze vier modules vormen samen een NTP-Server.

Eenvoudig gezegd is een NTP-server niets anders dan een ‘internet-adres’, dat bij correcte aanroep tamelijk exacte tijddata via het Network Time Protocol terugstuurt. Wie geïnteresseerd is in de details van NTP kan op bijvoorbeeld Wikipedia nadere informatie en links vinden. Om PC’s en andere met het internet verbonden microprocessoren na inschakelen van de juiste tijd (en datum) te voorzien, beschikken deze praktisch allemaal over een met een batterij gebufferde

RTC (Real Time Clock). Deze levert de data dan aan het OS van de processor en daarmee ook aan de daarop draaiende software. Hiermee wordt niet alleen gegarandeerd dat een kalender correct functioneert, maar ook krijgen zo weggeschreven of veranderde bestanden een correct tijdstempel en worden op basis van de tijddata belangrijke taken uitgevoerd.

Die tijd moet echter ook op de langere duur blijven kloppen en niet afhankelijk

zijn van een interne klokgenerator (lees: het kristal van de RTC). Daarom is het zinvol om besturingssystemen van computers met een eigen RTC-tijd regelmatig te synchroniseren met een referentie. En dat gebeurt dus via NTP.

Natuurlijk kan de tijd ook via DCF77 of een GPS worden verkregen, maar op de eerste plaats is een software-routine voor het afvragen van een NTP-server eenvoudiger en goedkoper dan speciale eigen hardware. En op de tweede plaats bren-

gen beide tijdsbronnen zo hun eigen problemen met zich mee: de DCF77-zender in Mainflingen heeft weliswaar een grote maar niettemin beperkte reikwijdte en is buiten Europa niet te ontvangen. GPS is weer wel internationaal te ontvangen, maar helaas niet goed binnen gebouwen, omdat de gebruikte bouwmaterialen de hoogfrequente satellietsignalen te sterk dempen. De standaard voor tijdsynchronisatie is daarom NTP.

Eigen tijdserver?

Blijft de vraag: waarom een eigen NTP-server, als die al op het internet beschikbaar is? Daar zijn tal van openbare, vrij toegankelijke NTP-servers te vinden.

Zoals reeds in de inleiding aangegeven, zijn er situaties waarin – relatief bescheiden – moeite voor een eigen NTP-server zin heeft. Zo zijn er bijvoorbeeld netwerken die om goede redenen geen verbinding met het internet hebben. Terwijl ook in dergelijke eiland-netwerken de ermee verbonden computers en vergelijkbare apparaten over de juiste tijd willen kunnen beschikken. Het handmatig elke paar maanden moeten corrigeren van alle ‘klokken’ in de netwerkknoppunten is bepaald geen pretje.

Ook wie apparaten (en/of softwaretools) ontwikkelt die hun tijd via NTP ophalen, zal het comfort weten te waarderen van het voor testdoeleinden kunnen manipuleren van de door een NTP-server verzonden data. Ook door Elektor zijn diverse projecten gepubliceerd die NTP-gegevens nodig hebben. Bovendien is de toegang tot een lokaal LAN-adres sneller en stabieler dan een aanvraag naar buiten op het internet. Vooral wie een hele batterij verschillende microcomputers en -controllers draait, die zo synchroon mogelijk moeten zijn, zal profiteren van een eigen NTP-server.

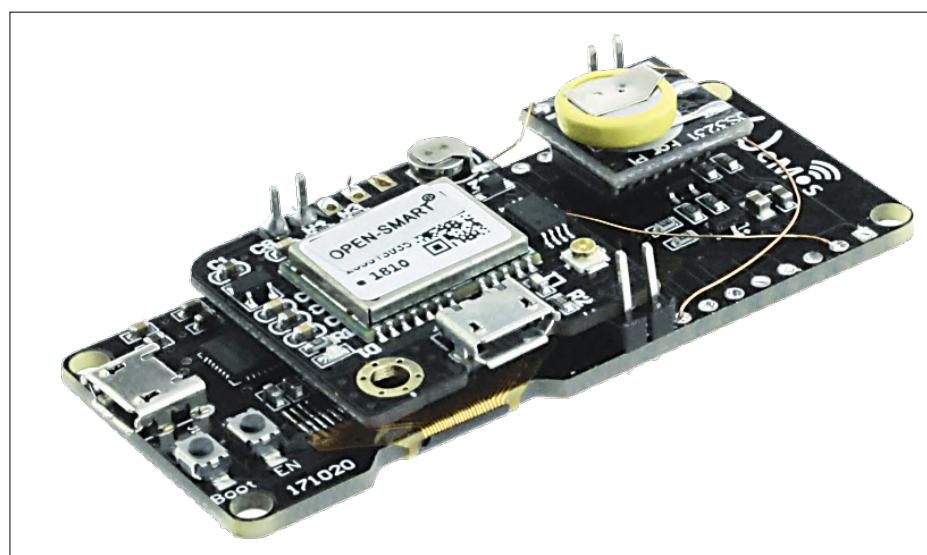
Er zijn dus nogal wat goede redenen om een eigen kleine NTP-server te bouwen en te gebruiken. De benodigde hardware is tegenwoordig goedkoop verkrijgbaar en is prima uit kant-en-klare modules samen te stellen. In dit geval is de rekenkracht van een ESP32-module meer dan voldoende. Deze module verzorgt gelijk ook de draadloze verbinding met het netwerk en kan dus probleemloos (bijna) overal worden opgesteld. De tijddata verkrijgt de eigen NTP-server in dit project via GPS – ook daarvoor zijn goedkope kant-en-klare modules beschikbaar. Dan hebben we nog een

Real Time Clock nodig voor de perioden dat GPS-ontvangst niet mogelijk of gestoord is. Om het functioneren van de server te allen tijde te kunnen monitoren, hebben we nog een klein display nodig. En tot slot hebben we nog wat software nodig waarmee de server tot leven kan worden gewekt.

Geschiedenis

De eerste versie van onze mini-NTP-server werd gebouwd voor een tentoonstelling in Frankrijk waar Elektor een stand had. Die eerste versie was opgebouwd rond een ESP8266, met als toevoegingen een RTC-chip van het type DS3231 en een klein 0,96" OLED-display. Met deze server konden de tentoongestelde Elektor-schakelingen ook de juiste tijd ophalen in perioden dat het Internet niet

ver was dus hard aan een update toe. Daarbij werd de ESP8266-module vervangen door de modernere ESP32 en een GPS-ontvanger uit de Elektor Shop toegevoegd voor automatische tijdsynchronisatie. De gebruikte ESP32-module was al met een OLED-display uitgerust. Dit display reageert niet zoals gebruikelijk op I²C-adres 0x78, maar op 0x7A, waardoor zelfs een tweede, extern aangesloten display op 0x78 mogelijk was. In de gebruikte library U8G2 kunnen de display-adressen worden ingesteld. Op het ingebouwde display wordt dan tijd en datum in GMT afgebeeld, evenals het IP-adres van de NTP-server en de actuele status van de ESP32 (Client of Accesspoint). Het tweede display is verantwoordelijk voor de lokale tijd en de GPS-positie. Als RTC werd een in het



Figuur 2. De zwevend bedrade sandwich bestaande uit ESP32-module met geïntegreerd OLED-display, RTC en GPS-module.

beschikbaar was, of de verbinding ermee te slecht. Het prototype deed ook dienst tijdens Electronica 2018 in München. In het lab werd de schakeling gebruikt als tijdbron voor enkele Elektor-klokken en RPi-projecten. Vooral de mogelijkheid eenvoudig de NTP-data te kunnen manipuleren bleek erg handig.

Die eerste server-versie was uitgerust met een RTC-chip die de voor een DS3231 vigerende toleranties duidelijk overschreed. Zijn afwijking begon al snel te irriteren, omdat de klok elke maand opnieuw moest worden ingesteld. Dat is voor een korte beursperiode weliswaar niet zo'n probleem, maar in een laboratorium al snel onacceptabel. Die NTP-ser-

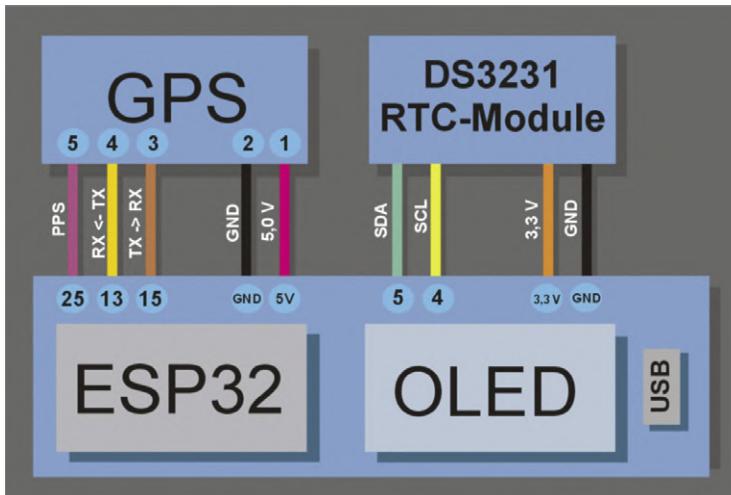
lab voorhanden zijnde DS3231-module gebruikt. En daarmee was de hardware-update klaar.

Module

Voor de nieuwe NTP-server moeten dus verschillende units worden samengevoegd, zoals te zien in **figuur 1**. De ESP32 is het hart van de schakeling, en daar komen een OLED-display met SSD1306-controller, een RTC en een GPS-module bij. In totaal vier eenheden. Het kan nog wat compacter, als u net als wij een ESP32-module met on-board OLED-display toepast. Samen met de DS3231-RTC en de GPS-module krijgen we dan een com-



Figuur 3. Alles van figuur 2 ingebouwd in een met een 3D-printer vervaardigde behuizing.



Figuur 4. Zo worden de drie modules elektrisch met elkaar verbonden.

Interrupt “Pulse Per Second”

Voor de PPS-interrupt (Pulse Per Second) gebruikt de software GPIO25. Om te voorkomen dat de pin een ongedefinieerde toestand aanneemt, wordt de interne pullup-weerstand ingeschakeld. De interrupt wordt door een stijgende flank geactiveerd, wat de functie `handlePPSInterrupt` aanroeft. Deze functie moet het attribuut `IRAM_ATTR` bezitten – de code moet dus in het RAM van de ESP32 vorhanden zijn. Niet in het RAM gekopieerde code zou door het uitlezen van de externe flash vertraagd worden, of erger: uitgeschakelde flash zou resulteren in een exception en een reboot van de ESP32. Het gebruik van variabelen van het type `float` is overigens in interrupts problematisch, omdat daarvoor de floating-point unit benut zal worden, wat in het actuele FreeRTOS tot problemen en onvoorspelbare resultaten na de interrupt kan leiden. Aan het einde van de routine komt `xSemaphoreGiveFromISR` voor het updaten van het OLED-display.

pakte sandwich (**figuur 2**), waarvoor een passende 3D-geprinte behuizing is gemaakt (**figuur 3**). Zo krijgen we een kleine maar vooral mobiele NTP-server, die overal kan worden ingezet waar hij verbinding met het eigen WLAN heeft en GPS-signalen kan ontvangen.

Voor de I²C-aansturing van de RTC worden niet de gebruikelijke pinnen van de ESP32-module gebruikt. Als je een RTC via I²C wilt aansluiten, dan ligt SCL in plaats daarvan aan GPIO4 en SDA aan GPIO5. De GPS-module maakt gebruik van een normale seriële verbinding. Hier fungeert GPIO13 als RX-pin en GPIO15 als TX-pin. De seriële verbinding vereist gekruiste signalen: ESP32-RX moet dus worden verbonden met GPS-TX en ESP-TX met GPS-RX. Beide modules moeten met 5 V worden gevoed, welke spanning en massa direct van de ESP32-module worden betrokken. Bij de GPS-module moet nog het signaal met de secondepuls worden verbonden met de corresponderende ingang van de ESP32-module (pin 25). **Figuur 4** toont de benodigde bedrading.

Om te zien met welke problemen we bij de nog te schrijven software te maken kunnen krijgen, hebben we eerst de gratis verkrijgbare code *ESPNTPServer* [1] gebruikt. Voor de ‘juiste’ NTP-server-software werden enige codebestanddelen van andere Elektor-projecten gerecycled – in het bijzonder de tijdcorrectiefuncties van de Flipper-klok [2], inclusief de mogelijkheid om met verschillende bronnen met verschillende prioriteiten te synchroniseren.

Als RTC kan zowel de DS3231 als de DS1307 worden gebruikt. Na het booten wordt op I²C-adres 0x68 naar een RTC gezocht. Als daar een bouwsteen reageert wordt aangenomen dat het een DS3231 is. Bij een DS1307 moet de code opnieuw worden gecompileerd, en in *Firmware.ino* de regel `RTC_DS3231 rtc_clock;` worden vervangen door `RTC_DS1307 rtc_clock;`.

De RTC stelt de tijd ter beschikking, ook als nog geen GPS-signaal is ontvangen of als dit uitvalt. Ook als de ESP32 geen RTC-module vindt loopt de server, maar nu zonder de tijd van de RTC-module. De nood-klok van de ESP32 start simpelweg op 01-01-1970 middernacht, totdat voor de eerste keer GPS-data binnenkomen. De server loopt dus ook zonder RTC. Ook zonder GPS-module loopt de NTP-server in de ESP32. Maar dan moet de tijd steeds weer handmatig worden

ingesteld. Dat hiervoor de NTP-server via een configuratie-website bereikbaar moet zijn spreekt voor zich.

NTP-Data

NTP maakt gebruik van UDP (User Datagram Protocol) voor de informatie-overdracht. De NTP-server luistert eenvoudig naar poort 123 naar inkomende NTP-aanvragen en beantwoordt deze met een gestandaardiseerd datapakket van 56 bytes lang in het formaat NTPv4 (RFC 5905). Het bijbehorende datagram van **figuur 5** toont welke data vereist zijn. LI, VN en MODE zijn bitvelden: LI geeft aan of er aan het einde van de minuut een schrikkelseconde komt, VN geeft de gebruikte protocolversie aan en MODE staat voor de modus van het systeem (hier ‘server’).

Root Delay heeft betrekking op de afstand tussen deze en de volgende server. Omdat wij natuurlijk de enige in de (lokale) wereld zijn, bevat dit veld de waarde ‘1’. Ook bij *Stratum* moet een ‘1’ staan, omdat er naast deze geen andere NTP-servers zijn. Dezelfde waarde komt ook in het veld *Root Dispersion*. Schakelseconden worden genegeerd. Een *Reference Identifier* met de waarde ‘PPS’ (Puls Per Second) geeft aan, dat met een impulsfrequentie van 1 Hz wordt gerekend.

Alle tijden beginnen op 01-01-1900 en daarmee 70 jaar of 2208988800 seconden voor de start van de in de tijd kern gebruikte Unixtime. Elke tijdstempel bezit twee 32-bit velden – één voor de seconden van de actuele tijd en één voor het fractionele deel. De *Reference Timestamp* moet de actuele tijd bevatten. De server werkt uitsluitend met hele seconden en daarom bevat de *Reference Timestamp Fraction* een ‘0’. Dat geldt eveneens voor de tijdstempels voor ontvangen en verzenden. In *Precision* komt

$\log_2(\text{Clock Jitter})$. De jitter zelf is het resultaat van de aanroepijd van de functie voor het uitlezen van de huidige tijd plus één seconde, die we krijgen door het op nul gezette ontbrekende fractionele deel. Deze oplossing is niet perfect, maar voor hier afdoende.

Tijdbronnen

Zoals eerder gesteld, werkt de NTP-server met twee externe tijdbronnen. Eén daarvan is een RTC en de andere is het GPS-signalen met zijn secondepuls. Bij instabiele ontvangst kan dit GPS-signalen ook af en toe wegvalLEN. Het signaal activeert een interrupt en wordt vervolgens gebruikt als het beschikbaar is. In dit geval leidt dit tot het omschakelen van de interne secondentelling naar de externe via GPS. Bovendien wordt de GPS-secondepuls bewaakt. Als de puls langer dan 1400 ms op zich laat wachten, wordt aangenomen dat de ontvangst is weggevalLEN. In dat geval wordt overgeschakeld op de interne tijdbron. Dit betekent dat een seconde verder geschakeld wordt en dit 600 ms later moet worden herhaald, omdat de software 400 ms na de verwachte puls heeft omgeschakeld.

Bij GPS-ontvangst levert de GPS-module overeenkomstige tijdsgegevens. Om te synchroniseren met de exacte wereldtijd, is het voldoende om deze GPS-tijdsgegevens slechts om de 10 minuten te verwerken en de interne (of RTC-klok) hiermee te synchroniseren. Zolang de GPS-module ontvangt, levert deze een precieze secondepuls af, die de interne klok van de ESP32 een seconde verder zet.

Voor bepaalde doeleinden kan de synchronisatie met de GPS-tijd worden gedeactiveerd. Dan wordt alleen nog de secondepuls gebruikt. Zo kan worden getest hoe systemen in het LAN zich gedragen onder bijzondere omstandigheden, zoals de overgang van zomer-naar winter tijd.



Figuur 5. Datagram van een NTP-datapakket.

heden, zoals de overgang van zomer-naar winter tijd.

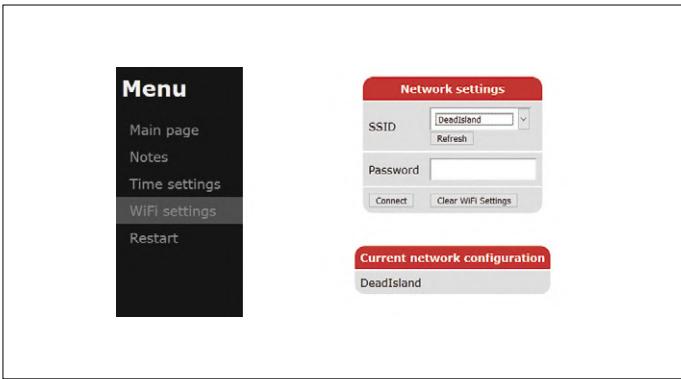
Software

Hier komen een aantal eigenaardigheden van de ESP32 aan het licht. Het actualiseren van het display wordt in een eigen taak in een eindelozelus gerealized, zonder dat dit de ESP32 blokkeert. Door interprocesscommunicatie wordt de taak meegedeeld, dat hij alleen één keer per seconde de inhoud van de twee displays hoeft te berekenen. De taak is dus inactief totdat een semafoor wordt geactiveerd – in de tussentijd heeft hij geen actieve rekentijd nodig.

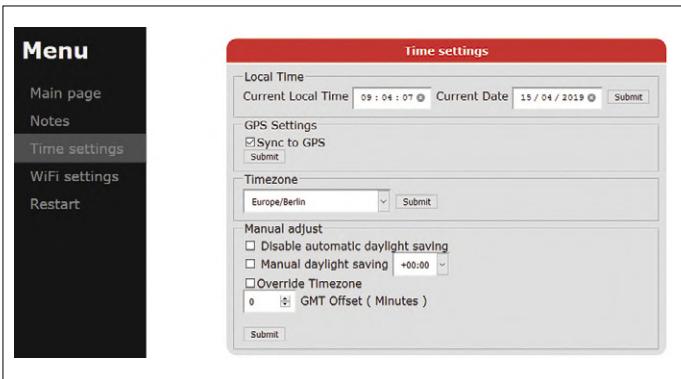
Ook wordt door een Mutex verhindert dat de lus of de displaytaak ongecontroleerd de I2C-bus benadert. Deze functies zijn beschikbaar in het Real Time Operating System FreeRTOS, dat normalerwijze

Weblinks

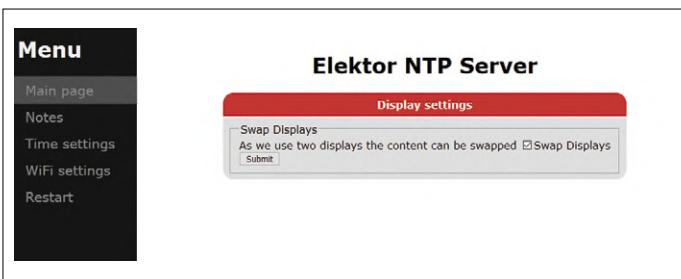
- [1] ESPNTPServer: <https://github.com/liebman/ESPNTPServer>
- [2] Flipperklok, Elektorlabs maart/april 2019: www.elektormagazine.nl/180307-03
- [3] Projectpagina bij dit artikel: www.elektormagazine.nl/180662-03
- [4] Firmware op GitHub: <https://github.com/ElektorLabs/180662-mini-NTP-ESP32>
- [5] Arduino voor ESP32 voorbereiden: https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards_manager.md
- [6] Arduino ESP32-Filesystem-Uploader: <https://github.com/me-no-dev/arduino-esp32fs-plugin>
- [7] Handleiding voor het uploaden van bestanden: <https://techtutorialsx.com/2018/08/24/esp32-arduino-spiffs-file-upload-ide-plugin/>



Figuur 6. Menu van de website voor de WLAN-instellingen.



Figuur 7. Menu voor de tijd-instellingen.



Figuur 8. Menu voor de monitor-weergave.

door het Arduino-framework verborgen wordt gehouden. Een beschrijving van FreeRTOS, zijn mogelijkheden en struikelblokken ligt buiten het bestek van dit artikel, maar een specifiek hieraan gewijd artikel is in voorbereiding.

De benodigde firmware voor de ESP32 is gratis op de bij dit project behorende Elektor-webpagina beschikbaar [3]. Op deze webpagina worden ook eventuele updates gepubliceerd. Vanzelfsprekend zijn deze ook direct op GitHub te vinden [4]. Daar zijn ook nieuwe preview-versies of modificaties van de originele versie te vinden.

Begin met het installeren van de nieuwste Arduino-IDE op uw PC en vervolgens het pakket voor het ESP32-board [5]. Om

de code van de firmware zelf te kunnen compileren, moeten aanvullend nog de volgende libraries worden geïnstalleerd:

- U8G2
- Time
- Ticker
- TinyGPS++
- RTCLib
- ArduinoJson 6.x
- CRC32

In eventuele nieuwe versies van het project kan deze lijst mogelijk veranderen. Nadat de gecompileerde code naar de ESP32-module is overgedragen, moet ook nog de bijbehorende website worden geladen. Hoe dat gaat, is op het internet en in enkele Elektor-projecten beschreven waarin de ESP32 is toegepast. De Arduino *ESP32-Filesystem-Uploader* en nadere info is bij GitHub onder [6] te vinden, een handleiding voor het uploaden van bestanden (een website is niets anders dan een bestand) onder [7]. Na een restart en daaropvolgend boot-proces staat de nieuwe NTP-server ter beschikking.

Server-configuratie

De inrichting van de server is heel eenvoudig. Na het succesvol uploaden van code en website start de server eerst als Access-Point. In deze modus kan met de server, resp. zijn website, verbinding worden gemaakt op het lokale adres 192.168.4.1. Vervolgens moeten SSID en wachtwoord worden ingevoerd (**figuur 6**). Via de web-interface kunnen ook de tijdrelevante instellingen worden aangepast.

Nadat deze instellingen actief zijn geworden, probeert de server een verbinding met het WLAN tot stand te brengen en via DHCP een lokaal IP-adres te verkrijgen.

Aansluitend kunnen via de web-interface verdere functies worden geconfigureerd (**figuur 7**). Onder de menuoptie *Main Page* kan de inhoud van twee display worden verwisseld. Als alleen een monitor aangesloten is, kan hier worden gekozen welke inhoud moet worden afgebeeld. Daarmee is de configuratie voltooid. Uw hoogsteigen Mini-NTP-server kan vanaf nu inkommende NTB-aanvragen afhandelen. ▶

180662-03

IN DE STORE



→ OPEN SMART GPS
www.elektor.nl/18733

- Lolin ESP32 OLED
www.elektor.nl/18575
- Adalogger FeatherWing - RTC + SD
www.elektor.nl/18692
- 0,96" I²C OLED (optioneel)
www.elektor.nl/18747

Het SCCC-project (3)

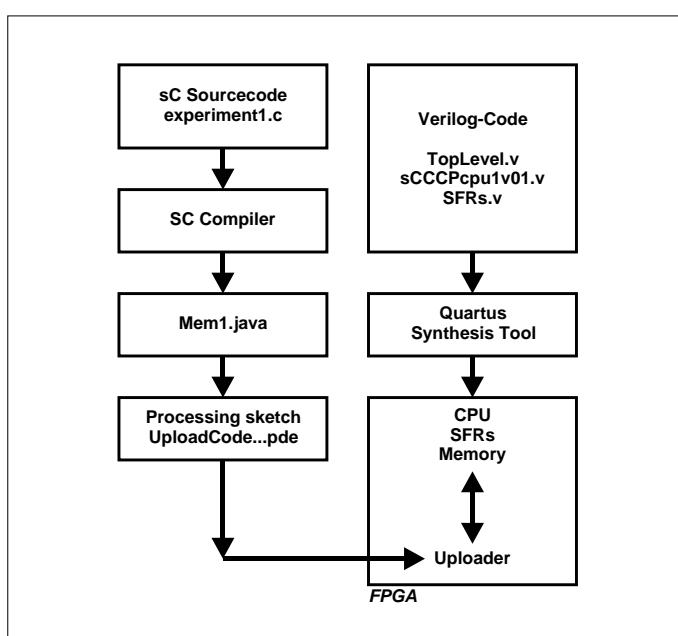
softcore-processor en C-compiler voor zelfbouw

Martin Oßmann (Duitsland)

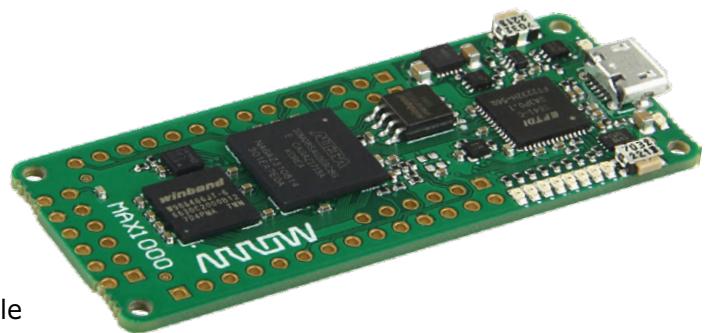
In het vorige deel van deze serie hebben we ons beperkt tot het simuleren van onze zelfgebouwde CPU. Dit keer gaan we weer aan de slag met het MAX1000-board. We verbinden de kleine print via één of twee seriële interfaces met de PC om programma's te uploaden en seriële in- en uitvoer mogelijk te maken. Verder gaan we de versnellingssensor op de kaart uitlezen.

In het eerste deel van de serie [1] hebben we de MAX1000-kaart al gebruikt. We willen nu in het kort bekijken hoe we een gecompileerd programma in de FPGA krijgen. Het compileren van het C-programma gaat vrij snel. Daarbij ontstaan twee bestanden: *codemem.txt* en *datamem.txt*. De inhoud van die bestanden moet nu in het geheugen van de CPU, die zich in de FPGA bevindt, worden geladen.

Dat is het doel van de volgende regels in het Verilog-codebestand *sCCCPcpu1v01.v* (dat de CPU definieert):



Figuur 1: Met de uploader kunnen we gecompileerde programma's van de gebruiker ook na de synthese van de CPU naar de FPGA sturen. Dat scheelt veel ontwikkeltijd!



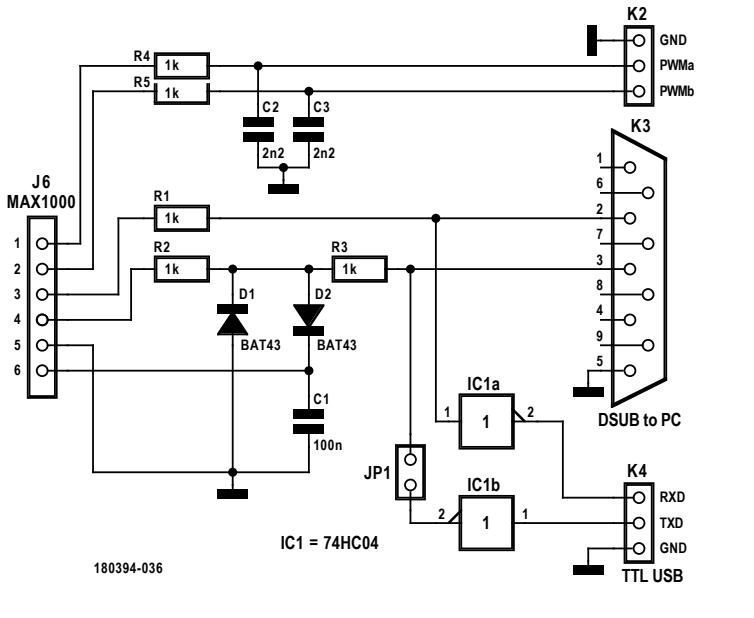
```

parameter sCcodeMemAddressWidth=12 ;
parameter sCdataMemAddressWidth=10 ;
reg [40-1:0] sCcodeMem [0:(1 <<
    sCcodeMemAddressWidth)-1] ;
reg [32-1:0] sCdataMem [0:(1 <<
    sCdataMemAddressWidth)-1] ;
initial begin
    $readmemh("CodeMem.txt",sCcodeMem) ;
    $readmemh("DataMem.txt",sCdataMem) ;
end
  
```

In de eerste vier regels worden de woordbreedte en het aantal woorden van het programma- en datageheugen vastgelegd. De regels na *initial begin* geven aan, dat het programma- en het datageheugen moeten worden geïnitialiseerd met de inhoud van deze bestanden. Om dat mogelijk te maken, moeten deze bestanden al beschikbaar zijn op het moment van het synthetiseren van de FPGA. We synthetiseren de FPGA dus na het compileren van het C-programma. Dan krijgen we een bitfile dat we in de FPGA laden. Deze stap kost vrij veel tijd (bij de auteur duurt het ongeveer vijf minuten). Na elke verandering in het C-programma moet de synthese opnieuw worden uitgevoerd.

Dat is heel vervelend en het zou veel prettiger zijn, als we het programma- en datageheugen pas na de synthesestap in de FPGA zouden kunnen laden. Gelukkig is dat mogelijk door een soort uploader in de FPGA te programmeren.

Laten we dat eens proberen. In de werkmap van *experiment5* (te downloaden onder [3]) staat een Quartus-project (.qpf). Om de uploader te kunnen gebruiken, moet u dit project synthetiseren en de code in de FPGA laden. De workflow is te zien in **figuur 1**. Voor het naar de FPGA verzenden van de data en het programma gaan we een seriële interface gebruiken. Daarover eerst een paar opmerkingen.



Figuur 2: Hardware-uitbreiding voor de service-interface voor het uploaden van programma's. De PWM-uitvoer hebben we voorlopig nog niet nodig.

Onze eerste hardware-uitbreiding

We gebruiken de seriële interfaces standaard in de modus 115200 8N1. Om de seriële interface van de PC te verbinden met de MAX1000-kaart, gebruiken we de hardware in **figuur 2**. Als de PC een echte seriële interface heeft, dan kunnen we die verbinden met de 9-polige connector. IC1 kan dan worden weggeleggen (verwijder jumper JP1). Dit is hoe de auteur zijn seriële poorten gebruikt. Het is niet volgens de norm, maar het werkt goed.

Misschien hebben veel lezers een seriële USB-TTL-kabel, waarvan de signalen geïnverteerd zijn in vergelijking met normale RS232-spanningen. Zo'n kabel kan worden aangesloten op K4. Wie het nog eenvoudiger wil hebben, kan de USB-TTL-kabel ook rechtstreeks met de FPGA verbinden, maar dan moet er wel een invertering voor de beide signalen in de FPGA worden ingebouwd.

Seriële service-poort

Voor het uploaden van toepassingsprogramma's via de 'service-poort' wordt een klein programma op de ontwikkel-PC gebruikt. De auteur heeft dit programma gerealiseerd met Processing [4]. Dit programma is te vinden in de download [3] onder *Processing\UploadCodeViaSerial1v01*, u kunt het rechtstreeks starten in de Processing-ontwikkelomgeving. Maar eerst moet u nog de regel

```
String serialPort="COM3";
```

aanpassen voor de COM-poort, die u gebruikt voor de communicatie met de hardware.

Om te testen, of de service-interface werkt, kunnen we

1234x

naar de poort sturen (bijvoorbeeld met een terminalpro-

gramma). Als antwoord krijgen we dan een reeks hexadecimale karakters

```
000000001234 00000000000000 1234
```

waarbij de verstuurde hexadecimale data ook weer wordt weergegeven.

In de directory *Processing\UploadCodeViaSerial1v01* staat ook nog een bestand *Mem1.java*, als we bij het starten van de SCCP-ontwikkelomgeving de uitvoeroptie -p (voor Processing) hebben gekozen. Dit bestand bevat straks de objectcode van ons gebruikersprogramma en wordt door de Processing-omgeving gebruikt om de eigenlijke upload uit te voeren.

Periferie aansluiten

De CPU die we gebruiken bij Experiment 5 heeft niet alleen de service-UART aan boord, maar is ook nog uitgerust met andere functionele blokken. Dat zijn onder meer een teller (RTC), een pulsbreedtemodulator (PWM), een A/D-converter, een GPIO-interface en last but not least een UART voor de gebruiker. De UART voor gebruiker wordt hardwarematig verbonden met de PMOD2-inter-

face op de MAX1000-kaart (J6), zoals te zien is in **figuur 3**. We gaan nu kijken, hoe we de CPU in de FPGA kunnen uitbreiden met eenvoudige periferieblokken.

Om te beginnen willen we drie pennen gebruiken als General Purpose IO-pennen (GPIO's).

In Verilog gaat dat als volgt:

```
parameter GPIOINWIDTH = 3 ;
wire[GPIOINWIDTH-1:0] GPIOinp ;
parameter GPIOOUTWIDTH = 3 ;
reg[GPIOOUTWIDTH-1:0] GPIOoutp ;
```

Voor het lezen van data uit uitbreidingen laten we de CPU het commando **INPA c** uitvoeren, waarbij **c** (van Channel) het nummer is van de poort waaronder we de CPU-periferie in de FPGA hebben gerealiseerd.

Het mechanisme wordt in Verilog afgebeeld met de variabele **inpA**, waarvan de waarde door het verwerken van het bovenstaande commando in register R0 wordt geplaatst. Welke data daar naartoe gaat, wordt vastgelegd door een grote 'multiplexer', die als volgt is geprogrammeerd:

```
assign inpA=
  (inpAchannel==0) ? { 32'h12341234 } :
  (inpAchannel==1) ? sCuartRXdata :
  (inpAchannel==2) ? sCTXuartBitsLeft :
  (inpAchannel==3) ?{ 20'h12300 , ADCval[12-1:0] } :
  (inpAchannel==4) ? RTCTimer :
  (inpAchannel==5) ? debugReg1 :
  (inpAchannel==6) ? aDacSignal :
  (inpAchannel==7) ? bDacSignal :
  (inpAchannel==18) ? GPIOinp :
  32'h12341111 ;
```

Via kanaal 0 krijgen we bijvoorbeeld altijd de waarde '12341234' terug (als testdata). Via kanaal 1 krijgen we de ontvangen data van de gebruikers-UART. En op kanaal 18 zitten de I/O-pennen, die we in Experiment 5 willen gebruiken.

In de andere richting werkt het op een vergelijkbare manier. Als we data uitvoeren naar kanaal c met het commando `OUTA c`, dan wordt het signaal `outAstrobe` geactiveerd en de data gaat door een grote if-then-else-structuur naar de juiste plaats. In Verilog ziet dat er als volgt uit:

```
always @(posedge sCclk) begin
  if(outAstrobe) begin
    if( outAchannel==0) begin
      ...
    else if( outAchannel==5) begin
      LEDs<=outA[8-1:0] ;
    end
    ...
  else if( outAchannel==18) begin
    GPIOoutp<=outA[GPIOOUTPWDTH-1:0] ;
  end
  ...
end
end
```

We zien hier bijvoorbeeld dat de LED's aangesloten zijn op kanaal 5 , dat we in Experiment 1 hebben gebruikt. Via kanaal 18 worden onze I/O-pennen aangestuurd. Nu moeten we deze pennen nog koppelen aan de echte hardware-pennen.

Experiment 5: GPIO en software-SPI

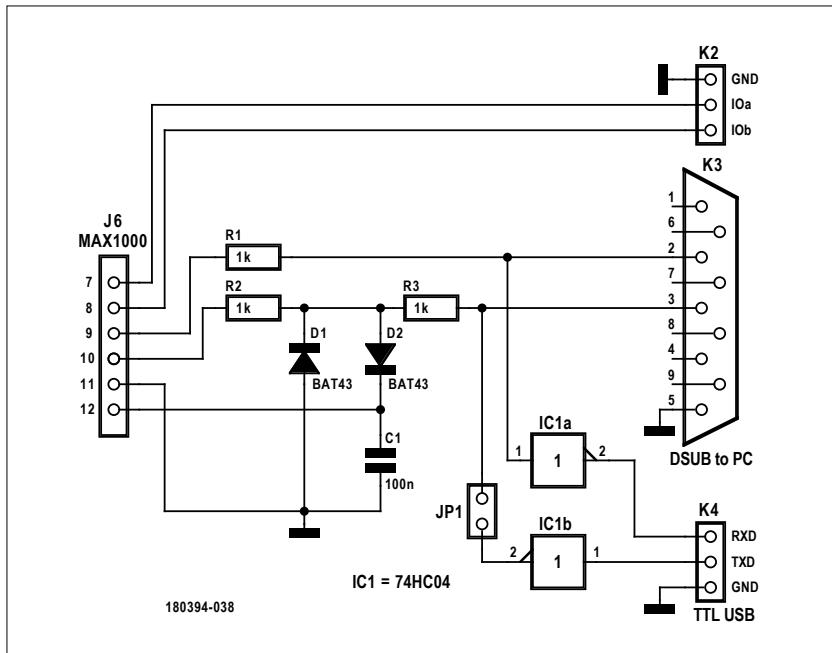
Op de MAX1000-kaart zit een versnellingssensor van het type LIS3DH van ST. Die willen we nu benaderen met onze CPU. We voorzien onze CPU daarom, zoals besproken, van General Purpose I/O-pennen, die we koppelen aan de pennen van de LIS3DH. Om te beginnen worden de fysieke pennen in de Verilog-code van de Toplevel-module gedefinieerd als in- en uitgangspennen:

```
...
input LISsenINT1,
input LISsenINT2,
input LISsenSDO,
output LISsenCS,
output LISsenSPC,
output LISsenSDI ...

```

Met de juiste `assign`-instructies verbinden we dan intern in de FPGA de LIS3DH-pennen met de I/O-pennen van onze DIY-CPU (listing 18):

```
assign GPIOinp[0] = LISsenINT1 ;
assign GPIOinp[1] = LISsenINT2 ;
assign GPIOinp[2] = LISsenSDO ;
assign LISsenCS = GPIOoutp[0] ;
assign LISsenSDI = GPIOoutp[1] ;
assign LISsenSPC = GPIOoutp[2] ;
```



Figuur 3: Met deze uitbreiding kunnen we eenvoudige I/O-experimenten uitvoeren.

Deze beide gedeelten van de Verilog-code vertellen de softcore-CPU hoe hij kan lezen en schrijven via INPA- resp. OUTA-commando's. Hier kunnen we de al beschikbare periferie van de CPU uitbreiden met eigen periferie-eenheden.

Nu moeten we nog toelichten hoe u in uw eigen C-programma's de periferie van de CPU kunt aanspreken.

Het geheel in C

We hebben al gezien, dat we in onze C-programma's ook assemblercode kunnen gebruiken. Dat maakt het aanspreken van de CPU-periferie (en daarmee van de externe hardware) heel gemakkelijk. Het voorbeeld in **listing 1** komt uit het bestand *experiment5/IOLib5.c*.

Om dit te begrijpen moet u weten hoe de compiler werkt: na het evalueren van een rekenkundige uitdrukking staat de uitkomst in R0. En wat er in R0 staat, werkt als retourwaarde van functieroutines. Met die kennis en met het commentaar in de code wordt de werking duidelijk. Nu kunnen we gebruik maken van C-functies, waarmee we de LIS3DG-versnellingsensor kunnen aansturen. Voor de programmeur ziet het eruit als een SPI-interface, geïmplementeerd door bit-banging van de universele I/O-lijnen. In C ziet dat eruit zoals in **listing 2**. Dan kunnen we nu beginnen met het eigenlijke experiment.

Actie: Go

De SCCC-ontwikkelomgeving wordt opgestart door *c.bat* in *experiment5/experiment* uit te voeren. Druk daarna op knop *MAKE* om de C-software te compileren.

c.bat:

```
Java
-jar c:\SCCCP\CompilerAsJar\sCCCP.jar
-s experiment5.c
-p ..\..\Processing\UploadCodeViaSerial1v01\
```

Listing 1: Periferiedata lezen en schrijven.

```
GPIOchannel EQU 18

GPIOout(int p){
    p ; // waarde van p wordt "berekend" en naar R0 gebracht
    #asm // nu verder in assembler
    OUTA GPIOchannel // waarde in R0 naar het GPIOchannel uitvoeren
    #endasm
}

GPIOin(){
    #asm // omschakelen naar assembler
    INPA GPIOchannel // waarde van GPIOchannel naar R0
    #endasm // omschakelen naar C
}
```

Met de optie `-p ...` geven we aan, dat de gegenereerde object-code als `mem1.java` in de upload-directory moet worden geschreven.

Daarna synthetiseren we het Quartus-project `experiment5.qpf` in de werkdirctory en laden we de bijbehorende bitfile in de FPGA. Vanaf nu kunnen we gebruik maken van de uploader om via de service-interface het data- en programmageheugen te laden. We kunnen dat testen door met een terminalprogramma een 'x' naar de service-interface te sturen. Er moet dan een reeks van hexadecimale tekens als antwoord komen.

Voor het uploaden maken we gebruik van de sketch

`UploadCodeViaSerial1v01.pde`,

die werkt in de Processing-programmeeromgeving.

Nu kunnen we (in de Processing-omgeving) de eigenlijke upload starten. Daarna is ons toepassingsprogramma in de FPGA geladen. Dit levert de waarden van de drie kanalen van de versnellingssensor LISD3 via de gebruikersinterface (met 115200 Baud). Dat ziet er als volgt uit:

```
x= -00000032 y= -00000128 z= +00016576
x= -00000048 y= -00000224 z= +00016688
```

```
x= -00000064 y= -00000240 z= +00016544
x= -00000032 y= -00000224 z= +00016640
x= +00000016 y= -00000256 z= +00016704
```

Als we de MAX1000-kaart draaien en kantelen, is te zien hoe de versnellingswaarden veranderen.

Als we nu het programma voor de CPU willen aanpassen, hoeven we alleen nog het nieuwe programma te compileren en het met een upload naar de MAX1000 te sturen. Dat gaat veel sneller dan wanneer we elke keer de CPU opnieuw zouden moeten synthetiseren. Als het programma uiteindelijk goed werkt, kunnen we in het bestand `MainInclude.txt` in het Verilog CPU-project de upload weer deactiveren, door de regels

```
`define bootload
//`define standalone
```

te vervangen door:

```
//`define bootload
`define standalone
```

Daardoor wordt de inhoud van het data- en programmageheugen weer meteen bij het synthetiseren ingevoegd.

Ons Experiment 5 is één van de ingewikkelder experimenten, omdat er een heleboel bestanden, programma's en componenten aan te pas komen. Als er iets niet lukt, controleer dan eerst de volgende dingen: zijn alle padnamen in de batch-bestanden aangepast aan de installatie op uw eigen PC? Zijn bij alle seriële interfaces de COM-poort-nummers en de baudrate goed ingesteld?

Als alles is gelukt, kunnen we ook eens een blik op het pro-

IN DE STORE



→ MAX1000 FPGA-ontwikkelkaart
www.elektor.nl/max1000

→ E-boek (Engelstalig):
“Microprocessor Design Using Verilog HDL”
www.elektor.nl/verilog

Weblinks

- [1] Het SCCC-project (1), Elektorlabs maart/april-2019: www.elektrormagazine.nl/180394-03
- [2] Het SCCC-project (2), Elektorlabs mei/juni-2019: www.elektrormagazine.nl/180394-B-03
- [3] Projectpagina bij dit artikel: www.elektrormagazine.nl/180394-C-03
- [4] Processing: <http://www.processing.org>

Listing 2: SPI-communicatie met behulp van bit-banging.

```
int GPIOoutShadow ;
#define bitMaskLISsenINT1 1
#define bitMaskLISsenINT2 2
#define bitMaskLISsenSD0  4

#define bitMaskLISsenCS   1
#define bitMaskLISsenSDI  2
#define bitMaskLISsenSPC  4

LISinit(){
    GPIOoutShadow= bitMaskLISsenCS | bitMaskLISsenSPC | bitMaskLISsenSDI ;
    GPIOout(GPIOoutShadow) ;
}

LSIclockPulse(){
    GPIOoutShadow = GPIOoutShadow & (~bitMaskLISsenSPC) ;
    GPIOout(GPIOoutShadow) ;
    GPIOoutShadow = GPIOoutShadow | bitMaskLISsenSPC ;
    GPIOout(GPIOoutShadow) ;
}

SPItxrxByteMSBfirst(int vTX){
    int vRX ;
    int k ;
    int GPIOinp ;
    vRX=0 ;
    for( k=0 ; k<8 ; k++){
        if(vTX & 0x80) {
            GPIOoutShadow =GPIOoutShadow | bitMaskLISsenSDI ;
        }
        else {
            GPIOoutShadow =GPIOoutShadow &(~ bitMaskLISsenSDI ) ;
        }
        LSIclockPulse() ;
        vRX = vRX << 1 ;
        GPIOinp=GPIOin() ;
        if( GPIOinp & bitMaskLISsenSD0 ) { vRX = vRX |1 ; }
        vTX=vTX<<1 ;
    }
    return vRX ;
}
```

gramma *experiment5.c* werpen. Dit programma gebruikt namelijk de gebruikers-UART voor de uitvoer, en zo kunnen we leren hoe dat in zijn werk gaat.

De versnellingssensor wordt in de code met bit-banging aangestuurd. We hadden de SPI-functionaliteit ook naar de FPGA kunnen verplaatsen. Maar bij onze eenvoudige en niet-tijdkritische toepassing was de aanpak met bit-banging het gemakkelijkst te realiseren.

In het volgende deel van de serie gaan we ons bezighouden met de uitvoer van analoge signalen. We gaan dan werken met sigma/delta-DAC's en interrupts. Tot dan! ◀

(180394-C03)

Van buizen naar silicium

met een interessante tussenstap

Dr. ir. Martin Beusekamp

De mensheid doet inmiddels royaal een eeuw aan elektronica (1907: uitvinding van de triode door Lee de Forest). Ruw afgerond hebben elektronici in de eerste halve eeuw van die periode alleen met elektronenbuizen gewerkt en in de laatste halve eeuw nagenoeg uitsluitend met silicium halfgeleiders in discrete of geïntegreerde vorm. In een betrekkelijk korte overgangsperiode zijn elektronische apparaten ontworpen en gebouwd die buizen noch silicium halfgeleiders bevatten, maar seleniumgelijkrichters, germaniumdioden en/of -transistoren. Laten we die componenten eens aan een nadere beschouwing onderwerpen.

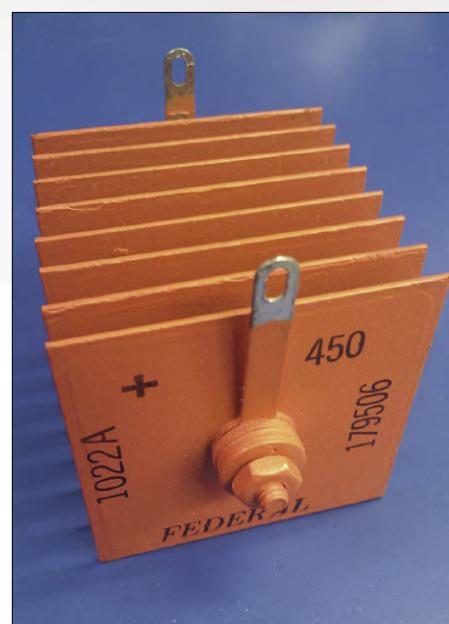
Selenium gelijkrichters

Figuur 1 toont een typische seleniumgelijkrichter, bestaande uit een aantal metalen plaatjes (staal of aluminium), bekleed met een dunne (ca. 1 µm) laag bismut (Bi) of nikkel (Ni) met daarop een dikker (ca. 60 µm) laag selenium (Se). Zo'n sandwich heeft de eigenschap dat elektronen de barrière tussen selenium en staal/aluminium in de ene richting makkelijker kunnen oversteken dan in de andere richting en werkt dus als een diode.

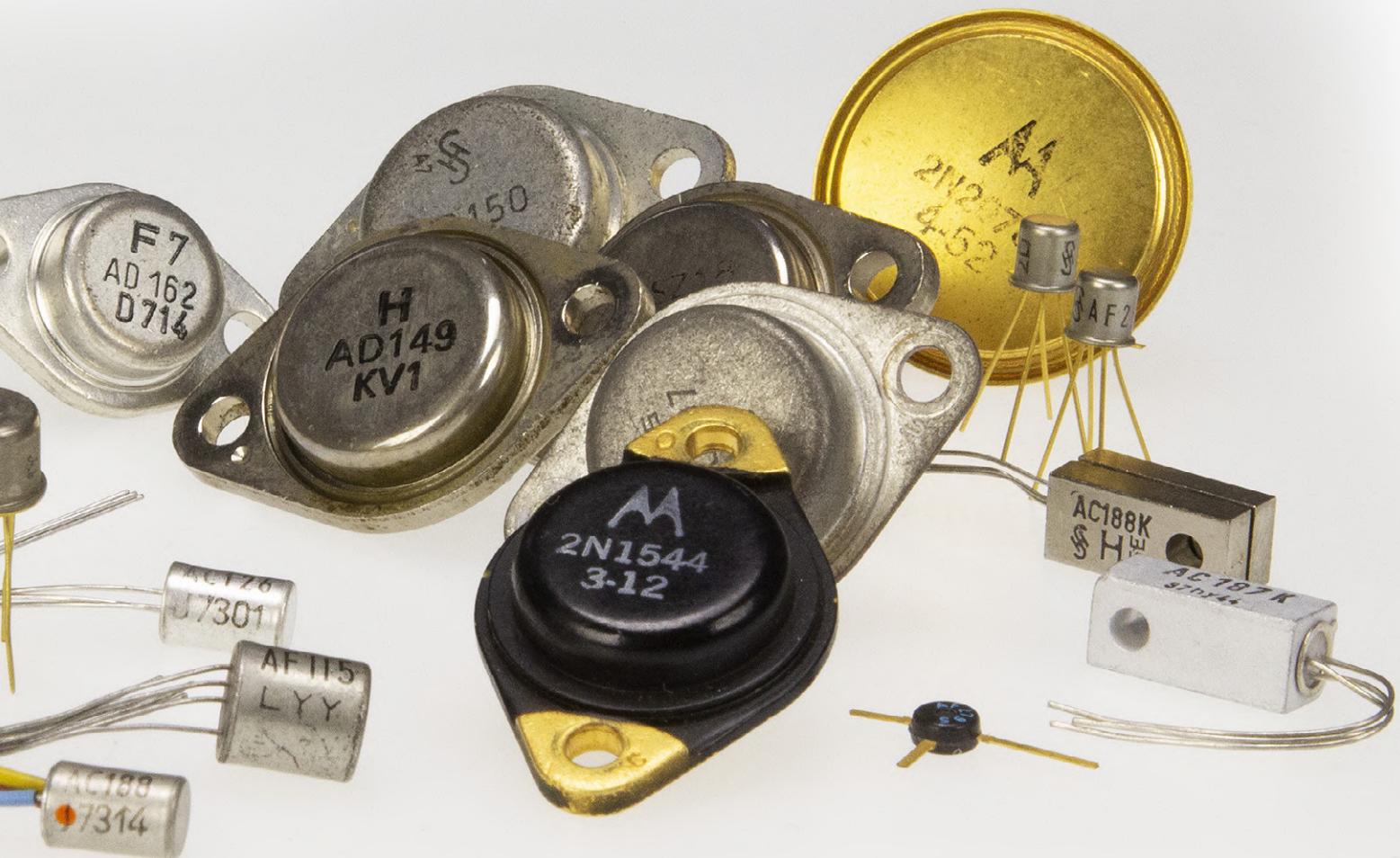
Ieder van die dioden heeft een sperspanning van ongeveer 30 - 50 volt, maar voor de veiligheid werd over het algemeen een maximum van 20 volt per diode aangehouden. De hele gelijkrichter uit figuur 1 bestaat uit acht plaatjes en heeft in de praktijk dus een maximale sperspanning van 160 volt. De maximale stroom door de gelijkrichter is recht evenredig met het oppervlak van

de metalen plaatjes en bedraagt in de praktijk ongeveer 25 - 30 mA/cm² [1]. Leuk, toch? Voor de eigenschappen van een selenium gelijkrichter hoeft je dus geen typenummer op te zoeken. Plaatjes tellen voor de sperspanning, plaatjes opmeten voor de doorlaatstroom, klaar. Juist vanwege de relatief hoge doorlaatstroom van seleniumgelijkrichters van flinke afmetingen werden deze gedurende een korte periode toegepast voor het gelijkrichten van stromen tot meerdere ampères. Een bekende wijsheid onder elektronici zegt echter dat het niet de vraag is óf een seleniumdiode kapot gaat, maar wanneer. Inderdaad was hun betrouwbaarheid en levensduur vrij beperkt, zeker vergeleken met de onverwoestbare siliciumdioden die we tegenwoordig gebruiken (mits spanning, stroom en vermogen binnen de opgegeven grenzen blijven).

Een seleniumdiode die de geest geeft,



Figuur 1. Typisch voorbeeld van een seleniumgelijkrichter. (Creative Commons Attribution-Share Alike 3.0 Unported)



ESTD 2004

www.elektor.tv

Retrotronica is een rubriek over oude elektronica en legendarische Elektor-ontwerpen. Bijdragen, suggesties en vragen zijn meer dan welkom; stuur uw telex of telegram naar jan.buiting@elektor.com

doet dat over het algemeen door oververhit te raken vanwege een verhoogde weerstand in doorlaatrichting. Dat gaat gepaard met een penetrante geur en het vrijkomende gas is in hogere concentraties ook bepaald niet ongevaarlijk (zie het kader: **Neus dicht, raam open, wegwezen**). Wie nog werkende seleniumgelijkrichters tegenkomt, doet er verstandig aan door ze te vervangen door een siliciumdiode plus een serieweerstand die samen dezelfde spanningsval vertonen als de seleniumgelijkrichter.

Germanium dioden en transistoren

De eerste halfgeleiders die op grote schaal beschikbaar kwamen, waren germaniumdioden en transistoren (zie ook het kader: **Buizencodes voor dioden en transistoren**). Germanium (Ge) is een element dat in het periodiek systeem recht onder silicium (Si) staat. Ger-

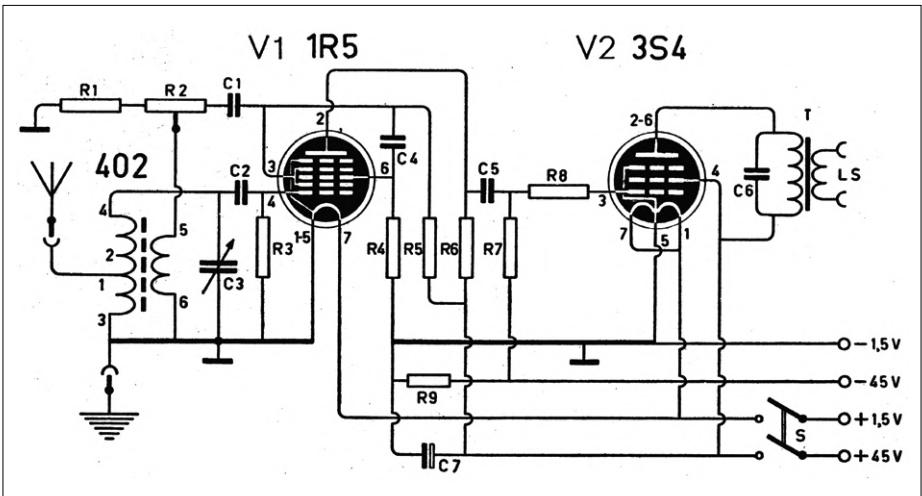
Neus dicht, raam open, wegwezen.

Zoals vermeld zijn seleniumgelijkrichters bepaald niet van gevaar ontbloot. Met de veroudering van een seleniumgelijkrichter neemt de serieweerstand toe en dus ook de spanningsval over de gelijkrichter. Daardoor gaat hij meer vermogen dissiperen en wordt hij warmer. Op een gegeven moment zelfs zo warm, dat het selenium (Se) spontaan begint te oxideren tot seleniumdioxide (SeO_2).

Seleniumdioxide is bij kamertemperatuur een vaste stof. Maar vanuit een oververhitte seleniumgelijkrichter ontsnapt het als een groenig gas. Een giftig gas. Behoorlijk giftig zelfs. Dat je er duizelig, misselijk en moe van kan worden, is nog tot daar aan toe. Maar op blijvende schade aan lever en milt zit niemand te wachten.

Gelukkig heeft de mens een natuurlijke verdediging tegen seleniumdioxidegas. De meeste mensen kunnen een concentratie vanaf 200 ng/m^3 al ruiken en raken daardoor gewaarschuwd vóór de situatie echt gevaarlijk wordt. Bovendien stinkt seleniumdioxide zo afschuwelijk, dat iedereen alleen daarom al weg wil van de bron. Selenium staat in het periodiek systeem recht onder zwavel (S). Selenium en zwavel hebben dus vergelijkbare eigenschappen, net zoals germanium en silicium die ook recht onder elkaar staan (zie tekst). Daarom lijkt de geur van seleniumdioxide wel een beetje op de geur van zwaveldioxide. Rottende uien met knoflook, dus ...

Zie ook Barry L. Ornitz - *Selenium Rectifier Replacement* [7] en "slapeloze nachten door berg uien" [8].



Figuur 2. Schema van de bouwdoos „Nucleon“ van Amroh, met twee buizen en twee batterijen als voeding.



manium heeft dus ook vier valentie-elektronen, die zorgen voor een regelmatige ruimtelijke kristalstructuur in de zuivere vorm van de stof.

Evenals silicium kan germanium gedoteerd worden met een element met één elektron meer in de buitenste schil, bijvoorbeeld fosfor (P) of arseen (As). Er ontstaat dan n-type germanium met een overmaat aan vrije elektronen. Dotering met een element met één elektron minder in de buitenste schil, bijvoorbeeld borium (B), zorgt voor een tekort aan

Figuur 3. Voorkant van het jongerentijdschrift „Radio Blan“ van juli 1960. Let ook op de prijs (uiteraard in guldencenten).

Buizencodes voor dioden en transistoren

In het midden van de vijftiger jaren kwamen de eerste germaniumdioden op de markt met typenummers als OA79 en OA81. Een paar jaar later volgden de eerste germaniumtransistoren van de types OC3, OC4, OC13 en OC14.

Die typenummers zijn gebaseerd op de gangbare (Europese) codering voor elektronen-buizen. Immers, de O staat voor een koude kathode (geen gloeidraad aanwezig), de A staat voor een enkele diode en de C voor een triode, het eenvoudigste versterkende element.

Pas later zijn speciale (Europese) coderingen voor halfgeleiders afgesproken en ingevoerd met de eerste letter A voor alle materialen met een verboden zone ("band gap") van 0,6 – 1,0 eV (in de praktijk altijd germanium); B voor alle materialen met een band gap van 1,0 – 1,3 eV (in de praktijk altijd silicium) en C voor alle materialen met een grotere band gap (bijvoorbeeld galliumarsenide). Vandaar typenummers zoals bijvoorbeeld CQY37 voor een bepaalde infrarode led en CQY80 voor een bepaalde opto-coupler.

Mocht u nog een diode met een OA-code of een transistor met een OC-code hebben, dan is die wellicht uitgevoerd als een glazen buisje dat aan de buitenkant zwart gelakt is. Als u de lak eraf krabt, heeft u een fotodiode of fototransistor!

vrije elektronen en dus p-type germanium. Germanium dioden en transistoren hebben dezelfde structuur en werking als hun broertjes van silicium en zien er ook ongeveer hetzelfde uit (zie kopfoto). Vanwege de verschillen in materiaaleigenschappen hebben ze wel andere elektrische parameters.

Zo is de spanningsval over een geleidende germaniumdiode of een geleidende basis-emitterovergang van een germaniumtransistor slechts 0,1 - 0,2 V, tegen 0,6 - 0,7 V bij siliciumtypes. Tegenover dat voordeel staan diverse nadelen, bijvoorbeeld de hogere serieweerstand van een germaniumdiode en de hogere lekstroom in sperrichting (Ge: ordegrootte 1 μ A, Si: ordegrootte 1 pA). Een germanium transistor is ook niet helemaal "uit" te zetten. Ook zonder basisstroom loopt er een collectorstroom I_{CEO} van enkele μ A bij een transistor voor lage vermogens tot enkele mA bij vermogenstransistoren. Allerlei andere problemen, zoals temperatuurafhankelijkheid, verlopende parameters bij veroudering en exemplarische spreiding bij serieproductie, waren bij germanium halfgeleiders ook beduidend groter dan later bij hun silicium vervangers (voor nóg een probleem, zie het kader: **Scheelt maar een haartje...**). Toen de technologie om silicium halfgeleiders te produceren ver genoeg gevorderd was, zijn germanium halfgeleiders dan ook vlot van het toneel verdwenen, behalve bij enkele toepassingen waarin de lage doorlaatspanning van essentieel belang was.

Bouwdozen voor jongeren

Vóór de uitvinding van germanium halfgeleiders bestonden er al veel zelfbouw-kits met elektronenbuizen, bijvoorbeeld de Philips "Senior" bouwdozen voor verschillende middengolfontvangers en versterkers. Die zelfbouwprojecten waren uitsluitend geschikt voor volwassenen, vanwege de hoge spanningen in de apparaten en de noodzaak om te solderen. Fabrikant Amroh probeerde daar iets aan te doen door enkele bouwdozen met elektronenbuizen op de markt te brengen voor gebruik op batterijen. De middengolfontvanger Amroh Nucleon (**figuur 2**) met luidsprekerweergave gebruikte bijvoorbeeld twee buizen, een DK91 (= 1R5) en een DL92 (= 3S4) met gloestromen van 50 mA, respectievelijk 100 mA bij 1,4 V. Een LR20-batterij (ook wel D-type of "dikke staaf" genoemd) van goede kwaliteit kon die gloeidra-

den enkele uren op temperatuur houden, waarna de klemspanning tot 1,2 V of lager was gezakt en de werking van de radio tot een minimum was gereduceerd. De anodespanning van de Nucleon werd verzorgd door een anodebatterij van 45 V. Tot en met de vijftiger jaren waren anodebatterijen met hoge spanningen (vaak ook 90 V) algemeen verkrijgbaar voor wie een flinke portemonnee wilde meebrengen.

En toen kwamen de germaniumdioden, waarmee ook relatief jonge kinderen een kristalontvanger konden bouwen, zoals de Pionier I van Philips of de "jampotontvanger" uit het jongerenblad "Radio Blan" (**figuur 3**). Let ook op de prijs van dat blad: vijftien (gulden)centen. Omgerekend, maar niet inflatiegecorrigeerd iets minder dan € 0,07. Ook voor die tijd (1960) goedkoop, maar gesponsord door enkele tientallen onderdelenhandelaren die een nieuwe markt roken.

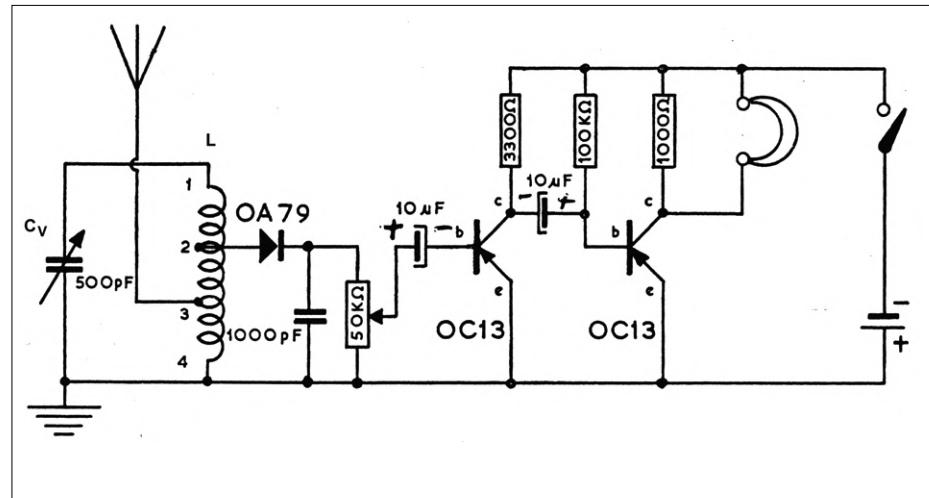
Vlot daarna volgden germaniumtransistoren, waarmee selectievere middengolfradio's met kristaltefoon (zoals de Philips Pionier II, met 2 × OC13, zie **figuur 4**), of zelfs met luidsprekerweergave (zoals de Philips Pionier III, met een extra OC14 als eindversterker) gebouwd konden worden.

Niet veel later kwam Philips met de experimenteerdozen EE8 en EE20 (EE = Electronic Engineer, het getal slaat op het aantal verschillende apparaten dat met de betreffende bouwdoos gemaakt kon worden). Daarin werden iets moderne germanium transistoren gebruikt: AF116 en AC126. Een breed spectrum aan toestellen kon worden gebouwd: grammofoonversterker, elektronisch orgel, intercom, morsecode-oefenapparaat, knipperlicht, akoestisch relais, vochtigheidsindicator, tijdschakelaar, meetbrug en uiteraard diverse radiootjes voor de middengolf, waarop toen nog iets te beluisteren viel.

Best of three worlds

Ook de industrie maakte uiteraard gebruik van selenium en germanium componenten als die een instrument beter, kleiner, goedkoper en/of betrouwbaarder konden maken. Nog vóór germanium transistoren alle elektronenbuizen hadden kunnen vervangen, waren silicium componenten echter al op het toneel verschenen.

De Philips oscilloscoop type PM3230 die we zien in **figuur 5** is een typisch voorbeeld van "the best of three



Figuur 4. Schema van de bouwdoos „Pionier II” van Philips, met een germanium diode type OA79 en twee germanium transistoren type OC13.

"worlds" op dat moment. Naast de kathodestraalbuis bevatten dit apparaat nog zeven andere elektronenbuizen (5 × ECC88 en 2 × E180F). Maar daarnaast telt deze oscilloscoop ook 81 halfgeleiders, waarvan 17 van germanium (vooral AF11x hoogfrequent transistoren) en 64 van silicium (voornamelijk dioden en bruggelijckrichters).

Waarom hadden de ontwerpers en ontwikkelaars van Philips daarvoor gekozen? Wel, een analoge oscilloscoop maakt gebruik van een breed scala aan ver-

Figuur 5. De Philips PM3230 oscilloscoop met elektronenbuizen, germaniumtransistoren, siliciumdioden én siliciumtransistoren aan boord. Fotografie: Gerrit Busscher.



Gekookt of gebakken?

Halfgeleiders zijn gevoelig voor een te hoge temperatuur. Lekker vlot solderen, dus. Germanium halfgeleiders waren nog beduidend gevoeliger voor oververhitting tijdens het solderen dan silicium halfgeleiders.

Er deden dan ook diverse trucjes de ronde om germanium componenten heel te houden bij het bestukken van een printje. Een los krokodillenbekje op de te solderen draad geklemd om de warmte af te voeren was meestal voldoende. Het jeugdblad *Radio Blan* (zie figuur 3) begreep dat jongelui geen grote soldeerervaring konden hebben en bovendien lang moesten sparen voor één diode of één transistor. Omgereden en voor inflatie gecorrigeerd kostte een germaniumdiode in 1960 ongeveer drie euro en een germanium transistor rond de twintig euro.

De tip van Opa Blan, de allesweter op elektronica gebied in het blad, om oververhitting tegen te gaan was navenant: "Vraag je moeder om een plakje rauwe aardappel. Snijd dat tot halverwege in en leg het om de aansluitdraad van de diode of transistor die gesoldeerd moet worden. Trek de aardappel na het solderen weer voorzichtig weg."



Figuur 6. De middengolfradio IC2000 van Philips (1967). Het ronde radiootje past in het etui, wat er rechts aan vast zit is de netvoeding / oplader.

schillende soorten elektronische schakelingen. De ingangsversterkers moeten signaalspanningen in de orde van millivolts verwerken, de naversnelingsspanning van de kathodestraalbuis daarentegen bedraagt vele kilovolts. De voeding produceert gelijkspanningen, maar de ingangssignalen kunnen

(in de PM3230) frequenties aannemen tot 10 MHz, in luxere, duurdere oscilloscopen uit die tijd nog een ordegrootte hoger. Sommige van die functies konden destijds nog het beste met elektronenbuizen worden uitgevoerd, voor andere functies waren germaniumtransistoren of zelfs al siliciumtransistoren een betere

oplossing. In de service manual [2] van de PM3230 kan de geïnteresseerde lezer veel meer details vinden.

Soms lagen er ook verkooptechnische overwegingen ten grondslag aan het gebruik van meerdere soorten actieve componenten. Op de Funkausstellung in Berlijn van 1967 kwam Philips met een (toen) futuristisch ogend middengolfradiootje, de IC2000, zie **figuur 6** voor een opmerkelijk gaaf exemplaar.

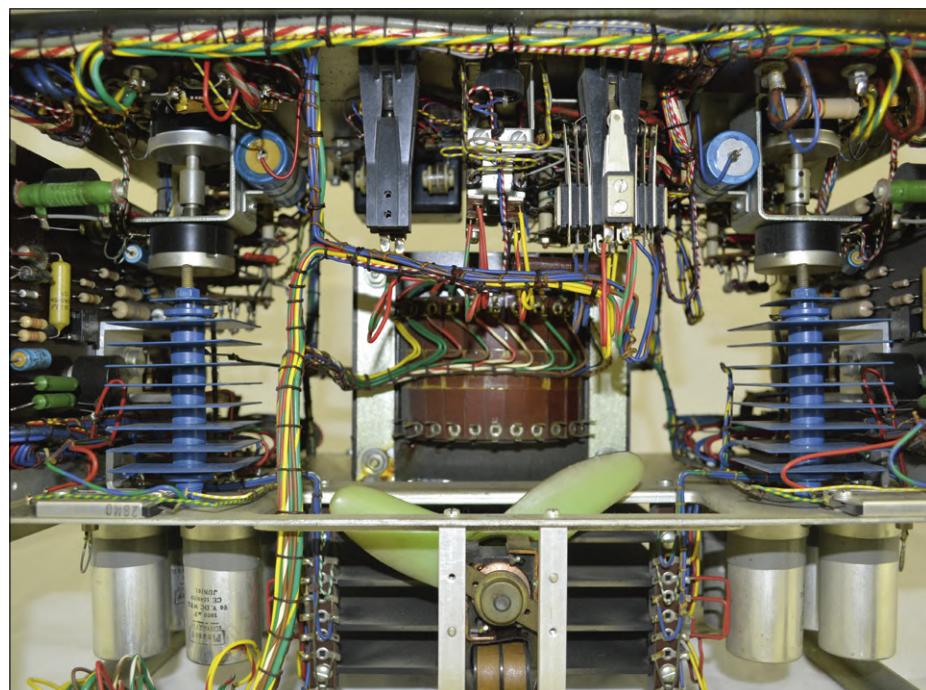
De ronde behuizing van deze radio van 7 cm diameter en 3 cm dik wordt voor ongeveer drie kwart ingenomen door het luidsprekertje. Om de magneet daarvan zit een rond printje met onder andere twee IC's. Eén daarvan (een TAA263) bevat maar liefst drie transistoren! Daarnaast maakt het apparaatje gebruik van één discrete siliciumtransistor, één siliciumdiode en twee germaniumtransistoren in de eindtrap.

Het gebruik van IC's in dit relatief simpele ontvanger is beslist niet essentieel, maar past bij het uiterlijk en bij het gekozen typenummer IC2000. In die tijd was "IC" een soort toverwoord en overal waar het getal 2000 opdook werd dit in de laatste decennia van de vorige eeuw in verband gebracht met een hoogtechnologische toekomst. Er waren in die tijd wel meer fabrikanten die hun verkoopcijfers probeerden te verhogen door het gebruik van "IC" en "2000" in hun typenummers. Wie details zoekt over het hier beschreven apparaat, zoekt de service manual [3] het beste in het archief van de Nederlandse Vereniging voor de Historie van de Radio (NVHR).

Best of one world

Heel anders is het in een relatief simpel apparaat als een industriële (niet-schakelende) gelijkspanningsvoeding. Geen hoge frequenties, geen heel lage of heel hoge spanningen, een rechttoe-recht-aan ontwerp en geen noodzaak om het grote publiek aan te spreken. Daarom zijn het in de Philips gelijkspanningsvoedingen PE4802 (0-15 V / 6 A) en PE4804 (2 x 0-30 V / 2 x 2 A) van begin zestiger jaren uitsluitend selenium- en germaniumcomponenten die de dienst uitmaken.

Figuur 7 toont het inwendige van een PE4804, in het bezit van de Historische Studieverzameling van de faculteit Elektrotechniek, Wiskunde & Informatica van de Universiteit Twente [4], die ook een PE4802 bezit. De grote, blauwe componenten zijn uiteraard de seleniumgelijk-



Figuur 7. Inwendige van een Philips voeding type PE4804, met uitsluitend selenium gelijkrichtcellen en germanium halfgeleiders. Schenking van E.G. Pleiter – Hengelo. Fotografie: Rob Quentemeijer.

Scheelt maar een haartje ...

De eerste auto, de eerste trein, ze functioneerden lang niet zo betrouwbaar als de beste diligences uit die tijd. Niet voor niets zei autopionier en -fabrikant Henry Ford: "Als ik naar mijn klanten had geluisterd, had ik snellere paarden moeten fokken." Zo was het aan het einde van het buizentijdperk ook met (germanium-) transistoren. Die waren meteen al een heel stuk kleiner en energiezuiniger dan elektronenbuizen, maar zeker nog niet betrouwbaarder.

De fabrikanten van vroege germaniumtransistoren (OC170-OC171, AF114-AF117, etc.) maakten goed functionerende producten, maar konden vanzelfsprekend nog niet weten welke levensduur hun nieuwe wonderkinderen in de praktijk zouden hebben. En die levensduur bleek beperkt te worden door de spontane groei van metaalhaartjes aan de binnenkant van de tinnen behuizing

van de betreffende transistoren. Laat tin nou net één van die metalen zijn (samen met antimoon, cadmium, indium en zink) die gevoelig is voor de vorming van haartjes op het oppervlak (zie **figuur 8** en de uitleg van Wikipedia [9]).

De tinnen behuizing van vroege germaniumtransistoren was hol. De

eigenlijke transistor was wel ingebed in een kloddertje siliconenvet, maar meestal bevatte de behuizing ook een luchtbel. Spontaan groeiende metaalhaartjes, enkele micrometers dik en woekerend in die luchtbel, konden op den duur sluiting veroorzaken tussen de behuizing van de transistor en de eigenlijke elektroden van de transistor: collector, basis en/of (!) emitter.

Dat probleem heeft in de vijftiger en zestiger jaren van de vorige eeuw alle fabrikanten van elektronica beziggehouden en zelfs de NASA heeft zich er nadrukkelijk over gebogen. Het probleem is volledig opgelost met de komst van transistoren in een plastic behuizing of in een metalen behuizing van een legering die niet gevoelig is voor het ontstaan van metaalhaartjes. Zie ook: Het raadsel van de OC171 [10] en het artikel van Mark Hennessy [11].



Figuur 8. Vorming van metaalhaartjes op een zinken beugeltje (bron: Wikipedia).

richters en uit de symmetrie tussen de linker- en de rechterkant van de foto is wel duidelijk dat het hier om een dubbele voeding gaat.

Onder de lichtgroene schoepen van de ventilator kunnen de zwarte koelplaten van de germanium vermogenstransistoren worden herkend. Overigens, het motortje dat de ventilatorschoepen aandrijft, is een standaard synchroonmotorje voor netspanning dat in die tijd in alle Philips platenspelers werd toegepast. Ook van dit apparaat zijn meer details

te vinden in de service manual op de website van de NVHR [5][6].

Collectors items zijn het, die Philips voedingen PE4802 en PE4804. Niet omdat ze technisch zo hoogstaand zijn, wel uit historisch perspectief: geen buizen, geen silicium. ►

180574-D-01

Links en Literatuur

- [1] Bron: E.A. Harty – Characteristics and applications of Selenium-Rectifier Cells – Transactions on Electrical Engineering, volume 62, October 1943.
- [2] Philips PM3230 oscilloscoop : www.radiodatabase.nl/meetapparatuur/philips/PM3230/PM3230.pdf
- [3] Philips IC2000 radio : https://nvrbiblio.nl/schema/Philips_22IC2000.pdf
- [4] Historische Studieverzameling van de faculteit Elektrotechniek, Wiskunde & Informatica van de Universiteit Twente : <http://studieverzameling.utwente.nl>
- [5] Philips PE4804 documentatie: https://nvrbiblio.nl/schema/Philips_PE4804.pdf
- [6] Philips PE4804-05 documentatie: https://nvrbiblio.nl/schema/Philips_PE4804-05.pdf
- [7] Vervangen van selenium brugcellen: https://yarchive.net/electr/selenium_rectifiers.html
- [8] Uien verstoren de nachtrust: www.telegraaf.nl/nieuws/1228796/slapeoze-nachten-door-berg-uien
- [9] Metaalhaartjes (Wikipedia): <https://nl.wikipedia.org/wiki/Metaalhaartjes>
- [10] Het raadsel van de OC171: www.elektormagazine.nl/magazine/elektor-200807/15593
- [11] Het AF11x probleem: https://www.markhennessy.co.uk/articles/vintage_transistors.htm#the_af11x_problem



welkom in je WEB STORE

TIP VAN DE REDACTIE



LabNation WiFi Bridge voor SmartScope

De SmartScope is een uitzonderlijk handige en veelzijdige USB-oscilloscoop die u kunt aansluiten op elke computer die draait onder Windows, OS X, iOS, Android of Linux. Nu is een praktische uitbreidingsmodule verkrijgbaar waarmee de SmartScope draadloos via WiFi gebruikt kan worden: de WiFi Bridge.

De WiFi Bridge van LabNation is een prima uitbreiding voor de SmartScope en bijzonder interessant voor iedereen die draadloos of op enige afstand wil meten. Van harte aanbevolen voor alle bezitters van een SmartScope!

Harry Baggen
(Elektor Labs)



www.elektor.nl/wifi-bridge-smarscope

Embedded in Embedded



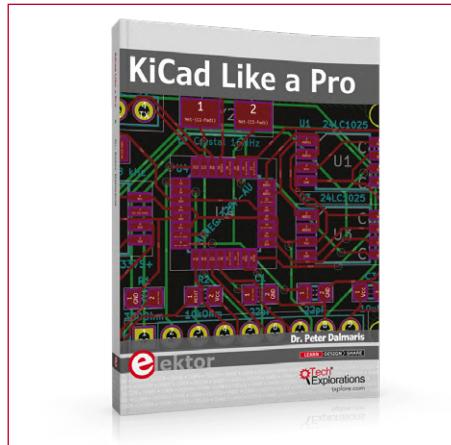
Dit boek behandelt embedded ontwerpen, inclusief een besprekking van de hardware, de processor-initialisatie, de ontwikkeling van low-level drivers, en het ontwerp van een applicatie-interface voor een product. Hoewel de auteur dit beschrijft aan de hand van een specifieke applicatie van een Cortex-M3 ontwikkelboard, slaagt hij er in de lezer de basisvaardigheden aan te reiken die nodig zijn om een goed productontwikkelaar te worden.



Ledenprijs: € 44,96

www.elektor.nl/embedded-in-embedded

KiCad Like a Pro



Dit (Engelse) boek leert u KiCad te gebruiken. Of nu u een hobbyist bent of een professionele elektronicus, dit boek helpt u om snel productief te worden en te beginnen met het ontwerpen van eigen printen. Het boek volgt een praktische benadering om te leren: het bestaat uit vier projecten van oplopende moeilijkheidsgraad en verschillende recepten.



Ledenprijs: € 35,96

www.elektor.nl/kicad-like-a-pro

Elektor Bestsellers

1. SDR Hands-on Book

www.elektor.nl/sdr-hands-on-book



2. Raspberry Pi 3 B+

www.elektor.nl/rpi-3-plus

3. Elektor SDR Shield 2.0

www.elektor.nl/170515-91

4. Bakeey UM25C USB Multimeter Logger

www.elektor.nl/bakeey-um25c

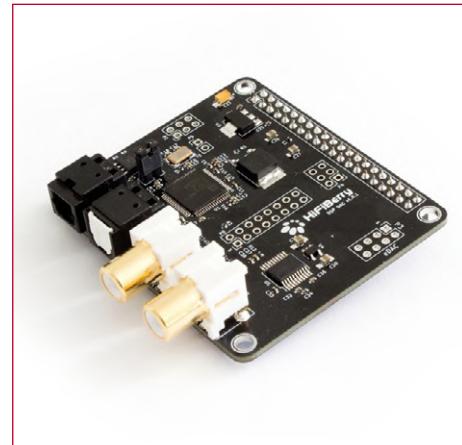
5. Miniware TS100 soldeerbout

www.elektor.nl/miniware-ts100

6. PCBite Kit

www.elektor.nl/pcbite-kit

HiFiBerry DAC+ DSP



De HiFiBerry DAC+ DSP is een high-resolution digitaal/analoog-converter voor de Raspberry Pi. Hij combineert een Burr-Brown DAC met digitale in- en uitgangen met een krachtige digitale signaalprocessor.

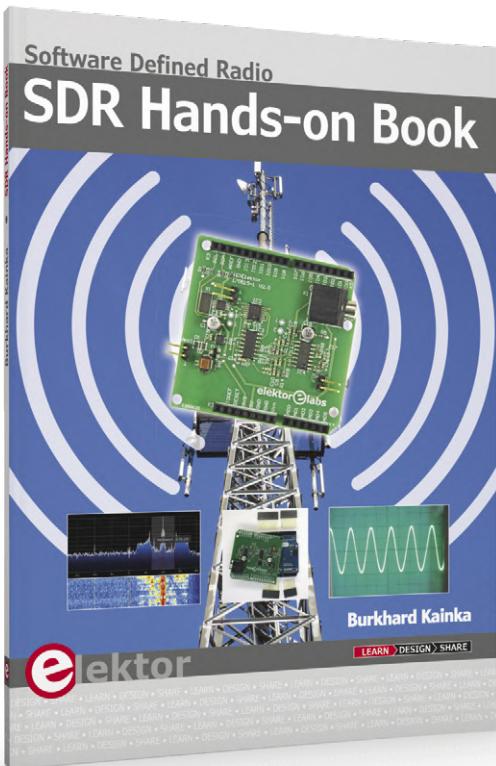


Ledenprijs: € 71,96

www.elektor.nl/hifiberry-dac-dsp



nieuw



SDR Hands-on Book

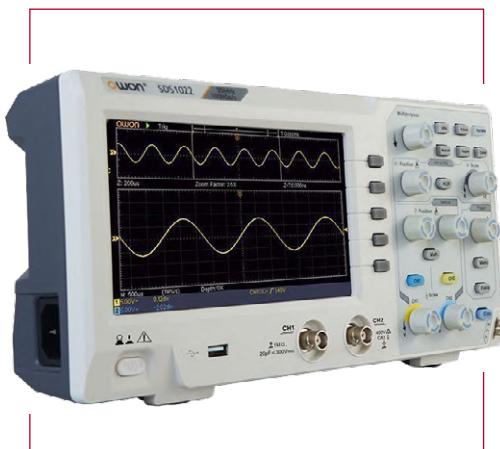
Het Elektor SDR Shield is een veelzijdige kortegolfontvanger tot 30 MHz. Met een Arduino en de passende software kunnen radiostations, morsesignalen, SSB-stations en digitale signalen worden ontvangen.

In dit boek beschrijft de bekende auteur en radio-amateur Burkhard Kainka de moderne praktijk van software-defined radio met behulp van het Elektor SDR Shield. Hij beperkt zich daarbij niet tot de theoretische achtergrond maar geeft ook tekst en uitleg over talloze open source softwaretools.



Ledenprijs: € 26,96
www.elektor.nl/sdr-hands-on-book

OWON SDS1022 tweekanaals-oscilloscoop (20 MHz)



De OWON SDS1022 is een aantrekkelijk geprijsde oscilloscoop met een bemonsteringssnelheid van 1 GSa/s en een bandbreedte van 20 MHz. Bovendien heeft deze 'scoop twee kanalen in de aanbieding en is het mogelijk 10 K punten per meting op te slaan. Op de achterzijde van het instrument bevindt zich een USB-connector om meetresultaten naar een PC over te brengen, en aan de voorzijde een USB-poort om metingen op een USB-stick op te slaan.



Ledenprijs: € 170,96

www.elektor.nl/owon-sds1022

The Complete ESP32 Projects Guide



Dit boek wil u de Arduino IDE en MicroPython programmeertalen voor projecten op basis van de ESP32 bijbrengen, onder gebruikmaking van het bijzonder populaire ESP32 DevKitC ontwikkelboard. In het boek wordt een groot aantal projecten met de Arduino IDE en de ESP32 DevKitC gepresenteerd, variërend van simpel tot gevorderd. Alle projecten zijn getest en werken. Voor alle projecten worden de (blok)schema's en complete programmalistings inclusief commentaar gegeven.



Ledenprijs: € 35,96

www.elektor.nl/esp32-projects-guide

PCBite Kit incl. vier PCBite-probes en meetsnoeren



PCBite is het perfecte tool om uw printen tijdens het ontwikkelproces te hanteren. Krachtige magneten bieden, in combinatie met een roestvrijstaal basisplaat, flexibiliteit, mobiliteit en gebruiksvriendelijkheid. De houder kan gemakkelijk worden verplaatst om printen van verschillende vorm en afmetingen te hanteren. De probe is tegelijk stevig en buigzaam, en kan worden gebruikt voor directe metingen of voor volledig handsfree gebruik met uw multimeter of ander meetinstrument.



Ledenprijs: € 121,46

www.elektor.nl/pcbite-kit

Hexadoku puzzelen voor elektronici

PC, oscilloscoop en soldeerbout kunnen weer even op adem komen terwijl u uw hersenen pijnigt met onze Hexadoku. De instructies voor deze puzzel zijn heel eenvoudig. De Hexadoku werkt met de hexadecimale cijfers 0 t/m F, helemaal in de stijl van elektronici en programmeurs.

Vul het diagram van 16 x 16 hokjes zodanig in dat **alle** hexadecimale cijfers van 0 t/m F (dus 0...9 en A...F) precies éénmaal voorkomen in elke rij, in elke kolom en in elk vak van 4 x 4 hokjes (gemarkerd door de dikdere zwarte lijnen).

Een aantal cijfers is al aangegeven en deze bepalen de uitgangssituatie voor de puzzel.

Onder de inzenders met de goede oplossing verloten we vijf waardebonnen. Om mee te dingen naar een van deze prijzen dient u **de cijfers in de grijze hokjes** naar ons op te sturen.



Doe mee en win! Onder de internationale inzenders met het juiste antwoord verloten we **vijf Elektor-waardebonnen**, elk ter waarde van **50 Euro**.

Het is dus zeker de moeite waard om mee te doen!

Stuur uw antwoord (de getallen in de grijze hokjes) **vóór 20 juli 2019** naar: hexadoku@elektor.nl

De prijswinnaars

De juiste oplossing van de Hexadoku uit het mei/juni-nummer is: **46F7C**.

De Elektor-waardebonnen van 50 Euro zijn gewonnen door Guillaume Demierbe (België), Alexandr Papazyan (Rusland), Michalis Fostiropoulos (Griekenland), Sune Johansson (Zweden), Ruth Hanselmann (Duitsland).

Allemaal van harte gefeliciteerd!

6		D	3	A	5				C	4					
		C			E	3		9	0	5	6				
	F		5	1	6					2					
5	1			9	B	6	C	F		3					
8				3		9	7	0	F		D				
	3		A	F	7	4	B	2							
A	D	7	2	6				E		4					
	4	E	B				2	3	6	8	0				
1	6	F	4	2				0	C	8					
2		5					3	4	1	0	D				
	0	1	8	F	A	6			4						
C	A	D	7	0		2					F				
8		A	C		F	D	6			0	2				
C					0	9	E			3					
F	D	6	0	B	3			E							
0	7				2	F	C	D			8				

B	A	E	3	5	1	0	8	4	F	C	7	D	9	2	6
D	F	1	4	6	9	7	A	E	5	2	3	B	0	8	C
0	2	7	5	D	3	C	F	B	6	8	9	E	4	1	A
9	C	6	8	E	2	B	4	A	D	0	1	3	5	F	7
A	7	4	B	9	0	D	E	6	2	3	F	C	8	5	1
C	1	D	6	B	4	2	3	5	E	7	8	9	A	0	F
2	3	8	0	7	5	F	1	9	A	D	C	4	B	6	E
E	5	9	F	8	6	A	C	0	B	1	4	2	3	7	D
1	B	5	C	3	7	8	D	2	0	9	E	F	6	A	4
3	D	0	E	A	B	9	2	1	4	F	6	5	7	C	8
4	6	F	7	C	E	1	5	3	8	A	D	0	2	9	B
8	9	A	2	0	F	4	6	7	C	B	5	1	D	E	3
F	E	2	D	1	A	3	9	8	7	4	0	6	C	B	5
6	8	B	9	F	D	E	0	C	3	5	A	7	1	4	2
5	4	C	1	2	8	6	7	D	9	E	B	A	F	3	0
7	0	3	A	4	C	5	B	F	1	6	2	8	E	D	9

Medewerkers van Elektor International Media en hun familieleden zijn van deelname uitgesloten.

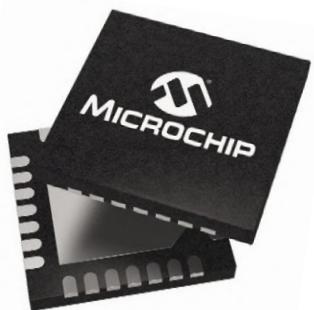


Maak het verschil met uw toepassing

Geef uw embedded ontwerp een persoonlijk cachet met flexibele, geïntegreerde bouwstenen

Microchip begrijpt dat uw ontwerpeisen uniek zijn. Daarom zijn onze producten voorzien van een robuuste reeks geïntegreerde periferiebouwstenen voor de meest uiteenlopende toepassingen:

- Voeg doeltreffende schermen en aanrakingsgevoelige bediening toe voor intuitieve interactie van de gebruikers
- Verbind uw toepassing met de hele wereld — met en zonder kabeltjes
- Regel uw motor of spanningsomzetting
- Beveilig de data in uw toepassing



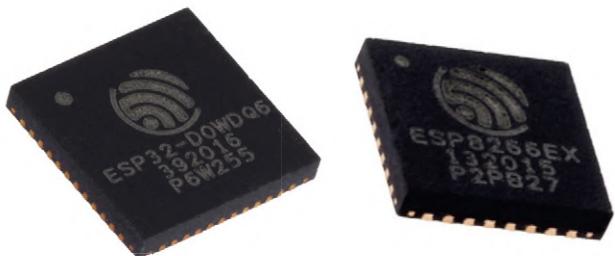
Geef uw product een persoonlijk tintje met de ingebouwde periferiebouwstenen en bespaar op ontwerp kosten en tijd.

**Pas uw ontwerp naar eigen voorkeur aan op
www.microchip.com/FlexibleFunctions**

BEKIJK ONS
VOLLEDIGE AANBOD



SoC - geïntegreerde schakelingen



IoT-modules



Ontwikkelingstools



ESPRESSIF

WIFI- EN BLUETOOTH -CHIPSET EN -OPLOSSINGEN

BETROUWBAAR

ROBUUST

VEILIG

ECONOMISCH

ENERGIEZUINIG

VEILIGE EN STABIELE

IOT-CONNECTIVITEITSOPLOSSINGEN