

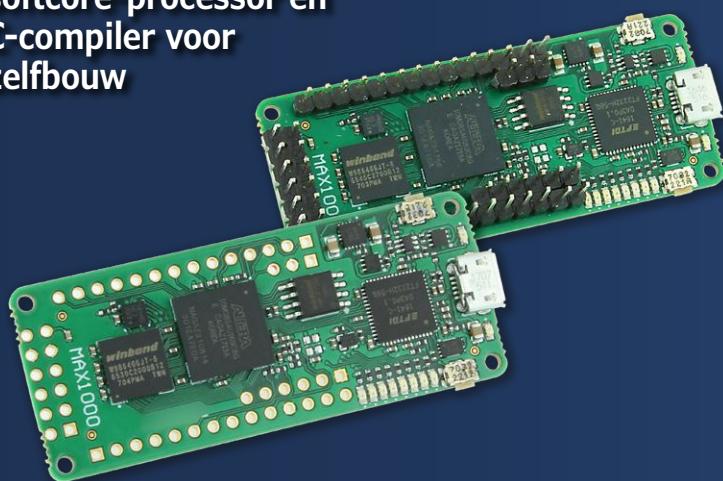
Versla de Elektor-Flipperklok



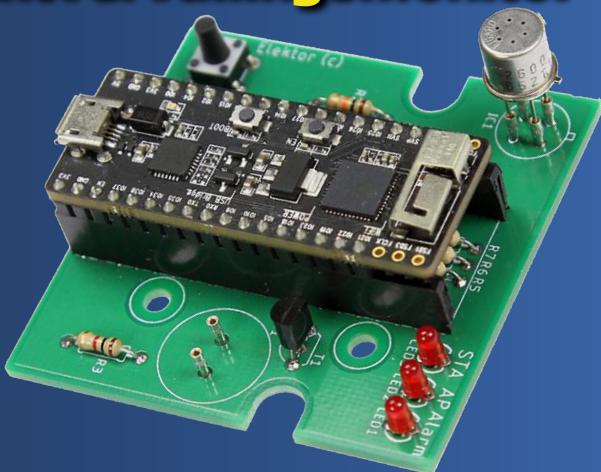
aaah... dat geluid van ratelende cijfers

Het SCCC-Project (1)

softcore-processor en C-compiler voor zelfbouw



Luchtvervuilingsmonitor



rookt daar iemand op het toilet?

Microcontroller-kits voor dummies ▷ Ventilatorsturing ▷ Achteruitkijkradar voor tweewielers ▷ Opamps in de praktijk (2) ▷ Wolverine CNC-portaalrobot ▷ Stappenmotorbesturing met antiresonantietechniek ▷ Hardware-ontwerp met (V)HDL (3) ▷ eBoB voor ENS210-dauwpuntsensor ▷ Automotive-stoorbronnen ▷ Digitale modulatietechnieken (1) ▷ De Raspberry Pi-lineaal ▷

- ✓ Meer dan 100.000 producten
- ✓ Geweldige prijs-kwaliteitverhouding
- ✓ Hoge beschikbaarheid en betrouwbare levering
- ✓ Sterke merken en voortreffelijke kwaliteit

PROFESSIONELE TECHNOLOGIE VOOR WERKPLAATS EN LABORATORIUM VOLLEDIG VERMOGEN OP DE WERKPLEK!

Het Weller WE1010 Education Bundle

70 watt vermogen voor professionele eisen!

Het sterkste soldeerstation in zijn klasse!

Met dit voordeelpakket bent u perfect uitgerust voor alle taken. Naast het WE1010 soldeerstation bevat deze set een 170 mm Weller Xcelite zijsnijtang, een rol Weller WSW SCN N1溶deertin alsmede een extra soldeerpoint ETB 2,4 mm.

- ESD-veilig soldeerstation, soldeerbout en hittebestendige siliconenkabel
- Temperatuurstabiliteit en -vergrendeling beschermt de soldeerpoint en de componenten
- Wachtwoordbeveiliging slaat alle instellingen op

Bestelnr.: WELLER WE1010EB

NIEUW **202,55**
(€ 167,40)



Laboratoriumvoeding PeakTech® 0 - 30 V, 0 - 10 A

Door het zeer handige ontwerp en het lage gewicht is deze schakelende voedingseenheid bij uitstek geschikt voor toepassingen in de ontwikkeling en het onderwijs.

- Spanning en stroomvoorkeuze met schakelbare uitgang
- Modus voor constante stroom en constante spanning met aangesloten belasting
- automatische uitschakeling bij kortsluiting
- fijn instelbare uitgangswaarden: 10 mV, 10 mA



**KWALITEIT VOOR OPLEIDING,
LABORATORIUM EN SERVICE**

Bestelnr.: PEAKTECH 6226

97,71
(€ 80,75) **BEST SELLER**

Dagprijzen · Prijzenstand: 30. 1. 2019

De wettelijke herroepingsregelingen zijn van toepassing. Alle aangegeven prijzen in euro inclusief de wettelijke BTW, excl. verzendkosten voor de totale winkelwagen. Uitsluitend onze Algemene Voorwaarden (zie hierover <https://rch.lt/AV-NL> of op aanvraag) zijn van toepassing. Afbeeldingen kunnen afwijken. Drukfouten, vergissingen en prijswijzigingen voorbehouden.

reichelt elektronik GmbH & Co. KG, Elektronikring 1, D-26452 Sande, Tel.: +49 (0)4422 955-360

Digitale opslag oscilloscoop voor beginners! 50 MHz, 2 kanalen

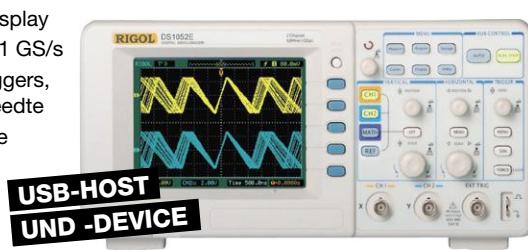
RIGOL
Innovation or nothing

Veelzijdige digitale opslag oscilloscoop, ook ideaal voor mobiel gebruik.

- verlicht full-colour TFT-display
- Bemonsteringssnelheid: 1 GS/s
- verschillende soorten triggers, incl. edge, video, pulsbreedte
- met Ultra Sigma Software

Bestelnr.:
RIGOL DS1052E

305,12
(€ 252,16)



Compacte multimeter met 2.000 tellingen

- Spanningsmeting: 200 mV - 600 V AC/DC
- Stroommeting: 200 µA tot 200 mA
- Weerstandsmeting: 200 Ohm tot 20 MOhm
- Continuïteit (zoemer) en diode testen

Bestelnr.: CHAU P01191740Z

50,85
(€ 42,03) **NIEUW**

Betaalwijzen: Sofort. VISA MasterCard Maestro PayPal IDEAL amazon pay

CHAUVIN ARNOUX



www.reichelt.nl

BESTEL-HOTLINE: +49 (0)4422 955-360

Colofon

59^e jaargang nr. 652,
maart/april 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: Senefelder Misfit, Doetinchem
Distributie: Betapress, Gilze



Niet geheel geruisloos

Ook in dit nummer gaan we weer los! Nadat we door het Lab helemaal gek zijn gemaakt met hun getest met de Flipperklok, kunnen we hem dan nu eindelijk presenteren. Het is op zijn zachtst gezegd geen opvallende schakeling... – “Wat zeg je?” HET IS GEEN STILLE SCHAKELING!!! Zo, ik hoop dat de hint subtel genoeg is.

Maar het lab maakt ook subtielere schakelingen, zoals de Luchtvervuilingsmonitor. Voor dit project hebben we een sensor van Figaro gekozen en hem in een schakeling met een ESP32 ingebouwd. Hiermee kunnen we meteen meedoen aan het Internet of Things en onze meetresultaten via ThingSpeak online voor de hele (internet)wereld beschikbaar maken.

In deze uitgave vindt u ook het eerste deel van een artikelserie waarin we ingaan op het gebruik van “softcores”. Als u zich afvraagt of dat het tegenovergestelde is van “hardcore” en dat u dan aan een bepaalde entertainmentsector denkt — welke, dat hangt geheel van uw eigen interessegebied af — dan bent u A) geen elektronicus en hebt u per ongeluk het verkeerde blaadje in handen (maar lees vooral verder!), of B) niet thuis in FPGA-techniek en is het hoog tijd dat u de artikelserie over het SCCC-project volgt! Het mag dan wel een onderwerp met weinig ‘groen’ (aka printplaat) en veel code zijn, maar vergeet niet dat FPGA’s vaak de kern vormen van elektronische systemen waar veel processen parallel verlopen. Denk bijvoorbeeld aan een gsm-zendmast die tig mobieljes met gesprekken en dataverkeer tegelijk moet kunnen ondersteunen. Of een datacenter met tal van servers en glasvezelverbindingen in de terabits.

Enfin, ik hoop dat we genoeg ‘luchtige’ artikelen tussen al dit geweld door hebben gestrooid om deze editie uit te balanceren. Ik wens u in ieder geval weer veel leesplezier!

En o ja, lees beslist eventjes wat we gedaan hebben met de Raspberry Pi-lineaal (die u misschien al van ons cadeau hebt gekregen op een beurs... en anders: kom eens langs!)

Thijs Beckers

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 Advertentiewesen gedeponeerd bij de rechtbanken in Nederland. Een exemplaar van de Regelen voor het Advertentiewesen 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 gebruikte vergoeding.

Ons team

Hoofdredacteur:

Thijs Beckers (thijs.beckers@eimworld.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, Eric Bogers,

Kees de Groot, Martien Jansen, Evelien Snel

Ledenmanagement:

Raoul Morreau (raoul.morreau@eimworld.com)

Vormgeving & Layout:

Giel Dols

Hoofd Online:

Daniëlle Mertens

Directeur:

Don Akkermans

in dit nummer

59^e jaargang – nr. 652

maart/april 2019

- 45 Vreemde onderdelen
spanningsreferentie LTZ1000
- 60 HomeLab Helicopter
- 71 Elektor Labs Pipeline
- 76 Shop-highlight
waar elektronici van dromen...
- 86 V & A
behuizingen
- 102 Escaped from the Labs
storende ledlampen
- 104 Retrotronica
met de groeten van de tekenkamer
- 112 Elektor Store
- 114 Hexadoku

achtergrond & info

- 6 Microcontroller-kits voor dummies
alle begin is makkelijk!
- 36 De NXP Cup 2019
groter en beter dan ooit!
- 56 Opamps in de praktijk
Deel 2: FET-opamps en breedbandtoepassingen
- 64 Digitale modulatietechnieken: de basis
GMSK, OQPSK, QAM, OFDM en co.
- 78 Hardware-ontwerp met (V)HDL (3)
Een WS2812-LED-matrix aansturen
- 98 Automotive-stoorbronnen
En hoe je die elimineert
- 110 Elektor Ethisiek
Hoe implementeer je ethische waarden concreet in je project?

Versla de

aaah... dat geluid

Het SCCC-Project (1)

softcore-processor en C-compiler
voor zelfbouw



24

Soms hebben we in een project zowel een FPGA als een klassieke CPU nodig, waarmee ook complexe besturingsprogramma's eenvoudig kunnen worden geprogrammeerd. Een manier om dat op te lossen is door gebruik te maken van een processor die in de FPGA is gerealiseerd. Zo'n 'softcore' kan kant-en-klaar worden gekocht, maar is vaak niet bepaald goedkoop. In dit project presenteren we een eenvoudige zelfbouw-CPU voor FPGA's. Dankzij de open broncode kan iedereen die aan zijn of haar eigen behoeften aanpassen. Daar hoort ook een zelfbouw-C-compiler bij, die eveneens kan worden aangepast en uitgebreid. In deze serie artikelen vullen we de theorie zoals gewoonlijk aan met veel praktische voorbeelden.

projecten

- 14 Versla de Elektor-flipperklok!
aaah... dat geluid van ratelende cijfers
- 24 Het SCCC-Project (1)
softcore-processor en C-compiler
voor zelfbouw
- 32 Ventilatorsturing
naar keus met Arduino of analoge elektronica



104

Retrotronica
met de groeten
van de tekenkamer

Elektor-flipperklok!

14 van ratelende cijfers

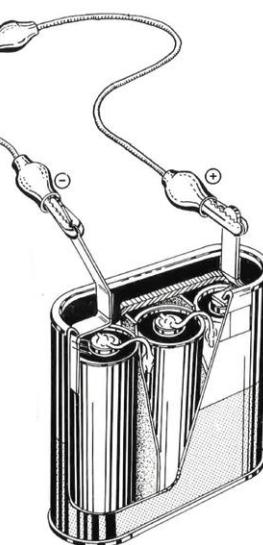
Luchtvervuilingsmonitor

rookt daar iemand op het toilet?

De kwaliteit van de lucht die we inademen zou ons allemaal zeer ter

ten gaan. We weten allemaal dat in de lucht astmapatiënten de nacht wakker houdt, maar ook de aanwezigheid van bepaalde gassen kan erg schadelijk zijn.

Dit project meet de aanwezigheid van vervuilende gassen in de lucht en laat een alarm klinken als de concentratie een bepaald niveau overschrijdt, bijvoorbeeld wanneer iemand een sigaret opsteekt...



- 37 Stappenmotorbesturing met antiresonantietechniek**
groter rendement en minder trillingen bij stappemotoren
 - 42 Achteruitkijkradar voor tweewielers**
veiliger op de fiets
met een simpel schakelingetje
 - 46 De Raspberry Pi-lineaal**
pret met een time of flight-sensor

ElektorLabs magazine

- 50 Wolverine CNC-portaalrobot**
multifunctionele krachtpatser
 - 72 Energiezuinige ledklok**
toont de tijd alleen wanneer nodig
 - 83 eBob voor ENS210-dauwpuntsensor**
met voorbeeld voor de Arduino UNO
 - 90 Luchtvervuilingsmonitor**
rookt daar iemand op het toilet?
 - 94 I²C-bus datalezer**
via USB alles op het pc-scherm



binnenkort

Elektorlabs magazine mei/juni 2019

Elektorlabs magazine mei/juni 2019 • Fijnstofsensor voor het Weerstation • Wekker met ESP32 • USB-stroommeter • Leidingen opzoeken • Eenvoudige audiospectrummeter • Microcontroller-kits voor gevorderden • Accubeveiliging • Het IOTA-project en een FPGA voor de Raspberry Pi • Opamps in de praktijk (3) ...en nog veel meer.

Aankondigingen onder voorbehoud.
Ektorlabs mei/juni 2019 verschijnt op 18 april 2019

Elektor Industry 2/2019

Editie 2/2019 heeft als onderwerp sensoren, apparatuur en methodes voor testen en meten. De contributies komen van bedrijven, specialisten uit de industrie, Elektor-redactie en freelance-auteurs. Daarnaast schenken we uitgebreid aandacht aan de Sensor & Test-expositie in Neurenberg, Duitsland, 25 t/m 27 juni 2019.

Elektor Industry 2/2019 verschijnt op 18 juni 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.elektermagazine.com. Aankondigingen onder voorbehoud.

Microcontroller-kits voor dummies

alle begin is makkelijk!

Menig elektronicus heeft enige drempelvrees voor het programmeren van microcontrollers. Niet nodig! Het is helemaal niet zo moeilijk, zoals we zullen laten zien aan de hand van vier low-cost-boards. Er worden een heleboel demo-programma's bij die boards bijgeleverd, zodat u al snel resultaten kunt zien. In dit eerste artikel probeert de bekende Elektor-auteur Burkhard Kainka een bijzonder gunstig geprijsde Arduino-kloon uit. Zijn zoon Fabian Kainka zet de eerste stappen met een NodeMCU-board, dat zich bij uitstek leent voor projecten rond het Internet.

(Te) veel elektronici zitten nogal eens aan te hikken tegen het programmeren van microcontrollers. Eén van de redenen is dat het nogal kan tegenvallen om in het onoverzichtelijke woud van microcontroller-kits de juiste te vinden, die niet te ingewikkeld is, die fatsoenlijk met hard- en software wordt ondersteund door de fabrikant en door andere partijen, en waar een *community* aan verbonden is waar je als beginner terecht kunt als je omhoog zit (lees: waar je niet bang hoeft te zijn om domme vragen te stellen).

Wij hebben vier controllerboards/-kits uitgekozen die geschikt zijn voor beginnende programmeurs. Die hebben we door vier elektronici laten uitproberen. U zult zien dat de eerste stappen in de microcontroller-wereld helemaal niet zo moeilijk zijn.

In dit nummer van Elektorlabs wordt het spits afgebeten door

Burkhard Kainka en zijn zoon Fabian. Eerstgenoemde is een alom bekend als ontwerper van een zeer groot aantal interessante schakelingen voor zelfbouw, auteur van talrijke boeken en lespakketten en exploitant van een website voor hobby-elektronica. Hij heeft zich op een Arduino Nano-kloon.

Zijn zoon Fabian is in de voetsporen van zijn vader getreden en heeft zich gespecialiseerd in ontwikkelingen en publicaties rond het thema Internet of Things. Daarom kreeg hij een microcontroller-kit onder handen die vooral daarop is gericht, een NodeMCU-kit.

In de volgende uitgave van Elektorlabs gaan we onder andere in op een ESP32-WLAN/Bluetooth-board met geïntegreerd OLED-display.

JOY-iT Nano V3

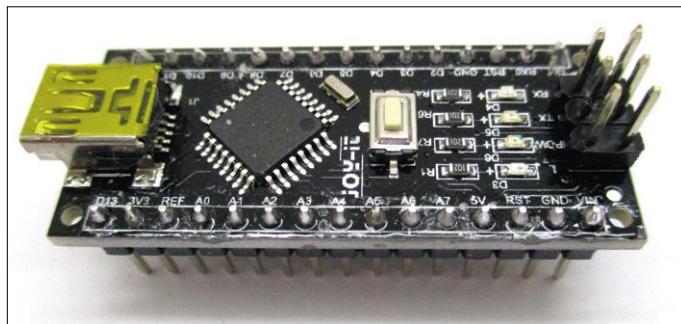
Burkhard Kainka (Duitsland)

De JOY-iT Nano V3 is een compatibele Arduino Nano van – u raadt het al – de firma JOY-iT, in de Elektor-shop verkrijgbaar voor nog geen 12 euro. Daarbij hoort een Arduino Supplement Kit met additionele onderdelen om een simpele en snelle start met dit microcontrollerboard te vergemakkelijken.

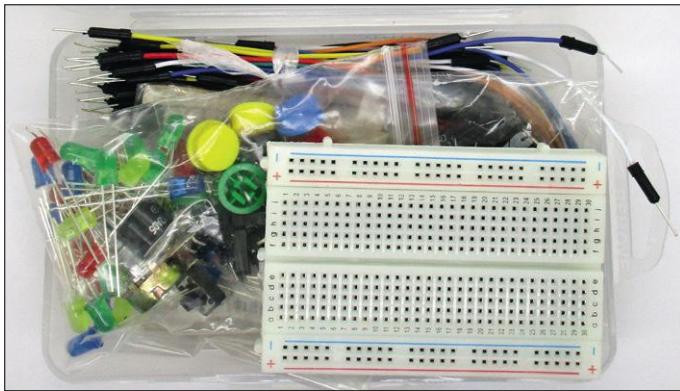
U zou eigenlijk altijd een Arduino op voorraad moeten hebben, want zo'n ding bewijst u waardevolle diensten in het elektronica-lab en bij het uitproberen van nieuwe ideeën. Nu gebruik ik zelf meestal het standaardmodel, de Arduino Uno, maar de Nano biedt bijna dezelfde hardware rond dezelfde ATmega328, alleen op een veel kleiner printje. Dat betekent een geheugenbereik dat ook groot genoeg is voor grotere projecten, en meer rekenkracht dan je normaal gesproken nodig hebt.

Eerste experimenten

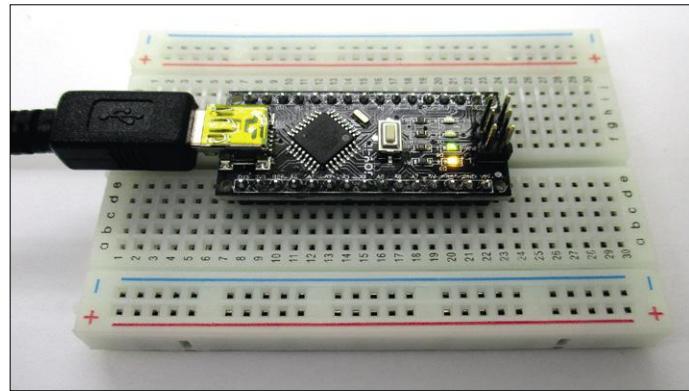
Wie zich afvraagt wat het verschil is tussen de originele Arduino Nano en de kloon van JOY-iT (**figuur 1**), die moet de onder-



Figuur 1. De Nano V3.



Figuur 2. De inhoud van de Supplement Kit.



Figuur 3. De Arduino Nano aan het werk.

kant eens bekijken. Daar bevindt zich de USB-chip, een FT232R bij het origineel en een CH340G bij de kloon. Die doen hetzelfde, ze vertalen de seriële poort van de ATmega naar een USB-poort. Sluit u hem aan uw PC aan, dan ziet u daar een virtuele seriële poort. Die heet dan bijvoorbeeld COM2, COM3 of COM99, al naargelang hoeveel andere componenten met een seriële poort uw PC al gezien heeft. Met een FT232 wordt er steeds weer een nieuw COM-nummer aangemaakt, maar bij de CH340 is dat niet het geval. Daarbij heeft een volgend board steeds hetzelfde COM-poort-nummer, dat u overigens kunt veranderen als u meerdere boards tegelijk wilt aansluiten. Mocht u op uw PC problemen met uw USB-driver ontdekken, dan moet u die van het internet downloaden [1] en ‘met de hand’ installeren.

De Arduino Supplement Kit (**figuur 2**) bestaat uit een klein breadboard en een heleboel nuttige onderdelen, van kabels tot een potmeter. Daarmee heeft u meer LED’s, weerstanden, condensatoren, drukschakelaars en ander ‘klein grut’ dan u in een week verknutselt.

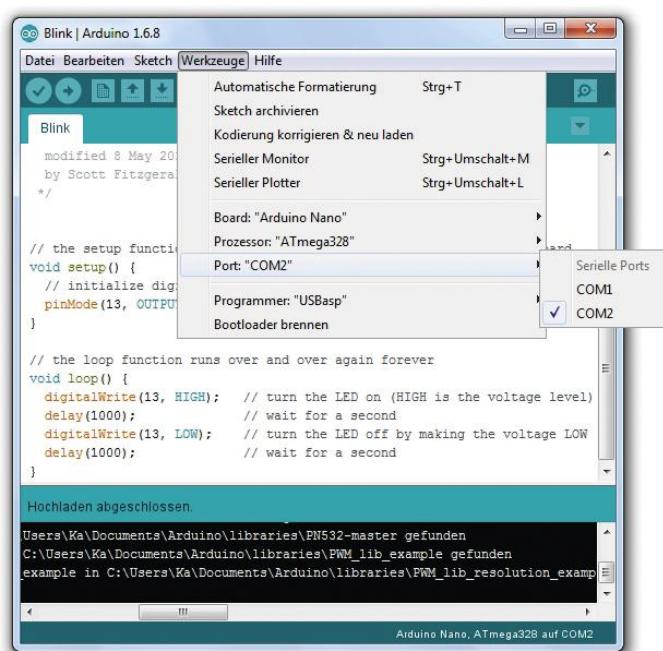
Tot zover de inleiding, nu aan de slag met deze Nano. Ik heb het printje op het breadboard geprikt (**figuur 3**) zodat ik mijn experimenten solide en zonder ongewenste kortsluitingen kan uitvoeren. We maken de verbinding met de PC met een mini-USB-kabel. Het groene LED’je op de print licht op ten teken dat voedingsspanning aanwezig is. De gele LED knippert met korte pulsjes. Dat betekent dat er nog geen programma geladen is en dat de bootloader op werk staat te wachten.

Dan starten we de Arduino-IDE. Telkens als ik een nieuwe Arduino test, pak ik als eerste het voorbeeldprogramma *Blink.ino* erbij (**figuur 4**), dat is inbegrepen in de IDE. Oeverigens praten we bij Arduino niet over programma maar over sketch. *Blink.ino* laat de gele LED langzaam in een tempo van 2 s knipperen. Voordat ik het programma in de Arduino kan laden moet ik natuurlijk eerst het juiste board kiezen in de IDE (Arduino Nano), en de juiste seriële poort (COM2). Dat het juist COM2 is geworden had ik verwacht, want onlangs had ik een nog andere controller gebruikt die ook met de CH340 was uitgerust, en die had ik op COM2 gezet.

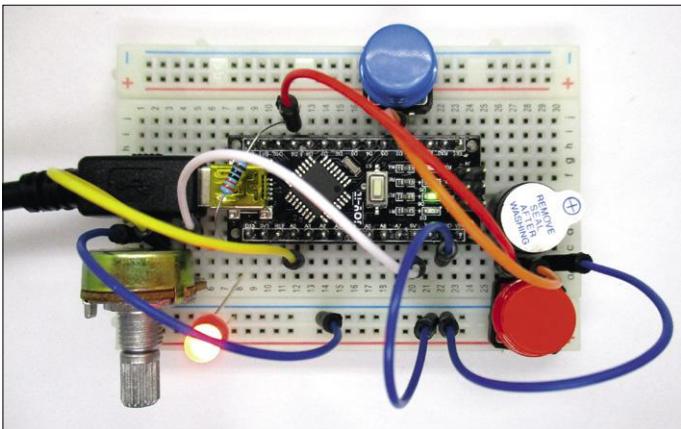
Dus: uploaden en afwachten. Met goed gevuld volbracht, de LED knippert langzaam. Om nu helemaal op zeker te spelen verander ik de delay-tijden in 2000 ms en upload de sketch nogmaals. Hij knippert langzamer.

En daarmee functioneert alles naar wens. Dus wat kunnen we verder nog doen? Mijn blik dwaalt over het materiaal in de Supplement Kit. Ik zie een buzzer die getest wil worden. Het is een actieve zoemer met ingebouwde elektronica. Een test aan de 3,3-V-uitgang van de Nano is succesvol, met 5 V is de zoemer luid. De sticker halen we eraf zoals aanbevolen, maar hij zoemt nu onaangenaam luid dus we plakken ‘m er meteen maar weer op. Met een weerstandje dan? Uitgeprobeerd en verworpen, met meer dan 100 Ω is er niet genoeg stroom om de zoemer te starten. Aan een poort moet het toch wel lukken. De zoemer wordt aangesloten op pin 13, dat is dezelfde als de gele LED op het board. En zoals te verwachten klinkt de zoemer tegelijk als de LED oplicht.

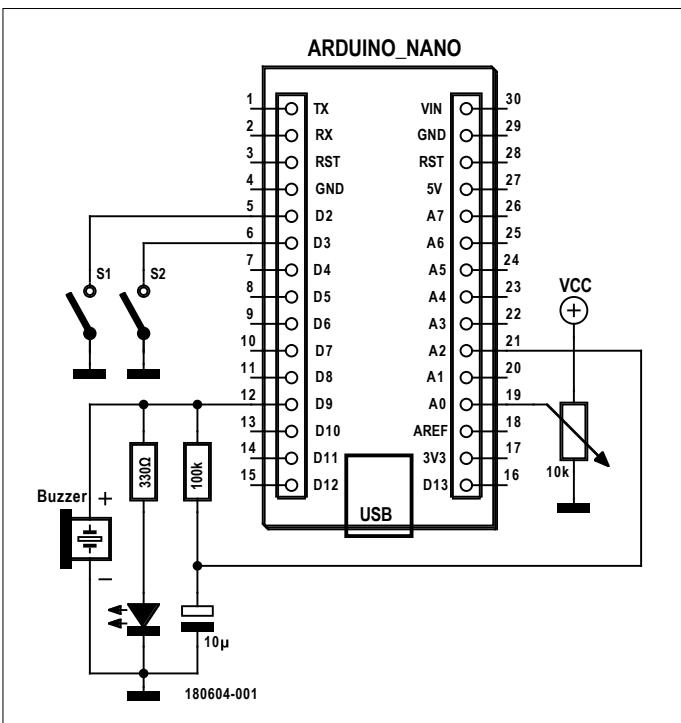
Zo langzamerhand krijg ik een beetje argwaan. Is het misschien helemaal geen piëzo-zoemer, zoals ik eerst vermoedde? Is het misschien een elektromagnetisch exemplaar? Ik hou een magneet erbij, die merkbaar aangetrokken wordt. Dat klopt, dergelijke onderdelen bestaan uit een magneetspoel met een transistor-generator en een magnetisch membraan. Ze trek-



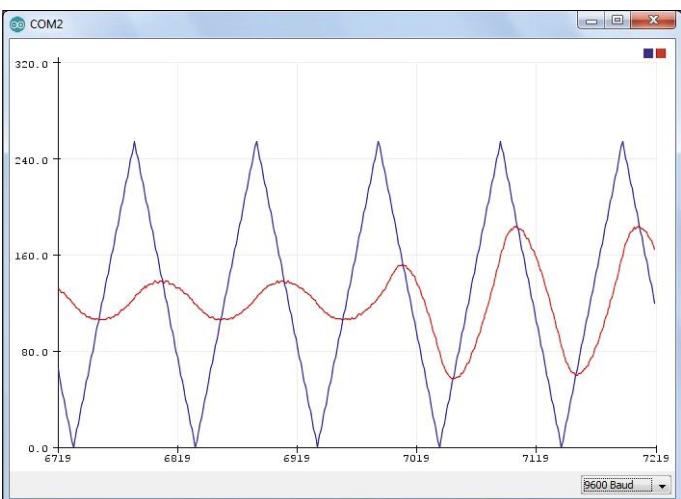
Figuur 4. De Nano instellen en laden in de IDE.



Figuur 5. Een probeersel.



Figuur 6. Uitbreiding met een laagdoorlaatfilter 100 k Ω / 10 μ F.



Figuur 7. Uitgangssignaal en gefilterde spanning.

ken veel meer stroom dan vergelijkbare piezo-zoemers. Dus daarom lukte het niet met die serieweerstand! Een poortpin van een ATmega328 levert echter stroom genoeg, zelfs genoeg voor zuinige DC-motoren zonder extra motor-drivers.

Zoemer aan de PWM-uitgang

Wat kunnen we verder nog proberen met al dat moois? Ik test een rode LED met een serieweerstand van 330 Ω aan D9 met de voorbeeldsketch *Fade.ino* uit de Arduino IDE. Dat werkt meteen. De LED werkt als ‘soft’ knipperlicht: hij wordt telkens helderder en dan weer donkerder. Dit alles dankzij PWM, dus door een poort snel aan en uit te schakelen.

Ik krijg steeds meer zin om te spelen. Kunnen we die zoemer ook gewoon direct op de PWM-uitgang aansluiten? Jawel, dat werkt, hoewel de zoemer eigenlijk niet daarvoor is bedoeld. In elk geval verandert de geluidssterkte in het tempo van de LED, en in enige mate ook de frequentie van de zoemer. En af en toe hoor je ‘mengproducten’ van de zoemerfrequentie en de PWM-frequentie.

Wat past er verder nog op het breadboard? Op het eerste gezicht wordt het al gauw erg vol, maar met wat passen en meten lukt het uiteindelijk allemaal toch. Twee extra grote drukschakelaars, een potmeter, de zoemer, de LED, een weerstand en wat bedrading; het past allemaal prima (**figuur 5**). De toetsen zitten maar met twee pootjes in het board, de andere twee hangen over de rand.

Dan breiden we het fade-voorbeeld wat uit. De rode knop moet de uitvoer stoppen en de zoemer uitschakelen. Met een druk op de blauwe knop gaat het weer verder. En met de potmeter stellen we de snelheid in. Zoals altijd werkt het eerst niet door een tikfoutje, en zoals altijd moet ik eerst iets nalezen (omdat ik normaal gesproken liever met Bascom werk). Maar tenslotte werkt alles zoals de bedoeling was. Dit project is weliswaar vrij zinloos en het eindresultaat is zelfs enigszins irritant (goed dat ik aan die stopknop S1 heb gedacht), maar alles bij elkaar is het toch een mooie vingerøefening. Bovendien is er de voldoening dat alles uiteindelijk functioneert zoals je voor ogen had.

IN DE STORE



→ JOY-iT Nano V3
www.elektor.nl/joy-it-nano-v3

→ Arduino Nano
www.elektor.nl/arduino-nano-3
→ Supplement-Kit
www.elektor.nl/arduino-supplement-kit

Boeken:

- Maik Schmidt, “Arduino” (Duitstalig)
www.elektor.de/arduino-dpunkt
- Bert van Dam, “Arduino ontdekken”
www.elektor.nl/arduino-uno-45-projecten-voor-beginners-en
- Michael Margolis, “Arduino Cookbook” (Engelstalig)
www.elektor.nl/arduino-cookbook
- Elektor Select, “Arduino Compilatie”:
www.elektor.nl/elektor-select-arduino-compilatie

Terwijl ik met het board aan het werk was viel me iets vreemds op waar ik niet meteen een verklaring voor had. Nadat ik een programma heb geladen, kan ik de USB-connector eruit halen en in plaats daarvan een USB-powerbank aansluiten. Zo wordt de applicatie draagbaar. Maar toen ik de kabel weer op de PC aansloot was het programma weg. De gele LED knipperde als bij een lege Nano. Wat was hier aan de hand? Met enig zoeken op het web vond ik het antwoord: het ligt aan de IDE in versie 1.6.8. Als die een nieuw aangesloten Arduino ontdekt, zet hij die direct in boot-modus. Waarom is me dat nog niet eerder opgevallen? Omdat ik liever met versie 1.6.5. werk: die is wel iets ouder maar heb je ook sneller startklaar.

Analoge signalen plotten

Soms zullen we echter toch versie 1.6.8. of hoger moeten gebruiken, vooral als we de seriële plotter willen gebruiken. Daarmee kunnen we veranderlijke grootheden weergeven. De plotter in versie 1.8.2. kan zelfs meerdere kanalen weergeven. Daarom is het programma nog een keer uitgebreid om de inhoud van de variabele *brightness* te plotten. Bovendien heb ik een laagdoorlaatfilter ($100\text{ k}\Omega / 10\text{ }\mu\text{F}$) toegevoegd om het PWM-singaal af te vlakken en te meten via analoge ingang A2 (zie **figuur 6** en **listing 1**). Het resultaat ziet u in **figuur 7**. De helderheid wordt aangestuurd met een driehoek-spanning, maar het RC-filter maakt daar een redelijke benadering van een sinus van. Na enige tijd heb ik aan de potmeter gedraaid om de uitvoer te vertragen. De periode van het signaal verandert ten opzichte van de constante tijdcyclische van het RC-filter. De amplitude van de ‘sinus’ wordt groter en het signaal wordt meer vervormd. Bij dergelijke experimenten heb je anders eigenlijk een hele batterij meetapparatuur nodig, maar de Nano doet dat zo ongeveer helemaal op eigen houtje.

ISP-programmeren

U hoeft overigens niet per se met de Arduino IDE te werken. In principe is elke ontwikkelomgeving voor AVR-controllers bruikbaar. De hex-file kunt u dan laden via de ISP-connector. Daarmee wordt dit boardje nog universeler. Zoals gezegd werk ik graag met Bascom. In de serie artikelen over de Arduino Uno (“Microcontrollers voor beginners”, vanaf Elektor april 2014) is een speciaal aangepaste Bascom-bootloader gebruikt, die ook op de Nano werkt omdat dezelfde ATmega328 en dezelfde klok van 16 Mhz worden gebruikt.

Listing 1. Het uitgebreide Fade-voorbeeld.

```
/*
Fade
This example shows how to fade an LED on pin 9
using the analogWrite() function.
This example code is in the public domain.
Additional: pot + s1 off + s2 on + AD + serial plot
*/
int led = 9;           // the pin that the LED is attached to
int brightness = 0;     // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by
int s1 = 2;
int s2 = 3;

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
  pinMode(s1, INPUT_PULLUP);
  pinMode(s2, INPUT_PULLUP);
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);
  Serial.print(brightness);
  int U = analogRead(A2);
  Serial.print(" ");
  Serial.print(U/4);
  Serial.println(" ");
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  // wait for ... milliseconds to see the dimming effect
  delay(analogRead(A0)/10+10);
  if (digitalRead(s1) == LOW){
    analogWrite(led,0);
    while (digitalRead(s2) == HIGH);
  }
}
```

Deze bootloader heb ik in de Nano geprogrammeerd met een STK500. En daarbij trad er een bijzonder probleem op. De Nano (zowel origineel als kloon) heeft een uitzonderlijk kleine pull-up-weerstand van $1\text{ k}\Omega$ aan de reset-pin, waardoor het

Weblinks

- [1] Driver voor de CH340G: www.wch.cn/download/CH341SER_ZIP.html
- [2] Website van de auteur: www.b-kainka.de/

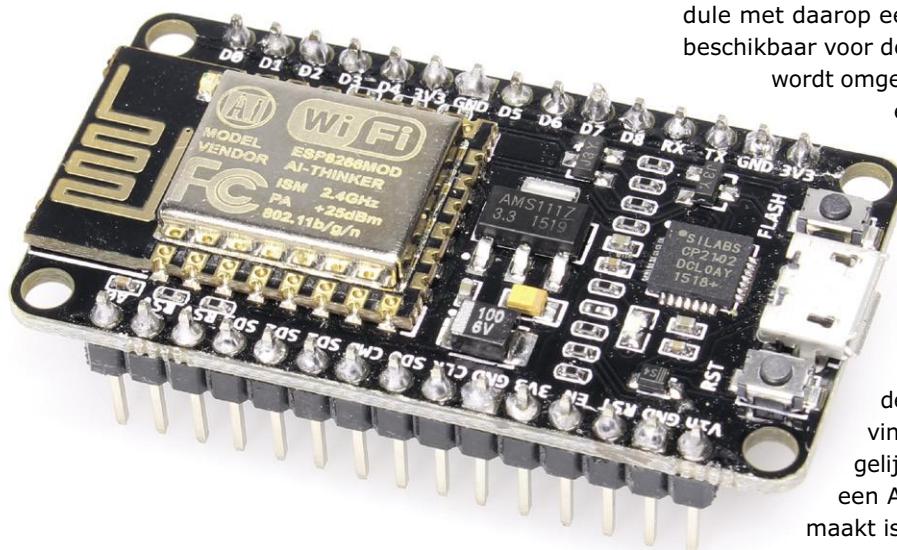
de STK500 te veel moeite kost om de pin stevig genoeg laag te maken. Daarvoor heb ik de volgende quick-and-dirty oplossing gevonden: tussen reset en GND zet ik extern nog eens 1 k. Daarmee wordt de reset nog net niet geactiveerd, maar met deze ondersteuning lukt het de programmer in de STK500

wel. Iets dergelijks zou je ook met andere programmers kunnen doen. Dus als het een keer niet lukt, kijk dan eens naar de spanning op de reset-ingang.

NodeMCU

Fabian Kainka (Duitsland)

NodeMCU is een ontwikkelboard gericht op het Internet of Things, gebaseerd op de 32-bits ESP8266-microcontroller van Espressif Systems.



Figuur 1. Het ontwikkelplatform NodeMCU.

Specificaties van de ESP8266.

- ESP12E-module
- 32 bits Low power CPU met 80 MHz klokfrequentie
- 4 MB flashgeheugen voor programma's en bestanden
- WiFi: 802.11 b/g/n tot 72,2 MBit/s
- Printantenne
- Voedingsspanning 3,3 V
- Programmeren en voeding via micro-USB-aansluiting
- USB naar serieel converter: DP2102
- 13 GPIO pinnen
- 10 bits ADC (successive approximation)
- SPI, I²C, I²S, 2 x UART, IRDA, PWM, RTC
- OnBoard LED, 2 druktoetsen

In **figuur 1** ziet u de hardware, met links een ESP12E WLAN-module met daarop een ESP8266. Er zijn bij elkaar 13 pennen beschikbaar voor de meest uiteenlopende toepassingen. USB

wordt omgezet naar serieel met een DP2102. Een LED

en twee druktoetsjes, waarvan één voor reset, zijn geïntegreerd op het board. Tot zover alles wat je van een ESP8266-ontwikkelboard mag verwachten.

NodeMCU is echter niet alleen de naam van het ontwikkelboard, maar ook van de vooraf geïnstalleerde firmware, met daarin een Lua-interpreter. Lua (Portugees voor maan) is een script-taal en eigenlijk is dat de reden waarom ik de NodeMCU interessant vind. In principe kun je een ESP8266 (en vergelijkbare boards) prima programmeren vanuit een Arduino IDE. Maar watLua zo interessant maakt is datLua-programma's in *runtime* draaien onder de interpreter. Je hoeft ze dus niet door een compiler te jagen. Dat maakt de software eenvoudiger: het is niks anders dan tekst die live, in runtime dus, door de interpreter begrepen en uitgevoerd wordt. U kunt het 't beste vergelijken met de console van Linux of de Power Shell in Windows, die commando's begrijpen en die script-files geautomatiseerd kunnen uitvoeren. En net als daar is de ontwikkelomgeving niet meer dan een simpele tekst-editor. En we kunnen het onszelf nog makkelijker maken, zoals we straks zullen zien.

De eerste stappen

De eerste stap was dat ik het board uit de verpakking gehaald heb en heb aangesloten op een USB-poort op mijn PC. Voor de eerste seriële verbinding heb ik het terminalprogramma Putty gebruikt, maar het kan net zo goed met elke andere seriële monitor. Toen verbinding was gemaakt op 9600 baud kon ik de eerste leesbare tekens ontvangen. Die onthulden dat het board was uitgerust met firmware versie 0.9.6, gecompileerd medio 2015. Dat vond ik een beetje te oud, dus ik ging op zoek naar de nieuwste versie. Daarbij vond ik direct een aantal interessante dingen.

Nieuwe firmware

Mijn zoektocht leidde me naar de overeenkomstige Github-site [1] (Github is een internetplatform voor broncode.) Hier staat alle belangrijke informatie bij elkaar en ik hoopte daar de nieuwste firmware te kunnen downloaden. Mijn verrassing was echter groot, want je vindt hier namelijk niet zomaar nieuwe firmware. In plaats daarvan kan elke gebruiker zijn eigen firmware samenstellen. Geheel naar eigen smaak en behoeft kunnen we inmiddels kiezen uit 65 verschillende functiemodules die we kunnen integreren in de Lua-firmware. Die modules passen natuurlijk niet allemaal tegelijk in het 4 MB grote flashgeheugen; dat zou net zoets zijn als dat je alle Arduino-libraries tegelijk wilt gebruiken. Maar dat betekent dus dat je als gebruiker al van tevoren een idee moet hebben welke kant je op wilt met je project.

In vrijwel iedere firmware-configuratie heb je standaardmodules nodig zoals een timer, UART, WiFi, GPIO en zo meer, terwijl je modules als HTTP, MQTT, SNTP en allerlei sensor-modules zoals DHT (temperatuur en luchtvuchtigheid), BME (idem, met luchtdruk), HMC5883L (digitaal drie-assig kompas) of TCS34725 (kleur- en lichtsensor) alleen opneemt als je die echt nodig hebt.

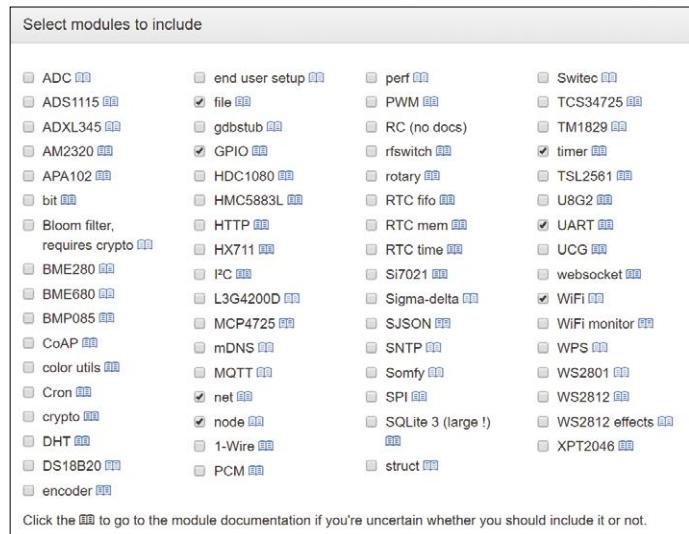
Dat mag dan allemaal klinken als een heleboel mogelijkheden, het betekent ook een heleboel werk om uit al die modules iets zinigs samen te stellen en tot bruikbare firmware te compileren. En inderdaad zou dat lastig zijn, ware daar niet de **Cloud Build Service (figuur 2)** [2] die heel makkelijk te gebruiken is. Daar kunnen we namelijk onze persoonlijke firmware met alle vereiste modules gewoon "bij elkaar klikken". En kunnen we als extra optie bijvoorbeeld aangeven welke branch (tak) van Github we willen gebruiken en of we TLS/SSL-ondersteuning of FatFS voor het lezen van SD-kaarten nodig hebben. Hebt u eenmaal uw keuze gemaakt, dan geeft u uw e-mailadres op en klikt u op **Start your build**. Het samenstellen van de firmware duurt een tijdje (drie minuten bij mij), en dan krijgt u een mail met een downloadlink die 24 uur geldig blijft.

Deze voorgecompileerde firmware moet nu alleen nog in het board worden geladen. Ook daarvoor is er een handige tool, namelijk de **NodeMCU PyFlasher (figuur 3)** die u via [3] kunt downloaden voor Windows of voor MacOS. Installeren is niet nodig, het is een direct uitvoerbaar programma (*executable*). Vervolgens kiest u de juiste COM-poort en de zojuist gedownloade firmware-file, en start u de upload. Bij mij ging dat allemaal zonder problemen. Een uitvoerige beschrijving van het samenstellen en flashen van de firmware vindt u op de uitstekende documentatie-site [4]. U vindt daar ook allerlei informatie over modules en fundamentele vragen en antwoorden in FAQ's.

Hello World – Lua-test

Op die documentatie-site vond ik ook een link [5] naar de zeer nuttige tool **ESPlorer**. Dit programma (**figuur 4**) is geen eenvoudige seriële monitor, maar meer een soort Zwitsers zakmes voor de ESP8266, en het maakt niet uit of je MicroPython, NodeMCU of andere AT-software gebruikt.

ESPlorer.jar is een executable. Toen ik die had gedownload en gestart heb ik opnieuw serieel verbinding met het board gemaakt. Met de nieuwe firmware was de baudrate dit keer 115200. Het programma probeerde nu automatisch met het board te communiceren, maar zonder resultaat. Pas toen ik de reset-toets op het board had ingedrukt kreeg ik de eerste



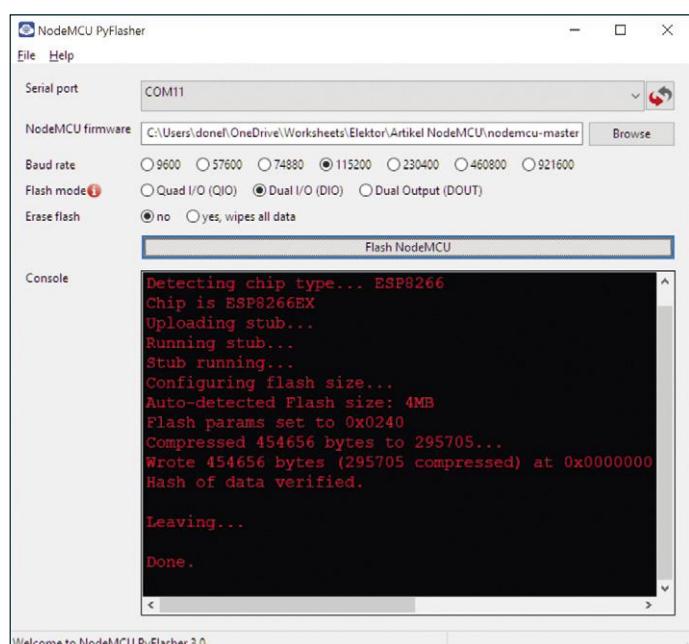
Figuur 2. Firmware samenstellen uit alle beschikbare modules.

berichten met de gebruikte firmware plus alle modules. Dus de communicatie werkte.

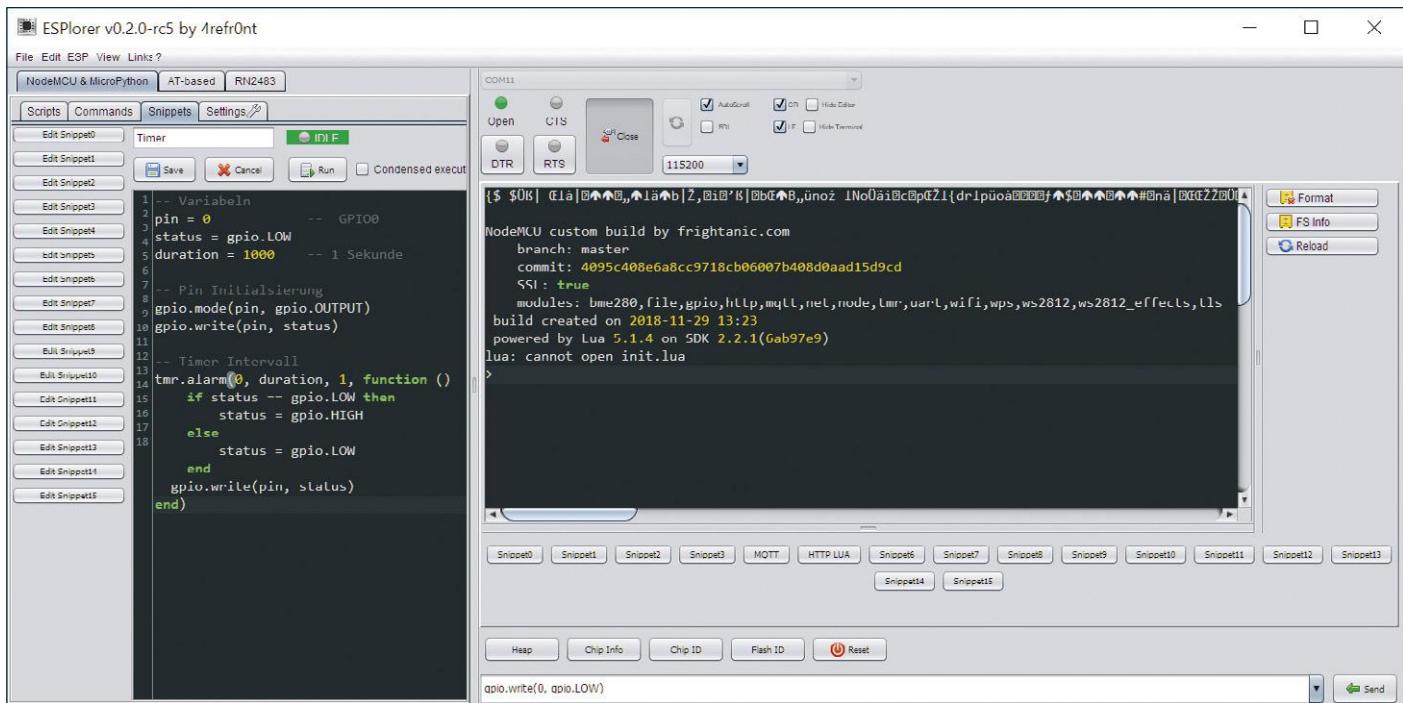
In ESPlorer heb je een paar knoppen aan de rand van het venster, die automatische commando's naar het board sturen. Zo kon ik bijvoorbeeld het chip-ID en informatie over het file-systeem uitlezen. Verder is er uiteraard ook een venster waar je direct commando's kunt invoeren. Ik heb als eerste deze commando's geprobeerd:

```
gpio.mode(0, gpio.OUTPUT)  
gpio.write(0, gpio.LOW)
```

Dit doet denken aan Arduino-broncode, wat deze commando's heel goed leesbaar maakt. Eerst wordt hier pin 0 als uitgang gedefinieerd en vervolgens op nul (LOW) gezet. Aan dit pootje hangt ook de LED op het board, die naar VCC is geschakeld en



Figuur 3. PyFlasher.



Figuur 4. ESPlorer, rechts de terminal, links de snippets.

daarom nu dus oplicht. Dat was een eerste simpele test of de commando's ook echt werken, zonder compileren. Dit venster voor handmatige invoer is heel handig als je maar één of twee commando's hoeft in te voeren, ook omdat de historie wordt onthouden, en het dus heel makkelijk is om een commando opnieuw uit te voeren. Maar in de meeste gevallen wil je meteen een hele reeks van commando's uitproberen. Ik wilde het klassieke "Hello World"-voorbeeld implementeren en een LED laten knipperen. Dat kost al een paar regels meer, maar gelukkig kun je met ESPlorer complete stukjes code samenstellen en zelfs opslaan op de PC. Deze script-knipsels heten *snippets* en je kunt ze met één klik naar het board sturen. Onderstaand knipperprogramma heb ik ingevoerd in één van de 15 snippets die je kunt bewerken, en vervolgens heb ik het gestart met *Run*.

```
-- Variables
pin = 0          -- GPIO0
status = gpio.LOW
duration = 1000   -- 1 sec
-- Pin Initialization
gpio.mode(pin, gpio.OUTPUT)
gpio.write(pin, status)
-- Timer Intervall
tmr.alarm(0, duration, 1, function ()
    if status == gpio.LOW then
        status = gpio.HIGH
    else
        status = gpio.LOW
    end
    gpio.write(pin, status)
end)
```

U ziet dat variabelen niet eerst gedeclareerd hoeven te worden. Als u al eens iets in JavaScript heeft geprogrammeerd dan komt u dat waarschijnlijk bekend voor. Ook interessant is hoe we met de timer omgaan. In dit geval geven we de timer simpelweg een complete direct in de aanroep gedefinieerde functie (een zogenaamde anonieme functie) mee die door de timer elke seconde moet worden uitgevoerd. Het was dus heel eenvoudig om een klassiek knipper-LED'je te maken, het programma is heel inzichtelijk. Maar omdat het hier eigenlijk een IoT-board betreft, heb ik de volgende snippet gemaakt om een verbinding met mijn router te maken:

```
wifi.setmode(wifi.STATION)
station_cfg={}
station_cfg.ssid="YourSSID"
station_cfg.pwd="YourPassword"
station_cfg.save=true
station_cfg.auto=true
wifi.sta.config(station_cfg)
wifi.sta.connect()

wifi.eventmon.register(wifi.eventmon.STA_CONNECTED,
    function(T)
        print("WiFi connected!")
    end)
```

Nadat de verbinding succesvol tot stand was gebracht, geeft het programma de melding "WiFi connected!" Daarvoor moet ik weer een anonieme functie maken, die wordt uitgevoerd als de gebeurtenis "verbonden met WiFi" optreedt. Deze manier van gebeurtenis-gestuurd (*event driven*) programmeren is kenmerkend voor Lua.

Het opzetten van de verbinding verliep probleemloos. De confi-

guratie die ik hier gekozen heb zorgt er bovendien voor dat de WiFi-data wordt opgeslagen en ook na een reset nog behouden blijven. Bovendien zorgt `station_cfg.auto=true` ervoor dat het board steeds uit zichzelf verbinding maakt met het opgeslagen netwerk. Zonder corresponderend programma verschijnt de melding "Wifi connected" echter niet, want dit gedeelte bewaart het board slechts tot de eerstvolgende reset.

Aangezien de verbinding zo mooi werkte, wilde ik meteen de MQTT-module testen. In de documentatie stond een goed voorbeeld dat ik direct in een snippet heb gekopieerd, een beetje heb aangepast en "eventjes" live op het board heb getest [6]. Ook dit verliep allemaal ongewoon soepel. Waardoor ik zin kreeg om grotere dingen te gaan proberen.

Daarom wilde ik als volgende stap een programma maken dat permanent op het board opgeslagen blijft en automatisch opstart zodra het board wordt ingeschakeld. En ook dat was heel makkelijk, vooropgesteld dat we het volgende in aanmerking nemen:

- de firmware stelt een eigen bestandssysteem ter beschikking waar we bestanden naartoe kunnen uploaden. Dat kunnen script-files zijn, die dan met het commando `dofile(filenaam)` worden gestart. Maar het kunnen ook websites zijn, of plaatjes of andere bestanden die u op het board bij de hand moet hebben;
- bij het opstarten kijkt het board of het bestand `init.lua` bestaat. Is dat het geval, dan wordt dat automatisch uitgevoerd.

Concreet betekende dat voor mij dat ik in de tekst-editor van ESPlorer gewoon moest omschakelen van het tabblad Snippets naar het tabblad Scripts. Daar schreef ik de volgende regels, die ik meteen op mijn PC opsloeg met de naam `init.lua` en vervolgens met Upload in de NodeMCU zette:

```
wifi.eventmon.register(wifi.eventmon.STA_CONNECTED,
    function(T)
        print("WiFi connected!")
        http.get("https://pastebin.com/raw/k4ccGx3T",
        nil, function(code, data)
            if (code == 200) then
                if file.open("run.lua","w+") then
                    file.write(data)
                    file.close()
                end
            else
                print("HTTP request failed")
            end

            if file.exists("run.lua") then dofile('run.
            lua') else print("run.lua not found") end
            end)
        end)
```

Wat hier gebeurt kan eigenlijk alleen met een script-taal. Als eerste wacht het programma tot er verbinding is met een WiFi-net. Is die er, dan haalt het in runtime een ander programma op van internet, slaat dat op met de naam run.lua en lanceert

Weblinks

- [1] Git-repository: <https://github.com/nodemcu/nodemcu-firmware>
- [2] Firmware samenstellen: <https://nodemcu-build.com/>
- [3] PyFlasher: <https://github.com/marcelstoer/nodemcu-pyflasher/releases>
- [4] Documentatie: <https://nodemcu.readthedocs.io/>
- [5] ESPlorer: <https://esp8266.ru/esplorer/#download>
- [6] MQTT-voorbeeld: <https://nodemcu.readthedocs.io/en/master/en/modules/mqtt/#example>
- [7] Website van de auteur: <https://fkainka.de/>

het. In dit voorbeeld is het een simpel knipper-LED'je dat van *pastebin* wordt gehaald, maar u kunt hier natuurlijk allerlei andere bronnen gebruiken. In deze firmware worden zowel de HTTP-module en de TLS/SSL-optie gebruikt.

Hier wordt de interpreter-taal Lua echt spannend. Ik stel me zo voor dat je een hele serie programma's ergens op een webserver neerzet die je daar steeds aanpast en verbetert. Met behulp van een website of via MQTT kiest men dan zijn of haar programma en laadt dat in het board. Nog meer modular en simpeler kan het bijna niet. ▶

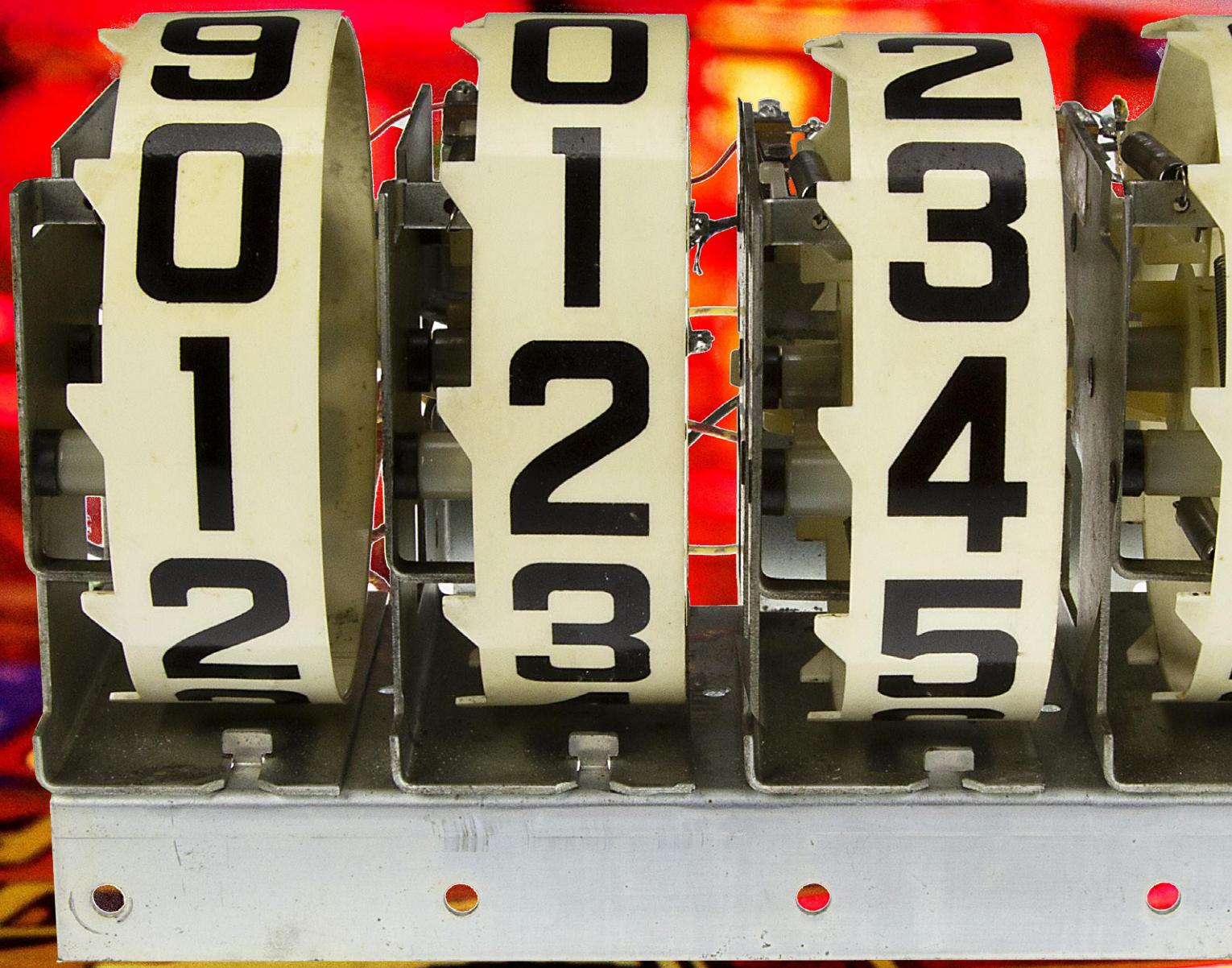
180604-03



IN DE STORE

- **NodeMCU-Mikrocontrollerboard**
www.elektor.nl/nodemcu-microcontroller-board-with-esp8266-and-lua
- **Erik Bartmann: "Das ESP9266-Praxisbuch"** (Duitstalig)
www.elektor.nl/das-esp8266-praxisbuch
- **Dogan en Ahmet Ibrahim: "ESP8266 and MicroPython"** (Engelstalig)
www.elektor.nl/esp8266-and-micropython

versla de



Elektor- Flipper- klok!

0
1
2

PROJECT-INFO	
	pinball retro ESP12-F ESP8266 eBay
	beginners ➔ gevorderden experts
	ca. 2 uur (exclusief schoonmaken van de tellers)
	3,3 V FTDI-kabel of andere USB/UART-interface
	ca. € 75 (exclusief cijferwielen)

aaah... dat geluid van rategende cijfers

Luc Lemmens (Elektor Labs)

Ouderwetse displays trekken gegarandeerd de aandacht, en niet alleen omdat ze er leuk uitzien. In dit project, waar we vrolijk techniek uit de jaren '60 combineren met WiFi en SMD, produceren ze ook de nodige **herrie** – hiermee verstrijkt de tijd hoorbaar..

Zeker, Elektor heeft talloze projecten met 'vintage' displays zoals Nixie-buizen, Numitrons, Dekatrons en VFD-buizen gepubliceerd, en ook in de vaste Retrotronica-rubriek zijn ze regelmatig ter sprake gekomen, maar tot op heden was er pas één thermometerproject dat gebruikmaakte van elektromechanische (EM) flipperkast-cijferwielen [1], gevuld door een nieuwe versie, waarin een 'credit'-wiel werd gebruikt om de temperatuur aan te geven [2]. Hoog tijd voor een vervolg! Dit keer maken we een klok met cijferwielen. Misschien niet de meest originele toepassing voor hergebruik van oude displays, maar wel een met een hoog bruikbaarheidsgehalte én een hoge WAF (*Wife Acceptance Factor*), dat wil zeggen tot 12:00 uur, maar daarover later meer.

Om recht te doen aan de oorspronkelijke toepassing van deze displays in bars, kelders en café's gaat in dit project ook een simpele flipperkast schuil waarop via computer of smartphone kan worden gespeeld. Zo worden de wielen gebruikt voor hun oorspronkelijke doel: het bijhouden van de score.

Op het internet is een overvloed aan flipperkast-klokken te vinden, maar de onze zal ongetwijfeld hoog scoren omdat het een geavanceerd en anachronistisch ontwerp betreft, met een ESP8266 die de klok via WiFi verbindt met het internet voor tijd- en datumsynchronisatie. De klok kan via WiFi worden gelijkgezet vanaf PC, smartphone of tablet. Tijd en datum kunnen ook handmatig worden ingesteld terwijl een DS3231 (Real Time Clock) met backup-batterij de tijd bijhoudt met een nauwkeurigheid van 5 ppm, zelfs als de voeding uitvalt.

Per dag (of liever: nacht) kan een periode worden ingesteld waarin de teller-wielen en klokalarmen stil blijven. De klok

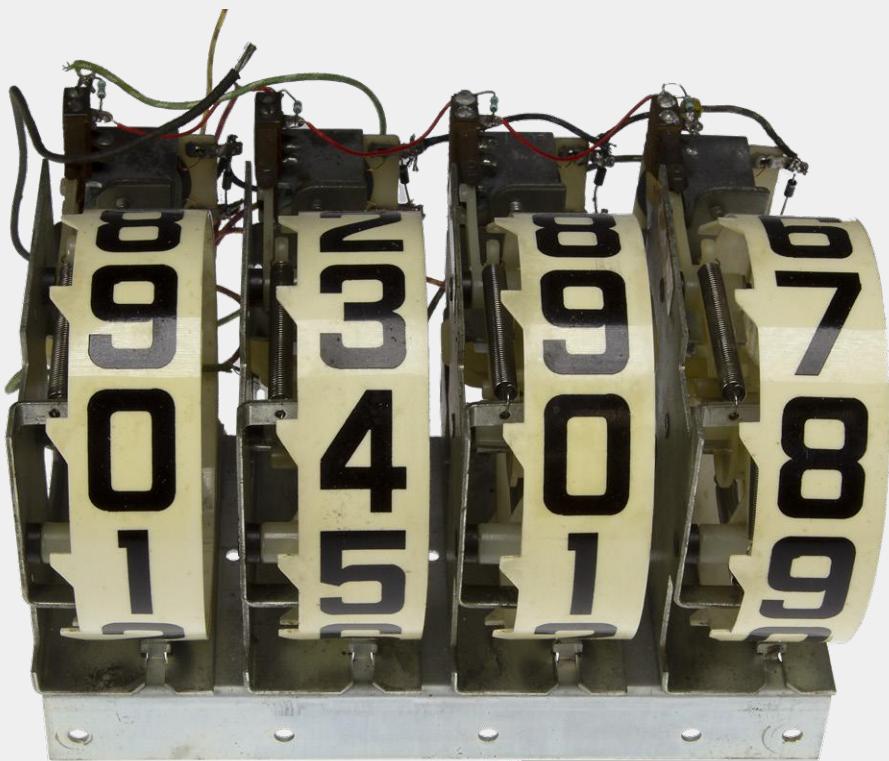
kan ook met de hand worden in- en uitgeschakeld (maar u kunt natuurlijk ook gewoon de stekker eruit trekken...). Tijdens normaal bedrijf kan de tijd in 12- of 24-uurs formaat worden weergegeven, terwijl de datum iedere 30 minuten wordt getoond (in Europees of US-formaat).

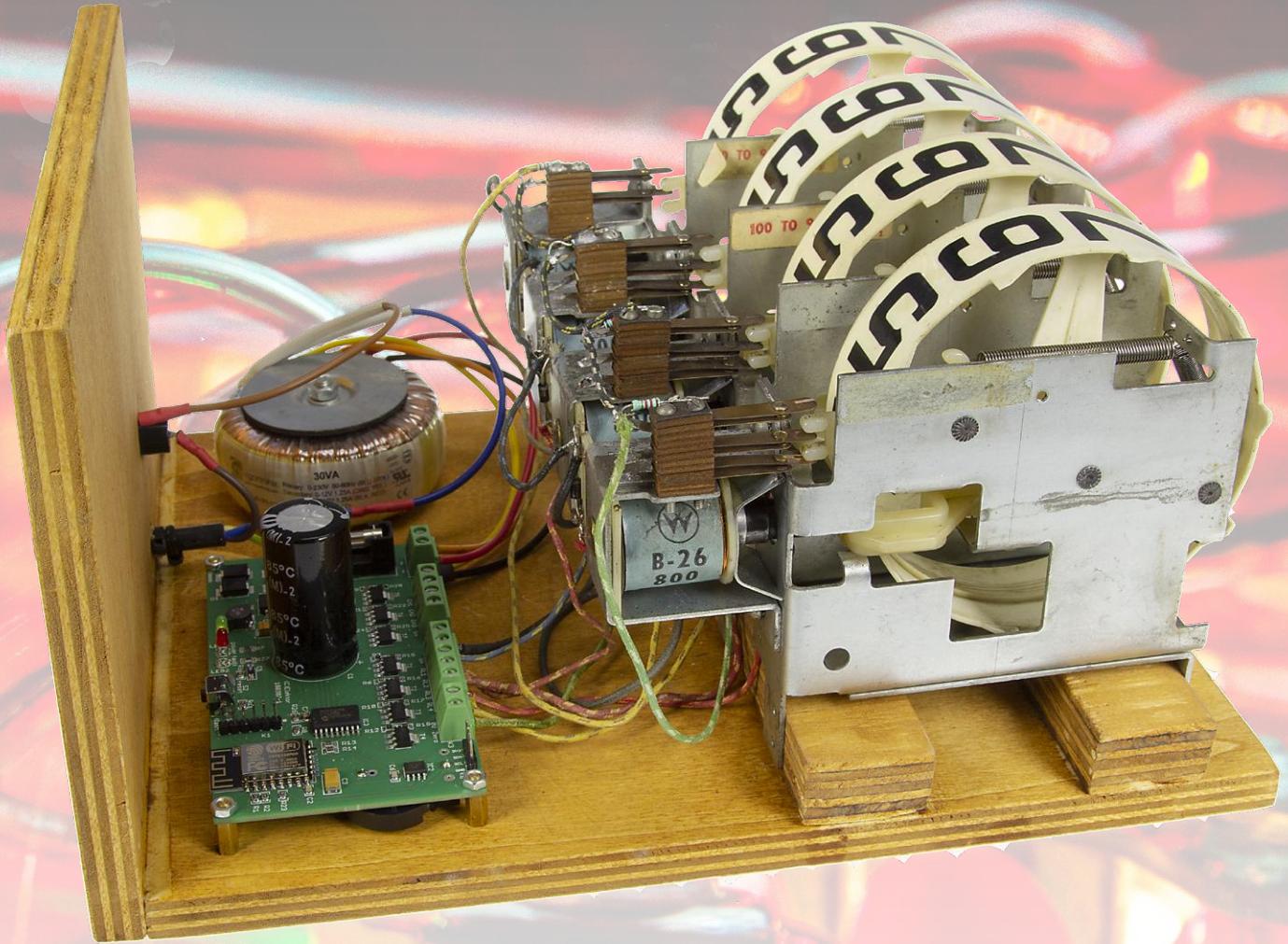
Toen de wereld nog elektromechanisch was

De meeste in de jaren '50 tot midden jaren '70 gebouwde flipperkasten maakten gebruik van elektromechanische cijferwielen (*score drums*) waarop de score van de speler werd weergegeven. Het

werkingsprincipe is door de jaren heen nauwelijks veranderd: er wordt een korte puls op een spoel gezet, waardoor het wiel één stapje verder draait. Een schakelaar detecteerde de nul-positie, zodat de flipperkast kon worden gereset aan het begin van een nieuw spel.

De werkelijkheid is altijd iets ingewikelder. Een cijferwheel bevatte meer schakelaars, bijvoorbeeld één om te controleren of een stap is voltooid, één die aangeeft dat het cijfer '9' is bereikt zodat de volgende decade kan worden opgehoogd; en sommige tellers registreerden zelfs elke positie, voor vrij-spel scores en de





Waar koop je cijferwielen, en welke moeten het zijn?

Elke marktplaats of gespecialiseerde site is een goede bron, maar er zijn ook (web)shops met gebruikte of opgeknapte onderdelen. Alle cijferwielen van bekende merken zoals Gottlieb, Bally en Williams zijn voor onze klok te gebruiken, maar er zullen zeker andere merken zijn die het ook doen.

Voor het aanzien verdient het aanbeveling vier identieke wielen te gebruiken; ook zijn er wielen met diverse afmetingen en lettertypen, afhankelijk van merk en leeftijd van de tellers.

Ontkolen

Zodra u de vier tellers op je werktafel hebt liggen, begint het vervelendste, maar ook dankbaarste klusje: schoonmaken! Hoogstwaarschijnlijk gaan de wielen schuil onder vuil, stof en roet (van daar dat ontkolen) en dat ziet er niet alleen lelijk uit, maar doorgaans moeten de mechanische delen worden gereinigd voordat ze weer goed functioneren. Doe uzelf een plezier en haal ze helemaal (!)

uit elkaar en maak ze grondig schoon voordat u ze weer in elkaar zet. Als u alleen 'op het oog' schoonmaakt werkt het straks waarschijnlijk niet.

Enkele *do's* en *don'ts* op basis van jarenlange ervaring:

- het belangrijkste: wees voorzichtig met de wielen, de belettering kan makkelijk worden beschadigd of zelfs uitgewist als ze niet met zorg worden schoongemaakt. De vroege Gottlieb-wielen zijn kennelijk bedrukt met een inkt op waterbasis... Over het algemeen: niet weken, schrobben of krabben en geen (agressieve) schoonmaakkmiddelen gebruiken;
- bij het weer in elkaar zetten: er is altijd een nokje of markering of zoets voor de juiste oriëntatie van het wiel op de as; onjuiste montage kan de hele module beschadigen en verschuift de nul-positie van het wiel;
- bewegende delen hebben metalen of kunststof lagers, als een wiel

niet lekker loopt hebt u het óf niet goed schoongemaakt, óf verkeerd gemonteerd. Gebruik in geen geval smeermiddelen!

- nulschakelaars zijn niet moeilijk te identificeren: zoek naar een schakelaar die open als het wiel in de nulstand komt. Bij recentere Bally- en Williams-wielen is het makkelijk: het is de schakelaar bovenop de module. In dat opzicht zijn de Gottlieb-machines veel minder reparatievriendelijk, met de schakelaars verborgen in de module. Zoek hier naar de nulschakelaar als je de module uit elkaar hebt liggen voor reiniging. In alle gevallen: maak de contactpunten schoon met heel fijn schuurpapier en regel indien nodig ook de contactveren af.

Op het internet is een video te vinden met instructies voor het (de-)monteren, schoonmaken en repareren van cijferwielen. Maar waar zijn al die bellen gebleven?

'match'-optie. Maar in ons project gebruiken we die allemaal niet, voor de klok zijn alleen de spoel en de nulschakelaar van belang.

In de loop der jaren was er wel een belangrijke verandering bij die cijferwielen: de diameter werd teruggebracht van 5" in de jaren '60 tot 4" en later zelfs 3". De traagheid van de grote tellers beperkte namelijk de timing van de score-weergave en kleinere cijferwielen

zijn nu eenmaal sneller. Dat is waarom oudere flipperkasten langzamer klinken – minder 'scherp' – dan moderne varianten. Om de beleving van de echte flipperkast (waarvoor de cijferwielen immers waren bedoeld) zo goed mogelijk te benaderen, kan ons klok-display aan de diameter van de cijferwielen worden aangepast.

Flipperkastellers kunnen alleen optellen, dus als een '1' een '0' moet worden, kan

dat alleen door negen pulsen toe te dienen. Het zal duidelijk zijn dat dit ongetwijfeld de rumoerigste klok zal zijn die Elektor ooit heeft gepresenteerd. Om hier echt helemaal zeker van te zijn, en voor de echte flipperaars onder ons: optioneel kan een echte drietonige flipperbel worden aangesloten voor het aangeven van het kwartier, halve en hele uur.

Dat wordt dan meteen ook het duurste onderdeel van onze klok, want deze bellen zijn steeds moeilijker te vinden, en worden dus steeds duurder. Als alternatief is ook een simpele bel van een oudere flipperkast bruikbaar.

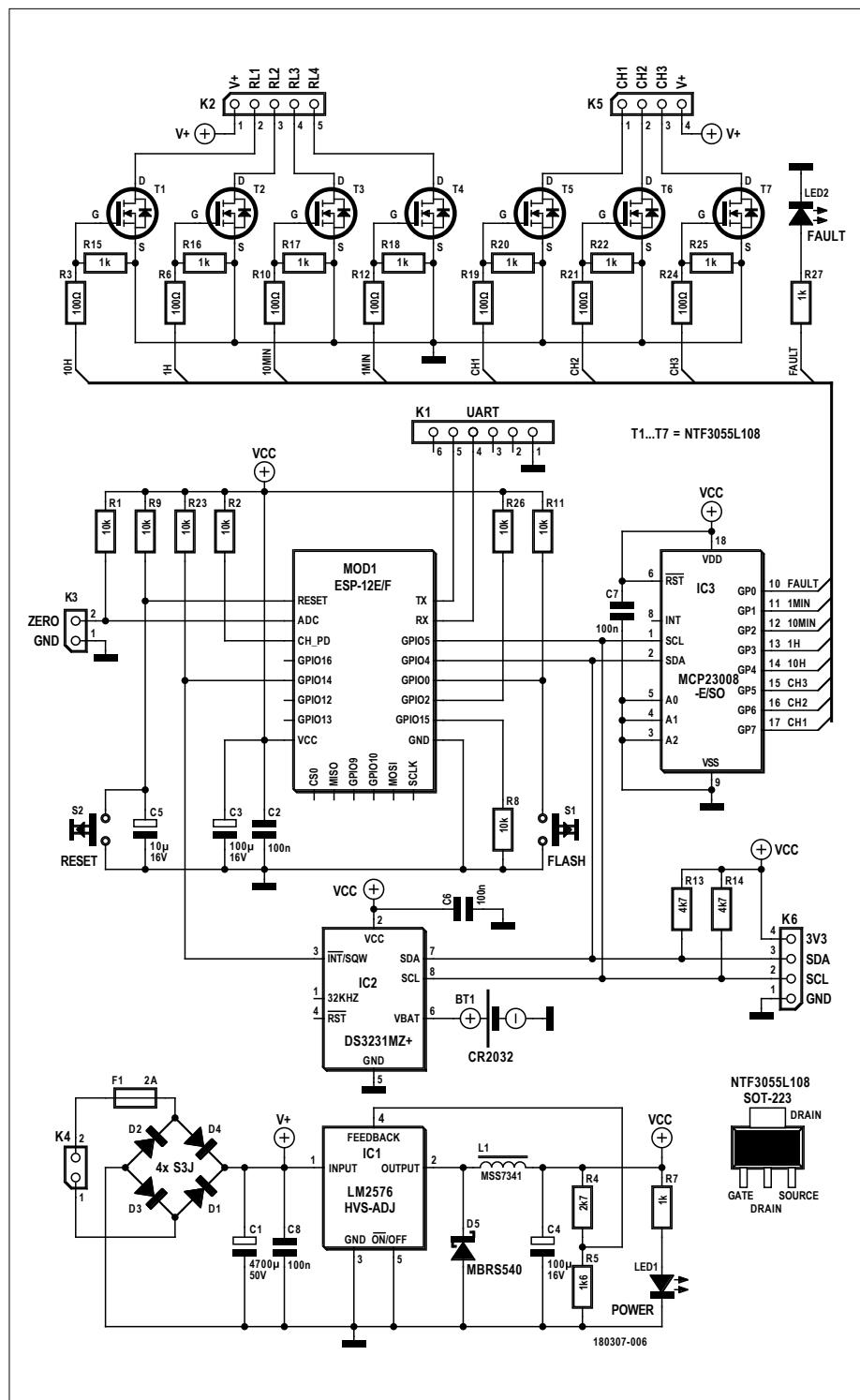
De spoelen in elektromechanische flipperkasten worden met wisselstroom bekrachtigd, behalve die van de flippers en de zogenaamde 'pop bumpers' in latere modellen. AC-spoelen kunnen ook met DC worden bekrachtigd, maar dan wel met een lagere spanning dan aangegeven voor AC [3]. Let wel: in dat document gaat het over het permanent bekrachtigen van AC-spoelen met DC. In onze klok en in de meeste flipperkasten krijgen de spoelen slechts korte pulsen te verduren, waardoor oververhitting niet erg waarschijnlijk is. Dat kan alleen gebeuren als een spoel permanent wordt bekrachtigd ten gevolge van een fout in de klok. In dat geval brandt een zekering door, zodat de stroom wordt onderbroken voordat een spoel geroosterd wordt. In een echte flipperkast veroorzaakt een vastzittende score-schakelaar een stinkende, verbrande en massief kortgesloten spoel en vliegt de 10-A-zekering eruit. Kortom: deze spoelen zijn nooit ontworpen om permanent te worden bekrachtigd!

Behalve in de allereerste Williams-machines (van voor 1963: 50 VAC) bedroeg de spoelspanning in zowel de Williams- als de Gottlieb-flipperkasten uit de EM-periode 28 VAC. Alle EM-kasten van Bally gebruikten 50 VAC. Alle tellers werken goed bij een spanning van ca. 35 VDC, hoewel een lagere spanning (zo'n 20 VDC) voor Gottlieb en Williams al voldoende is.

En dan nu de elektronica

Het schema van de flipperklok in **figuur 1** is niet echt ingewikkeld, maar we nemen toch even alle secties in detail door.

De voeding van onze klok is recht-toe-rechtaan. Een 2x12-V ringkerntrafo met de secundaire wikkelingen in serie is verbonden met K4 en levert de 32 VDC spoelspanning op de positieve aansluiting



Figuur 1: Schema van de flipperklok. Met of zonder bellen is dit een **luidruchtige** schakeling!

van de bruggelijkerichter D1...D4. Dat wil zeggen in theorie: in de praktijk zullen de onbelaste trafowikkelingen een spanning produceren die veel hoger is dan 35 VDC. Voor de Williams- of Gottlieb-tellers volstaat een 2x9-V trafo.

IC1 is een LM2576 stepdown-converter die de 3,3 V levert voor de digitale sectie van de flipperklok. Om aan de veilige kant te blijven kozen we voor de HV-versie die een maximum ingangsspanning van 60 V kan hebben, terwijl de 'normale' versie slechts 40 V kan verdragen. LED1 licht op als de 3,3-V-voeding is ingeschakeld.

Een opmerking over elektrolytische condensator C1, één van de weinige 'through-hole'-componenten in dit project. Dit lijkt misschien een vreemde eend op de print, maar het is een bewuste keuze. Om te beginnen zijn dikke SMD-elco's voor spanningen van meer dan 35 V moeilijk, zo niet onmogelijk te vinden. Bovendien is een exemplaar met aansluitdraden beter uit een oogpunt van mechanische stabiliteit en – het belangrijkste! – alom verkrijgbaar met werkspanningen van 50 V of hoger. De spoelen van de cijferwielen en de bellen worden aangestuurd door NTF3055 N-channel MOSFET's, die een voldoende lage V_{th} hebben om de spoelen betrouwbaar te kunnen pulsen vanuit logische 3,3-V-uitgangen. In ons eerste prototype werden deze FET's direct aangestuurd door de GPIO's van de ESP8266. Dat bleek een slechte keus: de ESP8266 heeft niet zo veel I/O-pennen en sommige daarvan hebben speciale functies tijdens opstarten, waardoor het bijzonder lastig wordt de spoelen volledig spanningsloos te houden tijdens de start. IC3, een MCP2300 I²C I/O-expander maakt de hardware een stuk eenvoudiger aan te sturen en biedt bovendien nog enige bescherming en buffering tussen de ESP en de power-FET's.

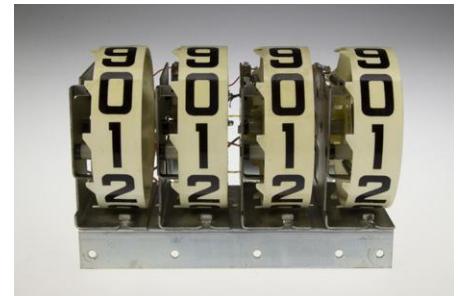
In het eerste ontwerp waren de vier nulschakelaars verbonden met de ADC van de ESP om I/O-pennen uit te sparen. Hoewel de ADC niet echt nauwkeurig is, werkte deze oplossing goed. De ESP kan alleen een verandering van de ADC-waarde vaststellen als een nulschakelaar opent nadat de teller de nulpositie heeft bereikt. Als gevolg hiervan zullen bij een complete reset alle tellers draaien, ook al stonden ze al op nul.

Over alle nulschakelaars moet een 1,2-kΩ-weerstand worden gesoldeerd (zie alvast **figuur 2**). De schakelaars

staan in serie en zijn verbonden met K3. Als na 12 opeenvolgende pulsen op een cijferwiel geen verandering van de ADC-waarde optreedt, gaat 'FAULT'-LED2 branden en worden pulsen naar dit wiel onderdrukt totdat de klok is gereset. Maar controleer eerst de nulschakelaars en hun bedrading voordat u dat doet. Dit mechanisme is niet alleen ingebouwd ter bescherming van geplaatste zenuwen, maar ook om de spoel te beveiligen tegen oververhitting als gevolg van aanhoudend pulsen naar een nulstand die nooit bereikt zal worden.

Op K6 kan een I²C-OLED-display worden aangesloten waarop statusinformatie wordt weergegeven. Deze connector kan ook worden gebruikt voor toekomstige uitbreidingen, zoals een I²C-temperatuursensor.

S1 is de resetknop. Als u S2 ingedrukt houdt totdat S1 wordt losgelaten, start de ESP8266 in bootloader-modus, waardoor de klok opnieuw kan worden geprogrammeerd via UART-connector K1. Met S2 kunnen we ook terug naar default (fabrieksinstelling), bijvoorbeeld als de klok met een ander WiFi-netwerk moet worden verbonden. Dat gaat als volgt: reset de klok en druk S2 in terwijl de

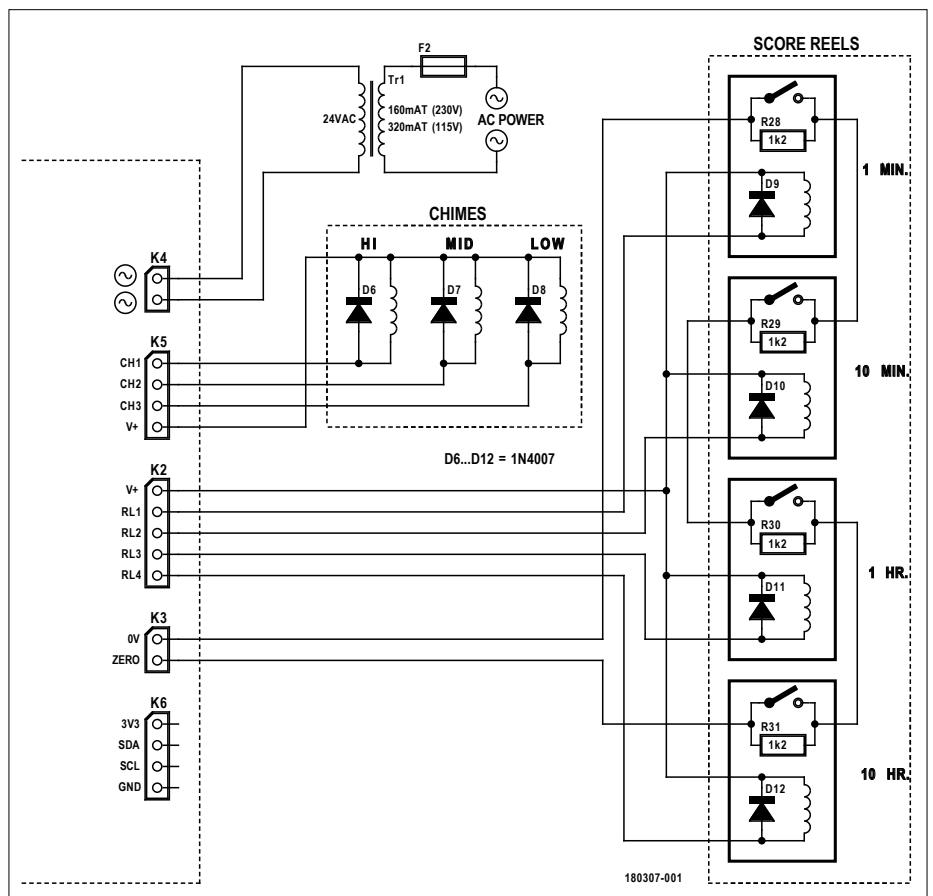


rode 'FAULT'-LED brandt.

IC2 is de zeer nauwkeurige (wow, 5 ppm!) DS3231 I²C Real Time Clock. Een CR2032 knoopcel zorgt ervoor dat de juiste tijd behouden blijft als de flipperklok wordt uitgeschakeld. Hij hoeft dus niet iedere keer opnieuw te worden ingesteld na inschakelen of na verlies van de NTP-synchronisatie.

Bouw en bedrading

Eerst moet u aan vier cijferwielen op de kop zien te tikken voordat u de andere hardware voor dit project aanschaft. Een drietonige bel of eventueel een enkeltonige flipperbel is een leuk extraatje, maar niet noodzakelijk om deze klok te bouwen; en zelfs als u ze wel hebt, is de kans groot dat u ze vanwege de herrie al snel het zwijgen oplegt...



Figuur 2: Bedradingsschema van de flipperklok.

Bestel de kit via GroupGets.com:
www.elektormagazine.com/pinball-clock



ONDERDELENLIJST

Weerstanden:

R1,R2,R8,R9,R11,R23,R26 = 10 k, dikke film,
 5%, 0,1 W, 150 V
 R3,R6,R10,R12,R19,R21,R24 = 100 Ω, dikke
 film, 5%, 0,1 W, 150 V
 R4 = 2k7, dikke film, 5%, 0,1 W, 150 V
 R5 = 1k6, dikke film, 1%, 0,1 W, 150 V
 R7,R15,R16,R17,R18,R20,R22,R25,R27 = 1 k,
 dikke film, 5%, 0,1 W, 150 V
 R13,R14 = 4k7, dikke film, 5%, 0,1 W, 150 V

Condensatoren:

C1 = 4700 μF/50 V, steek 10 mm, 22 x 41mm
 C2,C6,C7,C8 = 100 nF/50 V, X7R, 0805
 C3,C4 = 100 μF/16 V, 2312
 C5 = 10 μF/16 V, 1206

Spoel:

L1 = 470 μH power-inductor (SMD),
 afgeschermd, 270 mA, MSS7341T-serie

Halfgeleiders:

D1,D2,D3,D4 = S3J+, 600 V, 3 A
 D5 = MBR540, 40 V, 5 A,
 $V_f = 550 \text{ mV} @ I_f = 5 \text{ A}$
 LED1 = groen, 3 mm
 LED2 = rood, 3 mm
 T1,T2,T3,T4,T5,T6,T7 = NTF3055L108T1G
 MOSFET, N-channel, 3 A, 60 V, 120 mΩ
 IC1 = LM2576HVS-ADJ, step-down regulator,
 4–60 V, 3 A
 IC2 = DS3231MZ+ RTC, I²C, 5 ppm
 IC3 = MCP23008-E/SO, 8-bit I²C
 I/O-expander
 MOD1 = ESP-12F, ESP8266-based gebaseerde
 WiFi module

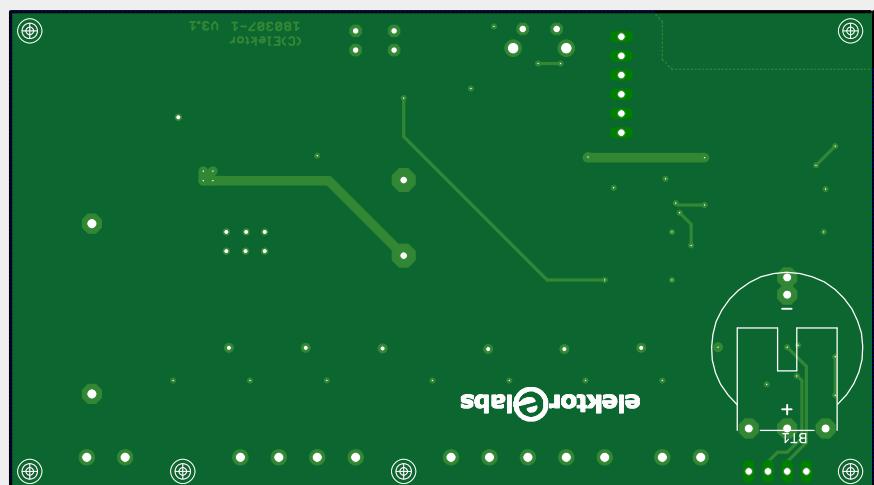
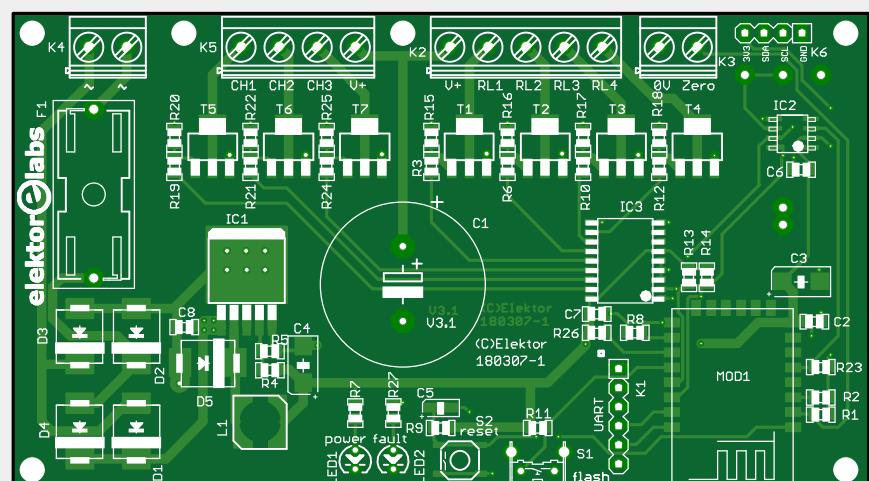
Diversen, en/of niet op de print:

Bt1 = CR2032-batterij, met houder, bijv.
 Keystone 1066
 D6,D7,D8,D9,D10,D11,D12 = 1N4007, 1000 V,
 1 A (niet op de print!)
 F1 = 2 AT zekering, 5 x 20 mm, met houder,
 printmontage, steek 22,6 mm
 F2 = 160 mAAT zekering, 5 x 20 mm
 (230-V-lichtnet) met houder,
 paneelmontage, soldeer-aansluitingen
 F2 = 320 mAAT zekering, 5 x 20 mm
 (115-V-lichtnet) met houder, paneelmontage,
 soldeer-aansluitingen
 K1 = pinheader 6-polig, steek 0,1", verticaal
 K2 = 5-polige printkroonsteen, steek 0,2",
 630 V
 K3,K4 = 2-polige printkroonsteen, steek 0,2",
 630 V
 K5 = 4-polige printkroonsteen, steek 0,2",
 630 V
 K6 = pinheader 4-polig, steek 0,1", verticaal
 LCD1 = 0,96" 128x64 OLED LC-display voor
 Arduino (optioneel)
 R28,R29,R30,R31 = 1k2, koolfilm, 5%, 0,25 W,
 250 V (niet op de print!)
 S1 = druktoets, bijv. SKHHLVA010

S2 = druktoets, 12 V, 50 mA, Multicomp
 TM-serie

Tr1 = ringkerentransformator, 2 x 12 V/30 VA,
 bijv. Multicomp MCTA030/12

Voedingsconnector, IEC320 C8 + kabel
 Print 180307-1 V3.1



Hoewel de meeste onderdelen op de print SMD-exemplaren zijn, zal het met een fijne soldeerbout en dunne (< 1 mm) soldeerterin best lukken. Soldeer van een component steeds eerst één aansluiting vast, controleer dan of het goed op zijn plaats én in de goede richting zit en soldeer dan pas de overige aansluitingen.

Vervolgens worden de onderdelen en modules aangesloten die niet op de print zitten; zie hiertoe het bedradingsschema van **figuur 2**.

Maak de cijferwielen schoon en vergeet niet de 1N4007-vrijloopdiodes (D6...D12) die over elke spoel én over de optionele belpoelen moeten worden gemonteerd. Verbind die kant van de spoelen waar de kathode van de diodes op zijn aangesloten met de V+-aansluiting op de print (K2 voor de cijferwielen en K5 voor de bellen). De andere zijde van het meest rechtse wielspoel (minuten) moet worden verbonden met RL1, de tweede (10 minuten) met RL2 op K2 etc., de bel met de hoogste noot op CH1 van K5, enzovoort.

De weerstanden voor de nuldetectie van de cijferwielen (R28...R31) worden direct op de aansluitingen van de nulpositie-schakelaars gesoldeerd. De schakelaars worden vervolgens in serie geschakeld en de uiteinden van de keten worden aangesloten op K3.

De secundaire wikkelingen van de voedingstrafo worden verbonden met K4. Vergoet niet de netzekering aan de primaire zijde van de trafo.

Software en firmware flashen

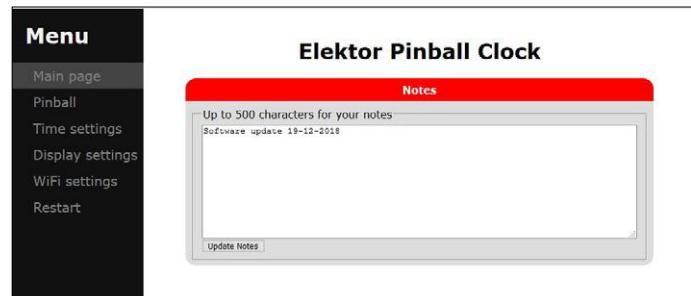
Er moet heel wat werk worden verzet voordat we kunnen beginnen met het compileren en uploaden van de code naar de ESP12-module van de klok.

Eerst moeten de Arduino IDE en de plugin voor de ESP8266-boards op de PC worden geïnstalleerd. Daarna hebt u de arduino-esp8266fs-plugin nodig om de web-content van de klok te uploaden [5]. Download de firmware voor dit project van [6]. Om deze code te compileren moeten de volgende libraries zijn geïnstalleerd:

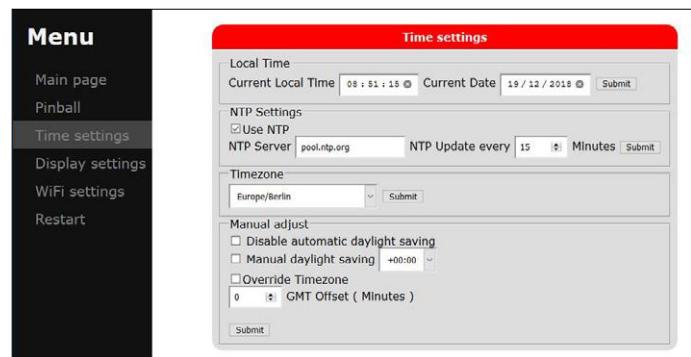
- RTCLib van Adafruit;
- MCP23008 van Adafruit;
- TimeLib van Michael Margolis;
- Arduino JSON 5.x;
- WebSockets van Markus Sattler;
- NtpClientLib form [7] (gebruik de AsyncUDP-branch);
- CRC32 van Christopher Baker.

Om u het leven een beetje gemakkelijker te maken, hebben we in het download-archief voor dit project [8] onze Arduino ‘portable’-map opgenomen, die de complete sketch bevat, evenals de libraries en de ESP8266FS-plugin. Om de omvang van de download te beperken, hebben we de ESP8266 boards-plugin er **niet** bij gedaan! Kopieer de gehele portable-map naar de Arduino-installatiemap op uw PC.

- Schakel de klok in en verbind een 3,3-V FTDI-kabel met K1 (de zwarte GND-draad wijst naar IC3) en een USB-poort op uw PC. Houd drukknop S1 (flash) ingedrukt, druk reset-knop S1 in en laat deze weer los om de ESP in bootloader-mode op te starten.
- Selecteer in de Arduino IDE de COM-poort waarmee de



Figuur 3: Via deze webpagina kunnen alle parameters en opties van de flipperklok worden ingesteld.



Figuur 4: Time settings-menu.

JACKPOT! Elektor-flipperklok op Hackaday: <https://bit.ly/2Rw9xln>

FTDI-kabel is verbonden en kies ‘nodeMCU 1.0 ESP-12E Module’ als targetboard.

- Open vanuit de Arduino IDE PinballClock.ino in deze map:
- ..\arduino\portable\sketchbook\PinballClock
- en compileer en upload de firmware vervolgens naar het board.

Tot slot moeten we de ESP-webpagina uploaden. In de Arduino-sketchmap staat een submap met de naam ‘data’ die alle bestanden voor deze webpagina bevat. Het upload-tool zal automatisch de hele inhoud van deze map naar het flashgeheugen van de ESP sturen. Controleer eerst of ‘Flash-size’ in het Tools-menu van de IDE op ‘4M (1M SPIFFS)’ staat. Herhaal de S1/S2-procedure om de ESP in bootloader-mode te laten starten en upload de webpagina van de ESP door in het Tools-menu te kiezen voor ‘ESP8266 Sketch Data Upload’. Dit duurt wel even, maar als de statusbalk in de IDE ‘SPIFFS Image Uploaded’ meldt, is de flipperklok klaar om te configureren.

Gebruik

Na het uploaden van de webpagina zal de klok zichzelf opnieuw resetten, de vier cijferwielen in de nulstand zetten en een willekeurige tijd uit de RTC (die nog niet is gesynchroniseerd) tonen. Houd ‘FAULT’ (LED2) in de gaten en als deze aangaat, con-

The screenshot shows the 'Display settings' menu. It includes sections for 'Reel Size' (set to 3" (100ms)), 'Reel status' (listing Reel 0 to 3 with 'No fault'), 'Time/Date format' (Date: Day-Month, Time: 24 hour), 'Silent mode' (checkbox checked for 'Silent mode Enable', with start and end times set to 21:00:00 and 09:00:00 respectively), and 'Display/Chime(s) ON-OFF' (checkboxes for 'Disable display' and 'Disable chime(s)'). A 'Submit' button is at the bottom of each section.

Figuur 5: Display settings-menu.

The screenshot shows the 'WiFi settings' menu. It displays 'Available networks' with 'SSID' set to 'Elektor-Wlan' and a 'Connect' button. Below it, 'Current connection' shows 'Elektor-User/Guest'. A 'Clear WiFi Settings' button is also present.

Figuur 6: WiFi settings-menu.

troleer dan de stand van de vier nulschakelaars op de cijferwielen, de weerstanden en hun bedrading en reset daarna opnieuw met S1.

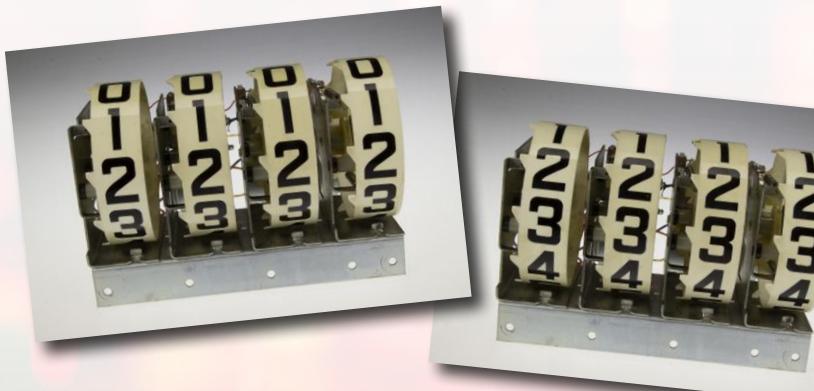
Als de klok niet met een WiFi-netwerk verbinding kan maken (zoals na het flashen van nieuwe firmware of als het netwerk down is), dan zet hij een open WiFi-accesspoint op met de SSID 'Elektor Pinball Clock'. Verbind uw PC, laptop of smartphone met dit netwerk, open uw browser en zoek naar IP-adres

192.168.4.1, dat de webpagina van de klok opent (**figuur 3**). Deze bevat een tekstkader dat als kladblok kan worden gebruikt, bijvoorbeeld voor uw hoogste score bij het flipperen of notities over eventuele eigen modificaties van de firmware.

Onder het menu-item 'Time settings' (**figuur 4**) kunt u handmatig tijd en datum instellen, maar u kunt ook synchroniseren met een NTP-server op het internet. In het laatste geval zoekt u eerst de SSID op van uw lokale netwerk in de lijst 'WiFi Settings', vult het wachtwoord in en klikt dan op 'submit'. De ESP reset en maakt verbinding met dit WiFi-netwerk en ontvangt een IP-adres van de DHCP-server van het netwerk.

Er zijn diverse manieren om erachter te komen wat dit adres is:

- open, met de FTDI no steeds aangesloten tussen PC en klok, de seriële monitor van de Arduino IDE om te zien of de login succesvol is en wat het IP-adres van de klok is;
- of verbind het optionele OLED-display LCD1 om na reboot de netwerk-settings te bekijken;
- of kijk in uw netwerk (in de Windows-netwerkmap, of met tools zoals Free IP Scanner) om te zien welk IP-adres is toegevoegd toen de klok werd ingeschakeld.



Nu kunt u via uw lokale WiFi-netwerk de klok benaderen met het IP-adres dat u net hebt ontdekt. Vergeet niet uw PC of mobiel apparaat weer met dit netwerk te verbinden, want het AP van de klok wordt uitgeschakeld zodra hij inlogt op uw WiFi. Open nu opnieuw 'Time Settings' en vul 'pool.ntp.org' in (of welke andere server u voor synchronisatie wilt gebruiken) voor de NTP-server, en het interval waarin de klok moet worden gesynchroniseerd. Selecteer je tijdzone en overige tijdstellingen en vergeet niet de bijbehorende 'submit'-knoppen om de klok-settings op te slaan.

Settings, bent u daar ook zo dol op? Hier zijn er een paar.

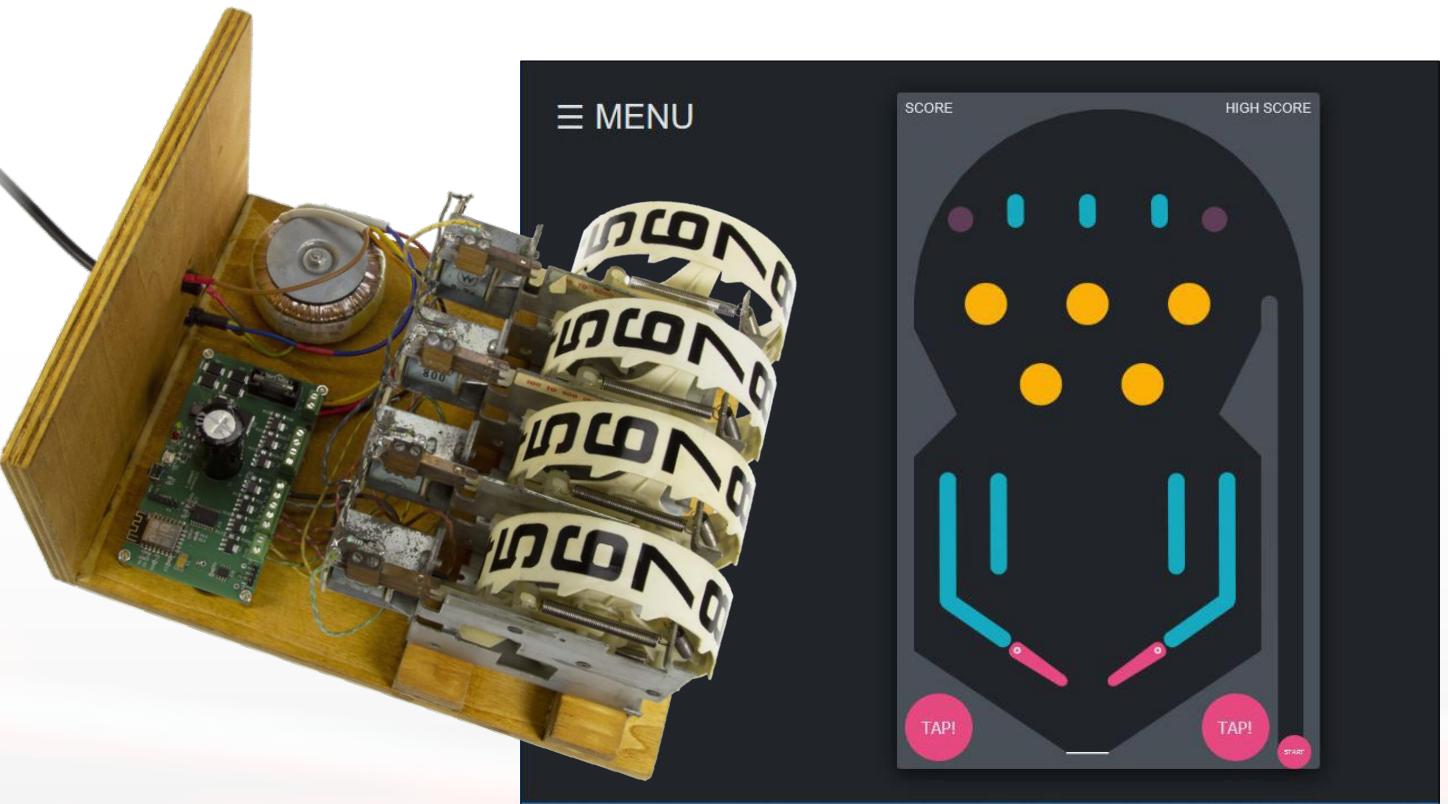
- Met 'Display settings' (**figuur 5**) kunt u timing en tempo van de pulsen naar de cijferwielen instellen. Het is niet per se nodig de standaardinstellingen aan te passen, want alle typen en merken zullen werken met de standaard 5"-parameters. Maar de kleinere 3"-wielen zijn ontworpen om sneller te 'stappen' en het oog en klinkt natuurlijker als ze sneller gaan. De grotere/oudere cijferwielen werken niet meer betrouwbaar met de snellere instellingen.
- Zes-inch-wielen hebben voor zover wij weten nooit bestaan, maar zijn toch aan het menu toegevoegd om meer opties te hebben; kies wat u het best bevalt.
- 'Reel status' laat zien of de nulschakelaars van de wielen worden geregistreerd; deze instelling kan worden

IN DE STORE

→ Kale print flipperklok
www.elektor.nl/pinball-clock-180307-1

→ ESP-12F, WiFi-module op basis van ESP8266,
www.elektor.nl/wi-fi-module-160100-92

→ (optioneel) 0.96" 128x64 OLED LC-display voor Arduino
www.elektor.nl/oled-display-for-arduino



Figuur 7: Het flipperspel ‘Easter Egg’ is opgenomen in de besturingsssoftware; de score wordt bijgehouden met de cijferwielen!

gebruikt om deze schakelaars af te regelen. De ‘Reset reels’-knop zet alle tellers op ‘0’ en wist alle ‘reel error’-meldingen (voor zover aanwezig natuurlijk).

- Met ‘Time/Date display’ kan het formaat van tijd (12 of 24u) en datum (mm-dd of dd-mm) worden gewijzigd.
- In ‘Silent mode’ kunnen wielen en bellen gedurende een bepaalde periode van de dag worden uitgeschakeld, meestal wanneer u van uw welverdiende nachtrust wilt genieten. Met de laatste optie in dit menu kunnen tellers en/of bellen ongeacht het tijdstip worden uitgeschakeld. Als het display wordt uitgeschakeld, worden de tellers op ‘9999’ gezet.
- ‘WiFi settings’ hebben we al gezien (**figuur 6**), maar nu de klok is verbonden met uw WiFi-netwerk toont dit de beschikbare netwerken en de naam van het netwerk dat de klok gebruikt in ‘Current network settings’. De klok onthoudt voorkeurnetwerk en wachtnaam zelfs als de voeding wordt uitgeschakeld.
- ‘Restart’ doet wat de naam suggerert: hiermee wordt de klok opnieuw opgestart. Merk op dat dit ook het IP-adres van de klok kan veranderen.

En dan nu het leukste, het bovenste item in de menulijst, dat we voor het

EXTRA BALLEN! Gratis flipperspel als onderdeel van de klok-appapp

toetje hebben bewaard: het flipperspel (**figuur 7**). Op het scherm verschijnt een simpel flipperspel als u dit item selecteert, de teller-wielen gaan naar ‘0000’ en houden de stand bij en natuurlijk zijn

de bellen actief! Het spel respecteert overigens de stille modus van de klok, dus u kunt het spel niet gebruiken om nietsvermoedende huisgenoten te wekken...

180307-03

Weblinks

- [1] Thermometer met flipperkast-cijferwielen (Elektor april 2012): www.elektormagazine.nl/magazine/elektor-201204/16541
- [2] Same PCB shoots again! (Elektor juli 2012): www.elektormagazine.nl/magazine/elektor-201207/16628
- [3] AC-spoelen op DC: www.ecmweb.com/content/using-ac-coils-dc-power
- [4] ESP8266 boards-plugin: <https://github.com/esp8266/Arduino>
- [5] Arduino esp8266fs-plugin: <https://github.com/esp8266/arduino-esp8266fs-plugin>
- [6] Pinball Clock softwarebundel voor dit project (Github): <https://github.com/ElektorLabs/180307-PinballClock>
- [7] NtpClientLib-form: <https://github.com/gmag11/NtpClient/issues/60>
- [8] Pinball Clock softwarebundel voor dit project (ElektorLabs magazine site): www.elektormagazine.nl/180307-03
- [9] Pinball Clock video (Elektor TV): <https://youtu.be/IQL2VkJHj-o>

Het SCCC-Project (1)

softcore-processor en C-compiler voor zelfbouw

Martin Oßmann (Duitsland)

Soms hebben we in een project zowel een FPGA als een klassieke CPU nodig, waarmee ook complexe besturingsprogramma's eenvoudig kunnen worden geprogrammeerd. Een manier om dat op te lossen is door gebruik te maken van een processor die in de FPGA is gerealiseerd. Zo'n 'softcore' kan kant-en-klaar worden gekocht, maar is vaak niet bepaald goedkoop. In dit project presenteren we een eenvoudige zelfbouw-CPU voor FPGA's. Dankzij de open broncode kan iedereen die aan zijn of haar eigen behoeften aanpassen.

Daar hoort ook een zelfbouw-C-compiler bij, die

eveneens kan worden aangepast en

uitgebreid. In deze serie

artikelen vullen we

de theorie zoals

gewoonlijk aan

met veel praktische

voorbeelden.



Wie zowel met FPGA's als met normale CPU's werkt, weet dat FPGA's in het voordeel zijn als een toepassing om veel paralleliteit vraagt terwijl het programmaverloop niet te ingewikkeld is. Als er echter een ingewikkeld besturingsprogramma nodig is, dan is een klassieke CPU beter op zijn plaats.

Het MAX1000-board

- Intel MAX®10 10M08SAU169C8G FPGA
- Arrow USB Programmer2
- 64 MBit SDRAM (16-bits databus)
- 64 MBit flashgeheugen
- 12-MHz MEMS-Oscillator
- Optionele MEMS-oscillator
- 8 rode vrij te gebruiken LED's
- 2 indicatie-LED's
- 2 vrij te gebruiken drukknoppen
- 3-assige-versnellingsopnemer
- 12-pens PMOD-connector (zelf solderen)
- Arduino MKR-connector (zelf solderen)
- JTAG-connector (zelf solderen)
- 3-pens I/O-connector (zelf solderen)

Daarom vinden we in veel projecten zowel een FPGA als een microcontroller. Een alternatief is om een CPU in de FPGA te realiseren. Zulke 'softcores' worden vaak door FPGA-fabrikanten aangeboden. Meestal betreft dat dan merkgebonden software, waaraan vaak een stug prijskaartje hangt.

Dit is waar het *Small C Compiler CPU Project (SCCCP)* van pas komt: hiermee kan een eenvoudige CPU worden gerealiseerd die we, omdat de broncode beschikbaar is, ook aan onze eigen eisen kunnen aanpassen. Daarnaast hebben we een taal nodig om die CPU te programmeren. Hiervoor is een compiler in het leven geroepen, die een subset van de taal C ('smallC') kan vertalen. Deze taal is op integer-variabelen gebaseerd. Ook die compiler kunnen we desgewenst zelf uitbreiden.

Hardware

Als hardware gebruiken we in dit project de MAX1000-kaart van Arrow (deze kost minder dan € 30 [1]). Hij is uitgerust met een MAX10-FPGA van Intel/Altera, een USB-programmeerinterface, twee drukknoppen en acht vrij te gebruiken LED's en verschillende uitbreidingsconnectoren (zie tekstkader). Om vertrouwd te raken met de kaart en de Quartus-software (van Intel/Altera), kunnen we al een eerste experiment uitvoeren. Onder [2] vinden we de *MAX1000 User Guide*. Daarin

wordt beschreven hoe we de Quartus-ontwikkelflitsware kunnen installeren. Met die software kunnen we programmeerfiles maken, die we dan moeten uploaden naar de FPGA het inwendige daarvan te configureren. Dat proces wordt ook wel synthese genoemd. Als hardwarebeschrijvingstaal gebruiken we bij dit project Verilog. In het januari/februari-nummer hebben we de beginselen van Verilog uit de doeken gedaan [3].

Experiment 1: test

Met experiment 1 kunnen we testen of Quartus goed is geïnstalleerd. Download eerst het software-archief van de webpagina van dit project [4]. Dit bevat de verschillende demo's voor het SCCC-project. Pak het archief uit en zet het in een map voor dit project op uw eigen PC: *C:/SCCCP/*.

Open nu met Quartus simpelweg het QPF-bestand (Quartus-ProjectFile) voor experiment 1: (*C:/SCCCP/experiment1/experiment1/experiment1.qpf*). Synthetiseer dan het project en laad de gegenereerde programmeerfile in de MAX1000-kaart. Als dat allemaal is gelukt, zou er nu een LED-looplicht (met de gebruiker-LED's) te zien moeten zijn.

Hiermee hebben we voor het eerst de zelfbouw-processor gesynthetiseerd en op de FPGA laten werken. Het C-programma stond al kant-en-klaar gecompileerd in de directory van experiment 1. De broncode staat in **listing 1**.

Wanneer u nieuwsgierig bent, kunt u een kijkje nemen in de directory van experiment 1. Daar staan de bestanden *codemem.txt* en *datamem.txt*. Deze zijn aangemaakt door de C-compiler van de auteur. Bij het uploaden van de FPGA-programmeerfile wordt ook de inhoud van deze bestanden in het geheugen van de FPGA gezet. De inhoud van *codemem.txt* legt de initiële inhoud van het programmageheugen van de CPU vast (met de instructies in machinetaal), *datamem.txt* bepaalt de initiële inhoud van het datageheugen voor deze CPU (zie onder). Als we in die bestanden kijken, kunnen we zien dat de broncode er ook in staat. Zo kunnen we later altijd nog de broncode zien. In de werkdirctory bevindt zich trouwens ook nog het bestand *sCCCPcpu1v01.v*. Dat is onze CPU in de vorm van een Verilog-bestand; hij bestaat uit ongeveer 1000 regels code. De workflow met de verschillende tools en bestanden is te zien in **figuur 1**. Op de computer van de auteur duurt een syntheseproces ongeveer 4 minuten. Dat maakt het zoeken naar fouten nogal frustrerend en tijdverdorpend. Er moet een betere manier zijn, zoals we straks nog zullen zien.

Geschiedenis

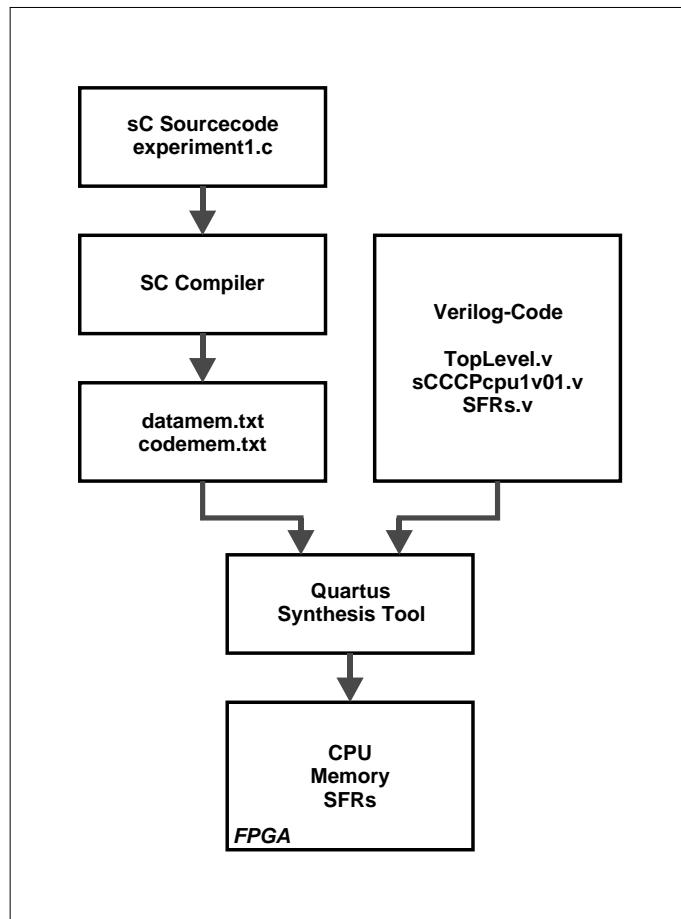
Een CPU (en een bijpassende compiler) ontwerpen is natuurlijk niet iets dat je op een regenachtige zondagmiddag evenjes doet. Het is dan een goed idee om uit te gaan van een verstandig ontwerp. De inspiratie voor SCCC kommt van het 'small c'-project van Ron Cain (Ron Cain, 'A Small C Compiler for the 8080's', Dr. Dobb's Journal, april-mei 1980, pp. 5...19). Hij heeft een eenvoudige maar toch krachtige, subset van C ontwikkeld; de compiler genereert code voor de 8080 of Z80. Daarbij worden relatief weinig verschillende 8080-commando's gegenereerd. Een dergelijke 'smalspoor'-CPU zou relatief gemakkelijk als softcore uitgevoerd moeten kunnen worden. Ons SCCC-CPU-ontwerp is zó ontwikkeld dat het de machine-instructies kan verwerken die de compiler genereert bij het vertalen van smallC-code. Wie zich de tijd van de 8080-processor nog kan herinneren, zal al snel de verwantschap met onze CPU zien.

Listing 1. Looplicht

```
setLeds(int pattern){ // output pattern to LED
    pattern ;
    #asm
        OUTA 5           // LEDs are attached to channel 5
    #endasm
}

delay(int time){
    int k ;
    for(k=0 ; k<time ; k++){
    }
}

main(){
    int k ;
    while(1){
        for(k=0 ; k<8 ; k++){
            setLeds(1 << k) ;
            delay(500000) ;
        }
    }
}
```



Figuur 1: De C-compiler zet het C-programma om in machinetaal en maakt het bestand 'codemem.txt' aan. De binair code wordt in het RAM van de FPGA geladen en wordt daar door de gesynthetiseerde CPU uitgevoerd.

Listing 2. Looplicht met interrupt-afhandeling

```
#asm
RTCintVector      EQU  8
CPUflagsChannel   EQU  6
LEDsChannel       EQU  5
ORG    RTCintVector
PUSH   R0
PUSH   R1
CALL   RTCinterrupt
POP    R1
POP    R0
RET
#endasm

setCPUflags(int x) {
    x ;
#asm
    OUTA  CPUflagsChannel
#endasm
}

setLeds(int pattern){
    pattern ;
#asm
    OUTA  LEDsChannel
#endasm
}

RTCinterrupt(){
    setLeds(0xFF) ;
}

main(){
    int k ;
    setCPUflags(1) ;
    while(1){
        for(k=0 ; k<8 ; k++){
            setLeds(1 << k) ;
        }
    }
}
```

CPU-architectuur

Onze SCCC-P CPU realiseert een zogenaamde gemodificeerde Harvard-architectuur, d.w.z. dat het programma- en datageheugen van elkaar gescheiden zijn. In de standaardconfiguratie hebben we plaats voor 4096 commando- en 1024 datawoorden. Verder beschikt de CPU over een programmateller (PC) en een stackpointer (SP). De beide registers zijn zo breed, dat we alle woorden rechtstreeks kunnen adresseren, de programcounter is dus 12 bit breed en de stackpointer 10.

Daarnaast zijn er nog twee registers: R0 is het primaire register; de meeste data gaat via dit register. Register R1 is het secundaire register. Dat is nodig als we met twee operanden willen werken (in Ron Cain's smallIC-implementatie komt R0 overeen met het HL-register en R1 met het DE-register). Deze registers zijn elk 32 bit breed.

Het datageheugen is ook 32 bit breed, het programmageheugen heeft een breedte van 40 bit. In een commandowoord

passen dus een 32-bit woord voor de data en 8 bit voor de opcode-decodering.

Laten we hier als voorbeeld eens een commando voor onze CPU bekijken. Het commando:

```
LDI      1234      ; LoaDImmediate
```

schrijft de waarde 1234 in het register R0.

Onze machine kent alleen 32-bit woorden en alle bewerkingen worden uitgevoerd op 32 bit. Als we met char-waarden willen rekenen, gebruiken we alleen de laagste 8 bit. Dat lijkt op het eerste gezicht verkwisting, maar het maakt het wel veel gemakkelijker om een compiler te ontwikkelen.

Periferie

Verder kan de CPU ook werken met andere waarden uit onze hardware, die eveneens 32 bit breed zijn.

Als we de CPU bijvoorbeeld uitbreiden met een Real Time Counter (RTC), dan willen we de waarde van de RTC kunnen lezen en schrijven, en we willen de periodeduur kunnen aangeven. Er zijn in dit geval dus twee waarden (`RTCTimer`, `RTCPériode`) die van de CPU komen, en één waarde die de CPU kan uitlezen, namelijk de huidige tellerstand in `RTCTimer`. Alle parameters in de periferie van de CPU die worden gelezen-of geschreven, krijgen een nummer toegewezen (SFR-adres = adres van een Special Function Register). Om waarden te lezen gebruiken we het commando:

```
INPA nn
```

Dit commando leest het functieregister nn uit en schrijft de waarde naar R0.

Het commando:

```
OUT A nn
```

schrijft de inhoud van R0 naar functieregister nn.

Bij onze experimenten gaan we de kale CPU uitbreiden met dit soort bijkomende functionaliteit (seriële interface, A/D-convertisers, PWM, RTC, interrupts). De CPU heeft dan toegang tot de betreffende SFR's met commando's INPA en OUTA.

Als we bijvoorbeeld de RTC op een bepaalde waarde willen instellen of de huidige waarde ervan willen uitlezen, dan gebruiken we het SFR-adres 4. De periodeduur `RTCTop` is te vinden op SFR-adres 7. Om dus bijvoorbeeld de RTC in te stellen op 256 en de periodeduur op 500, kunnen we in assemblertaal schrijven:

```
LDI      256 // put constant in
          // Register R0
OUTA    4   // output via SFRaddress 4,
          // i.e. RTCTimer
LDI      500 // put constant in
          // Register R0
OUTA    7   // output via SFRaddress 4,
          // i.e. RTCTimerTop
```

Via de SFR-adressen hebben we ook toegang tot de I/O-pennen van de FPGA, die op hun beurt weer met externe periferie zoals bijvoorbeeld de LED's op de kaart zijn verbonden.

Debuggen van het CPU-ontwerp

Het debuggen van een CPU in Verilog gaat erg lang duren als voor elke iteratie een volledige synthese van het FPGA-ontwerp moet worden uitgevoerd. Gelukkig kunnen we in plaats daarvan gebruik maken van een Verilog-simulatie wanneer we in de Verilog-code van de CPU (en de uitbreidingen daarvan) naar fouten willen zoeken. We doen dat met de gratis simulator *Icarus Verilog* [5]. Een installatiehandleiding is te vinden onder [6].

Voordat u experiment 2 kunt uitvoeren, moet u eerst de twee batch-bestanden *c.bat* en *r.bat* in de directory van experiment 2 zó veranderen dat daarmee de Icarus-Verilog-simulator correct wordt aangeroepen. Bij de auteur staan de exe-bestanden van Icarus in de directory *C:/iverilog/bin/*. Daarom staat in het bestand *c.bat* (c voor ‘compile’) het volgende:

```
c:/iverilog/bin/iverilog.exe -o Vobject TestBench1v01.v
```

Het batch-bestand *r.bat* (r voor ‘run’) ziet er als volgt uit:

```
c:/iverilog/bin/vvp.exe Vobject
```

U moet dus in de bestanden *c.bat* en *r.bat* het juiste pad naar uw eigen Icarus-installatie invoegen.

De SCCC-omgeving

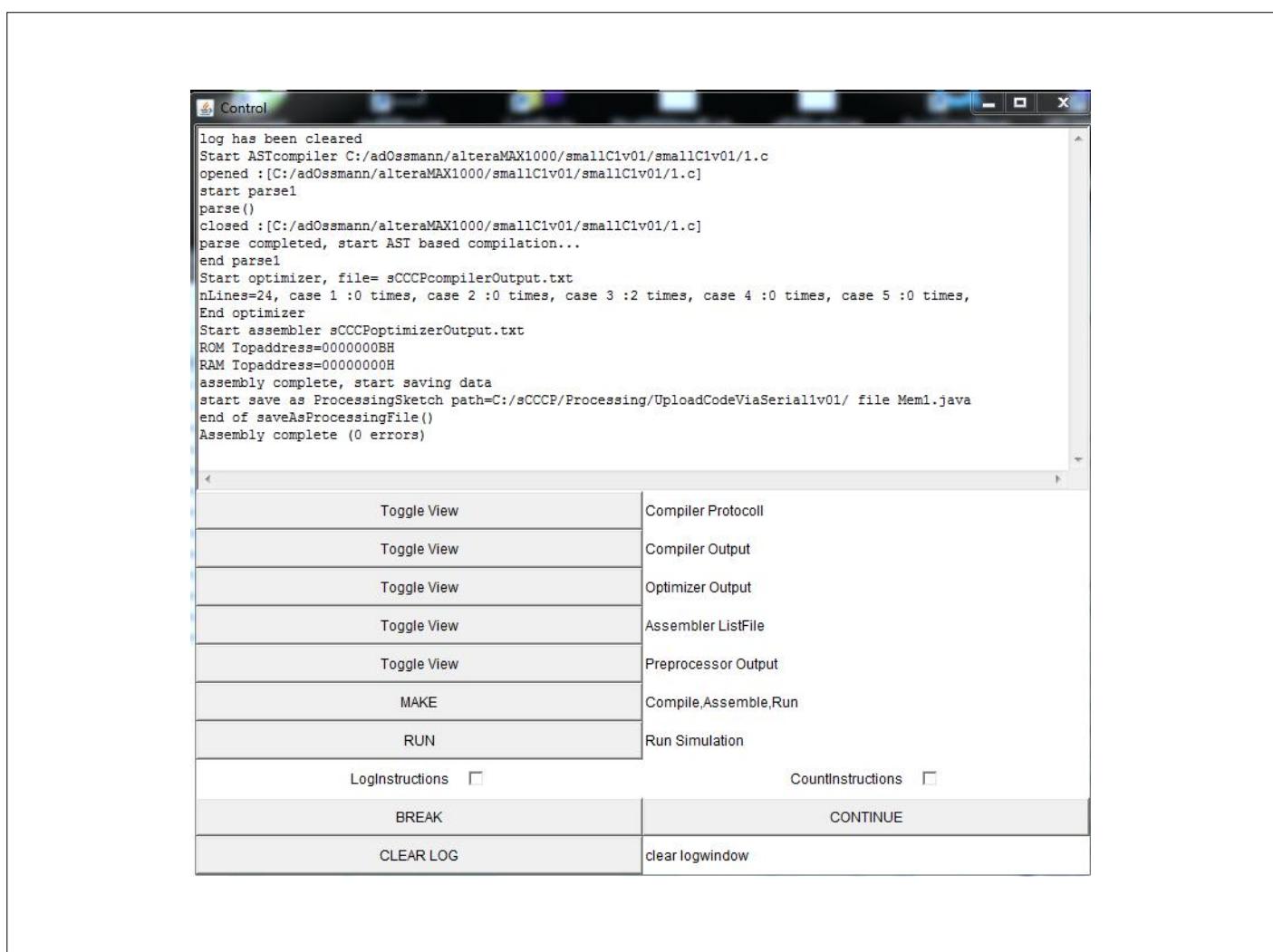
In de map van experiment 1 stond het toepassingsprogramma voor onze CPU al kant-en-klaar gecompileerd ter beschikking in de vorm van *codemem.txt*, een image van het programmagheugen. Maar normaal gesproken willen gebruikers natuurlijk hun eigen C-programma’s schrijven, compileren en op de CPU uitvoeren. Hiervoor heeft de auteur een kleine ontwikkelomgeving in de programmeertaal Java gemaakt. Die kan vanaf de commandline worden aangeroepen met

```
java -jar c:/SCCCP/CompilerAsJar/sCCCP.jar -s  
experiment2.c -i c:/SCCCP/experiment2/experiment2/
```

Bij de aanroep met de hier getoonde parameters wordt de C-broncode *experiment2.c* in de ontwikkelomgeving geladen.

Om de lezer het foutgevoelige getype te besparen, is de aanroep in de vorm van een batch-bestand gerealiseerd. Dit bevindt zich in de map van experiment 2 en draagt de naam *c_compile.bat*.

De GUI van onze kleine ontwikkelomgeving is te zien in **figuur 2**. Met de knop *MAKE* kunnen we het C-programma compileren. Daarbij zijn diverse opties voor het debuggen en simuleren beschikbaar; we komen daar in de loop van dit artikel nog op terug.



Figuur 2: De kleine ontwikkelomgeving met compiler en simulator is geschreven in Java.

Listing 3. Debug-info

```
VCD info: dumpfile test1.vcd opened for output.  
SC reset  
SC reset  
-- set CPUflags 00000001H  
-- out channel 00000005H data = 00000001H  
-- out channel 00000005H data = 00000002H  
-- out channel 00000005H data = 00000004H  
-- out channel 00000005H data = 00000008H  
-- out channel 00000005H data = 00000010H  
//  
Interrupt irq0 CPUflags=00000001H  
***** interrupt ACK selectedIRQ=0001H  
    serviceCode at 0008H simTimer=0000012eH save  
    PC=00000028H  
  
-- out channel 00000005H data = 000000ffH  
-- out channel 00000005H data = 00000020H  
-- out channel 00000005H data = 00000040H  
-- out channel 00000005H data = 00000080H  
-- out channel 00000005H data = 00000001H  
-- out channel 00000005H data = 00000002H  
-- out channel 00000005H data = 00000004H
```

Experiment 2: simulatie met Icarus

Bij experiment 2 gebruiken we opnieuw de looplicht-code van experiment 1, maar nu uitgebreid met een interrupt-handler voor interrupt 1 (**listing 2**). Roep eerst met *c_compile.bat* de SCCC omgeving op en bouw met MAKE de vertaalde code (in de werkdirectory C:/sccc/experiment/experiment2).

Vervolgens vertalen we met het aanroepen van *c.bat* de code in het interne Icarus-formaat. Met *r.bat* kunnen we dan de simulatie starten. Het resultaat is de uitvoer van **listing 3**. Hierin is te zien dat we de waarden 01H, 02H, 04H enz. krijgen, die verantwoordelijk zijn voor het LED-patroon. De LED's kunnen worden aangesproken op SFR-adres 5. Verder zien we hoe tussentijds een interrupt wordt afgehandeld. Maar veel meer informatie dan in experiment 1 krijgen we niet. Dat komt doordat de *defines* voor het debuggen nog uitgecommentarieerd zijn in de Verilog-code voor de CPU. We kunnen de *defines* in de eerste regels van *sCCCPcpu1v01.v* activeren door de // -tekens te verwijderen.

```
/*'define VERBOSE  
/*'define VERBOSE1  
/*'define VERBOSE3  
/*'define VERBOSE_INPA1  
/*'define VERBOSE_OUTAchrOut
```

Na opnieuw compileren krijgen we nu in detail te zien, wat er in de CPU gebeurt. De uitvoer voor de eerste vier commando's zien we in **listing 4**.

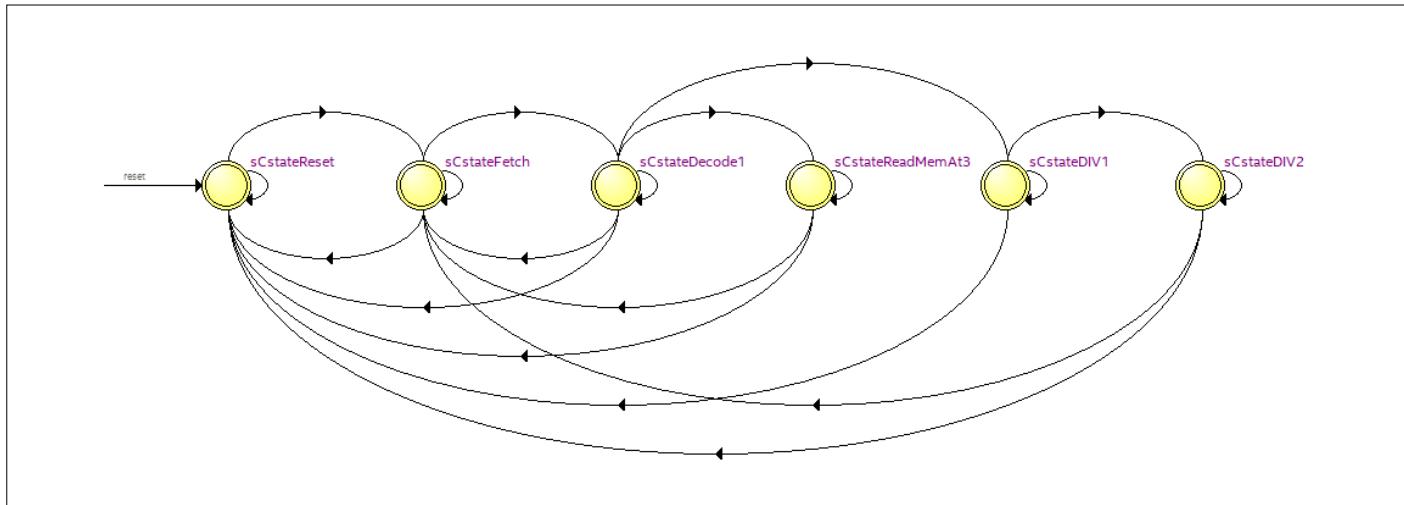
Zoals u ziet, wordt voor elk commando de inhoud van alle

Listing 4. Gedetailleerde info

```
FETCH PC=0000H Time=000002 SP=03ffH OPC=2500000018H R0=00000000H R1=00000000H  
decode: CALL 00000018H  
pending write, store adr=000003feH val=00000001H  
  
FETCH PC=0018H Time=000004 SP=03feH OPC=2afffffffH R0=00000000H R1=00000000H  
decode: ADDSP ffffffffH  
  
FETCH PC=0019H Time=000006 SP=03fdH OPC=4000000001H R0=00000000H R1=00000000H  
decode: PUSHI 00000001H  
pending write, store adr=000003fcH val=00000001H  
  
FETCH PC=001aH Time=000008 SP=03fcH OPC=25000000eH R0=00000001H R1=00000000H  
decode: CALL 0000000eH  
pending write, store adr=000003fbH val=0000001bH
```

Listing 5. De commando's NOP en LDI in Verilog

```
if( sCstate==sCstateDecode1 ) begin  
    if (sCopcode==sCopcNOP) begin  
        sCstate<=sCstateFetch ;           // NOP instruction: nothing to do  
        end                         // goto fetch phase of next instruction  
    else if (sCopcode==sCopcLDI) begin  
        sCregR0 <= sCoperand1 ;       // Data from instruction goes to R0  
        sCstate<=sCstateFetch ;       // goto fetch of next instruction  
        end  
    else  
    ...
```



Figuur 3: Toestandsdiagram van de *state machine* voor de verwerking van de machine-instructies.

registers weergegeven, plus de mnemonic van het betreffende commando. We krijgen dus heel gedetailleerde informatie. Maar dit is wel een ware stortvloed aan informatie. We moeten dus creatief te werk gaan om alleen die informatie te laten weergeven, die we nodig hebben. Soms is zoveel informatie wel nodig om een fout op te sporen. En dan kan het vaak een probleem zijn, dat de simulatie niet alle aspecten van het verloop van het programma kan laten zien, omdat het al gauw te omslachtig is om alle aspecten van het werkelijke ontwerp af te beelden. Stel u zich eens voor dat we in een echt ontwerp draadloze signalen willen analyseren. Dan zouden we die draadloze signalen met de juiste timing moeten simuleren, wat vaak heel omslachtig is.

Maar als we bijvoorbeeld fouten in de CPU-simulatie willen opsporen, dan is de Icarus-Verilog-simulator werkelijk waardevol. Op die manier hebben we diverse fouten in de CPU opgespoord en vervolgens verholpen.

Programmeren van de CPU in Verilog

We gaan nu aan de hand van enkele specifieke commando's verklaren hoe de SCCCP-CPU in Verilog wordt gerealiseerd. De commando's worden verwerkt in een toestandsmachine (*finite state machine*); in **figuur 3** zien we het door Quartus gegenereerde toestandsdiagram. De eerste fase heet `sCstateFetch`; daarin wordt alleen het commando geladen en de programcounter (`sCregPC`) met 1 verhoogd. De volgende toestand is `sCstateDecode1`. In Verilog ziet die er zo uit:

```

if( sCstate==sCstateFetch ) begin
    sCinstruction <= sCcodeMem[sCregPC] ;
    sCregPC <= sCregPC + 1'b1 ;
    sCstate <= sCstateDecode1 ;
end

```

In de toestand `sCstateDecode` wordt het commando gedecodeerd; daarna wordt begonnen met de eigenlijke uitvoering. Een aantal commando's wordt ook al volledig verwerkt. De Verilog-code voor de commando's NOP en LDI is te zien in **listing 5** (als volgende toestand wordt weer `sCstateFetch` gezet, waarmee het volgende commando begint).

Er zijn natuurlijk ook complexere instructies. De instructie `LDPUSH aa` zet de waarde, die op adres aa staat, op de stack en in R0. In de toestand `sCstateDecode1` wordt de waarde van aa overgebracht naar het tussengeheugen `readAddress`. Daarna volgt de toestand `sCstateReadMemAt3`. Daarin wordt de waarde uit het datageheugen gelezen en in register R0 gezet. Tegelijk wordt de schrijfactie naar de stack voorbereid door de waarde van de stackpointer `sCregSP-1` naar de variabele `writeAddress` te schrijven; deze waarde wordt ook ingesteld als nieuwe waarde van `sCregSP`. Als te schrijven waarde `writeValue` wordt de eerder uitgelezen waarde gebruikt. Het schrijven zouden we nu in een vierde toestand kunnen doen. Maar omdat onze CPU een gemodificeerde hardware-architectuur heeft, kunnen we tegelijkertijd het datageheugen en het programmageheugen aansturen. Daarom wordt de nog

Weblinks

- [1] MAX1000 Board: <https://shop.trenz-electronic.de/de/Produkte/Trenz-Electronic/MAX1000-Intel-MAX10/>
- [2] MAX1000-Board Manual met installatiehandleiding voor Quartus: www.trenz-electronic.de/fileadmin/docs/Trenz_Electronic/Modules_and_Module_Carriers/2.5x6.15/TEI0001/User_Guide/MAX1000%20User%20Guide.pdf
- [3] 'Verilog Basics', Elektorlabs 1-2/2019: www.elektrormagazine.nl/180562-03
- [4] Projectpagina met software-download: www.elektrormagazine.nl/180394-03
- [5] Simulator Icarus Verilog: <http://iverilog.icarus.com>
- [6] Installatiehandleiding voor Icarus: http://iverilog.wikia.com/wiki/Installation_Guide

Overzicht van de CPU-commando's

instruction set (all operations 32 bit wide):

Name	cycles	meaning	explanation
NOP	2	no Operation	
IDIV	37	signed integer division	R0=R1 / R0
IMOD	37	modulus operation	R0=R1 % R0 for positive operands
HALT	2	halts execution	
LDIND	3	load indirect	R0=memory[R0]
LDI n	2	load immediate	R0=n
ADDI n	2	add immediate	R0=R0+n
ADDSP n	2	add to stack pointer	SP=SP+n
JMP adr	2	jump	PC=adr
PCR0	2	load PC with R0	PC=R0
JFALSE adr	2	jump if HL==0	if(HL==0) PC=adr
LD @adr	3	load	R0=memory[adr]
PUSH R0	2+	save R0 on stack	memory[--SP]=R0
PUSH R1	2+	save R1 on stack	memory[--SP]=R1
CALL adr	2+	call subroutine	memory[SP--]=PC , PC=adr
POP R1	3	fetch R1 from stack	R1=memory[SP++]
POP R0	3	fetch R0 from stack	R0=memory[SP++]
XTR0	3	xchange top of stack and R0	R0=memory[SP], memory[SP]=R0
RET	3	return from subroutine or interrupt	PC=memory[SP++]
IADD	2	integer addition	R0=R1 + R0
IADD SP	2	add stackpointer to R0	R0=R0 + SP
IXOR	2	bitwise XOR	R0=R1 ^ R0
IOR	2	bitwise OR	R0=R1 R0
IAND	2	bitwise AND	R0=R1 & R0
ICOM	2	bitwise complement	R0= ~ R0
INEG	2	signed negate	R0= - R0
IMUL	2	signed multiply	R0=R1 * R0
IASR	2	arithmetic shift right	R0=R1 >> R0
IASL	2	arithmetic shift left	R0=R1 << R0
ST @adr	2+	store	memory[adr]=R0
STIND @R1	2+	store indirect	memory[R1]=R0
OUTA ch	2	output R0 to channel ch	channel n=R0
INPA ch	2	input R0 from channel ch	R0=channel n
CMPEQ	2	compare equal	R0=(R1==R0)
CMPNE	2	compare not equal	R0=(R1!=R0)
CMPLT	2	compare less than	R0=(R1<R0)
CMPLE	2	compare less equal	R0=(R1<=R0)
CMPGT	2	compare greater than	R0=(R1>R0)
CMPGE	2	compare greater equal	R0=(R1>=R0)
CMPULT	2	compare unsigned less than	R0=(R1<R0)
CMPULE	2	compare unsigned less equal	R0=(R1<=R0)
CMPUGT	2	compare unsigned greater than	R0=(R1>R0)
CMPUGE	2	compare unsigned greater equal	R0=(R1>=R0)

The following instructions were added later to increase efficiency (higher speed, smaller code).

These instructions are not necessary. They are used by a simple peephole-optimizer.

Switching the optimizer off eliminates usage of these instructions.

PUSHI n	2+	push immediate value	R0=n ; SP-- ; memory[SP]=R0
TWOOP n	2	prepare two operands	R1=R0 ; R0=n ;
LDPUSH adr	3+	load var and push	R0=memory[adr] ; SP-- ; memory[SP]=R0
ADDSPI n	2	load R0 by SP+offset	R0=SP+n
GETLCL n	3	get local variable	R0=SP+n ; R0=memory[R0]

Listing 6. Het commando LDPUSH in Verilog

```

if( sCstate==sCstateDecode1 )
    readAddress <= sCoperand1 ;
    sCstate<=sCstateReadMemAt3 ;

if( sCstate== sCstateReadMemAt3 )
    sCregR0 <=sCdataMem[readAddress] ;
    writeAddress <= sCregSP-1'b1 ;
    writeValue <= sCdataMem[readAddress] ;
    writePending <= 1 ;
    sCregSP <= sCregSP-1'b1 ;
    sCstate<=sCstateFetch ;

if( sCstate== sCstateFetch )
    if(writePending) begin
        sCdataMem[writeAddress[sCdataMemAddressWidth-1:0]]<=writeValue ;
    end

```

Listing 7. Het commando LDPUSH in de Java-simulator

```

instruction=ram[regPC] ;
opcode=(instruction>>32) & 0xFF ;      // 8 Bit opcode
operand1=instruction & 0xFFFFFFFF ;   // 32 Bit immediate operand1

switch(opcode){
    case(ccNOP) : { return ;}
    case(ccLDI) : { regR0=operand1 ; return ;}
    case(..ccLDPUSH) : { regR0=readDataMemX(operand1) ;
        regSP-- ;
        writeDataMemX(regSP,regR0) ;
        return ;}
}

```

uitstaande schrijfoperatie naar het datageheugen tegelijk met de volgende fetch-fase uitgevoerd. De toestandsmachine herkent dat aan het gezette bit `writePending`. Zodoende wordt de uitvoeringstijd 4 naar 3 toestanden bekort.

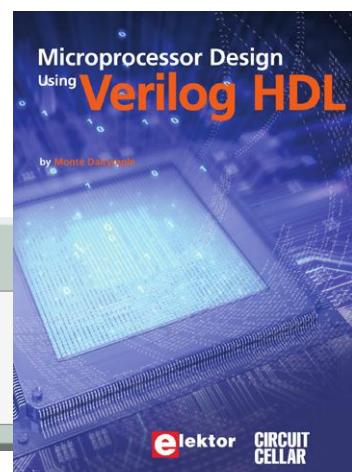
Het uitvoeren van het commando LDPUSH gebeurt zoals te zien is in **listing 6**.

Dit soort code moet voor elke instructie (zie het **tekstkader Overzicht van de commando's**) worden geschreven. Dat is niet echt moeilijk. Maar voordat alles feilloos werkte, waren toch wel een paar dagen programmeren nodig.

De auteur heeft in de SCCCP-ontwikkelomgeving een kleine PC-simulator voor de CPU ingebouwd. In Java ziet de implementatie van het commando LDPUSH er veel eenvoudiger uit (**listing 7**).

Met de simulatie van de CPU gaan we ons bezig houden in experiment 3, dat u in het tweede deel van deze artikelreeks kunt vinden. ▶

(180394-03)



IN DE STORE



→ E-Book (Engelstalig): 'Microprocessor Design Using Verilog HDL'
www.elektor.nl/microprocessor-design-using-verilog-hdl-e-book

→ MAX1000 FPGA-ontwikkelboard
www.elektor.nl/max1000

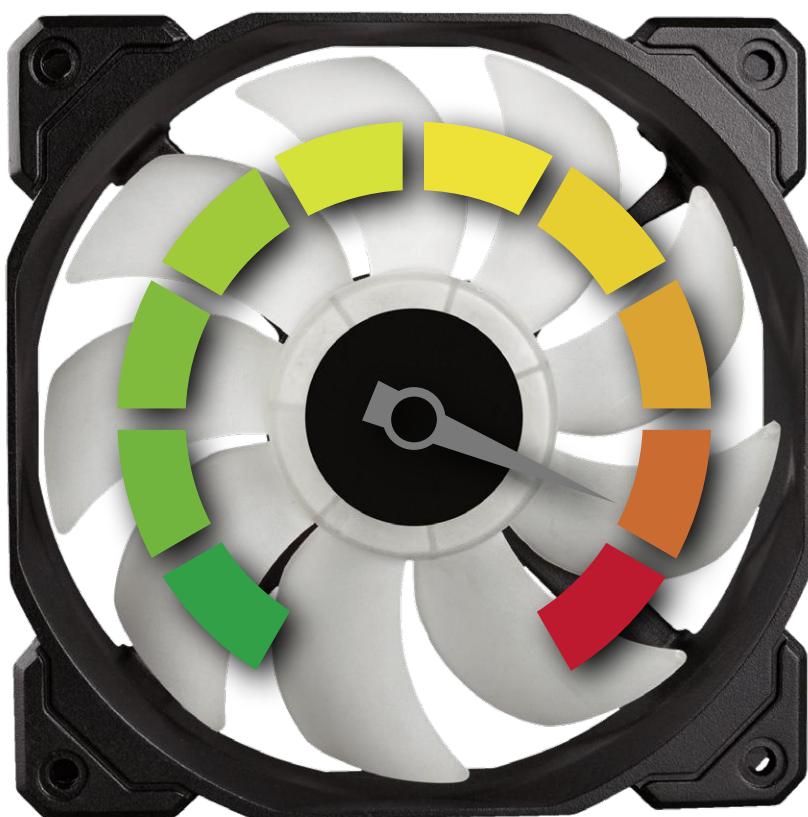
elektor CIRCUIT CELLAR

Ventilatorsturing

Naar keus met Arduino of analoge elektronica

Jac Hettema (Nederland)

In schakelingen waar veel energie gedissipeerd wordt, is in een aantal gevallen een actieve koeling gevraagd. Dat wil zeggen: met een ventilator. En om de slijtage en met name het geluid te beperken, is het wenselijk de draaisnelheid van de ventilator aan te passen aan het te dissiperen vermogen. Daarvoor is een sturing nodig. In dit artikel bespreken we twee versies: 'ouderwets' analoog en 'nieuwerwets' met een Arduino en software.



Elektronische schakelingen hebben vaak te kampen met warmteontwikkeling. Wanneer de hoeveelheid af te voeren warmte gering is, kunnen we vaak volstaan met een passieve koeling, oftewel een eenvoudig koellichaam. Maar wanneer het vermogen groter wordt, dient het koellichaam eveneens groter te worden. Dat levert soms onoverkomelijke problemen op en de oplossing kan dan liggen in actieve koeling — met een ventilator. Hoe je een dergelijke ventilator aan kan sturen, tonen we hier.

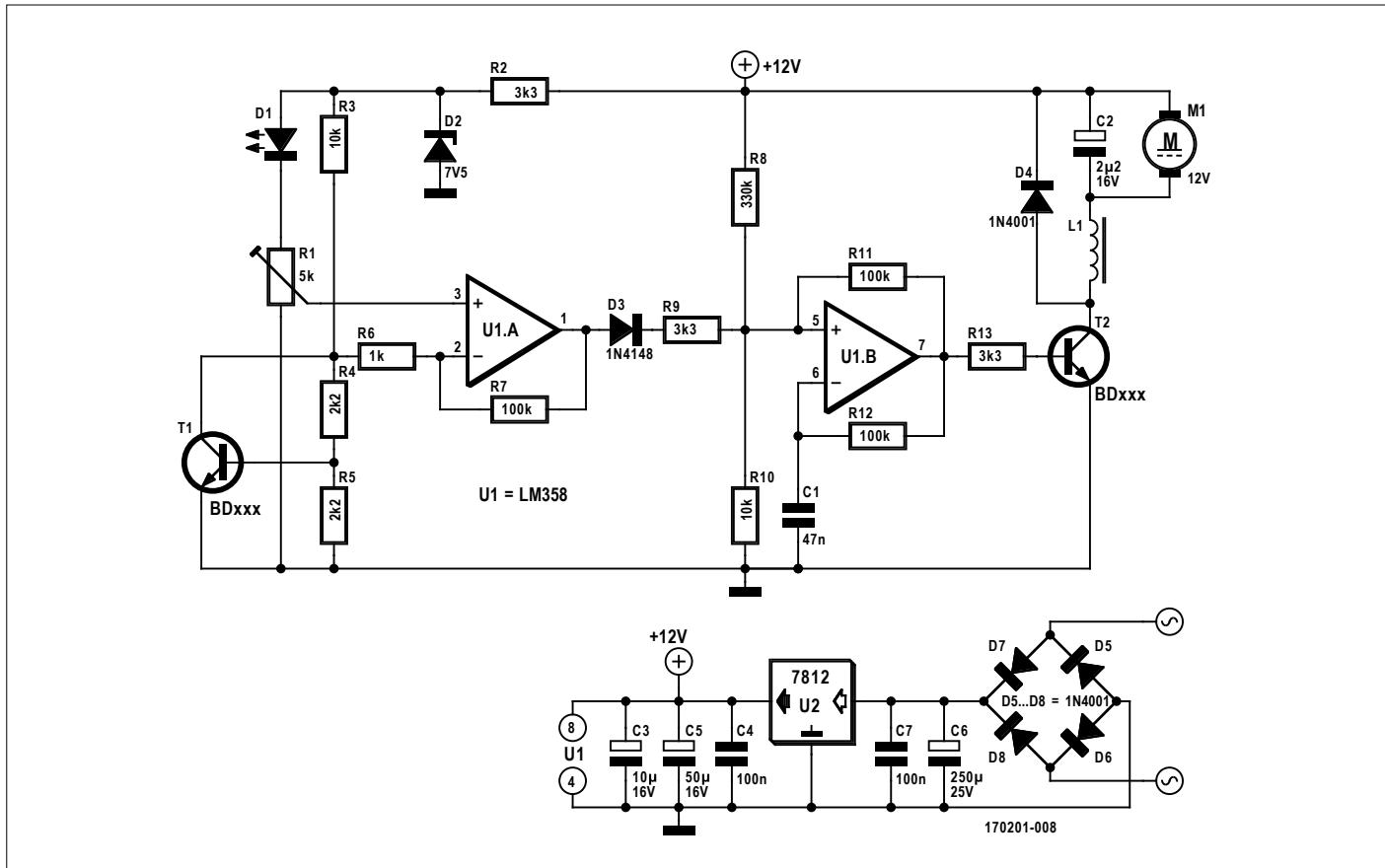
Analoog elektronica of software?

De schakeling die we gebruiken is zuinig met energie. Voor de aansturing stellen we twee varianten voor: de eerste met oude vertrouwde analoge elektronica, de tweede met een Arduino (Pro mini) met bijbehorende software (firmware). Deze aansturing stuurt met een PWM-signalen een regeleenheid aan, waarvoor we een transistor gebruiken. Door in het uitgangscircuit een zelfinductie, diode en condensator op te nemen, krijgt het geheel de kenmerken van een schakelende voeding met de daarbij horende eigenschappen. We beginnen met de analoge versie, zie **figuur 1**.

Analoog

Bij de analoge versie gebruiken we een transistor als temperatuursensor. We gebruiken de temperatuurcoëfficiënt van de basis-emitterovergang (ongeveer $-2,2 \text{ mV/K}$). We sluiten de basis aan op een spanningsdeler tussen collector en emitter (R_4, R_5), waarmee we de afgegeven spanning vergroten. Deze spanning wordt door U_{1A} vergeleken met de spanning aan de loper van potmeter R_1 . Met R_1 kunnen we dus de ventilatorregeling beïnvloeden. R_1 krijgt zijn voeding via zenerdiode D_2 , die de spanning in eerste instantie beperkt tot 7,5 V. De stroom voor R_1 loopt ook door D_1 (een blauwe of groene led), die de spanning verder vermindert. Deze opzet heeft een paar (kleine) voordelen:

- De zenerdiode heeft een waarde waarbij de temperatuurcoëfficiënt minimaal is, de led heeft een negatieve temperatuurcoëfficiënt, dus de spanning over de potmeter heeft nu de positieve temperatuurcoëfficiënt van de voorschakelweerstand voor de led (R_2), waarmee de omgevings temperatuur 'gemeten' wordt.



Figuur 1. In de analoge versie van de ventilatorsturing zien we de LM358 als a-stabiele multivibrator.

- T1 heeft een negatieve temperatuurcoëfficiënt en meet de temperatuur van het te koelen object. Per saldo is de verschilspanning zowel afhankelijk van de temperatuur van het te koelen object als van de omgevingstemperatuur, zodanig dat bij een hogere omgevingstemperatuur de stuurspanning groter wordt en de ventilator sneller gaat draaien.
- De led is meteen een indicator dat er spanning op de schakeling staat.

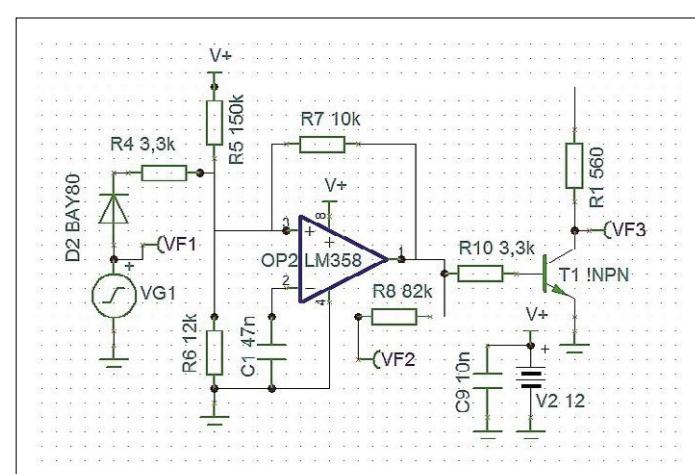
De sensorspanning voeren we toe aan opamp U1A, een LM358, die de spanning 100 maal versterkt. Dat is redelijk veel, zodat kleine temperatuurvariaties veel invloed hebben. Als dat niet nodig is, kan de versterking aangepast worden met de verhouding van de weerstanden R6 en R7. Wanneer we voor R7 een kleinere waarde nemen, is de versterkingsfactor ook kleiner. Dit is natuurlijk ook regelbaar te maken door voor R7 een potmeter te nemen met de loper aan één van de andere pennen verbonden. Met een 100k-potmeter kun je zo de versterking tussen (ongeveer) 1 en 100 maal instellen.

De tweede opamp van de LM358, U1A, staat geschakeld als een a-stabiele multivibrator. De niet-inverterende ingang is via een spanningsdeler (R8 en R10) en R9 en D3 aangesloten op de uitgang van U1A. Voor meer technische achtergrond over de werking van de opamp-multivibrator kun je goed op internet terecht, bijvoorbeeld op [1].

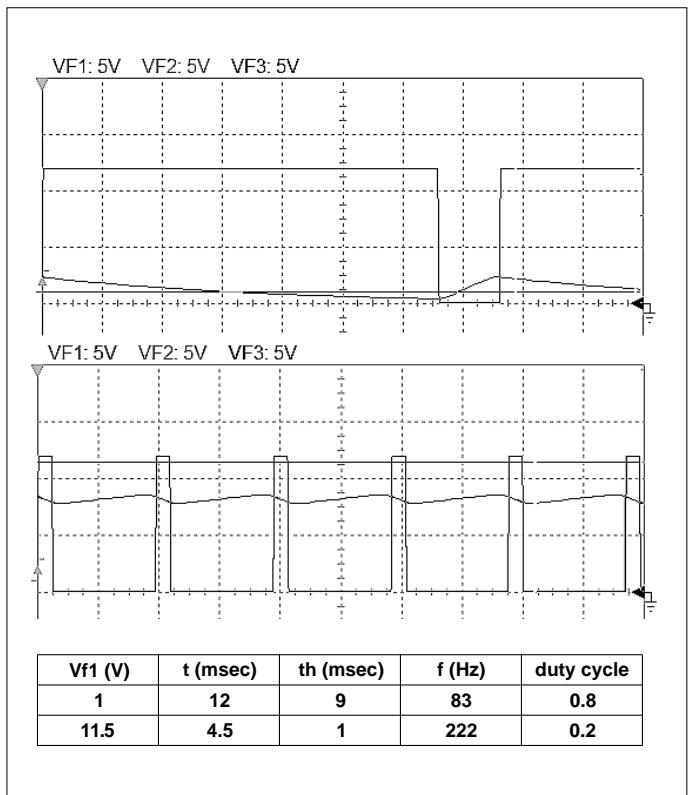
De duty-cycle van de multivibrator wordt bepaald door de spanning op de niet-inverterende ingang van de opamp. Is de spanning gelijk aan de halve voedingsspanning, dan zal de

duty-cycle 50% zijn. Wanneer de spanning op de niet-inverterende ingang lager is, zal de uitgangsspanning langer hoog moeten zijn om de opamp om te laten slaan. Zo kun je dus de duty-cycle beïnvloeden. Een lagere ingangsspanning betekent een hogere duty-cycle en omgekeerd.

In Tina TI V9 [2] kunnen we dit prima te simuleren. **Figuur 2** toont het schema dat we daarvoor gebruikt hebben. In **figuur 3** tonen we de resultaten van de simulatie. Hier is duidelijk te zien welke invloed de ingangsspanning heeft op de duty-cycle. Zoals de oplettende lezer misschien al was opgevallen, veran-



Figuur 2. De analoge schakeling is prima te simuleren in Tina TI.



Figuur 3. In de simulatieresultaten zien we duidelijk dat niet alleen de duty-cycle veranderd, maar ook de frequentie.

dert niet alleen de duty-cycle, maar ook de frequentie! Dat is in deze schakeling echter geen bezwaar.

Voor T1 is in het schema van figuur 1 een NPN-transistor aangegeven, maar dit kan even goed een PNP-transistor zijn, waarbij je dan de collector en emitter dient te verwisselen. Het voordeel kan zijn dat een transistor van het BD-type dan met zijn collector aan massa is verbonden, zodat bij de montage van de sensortransistor op het koelblok in principe geen isolatie nodig is en de warmteoverdracht beter is.

Bij L1 is opzettelijk geen waarde vermeld in het schema. Deze wordt namelijk experimenteel bepaald: gewoon in de voorraad kijken wat er ligt en uitproberen hoe de schakeling zich gedraagt. Zijn aanwezigheid werkt als laagdoorlaatfilter voor het PWM-signal en geeft een rustigere loop van de beschikbare ventilator (waar ook geen exact type van is vermeld).

Digitaal

Bij de microcontrollergestuurde versie zijn we uitgegaan van een Arduino Pro Mini. Dat zijn bordjes die voor weinig geld

op Ebay en dergelijke te vinden zijn en voor ons doel prima geschikt zijn. Het is mogelijk om de schakeling nog kleiner te maken door gebruik te maken van een Digispark met de Attiny85, maar zover zijn we (nog) niet gegaan. In principe kan het met iedere Arduino werken, maar het gemakkelijkst is om er eentje te gebruiken met dezelfde microcontroller. Op de website bij Arduino zelf [3] is een handig specificatieoverzicht te vinden.

De schakeling is eenvoudiger dan de analoge versie, zie het schema in **figuur 4**. Wat hier meteen in opvalt, is dat voor de temperatuuropnemer een gewone signaal diode is toegepast en niet een transistor zoals bij de analoge versie. Dat heeft te maken met de keuze van de referentiespanning van de Arduino. Hier hebben we gebruik gemaakt van de interne referentiespanning van 1100 mV van de Arduino. Met een transistor als thermosensor zou de ingangsspanning al meer dan de referentiespanning zijn. En dat kan de microcontroller niet verwerken. De resolutie van de interne A/D-omzetter is 10 bits, hetgeen neerkomt op een stapspanning van 1,1 mV. Deze stap komt bij een enkele diode neer op ongeveer 0,5 K, wat voldoende nauwkeurig is voor deze toepassing.

Omdat zowel de referentiespanning van de Arduino als de eigenschappen van de gebruikte diode per exemplaar variëren, is het aan te bevelen om een aantal ijkingen uit te voeren. Dit doe je door de spanning over de diode bij 0 °C en bij 100 °C met de Arduino te meten en uit te lezen. Als de spanning bij deze twee punten bekend is, kun je eenvoudig voor alle tussenliggende waarden uitrekenen welke temperatuur de sensor heeft. Ook kun je dan uitrekenen welke waarde het PWM-signal moet aannemen bij elke gemeten temperatuur.

Natuurlijk kun je er ook voor kiezen om de temperatuur met een digitale opnemer te meten, zoals de 18B20. Maar die ligt niet altijd voor het graaien in het rommelbakje en dan is het goed om te weten dat het ook met een gewone diode kan. De 18B20 kan zonder aanpassingen op de plaats van de diode aangesloten worden. Hij werkt dan in de ‘one-wire-mode’, wat voor korte draadlengtes geen enkel bezwaar is.

Bij de software is overigens nog ‘ruimte voor verbetering’. De temperatuuruitlezing functioneert naar behoren, maar het omschakelen van het PWM-signal als gevolg van de temperatuur is bijvoorbeeld nog niet gecheckt.

Motoraansturing en voeding

In onze schakeling zijn we uitgegaan van een eenvoudige 12V-ventilator met niet al te veel vermogen. Voor de aansturing hiervan gebruiken we een BD135 of iets dergelijks (T2). De collector is via een smoorspoel (L1) en aanverwante componenten met de ventilatormotor gekoppeld.

Natuurlijk is de schakeling niet beperkt tot het gebruik van een BD135. Met een krachtigere transistor zijn veel zwaardere belastingen te sturen (let dan ook op of de Arduino of de LM358 de benodigde stuurstroom voor de zwaardere transistor kan leveren).

Voor de voeding van beide versies van de schakeling kunnen we terugrijpen op de oude vertrouwde 7812-stabilisator. Diodenennetwerk D5-D8 zorgt ervoor dat zowel gelijk- als wisselspanning (tot maximaal zo'n 18 VRMS, afhankelijk van de koeling en belasting van de 7812) als voeding aangesloten kan worden. Verder zijn alleen een aantal buffercondensatoren nodig voor de correcte werking van de 7812. De meeste stroom zal de ventilator gebruiken, dus wanneer u bij de voeding zorgt voor

IN DE STORE



→ JOY-iT Uno R3

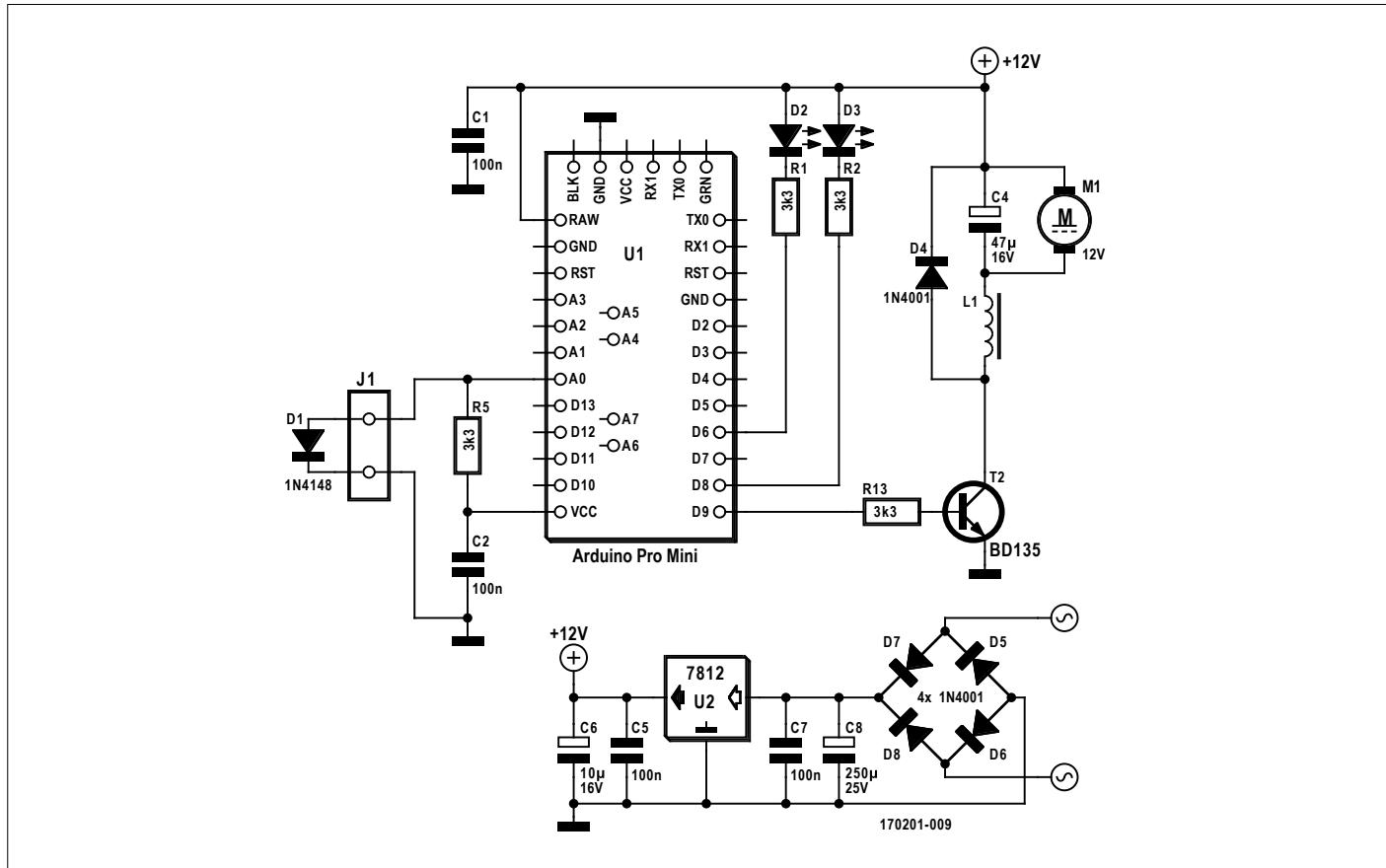
www.elektor.nl/joy-it-uno-r3

→ Elektor Sensor Kit

www.elektor.nl/arduino-sensor-kit

→ Boek: Arduino in Control

www.elektor.nl/arduino-in-control-3de-versie



Figuur 4. In de digitale versie zien we dat het eerste gedeelte vervangen is door de Arduino Pro mini en dat het tweede gedeelte zo goed als onveranderd kan blijven.

een kleine overcapaciteit van bijvoorbeeld 100 mA bovenop de nominale stroom van de ventilator, zit u vrijwel altijd goed. De Arduino Pro mini voeden we vanuit de 12V-stabilisator. Hij heeft zelf een stabilisator aan boord die de spanning redueert tot 5 V voor de microcontroller. Deze 5 V staat ook ter beschikking aan de VCC-pen van de module. Daar maken we graag gebruik van om de sensordiode van spanning te voorzien.

Printontwerp

Voor deze twee schakelingen heeft de auteur twee PCB's ontworpen die ter download beschikbaar staan op de webpagina bij dit project [4]. De print voor de digitale versie heeft een aantal uitbreidingen ten opzichte van de analoge versie:

- Hij is uitgebreid met twee uitgangen voor bijvoorbeeld een led of een optocoupler. De schakelpunten daarvoor zijn eenvoudig in de software op te nemen.
- Er is een montagemogelijkheid voor een header waarop +5V, GND, A4 en A5 beschikbaar zijn. Hierop kan bijvoorbeeld een module die via I2C communiceert aangesloten worden.
- Het is mogelijk de ventilator met een externe voeding te voeden. Hiervoor is een header met jumper voorzien. Als de spanning van de externe voeding hoger is dan 12 V, dient erop gelet te worden dat de BD135 daarmee overweg kan. Mogelijkkerwijs is een andere transistor (en/of koeling) noodzakelijk. Ook de werkspanning van C4 dient hiervoor voldoende hoog te zijn.

Aan de slag

Beide schema's zijn getekend in Ulticap 5.72 en de PCB's zijn in Sprint 5 ontworpen. De software in de vorm van een Arduino-sketch is door de auteur ter beschikking gesteld. Voor het programmeren is de gratis Arduino-IDE benodigd [5]. De Ulticap-schema's, Sprint-layouts en Arduino-sketch zijn te vinden op [4]. Met dit alles kunt u aan de slag!

170201-01

Weblinks

- [1] Werking van de opamp-multivibrator:
www.electronics-tutorials.ws/opamp/op-amp-multivibrator.html
- [2] TINA-TI: www.ti.com/tool/TINA-TI
- [3] Specificatieoverzicht Arduino:
www.arduino.cc/en/products.compare
- [4] Webpagina bij dit project :
www.elektrormagazine.nl/170201-01
- [5] Arduino-IDE: www.arduino.cc/en/Main/Software

De NXP Cup 2019 groter en beter dan ooit!

Clemens Valens (Elektor Labs)

De NXP Cup is een jaarlijkse wereldwijde wedstrijd voor studententeams en roboticaclubs die zich met elkaar willen meten in het bouwen en programmeren van en racen met een volledig autonoom smartcar-model op een voorgeschreven parcours. En niet alleen voor de lol, want het winnende team wordt naar Silicon Valley gevlogen (en terug)!

Het doel van de NXP Cup is het bouwen van en racen met *pro-level* autonome auto's, maar dan zonder pro-level budget. Dat kan alleen worden bereikt met zeer goed ontwerpen en geoptimaliseerde hard- en software, een hele klus.



De NXP Cup is een uitstekende kans voor technische studenten om vertrouwd te raken met de vaardigheden die ze in hun professionele carrière ook nodig zullen hebben. Of, zoals Lars Reger, CTO van NXP Automotive het stelt: "De technologie, de universitaire ondersteuning en het fantastische partner-ecosysteem bieden studenten kansen om essentiële vaardigheden zoals samenwerking te leren en vertrouwd te raken met de kerntechnologieën om autonome robots te bouwen."

Het realiseren van een autonome auto die in deze wedstrijd kan meedoen en – zo mogelijk – kan winnen, vereist:

- het programmeren van embedded software en basale hardwarekennis;
- het optimaliseren van de hard- en software voor de motorbesturing voor een zo groot mogelijke snelheid en nauwkeurigheid;
- het integreren van een camera om de auto in staat te stellen het parcours te volgen.



Om de deelnemende teams te helpen hun doelen te behalen, zijn voor de 2019-editie van de NXP Cup de regels herzien, waardoor meer vrijheid ontstaat:

- onbegrensd aantal sensoren;
- vrije keuze van een onbegrensd aantal NXP-processoren;
- meer uitdagingen om extra punten te behalen met zaken als tijdrit, figuur-8, obstakels vermijden en zich houden aan de maximum snelheid.

Naast de soepeler regels en de extra uitdagingen is ook het parcours zelf verbeterd; dit biedt nu een betere grip, waardoor hogere snelheden mogelijk zijn. ▶

180692-03

Weblink

[1] NXP Cup EMEA:

<https://community.nxp.com/groups/tfc-emea>



Stappenmotorbesturing met antiresonantietechniek

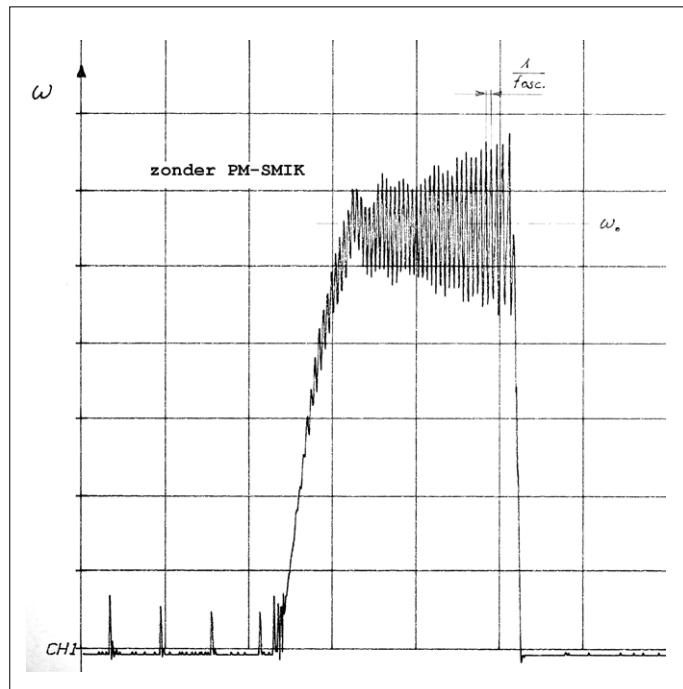
groter rendement en minder trillingen bij stappenmotoren

René Merz (Magnetron.ch)

Stappenmotoren zijn uit de moderne elektromechanica niet meer weg te denken. Je komt ze overal tegen: in 3D-printers, harddisks en in talloze machines. Maar stappenmotoren zijn niet zonder problemen. Ze hebben vaak last van resonantieverschijnselen en het rendement is vatbaar voor verbetering. In dit artikel beschrijven we hoe we daar iets aan kunnen doen.

Stappenmotoren zijn er in soorten en maten. Ze kunnen heel verschillende stapgroottes, frequentiegebieden en bouwvormen hebben, passend voor de betreffende applicatie. Vooral hybride stappenmotoren en klawpoolmotoren worden steeds vaker gebruikt voor lineaire spindel-aandrijvingen in 3D-printers.

Vaak worden ze daarbij ingezet onder niet-optimale omstandigheden, dus bij hoge frequenties en kleine belasting, waarbij heel gemakkelijk storende verschijnselen zoals mechanische resonanties optreden. Om stappenmotoren optimaal te kunnen inzetten, moeten we daarom bij het aansturen al rekening houden met die verschijnselen op basis van een Raspberry Pi en een passende HAT.



Figuur 1: Typische rotoroscillaties bij kleine stappenmotoren, gemeten over een shunt. X = 500 ms/div en Y = 200 mV/div. Zonder antiresonantiemaatregelen ontstaat een oscillatiefrequentie van 34 Hz in de rotor. Na het opslingereren blokkeert de motor.

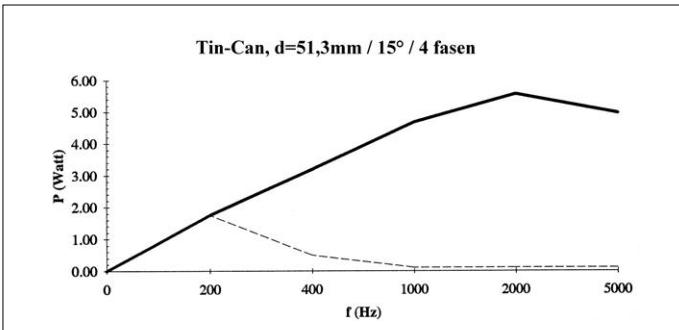
Resonanties

We maken bij stappenmotoren onderscheid tussen de eigenresonanties en het zogenaamde parametrische instabiliteitsgebied. Eigenresonanties bij frequenties van 1...300 Hz kunnen we binnen de perken houden door mechanische tegenmaatregelen (keuze van de overbrenging, damping etc.). In dit bereik haalt een stappenmotor typisch een rendement van minder dan 10 %.

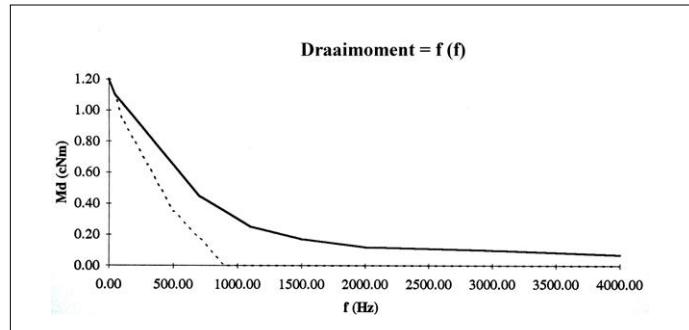
Boven 300 Hz (resp. stappen/s) treden in bepaalde frequentiebanden zogenaamde parametrische resonanties op. Daarbij ontstaan in bepaalde frequentiegebieden oscillaties van de rotor om zijn gewenste positie (zie **figuur 1**). Deze trillingen van de rotor kunnen ten gevolge van die resonantie een grote amplitude bereiken (opslingereng), wat ertoe kan leiden dat de rotor niet meer synchroon werkt met de aanstuurfrequentie en stappen 'mist' of uiteindelijk zelfs stilstaat. Daardoor kunnen we, zonder de juiste tegenmaatregelen, het vermogensbereik niet volledig benutten (zie **figuur 2**). De gestippelde lijn in **figuur 3** laat zien, dat zonder tegenmaatregelen zowel draaimoment (koppel) als maximale aanstuurfrequentie duidelijk suboptimaal zijn.

Een oplossing

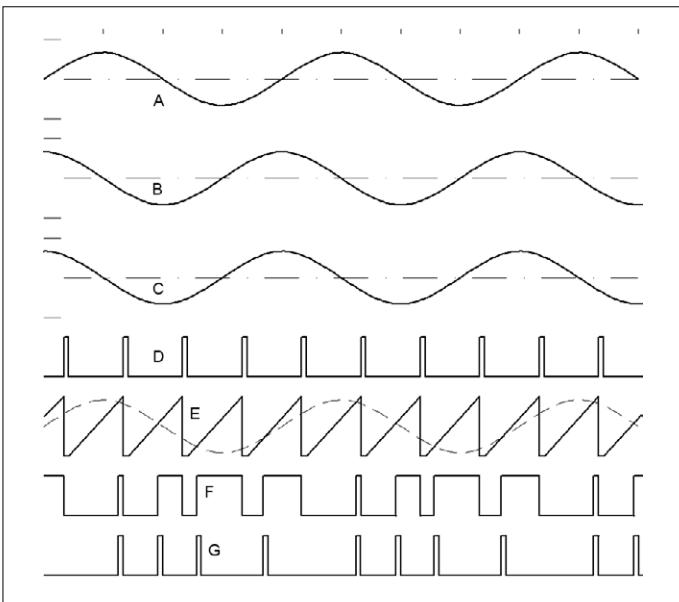
Voor het meten van resonantieverschijnselen kunnen we met de oscillaties corresponderende signalen aftappen met behulp van een shuntweerstand: de spanningsval is evenredig met de



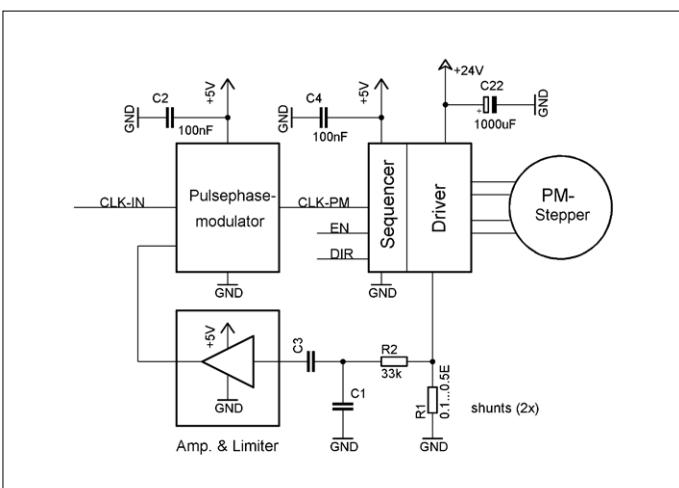
Figuur 2: Mechanisch motorvermogen in W bij $I = 0,4$ A/fase en ED = 30 %. De gestippelde curve geeft het gedrag zonder antiresonantieschakeling. Daarmee verliest u niet alleen twee derde van het haalbare vermogen, maar ook de mogelijkheid om met hogere frequenties te werken.



Figuur 3: Draaimoment als functie van de frequentie. De doorgetrokken lijn toont de enorme verbetering door de antiresonantieschakeling.



Figuur 4: Faserelaties van de signalen van de motorsturing. A: rotorsnelheid ω ; B: lasthoek rotor γ ; C: $U = f(I)$, in fase met signaal B; D: clock-in (f_0) van de frequentiegenerator; E: zaagtandsignaal en U' na het filter, 90° in fase verschoven; F: PWM; G: clock-PM, voedt de sequencer.



Figuur 5: Werkingsprincipe van de antiresonantieschakeling. Het signaal van een shunt wordt gebruikt voor het verschuiven van de fase van de stuurstroombussen.

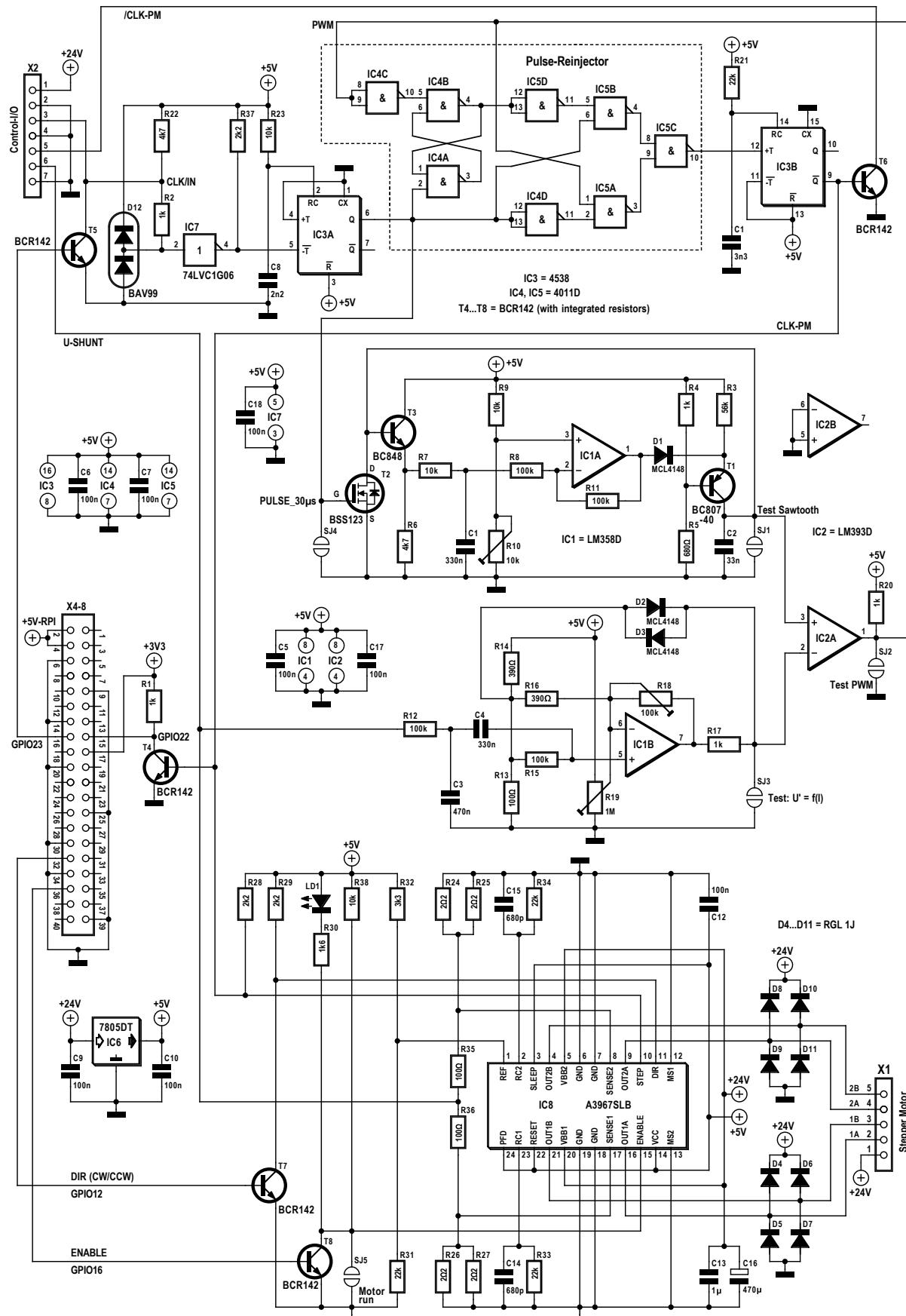
stroom door de rotorwikkeling. In **figuur 4** zien we de geïdealiseerde faserelaties tussen de rotor en de stuursignalen.

In **figuur 5** zien we het principe van een mogelijke tegemaatregel, die in 1986 is ontwikkeld door Pierre Boillat [1]. De patentbescherming van dit systeem is intussen afgelopen. Bij kleinere klauwpoolmotoren (tin-can-motoren) hebben de oscillaties typisch een frequentie van 5...80 Hz. Het shuntsignaal (onderaan) wordt gefilterd en versterkt en stuurt een pulsfasemodulator aan. Het uitgangssignaal CLK-PM stuurt de sequencer-ingang van de eigenlijke motordriver aan met een gecorregeerde faseverhouding. Zo'n pulsfasemodulator kan worden ingezet als de eigenlijke pulsgenerator (meestal een microcontroller) zelf niet geschikt is om de fasemodulatie uit te voeren.

Op deze manier kunnen we heel hoge stapsnelheden tot wel 15.000 s/sec bereiken, waardoor we de stappenmotoren efficiënter kunnen benutten. Een ander voordeel is dat we geen speciale versnellingsprofielen hoeven te gebruiken om de instabiliteitsgebieden te passeren. Het is ook heel prettig dat een op deze manier gestabiliseerde stappenmotor ook minder geluid maakt. Belangrijk om te weten: deze methode van stabilisatie heeft niets te maken met de PWM-chopper-stroomsbesturingen die tegenwoordig in veel geïntegreerde stappenmotordrivers zijn ingebouwd.

Antiresonantieschakeling

De auteur heeft de schakeling van **figuur 6** ontwikkeld om het principe in de praktijk te demonstreren. Het is een uitbreidingsmodule (HAT) die rechtstreeks op een Raspberry Pi van het formaat A+ kan worden gestoken. Het inkomende signaal van een externe microcontroller wordt aangesloten op CLK-IN, dus op pin 3 van de printkroonsteen X2. Het signaal wordt eerst geconditioneerd met inverter IC7 en bereikt dan de monoflop IC3A, die triggert op de negatieve flanken. Op uitgang Q staan dan impulsen met een constante lengte van ongeveer 30 µs. Deze impulsen ontladen via MOSFET T2 condensator C2 van de rechthoek/zaagtand-converter rondom IC1A en T1...T3. De geregelde stroombron rondom T1 zorgt ervoor dat C2 nadat hij is ontladen met T2 door T1 met een lineaire stroom wordt opgeladen tot de maximale waarde. Het resultaat is een heel zuiver zaagtandsignaal op de positieve ingang (pin 3) van comparatoren IC2A. Samen met het modulatiesignaal U' op de negatieve ingang (pin 2) zien we aan de uitgang (pin 1) een impulsue signaal met de gewenste (lineaire) fasemodulatie.





Figuur 7: Signalen van de antiresonantieschakeling bij een stapgroote van 15° en 901,75 Hz. C1: 30- μ s-impulsen op SJ4; C2: zaagtand op SJ1; C3: PWM-signalen op SJ2; C4: shuntsignaal, $U = f (IT)$ op X2-6.

Voor een groot frequentiebereik moet de zaagtand een amplitude hebben die niet afhankelijk is van de frequentie. Om dat te bereiken, wordt de stroombron rondom T1 aangestuurd door IC1A. De amplitude wordt constant gehouden door de gemiddelde waarde (de DC-component) van de zaagtand te bufferen met T3 en te filteren met het laagdoorlaatfilter R7/C1, waarna die wordt gebruikt als tegenkoppeling. Het stuursignaal wordt via D1 naar de stroombron geleid.

Bij een heel snelle sweep (bijvoorbeeld een overgang van 0 Hz naar 4 kHz in slechts 200 ms) is deze regeling te traag vanwege de tijdconstante van R7/C1. Dan zou de amplitude van de zaagtand instorten en daardoor zouden impulsen aan het einde van de sweeps bij de comparator IC2A kunnen wegval-

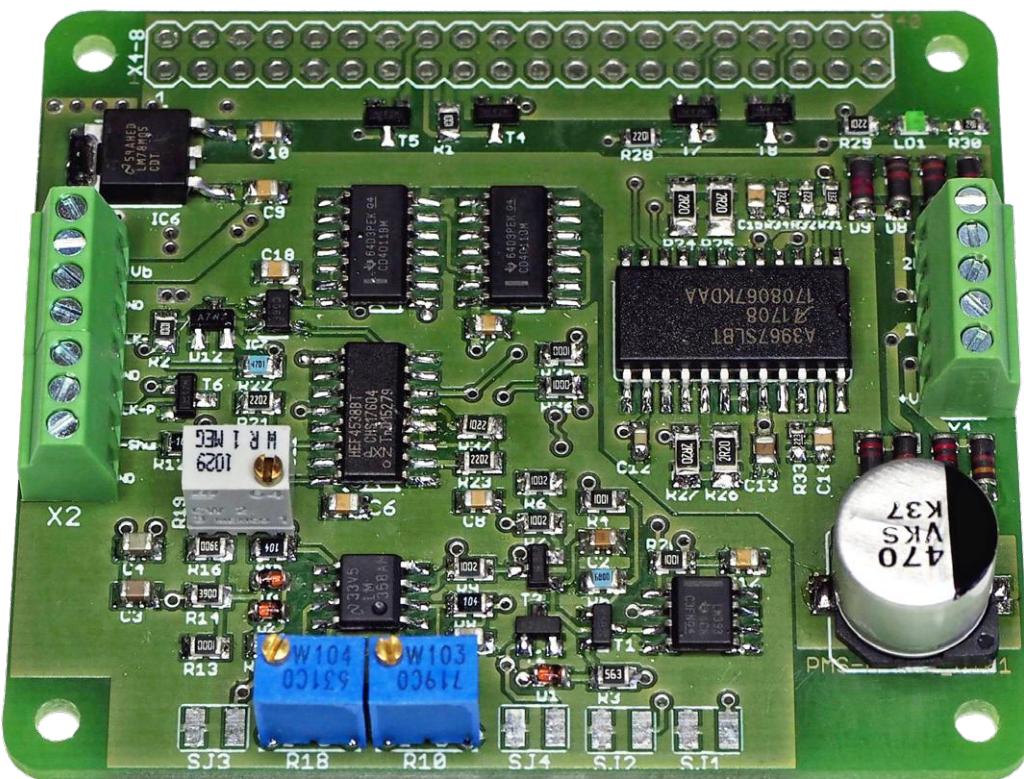
len. Dat zou fataal zijn voor een positioneringssysteem! Om het uitbliven van impulsen te voorkomen, is de logica rondom de NAND-gates IC4 en IC5 (pulse-reinjector) toegevoegd. De monoflop IC3B is nodig omdat alleen de positieve signaalflanken worden gemoduleerd, terwijl sommige gangbare sequencer-drivers triggeren op de negatieve flank. Dankzij de open-collector-schakeling kan driver-schakeling aan de uitgang voor niveau-aanpassing zorgen met behulp van een pull up-weerstand.

De modulatiespanning U' wordt verkregen uit het signaal op de shuntweerstanden R24...R27 door een banddoorlaatfilter met R12, C3, C4, R15 en versterking met IC1B. Het kan nodig zijn (in uitzonderlijke gevallen) om het banddoorlaatfilter aan te passen aan het traagheidsmoment van de rotor. Met R18 kan de versterking ingesteld worden tussen een factor 1...250. In de praktijk is gebleken dat een versterking met een factor 50 meestal toereikend is. De pads SJ1...SJ4 op de print dienen als testpunten voor een oscilloscoop. De in het oscilloscoop in **figuur 7** weergegeven signalen kunnen met de instelpotmeters R10, R18 en R19 worden ingesteld. R19 dient voor het instellen van het DC-niveau op de uitgang van IC1B. Op SJ3 moet de helft van de piekwaarde van de zaagtand SJ1 staan en dit moet tussen 1...1,3 V liggen.

Helemaal aan het eind zit IC8: de geïntegreerde stappenmotordriver van Allegro [2]. Het IC A3967SLB bevat een sequencer, een PWM-chopper-stroomsturing voor microstepping en een eindtrap met bipolaire transistoren voor 0,75 A/fase. De referentiespanning op pen 1 van IC8 kan met de spanningsdeler R31/R32 worden aangepast aan de stroomspecificaties van de stappenmotor.

RPi-HAT

De HAT-print in **figuur 8** past dankzij de 2 x20-polige header rechtstreeks op een RPi van het type A+. Die kan dan de benodigde besturingssignalen voor de motor voor allerlei praktische toepassingen genereren. De besturingspulsen met een niveau van 3,3 V worden via poort GPIO23 en T5 naar de antiresonantieschakeling geleid. Onze schakeling werkt met niveaus van 5 V. De niveau-aanpassing voor CLK-IN wordt verzorgd door een pull up-weerstand (R22). Ook voor de besturinglijnen /Enable (vrijgave van de motor, actief laag) en CW/CCW (draairichting) is gezorgd. Als de HAT op een RPi moet worden gestoken, zijn natuurlijk nog vier afstandsbusen en schroeven nodig voor de mechanische bevestiging. De beide boards



Figuur 8: Zo ziet het prototype van een stappenmotorbesturing met antiresonantieschakeling in de vorm van een RPi-HAT voor de RPi model A+ eruit.

moeten elektrisch gescheiden voedingen hebben (uit twee aparte netvoedingen). Vrijwel alle kleine tweefase-stappemotoren zijn geschikt voor deze test, of het nu klappool- of hybride typen (NEMA17) zijn, bipolaire of unipolaire, als de weerstand van de wikkelingen maar in het bereik van 5...100 Ω /fase ligt en het vermogen tussen 1 en 15 W.

Bij een weerstand van minder dan 5 Ω /fase moet de stroominstelling aan de driver erg voorzichtig worden uitgevoerd. De motorlijnen mogen in geen geval onderbroken worden als de schakeling in werking is.

Hoewel IC8 is geconfigureerd voor bipolaire stappemotoren, kunnen ook unipolaire typen worden aangesloten aan de schroefklemmen X1. De middenaftakking van de spoelen komt dan aan X1-1 (+24 V).

Als u de HAT-kaart wilt nabouwen, vindt u de schema- en layout-bestanden op de webpagina bij dit artikel [3]. De **technische gegevens** zijn te vinden in de tabel hiernaast. Tenslotte hoeft u alleen uw eigen toepassing met een geschikte pulsgenerator te programmeren op de RPi. Daar kunnen we u niet bij helpen...

Overwegingen en achtergronden

De auteur nam in 2014 een 3D-printer in gebruik, waarvan verschillende aandrijfassen waren uitgevoerd met bipolaire, hybride stappemotoren (vierdraads, NEMA17). De wikkelingen zijn hier in serie geschakeld. De gekozen aanstuurfrequenties lagen helaas in frequentiegebieden waar de behuizing van de printer sterk resoneerde. De grote hybride stappemotoren hadden vervangen kunnen worden door kleinere versies, als in de besturing een antiresonantieschakeling en een aangepaste overbrenging zouden zijn gebruikt. Veel gebruiker gaan voor het dempen in het zelfresonantiegebied over op microstepping-modus, ook vanwege het grotere oplossend vermogen. Maar een groter oplossend vermogen betekent nog lang geen grotere nauwkeurigheid.

De oorzaak ligt in fouten in de geometrie van de stator, spreiding in de magnetische eigenschappen van het metaal van de stator en afwijkingen van de gewenste stroomwaarde (PWM-chopper, 1/8...1/16 stap) bij kleine fasestromen!

Bij gelijke aanstuurfrequentie van de sequencer en acht microstappen wordt trouwens de snelheid van de motoras gereduceerd met een factor 8. In de formule voor het mechanische afgegeven vermogen...

$$P_{\text{mech}} = M_d \times \omega$$

... zien we namelijk dat de hoeksnelheid ω van de motoras, naast het draaimoment (M_d) ook een belangrijke rol speelt: De last moet immers zo snel en nauwkeurig mogelijk kunnen worden gepositioneerd! De microcontroller zou één of meerdere geschikt pulsgevers aan boord moeten hebben. Het is dus belangrijk dat de μ C over voldoende geheugen (NV-RAM of Flash) beschikt om de waarden voor frequenties, aantal burstimpulsen en zaagtanden op te kunnen slaan. Bij de in de RPi gebruikte SoC's is dat geen groot probleem.

Om in de toekomst gemakkelijker stappemotoren in te kunnen zetten, zouden de fabrikanten daarom ook het rendement en/of de verhouding tussen opgenomen elektrisch vermogen en afgegeven mechanisch vermogen als functie van de aanstuurfrequentie in de datasheets duidelijk moeten documenteren. Om werkelijk alles uit stappemotoren te kunnen halen, zijn

Technische gegevens

Voedingsspanning:	7...24 V
Aanstuurfrequentie-burst:	1 mHz...25 kHz
Tijdvertraging CLK-In/Out:	ca. 5 ms bij 1 kHz
Fasestroom max.:	0,75 A/Ph

complexe algoritmes met versnellings- en vertragingsprofielen essentieel. Natuurlijk brengt dat extra programmeerwerk met zich mee.

Het is verstandig om zoveel mogelijk te voorkomen dat de motor in het gebied van zijn eigenresonanties wordt gebruikt. Dat stelt dus ook voorwaarden aan de manier van inbouwen: zorg ervoor dat de ophanging van de motor een grote inwendige demping heeft. Bij de afweging of u een spelingsvrije overbrenging (Harmonic-drive) of een tandriemoverbrenging wilt gebruiken, moet u letten op de kosten/baten-verhouding. Misschien zijn er in de nabije toekomst goedkope driver-IC's verkrijgbaar waarin een klasse-D-eindtrap zelfs meteen gefilterde sinus- en cosinus-uitgangssignalen voor de motorfasen genereert. Ook het gebruik van statormaterialen met kleinere magnetiserings- en hystereseverliezen zou in de toekomst welkom zijn, want tegenwoordig zijn chopper-frequenties van meer dan 18 kHz al gebruikelijk. ◀

180279-03

Over

René Merz werkte van 1985 tot 1990 bij de SAIA AG (nu Johnson Electric International AG in Murten in Zwitserland) aan magnetiseringsspoelen en aan de ontwikkeling van nieuwe magnetische rotoren voor stappen- en synchroonmotoren. Sindsdien houdt René Merz zich ook bezig met besturingen voor stappenmotoren. Sinds eind 1992 is hij eigenaar van de firma Magnetron Labs Merz.

Links & literatuur

- [1] Pierre Boillat, "Restraining the instability of a stepper motor", US-Patenten US4673855 en US4683409
- [2] Allegro A3967: <https://bit.ly/2UoRnk1>
- [3] Webpagina bij dit artikel: www.elektormagazine.nl/180279-03



IN DE STORE

- Boek (Duitstalig): "Motorsteuerung mit Arduino und Raspberry Pi" www.elektor.nl/motorsteuerung-mit-arduino-und-raspberry-pi
- Boek (Engelstalig): "Motors for Makers" www.elektor.nl/motors-for-makers
- Mendocino-Solar-Motor X-8: www.elektor.nl/mendocino-motor-x-8

Achteruitkijkradar voor tweewielers

veiliger op de fiets met een simpel schakelingetje

Laurent Labbe (Hong Kong)



Ik woon al een hele tijd in Azië (eerst in Shenzhen in China, tegenwoordig in Hong Kong) en ik stap regelmatig op de fiets, hetzij om naar mijn werk te gaan, hetzij om te trainen voor een triathlon. Maar ik moet wel voortdurend al het verkeer om me heen in de gaten houden. En omdat ik van het steeds omkijken geen nekkamp wil krijgen, heb ik een schakeling bedacht rond een achtbenige microcontroller.

Heel lang geleden, toen ik op de fiets door het stadsverkeer naar school ging, had ik achteruitkijkspiegels links en rechts op mijn stuur, zodat ik kon zien of en welke auto's me zouden gaan inhalen. Zulke accessoires zijn tegenwoordig echter wat moeilijker te vinden. Daarom bedacht ik een systeem dat me waarschuwt voor verkeer achter me, zonder dat ik daarvoor mijn hoofd hoeft om te draaien en waardoor ik me dus op de weg voor me kan concentreren. In China komen de auto's en de bussen echt overal vandaan, en meestal nog onverwachts ook. Je moet dus heel goed opletten. In Hong Kong is het zelfs nog erger. In de smalle straten razen de bussen, taxi's en minibussen voortdurend rakelings langs je heen.

Het hier gepresenteerde systeem is een soort radar, bestaande uit twee ultrasone sensoren die onder een hoek van 90° zijn

gemonteerd. De sensoren meten met vaste tussenpozen de afstand tussen het fietszadel, waar de schakeling is geplaatst, en de voertuigen die zich achter of rechts van de fiets bevinden (het verkeer in Hong Kong houdt links). Er kan eventueel een derde sensor worden toegevoegd. Het prototype (**figuur 1**) heeft die niet, maar de firmware (die van [1] kan worden gedownload) kan er wel mee overweg. De positionering van de schakeling hangt af van aan welke kant van de weg het verkeer rijdt (links of rechts), dus van het land waarin u fietst. De sensoren moeten wijzen in de richting vanwaar het achteropkomend verkeer komt. Dus rijdt het verkeer links, dan zal de module meer naar rechts moeten wijzen, en andersom. Wanneer de module zo wordt aangebracht dat de sensoren elk een hoek van 45° ten opzichte van de lengterichting van de

fiets maken, blijkt dat geparkeerde auto's (links in Hong Kong en rechts in China) ten onrechte alarm veroorzaken.

Hoe waarschuwt je een fietser? Hij of zij moet de blik op de weg houden. Ik heb gekozen voor twee trilalarmen links en rechts onder het zadel, om aan te geven waar een auto vanaf komt. Het aantal trillingen (1, 2 of 3) geeft een idee van de afstand. U zou die 'trillers' eventueel ook links en rechts aan of in het stuur kunnen monteren, met langere bekabeling. Voor die trillers heb ik kleine motortjes met een excentrisch gewichtje uit oude mobiele telefoons gesloopt.

Dan is er nog de voedingskwestie. Het energieverbruik moet zo gering mogelijk zijn; daarom beperkt de schakeling zich tot één functie: de radar. Hij wordt gevoed uit een lithiumcel, die via een speciaal IC wordt bijgeladen (hier de MAX1555). Een spanningsregelaar vond ik niet nodig. Je zou de schakeling kunnen laden met een zonnecel; misschien wordt dat iets voor een volgende versie.

De schakeling

Ik wilde wel de batterijspanning kunnen meten. Daarom moest de microcontroller een interne referentiespanning aan boord hebben, maar desondanks zo klein mogelijk zijn (acht pootjes) om het energieverbruik laag te houden. Ik heb gekozen voor

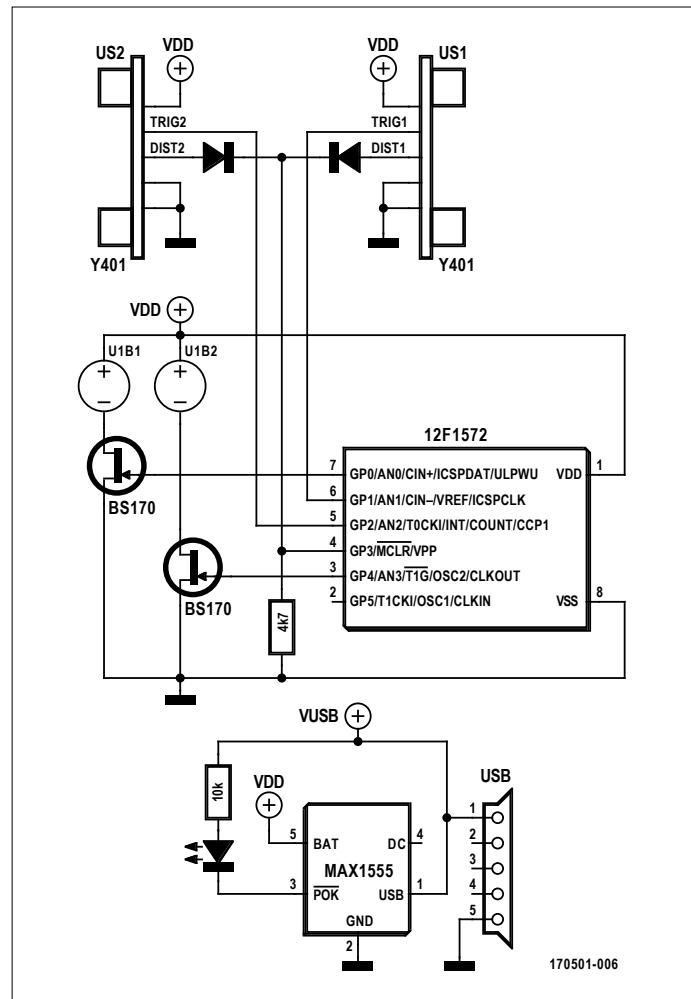
de 12F1512, die gevoed kan worden met een spanning van 3...5 V. Ultrasone sensoren zijn er in overvloed, maar er zijn er maar weinig die je kunt voeden met een spanning van 3...5 V. Met de types Y401 en US100 kan dat wel. We meten de afstand aan de hand van de duur van de uitgangspuls. Met deze sensoren zouden we de afstand ook via een seriële link als binaire waarde kunnen meten, maar die optie gebruiken we niet.

Het schema is heel simpel (**figuur 2**): twee µC-uitgangen GP1 en GP2 triggeren de twee ultrasone sensoren, en via een met diodes opgebouwde OF-poort meten we de tijd tot aan het retoursignaal. In de firmware vinden de triggering en de meting van de afstand na elkaar plaats, daardoor kunnen we volstaan met een OF-poort met een pulldown-weerstand.

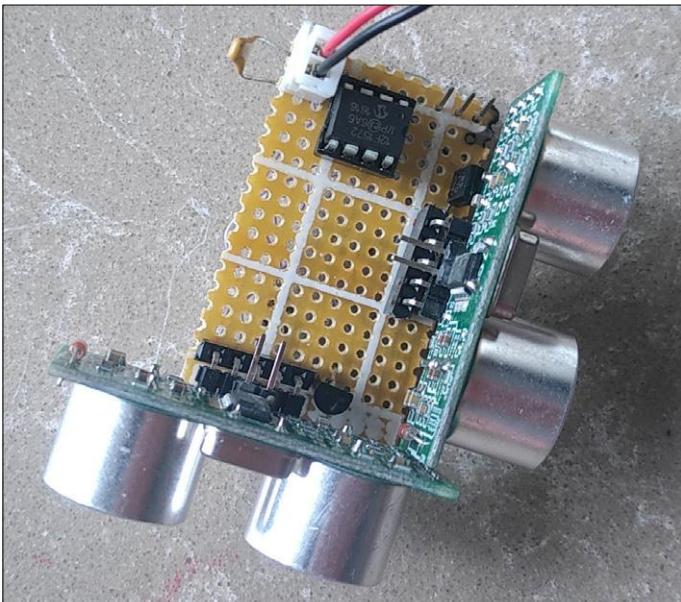
We meten met behulp van timer1, die de toestand op pin 4 meet (GP3, **T1GVAL**). Het timer-register wordt opgehoogd zolang pin 4 hoog is. Als pin 4 laag is, gebeurt er niets. De firmware wacht zolang het signaal laag is en meet vervolgens hoe lang het hoog is, of stopt de meting als die tijd de grenswaarde **TIME_LIMIT_WAIT** overschrijdt. Dat laatste is het geval als er geen obstakel is dat iets terugkaatst naar de sensor. De uitgang van de OF-poort gaat dus naar deze ingang, en start de meting met timer1. De tijdsduur van de puls wordt bijgehouden in het timer-register totdat de uitgang van de OF-poort nul wordt.



Figuur 1. Prototype van de auteur *in situ*, met een sensor naar rechts en naar achteren gericht. U ziet ook de bekabeling naar de trillers naar weerszijden van het zadelframe.



Figuur 2. Schema van de radar, simpel maar doeltreffend.



Figuur 3. De schakeling is opgebouwd op dubbelzijdige gaatjesprint.

De afstand in centimeters berekenen we met de volgende formule:

$$d = t \text{ (tijdsduur van het hoog-signal in } \mu\text{s}) * 34 / 200$$

De trillers worden geactiveerd met twee FET's van het type BS170.

Zoals gezegd wordt de module gevoed uit een lithium-accu (merk en type onbelangrijk). Via een MAX1555 kan de accu vanuit een USB-poort worden opgeladen.

De firmware

Ook de firmware is niet moeilijk. Elke twee seconden activeren we de linkersensor, meten de afstand en activeren we de linkertriller als er een obstakel is gedetecteerd; vervolgens doen we hetzelfde maar dan voor de rechtersensor en -triller. Als er een derde sensor is geplaatst voor het midden (uitgang GP5), dan activeert de firmware beide trillers tegelijk.

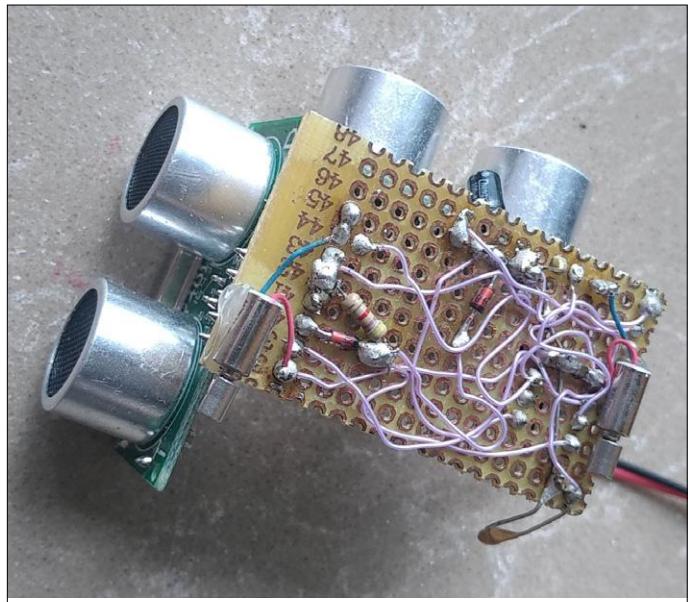
De afstand waarop gewaarschuwd wordt is vastgelegd in de firmware.

De firmware is ontwikkeld met de laatste versie van MikroC, dat is de enige versie die de 12F1512-microcontroller ondersteunt. De gratis versie volstaat, want het is maar een heel klein programmaatje. Ik gebruik een PICkit3 om de firmware in de µC te laden.

Montage op de fiets

Dit is niet het eenvoudigste onderdeel van dit project! Enkele tips voor zowel de sensorprint als voor de trillers:

- de radarprint komt onder het zadel. Ik heb dat gedaan met kabelbinders. U moet goed opletten dat de sensoren netjes horizontaal staan, anders 'detecteren' ze de bestrating...
- zoals hierboven uiteengezet dient u de sensoren uit te richten al naargelang aan welke kant van de weg het verkeer rijdt;



- de simpelste positie voor de trillers is onder het zadel.

Ik heb de trillers verpakt in krimpkous en ze elk met een kabelbinder aan de metalen beugels links en rechts onder het zadel bevestigd. Diverse tests wezen uit dat hier de minste beweging zit. Meer naar voren onder het zadel voel je met het bewegen van je benen de trillingen niet. Als u er niet tegen opziet om bekabeling naar het stuur te trekken, dan is dat waarschijnlijk de meest effectieve oplossing...

Dit 'veiligheidsschakelingetje' is best geschikt voor verdere perfectionering (sensoren, voeding...). Ik wens u veel plezier ermee! ▶

(170501-04)

Weblink

- [1] Webpagina bij dit artikel
www.elektormagazine.nl/170501-04



IN DE STORE

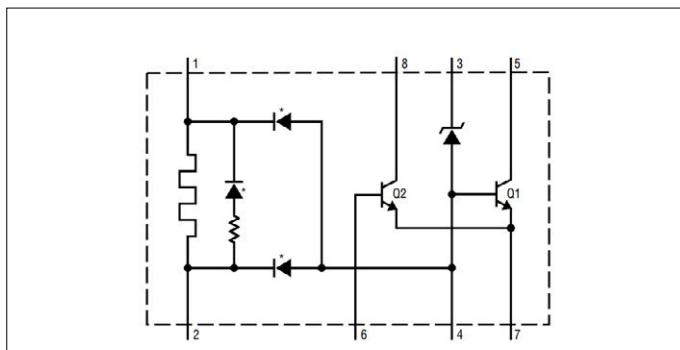
→ Boek (Franstalig):

'Les microcontrôleurs PIC pour les débutants', Rémy Mallard
www.elektor.fr/les-microcontroleurs-pic-pour-les-debutants

Spanningsreferentie LTZ1000 met 'begraafde zener' vreemde onderdelen

Neil Gruending (Canada)

Als je snel een nauwkeurige referentiespanning nodig hebt, is een bandgap-spanningsreferentie prima geschikt. Maar als de stabiliteit als functie van tijd en temperatuur belangrijker is dan de absolute uitgangsspanning, wat dan? Het antwoord is te vinden in de beroemde HP 3458A voltmeter: de LTZ1000-spanningsreferentie van Linear Technology. Laten we deze unieke 'begraaf-zener'-referentie eens van dichtbij bekijken.



in high-end voltmeters zoals de HP 3458A en in transferstandaarden zoals de Fluke 7001.

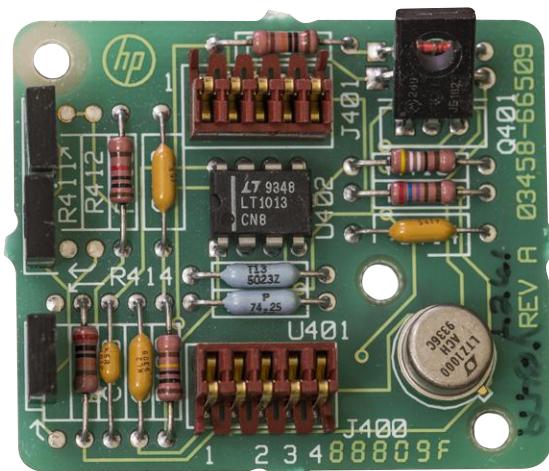
Deze prestaties hebben echter een prijs, want er zijn overal verborgen thermische effecten die de LTZ1000 kunnen beïnvloeden. Zo vormen de verbindingen tussen de Kovar-aansluitdraden en de koperen printspleten thermokoppels die spanningen tot $35 \mu\text{V}/^\circ\text{C}$ kunnen opwekken, en dus is het een vereiste dat alle draden dezelfde temperatuur hebben. Zelfs de weerstanden in de externe schakeling zijn van belang, omdat deze ook thermokoppeleffecten kunnen hebben. Daarom worden hiervoor draadgewonden of uiterst stabiele hermetisch gesloten exemplaren gebruikt.

Als u ermee wilt experimenteren: de LTZ1000 wordt nog steeds geproduceerd, en er zijn zelfs nog tweedehands spanningsreferenties uit de HP 3458 (figuur 2) [4] verkrijgbaar. Velen hebben zich bezig gehouden met het bouwen en testen van schakelingen [3] met het IC, maar de grootste uitdaging blijft toch hoe je daaraan gaat meten! ▶

(180464-04)

Figuur 1. Blokschema LTZ1000.

Zoals figuur 1 [1] laat zien, is een LTZ1000 een schijnbaar eenvoudig IC dat alleen de essentiële analoge onderdelen voor de referentie bevat. Q1 zorgt voor temperatuurcompensatie van de zenerdiode die met pen 3 is verbonden en Q2 dient als temperatuursensor voor het verwarmingselement tussen de pennen 1 en 2. Omdat de temperatuur invloed heeft op de zenerspanning, wordt het verwarmingselement met een extern regelcircuit op een constante temperatuur gehouden die iets hoger is dan de omgevingstemperatuur. De ontwerpers hebben de stabiliteit nog verder verbeterd door ervoor te zorgen dat alles in de (oven-)behuizing exact dezelfde temperatuur heeft. Het resultaat is een ultrastabiele 7V-referentie. Ontwerper Carl Nelson zei hierover: "Het doel is om een onderdeel te maken dat zo goed is dat je het niet kunt meten" [2]. De stabiliteit wordt alleen overtroffen door een standaardcel of Josephson-junctie-referentie. Daarom vinden we de LTZ1000



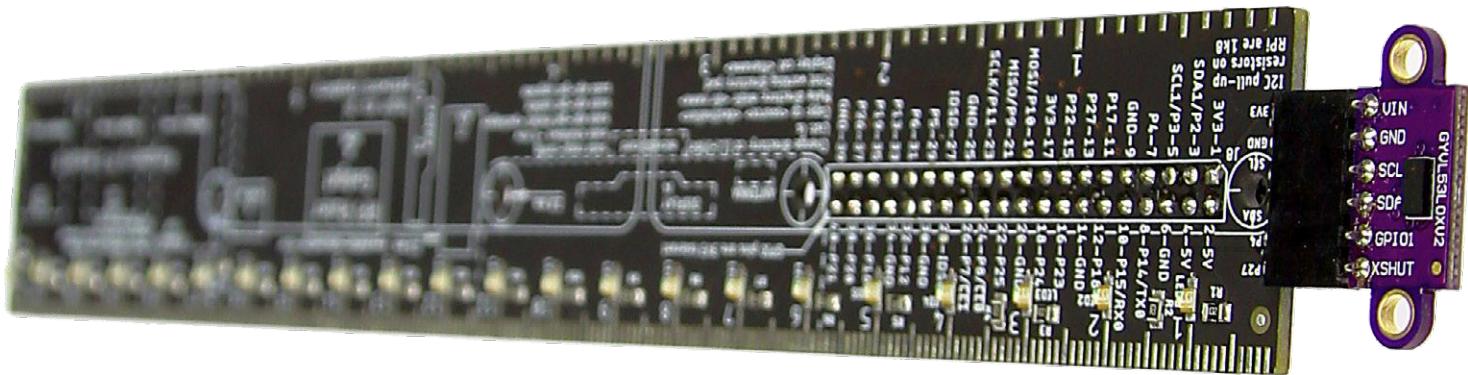
Figuur 2. Spanningsreferentie-eenheid uit een HP 3458 voltmeter.

Weblinks

- [1] LTZ1000-blokschema, bron afbeelding: www.analog.com/media/en/technical-documentation/data-sheets/1000afe.pdf
- [2] Interview met Carl Nelson: www.electronicdesign.com/analog/interview-analog-guru-carl-nelson
- [3] HP 3458 spanningsreferentie-eenheid, bron afbeelding: https://xdevs.com/doc/HP_Agilent_Keysight/3458A/img/a9_top.jpg
- [4] LTZ1000-schakelingen en -tests: <http://www.eevblog.com/forum/projects/project-kx-diy-calibrator-reference-sourcemeter/>

De Raspberry Pi-lineaal

pret met een time of flight-sensor



Clemens Valens (Elektor Labs)

Deze liniaal kan elektronisch afstanden meten tot twee meter met behulp van een VL53L0X time of flight-sensor (ToF). Hij werd ontworpen als weggeef-hebbedingetje voor beurzen en soortgelijke evenementen waar Elektor een stand heeft.

Specs in het kort

- Veel praktische mechanische details van de RPi
- VL53L0X Time of Flight-sensor
- 22 LED's
- I²C-connector
- liniaalprint gratis voor bezoekers van de Elektor-stand

praktische (mechanische en elektrische) informatie over de Raspberry Pi op en doet hij het prima als liniaal. Hij kan ook worden omgebouwd tot een echt high-tech gadget als je alle onderdelen monteert, een VL53L0X time of flight-sensor (ToF) aansluit op de I²C-connector en natuurlijk de liniaal op een RPI prikt.

De VL53L0X time of flight-sensor (ToF)

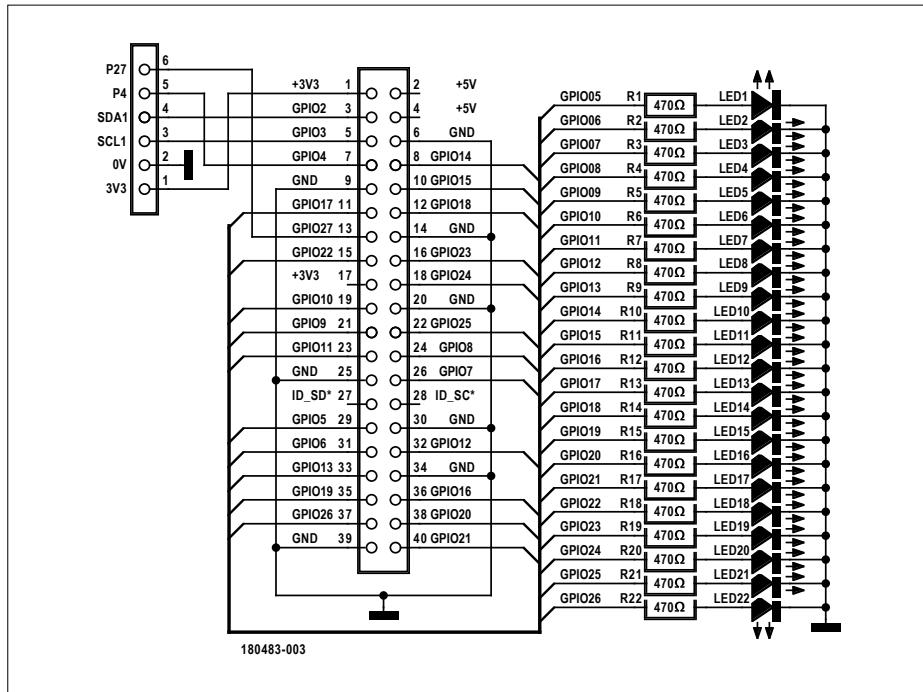
De VL53L0X-sensor die in dit project wordt gebruikt is best verbazingwekkend. De module met de laser-afstands-meter wordt gemaakt door ST, en is volgens de datasheet ondergebracht in de op dat moment (stand: 2018) kleinste behuizing. Hij meet nauwkeurig afstanden ongeacht de reflecterende eigenschappen van het doel, en kan absolute afstanden tot twee meter meten. De sensor schat de afstand tot een object niet door de hoeveelheid teruggekaatst licht te meten – die is immers sterk afhankelijk van de kleur en het oppervlak van het object – maar meet nauwkeurig de looptijd van het licht eraan het object en weer terug (vandaar de benaming ‘Time of Flight’).

Als je toevallig een van deze linialen hebt omdat je vorig jaar november de stand van Elektor hebt bezocht tijdens de electronica-beurs in München of de WoTS in Utrecht, zul je gemerkt hebben dat er nog plaats voor onderdelen (waaronder twee connectoren) op is. Dat komt omdat de liniaal kan worden voorzien van 22 LED's (in 0805 SMD-formaat) met serieweerstanden (0805-SMD's van 1 kΩ voldoen prima), een 40-polige Raspberry Pi-uitbreidingsheader (minimaal 16 mm hoog bij gebruik met een ‘normale’ RPi) en een 6-pens I²C-header/socket (of aansluitdraden, wat je maar wilt) – zie **figuur 1**.

Toegewezen, de RPi-liniaal per se is niet uitgesproken nuttig. Toch staat er veel

SPAD

Het gereflecteerde licht wordt gedetecteerd door een lawinediode die aan een enkel foton genoeg heeft (*single-photon avalanche diode, SPAD*). Deze diode is vergelijkbaar met een ‘normale’ lawine-fotodiode (*avalanche photodiode, APD*), behalve dat er maar één enkel foton nodig is om hem te laten doorslaan (*avalanche*) wat een stroom van enkele milliampères veroorzaakt. Wanneer de ‘lawine’ wordt onderbroken (*quenched*), kan de SPAD zich herstellen en een tweede foton detecteren. Het belangrijkste verschil tussen een SPAD en een APD is dat de laatste een lineaire versterker is voor optische signalen, de eerste is een triggercomponent. De VL53L0X komt niet verder dan twee meter en bij praktische toepassingen hangt de maximale afstand af van het omgevingslicht en het reflecterende oppervlak waar hij op wordt gericht. Binnenshuis is een afstand tot één meter goed haalbaar; buitenhuis kan het wat minder zijn. De nauwkeurigheid van de sensor is in de meeste gevallen beter dan 10%, maar nogmaals, dit is afhankelijk van de gebruiksomstandigheden.



Figuur 1: Het schema van de RPI-liniaal omvat slechts 22 LED's met serieweerstanden en twee connectoren.

De data die de VL53L0X produceert kan via een handige I²C-interface worden uitgelezen.

Handig BOB'je

De interessante module werd al snel opgemerkt door (Chinese?) fabrikanten van breakout boards (BOB's); handige uitvoeringen ervan zijn op internet gemakkelijk te vinden (**figuur 2**). De fabrikant heeft veel energie gestoken in het schrijven van software voor de sensor; samen met de breakout-boards heeft dat geresulteerd in verschillende open source-driverbibliotheeken voor Arduino en Raspberry Pi.

Elektronische RPi-liniaal

In dit project gaan we de LED's op onze RPi-liniaal gebruiken om de afstand weer te geven. Aangezien er slechts 22 LED's op de liniaal zitten, blijft de uitlezing beperkt tot 22 centimeter. Gelukkig kunnen er betere weergavemogelijkheden worden gecreëerd door de LED's te multiplexen. Daarom heb ik rode LED's gemonteerd voor de waarden 10 en 20, en groene LED's voor de rest. Merk op dat de afstand in millimeters ook op een display verschijnt dat aangesloten is op de HDMI-poort van de Pi. Verder kan dit project ook zonder de liniaal worden gebouwd; sluit de VL53L0X gewoon aan op de I²C-poort van de Pi.

De software werd ontwikkeld op een

Raspberry Pi 3 model B V1.2 maar zou ook moeten werken op andere Pi's met een 40-polige uitbreidingsconnector. Prik de liniaal eerst op de Pi voordat je hem inschakelt.

Installatie van de software op de RPi

De VL53L0X-driver is geschreven in C++, daarom is de software voor dit project ook in C++ geschreven. Er bestaan ook Python-bibliotheek voor de sensor, dus als u er de voorkeur aan geeft om met Python te werken, dan is dat geen probleem. Voor alle duidelijkheid: de volgende stappen worden op uw Raspberry Pi uitgevoerd, niet op een Windows/Linux/MacOS-computer:

- download en installeer wiringPi [1]. Deze bibliotheek wordt gebruikt om in C++ toegang te krijgen tot de GPIO-pennen van de Pi, zodat we de LEDs in en uit kunnen schakelen;
- download en pak de VL53L0X-driver STSW-IMG005 van ST uit, ga daarvoor naar 'Get Software' in [2];
- download en pak de Raspberry Pi-adapter VL53L0X_rasp van Casou (Mathieu Cassard) uit [3];
- download de Makefile van [4] en kopieer deze naar de map VL53L0X_rasp (overschrijf het bestaande bestand);

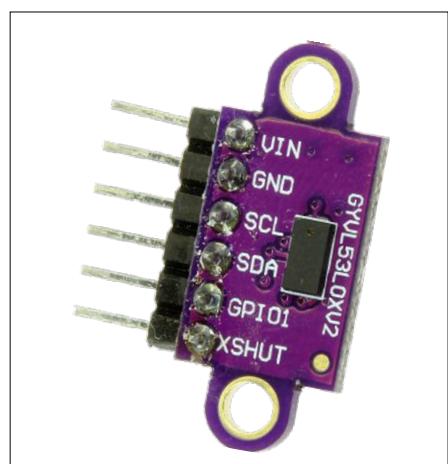


- download het bestand rpi_ruler.c van [4] en plaats het in de map VL53L0X_rasp/examples.

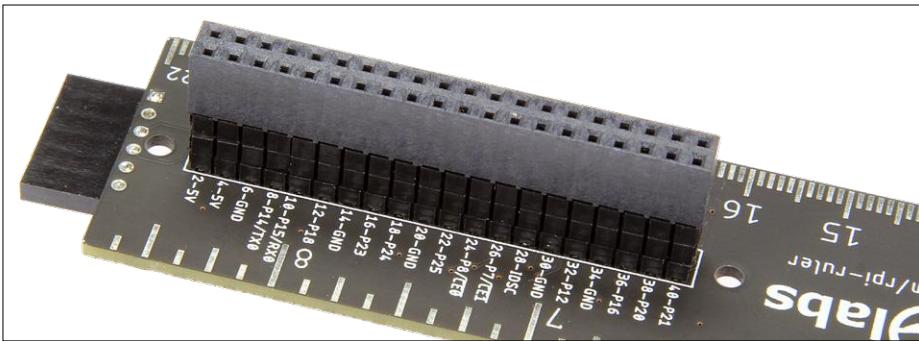
Als u de bovenstaande stappen hebt uitgevoerd en vanuit de home-directory werkt, voer dan nog het volgende uit:

```
cd VL53L0X_rasp
make API_DIR=~/VL53L0X_1.0.2/
make examples API_DIR=~/VL53L0X_1.0.2/
```

Er mogen geen fouten of waarschuwingen komen. Toets voor het uitvoeren van



Figuur 2: Een bijzonder populair breakout board (BoB) voor de VL53L0X Time of Flight-sensor.



Figuur 3: Als de liniaal op een 'normale' Raspberry Pi wordt aangesloten, dan moet de connector minstens 16 mm hoog zijn.

het RPI-liniaal-programma in:

```
./bin/rpi_ruler
```

Weergave op het display

Sluit een display aan op de RPi en de resultaten zouden voorbij moeten scrolen. Als u uw hand of een voorwerp voor de sensor houdt, moet dit een geloofwaardige afstand in millimeters opleveren. Objecten voor de sensor zullen nooit de waarde 8192 (d.w.z. buiten bereik) produceren. Als een object zich binnen 22 cm van de sensor bevindt, wordt de afstand aangegeven door de LEDs. Omdat in dat geval de afstand vaak een niet-gehele

centimeterwaarde is, bestaat de mogelijkheid dat twee LED's tegelijk oplichten. Hun helderheid wordt bepaald door de decimale waarde waarbij de LED die het dichtst bij de decimale waarde ligt, het helderst schijnt. Een voorbeeld: wanneer de gemeten afstand 12,4 cm is, zal LED12 een helderheid van 60% hebben en LED13 zal oplichten met 40% (inderdaad, voor optimale leesbaarheid zou een logaritmische schaal beter zijn dan een lineaire...). De helderheid wordt geregeld door de functie `softPwmWrite` in `wiringPi`.

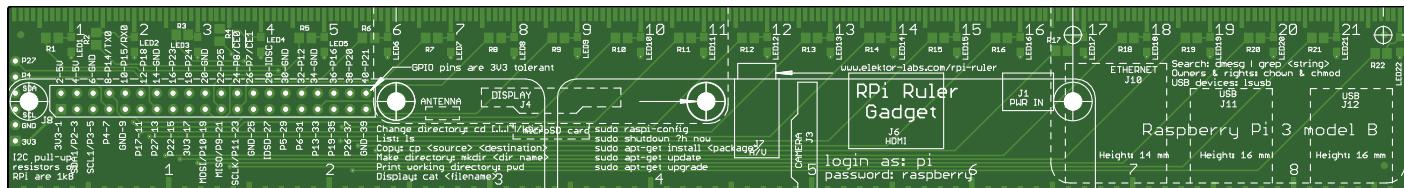
Opmerkingen

Lees het volgende aandachtig door

voor 'optimale resultaten en de beste gebruikerservaring':

- gebruik voor de Raspberry Pi-uitbreidingsconnector een exemplaar met een hoogte van tenminste 16 mm omdat de liniaal boven de USB-connector van de Pi uit moet komen (**figuur 3**). Deze opmerking heeft geen betrekking op de Raspberry Pi Zero omdat deze geen hoge connectoren heeft;
- de VL53L0X-sensor kan worden gevoed met 2,6...3,5 V. VL53L0X breakout modules hebben echter meestal een lage drop-out (LDO) spanningsregelaar om voeding vanuit 5 V mogelijk te maken. De I²C-connector op de liniaal levert slechts 3,3 V. Dat is geen probleem, omdat veel LDO's gewoon de ingangsspanning doorgeven wanneer deze onder de minimale waarde daalt. Als u om wat voor reden dan ook echt 5 V nodig hebt, gebruik dan gewoon de 5 V van de uitbreidingsconnector van de RPi.
- uw VL53L0X-sensor kan van een (vaak gelige) beschermfolie voorzien zijn. Als dat het geval is, verwijder die dan voordat u de sensor gebruikt. ▶

(180483-B-04)



ONDERDELENLIJST

Weerstanden:

R1...R22 = 1k, 0805

Halfgeleiders:

LED1...LED22 = groen, 0805

Diversen:

40-polige (2x20) busstrip, steek
0,1", hoogte 16 mm (bijv. Samtec
ESQ-120-33-T-D)
6-polige SIL-pinheader, steek 0,1"
6-polige SIL-busstrip, steek 0,1"
VL53L0X ToF-sensormodule

Print # 180483-1 (Elektor-shop)



IN DE STORE

→ 180483-1: kale print RPi-liniaal:
www.elektor.nl/elektor-raspberry-pi-ruler

→ 180483-71: kit met onderdelen (SKU18721):
www.elektor.nl/elektor-raspberry-pi-ruler-kit

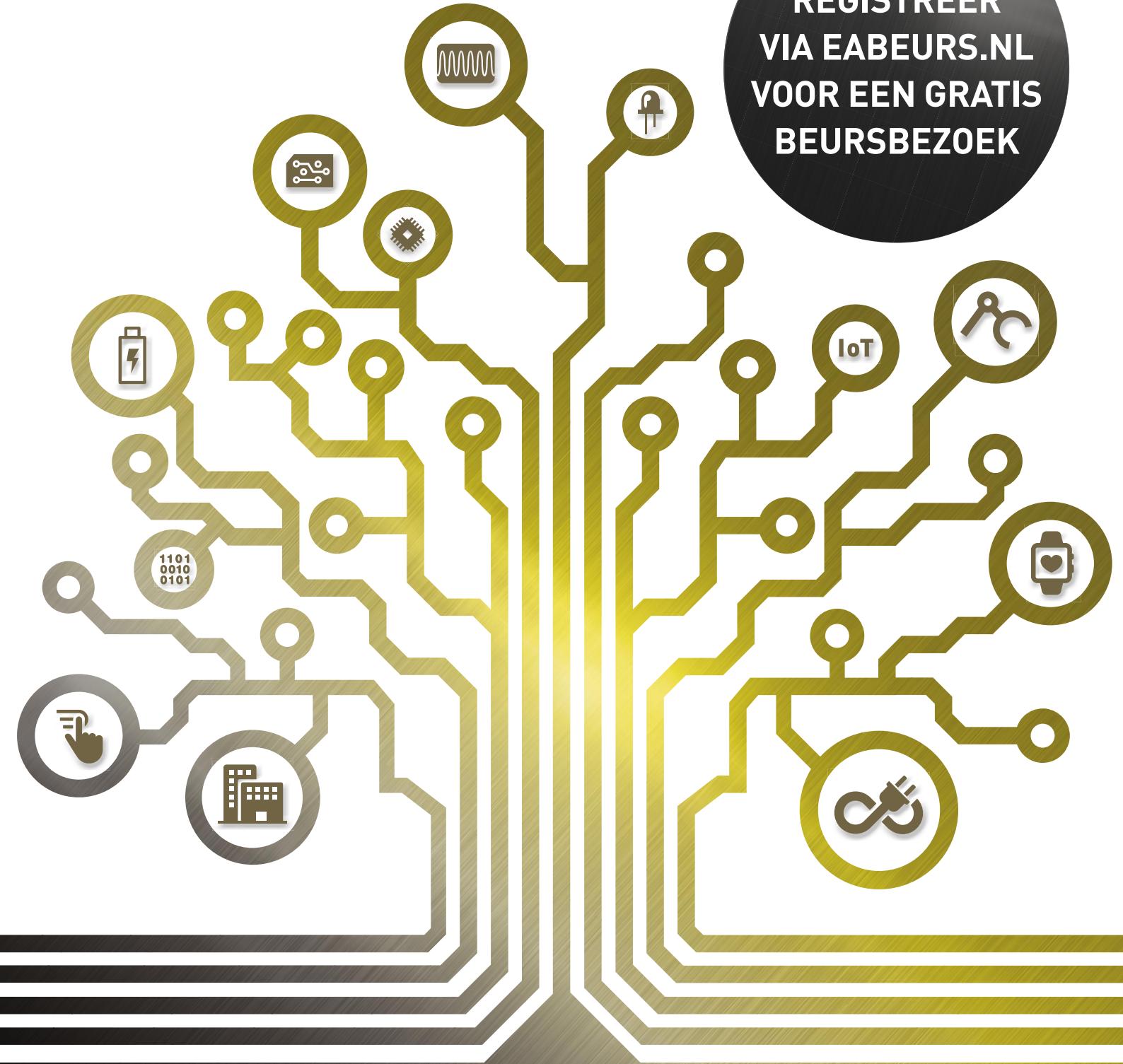
→ VL53L0X Time of Flight-sensor:
www.elektor.nl/gv-vl53l0xv2-time-of-flight-distance-sensor

Weblinks

- [1] wiringPi: <http://wiringpi.com/>
- [2] Officiële VL53L0X-driver:
www.st.com/en/embedded-software/stsw-img005.html
- [3] RPi VL53L0X-driver: https://github.com/cassou/VL53L0X_rasp
- [4] Makefile en hoofdprogramma: www.electormagazine.com/labs/rpi-ruler

ELECTRONICS & APPLICATIONS

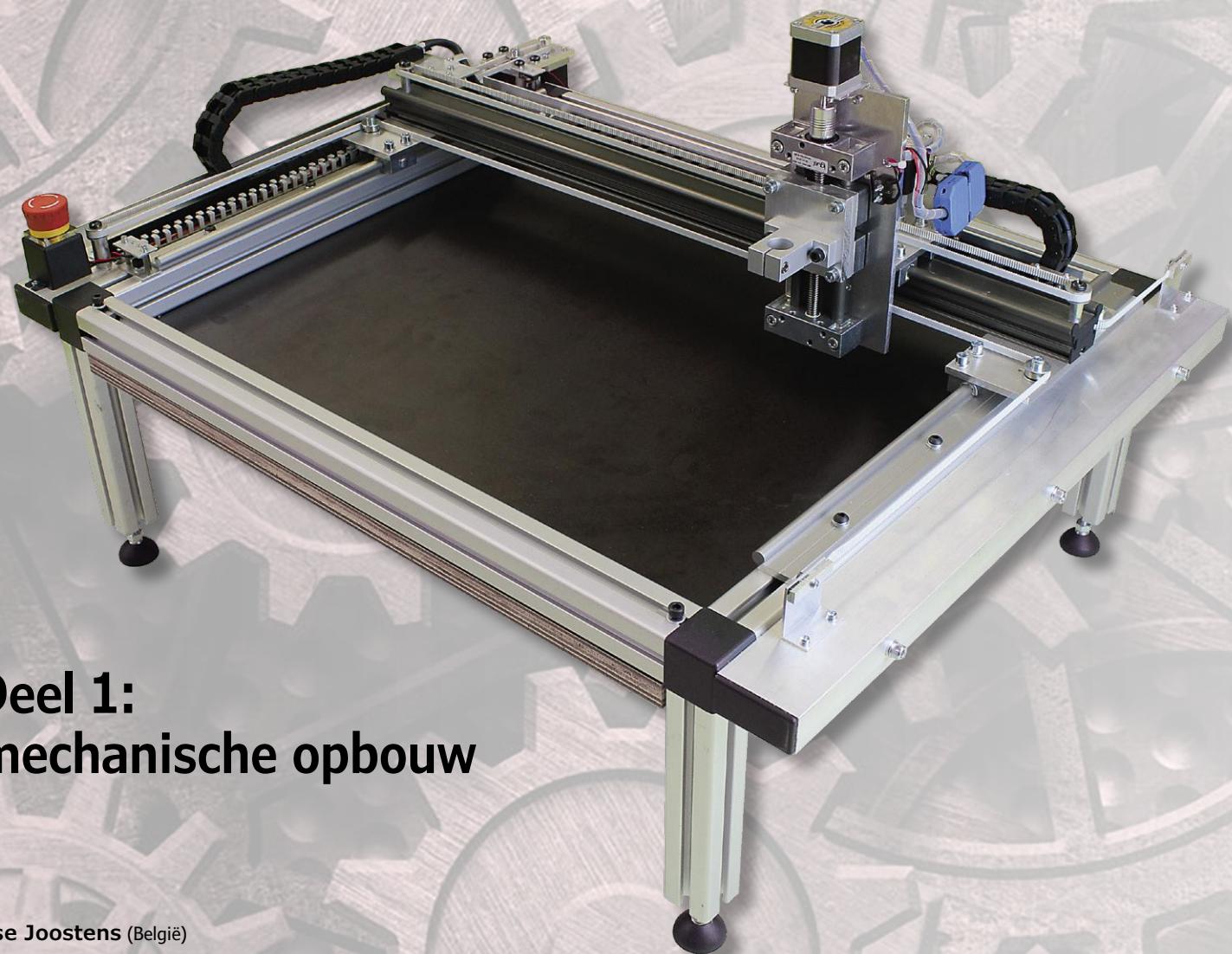
14 T/M 16 MEI 2019 JAARBEURS UTRECHT



REGISTREER
VIA EABEURS.NL
VOOR EEN GRATIS
BEURSBEZOEK

Wolverine CNC-portaalrobot

multifunctionele krachtpatser



Deel 1: mechanische opbouw

Ilse Joostens (België)

Heel wat hobbyisten, fanaten en *makers* worstelen met de mechanische aspecten van hun projecten. Een printje ontwerpen en opbouwen of een microcontroller programmeren is voor de doorgewinterde elektronicus doorgaans geen probleem, maar zodra de schakeling ingebouwd moet worden, is het enthousiasme gewoonlijk meteen een stuk bekoeld. En dan hebben we het nog niet eens over het zelf maken van relatief eenvoudige mechanische onderdelen zoals een tandwiel of een afdekplaatje. Met de opkomst van 3D-printers, betaalbare CNC-machines en online dienstverleners is mechanisch ontwerpen al een stuk toegankelijker geworden. Met de hier gepresenteerde CNC-portaalrobot willen we u verder op weg helpen binnen deze fascinerende wereld.

Eigenschappen

- werkvlak van ongeveer 50 × 30 cm
- lineaire geleidingen met kunststof glilagiers, ongevoelig voor vervuiling
- maximale snelheid op X- en Y-as: 15000 mm/min, Z-as: 750 mm/min
- herhalingsnauwkeurigheid: circa 0,2 mm
- nema-17 stappenumotoren
- aandrijving via versterkte GT2-tandriemen (X- en Y-as) en schroefspindel (Z-as uitbreiding)
- Z-as uitbreiding geschikt voor boren, frezen en graveren in hout, kunststoffen en non-ferro metalen (aluminium, messing). Met Proxxon LBS/E spindel (5.000 – 20.000 tpm, 100 W)
- professionele sturing en software van Eding CNC (cnc310 controllerbord)
- stappenumotordrivers op basis van Trinamic TMC2160 controller IC's voor een geruisloze werking
- 2.5D software uit te breiden naar full 3D
- multifunctionele opzet, uit te breiden voor andere toepassingen
- uit te breiden met een pendant (= console met knoppen en draaiwielen voor manuele bediening)
- maximaal stroomverbruik zonder boorslijper: circa 100 W
- totaalgewicht van de machine inclusief z-as uitbreiding en Proxxon LBS/E boorslijper: 24 kg

Indien u ooit zou overwegen om zelf mechanische onderdelen te ontwerpen, is de kans groot dat u daarbij in eerste instantie aan een 3D-printer denkt. 3D-printers bestaan al langer, maar dankzij de opkomst van betaalbare machines is deze technologie in korte tijd razend populair geworden. De meerderheid van deze machines gebruikt FDM-technologie (Fused Deposition Modeling), waarbij een plastic filament gesmolten en in laagjes geëxtrudeerd wordt. Ook machines op basis van stereolithografie (fotopolymerisatie) komen voor in het hobbylab.

Nadelen van 3D-printers

Ondanks de grote populariteit hebben 3D-printers wat nadelen. Niet alleen zijn de gebruikte materialen (filament, UV-hars, ...) relatief duur, het is ook een langzaam proces en daardoor niet echt geschikt voor de productie van grotere aantallen. Daarnaast laten vooral FDM-3D-printers een vrij ruw oppervlak achter, waardoor nabewerking soms nodig is. Het 3D-printen van voorwerpen die in contact komen met voedsel is volstrekt af te raden, omdat deze door het ruwe oppervlak na gebruik niet goed gereinigd kunnen worden.

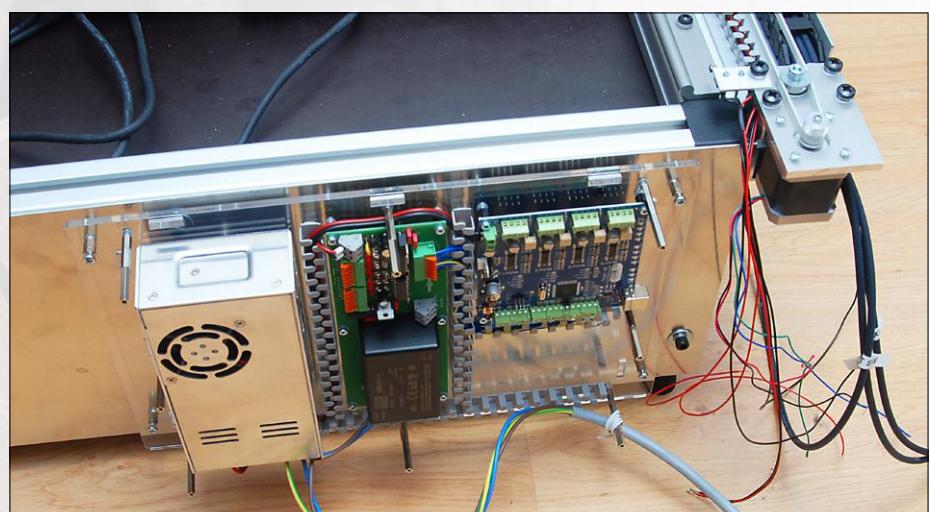
In vergelijking met 3D-printers en hun gesofisticeerde technologie zien eenvoudige CNC-machines er eerder uit als een primitieve kolomboor op steroïden. Toch zijn ze in staat om veel sneller onderdelen te produceren dan 3D-printers, met een betere afwerking en meer keuze qua materialen. Om u een idee te geven, een gemiddelde kleinere job op een CNC-machine duurt zo'n 5 minuten, terwijl u bij een 3D-printer eerder moet denken in

uren. Het is uiteraard niet onze bedoeling om 3D-printers te bekritiseren, voor ingewikkelde 3D-vormen of prototyping zijn ze nog altijd een goede keuze. De populariteit van 3D-printers heeft voor ons project een groot voordeel: het is tegenwoordig veel gemakkelijker om aan mechanische onderdelen als geleiders, lagers, tandriemen en koppelingen te komen dan pakweg 10 jaar geleden. Onze CNC-portaalrobot is dan ook opgebouwd met gebruik van dergelijke gemakkelijk verkrijgbare onderdelen die u eenvoudig online (bijvoorbeeld via eBay) kunt bestellen. In dit eerste deel gaan we dieper in op de mechanische opbouw van de machine.

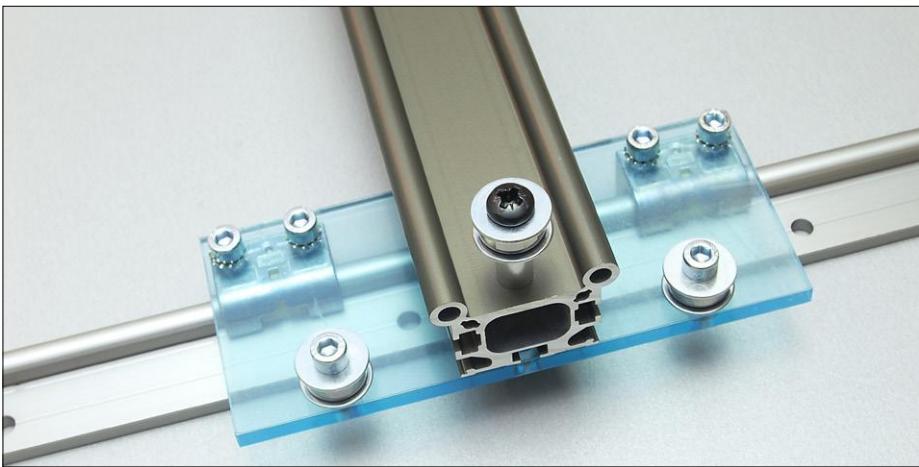
Frame en algemene opbouw

Vergelijken met standaard desktop-CNC-portaal freesmachines is onze portaalrobot een opvallende ver-

schijning. Waar portaal freesmachines gewoonlijk zijn opgebouwd rond een vast werkoppervlak, hebben wij voor een stevig aluminium frame gekozen. De 40 x 40 mm aluminium profielen dragen niet alleen bij aan de algemene stevigheid van de machine, maar vergroten ook de gebruiksmogelijkheden en bieden meer opties voor latere uitbreidingen. Het werkvlak van ons prototype bestaat uit 20 mm dik betonplex dat dankzij de frameconstructie gemakkelijk verwijderd kan worden. Niet alleen kan zo een werkvlak gekozen worden in functie van de toepassing, maar kan het werkvlak ook op een aangepaste hoogte gemonterd worden. Het is zelfs mogelijk om de machine zonder werkvlak over een werkstuk heen te zetten, bijvoorbeeld om een groter voorwerp te graveren. Een mogelijke toekomstige uitbreiding zou bijvoorbeeld een in hoogte verstelbare



Figuur 1. De elektronica vindt een prima plekje aan de achterkant van de machine.



Figuur 2. Voor de Y-as zetten we twee enkele rails type WS-10 in.



Figuur 3. Voor de X-as gebruiken we een dubbele rail van het type WSX-10-40.

tafel kunnen zijn voor CO₂-lasergraveren en -snijden of 3D-printen.

Indien de machine ingezet wordt als een pick & place machine voor het bestukken van printen met SMD's, kan het frame gebruikt worden voor het bevestigen van reels met componenten en zelfs SMD-feeders, mocht dat gewenst zijn. Dankzij het frame kunnen de elektronica voor de besturing en de voeding achteraan de machine gemonteerd worden (**figuur 1**). Hierdoor is de machine als geheel niet alleen compacter, maar is de elektronica ook beter beschermd tegen wegvliegende spanen en stof. Met de verstelbare voetjes kan de machine ook op een oneffen ondergrond stabiel geplaatst worden. Het is natuurlijk ook mogelijk om de machine zonder frame op te bouwen met een stuk betonplex als basis.

Ondanks de complexiteit is de machine redelijk nabouwvriendelijk. Hij is zonder specialistische gereedschappen en

machines op te bouwen. Enige kennis van metaalbewerking en (toegang tot) een kolomboormachine of boorstandaard, een afkortzaag met zaagblad voor aluminium, een draadtapset en een trapboor zijn echter noodzakelijk. Optioneel kunt u ook een werkvlak uit MDF-plaat met T-slot groeven maken (spoilboard) voor het inklemmen van werkstukken in combinatie met Proxxon spanklauwen (24257). Dit kan op de betonplex plaat bevestigd worden en eenvoudig vervangen worden bij beschadigingen. Hiervoor hebt u wel een bovenfrees met T-slot freesje nodig. De mechanische tekeningen, onderdelenlijst en een bouwhandleiding zijn als download beschikbaar via de Elektor-website [1].

X- en Y-as

• Geleidingen en lagerblokken

Voor de geleidingen van de X- en Y-as hebben we gekozen voor profielen en lagerblokken uit het DryLin-W-gamma

van het Duitse bedrijf Igus [2]. Deze hard geanodiseerde aluminium profielen en kunststof glijlagers bieden een zeer goede prijs-/prestatieverhouding en zijn robuust en weinig gevoelig voor stof en vervuiling. Bovendien kan de kunststof folie van de lagerblokken makkelijk vervangen worden, mocht dat ooit nodig zijn. Een bijkomend voordeel is dat de uitlijning wat minder nauw komt vergeleken met kogelomloop-leidingen wat de montage van de machine makkelijker maakt.

Voor de Y-as hebben we twee enkele rails, type WS-10, van 500 mm gebruikt, zie **figuur 2**, terwijl de X-traverse opgebouwd is rond een dubbele rail, type WSX-10-40, van 750 mm (**figuur 3**). Aangezien de X-traverse over een redelijke afstand zelfdragend dient te zijn, hebben we specifiek gekozen voor een verhoogde rail. Mocht u zelf een grotere machine willen bouwen, dan kan het ontwerp binnen zekere grenzen nog wat opgeschaald worden.

Voor de lagerblokken van de X-as hebt u, afhankelijk van de gewenste toepassingen voor de machine, de keuze uit het type WJUME-01-10 met instelbare speling, preload lagerblokken van het type WJ200UM-01-10-P40 / -P90 / -P140 of precisielagerblokken van het type WJUM-01-10K met JUMO-01-10K lagerfolie (voor de Z-as dient de lagerfolie JUMO-01-10K gebruikt te worden, omdat de huidige Z-as specifiek voor frezen is ontworpen).

De lagerblokken met instelbare speling en de preload lagerblokken zorgen, mits goed ingesteld, voor een soepele loop van de X-as, waarbij de aandrijfmotor weinig kracht hoeft te leveren. Dit is ideaal voor toepassingen waarbij weinig krachten op de X-traverse ontstaan, zoals lasergraveren/-snijden, 3D-printen, pick & place, ... Indien u de machine als CNC-freesmachine wilt gaan gebruiken, raden we de precisielagerblokken WJUM-01-10K aan. Met deze lagerblokken krijgt de X-as een betere rigiditeit (minder speling), wat voor de montage van een Z-as met freesspindel absoluut noodzakelijk is, zeker als u wat hardere materialen wilt bewerken.

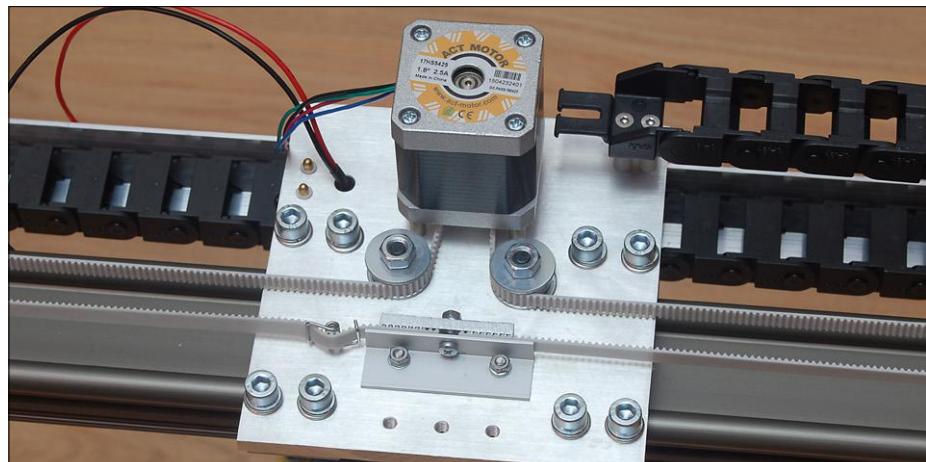
Een nadeel van deze lagerblokken is de stroeve loop van de X-as. Op zich hebben de motoren door de configuratie van de drijfriemen (katrolsystemen) voldoende kracht, ook bij wat hogere snelheden. Optioneel kunt u de lagerblokken voorzichtig smeren met een

kleine hoeveelheid wapenolie (type Ballistol GunCer, verkrijgbaar bij de betere wapenhandel).

De lagerblokken van de Y-as zijn een hybride type (WJRM-01-10) dat zowel over de rail rolt als glijdt, wat stabiever is bij hogere snelheden ($> 10 \text{ m/minuut}$).

• Aandrijfsysteem en motoren

Omdat de machine multifunctioneel van opzet is en we vooral voor pick & place en lasergraveren een snelle machine willen, wat met spindelaangedreven machines wat moeilijker ligt, hebben we met staaldraad versterkte 6mm-GT2-tandriemen gekozen voor de aandrijving van de X-en Y-as. Deze zijn gewoonlijk wit van



Figuur 4. Met een katrolsysteem wordt de kracht van de motoren vergroot.

Door een katrolsysteem wordt de kracht verdubbeld
Met staaldraad versterkte GT2-riemen zijn veel sterker dan neopreen

kleur en erg sterk vergeleken met de standaard zwarte neopreen tandriemen. Typisch voor riemen met GT2-profiel is eveneens de erg kleine speling. Het is daarbij wel belangrijk om de riemen vrij strak op te spannen.

GT2-riemen zijn erg makkelijk via eBay te vinden bij verkopers die materiaal voor zelfbouw 3D-printers aanbieden. Voor de geleidingen van de riemen zijn goed verkrijgbare 606 2RS-kogellagers gebruikt, gemonteerd tussen M6-sluitringen met een buitendiameter van 20 mm.

Voor de motoren hebben we standaard nema-17-stappenummotoren met 5 mm asdiameter gekozen. Door een katrolsysteem te gebruiken bij de tandriemen (zie **figuur 4**), wordt de kracht van de motoren verdubbeld, waardoor deze dan ook meer dan krachtig genoeg zijn voor licht freeswerk en alle andere doegeleinden waarvoor deze machine ontworpen is. Met een houdkoppel van 0,58 Nm voor de stappenummotoren (type 17HS5425) en een diameter van de poelies van 12,8 mm komen we theoretisch aan een maximale kracht van $(2 \times (0,58 \text{ Nm} \times 71\% / 0,0064 \text{ m}) =) 128,7 \text{ N}$ op de X- en Y-as bij lage snelheid. We zijn hierbij uitgegaan van een theoretisch dynamisch koppel van 71 % van het houdkoppel bij stuuring aan lage snelheid d.m.v. microstepping. In de praktijk sturen we de motoren niet met de maximale stroom aan en zijn er ook verliezen door de bekabeling, de mechanische overbrenging en

wrijving in de lineaire geleidingen, waardoor zo'n 60 tot 70 N realistischer is. Om u een idee te geven, wanneer we de X-traverse bij lage snelheid tegenhouden, is de machine in staat zichzelf over de grond te verschuiven.

Grottere motoren hebben als grootste nadeel meer gewicht en dus meer massa-traagheid bij snelle acceleraties en deceleraties. Bovendien zijn soms ook duurdere stappenmotordrivers die meer stroom kunnen leveren noodzakelijk. De Y-as is aangedreven met één enkele stappenummotor. Dergelijke assen worden nogal eens met twee stappenummotoren aangedreven. Dit gaat goed zolang beide motoren in de pas lopen. Mocht echter een motor stappen overslaan, kan de traverse schuin beginnen te lopen met alle problemen van dien.

Met de gebruikte GT2-poelies met 20 tanden is de verplaatsing van de X-as ongeveer 20 mm per omwenteling van de stappenummotor. Bij de Y-as speelt ook de dikte van de tandrem een rol en bij ons prototype kwamen we op een verplaatsing van 20,6 mm per omwenteling. Dit zijn richtwaarden en het is belangrijk om dat bij een opgebouwde machine met een schuifmaat te kalibreren.

De dragerplaten voor de kogel- en glijlagers en de motoren bestaan uit 6 mm dik 6082 T6 aluminium. Dit materiaal is nog relatief goed te bewerken en geeft de machine als geheel voldoende rigiditeit. Als de machine enkel voor lichte

toepassingen gebruikt wordt, kunt u ook 6 mm HPL-plaat gebruiken in de plaats van aluminium. Dat is een stuk makkelijker te bewerken.

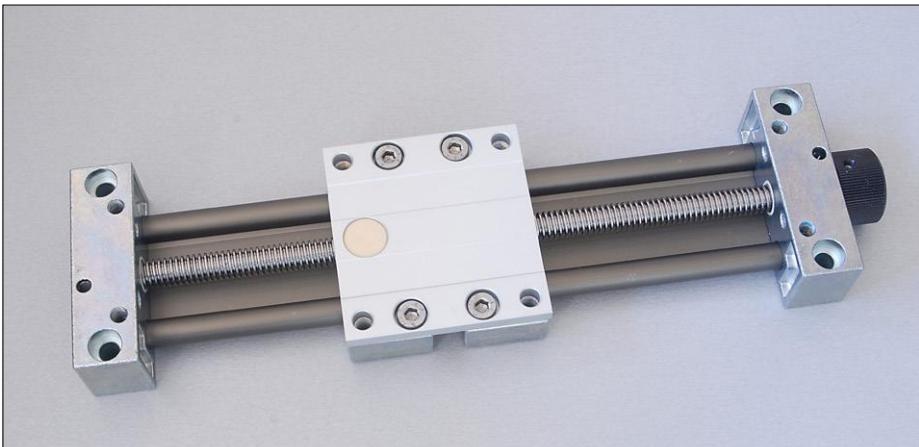
Z-as

De dragerplaat voor de X-as motor is voorzien van drie gaten met M5-schroefdraad voor de montage van diverse uitbreidingen. Onze eerste uitbreiding is een Z-as met freesspindel voor frezen, graveren en boren.

Omdat bij deze uitbreiding de precisie belangrijker is dan de snelheid, is hier een spindelaangedreven lineaire geleiding gebruikt. Deze kan als geheel bij Igus gekocht worden met bestelcode SLW-1040 (zie **figuur 5**). Daarbij is het mogelijk een aantal parameters zelf te kiezen, zoals de slaglengte. Hiervoor is 60 mm meestal voldoende. Het is belangrijk om de slaglengte niet te lang te kiezen in verband met de stabiliteit. Omdat de axiale belasting vooral naar beneden is gericht door het gewicht van de freesspindel, is een preload-moer op de spindelaandrijving in verband met speling niet nodig.

Voor de lagerblokken van de SLW-1040 dient u ook de nieuwe WJUM-01-10K kunststof folie te kiezen (speciale bestelling bij Igus), anders is er teveel radiale speling op de freesspindel.

De aandrijving wordt net als bij de andere assen verzorgd door een nema-17-stappenummotor via een flexibele koppeling.



Figuur 5. In de z-as zorgt een spindelaangedreven geleiding van het type SLW-1040 voor een hoge nauwkeurigheid.



Figuur 6. Deze Proxxon LBS/E-boorslijper is niet de enige geschikte kandidaat.

Theoretisch ligt de kracht met een verplaatsing van 2 mm per omwenteling hier een stuk hoger dan bij de X-as en Y-as, maar we dienen rekening te houden met de maximale axiale belastbaarheid van de SLW-1040 (700 N statisch) en de belasting van de Z-as als geheel op de motordragerplaat en glijlagers van de X-as. Het is daarom verstandig om dit in de praktijk beperkt te houden tot circa 20 N, wat voor de meeste toepassingen voldoende is. Een SK20-houder voor 20mm-stangen wordt gebruikt voor

de montage van de boor- en freesspindel. Voor ons prototype hebben we een Proxxon LBS/E-boorslijper gebruikt (**figuur 6**). Deze is geschikt voor het lichtere freeswerk. Maar merken als Kress leveren ook prima modellen. Bovendien belet niets u om al dan niet met behulp van een aangepaste houder een echte freesspindel te monteren.

De montage van de Z-as op de dragerplaat van de X-as gebeurt door middel van vier M5-schroeven en een stuk vierkante aluminiumstaaf. De Z-as kan daar-

bij op vijf vaste posities met verschillende hoogtes gemonteerd worden. Als u de Proxxon LBS/E gebruikt, kunt u die ook nog in hoogte verstellen, waardoor u over heel wat flexibiliteit beschikt.

Bedrading en eindschakelaars

De kabels die de motoren en eindschakelaars met de elektronica verbinden zijn constant in beweging met kleine buigradii en onderhevig aan vervuiling. Er worden dan ook hoge eisen aan gesteld waar gewone kabels niet aan voldoen. De toegepaste kabels zijn van het type ChainFlex (CF9.02.06, zie **figuur 7**) met 6 geleiders met 0,25 mm² doorsnede, wat voor nema-17-motoren en stromen tot 2,5 A nog net voldoende is. De koperdraden lopen bij dit type kabel spiraalvormig rond een plastic kern waardoor de buiging van de kabel als geheel minder overgedragen wordt naar de koperen geleiders. De plastic kern en kunststofvezels tussen de geleiders en buitenmantel zorgen voor de nodige versteviging en weerstand tegen trekkrachten. De buitenmantel weerstaat vervuiling door gangbare chemische producten, oliën en vetten. De fabrikant garandeert een foutloze werking gedurende minimaal 5 miljoen heen- en weergaande bewegingen bij temperaturen van -35 tot +100 °C. Het is af te raden om dikkere kabels met grotere koperdoorsnede te gebruiken. Deze zijn namelijk een stuk stijver bij het buigen. Om dezelfde reden is er gekozen om drie kabels met zes geleiders naast elkaar te gebruiken in plaats van één of twee dikkere kabels met meer geleiders. Om de kabels in positie te houden zijn twee kabelrupsen gebruikt, eveneens van Igus. De kabelrupsen zijn eigenlijk te lang voor de door de kabel afgelegde weg, maar dat spaart ons een extra kabelgoot of bedradingssokker. Naast kabels kan ook eventueel PVC-slang door de kabelgaten gevoerd worden voor bijvoorbeeld afzuiging, perslucht of koelmiddelen. Voer in elk geval niet het netsnoer van de Proxxon boorslijper door de kabelrupsen, want dit voldoet niet aan de eisen voor dergelijk gebruik!

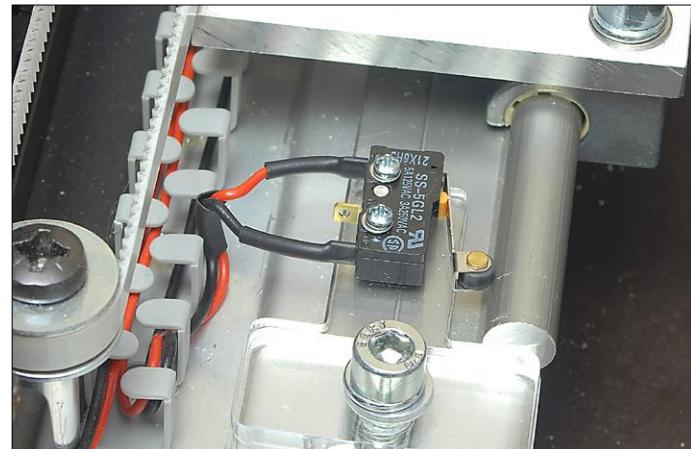
Voor de eindschakelaars zijn microschakelaars met rol gebruikt (**figuur 8**). Omwille van de veiligheid wordt het normaal gesloten contact gebruikt, omdat een contact verbreken zekerder is dan contact maken. Voor de X-as en Y-as zijn zowel aan het begin als aan het eind van de geleiderail eindschakelaars geplaatst. De schakelaars links en onderaan hebben

Weblinks

- [1] Webpagina bij dit artikel: www.elektormagazine.nl/180246-01
- [2] Igus: www.igus.com
- [3] Alu-messing-shop: www.alu-messing-shop.de
- [4] ACT Motor GmbH: <http://de.act-motor.com>
- [5] Proxxon: www.proxxon.com



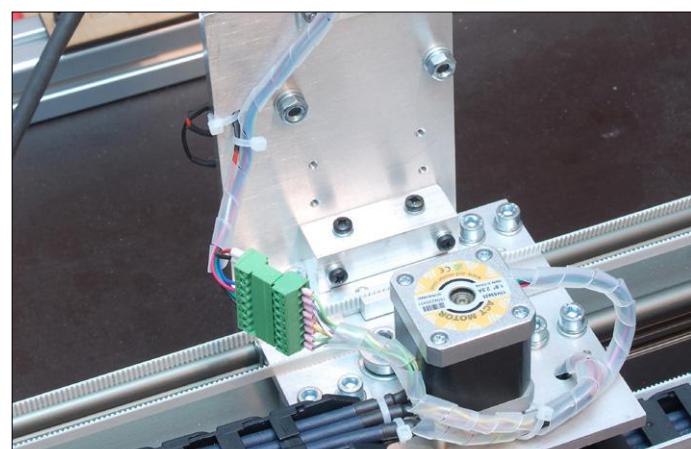
Figuur 7. Geschikte flexibele kabels weerstaan het duizenden malen buigen.



Figuur 8. De eindschakelaars met rol zet je het best op deze manier in.



Figuur 9. Een noodstop kan helpen voorkomen dat een verkeerde opdracht het freesje fataal wordt.



Figuur 10. Met een meerpolige connector is de z-as snel verwisseld voor een ander type.

zowel de functie van limietschakelaar als homing-schakelaar. De andere schakelaars hebben de functie van noodstop, mocht de machine door een foutconditie voorbij het fysieke einde van de X- of Y-as willen bewegen. Deze noodstopschakelaars zijn optioneel en hoeven niet noodzakelijk gemonteerd worden voor een goede werking van de machine. Ook vanuit de machinerichtlijn zijn deze schakelaars niet verplicht omdat de door de machine ontwikkelde krachten relatief beperkt blijven.

Voor de Z-as is slechts één limiet-/homing-schakelaar voorzien en wel voor de hoogste positie (Z_{max}). Een limietschakelaar voor de laagste positie is hierbij niet zinvol. Voordat die geactiveerd wordt, is de frees al ruimschoots door het materiaal heen.

Merk op dat alle homing-schakelaars in langsrichting gemonteerd staan ten opzichte van de assen. Doordat de schakelaars op die manier geen harde limiet

vormen kan het homen van de assen sneller gaan. Op het moment dat de homing schakelaar geactiveerd wordt, is er nog wat bewegingsruimte om de as af te remmen zonder risico op een harde botsing met de schakelaar.

Tot slot is naast een aansluiting voor een Z-probe ook een noodstop voorzien (**figuur 9**). Dit lijkt misschien overbodig, maar de extra kosten zijn al meteen uitgespaard als men dankzij de noodstop kan voorkomen dat een freesje breekt. De Z-as is via een meerpolige connector aangesloten op de rest van de machine (zie **figuur 10**). Op die manier kan de Z-as makkelijk losgemaakt worden en

omgewisseld worden met een andere uitbreiding, bijvoorbeeld een laserkop.

Elektronische sturing en software: deel 2

In het volgende artikel rond onze CNC-machine gaan we dieper in op de sturing, de stappenmotordrivers en de software.

We belichten dan ook G-code, CAM-software en de workflow om van CAD-tekening tot afgewerkt onderdeel te komen.



180246-01


IN DE STORE

→ Boek: 3D Printing and CNC Fabrication with SketchUp
www.elektor.nl/3d-printing-and-cnc-fabrication-with-sketchup



Opamps in de praktijk

deel 2: FET-opamps en breedbandtoepassingen

Burkhard Kainka (Duitsland)

Voor veel toepassingen is het van belang dat de ingangen van een opamp extreem hoogohmig zijn.

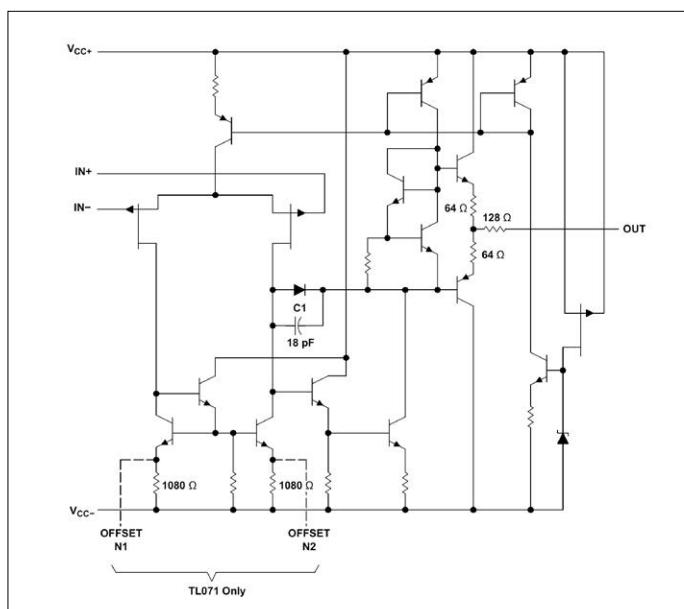
Aangezien dat bij bipolaire standaardtypes niet het geval is, gebruikt men in die gevallen opamps met FET-ingangen.

Bij een standaard-opamp zoals de LM358 lopen in de ingangen stromen van zo'n 30 nA. Over een ingangsweerstand van 1 M Ω treedt dan een spanningsval op van 30 mV, hetgeen voor veel toepassingen niet acceptabel is. In de ingangen van een opamp met FET-ingangstransistoren loopt daarentegen vrijwel geen stroom. Met JFET-ingangen worden ingangsstromen gerealiseerd die zo'n 1000 keer kleiner zijn dan bij bipolaire ingangen. Typische voorbeelden van JFET-opamps zijn de TL071 (enkelvoudig), de TL072 (tweevoudig) en de TL074 (viervoudig). Deze versterkers gebruiken JFET's in de ingangstrappen en bipolaire transistoren in de rest van de schakeling, zoals **figuur 1** laat zien. Een nog betere isolatie van de ingangen wordt bereikt met MOSFET's in de ingangstrappen. Een typisch voorbeeld van deze technologie is de CA3140 (**figuur 2**). Deze opamp werkt met een enkele of dubbele voedingsspanning vanaf 4 V. De ingangsspanning mag tot 0,5 V onder de negatieve voedingsspanning liggen. Omdat de ingangen van MOSFET's zeer gevoelig zijn voor overspanning, zijn ter bescherming zenerdiodes ingebouwd. De ingangsstromen zijn hier slechts 2 pA. De uit-

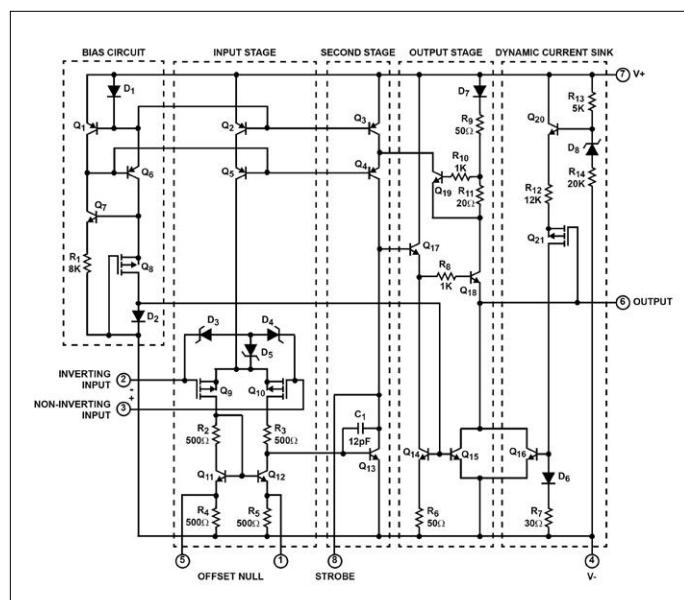
gangstrap van de CA3140 is conventioneel opgebouwd met bipolaire transistoren. De fabrikanten spreken in dit verband daarom ook van BiMOS-versterkers. Naast de enkelvoudige versie kennen we ook nog de dubbele opamp CA3240. Een verdere verbetering brengt de CA3160 met zijn complementaire MOS-uitgangstrap (CMOS). Hiermee wordt het mogelijk de uitgangsspanning uit te sturen tot 10 mV van de negatieve, resp. positieve voedingsspanning (rail-to-rail).

Zaagtandspanning

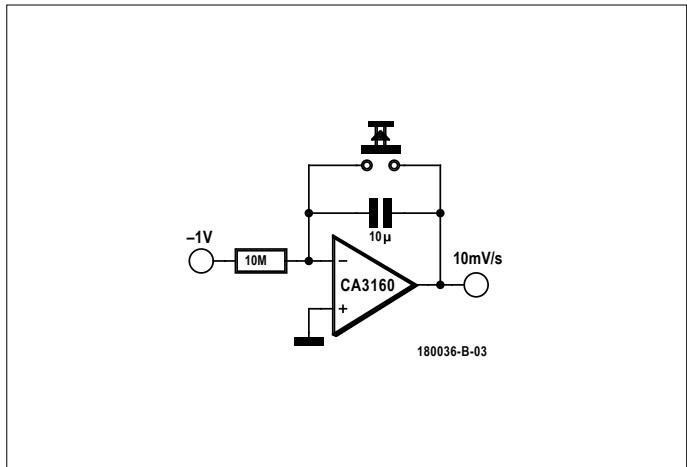
Een opamp met FET-ingang is heel goed bruikbaar om trage golfvormen zoals een zaagtandspanning op te wekken. **Figuur 3** toont een typische zaagtandgenerator, die bijvoorbeeld kan worden gebruikt voor het automatisch opnemen van karakteristieken. De opamp is geschakeld als integrator. De steilheid van de zaagtand wordt bepaald door de geringe laadstroom in de inverterende ingang. Door de hoge ingangsweerstand van de opamp hoeven we niet bang dat de ingangsstromen het resultaat beïnvloeden. Met een schakelaar parallel aan de



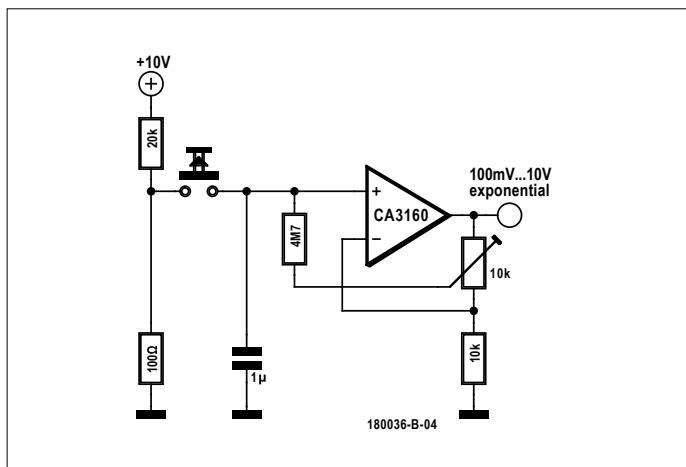
Figuur 1. Inwendig schema van de TL071 (bron: Texas Instruments).



Figuur 2. Inwendig schema van de CA3140 (bron: Renesas).



Figuur 3. Zaagtandgenerator.



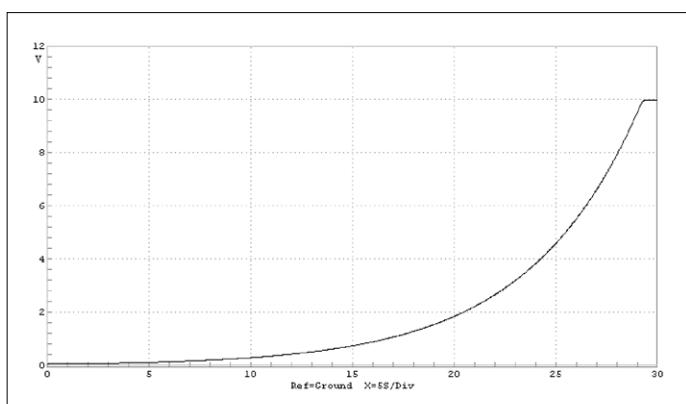
Figuur 4. Exponentiële zaagtandgenerator.

integrerende condensator kan deze worden ontladen en een nieuwe zaagtand worden gestart.

Soms hebben we in plaats van een lineair toenemende spanning een exponentieel verloop nodig. Een exponentieel werkende zaagtandgenerator zoals in **figuur 4** kan bijvoorbeeld worden gebruikt om een toongenerator aan te sturen voor audio-meetdoeleinden. De laadstroom van de condensator wordt hier opgewekt via een spanning die zelf evenredig is met de condensatorspanning. Hierdoor ontstaat een exponentiële toename (**figuur 5**). Om deze schakeling te laten opstarten is een zekere voorspanning noodzakelijk. Met 50 mV krijgen we vanwege de tweevoudige versterking een beginspanning van 100 mV aan de uitgang. Met een hoogohmige BiMOS-opamp kunnen de parameters van deze schakeling over een breed gebied worden ingesteld.

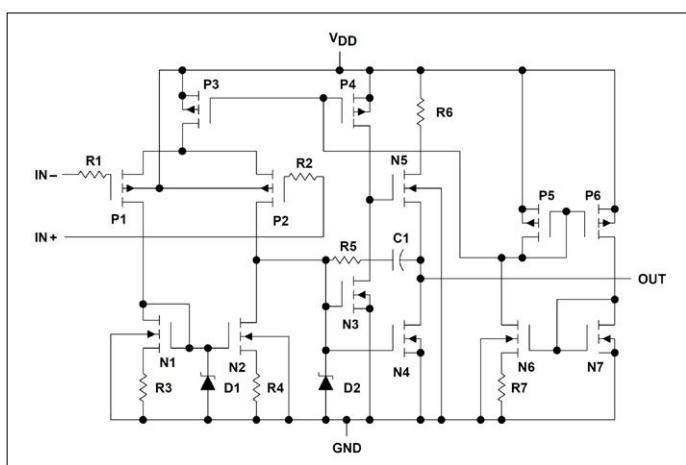
Veel moderne opamps zijn geheel in CMOS-technologie opgebouwd en bevatten geen bipolaire transistoren meer. Een typisch voorbeeld hiervan is de tweevoudige CMOS-opamp TLC272 met ingangsstromen van minder dan 1 pA! **figuur 6** toont het inwendige. Sommige CMOS-opamps kunnen aan zowel in- als uitgang met spanningen tot aan de voedingsspanning overweg. Doorlopend worden nieuwe opamps ontwikkeld met steeds betere eigenschappen. Deels betreft dat dan een nog hoogohmiger ingangsweerstand of een nog geringere ingangs-offsetspanning. Of er wordt geprobeerd echte rail-to-rail eigenschappen aan zowel in- als uitgang te bereiken en bovendien de mogelijkheid om relatief laagohmige belastingen aan te sturen. Zo wordt op sommige punten een bijna ideale opamp bereikt. Maar vaak betaalt men een bijzonder goede eigenschap met slechtere prestaties op andere punten. Zo hebben zeer hoogohmige CMOS-opamps meestal een grotere ingangs-offsetspanning en hebben exemplaren waarvan de offset is geoptimaliseerd een lagere kantelfrequentie.

Een universeel inzetbare viervoudige opamp van de nieuwe generatie is de TS914 met een rail-to-rail bereik aan zowel in- als uitgang. De inwendige schakeling (**figuur 7**) toont een complementaire verschiltrap aan de ingang en een push-pull uitgangstrap met MOSFET's. Dit type is in veel gevallen een goede keus en wordt bijvoorbeeld ook in de Elektor-SDR toegepast. Hij heeft een ingangsstroom van slechts 1 pA bij een offsetspanning van 5 mV, een vermogensbandbreedte van 800 kHz en kan laagohmige belastingen van minder dan 100 Ω aansturen.

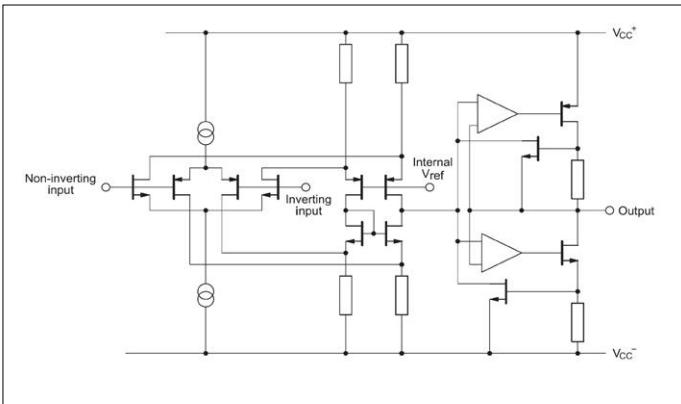


Figuur 5. De uitgangsspanning van een exponentiële zaagtandgenerator.

Extreem hoogohmige opamps worden met name toegepast in instrumentatieversterkers voor meetdoeleinden. Een instrumentatieversterker volgens **figuur 8** heeft differentiële ingangen en een unipolaire uitgang. Hiermee kunnen ook metingen worden verricht aan meetobjecten die met geen enkele aansluiting aan massa liggen. De spanning ten opzichte van massa moet wel binnen het in- en uitgangsstuurbereik van de opamps liggen.



Figuur 6. Inwendig schema van de TLC272 (Bron: Texas Instruments).

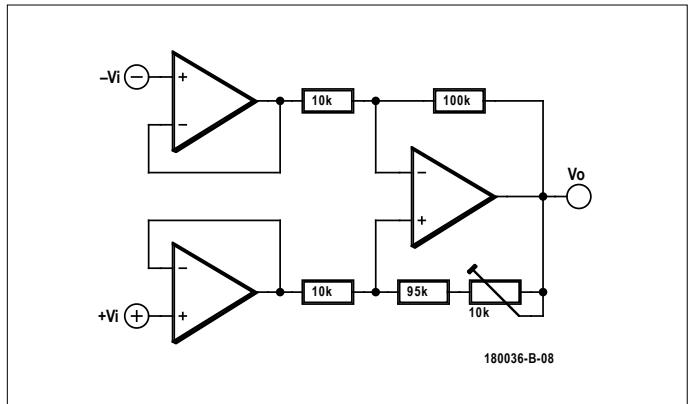


Figuur 7. Inwendig schema van de TS914 (bron: STMicroelectronics).

De schakeling bestaat uit een instelbare verschilversterker en twee voorgeschakelde impedantieomzetters om een extreem hoge ingangsweerstand te bereiken. De verschilversterker moet worden afgeregeld om de beste common-modeonderdrukking en stabiliteit te verkrijgen.

Breedbandtoepassingen

Hoewel een opamp eigenlijk als gelijkspanningsversterker is ontworpen, kunnen er ook wisselspanningen mee worden versterkt — zoals bij een microfoonversterker (**figuur 9**). Bij een enkelvoudige voedingsspanning wordt doorgaans een kunstmatige middenspanning gebruikt van bijvoorbeeld de halve voedingsspanning. De schakeling gedraagt zich dan alsof hij een positieve en een negatieve voedingsspanning heeft. Hoewel een dergelijke versterkerschakeling in een breed frequentiebereik kan werken, worden we in de praktijk toch vaak verrast hoe snel een opamp tegen zijn grenzen aanloopt. Als een versterker wordt ontworpen voor een versterking van 200 maal en een bandbreedte van 20 kHz, dan moet zijn versterkingsbandbreedte-product 2 MHz bedragen. De meeste typen halen echter hoogstens 1 MHz. Bij 20 kHz is dan geen sprake meer van voldoende versterking, waardoor de tegenkoppeling niet meer als gewenst functioneert. Daardoor ontstaat vervorming omdat interne trappen in verzadiging raken. Op een

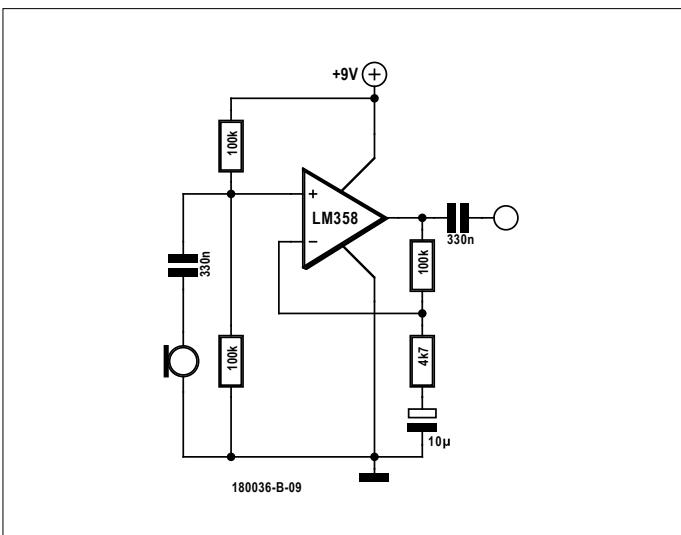


Figuur 8. Instrumentatieversterker.

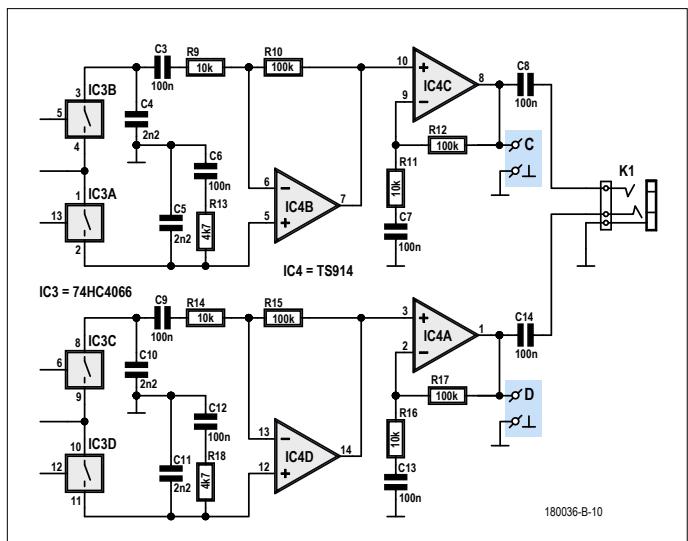
oscilloscoop is dat vaak herkenbaar aan het feit dat een sinus meer op een driehoek gaat lijken. Intern leidt oversturing tot blokgolfsignalen, die in de eindtrap door diens eindige stijgtijd tot driehoeken worden geïntegreerd.

Het Elektor SDR-shield bevat een viervoudige opamp TS914, die een versterkingsbandbreedte-product heeft van 0,8 MHz (**figuur 10**). Om een grotere bandbreedte te bereiken, wordt de versterking van 100 maal verdeeld over twee trappen met elk een versterking van 10 maal. Hierdoor wordt de bandbreedte 80 kHz, wat voor deze toepassing voldoende is. De signalen komen binnen op de stereo-ingang van een geluidskaart en worden daarna door de SDR-software verder bewerkt. Afhankelijk van de geluidskaart kan een bandbreedte van 24 of 48 kHz worden gebruikt, waaruit vanwege de over twee kanalen verdeelde bewerking een effectieve ontvangstbandbreedte van 48 of 96 kHz resulteert.

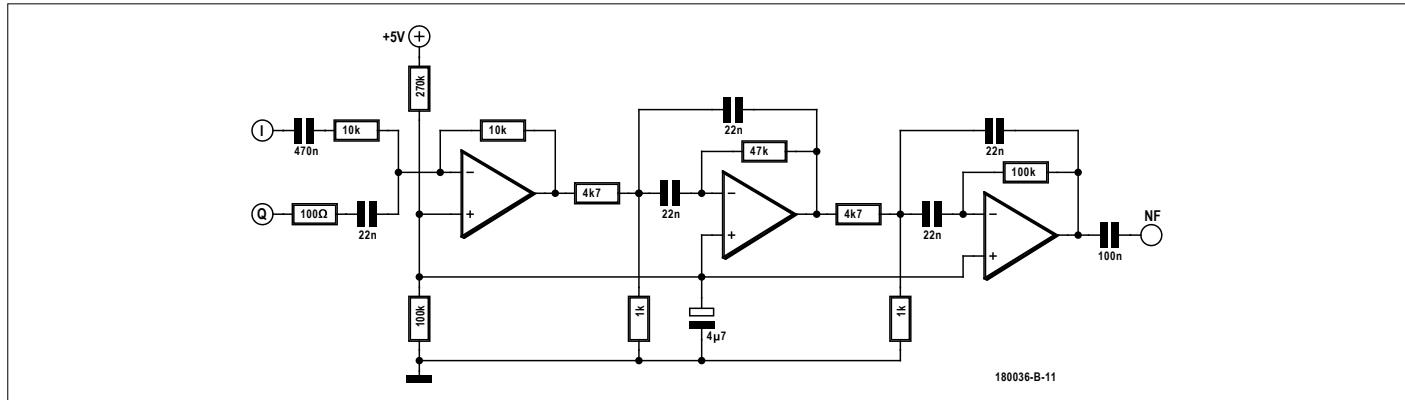
Voor stand-alone toepassingen van de ontvanger moet de signaalverwerking door de PC door een puur elektronische oplossing worden vervangen. Daarvoor gebruiken we een IQ-CW-filter van 750 Hz (**figuur 11**). Aan de ingang zien we een optelschakeling voor het I- en het Q-signalen, waarbij tussen beide een faseverschuiving van 90° bij 750 Hz wordt igerealiseerd. Hierna volgt een tweetraps banddoorlaatfilter voor ongeveer 750 Hz. Hiervoor worden tweevoudige opamps van het type LM358 gebruikt.



Figuur 9. Microfoonversterker met een opamp.



Figuur 10. De TS914 in het Elektor SDR-shield.



Figuur 11. Een IQ-CW-filter.

De filtertrappen zijn van tevoren met LTspice gesimuleerd en ontwikkeld. Maar let op: ook hier speelt de eindige bandbreedte van de opamps een rol! U zou natuurlijk kunnen denken: alles blijft binnen het LF-gebied, dus er zijn geen problemen te verwachten. De tweede filtertrap heeft een versterking van 100 maal. Voor een signaal van 800 Hz hebben we dus een versterkingsbandbreedte-product nodig van slechts 80 kHz. Er kunnen echter ook hogere frequenties in het signaal voorkomen, die niet tot vervorming mogen leiden. Nog kritischer wordt het, als je filters met een hogere Q voor hogere frequenties boven de 10 kHz wil bouwen. ▶

180036-B-03

IN DE STORE

→ Boek: The Circuit Designer's Companion (3rd Edition)
www.elektor.de/the-circuit-designers-companion-3rd-edition

→ Boek: Learning the Art of Electronics
www.elektor.de/learning-the-art-of-electronics-3rd-edition

→ Boek: Elektronik - gar nicht schwer 5
Experimente mit Operationsverstärkern
www.elektor.de/elektronik-gar-nicht-schwer-5

Advertentie

Word lid van de Elektor Community

Neem nu een GOLD lidmaatschap!

GOLD € 2,45/week

- ✓ Toegang tot ons web-archief
- ✓ 10% korting in onze webshop
- ✓ 6x ElektorLabs Magazine (Print)
- ✓ 6x ElektorLabs Magazine (PDF)
- ✓ Exclusieve aanbiedingen
- ✓ Toegang tot meer dan 1000 Gerberfiles
- ✓ Elektor's jaarlijkse DVD-ROM

GREEN € 1,78/week

- ✓ Toegang tot ons web-archief
- ✓ 10% korting in onze webshop
- ✓ 6x ElektorLabs Magazine (PDF)
- ✓ Exclusieve aanbiedingen
- ✓ Toegang tot meer dan 1000 Gerberfiles

Ook verkrijgbaar:

Het digitale GREEN lidmaatschap!

www.elektor.nl/member



HomeLab Helicopter

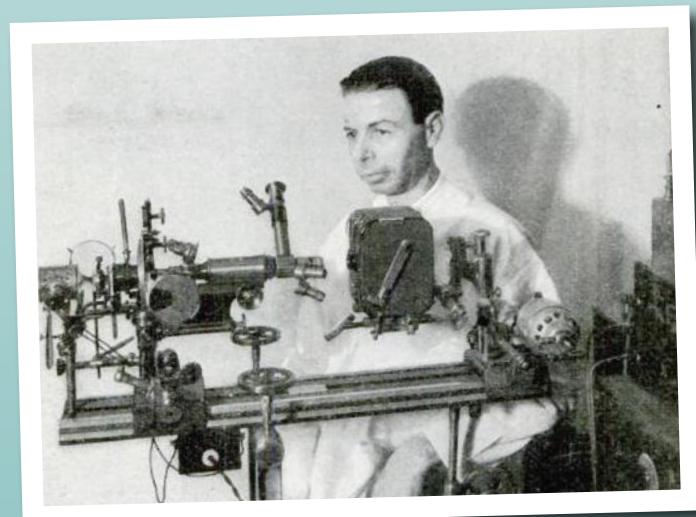
de wondere wereld der elektronica in vogelvlucht

samengesteld door **Clemens Valens** (Elektor Labs)



sympathische resonantie – “het lijkt wel tovenarij!”

Het internet staat vol met technologieën, producten en interessegroepen die er op wachten om te worden ontdekt. Soms struikel je bij het surfen op het web over iets dat je nog niet

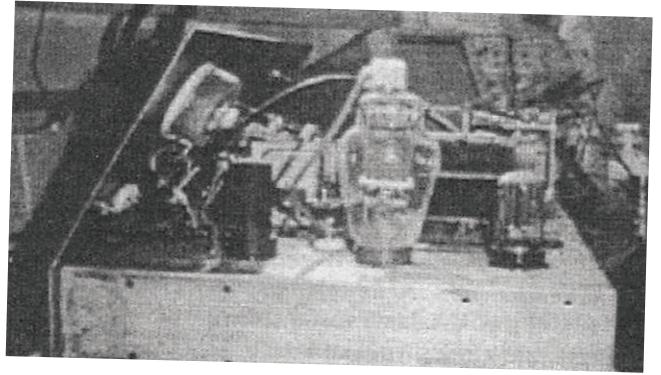


Royal Raymond Rife in 1931 met een van de microscopen die hij gebruikte om het effect van sympathische resonantie op ziektekliemen te bestuderen (bron: Wikipedia).

kent* en dat op de een of andere manier je aandacht trekt. Na het aanklikken van wat links vind je jezelf terug temidden van een groep mensen voor wie het leven om deze bijzondere technologie of dit product lijkt te draaien. Voor mij was de Rife Machine zo'n product. Ik had er nog nooit van gehoord en toch schijnt deze machine nogal wat aanhangers te hebben. Toch? Laten we voor dat we verder gaan vaststellen dat Rife Machines en de theorie erachter op z'n minst controversieel zijn. Bij het onderzoeken van deze technologie denk je al gauw aan pseudowetenschappen zoals magneettherapie en piramidekrachten. De Rife Machine werd in de jaren 1920 uitgevonden door Royal Raymond Rife (1888–1971). Het is een elektronisch apparaat bedoeld om gevaarlijke micro-organismen zoals bacteriën, virusen en kankercellen te vernietigen. Het idee erachter is dat ieder levend wezen een bepaalde resonantiefrequentie heeft. Als het organisme door die frequentie wordt gestimuleerd, zal het worden gedood of onschadelijk gemaakt. Royal Rife noemde deze frequentie het dodelijke trillingsgetal (Mortal Oscillatory Rate, MOR), anderen noemden het de Royal Rife-frequentie. Hoe rampzalig resonantie kan zijn, is te zien in de filmopname van het instorten van de Tacoma Narrows Bridge in 1940. Het werk van Rife werd een jaar of tien geleden herontdekt door muziekdocent Anthony Holland. U vindt zijn werk op YouTube.

* Er was ooit een website met de naam StumbleUpon, die speciaal hiervoor was bedoeld.

Een Rife Machine bevat een gewone functiegenerator die signalen opwekt met speciale, empirisch bepaalde, 'dodelijke' frequenties. Deze signalen worden vervolgens op een of andere manier overgebracht naar het organisme waarin zich de schadelijke ziektekiemen zouden bevinden die door middel van sympathische resonantie moeten worden vernietigd. Het effect is hetzelfde als bij een zanger die een wijnglas aan scherven zingt. Er zijn drie manieren om het signaal naar de ziektekiemen over te brengen: via direct contact, bijvoorbeeld met hand-elektrodes; draadloos met behulp van plasma-antennes en op afstand door het gebruik van non-lokale ruimte, quantumphysica en de antenne-eigenschappen van DNA.



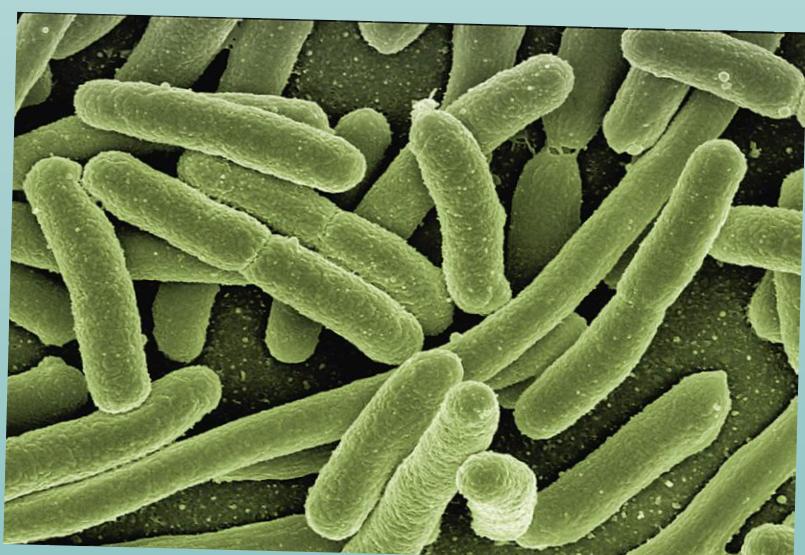
Een Rife Machine uit 1931 (bron: Wikipedia).

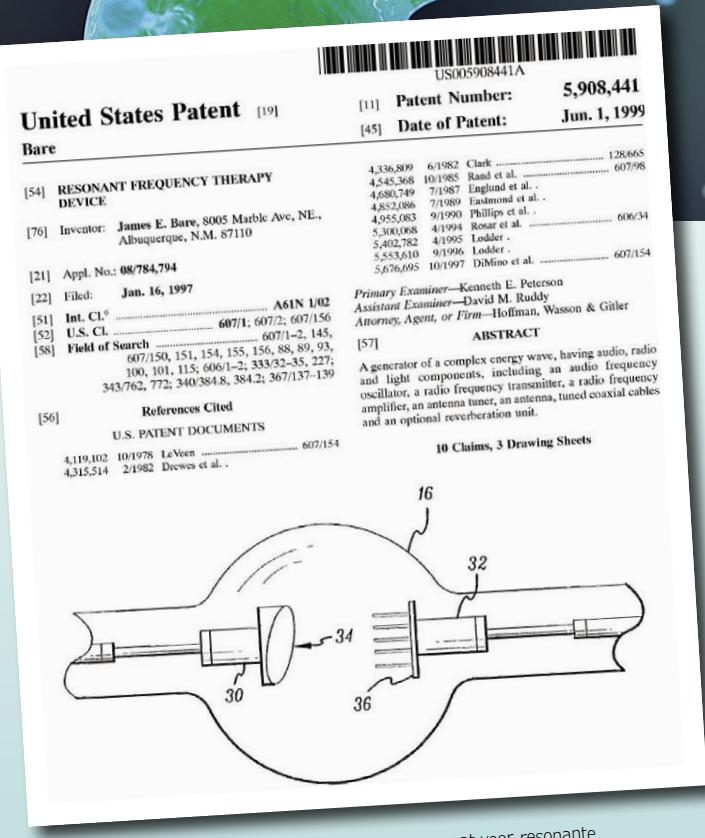


In een plasma-antenne zijn de gebruikelijke transmissielijn-elementen of antenne-oppervlakken vervangen door sterk geïoniseerd plasma, dat een goede geleider is. Omdat een plasma geen fysieke lengte heeft, kan een plasma-antenne eenvoudig worden afgestemd. Al in 1919 werd voor deze technologie patent verleend aan J. Hettinger. Meer recent verkreeg James E. Bare in 1999 het US patent #5,908,441 voor zijn op een plasma-antenne gebaseerde Resonant Frequency Therapy Device.

Het grote voordeel van de quantum-DNA-methode is dat de machines over afstanden van duizenden kilometers werken. Om de generator op de patiënt af te stemmen moet een stukje afgeknipte nagel in een doosje worden gelegd. Dit wordt met de generator verbonden die audiofrequente energie naar het DNA in het stukje nagel zendt. Op de website van de Spooky2 — een goedkope Rife Machine — staat: "Omdat DNA twee strengen heeft die spiraalvormig om elkaar heen zijn gewikkeld, werkt het als een antenne die de in energiesignalen gecodeerde informatie via non-lokale ruimte kan verzenden en ontvangen." Kennelijk is quantumverstrekking het mechanisme dat de energie via non-lokale ruimte overbrengt naar het bron-DNA,

dat zich alleen in de betreffende patiënt bevindt (het is dus een zeer nauwkeurige techniek). Verder lezen we op de Spooky2-website dat quantumverstrekking betekent "*dat als een deel van een systeem van het geheel wordt verwijderd, elke actie die op het deel wordt*





Eerste pagina van een patent uit 1999 voor een apparaat voor resonante frequentietherapie.

toegepast ook tegelijkertijd op het bronstelsel wordt toegepast, en vice versa." Hoewel dit klinkt als een oversimplificatie, worden vergelijkbare technieken al eeuwenlang gebruikt bij de zogenoemde sympathische magie. Beoefenaars van deze magie geloven dat een persoon die in contact komt met een voorwerp, een deel van zijn/haar energie op dit voorwerp overdraagt. Door zo'n voorwerp — een haarlok of een stukje nagel — in een (voodoo) pop te plaatsen en dan naalden in deze pop te steken, zal dit direct effect hebben op de persoon die er energetisch mee is verbonden. Het is duidelijk: quantumverstrekking is iets uit de oude doos.

En er zijn nog twee andere manieren om de dodelijke trillingen

naar het lichaam over te brengen: koude lasers of Low Level Laser Therapy (LLLT) met niet-thermische lichtfotonen — biofotonen volgens Fritz-Albert Popp — en Pulsed Electromagnetic Field Therapy (PEMFT). Zowel LLLT als PEMFT komen vaak voor in de alternatieve geneeskunde, en er lijken aanwijzingen te zijn dat ze een gering positief effect kunnen hebben. De Amerikaanse Food and Drug Administration (FDA) heeft zelfs sommige PEMFT-apparaten toegelaten die als 'wellness'-apparaten worden verkocht.

Omdat er geen wetenschappelijke of medische onderbouwing is voor het nut van Rice Machines op welk niveau dan ook, is de verkoop ervan in sommige landen verboden. Voor elektronici is dit geen probleem, want de meesten zullen de beschikking hebben over een functiegenerator. U hoeft alleen maar de uitgang van de generator op een veilige manier met een mens te verbinden. Een 'verbinding' op afstand is een goede optie: hierbij is er geen fysiek contact tussen de generator en de patiënt. Verbind de uitgang van de generator met een stukje afgeknipte nagel van de patiënt — zorg voor de juiste impedantie-aanpassing — en stel de frequentie in. Het vermogen is niet van belang want de DNA-helix-antenne is erg efficiënt. Pas alleen op dat u de nagel niet verbrandt. Sinds ik zo'n systeem in mijn lab heb, dat continu over een groot frequentiegebied zwaaait om veel ziektekiemen te bereiken, heb ik nooit meer de Spaanse griep gekregen.

Als dit artikel voldoende sympathische reacties genereert, zal ik in een volgende aflevering wellicht de Keely-motor in het zonnetje zetten, die is gebaseerd op 'vibrerende sympathie'.

Noot: ik aanvaard geen enkele verantwoordelijkheid voor schade of defecten ontstaan door toepassing van sympathische resonantie en/of magie op zowel levende wezens als levenloze voorwerpen.

**Luister naar
het geluid van Mars:
youtu.be/o3cxuIsEIRM**

Tips & trucs



Hoe kun je testen of een batterij vol of leeg is? Simpel, laat hem vanaf een hoogte van 20 cm op een hard plat oppervlak zoals een tafel vallen. Als de batterij terugstuitert is hij leeg, als hij gewoon omvalt is hij vol. Ongelooflijk, nietwaar? Toch werkt deze truc bij niet-oplaadbare alkalinebatterijen type AA of AAA. We hebben dit uitgebreid in ons lab uitgeprobeerd. De reden is dat de kleverige substantie in de batterij tijdens het ontladen hard wordt en hierdoor zijn schokabsorberende eigenschappen verliest: de batterij stuitert als hij een hard voorwerp raakt.

Zijn elektrische voertuigen te stil?



Veel mensen denken dat elektrische voertuigen (EV's) de oplossing zijn voor alle problemen, maar in de praktijk valt dat nogal tegen. Zonder duidelijk hoorbare verbrandingsmotor maken EV's niet genoeg geluid om voetgangers en fietsers te waarschuwen, en dat levert gevaarlijke situaties op. De ernst van dit probleem blijkt uit het feit dat in verschillende landen een onderzoek is gestart naar wetgeving die een *minimaal* geluidsniveau voor elektrische voertuigen voorschrijft. Voor het lokaliseren van EV's door slechtzienden, voetgangers en fietsers, is meer geluid nodig!

Gelukkig zijn er genoeg technieken om het probleem op te lossen: veel claxonneren zoals in India, een touw met 'pas getrouwde'-blikjes achter het voertuig... lawaai maken lukt altijd. Maar bij Elektor geven we de voorkeur aan elektronische

oplossingen, en wij gaan voor de application note AN-1552 van Analog Devices over het Electric Vehicle Warning Sound System (EVWSS). Een EVWSS "produceert een reeks geluiden, ontworpen om voetgangers op een EV te attenderen. Naast waarschuwingsignalen (zoals een claxon, maar dan minder indringend) zijn er geluiden die bij lage snelheden automatisch worden ingeschakeld. Deze geluiden variëren van kunstmatige klanken tot realistische nabootsing van motorgeluiden en het geluid van autobanden die over grind rollen". (Dachten ze al aan zwevende EV's zonder wielen toen ze dit schreven?) Dit doet me wel heel erg sterk denken aan de klassieker 'Monty Python and the Holy Grail' uit 1975 waarin koning Arthur's lakei (Patsy) twee halve kokosnoten tegen elkaar slaat om te suggereren dat de koning zijn koninklijke strijdros berijdt.

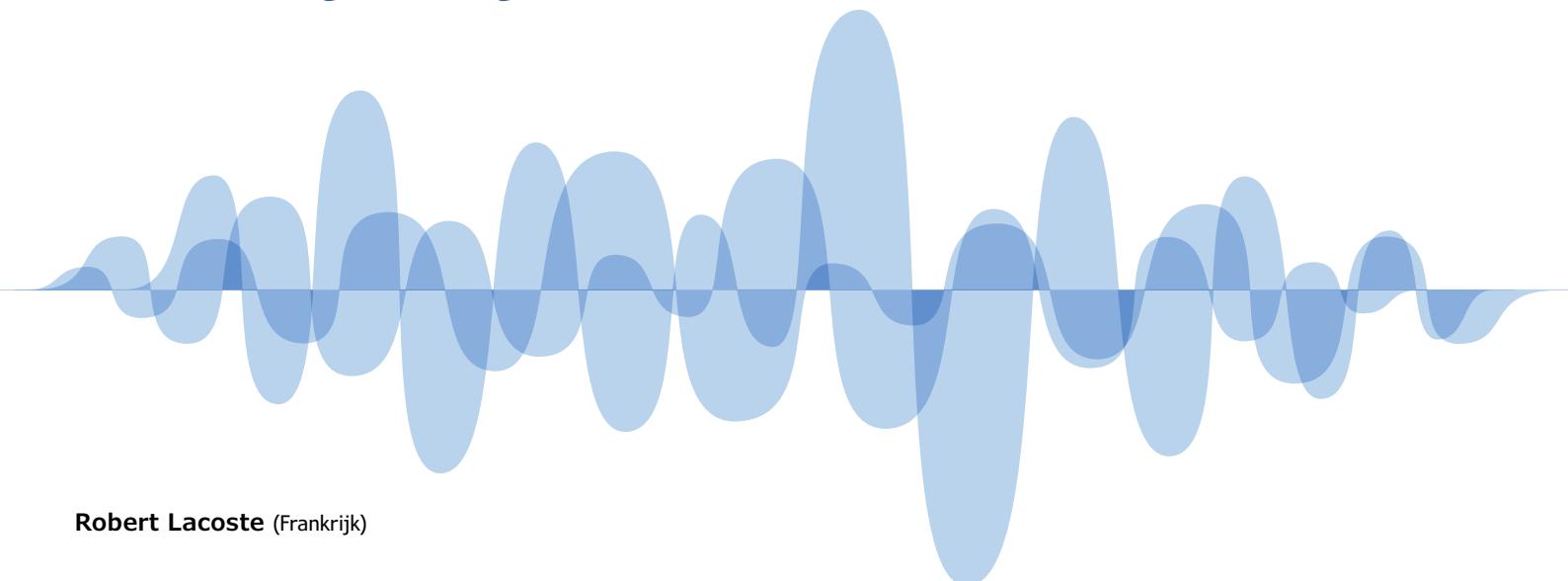
AD biedt twee EVWSS-oplossingen voor motorgeluiden binnen en buiten het voertuig, gebaseerd op de ADSP-BF706 BlackFin DSP. Voor instapmodellen is er de ADAU1450 SigmaDSP.

180567-B-03

Wilt u meedoen? Stuur uw commentaar, suggesties, tips en trucs naar labs@elektor.com

Digitale modulatietechnieken: de basis

GMSK, OQPSK, QAM, OFDM en co.



Robert Lacoste (Frankrijk)

In de jaren '80 van de vorige eeuw was ik apetrots op mijn Apple II en zijn snelle 6502-processor op 1 MHz. Om met de eerste BBS'en (bulletin board systems) te kunnen communiceren haalde ik mijn spaarpot leeg om een modem te kopen. Het uitwisselen van gegevens via een simpele telefoonlijn leek tovenarij! Het ging met een snelheid van 300 bits per seconde, en mijn ouders waren niet blij met de telefoonrekening, maar voor die tijd was het geweldig. Tegenwoordig gebruikt onze ADSL-box nog steeds zo'nzelfde koperverbinding, maar dan met een snelheid van enkele tientallen megabits per seconde (Mbps)... En niemand verbaast zich nog over de gigantische snelheden die met een eenvoudige 4G-smartphone worden gehaald.

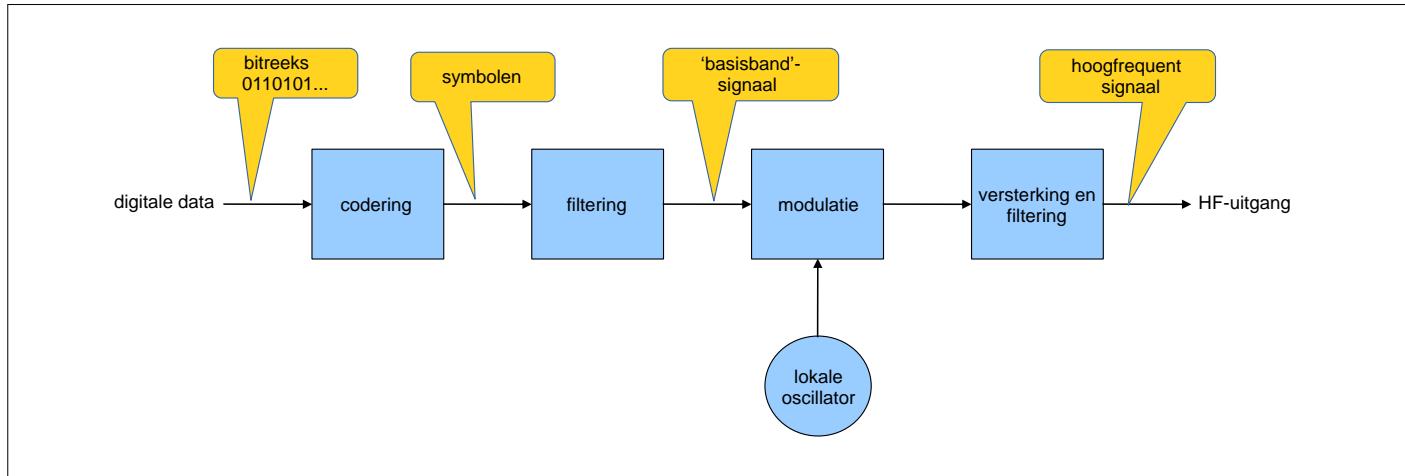
Waar komen deze explosieve ontwikkelingen vandaan? Voor een deel komen ze voort uit de wet van Moore en het gebruik van beter presterende onderdelen, maar de belangrijkste bijdrage is geleverd door wiskundigen en ingenieurs

die in signaalverwerking zijn gespecialiseerd. Dankzij deze mensen kunnen we communicatiekanalen op een veel efficiëntere manier benutten. Hoe? Door geavanceerde technieken voor codering, modulatie en demodulatie te gebruiken.

Bent u wel eens afkortingen tegengekomen zoals GMSK, OQPSK, QAM of OFDM? Weet u wat deze betekenen? Weet u hoe foutcorrectiesystemen werken? Zo niet, dan hebt u geluk, want we gaan ons hier de komende tijd in Elektorlabs Maga-

Tabel: Overzicht van modulatietechnieken

OOK	<i>On-Off Keying</i>	amplitudemodulatie (AM) met 0% of 100%
ASK	<i>Amplitude Shift Keying</i>	OOK met verschillende niveaus
FSK	<i>Frequency Shift Keying</i>	modulatie door frequentieververschuiving, verschillende niveaus mogelijk
MSK	<i>Minimal Shift Keying</i>	FSK waarbij de frequentieververschuiving gelijk is aan de helft van de data rate
BPSK	<i>Binary Phase Shift Keying</i>	fasemodulatie (0° en 180°)
DPSK	<i>Differential-PSK</i>	fasemodulatie waarbij de informatie is gecodeerd in het faseverschil tussen twee elkaar opvolgende signalen
QAM	<i>Quadrature Amplitude Modulation</i>	amplitudemodulatie van een draaggolf en amplitudemodulatie van een 90° in fase verschoven (kwadratuur-) versie van deze draaggolf



Figuur 1. In vrijwel ieder systeem voor datatransmissie wordt het bericht eerst gecodeerd en gefilterd, waarna er een draaggolf van een bepaalde frequentie mee wordt gemoduleerd die vervolgens versterkt en getransporteerd wordt.

zine mee bezighouden: dit is het eerste artikel van een mini-reeks over digitale modulatietechnieken!

Modulatie

Veronderstel dat u een draadloze data-zender wilt ontwerpen waarmee u een bericht dat bestaat uit een reeks enen en nullen met zo min mogelijk fouten naar iemand anders wilt sturen. Het mechanisme dat u hiervoor nodig hebt, zal in het algemeen uit vier stappen bestaan (**figuur 1**). Eerst gaat u de datareeks coderen. U voegt bijvoorbeeld aan het begin een paar synchronisatiebytes toe, die de ontvanger helpen om het begin van het bericht te herkennen. In dit stadium moeten ook mechanismen voor controle en/of foutcorrectie worden toegevoegd, op z'n minst een *checksum* om fouten te kunnen detecteren. Ook codeert u de data in een voor verzending geschikt formaat. De eenvoudigste techniek hiervoor, die NRZ wordt genoemd (*non return to zero*), bestaat uit... niets doen, dat wil zeggen een hoog niveau verzenden voor elke over te dragen '1' en een laag niveau voor elke '0'.

Deze methode heeft een paar nadelen, met name als het bericht lange reeksen nullen en enen bevat: de ontvanger kan dan uit synchronisatie raken. Er bestaan stabielere coderingsmethoden, zoals de Manchestercode waarbij de binaire waarden door stijgende en dalende flanken worden voorgesteld (hierbij wordt een één gecodeerd als '10' en een nul als '01'). Andere technieken, zoals foutcorrectie-algoritmes, kunnen de over-

dracht nog verbeteren. Onthoud voor dit moment dat zo'n codering nuttig kan zijn, ik kom hier in een volgend artikel op terug. Na de coderingsfase wordt het signaal gefilterd om de transmissie verder te optimaliseren. Ook die stap zal in een volgend artikel aan de orde komen. We gaan nu eerst kijken hoe we de nullen en enen op een draaggolf kunnen moduleren om ze vervolgens te verzenden. Deze stap vormt de kern van dit artikel, want hier bestaan veel verschillende technieken voor!

OOK

We beginnen met de eenvoudigste modulatietechniek: OOK (*On-Off Keying*). Dit is amplitudemodulatie (AM) met een modulatiediepte van 0% of 100% afhankelijk van de status van het te verzenden bit: tijdens een '1' wordt de draaggolf met maximaal vermogen uitgezonden, tijdens een '0' wordt er niets uitgezonden. Gezien de eenvoud wordt deze methode in veel goedkope producten toegepast. Zoals iedere vorm van amplitudemodulatie is OOK gevoelig voor storingen:

Listing 1. Scilab-script voor de berekening en weergave van het spectrum van een OOK-signaal.

```

// Geneneren van de draaggolf
fcarrier=1000000;
dt=1/(fcarrier*5);
npoints=128;
t=(0:npoints-1)*dt;
cw=sin(2*pi*fcarrier*t);

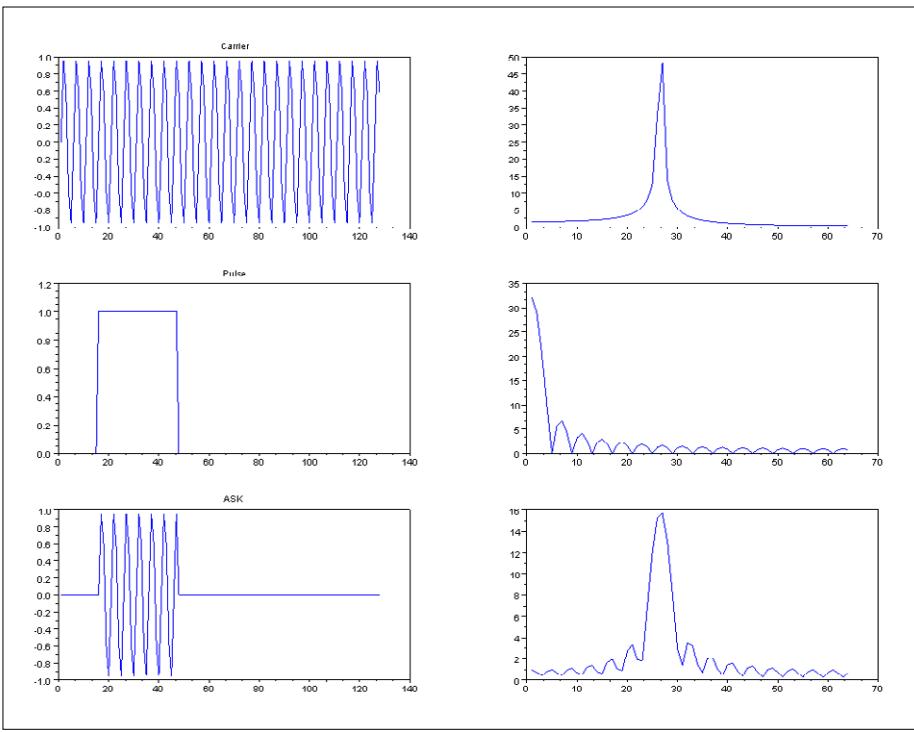
// Weergave van de draaggolf en zijn FFT
subplot(3,2,1); plot(cw); xtitle('Carrier');
spectrumc=abs(fft(cw)); subplot(3,2,2); plot(spectrumc(1:$/2));

// Genereren en weergeven van een rechthoekige puls en zijn FFT
pulse=zeros(1:npoints);
pulse(16:47)=1;
subplot(3,2,3); plot(pulse); xtitle('Pulse');
spectrump=abs(fft(pulse)); subplot(3,2,4); plot(spectrump(1:$/2));

// OOK-modulatie
ook=pulse.*cw;

// Weergave van het resultaat en de bijbehorende FFT
subplot(3,2,5); plot(ook); xtitle('OOK');
spectruma=abs(fft(ook)); subplot(3,2,6); plot(spectruma(1:$/2));

```



Figuur 2. In deze Scilab-simulatie staan de signalen in het tijddomein links en vindt u rechts hun frequentiespectra. Het spectrum van een rechthoekige puls is een curve die wordt gedefinieerd door $\sin(x)/x$, en dat zien we terug in het spectrum van een OOK-puls.

een stoorsignaal kan als een '1' worden gezien. Een ander probleem is dat voor een OOK-zender een radiokanaal nodig is waarvan de breedte afhangt van de

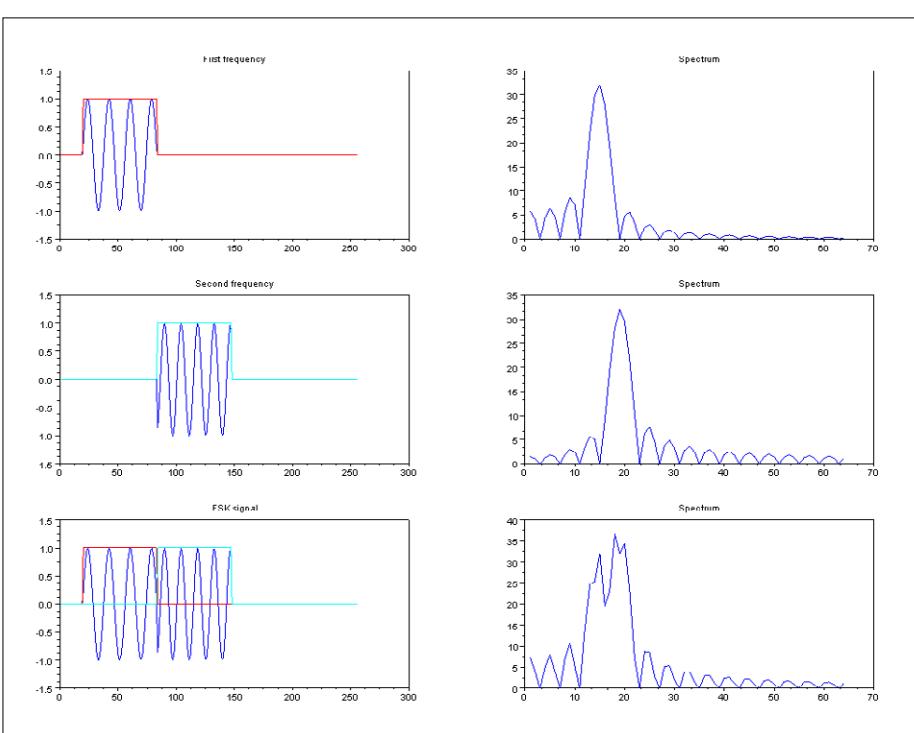
gegevensnelheid (*data rate*) van de verbinding. OOK kan dus niet worden gebruikt voor erg snelle verbindingen, want het kanaal wordt dan breder dan

is toegestaan. Hoe dat komt? Hiervoor moeten we het bijbehorende spectrum bekijken, dat wil zeggen overgaan naar het frequentiedomein. Je kunt dit zichtbaar maken door een spectrumanalyzer op de uitgang van een zender aan te sluiten, maar ik houd het hier bij een digitale simulatie (bestanden kunnen worden gedownload van [1]). Het frequentiespectrum van een signaal kan eenvoudig met een Fouriertransformatie worden berekend. We gebruiken hiervoor Scilab, een gratis digitaal open source-rekenkool. Met een klein script (**listing 1**) kunt u het frequentiespectrum van een met OOK gemoduleerde '1' berekenen en weergeven.

Het resultaat van deze simulatie ziet u in **figuur 2**. Het frequentiespectrum van een OOK-puls bevat uiteraard de draaggolf frequentie, maar ook talloze pieken die zich op regelmatige afstanden van deze draaggolf bevinden... Kijk nog eens goed naar figuur 2. Een OOK-signaal is in feite product van de draaggolf en een rechthoekige puls met een lengte van één bit. In het frequentiedomein is het spectrum van de draaggolf één enkele verticale piek. Als u mijn artikel over CIC-filters [2] hebt gelezen, dan herinnert u zich misschien dat het spectrum van een rechthoekige puls een curve is, die wiskundig is gedefinieerd als $\sin(x)/x$. Deze curve heeft een hoofdpiel, en daarnaast een oneindig aantal nevenpieken met afnemende amplitude. De eerste ligt 13 dB onder de hoofdpiel en dat is tamelijk hoog. De frequentieafstand tussen de pieken is omgekeerd evenredig met de tijdsduur van het bit (dus als de bitrate toeneemt, wordt het spectrum breder...). Ook hebben de wiskundigen ons geleerd dat het spectrum van het product van twee signalen (hier de draaggolf en de rechthoekige puls) overeenkomt met de convolutie van hun individuele spectra. Dat is hier het spectrum van de rechthoekige puls [$\sin(x)/x$], maar dan zodanig verschoven dat het precies rond de draaggolf frequentie ligt (zie figuur 2).

Meer niveaus

Van OOK gaan we naar een modulatietechniek die hieraan verwant is: ASK of *Amplitude Shift Keying*. Het is een OOK-variant waarbij het HF-vermogen niet nul is bij het verzenden van een '0'. Er kan bijvoorbeeld 10% van het vermogen voor de 'nullen' worden gebruikt en 100% voor de 'enen'. Zonder de filtering waar ik het in een volgend artikel



Figuur 3. Het spectrum van een FSK-signaal is de som van de spectra van twee OOK-signalen, het ene gesitueerd rond $f - df/2$ en het andere rond $f + df/2$. Hier is het frequentieverschil zodanig gekozen dat de piek van het ene signaal precies in het dal van het andere valt, waardoor de prestaties worden verbeterd.

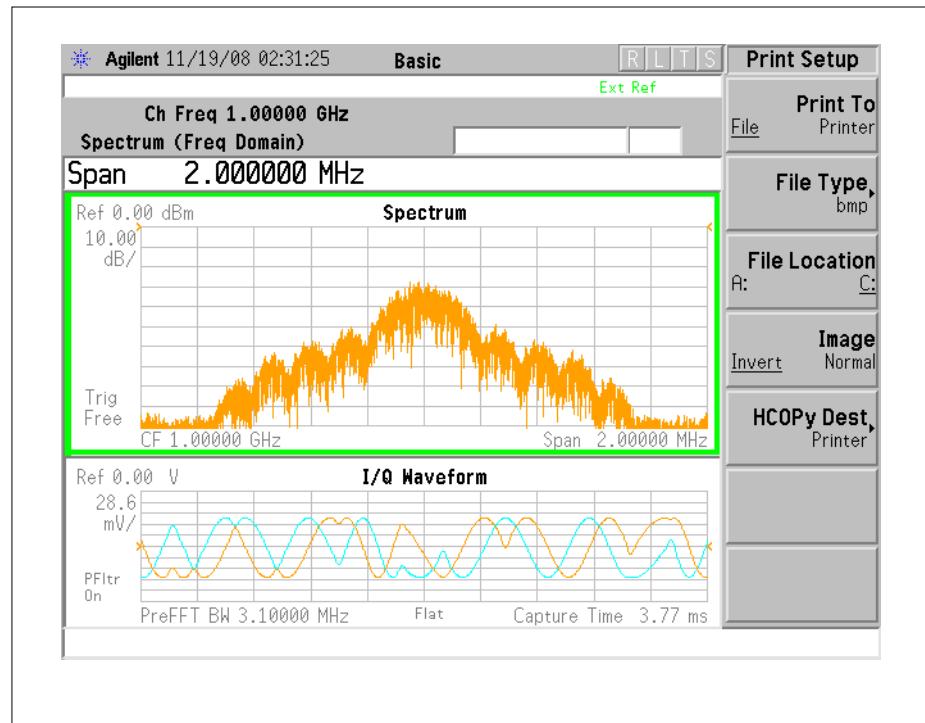
over zal hebben biedt deze methode niet veel voordeel, maar dit ASK-voorbeeld geeft mij de gelegenheid om het begrip *multi-level modulatie* te introduceren. Hierbij wordt de transmissiesnelheid verhoogd door meer dan twee verschillende vermogensniveaus te gebruiken. Zo wordt bij 4-ASK gebruikgemaakt van vier verschillende HF-niveaus, bijvoorbeeld 10%, 40%, 70% en 100%. Dit maakt het mogelijk om twee bits tegelijk te verzenden en zo de snelheid van de verbinding te verdubbelen. Hierbij worden de te verzenden bits twee aan twee samengevoegd in groepjes die symbolen worden genoemd. Ieder symbool (bitpaar) wordt nu aan een bepaald vermogensniveau gekoppeld. Bijvoorbeeld '00'=10%, '01'=40%, '10'=70% en '11'=100%. Een symbool van twee bits wordt dus in één keer verzonden. Natuurlijk kunnen we ook nog verder gaan en acht niveaus gebruiken voor symbolen van drie bits tegelijk (8-ASK), maar dat vergroot de kans op fouten want de verschillende symbolen zijn dan voor de ontvanger lastiger van elkaar te onderscheiden.

FSK en consorten

We gaan een stap verder. U weet waarschijnlijk dat frequentiemodulatie minder storingsgevoelig is dan amplitudemodulatie. In plaats van OOK of ASK te gebruiken, zijn de meeste transmissiesystemen dan ook gebaseerd op een modulatietechniek die FSK wordt genoemd (*Frequency Shift Keying*). Het principe is dat de draaggolf frequentie wordt verlaagd om een '0' aan te geven en wordt verhoogd voor een '1'. Als we het frequentieverschil tussen de enen en de nullen aanduiden met 'df', dan zijn de twee uitgezonden frequenties dus afwisselend $f_c - df/2$ en $f_c + df/2$, met f_c als centrale frequentie.

Wat gebeurt er in het frequentiedomein? Stelt u zich voor dat u in FSK een enkele '0' uitzendt, gevolgd door een enkele '1'. De '0' is gelijk aan een rechthoekige puls die een draaggolf met een frequentie van $f_c - df/2$ moduleert. Het spectrum heeft dus zoals bij OOK de vorm $\sin(x)/x$, maar nu gepositioneerd rond $f_c - df/2$. Ook voor de '1' heeft het spectrum deze vorm, maar nu gepositioneerd rond $f_c + df/2$. Het volledige spectrum van het FSK-signaal is de som van beide deelspectra, zoals in **figuur 3** is te zien.

Deze kleine analyse laat ons een ander erg belangrijk aspect zien. Om de gevoeligheid van de ontvanger te verbeteren



Figuur 4. Het echte spectrum van een MSK-signal. Herkent u $\sin(x)/x$?

moeten we de interferentie tussen de uitgezonden nullen en enen beperken. Kijk nog eens naar figuur 3. Als de afstand df tussen de twee frequenties gelijk is aan de breedte van de $\sin(x)/x$ -piek, dan zal de piek van het '0'-spectrum samenvalLEN met de plaats waar het vermogen van het '1'-spectrum minimaal is, en omgekeerd. Zo'n modulatie wordt orthogonaal genoemd en levert goede prestaties. Wat betekent dit in de praktijk? Dat is simpel: de breedte van de $\sin(x)/x$ piek is omgekeerd evenredig met de bitlengte, en die is gerelateerd aan de data rate. FSK-modulatie is dus orthogonaal als het frequentieverschil (tussen '1' en '0') gelijk is aan de data rate van de verbinding of een veelvoud hiervan. Als u bijvoorbeeld een zender hebt op 433,92 MHz en een data rate van 9600 bps, dan moeten de binaire FSK-frequenties in het ideale geval gelijk zijn aan 433,92 MHz ± 4800 Hz, of 433,92 MHz ± 9600 Hz enzovoort. Andere frequentieverzijlen kunnen ook worden gebruikt, maar geven minder goede resultaten. Bovendien zal het totale spectrum er 'netter' uitzien en hierdoor beter aan de wettelijke eisen voldoen.

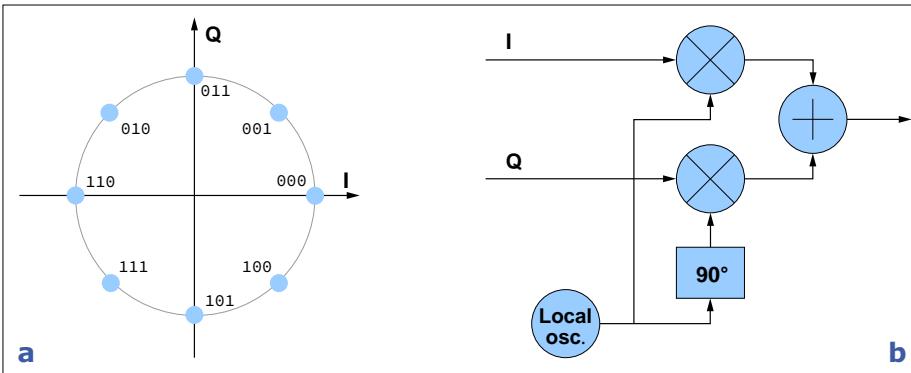
Natuurlijk kan er ook in FSK op meerdere niveaus worden gewerkt. U kunt bijvoorbeeld de bits in groepjes van vier samenvoegen en ieder symbool aan één van zestien verschillende fre-

quenties koppelen. Deze modulatie wordt 16-FSK genoemd, en hiermee is bij gelijkblijvende kanaalbreedte een vier keer hogere snelheid mogelijk dan met standaard-FSK. Ook hier is de prijs die moet worden betaald een geringere reikwijdte, omdat de gevoeligheid van de ontvanger onvermijdelijk lager zal zijn.

MSK

Nog een laatste opmerking over FSK: bij frequentiemodulatie is er ook nog een andere manier om de interferentie tussen de symbolen te minimaliseren. Als u het frequentieverschil heel nauwkeurig op de helft van de data rate instelt, dan is in theorie de interferentie ook nul. Dit is niet zichtbaar in figuur 3, en het is ook een beetje ingewikkeld om uit te leggen, want het heeft te maken met de fase van de signalen. Deze geoptimaliseerde modulatietechniek die MSK wordt genoemd (*Minimal Shift Keying*), wordt niet vaak toegepast in goedkope producten omdat de ontvanger nogal gecompliceerd is. Maar u bent er allemaal al mee in aanraking gekomen, het is de basis van GSM...

Even een klein lab-experiment om u een 'echt' MSK-signal te laten zien. Ik heb hiervoor gebruikgemaakt van een Keysight E4432B signaalgenerator, geconfigureerd om een MSK-signal te leveren, en ik heb de uitgang verbonden met



Figuur 5. a) Bij 8-PSK-modulatie worden acht verschillende fases gebruikt om drie bits tegelijk te coderen. b) Een IQ-modulator bestaat uit twee vermenigvuldigers die door een lokale oscillator worden aangestuurd, één in fase en één in kwadratuur. De twee signalen worden vervolgens bij elkaar opgeteld. Met de datasignalen die op I en Q worden aangeboden kan nu iedere fase tussen 0 en 360° en iedere amplitude worden ingesteld.

een vectoriële spectrumanalyzer Keysight E4406A (ik weet het, ik ben een geluksvogel...). En hopla! U ziet het resultaat in **figuur 4**. U herkent hier duidelijk $\sin(x)/x$, nietwaar?

Fasemodulatie

Naast amplitudemodulatie en frequentiemodulatie is er nog één andere mogelijkheid: fasemodulatie. Hierbij worden

amplitude en frequentie constant gehouden, maar wordt de fase van de draaggolf veranderd om onderscheid te maken tussen nullen en enen. De BPSK-modulatie (*Binary Phase Shift Keying*) gebruikt twee verschillende fases, 0° en 180° , voor het verzenden van respectievelijk nullen en enen. Fasemodulatie heeft een principeel nadeel: de ontvanger kan zonder een referentie op geen enkele manier

de absolute fase van een signaal vaststellen. Er zijn twee oplossingen voor dit probleem, die beide worden toegepast. Bij de eerste wordt aan ieder bericht een bitreeks toegevoegd om aan de ontvanger aan te geven wat de referentiefase is. Met een PLL kan de ontvanger vervolgens de verschillende fases detecteren. De andere oplossing is de informatie niet te coderen als absolute fases, maar als faseverschillen. Deze modulatietechniek wordt DPSK genoemd (*Differential Phase Shift Keying*).

Waarom zouden we zoveel moeite doen voor PSK? Omdat deze het mogelijk maakt om meer dan twee niveaus te gebruiken zonder dat het spectrum breder wordt zoals bij FSK, en zonder de storingsgevoeligheid te vergroten zoals bij ASK. Zo is er bijvoorbeeld een modulatietechniek met de welluidende naam QPSK, die gebruikmaakt van vier fases (0° , 90° , 180° en 270°) om twee bits tegelijk te coderen. Bij 8-PSK worden acht fases gebruikt die 45° uit elkaar liggen om drie bits tegelijk te coderen. Deze beide modulatietechnieken worden toegepast bij bijvoorbeeld Bluetooth.

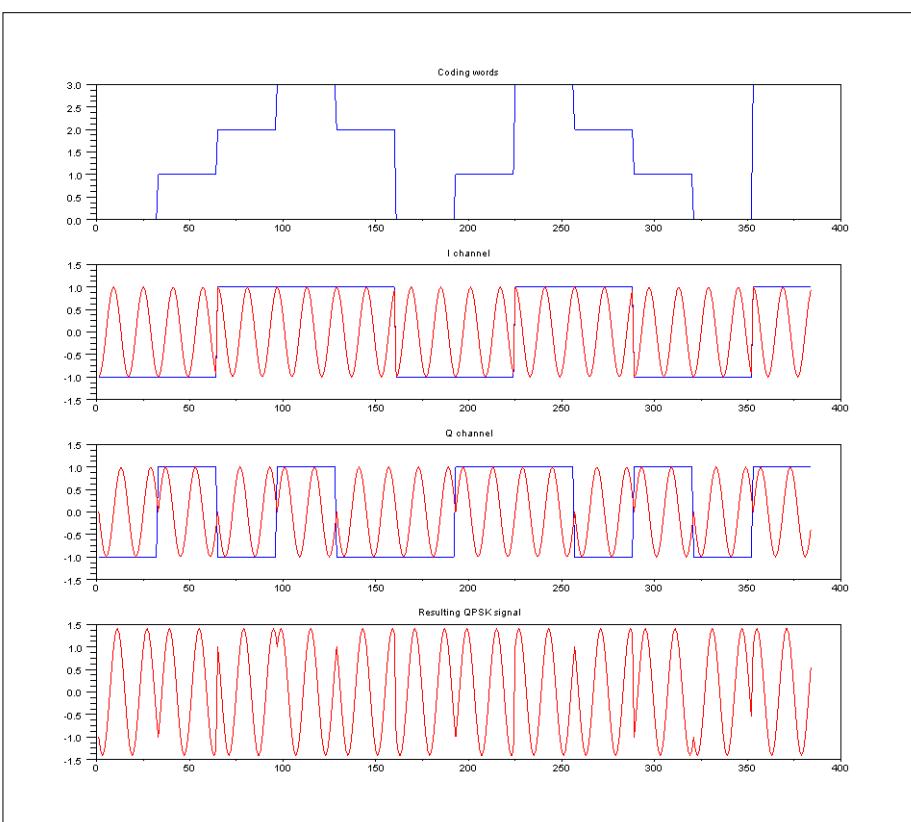
I/Q?

Een handige manier om fasemodulatie zoals 8-PSK weer te geven is met een pooldiagram, zoals in **figuur 5a** is te zien. Zo'n diagram is meer dan alleen maar tekening, het is een illustratie van de manier waarop een fasemodulator is de praktijk is geconstrueerd. Stel dat iemand u vraagt om een schakeling te maken waarmee een signaal over een variabele fasehoek kan worden verschoven. Niet eenvoudig? Dat valt wel mee, u hoeft alleen maar twee versies van de draaggolf te produceren, één in fase en de andere in kwadratuur (dat wil zeggen 90° in fase verschoven), en deze via regelbare versterkers bij elkaar op te tellen (zie **figuur 5b**). De respectievelijke amplitudes komen dan overeen met de projectie van de gewenste vector op de horizontale en verticale assen, die met I en Q worden aangeduid. Dit wordt een I/Q-modulator genoemd en hiermee kan een signaal met iedere willekeurige fase worden gegenereerd.

Dit werkt als volgt:

$$\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$$

Met $a = 2\pi f$ en $b = \varphi$ krijgen we het volgende:



Figuur 6. Een voorbeeld van QPSK-modulatie. Bovenaan ziet u de te verzenden symbolen (vier niveaus). De twee afbeeldingen in het midden zijn respectievelijk van het I- en het Q-signal (blauw) en van de uitgang van de vermenigvuldiger. Tot slot ziet u onderaan het uiteindelijke gemoduleerde signaal.

$\sin(2\pi f + \varphi) =$
 $\sin(2\pi f)\cos(\varphi) + \cos(2\pi f)\sin(\varphi)$
 Omdat $\cos(a) = \sin(a + \pi/2)$, kunnen we het geheel herschrijven als:

$$\sin(2\pi f + \varphi) = I \cdot \sin(2\pi f) + Q \cdot \sin(2\pi f + \pi/2)$$

waarin $I = \cos(\varphi)$ en $Q = \sin(\varphi)$ constanten zijn die afhangen van de gevraagde fase-afwijking.

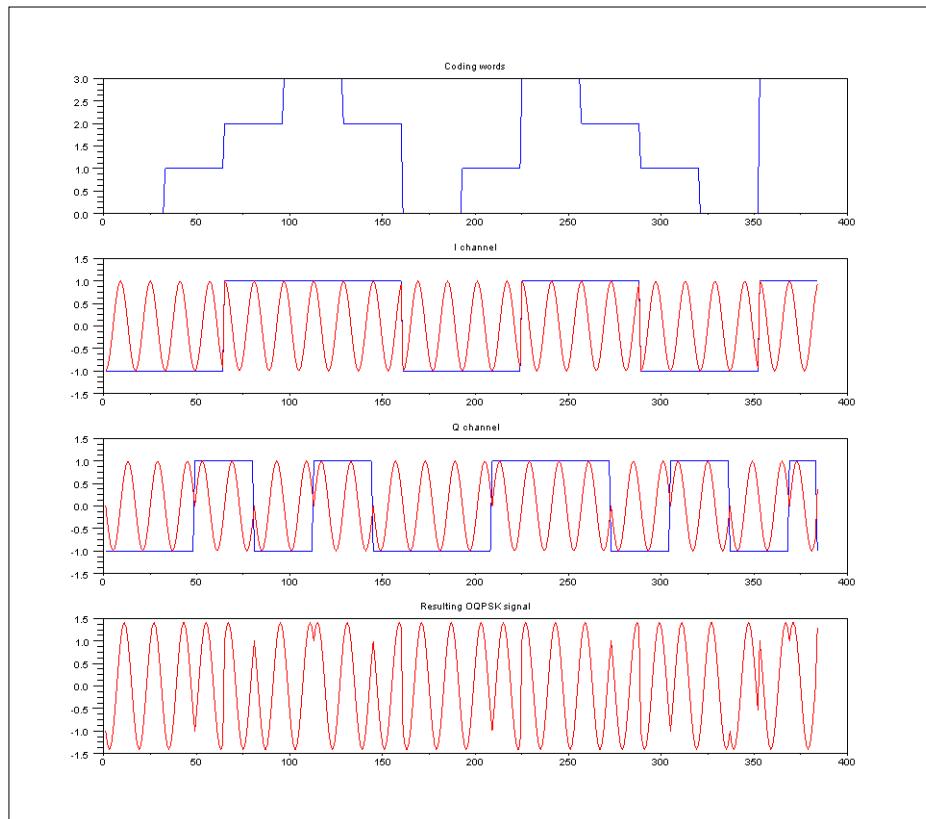
U herkent hier de twee draaggolven, in fase en in kwadratuur, vermenigvuldigd met de waarden I en Q en bij elkaar opgeteld. Maakt u zich geen zorgen als dit nog steeds wat mysterieus klinkt, ik ben voor een volgend nummer een artikel aan het voorbereiden dat speciaal over dit onderwerp gaat...

In **figuur 6** ziet u een voorbeeld van QPSK-modulatie zoals deze wordt gebruikt voor de WiFi-variant 802.11b (11 Mbps).

ASK+PSK=QAM

In figuur 5a ziet u dat bij PSK de verschillende waarden als punten op een cirkel worden weergegeven, omdat ze overeenkomen met verschillende fasen maar een constante amplitude hebben. Hoe kunnen we nu nog meer bits per symbool verzenden? Door niet alleen met de fase te spelen maar ook met de amplitude. Iedere combinatie van fase en amplitude vormt dan een symbool en hiermee wordt dus een soort combinatie van PSK en ASK gecreëerd. Eigenlijk is het efficiënter om de verschillende symbolen in het I/Q-vlak te plaatsen in plaats van combinaties van fase en amplitude te gebruiken, maar dit komt er erg dicht bij in de buurt. Deze techniek wordt QAM genoemd (*Quadrature Amplitude Modulation*). In **figuur 8** ziet u de symbolen die horen bij 16-QAM waarbij zestien combinaties van fase en amplitude worden gebruikt om vier bits tegelijk te coderen. Het goede nieuws is dat de IQ-modulator die hierboven is beschreven ook voor QAM kan worden gebruikt door de respectievelijke amplitudes te variëren. In **figuur 9** ziet u een Scilab-simulatie van 16-QAM (bestand te downloaden van [2]).

Gezien de prestaties wordt QAM meestal gebruikt voor toepassingen waarbij een hoge data rate bij een beperkte bandbreedte is gewenst. 16-QAM, 32-QAM en zelfs 256-QAM vormen bijvoorbeeld de basis voor de WiFi- en DVB-standaarden (DVB-T t/m DVB-C). Let op, een modula-



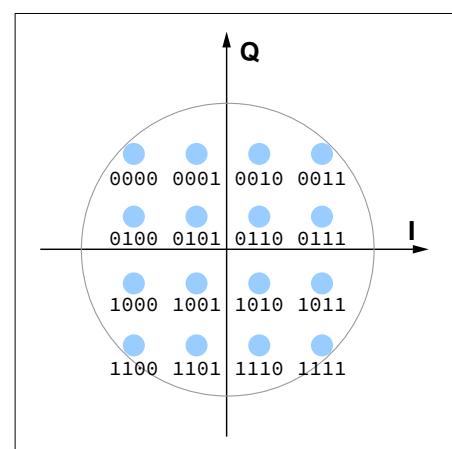
Figuur 7. OQPSK is een variant van QPSK waarbij het Q-kanaal een halve bit naar rechts is verschoven ten opzichte van het I-kanaal. Als u deze figuur vergelijkt met figuur 6 ziet u dat de faseovergangen wat minder abrupt zijn.

tietechniek zoals 256-QAM, waarbij een volledige byte in één keer door middel van 256 IQ-waarden wordt verzonden, is erg gevoelig voor interferentie en ruis. Zoals altijd gaat de snelheidswinst ten koste van de gevoeligheid van de ontvanger en dus van de reikwijdte van de verbinding.

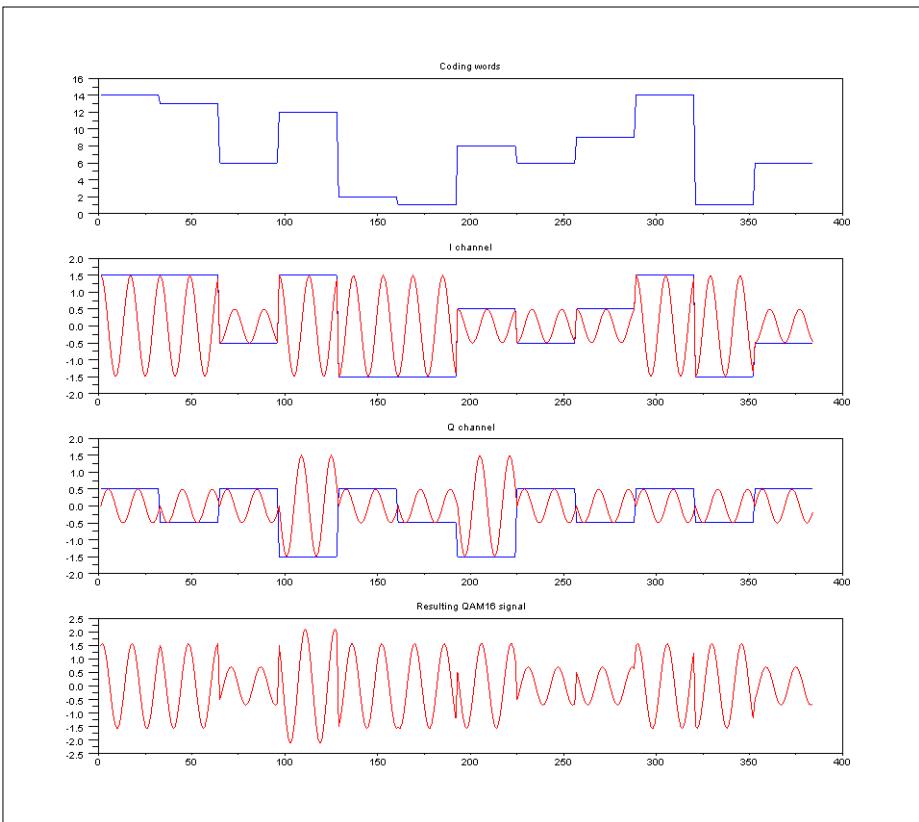
Van FSK naar OFDM...

Toen we het over FSK hadden heb ik u uitgelegd hoe de interferentie tussen symbolen kan worden geminimaliseerd door de frequentieververschuiving gelijk te maken aan een veelvoud van de binair data rate. Het gaat er om dat de piek van het spectrum van een van de twee frequenties precies in een minimum van het spectrum van de andere terechtkomt. Dit levert een orthogonale modulatie op. We vinden hetzelfde principe terug in een reeks modulatietechnieken die OFDM wordt genoemd (*Orthogonal Frequency Division Multiplexing*). OFDM is eigenlijk een soort super-FSK. Ten eerste gebruikt OFDM niet slechts twee, maar honderden frequenties, meestal met regelmatige tussenruimtes. Ten tweede zijn deze frequenties geen simpele sinusoïden

gemoduleerd door een rechthoekige puls, zoals bij FSK, maar wordt iedere frequentie gebruikt als sub-draaggolf voor een compleet transmissiekanaal. Elke van deze honderden sub-draaggolven is gemoduleerd met een digitale modulatietechniek, bijvoorbeeld QAM, waarbij rekening wordt gehouden met de juiste



Figuur 8. Een 16-QAM-configuratie waarin vier bits tegelijk zijn gecodeerd. Dit geeft zestien punten in het I/Q-vlak, elk corresponderend met een fase en een amplitude.



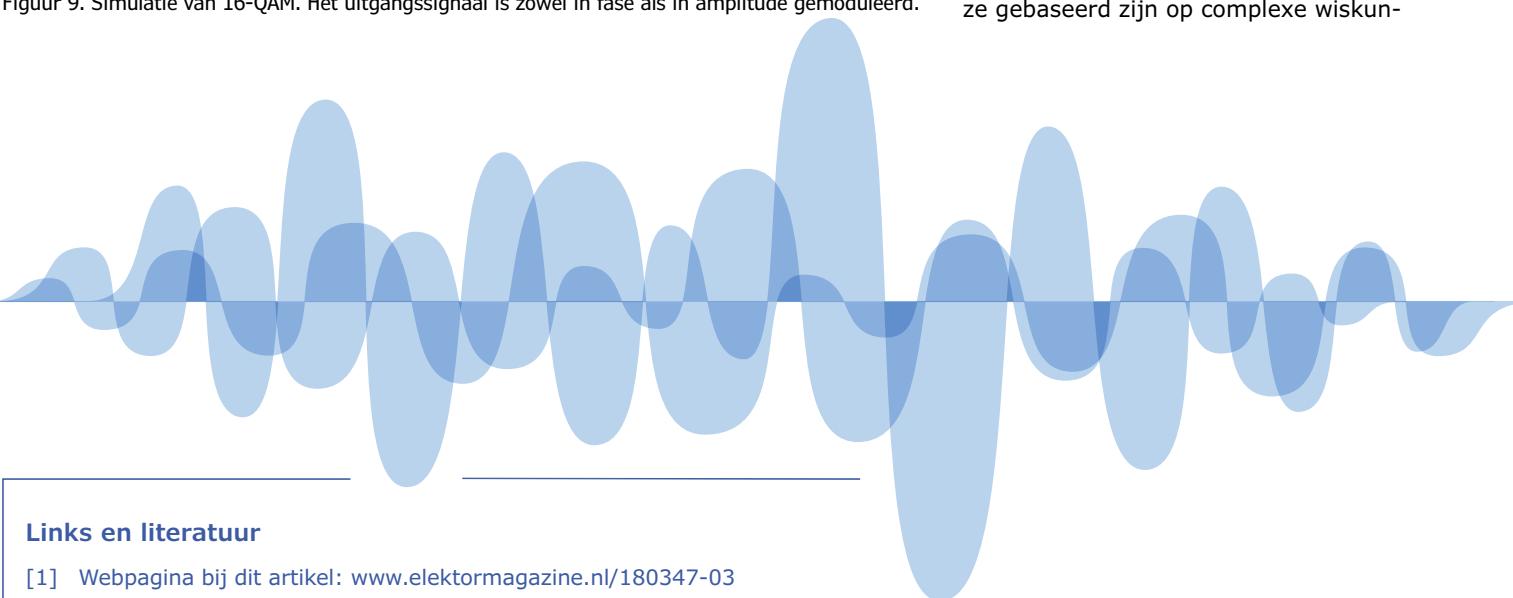
Figuur 9. Simulatie van 16-QAM. Het uitgangssignaal is zowel in fase als in amplitude gemoduleerd.

keuze van de verschillende data rates om de orthogonaliteit te waarborgen. Via deze honderden draaggolven wordt een stortvloed aan gegevens simultaan verzonden.

U kunt zich voorstellen dat de totale data rate enorm kan zijn! Daarom wordt OFDM toegepast bij ADSL, WiFi 802.11n, DAB-radio's, digitale video voor DVB-H en DVB-T enzovoort. Bij ADSL2+ bijvoorbeeld, kan de telefoonlijn frequenties van 0 tot 2,2 MHz transporteren. Deze band wordt verdeeld in 512 kanaal van ieder 4,3125 kHz waarbij voor ieder kanaal automatisch de juiste modulatie wordt gekozen waarmee, afhankelijk van de eigenschappen van het kanaal, 1 tot 15 bits per symbool kunnen worden verzonden. De maximale data rate van ADSL2+ is dus gelijk aan $512 \times 4,3125 \text{ kHz} \times 15 \text{ bits}$, en dat is ongeveer 33 Mbps. Niet slecht voor zo'n simpele telefoonlijn...

Tot slot

Digitale modulatietechnieken zijn niet altijd eenvoudig te doorgronden, omdat ze gebaseerd zijn op complexe wiskun-



Links en literatuur

- [1] Webpagina bij dit artikel: www.elektormagazine.nl/180347-03
- [2] 'Digitale filters eenvoudig uitgelegd: CIC-filters', Elektor 03/2016: www.elektormagazine.nl/150636
- [3] 'Digital Modulation in Communications Systems – An introduction', Agilent, Application Note 1298: <http://cp.literature.agilent.com/litweb/pdf/5965-7160E.pdf>
- [4] 'Wireless Digital Communications: Design and Theory', Tom McDermott, NEG, Tucson Amateur Packet Radio Corporation, ISBN 0-9644707-2-1: <http://www.tapr.org>
- [5] 'Intuitive Guide to Principles of Communications: All about modulation', Charan Langton: www.complextoreal.com/chapters/mod1.pdf
- [6] Scilab: www.scilab.org
- [7] Digitale signaalgenerator 250 kHz-3 GHz, Agilent: www.agilent.com
- [8] I/Q-modulatie generator E4406A, Agilent: www.agilent.com

dige begrippen. Ik hoop dat met dit kleine artikel de belangrijkste eigenschappen duidelijk zijn geworden. In het volgende artikel ga ik op dezelfde manier verder met een aspect dat hier al even voorbijkwam: de verschillende technieken om gegevens te coderen. ▀

(180347-03)

Dit artikel is eerder gepubliceerd in het tijdschrift Circuit Cellar (nr. 233, december 2009).

Elektor Labs Pipeline

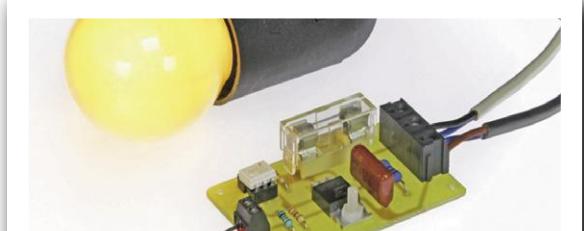


Projecten voor timers en klokken zijn altijd populair. Die dingen komen overal van pas en er is altijd veel ruimte voor creativiteit, vooral als het gaat om een visueel aantrekkelijke presentatie

 **Rijk en beroemd worden? Post uw projecten @ Elektor Labs**

LED/gloeilamp-converter met galvanische scheiding

De meeste mensen willen gloeilampen vervangen door LED-verlichting, maar wat als u nou net het omgekeerde wilt? Deze omzetter maakt dat mogelijk. Vervang gewoon de LED door een optocoupler die de lamp in- en uitschakelt.



@ Elektor Labs: www.elektor-labs.com/1704

Slaapkamerklok die ook de buitentemperatuur toont

Wilt u graag weten hoe koud het buiten is voordat u uit bed stapt? Dan is deze zelfbouwklok echt iets voor u. Een Raspberry Pi die dienstdoet als domotica-server, een ESP32 WiFi-module met daarop aangesloten een DHT22-sensor, en een tweede ESP32 die een LED-display aanstuurt. Meer is niet nodig. Klinkt moeilijk? Welkom in 2019.



@ Elektor Labs: www.elektor-labs.com/1605

Bouw een niet-programmeerbare timer zonder toeters en bellen (oftewel zandloper)

'Feature creep' is het fenomeen dat er allerlei toeters en bellen aan een product of een programma worden toegevoegd die eigenlijk helemaal niet nodig zijn. Neem bijvoorbeeld kleur- en lettertype-opties in smartphone-apps of zoveel andere obscure instelmogelijkheden. Daar heeft dit timertje geen last van. Of toch?



@ Elektor Labs: www.elektor-labs.com/478

Bouw een klok/scorebord/timer met XL-cijfers

Als u ooit een groot display nodig hebt om de score bij wedstrijden weer te geven, of voor een teller bij sportevenementen, dan is dit project iets voor u. Met vier grote (10 cm) 7-segment-displays kan hij van grote afstand prima worden afgelezen. Dit project is gebaseerd op LEDtron-modules, dus 7-segment-displays die met LED-filamenten zijn gemaakt die uit moderne LED-lampen zijn gesloopt. En ja, het is ook een klok! ▶

(180571-B-02)



@ Elektor Labs: www.elektor-labs.com/1299

Energiezuinige LED-klok

toont de tijd alleen wanneer nodig

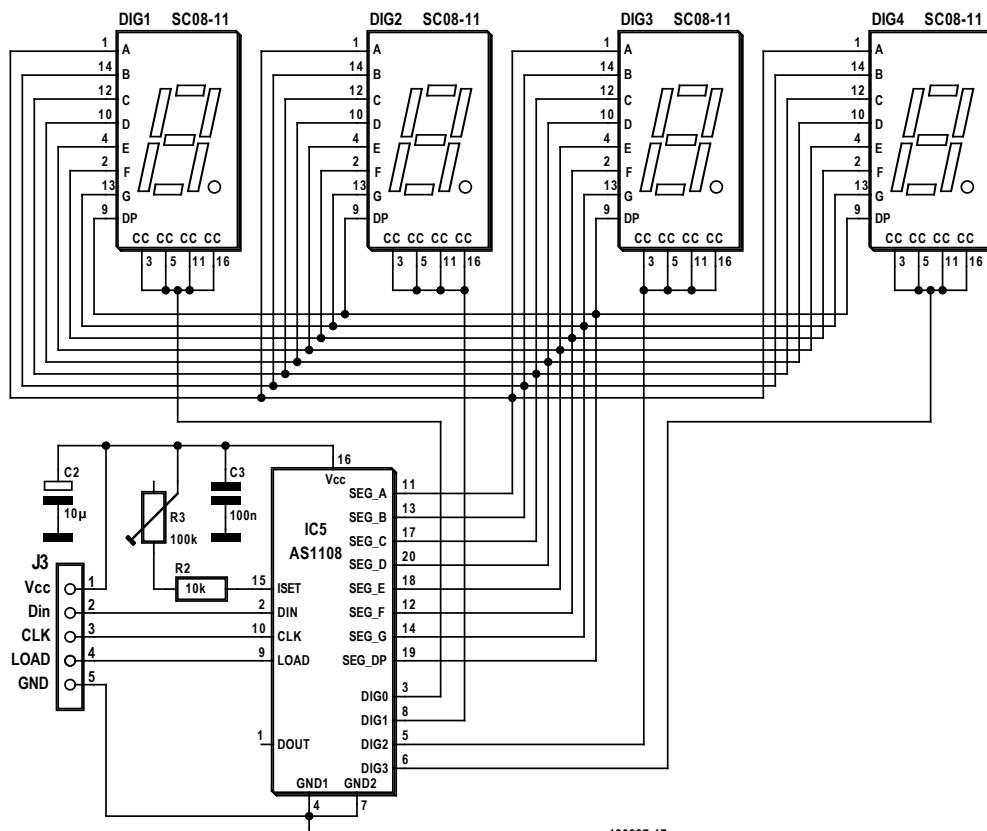
Miroslav Cina (Duitsland)

Klokken met een LC-display zijn in de schemering — zelfs met achtergrondverlichting — nauwelijks af te lezen. Maar met achtergrondverlichting gebruikt een LCD helaas net zoveel stroom als een LED-display. In een batterijgevoede wandklok kan echter zonder bezwaar een LED-display worden toegepast, als dat alleen dan wat laat zien als u het wilt aflezen. En daar is een PIR-bewegingssensor heel geschikt voor.

Wand- en andere klokken die vanwege afleesbaarheid met een LED-display zijn uitgerust, hebben een netvoeding en stopcontact nodig, om te voorkomen

dat u om de haverklap de batterijen moet vervangen. Maar vaak wilt u juist daar een klok zetten of hangen waar geen stopcontact vorhanden is. Een ener-

giezuinig LC-display is geen echt alternatief, omdat dat in het donker niet is af te lezen. Zeven-segment LED-display zijn wel contrastrijk en geven zelf licht,



Figuur 1. De schakeling van de display-module is heel eenvoudig: een display-driver en vier zeven-segment displays.

maar gebruiken te veel stroom. Hoe lossen we dat op?

Stroomparend LED-display

Er zijn meerdere mogelijkheden om het stroomverbruik van een LED-display te reduceren. We kunnen bijvoorbeeld minder stroom laten lopen, zodat de LED-segmenten nog maar zwak oplichten. In het donker geen probleem, maar in helder licht zien we dan niks meer. Beter zou het zijn als het LED-display alleen dan licht geeft als we de tijd willen weten. Om hiervoor een druckscha-kelaar te gebruiken is noch ergonomisch, noch erg praktisch. Daarom heb ik ervoor gekozen een bewegingsmelder in de vorm van een PIR-sensor toe te passen. Als we nu langs de klok lopen of de hand ernaartoe uitsteken, dan wordt gedurende een paar seconden de tijd getoond. De gemiddelde stroomopname is daardoor zeer laag.

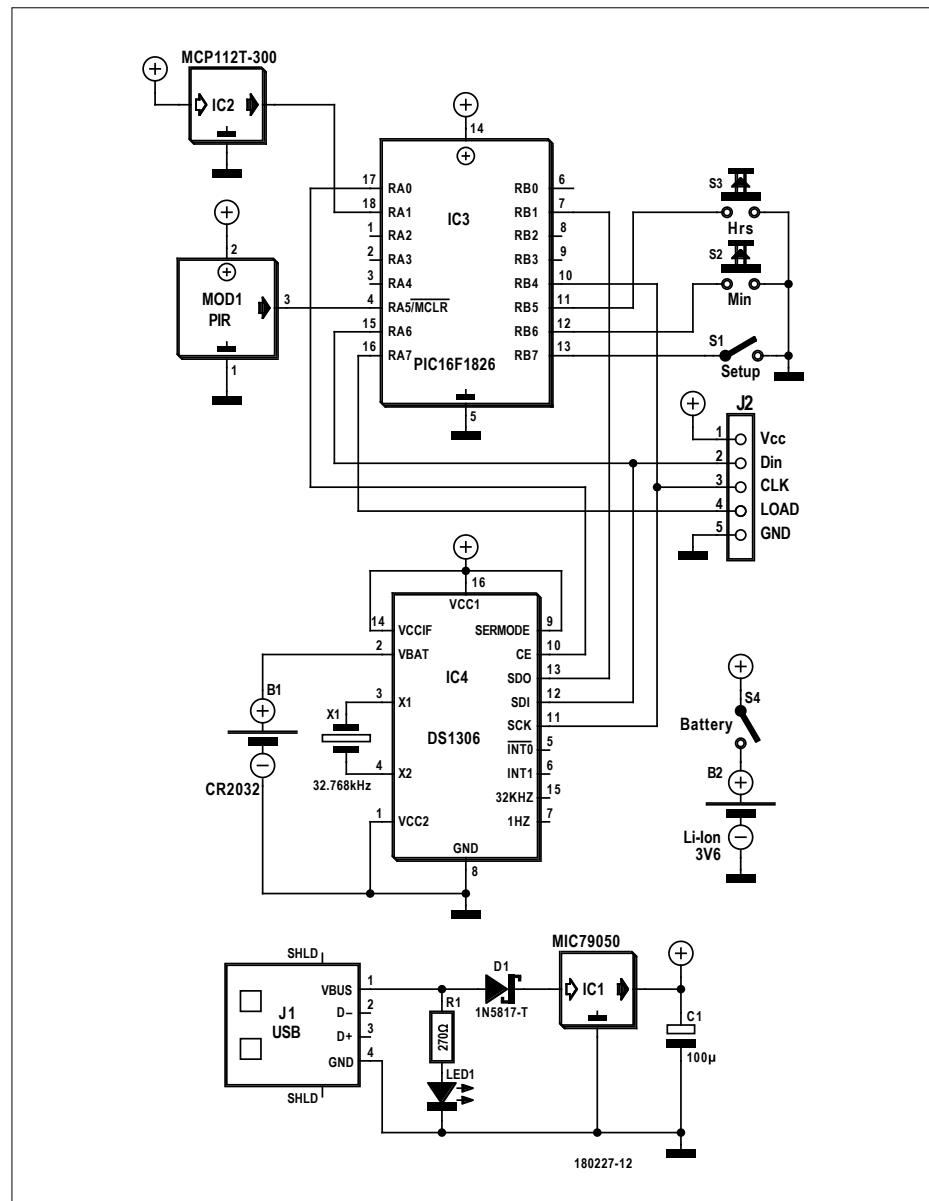
Overwegingen vooraf

Voor mijn PIR-klok volstaat het als alleen uren en minuten worden aangegeven. Dan zijn dus maar vier zevensegment displays nodig, hetgeen eveneens tot een geringer stroomverbruik leidt. Omdat netvoeding op bezwaren stuit, moet de klok met batterijen of accu's worden uitgerust. Verder moet de klok bij een batterijwissel gewoon doorlopen, zodat hij niet telkens opnieuw moet worden ingesteld. En tot slot is het — zoals eerder aangegeven — voldoende als alleen de tijd wordt getoond als er een persoon in de buurt is.

Mijn ideale klok wordt gevoed uit een Li-Ion accu, wat tevens batterijenwisseling overbodig maakt. Maar dan moet er dus wel een geschikte laadschakeling worden ingebouwd, die gelukkig tegenwoordig in IC-vorm verkrijgbaar is. Om ervoor te zorgen dat de klok bij een lege accu niet blijft stilstaan, is tevens een RTC-chip (Real Time Clock) nodig, met een kleine backup-batterij. Als aanwezigheidsdetector fungeert een geschikte PIR-sensor.

Hardware

De schakeling is zeer overzichtelijk. Wat we nodig hebben is een RTC-IC van Maxim, een microcontroller, een display-driver, vier zevensegment LED-displays en een PIR-sensor. Daarbij komt een Li-Ion cel plus laadschakeling en spanningsbewaking om te diepe ontlasting van de accu's te voorkomen.



Figuur 2. Ondanks de vier IC's is ook de eigenlijke klok niet echt ingewikkeld.

De hardware van de klok bestaat uit een display-module met -driver aan de ene kant en de eigenlijke klok met de overige functies aan de andere kant.

Displaymodule

De schakeling van de displaymodule is — zoals te zien in **figuur 1** — heel eenvoudig van opzet. De vier zevensegment displays van het type SC08-11 worden in matrix-configuratie aangestuurd door de displaydriver AS1108 van AMS. In principe zijn alle zevensegment display met gemeenschappelijke kathode geschikt. Ter vermindering van spanningsdips bij stroompieken of storingen, die zouden kunnen leiden tot een ongewenste reset van IC5, dienen de beide condensatoren C2 en C3. De helderheid van de displays

hangt af van de totale weerstand van de serieschakeling van R2 en trimpotmeter R3. Bij minimale waarde van R3 is het display het helderste. Met R3 kan de helderheid op een voor u prettige waarde worden ingesteld.

De hoofdmodule

Alle overige functies zitten op de hoofdmodule. Hoewel deze schakeling (**figuur 2**) uit vier IC's bestaat, is hij toch gemakkelijk te begrijpen. De eigenlijke klok zit in het RTC-IC DS1306 (IC4). Deze telt de tijd, maar bevat daarnaast nog een paar handige functies, zoals datum- en alarmopties. Als externe klok fungeert kristal X1 met de gebruikelijke frequentie van 32.768 kHz. De backup-batterij, een

MSSP

Een MSSP kan in veel verschillende modi worden gebruikt. Een MSSP kan bijvoorbeeld als I²C- of als SPI-interface worden geconfigureerd. Bij PIC's is in beide modi de configuratie als slave of als master mogelijk.

Veel microcontrollers bieden slechts een SSP, dus zonder 'M'. Deze kunnen slechts als slaaf worden geconfigureerd. Maar de 'M' (master) bespaart veel tijd en programmageheugen wanneer we de seriële interface als master willen gebruikend.

De SPI-bus is in 1979 oorspronkelijk door Motorola bij de invoering van legendarische 68K-CPU ontwikkeld. Het gaat hier om een synchrone seriële full-duplex interface. Dat maakt de zaak weliswaar wat complexer, omdat voor de bidirectionele communicatie naast de klok op SCLK en de beide datalijnen MOSI (Master Output, Slave Input) / SDO (Serial Data Out) en MISO (Master Input, Slave Output) / SDI (Serial Data In), ook nog minstens een CS-lijn (Chip Select) en massa nodig zijn. Eigenlijk hebben alle slaves eigen CS-lijnen nodig.

Hierdoor en door het gebruik van push-pull drivertrappen zijn in vergelijking met de open collector-technologie bij I²C hogere datasnelheden mogelijk. Nadeel is de begrenste kabellengte, ongeveer gelijk aan die bij RS232. Vanwege snelheden tot zo'n 80 MBit/s is SPI vooral geschikt voor koppelingen tussen chips met hoge datauitvoer zoals snelle ADC's of externe flash-geheugens. Veel microcontrollers ondersteunen beide protocollen gelijktijdig (bijvoorbeeld MSSP1 als I²C-master en MSSP2 als SPI-master) en zijn daardoor zeer flexibel inzetbaar.

Overigens spelen in een toepassing zoals deze klok hoge datasnelheden überhaupt geen rol.

als SPI-master. IC3 vraagt niet alleen via SPI de tijd op bij IC4, maar stuurt via deze bus ook display-driver IC5 op de display-module. IC3 heeft nog genoeg poorten vrij om eventuele eigen uitbreidings te realiseren.

IC2 is een spanningsdetector van het type MCP112T-300 van Microchip. Zijn uitgang wordt 'low' als de spanning op zijn ingang lager wordt dan 3,0 V. Daardoor weet IC3 dat accu B2 leeg is en kan maatregelen treffen om te voorkomen dat de accu te diep wordt ontladen. Bij een nieuwe batterij kan deze spanning maximaal 3,2 V bedragen. Dan draait IC4 nog geruime tijd op B2, zolang IC2 niet heeft geschakeld. IC2 is dus ook niet dwingend noodzakelijk. Als die wordt weggeleggen, moeten de pinnen 1 en 2 worden doorverbonden.

Als PIR-sensor dient het goed verkrijgbare type PIR 555-28027. Van belang is dat deze kan werken met spanningen tussen de 3 en 5 V en dat zijn uitgang 'high' wordt als een beweging wordt gedetecteerd.

IC1 tenslotte is een geïntegreerde laadschakeling voor een enkele Li-Ion cel. Deze laadt de accu op tot een spanning van 4,2 V met een stroom van maximaal 500 mA. De spanningsval over IC1 bedraagt 380 mV bij 500 mA, zodat de schakeling prima met een gewone 5-V-adapter via J1 kan worden gevoed. D1 beveilt tegen verkeerd-om aansluiten van de voedingsspanning en LED1 licht op als de schakeling vanuit het lichtnet wordt gevoed. Buffercondensator C1 verlaagt de — al behoorlijk lage — inwendige weerstand van B2 nog wat meer.

Tabel 1. SPI-routines.

Subroutine	Functie	Toepassing
spi_init_m0	MSSP-initialisatie SPI Master Mode 0	SPI Mode 0 voor de communicatie met display-driver AS1108
spi_init_m1	MSSP-Initialisatie SPI Master Mode 1	SPI-Mode 1 [TS1] voor de communicatie met de RTC DS1306
spi_send	Zenden en ontvangen via de SPI-interface	SPI-communicatie: 1 byte zenden of ontvangen (gelijktijdig)

makkelijk verkrijgbare Lithium-cel van het type CR2032, is aangesloten op pin 2 van IC4. De backup-logica werkt heel eenvoudig: als de spanning van accu B2 op pin 16 van IC4 hoger is dan die van B1, dan bevindt de klok zich in een normale situatie en wordt hij gevoed door B2. Is UB2 < UB1, dan is B2 kennelijk ontladen en voedt B1 IC4, maar dan ook uitsluitend IC4, dat via zijn SPI-communicatie de microcontroller informeert over de backup-toestand. De rest van de schakeling wordt nog steeds door B1 gevoed en de microcontroller meldt optisch dat de accu moet worden opgeladen. Een opmerking met betrekking tot de

SPI-communicatie van IC4: deze chip maakt gebruik van een nogal ongebruikelijk, quasi-geïnverteerd CS-signal: een 'high' op CS geldt hier als actief. Microcontroller PIC16F1826 van Microchip vormt als IC3 het hart van de schakeling. Deze biedt 2 KW flash-programmageheugen. De eenheid KW staat hier niet voor KiloWatt, maar voor KiloWoord — een adresseerbare geheugenplaats is namelijk 14 bit breed. Daarnaast hebben we nog de beschikking over 256 bytes RAM en 256 bytes EEPROM (die hier niet worden gebruikt). IC3 beschikt over een MSSP-interface (Master Synchronous Serial Port). Deze wordt geconfigureerd

Firmware

De firmware is geheel in assembler geschreven en heel compact. Van het programmageheugen worden minder dan 400 woorden gebruikt. De software is net als de hardware in tweeën gedeeld: bestand 91_SPI_comm_16F1826.inc bevat alle subroutines voor de SPI-communicatie en het bestand 01_PIR_clock_v1p09.asm bevat de eigenlijke klok, samen met de overige functies. De broncode voor dit project, die van de webpagina bij dit artikel kan worden gedownload [1], kan met een editor worden aangepast aan eigen wensen en kan met MPASM (onderdeel van de MPLAB IDE van Microchip) versie 5.65 worden gecompileerd. Een reeds gecompileerd en geassembleerd hex-bestand is eveneens via [1] te downloaden.

SPI-communicatie

Omdat de SPI-routines ook voor andere projecten bruikbaar zijn, worden deze hier beschreven. Informatie over de drie in het include-bestand *91_SPI_comm_16F1826.inc* opgenomen subroutines is te vinden in **tabel 1**. Deze zijn geschreven voor de MSSP-module van de PIC16F1826. Met enkele kleine aanpassingen zijn ze voor alle 8-bits microcontrollers met MSSP-module te gebruiken.

De subroutines voor de CS-signalen (voor AS1108 en DS1306) zijn heel eenvoudig en aangepast aan deze schakeling en de gebruikte IC's. Het gaat hier om subroutines *spi_cs0_en* en *spi_cs1_en*, die CS0 en CS1 activeren. CS0 is voor driver AS1108 en CS1 voor de RTC. Met *spi_cs0_dis* en *spi_cs1_dis* worden CS0 en CS1 weer gedeactiveerd. Bij deactivering worden overigens beide CS-signalen inactief.

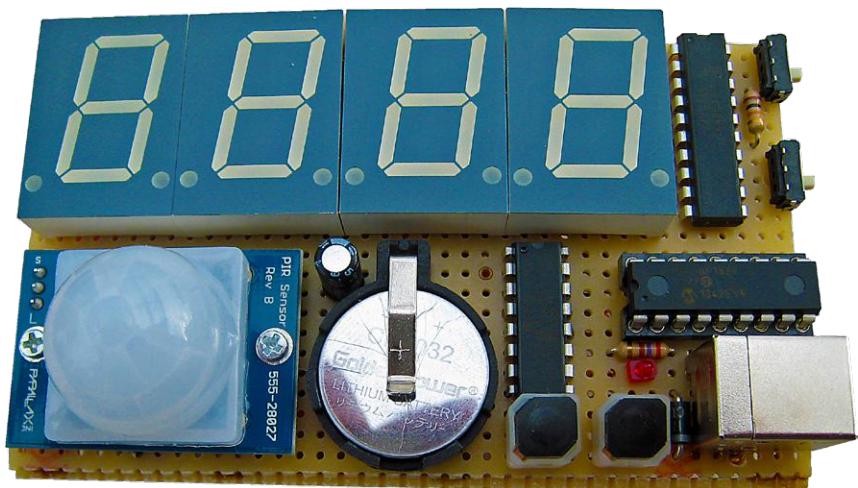
Hoofdprogramma

Zelfs het hoofdprogramma is eenvoudig gehouden. Na de initialisatie wordt in een eindeloze lus (*main_loop*) gesprongen, waar periodiek het volgende gebeurt:

- Afvragen van de PIR-sensor:
 - Geen beweging:
LED-display uit;
spring terug naar begin.
 - Beweging gedetecteerd:
ga verder bij het volgende punt.
- Opvragen tijdinformatie bij IC4
(subroutine *rtc_spi_read*).
- Converteeren ingelezen data naar weer te geven formaat.
- Geconverteerde data versturen naar display-driver IC5 om te worden weergegeven.
- Als de tijd moet worden bijgewerkt:
spring naar betreffende subroutine.

Na het bijwerken wordt de nieuwe tijd door de routine *rtc_spi_write* in IC4 geschreven.

Naast de hoofdlus en de tijdinstelling hebben we nog routines voor de display-driver. Deze zijn eenvoudig aan de prefix *AMS_* te herkennen.



Figuur 3. Het prototype van de auteur, opgebouwd op gaatjesprint.

Bediening

Er zijn slechts drie knoppen: S1, S2 en S3. Dat is genoeg om de tijd in te stellen. Daartoe zetten we eerst schakelaar S1 (*Setup*, pin RB6 van IC3) in de stand *Setup*. Om aan te geven dat we in de setup-mode zijn, lichten nu de decimale punten van de vier displays op.

Als we nu S2 (*Min*, pin RB6 van IC3) indrukken, tellen de minuten versneld omhoog. Na de waarde 59 komt de waarde 00. Op dezelfde wijze kunnen door het indrukken van S3 (*Hrs*, RB5 van IC3) de uren worden ingesteld. Na het terugzetten van S1 is de setup klaar en worden de (niet ingestelde) seconden intern op 00 gezet. Met S4 kan de accu worden ontkoppeld, waarna alleen de RTC nog verder loopt.

De klok geeft de tijd weer in 24-uur-formaat. Desgewenst kan dit in de broncode worden gewijzigd in 12-uur-formaat.

Als de microcontroller via IC2 een te lage spanning vaststelt, wordt de laatste decimale punt ingeschakeld. Dat is het teken dat B2 moet worden opgeladen. Gebeurt dat niet, dan zal IC4 naar de backup-mode omschakelen en wordt geen tijd afgebeeld, maar “--:--”. Dat zal ongetwijfeld opvallen.

Tot slot

Mijn klok (zie **figuur 3**) bevindt zich in de badkamer, waar alle gezinsleden

op weekdagen 's ochtends serieus de tijd in de gaten moeten houden. Dankzij PIR-sensor en 'display-on-demand' houden twee parallel geschakelde Li-Ion accu's van het formaat 18650 het drie maanden vol, voordat de klok weer moet worden opgeladen door hem een paar uur aan een adapter te hangen.

Er zijn nog wel meer mogelijkheden om het energieverbruik te reduceren. Zo zou bijvoorbeeld een lichtsensor kunnen worden toegevoegd, waarmee de helderheid van de displays afhankelijk van het omgevingslicht wordt ingesteld.

Voor vragen, verbetervoorstellingen en opmerkingen ben ik bereikbaar via miroslav.cina@t-online.de.



180277-04

Weblink

[1] Download: www.elektormagazine.nl/180277-04

SHOP-HIGHLIGHT

Meten, voeden en printen...

...met artikelen uit onze shop



Multimeter met Bluetooth-module

Een (hobby-)elektronicus heeft minstens twee apparaten nodig om te kunnen werken aan een elektronische schakeling: een multimeter en een soldeerbout. De meesten van ons hebben deze inmiddels wel, maar zo nu en dan moet er toch wel eens wat nieuws worden aangeschaft of is een extra multimeter handig om meer metingen gelijktijdig te kunnen verrichten.



Nu zijn er multimeters in overvloed te koop, met prijzen vanaf een tientje of zo tot duizenden euro's. Vanwege de degelijkheid en veiligheid is het aan te raden niet het allergoedkoopste exemplaar aan te schaffen.

Qua mogelijkheden is de hier besproken MS8238H van Mastech een doorsnee multimeter, met alle basisbereiken die je van zo'n apparaat mag verwachten: volt, spanning (DC en AC), weerstand, diodemeting en doorgangsspieper. Verder zijn er nog een temperatuurbereik (een thermokoppel wordt meegeleverd), twee batterij-meetstanden (voor 9 V en 1,5 V) en een NCV-stand om contactloos hoge (net)spanningen te detecteren. Sommige bereiken van de draaischakelaar hebben meerdere functies die met de Func-drukknop kunnen worden geselecteerd. Met een Hold-knop kan de uitlezing in het display worden bevroren. De Ran-knop dient om de gevoeligheid van een meetbereik handmatig te kiezen. Gewoonlijk kiest de meter automatisch het optimale bereik.

Tot slot zit er nog een blauwe Link-knop op de meter, waarmee de ingebouwde Bluetooth 4-module wordt ingeschakeld. Deze geeft dan de meetwaarde en de stand van de draaischakelaar door aan de bijbehorende app op een smartphone of tablet. Dat is best bijzonder voor een multimeter van nog (net) geen 50 euro!

Het 3 1/4-digit-display heeft geen achtergrondverlichting, maar is ook bij minder omgevingslicht nog redelijk af te lezen. De basisnauwkeurigheid is voor deze prijsklasse gebruikelijk met 0,5% voor DC-spanningsmetingen. Bij AC-metingen wordt gebruik gemaakt van true-RMS-meting.

IN DE STORE



→ Mastech-multimeter MS8238H

[www.elektor.nl/mastech-ms8238h-digital-multimeter
with-wireles](http://www.elektor.nl/mastech-ms8238h-digital-multimeter-with-wireles)

Voeding voor de Raspberry Pi

Opsteekmodules voor het voeden van de Raspberry Pi bestaan al langer. Niettemin heeft de StromPi 3 veel nieuws te bieden. Hij wordt met twee afstandsbusjes, twee schroeven en twee moeren geleverd, plus een heel beknopte handleiding in het Duits en in het Engels. Met behulp daarvan levert de montage van deze HAT geen onoverkomelijke problemen op.

De StromPi 3 ondersteunt de StromPi 3 Battery HAT, een 1000 mAh LiFePo4-accu met een nominale spanning van 3,2 V (3,2 Wh). Deze kan gewoon op de StromPi 3 worden gestoken (in spanningsloze toestand) en kan de Raspberry Pi dan voeden. De StromPi 3 is voorzien van verschillende spanningsingen van voor de Raspberry Pi. Eén daarvan heeft een zeer ruim bereik van 6 tot 61 V; daarnaast heeft hij ook een USB-poort met een buffercondensator van 1 F en een aansluiting voor de LiFePo4-accu. Daarnaast heeft de HAT een RTC, zodat de Pi ook bij stroom- of netwerkuitval over de kloklijn kan beschikken. De StromPi 3 is daarmee een prima hulpmiddel als de Raspberry uit een typisch 12-, 24- of 48-VDC-netwerk moet worden gevoed.



Voor het geval dat één van de voedingsbronnen van de StromPi 3 wegvalt, kan worden ingesteld, in welke volgorde op de resterende bronnen moet worden teruggevallen. Er zijn doorvoerconnectoren voor het stapelen van meerdere HAT's, die ook de PoE-aansluitingen van de Pi 3B+ doorlossen. Wat ook opvalt, zijn de USB-A- en twee microUSB-connectoren aan de zijkant. Daar kunnen apparaten aan de StromPi 3 worden aangesloten die meer dan 500 mA nodig hebben (zoals een harde schijf). De StromPi 3 is weliswaar ontworpen voor de Raspberry Pi, maar dankzij zijn constructie kan hij ook met andere SBC's, zoals de BeagleBoneBlack, worden gebruikt. De StromPi 3 is zowel een UPS (in combinatie met de battery HAT) als een universele voeding voor de Pi, met de nadruk op voeding. De hardware is in elk geval heel doordacht.



Anycubic Photon 3D-printer

De Anycubic Photon is een DLP 3D-printer (Digital Light Processing) die objecten print met behulp van vloeibare hars. Met UV-light wordt een beeld onder een doorzichtige bad met vloeibare hars geprojecteerd; vervolgens wordt het object laag voor laag opgebouwd op een platform dat langzaam omhoog wordt bewogen. In vergelijking met de conventionele FDM 3D-printer met een heet mondstuk om plastic draad in de gewenste vorm te extruderen is het in feite een eenvoudig proces zonder veel bewegende delen (alleen het bad met hars beweegt op en neer). Het printen gaat rechtstreeks met behulp van de gratis meegeleverde Photon slicer-software. Opgelet: de printer moet precies waterpas staan. Het printen zelf neemt enkele uren in beslag, ongeacht het aantal objecten dat tegelijk geprint wordt. Na het printen moet het product met alcohol worden schoongespoeld en in daglicht nog een tijdje uitharden. Een nadeel is wellicht dat het hars (die in verschillende kleuren verkrijgbaar is) niet werkelijk goedkoop is. Niettemin is deze printer, gelet op de prijs/kwaliteit-verhouding, een van de beste keuzes op de huidige printermarkt. ▶

180689-B-01

Tested by
elektorlabs

IN DE STORE



→ StromPi 3 – voeding voor de RPi

www.elektor.nl/strompi-3-power-solution-for-raspberry-pi

IN DE STORE



→ Anycubic Photon 3D-printer

www.elektor.nl/anycubic-photon-uv-resin-dlp-3d-printer

Hardware-ontwerp met (V)HDL (3)

een WS2812-LED-matrix aansturen

Jörg Zollmann (Duitsland)

In het vorige deel van deze serie hebben we een digitale klok met binaire tijdweergave gebouwd. Dit keer gaan we de klok uitrusten met een display bestaande uit een matrix van 64 WS2812-RGB-LED's. Het protocol waarmee de WS2812-LED's worden aangestuurd, is gelukkig heel simpel asynchroon. Voor onze hardware-driver in VHDL gebruiken we onder meer een toestandsmachine.

Een matrix van 64 WS2812-RGB-LED's is online bij verschillende leveranciers al vanaf 5 € te koop. Een WS2812-LED is meer dan alleen maar een RGB-LED. In de kleine behuizing zitten, naast de drie LED's zelf ook nog de drivers (stroombronnen) en een kleine controller. We kunnen die controller via een serieel eendraads-protocol vertellen hoe fel elk van de drie interne LED's moet oplichten. Er zijn op het internet veel microcontrollerprojecten (en software-bibliotheken) te vinden voor het aansturen van WS2812-LED's [1][2]. Wij gaan een laag dieper en 'programmeren' zelf een hardware-driver in VHDL om te communiceren met de controllers die in de LED's zijn ingebouwd.

Net als bij de voorgaande delen van de serie kan de software worden gedownload van de webpagina bij dit artikel [7]. De organisatie van de sourcecode en het gebruikte hardwareplatform hebben we in het eerste deel [4] al beschreven.

WS2812-protocol

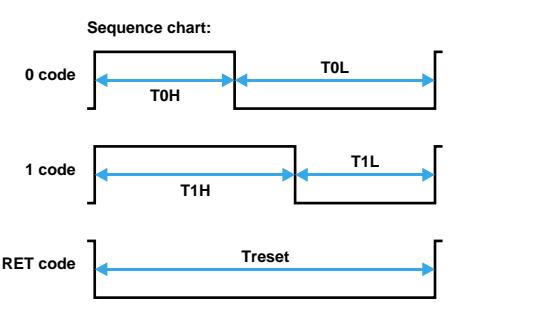
Als we in VHDL een hardware-driver willen schrijven die met een externe digitale chip moet communiceren, moeten we ons bezighouden met de interne details van de hardware en van het gebruikte protocol. Gelukkig werken de WS2812-LED's met een heel simpel asynchroon protocol. We kunnen daarmee LED's aan elkaar rijgen tot een keten: elke LED ontvangt 24 bit aan data via de seriële pin *DIN* en geeft de niet voor hem bestemde naar de volgende LED door via de uitgangspen *DOUT* [3].

Voor elk de drie kleuren wordt een 8-bits waarde tussen 0 en 255 gebruikt; die waarde bestuurt de stroomsterkte van de corresponderende LED-driver. Nadat de eerst 24 bits zijn ontvangen en opgeslagen, worden alle verder ontvangen bits versterkt en doorgegeven naar uitgang *DOUT*. Door de reconstructie van het signaal aan de uitgang is de data-overdracht erg robuust en kunnen we in theorie een willekeurig aantal LED's in serie schakelen. De grenzen worden bepaald door het stroomverbruik en de gewenste frame-update-snelheid. De ontvangen waarden worden pas doorgegeven aan de drivers als er een reset-code op de leiding wordt herkend. Die reset-code is een laag niveau van meer dan 50 µs op de lijn. De databits worden gecodeerd door hoge en lage pulsen van verschillende lengte op de lijn te zetten (zie **tabel 1** en **figuur 1**). Het tijdraster voor één bit is 1,25 µs. Als in die tijd de lijn een 0,35 µs hoog en 0,9 µs laag is, dan wordt dit bit herkend als 0. Om een 1 over te dragen moet de lijn 0,9 µs lang hoog zijn en 0,35 µs laag. De ontvangen data wordt geïnterpreteerd in de volgorde groen, rood, blauw en de volgorde van de bits is van MSB naar LSB.

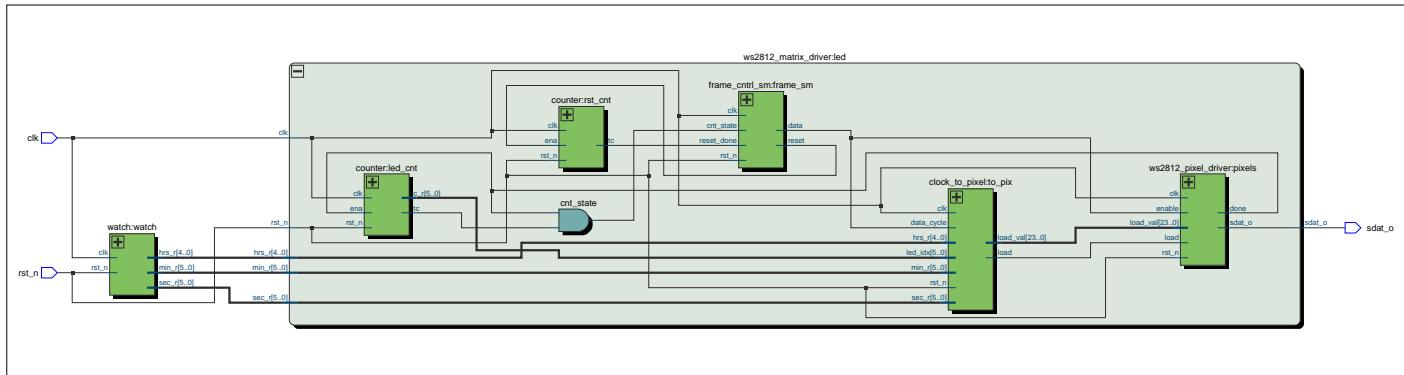
Bij een klokfrequentie van 50 MHz is het met de CPLD te bereiken oplossend vermogen in de tijd 20 ns. Bij het WS2812-protocol geldt de eis dat de pulsduren niet meer dan ±150 ns afwijken van de gespecificeerde waarden. Daar moeten we

Tabel 1. Datatransfer-tijden.

Naam	Beschrijving	Tijdsduur (TH+TL = 1,25 µs ± 150 ns)
T0H	0 code , high voltage time	0,35 µs
T1H	1 code , high voltage time	0,9 µs
T0L	0 code , low voltage time	0,9 µs
T1L	1 code , low voltage time	0,35 µs
RES	Reset-Code, low voltage time	> 50 µs



Figuur 1: WS2812-dataformaat [3].



Figuur 2: WS2812-watch in RTL-View.

ons bij onze pure VHDL-implementatie dus aan houden. De ervaring heeft ons trouwens geleerd dat de WS2812-LED's erg tolerant zijn voor kleine afwijkingen in de timing. Maar de lage pulsen mogen echt niet te lang uitvallen: ook pulsen die duidelijk korter zijn dan 50 µs worden door de LED-controllers soms al herkend als een reset-signalen.

Pixel-driver

De hier gepresenteerde WS2812-driver voor het weergeven van de kloktijd in binair formaat is hiërarchisch opgebouwd en bestaat uit meerdere modules (**figuur 2**). De module op het hoogste niveau instantieert de klok uit het vorige artikel [8] plus een instantie van de module *ws2812_matrix_driver*. Deze module heeft als ingangssignalen de gegevens uit het klokregister (hrs, sec, min), die op het 8x8-display moeten worden weergegeven. Binnenin de matrix-driver-module zitten ook weer verschillende componenten die elk een eigen taak vervullen. De component *ws2812_pixel_driver* is zó opgezet dat hij ook voor LED-strips of grotere matrices kan worden gebruikt (hier is dus ruimte voor eigen experimenten). De functie van de pixel-driver kan als volgt worden beschreven: bij het begin van een pixel wordt een 24-bits schuifregister (*do_reg*) gevuld met de helderheidswaarden voor de LED's. Het signaal *load* moet

daarvoor één klokplus lang op 1 gaan, dan wordt de waarde van het ingangssignaal *load_val* naar het schuifregister gekopieerd. Die inhoud moet dan serieel worden uitgeklokt (**listing 1**). Maar het protocol eist pulsen in plaats van pure 0- en 1-waarden op de datalijn. Dit wordt opgelost met twee processen. In het proces *code_proc* worden telkens de 'codes' voor een 0 of een 1 gegenereerd (**listing 2**). Welke code er dan op de uitgangslijn wordt gezet, wordt bepaald door het meest significante bit 23 in het schuifregister. Daarna volgen de andere bits. Deze uitgangsmultiplexer (Mux) verbindt dus de lijn *sdat_o* met de juiste code-generator, afhankelijk van de waarde in het schuifregister (**listing 3**).

Listing 1. Schuifregister

```
shift_proc: process (rst_n, clk)
begin
  if (rst_n = '0') then
    do_reg <= (others => '0');
  elsif rising_edge(clk) then
    if(enable= '1') then
      if (load = '1') then
        do_reg <= load_val;
      else
        if (bit_done) then
          do_reg <= do_reg(22 downto 0) & '0';
        end if;
      end if;
    end if;
  end process;
```

Listing 2. Zero One Code Process.

```
code_proc: process (rst_n, clk)
begin
  if (rst_n = '0') then
    zero_one_count <= 0;
    one_code <= '1';
    zero_code <= '1';
  elsif rising_edge(clk) then
    if (enable = '1') then
      if (bit_done)then
        zero_one_count <= 0;
        one_code <= '1';
        zero_code <= '1';
      else
        zero_one_count <= zero_one_count + 1;
        if (zero_one_count >= c_CLKS_T0_H_T1_L) then
          zero_code <= '0';
        end if;
        if (zero_one_count >= c_CLKS_T1_H_T0_L) then
          one_code <= '0';
        end if;
        end if;
      end if;
    end if;
  end process;
```

Listing 3. Uitgangsmultiplexer sdat_o.

```
mux : process (all) is
begin
  case (out_sel) is --
    when "00" => sdat_o <= '0';
    when "01" => sdat_o <= zero_code;
    when "10" => sdat_o <= '0';
    when "11" => sdat_o <= one_code;
    when others => sdat_o <= '0';
  end case;
end process;
```

De rest van de logica in de pixeldriver bestaat uit counters/tellers, zoals we die al in het vorige artikel hebben leren kennen. De tellers zijn geconfigureerd voor verschillende breedtes en tellen verschillende gebeurtenissen. Er is een counter die aangeeft wanneer er een bit is geschoven (dus elke 1,25 µs), en er is een counter die aangeeft wanneer er een byte is geschoven, en tenslotte een counter die aangeeft wanneer de pixel klaar is.

Van klok naar pixel

In ws2812_matrix_driver zitten nog twee andere tellers. Eén teller geeft aan, wanneer de tijdsduur voor een reset-code is

Listing 4. Frame Control toestandsmachine in één-proces-notatie.

```
-- Build an enumerated type for the state machine
type state_type is (data_cyc,reset_cyc);
-- Register to hold the current state
signal frame_state: state_type;
...
frame_ctrl: process (rst_n, clk)
begin
  if (rst_n = '0') then
    frame_state <= reset_cyc;
  elsif rising_edge(clk) then
    case frame_state is
      when reset_cyc => -- when current state
is reset_cyc
        frame_state <= reset_cyc;
        if(reset_done) then
          frame_state <= data_cyc;
        end if;
      when data_cyc => -- when current state is
data_cyc
        frame_state <= data_cyc;
        if(cnt_state) then
          frame_state <= reset_cyc;
        end if;
      when others => frame_state <= reset_cyc;
    end case;
  end if;
end process;
```

afgelopen en de andere geeft aan welk pixel op dit moment wordt uitgevoerd. Omdat de pixels van de matrix in serie geschakeld zijn, heeft elk van de 64 LED's een pixelnummer. De eigenlijke omzetting van de tellerwaarden van de klok naar de juiste 24-bits waarden voor de pixels wordt gedaan door de module *clock_to_pixel*. Deze module zorgt er in een synchroon proces voor dat het schuifregister in de pixel-driver telkens met nieuwe data wordt geladen. Daarvoor wordt het huidige pixelnummer opgehaald en geëvalueerd. Als het pixelnummer kleiner of groter is dan de pixelnummers, die bedoeld zijn voor het weergeven van de tijd, dan wordt een achtergrondkleur *led_bg* in het schuifregister geladen. Voor de tijdweergave worden 4 pixels gebruikt voor de uren en 6 pixels voor de weergave van de minuten en van de seconden. Omdat het aflezen in binair formaat wel wat oefening vereist, is voor elke weergave een hele rij van de LED-matrix gereserveerd. Welke rij dat is, is vastgelegd met constanten (een voorbeeld: *constant c_hrs_idx: natural := 15;*).

De weergave van de uren begint dus in de derde rij van de LED-matrix (twee rijen blijven vrij). Als de waarde van een bit van de kloktijd 1 is, wordt geel (*led_on*) in het schuifregister geladen, anders is de LED uit (*led_off*). De resterende LED's 5...8 of 7...8 van een rij op het display zijn altijd uit.

Om het vergelijken van het actuele pixelnummer goed te laten werken, moeten we letten op de verschillende datatypen die bij de vergelijking betrokken zijn. De pixelcounter (*led_cnt*) heeft als uitgangssignaal een signaal van het type *unsigned*, de constanten waarmee wordt vergeleken zijn van het type *natural*. Die vergelijking is alleen toegestaan, als we de functie *to_integer* uit de bibliotheek *Numeric_std* gebruiken (... *if (to_integer(led_idx) = c_hrs_idx +0) then ...*). Een handig overzicht van de in VHDL ingebouwde typeconversies is bijvoorbeeld onder [5] en [6] te vinden.

Toestandsmachine

De verdere besturing neemt de module *frame_ctrl_sm* voor zijn rekening. Dit is een toestandsmachine. Een toestandsmachine (*Finite State Machine*, FSM, is een sequentieel bestuurde logische schakeling die wordt opgezet voor het besturen van terugkerende taken en reeksen van handelingen. Er zijn nauwelijks grote digitale hardwaresystemen die niet met zo'n FSM werken. Een toestandsmachine bestaat uit een sequentieel deel, een reeks van registers die de actuele toestand opslaan, en uit één of twee combinatorische delen, die uit de actuele toestand de volgende toestand en de waarden op uitgangen berekenen. Afhankelijk van toepassing kan het nuttig zijn om de ingangen van te voren te synchroniseren en de uitgangen tussentijds op te slaan in een register. Er wordt onderscheid gemaakt tussen twee soorten toestandsmachines: de Moore- en de Mealy-FSM. Het verschil tussen die twee typen zit hem in de waarheidstabellen en in de toestandsovergangen. De uitgangen van een Moore-machine zijn alleen een functie van de actuele toestand, maar bij een Mealy-machine zijn de uitgangen een functie van de actuele toestand en van de data op de ingang. De twee typen zijn met behulp van Booleaanse algebra in elkaar om te zetten. In de praktijk maakt (naar oordeel van de auteur) dit onderscheid geen verschil, omdat het toch vooral zal gaan om het oplossen van een concreet probleem. De ervaring leert dat een Moore-implementatie meer toestanden (= mo(o)re) heeft dan een Mealy-implementatie, maar meestal wel gemakkelijker te begrijpen is.

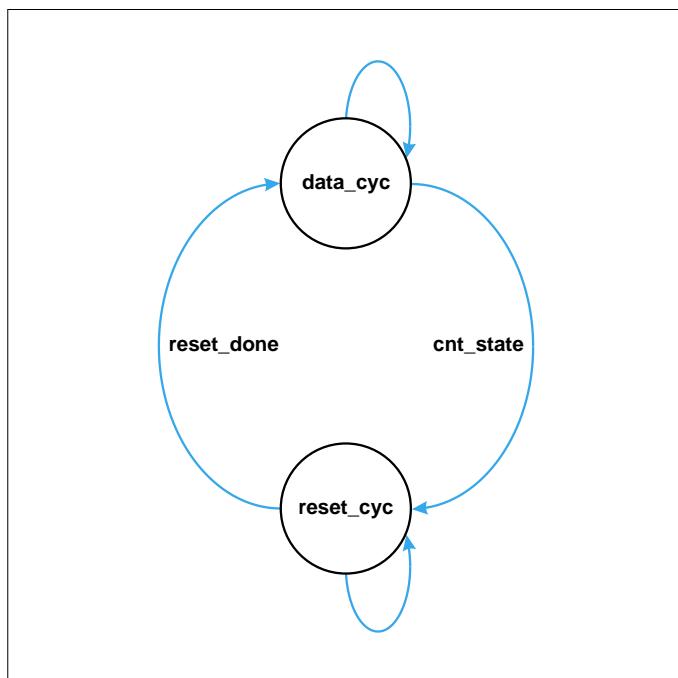
Er zijn verschillende manieren om een state machine te beschrijven in VHDL. De eerste is de methode met twee processen – volgens het boekje. De state machine wordt daarbij beschreven in twee gescheiden processen. Een proces is alleen verantwoordelijk voor het opslaan van de nieuw berekende toestandsvector; dat is de beschrijving van het sequentiële deel van de machine. Het tweede proces is de combinatorische logica, die nodig is voor het berekenen van de volgende toestand.

Daarnaast is er een aanpak waarbij de sequentiële en combinatorische delen, die nodig zijn voor het berekenen en opslaan van de toestand, binnen één proces worden beschreven en de resterende logica voor de uitgangen in een tweede proces (als dat nodig is). Deze aanpak heeft het voordeel, dat er minder geschreven hoeft te worden, waardoor de oplossing ook compact en eenvoudig te begrijpen is.

In **listing 4** zien we de sourcecode van de state machine, die de besturing van een frame bewaakt in onze driver. De state machine heeft maar twee toestanden. Die geven aan of op dit moment een pixel wordt uitgevoerd, of dat de reset-code wordt gestuurd. De toestanden van een FSM worden in de regel gedocumenteerd met behulp van toestandsdiagrammen. We kunnen in VHDL met het sleutelwoord **type** een *User-defined Type* aanmaken voor het coderen van de individuele toestanden. Met **type state_type is (data_cyc,reset_cyc);**

maken we een nieuw type met de naam **state_type** op basis van het VHDL-standaarddatatype *Enumeration*.

Het toestandsdiagram van de state machine voor een frame is te zien in **figuur 3**. Met Modelsim en Quartus kunnen we ook een toestandsdiagram weergeven, mits de state machine als zodanig wordt herkend. Als u bij Modelsim bij het vertalen van het VHDL-bestand de *Command Line Switch +cover* meegeeft, worden de meeste state machines herkend. Bij Quartus lukt dat alleen als we ons ook aan de voorgeschreven codeerstijl houden. De state machine van de WS2812-driver wordt echter toch niet herkend (vermoedelijk omdat hij maar twee toestanden heeft).



Figuur 3: Toestandsmachine voor het besturen van een frame.



SamacSys
ELECTRONIC COMPONENT LIBRARY SOLUTIONS

GRATIS downloaden

- Schematische symbolen
- PCB-voetafdrukken
- 3D-modellen

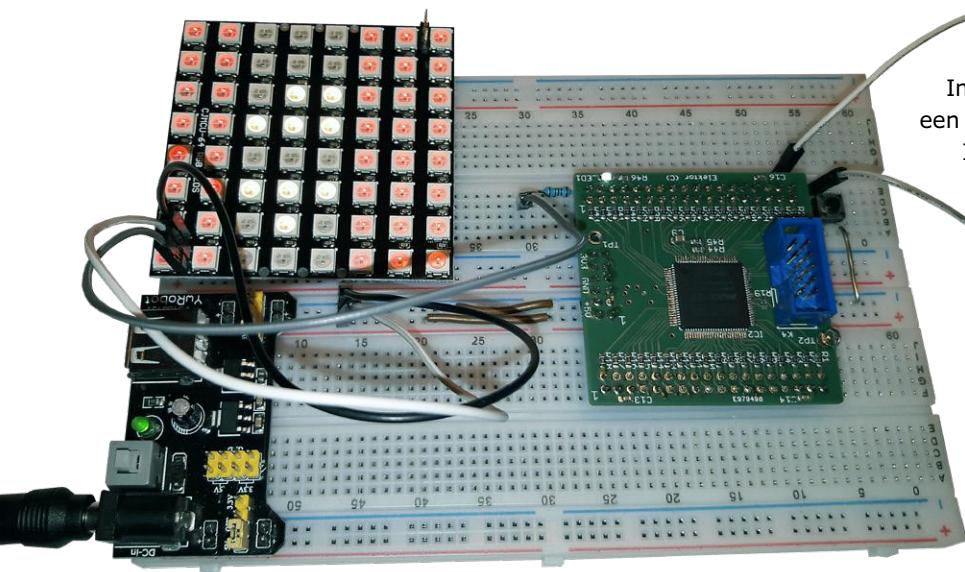
Klik & plaats in uw
ECAD-tool

De Nieuwste Producten voor Uw Nieuwste Ontwerpen™

nl.mouser.com/ecad

Helderheid

Een speurtocht op internet bracht aan het licht, dat één LED bij de grootste helderheid een stroom van maximaal 20 mA kan trekken. Met 64 x 3 LED's komen we dan toch al uit op een totaal van 3,9 A. Dat is genoeg om een kleine netspanningsadapter van een breadboard op de knieën te krijgen. Bij het ontwikkelen van de driver heeft de auteur daarom in het begin gewerkt met een aparte netvoeding die voldoende stroom kon leveren. Afgezien van het veel te grote stroomverbruik zijn de LED's, als ze op volle helderheid branden, te fel voor een tijdweergave. De helderheidswaarde moet dus worden gereduceerd. Hoe fel de LED's moeten oplichten en met welke kleur, kan in de module *clock_to_pixel* worden ingesteld met behulp van constanten. Daarvoor wordt een eigen *Array-Type* aangemaakt met std_logic-vectoren.



Figuur 4: Klok met 8x8-display.

Zo kan voor de verschillende kleuren die moeten worden weergegeven steeds een aparte constante worden aangemaakt.

```
type grb_lut is array (0 to 2) of std_logic_vector(7
  downto 0);
constant led_off: grb_lut:= (x"00", x"00", x"00");
-- all off
```

```
constant led_bg: grb_lut:= (x"00", x"10", x"00");
-- decent red
constant led_on : grb_lut:= (x"1F", x"1F", x"1F");
-- decent yellow
```

De standaard gekozen waarde van 31 (= 0x1F) is ruim voldoende. En dan is ook de uitgangsstroom van 700 mA van de breadboard-netvoeding voldoende (zie **figuur 4**). De seriële datalijn van de LED-matrix wordt rechtstreeks verbonden met de uitgang van de CPLD (PIN 54 van de CPLD = pin 5 op K1). Volgens de datasheet van de LED-matrix is dit strikt genomen niet toelaatbaar, omdat de minimale ingangsspanning voor een hoog niveau $0,7 * VDD = 3,5$ V moet zijn, wat ongeveer 200 mV meer is dan de 3,3 V van de LVTTL-uitgang. Afhankelijk van welke LED-matrix u gebruikt, kan het dus nodig zijn om een level-shifter toe te voegen.

Vooruitblik

In dit deel van de serie hebben we voor het eerst een wat grotere digitale schakeling bekeken.

In de 'Fitter Resource Usage Summary' in het *Quartus-Compilation-Report*-venster is te zien, dat daarmee de CPLD voor 36 % is gevuld. Er is dus nog wat plaats voor uitbreidingen. In de volgende aflevering gaan we in het kort de *Timing Constraints* behandelen die nodig zijn voor een werkende schakeling. Bovendien willen we een klein spel in hardware programmeren. ▶

180285-C-03

Weblinks

- [1] Aansturing van de WS2812: www.mikrocontroller.net/articles/WS2812_Ansteuerung
- [2] ESP8266-webserver voor NeoPixel-LED-strips, Elektorlabs september/oktober 2018: www.elektormagazine.nl/180355-03
- [3] WS2812-datasheet: <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [4] Hardware-ontwerp met (V)HDL (1), Elektorlabs september/oktober 2018: www.elektormagazine.nl/180285-04
- [5] VHDL-typeconversie: www.nandland.com/vhdl/tips/tip-convert-numeric-std-logic-vector-to-integer.html
- [6] VHDL-vectorwiskunde: www.doulos.com/knowhow/vhdl_designers_guide/numeric_std/
- [7] Webpagina bij dit artikel: www.elektormagazine.de/180285-C-01
- [8] Hardware-ontwerp met (V)HDL (2), Elektorlabs november/december 2018: www.elektormagazine.nl/180285-B-03

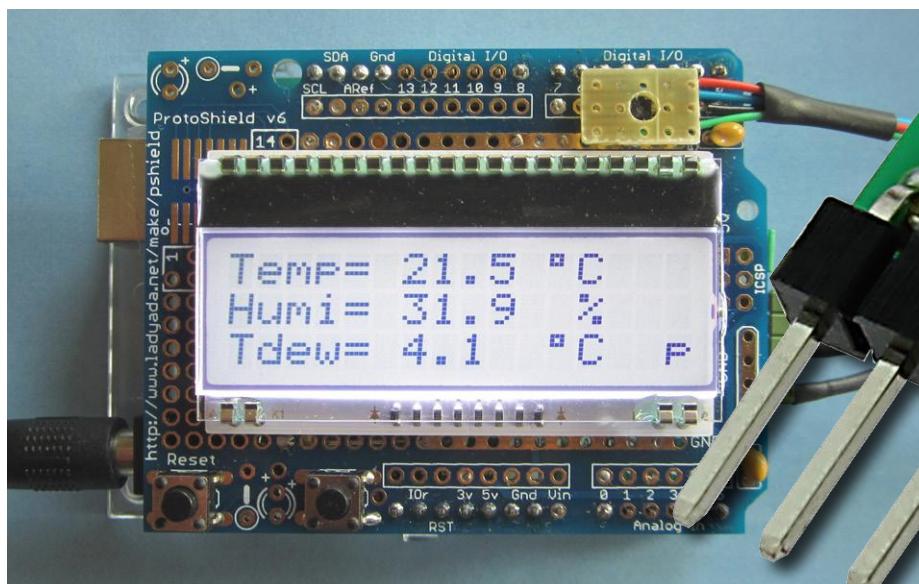


IN DE STORE

→ **CPLD Breakout Board**
www.elektor.nl/cpld-breakout-board-160425-91

eBoB voor ENS210-dauwpuntsensor

met voorbeeld voor de Arduino UNO



Roland Stiglmayr (Duitsland) en Luc Lemmens (Elektor Lab)

Het meten van de fysische grootheid met de naam ‘dauwpunt’ is niet alleen van belang om het risico van schimmelgroei maar ook van corrosie van stalen objecten zoals gereedschap te kunnen inschatten. Door te ventileren en te verwarmen kan dit tot op zekere hoogte worden beperkt, maar om hierop te kunnen anticiperen moet het dauwpunt bekend zijn. En dat is precies wat dit slimme apparaatje doet.

Specificaties

- Gemakkelijk toepasbaar breakout board (BoB)
- Zeer nauwkeurige en zeer kleine temperatuur- en vochtigheidssensor
- Voorbeeldtoepassing voor Arduino

Vergist u zich niet, het betreft hier een zeer nauwkeurig meetinstrument. Door de hoge nauwkeurigheid van de temperatuur- en vochtigheidsmetingen is het voor veel toepassingen bruikbaar. De gebruikte sensor, de ENS210 van a m s, is op dit moment waarschijnlijk de nauwkeurigste voorgekalibreerde sensor in de

markt. De temperatuur wordt gemeten met een nauwkeurigheid van $\pm 0,2^{\circ}\text{C}$ in een gebied van 0°C tot 70°C zonder dat enige kalibratie nodig is. Hierdoor is de ENS210 bruikbaar als referentie bij het kalibreren van temperatuur- en vochtigheidssensoren.

Wat is dat dauwpunt eigenlijk?

Het dauwpunt wordt berekend uit de gemeten waarden van temperatuur en relatieve luchtvochtigheid. Hiervoor wordt de ‘Magnus’-dauwpuntbenadering gebruikt; lezers die geïnteresseerd zijn in de wiskundige en theoretische achtergronden hiervan moeten beslist dit Wikipedia-lemma lezen [1]. Vervolgens

wordt het verschil tussen het dauwpunt en de actuele temperatuur bepaald, wat een uitstekende indicatie is voor het risico op schimmelgroei en corrosie.

Waarschuwing: kleine onderdelen!

Uw enthousiasme om met de nieuwste IC’s aan de slag te gaan wordt wellicht getemperd door hun extreem kleine afmetingen, volgens de industrie ‘omdat de klant dat zo wil’. De ENS210 is hierop geen uitzondering met zijn behuizing van slechts $2 \times 2 \text{ mm}$ (**figuur 1**). Gelukkig, zoals te zien in **figuur 2**, is het de auteur met een beetje moeite en vernuft gelukt het kleine chipje op een stukje experi-

PROJECT-INFO



Meten
Dauwpunt
Arduino BoB



beginners
→ gevorderden
experts



ca. 30 minuten



Arduino UNO,
IDE,
soldeerbout



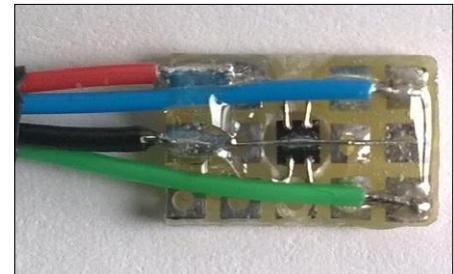
ca. € 15

menteerprint (gaatjes- of strokenprint of iets dergelijks) te solderen.

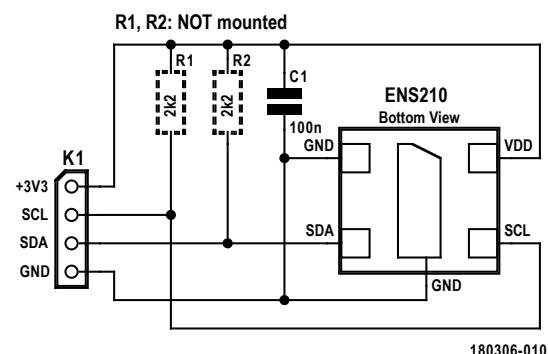
Frustratie over de onmogelijk kleine hedendaagse IC's is één ding, wanhoop en razernij ze überhaupt te kunnen hanteren is erger. Daarom heeft Elektor Labs, zoals u waarschijnlijk al had verwacht, een breakout board (eBoB) ontworpen voor deze voortreffelijke sensor. Onze eBoB voorkomt dat de ENS210



Figuur 1: Klein, kleiner, kleinste... de ENS210 dauwpuntsensor van a m s is echt heel klein. (onderzijde; foto van de auteur)



Figuur 2: Een geïmproviseerde, maar niettemin succesvolle methode om de ENS210 op experimenteerprint te monteren (foto van de auteur).



Figuur 3: Schema van de ENS210 eBoB. De I²C-pull ups R1 (SCL) en R2 (SDA) zijn optioneel.

kapot gaat, wordt gebraden of zelfs in de stofzuiger verdwijnt, en maakt de meetwaarden prima toegankelijk.

den R1 en R2 voor de SDA- en SCL-lijnen van de I²C-interface. R1 en R2 worden gevoed vanuit 3,3 V en moeten worden gemonteerd als:

- op de SDA- en SCL-lijnen van het systeem geen andere pull up-weerstanden zijn gemonteerd;
- spanningsbegrenzing is vereist vanwege componenten op de I²C-bus die op 5 V werken, zoals een ATMEGA met



ONDERDELENLIJST

Weerstanden:

R1,R2 = zie tekst

Condensatoren:

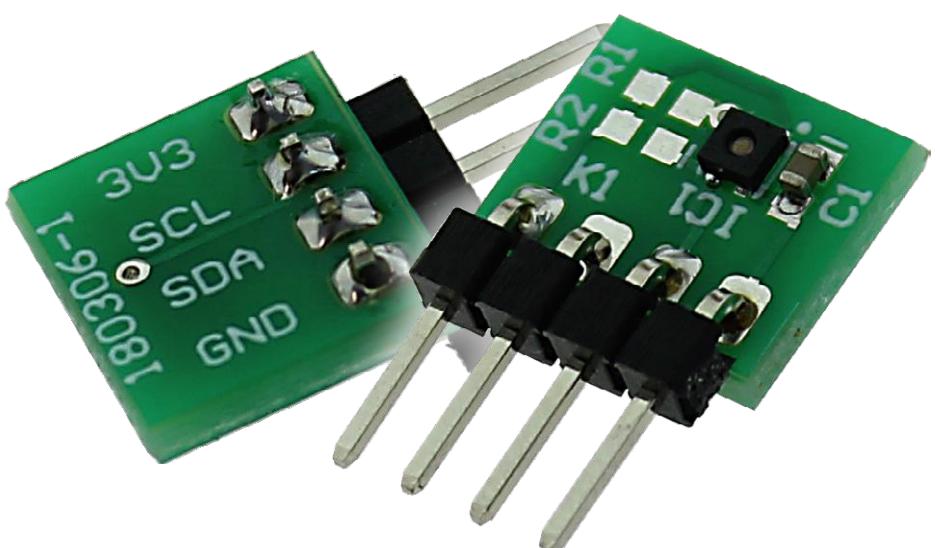
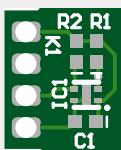
C1 = 100 nF, X7R, 0603

Halfgeleiders:

IC1 = ENS210-LQFM, (a m s), bijv. Mouser # 985-ENS210-LQFM

Diversen:

K1 = 4-polige SIL-pinheader, steek 0,1"
Print 180306-1 v1.0, Elektor-shop



zijn interne pull ups, met als resultaat een veilig hoog-niveau van 3,6 V op de SDA- en SCL-pinnen van de sensor.

De sensor mag onder geen beding zonder pull ups op SDA en SCL worden gebruikt. Uiteindelijk vindt alle dataverkeer en voeding (3,3 V) plaats via de vierpolige pinheader K1.

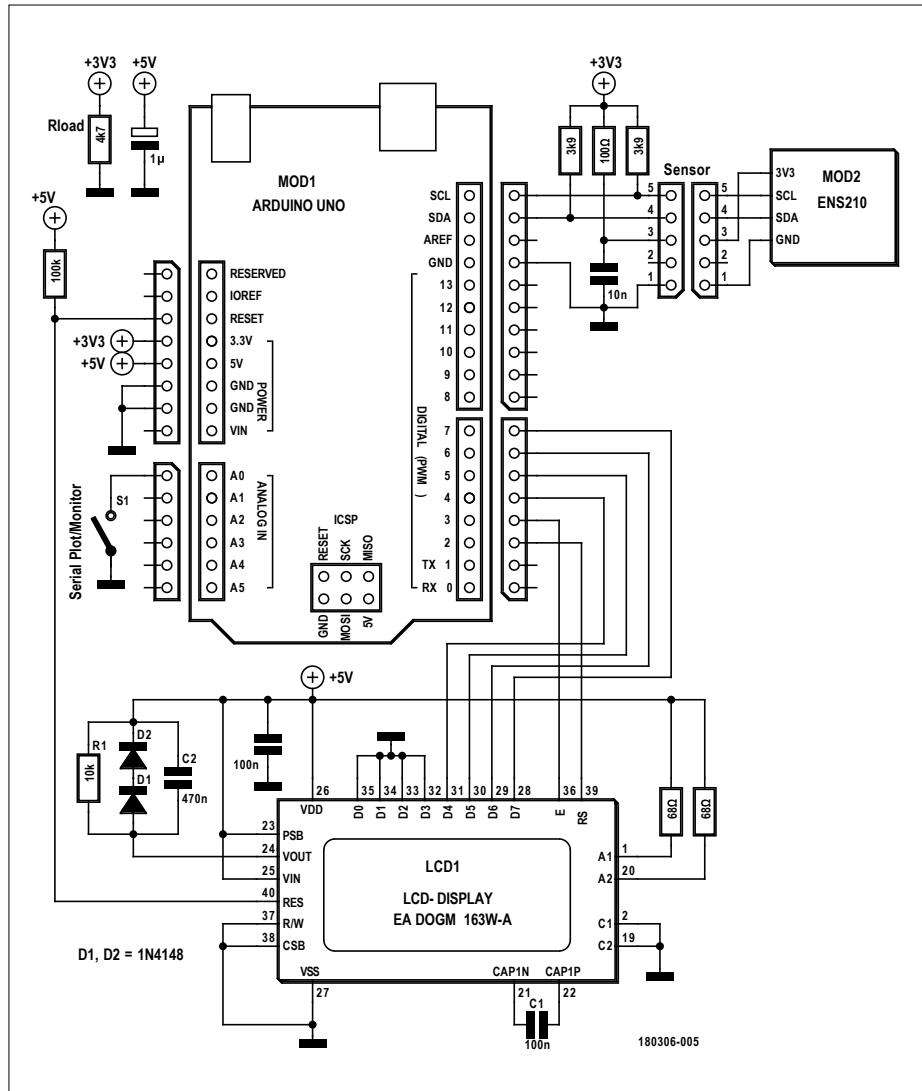
Combinatie met een Arduino

Om een praktische dauwpuntmeter te realiseren heeft de auteur zijn ENS210 aan een Arduino Uno geknoopt en het resultaat op de Elektor Labs-website gepubliceerd. Dankzij de toepassing van een Uno is de voor dit project benodigde hardware bijzonder simpel, met nog slechts één uitdaging: het aansluiten van een LC-display. De makkelijkste oplossing is om een Proto-shield te gebruiken, dat een connector heeft om de sensor via een kabel aan te sluiten.

Figuur 4 toont het schema van de dauwpuntmeter van de auteur, voorzien van nog een paar toeters en bellen, ontworpen voor de door hem geschreven software die het dauwpunt direct op het display toont. Het programma, of ‘sketch’ in Arduino-jargon, kan gratis worden gedownload [1]. Merk op dat deze schakeling en de Arduino-software niet zijn gekoppeld aan een door Elektor Labs ontworpen print, zodat dit deel van het artikel een puur **Homelab-project** is.

De ENS210 biedt een volledige I²C-interface, waardoor vrijwel geen andere hardware nodig is. Alle meetresultaten worden op het display getoond, maar daarnaast ook naar de seriële monitor of plotter van de Arduino IDE gestuurd. Door middel van een drukknop kan het seriële output-formaat worden geselecteerd voor monitor of plotter.

De auteur koos voor een fraai drieregelig LC-display met witte achtergrondverlichting, maar u kunt ook een ander type nemen. Dit 3-regelig display wordt ondersteund door de sketch ‘ENS210_3L_34’. Met slechts enkele aanvullingen kunnen grote gedeelten van de standaard ‘LiquidCrystal’-library worden gebruikt voor verdere uitbreidingen, zonder dat de library zelf hoeft te worden aangepast. Natuurlijk kan ook worden gebruikgemaakt van het Elektor LCD-shield in combinatie met de sketch ‘ENS210_2L_25’. Een andere suggestie voor uitbreiding van dit project zijn enkele drukknoppen waarmee een datalogger-functie kan worden bestuurd. Deze functie kan de



Figuur 4: Ontwerp van de auteur voor een Arduino ‘host’-systeem voor de ENS210 dauwpuntsensor. Door deze twee met een LCD te combineren ontstaat een interessant instrument om problemen met schimmelgroei en corrosie tijdig op het spoor te komen. Als de boost-functie van het LCD niet nodig is, kunnen C1, C2, D1, D2 en R1 worden weggelaten; da moet pin 24 worden doorverbonden met pin 25

meetdata niet-vluchtig opslaan in het EEPROM-geheugen.

MOD2 in het schema is de door de auteur gemaakte sensor-print zoals afgebeeld in figuur 2, dus niet de Elektor eBoB, die

een andere pinout heeft. De aanpassing is echter simpel, want het gaat slechts om de +3V3, SCL, SDA en GND tussen BoB en de UNO-host. ▀

180306-03



IN DE STORE

- Dauwpuntmeter e-BoB, opgebouwde print
www.elektor.nl/180306-1

Weblinks

- [1] Dauwpunt, theoretische achtergrond: <https://nl.wikipedia.org/wiki/Dauwpunt>
- [2] Arduino-sketches: <http://www.elektormagazine.nl/180306-03>



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

Met een fraaie behuizing op maat krijgt uw elektronica-project een professionele uitstraling. Vroeger was dat een lastige klus maar tegenwoordig is dit een stuk toegankelijker geworden.

Ilse Joostens (België)

V Welke materialen kunnen we gebruiken voor het ontwerpen van behuizingen?

A Er is een heel breed scala aan bruikbare materialen voor behuizingen. In de context van deze V&A beperken we ons tot de meer gangbare, makkelijk verkrijgbare en goed te bewerken materialen.

- **PMMA, polymethylmetacrylaat (acrylaat, plexiglas, perspex)**

PMMA is verkrijgbaar in een groot aantal kleuren en diktes en dit zowel transparant, doorschijnend als ondoorzichtig. Belangrijk om te weten is dat PMMA beschikbaar is in twee varianten, gegoten (GS) en geëxtrudeerd (XT). Wat het meest geschikt is hangt af van de toepassing en de gekozen bewerkingsmethode.

In tegenstelling tot geëxtrudeerd PMMA dat tussen rollen op dikte wordt gewalst, zitten er bij gegoten PMMA behoorlijke toleranties op de dikte tot wel +/- 15%.

Als de mechanische afmetingen en passvorm voor een ontwerp belangrijk zijn dient hier terdege rekening mee gehouden te worden en geniet geëxtrudeerd PMMA de voorkeur.

Gegoten PMMA is makkelijker mechanisch te bewerken (boren, zagen, frezen,...) en geeft mooiere resultaten bij lasergraveren (frost-look). Voor lasersnijden is naar onze ervaring geëxtrudeerd PMMA door het lagere smeltpunt beter geschikt. De snijkanten zijn mooier met een vlamgepolijst effect terwijl gegoten PMMA dikwijls wat ruwere snijkanten geeft. [1]

- **Polycarbonaat (Lexan)**

Polycarbonaat is net als PMMA in verschillende diktes verkrijgbaar maar met een beperkter aanbod aan kleuren. Het is erg stevig en slagvast en 250 maal sterker dan glas. Ondanks de stevigheid is het

goed mechanisch te bewerken waarbij er, vergeleken met PMMA, minder kans op barsten bestaat.

Voor lasersnijden is het minder geschikt omdat dan lelijke snijranden ontstaan. Polycarbonaat is erg geschikt als beschermvenster voor displays en is in die toepassing ook krasvaster dan PMMA.

- **Polyvinylchloride (PVC, Vikupor)**

PVC-platen zijn beschikbaar als harde PVC-platen en schuimplaten in diverse diktes en kleuren. De harde PVC-platen zijn erg vormvast en hebben een hoge chemische bestendigheid. Lasergraveren en -snijden zijn een absolute no-go bij dit materiaal omdat dan waterstofchloride vrijkomt, een sterk irriterend en giftig gas, wat in contact met water zoutzuur vormt en de lasersnijmachine aantast. Mechanische bewerkingen zijn wel probleemloos mogelijk.

Omdat PVC vlamvertragend en zelfdovend is, is het een zeer geschikt materiaal voor projecten met netspanning en/of vermogenselektronica.

- **Multiplex**

Multiplex is een verzamelnaam voor gelaagd plaatmateriaal uit diverse

houtsoorten. Het is steviger dan MDF, maar de dunne platen hebben de neiging om krom te trekken bij schommelingen in de luchtvochtigheid en temperatuur. Het materiaal is eenvoudig mechanisch te bewerken en als de dikte beperkt blijft, is ook lasersnijden en -graveren mogelijk. Dikker (>10 mm) multiplex lasersnijden is enkel mogelijk op lasersystemen met een groter vermogen, omdat anders de snijranden verkolen. Voor lasersnijden en -graveren is vooral multiplex uit "lichtere" houtsoorten zoals berk of populier aangewezen (**figuur 1**). Multiplex uit tropisch hardhout geeft eerder teleurstellende resultaten.

- **MDF (Medium-Density Fibreboard)**

MDF is zeer makkelijk mechanisch te bewerken en is mechanisch stabiel (trekt niet krom). Lasersnijden is mogelijk bij dunner materiaal, maar omdat de snijranden er donkerbruin tot zwart uitzien vindt niet iedereen dat even mooi (**figuur 2**).

- **Aluminium**

Aluminium kan alleen mechanisch bewerkt worden, laserbewerkingen zijn beperkt tot het markeren van geanodi-



Figuur 1. Dun multiplex leent zich goed voor lasersnijden.



Figuur 2. Het snijden van hout en MDF met een laser levert bruine randen op.

seerd aluminium of blank aluminium in combinatie met een chemisch product. Voor het makkelijk verzagen en (CNC) frezen is het aan te raden om een legering met goede verspaanbaarheid te kiezen. Goed verkrijgbare geschikte aluminiumlegeringen zijn bijvoorbeeld 6061 T4 en T6 of 6082 T6.

• Graveerbare materialen

(Rowmark, Lasermark, Gravoply, ...)

Deze materialen zijn meestal vrij dun en bestaan uit twee (of meer) lagen van een verschillend gekleurd materiaal. Door het materiaal van de toplaag weg te laseren of te frezen, komt het anderskleurige materiaal van de onderlaag tevoorschijn. Op deze manier kunnen makkelijk mooie frontplaten gemaakt worden (**figuur 3**).

V Hoe kan ik materialen bewerken?

A Vroeger waren, tenzij men over veel geld beschikte, de opties redelijk beperkt tot de meer klassieke middelen zoals boren, zagen, vijlen, ... Nu zijn de meer geavanceerde technieken een stuk toegankelijker geworden. Uiteraard is het ook mogelijk om meerdere technieken te combineren in één ontwerp. Een overzicht:

• 3D-printing

3D-printers zijn ondertussen al vele jaren erg populair en dikwijls de eerste keuze wanneer een behuizing op maat dient gemaakt te worden. 3D-printing is ook erg toegankelijk, omdat nogal wat printers aantrekkelijk geprijsd zijn en er voldoende gratis open source software te vinden is. Een nadeel is dat de techniek te traag is voor de productie van grotere aantallen behuizingen en dat de kosten voor filament of UV-hars kunnen oplopen. Daar tegenover staat dat zeer complexe vormen kunnen gemaakt worden.

• CNC

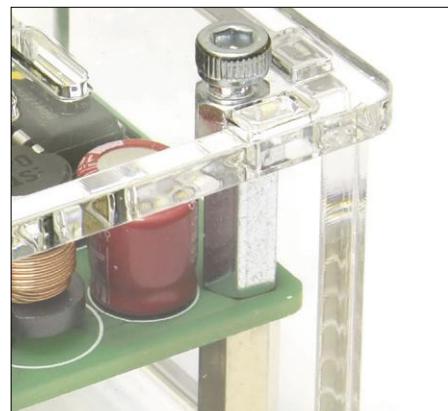
Net als 3D-printers worden meer en meer betaalbare CNC-freesmachines, al dan niet als bouwpakket, aangeboden. De betere machines kunnen een veelvoud van materialen bewerken, maar de leercurve is hier een stuk steiler in vergelijking met 3D-printing. Als tussenoplossing zijn ook goedkope, kleine, manueel bedienende freesmachines te koop. Deze zijn ideaal om af en toe eens een rechthoekig gat in een frontpaneel of bestaande behuizing te frezen.

• Lasersnijden

We hebben het hier voor alle duidelijkheid over CO₂-lasersystemen en niet over andere systemen zoals fiberlasers



Figuur 3. Tweekleurige gelaagd materiaal is heel geschikt om frontplaten mee te maken.



Figuur 4. Makkelijk: tabs, afstandsbusjes, bouten en moeren houden de boel bij elkaar.

en gecombineerde systemen met een zeer hoog prijskaartje. In tegenstelling tot 3D-printers en CNC-freesmachines blijven de prijzen voor lasersnijmachines vrij hoog. Naast de professionele productiemachines met metalen RF-laserbuis zijn er ook budgetlasersystemen met een glazen laserbuis beschikbaar voor thuisgebruik. Wie de prijzen hiervan nog te hoog vindt en wat avontuurlijk is aangelegd, kan ook overwegen om een Chinese lasersnijmachine aan te schaffen. Tot slot kunt u lasersnijwerk ook uitbesteden, waarbij u uw ontwerp instuurt bij een bedrijf dat dan uw onderdelen gaat produceren. Dit is trouwens ook mogelijk voor CNC-frezen en 3D-printen. U kunt ook altijd eens langs gaan bij een FabLab of Hackerspace.

Nog een tip: rastergraveren met een lasersnijmachine gaat erg traag, als snelheid belangrijk is, probeer dan zoveel mogelijk vectorieel (met lijnen) te graveren.

• Thermovormen

Thermoplastische kunststoffen kunnen in diverse vormen gebogen worden door het materiaal (plaatselijk) te verwarmen zodat het zacht wordt. Dit kan met hete lucht, een gloeidraad, maar ook met een lasersnijder. Hierbij wordt het materiaal opgewarmd met een snel bewegende laserstraal uit focus waarna de gravitatie zijn werk kan doen (laser origami). Dat laatste is misschien niet zo praktisch, maar het is wel een leuk idee.

Bij vacuümformen wordt een vel voorverwarmde kunststof over een driedimensionale mal getrokken met behulp van onderdruk. [2]

V Welke software kan ik gebruiken?

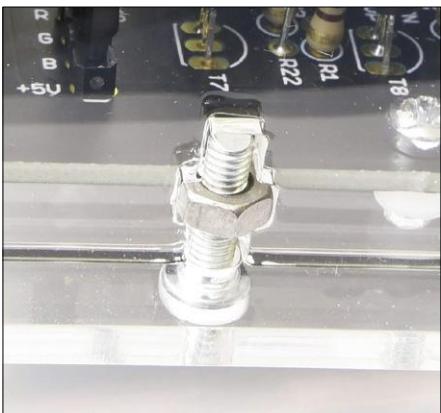
A Voor 3D-printing zijn tal van softwarepakketten beschikbaar, al dan niet gratis en/of open source. Voorbeelden zijn Blender, SketchUp, SolidWorks, Fusion 360, Inventor, TinkerCAD, etc. Een aantal van deze pakketten zijn ook bruikbaar voor 3D CNC-frezen, maar dan moet de CNC-machine hier ook geschikt voor zijn. Veel goedkopere CNC-freesmachines ondersteunen alleen 2,5D (met 2D-lagen een 3D-structuur bouwen) in plaats van echte 3D. Dikwijls is al dat 3D-geweld niet meteen noodzakelijk en kunt u ook gebruik maken van 2D CAD-programma's en (vectoriële) tekenprogramma's zoals DraftSight, AutoCAD LT, InkScape, CorelDraw of Adobe Illustrator. 3D-bewerkingen op een CNC-machine kosten net als 3D-printen veel tijd. Een driedimensionale behuizing samenstellen met een eenvoudige "2D-onderdelen" gaat nu eenmaal sneller en is ook makkelijker in aantallen te realiseren. Door materialen in lagen toe te passen kunt u ook complexe behuizingen maken met een professionele uitstraling.

V Hoe kan ik diverse onderdelen voor een behuizing verbinden?

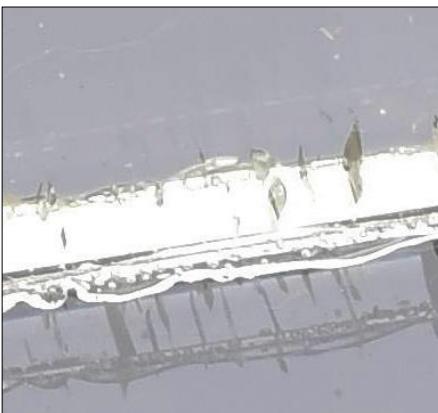
A De meest gangbare methoden zijn bevestigingsmaterialen, klemmen, kliksystemen en lijmen.

• Bevestigingsmaterialen

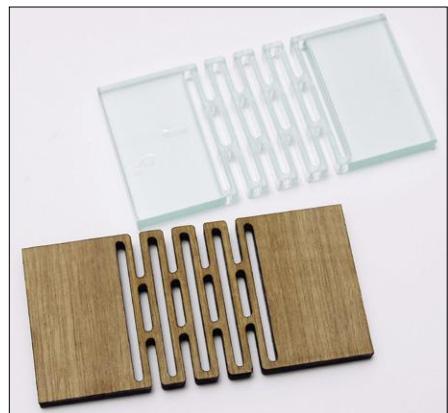
Hieronder verstaan we onder andere schroeven, moeren, bouten, afstandshouders, afstandsbusjes en verbindingsstukken. Hoewel deze montage-methode extra kosten met zich meebrengt voor de bevestigingsmaterialen heeft deze het voordeel dat de behuizing, eenmaal gemonteerd, later eenvoudig uit elkaar genomen kan wor-



Figuur 5. Een T-slot maakt bout-moer verbindingen eenvoudig.



Figuur 6. Om scheurtjes te voorkomen moet PMMA na lasersnijden eerst „ontspannen“ worden alvorens het te verlijmen.



Figuur 7. Living hinges kunnen ondermeer met hout en PMMA gemaakt worden.

den mocht dat nodig zijn. De meest eenvoudige methode is om de behuizing samen te houden met een combinatie van tabs, afstandshouders en bouten (**figuur 4**). Voor haakse verbindingen kan een "T-slot" (**figuur 5**, [3][4]) voorzien worden, waarna een ander stuk plaatmateriaal er onder een hoek van 90 graden tegen geschroefd kan worden. Hier is het belangrijk om de bout niet te stevig aan te draaien.

• Klemmen

Klemverbindingen [5] kunnen gemaakt worden met tabs en uitsparingen of met getande verbindingen. Het idee is om de tab of een deel ervan iets groter te maken dan het gat waar deze moet in passen zodat de verbinding spant. Bij getande verbindingen dient er een miniem verschil te zijn in afmeting tussen de tanden en de uitsparingen waarin de tanden vallen. Klemverbindingen kunnen zeer stevig vastzitten en ze opnieuw losmaken kan moeilijk zijn zonder schade te veroorzaken. Daarnaast bestaat ook het risico dat ze spontaan loskomen.

• Kliksystemen

Hier worden getande tabs gebruikt die, doordat het materiaal wat kan buigen, kunnen bewegen. De tab wordt dan gebogen en door een gat geforceerd, waarna het getande deel vastklikt nadat het door het gat komt. [6] Dit is een kwetsbaar systeem, vooral wan- neer breekbare materialen zoals PMMA gebruikt worden en het leent zich dan ook niet voor veelvuldige montage en demontage.

• Verlijmen

Lijnverbindingen kunnen zeer stevig zijn, zelfs even stevig als het verlijmde mate- riaal zelf. Dikwijls zijn speciale producten nodig en soms ook de beheersing van specifieke technieken. Dit soort verbin- dingen is meestal definitief en soms krijgt men maar één kans om een onderdeel goed te bevestigen. Bij het lasersnijden van PMMA ontstaan spanningen in het materiaal welke bij verlijmen scheurtjes kunnen doen ontstaan (**figuur 6**). Om dit te voorkomen dient PMMA vóór het verlij- men eerst ontlaten te worden in een oven.

V Wat zijn living hinges?

Een "Living hinge" is een flexibele scharnier die twee starre delen verbindt en uit hetzelfde materiaal als de starre delen bestaat. Een voorbeeld is de scharnier van het dekseltje van een Tic Tac-doosje. Door middel van laser- snijden of frezen kunnen plaatmaterialen voorzien worden van een patroon van inkervingen, waardoor een gedeelte of de gehele plaat buigzaam en flexibel wordt (**figuur 7**). Dit is mogelijk omdat starre materialen nog steeds enige flexibiliteit bezitten. De toepassingen hiervoor zijn legio en omvatten naast scharnieren en flexibele verbindingen ook het creëren van ronde vormen. Daarbij kunnen des- gewenst ook kunstige patronen gebruikt worden voor de inkervingen. Bedenk wel dat het lasernijden of frezen van een living hinge de nodige tijd kan kosten en dat living hinges soms fragiel kunnen zijn. [7][8]

180685-D-02

Weblinks

- [1] Acrylaat spiekbriefje: <https://makezine.com/2015/10/29/skill-builder-acrylic/>
- [2] Video over thermovormen: www.youtube.com/watch?v=P8a2KrEOzxU
- [3] Acrylaatverbindingen: www.ponoko.com/blog/how-to-make/how-to-make-snug-joints-in-acrylic/
- [4] Schroefverbindingen: www.instructables.com/id/How-to-Make-Anything-Using-Acrylic-and-Machine-Sc/
- [5] Klemverbindingen: www.ponoko.com/blog/how-to-make/how-to-create-better-nodes/
- [6] Kliksystemen: www.defproc.co.uk/blog/2013/so-whats-a-practical-laser-cut-clip-size/
- [7] Living hinge op Wikipedia: https://en.wikipedia.org/wiki/Living_hinge
- [8] Meer over living hinges: www.rs-online.com/designspark/laser-cut-living-hinges-for-neater-designs



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

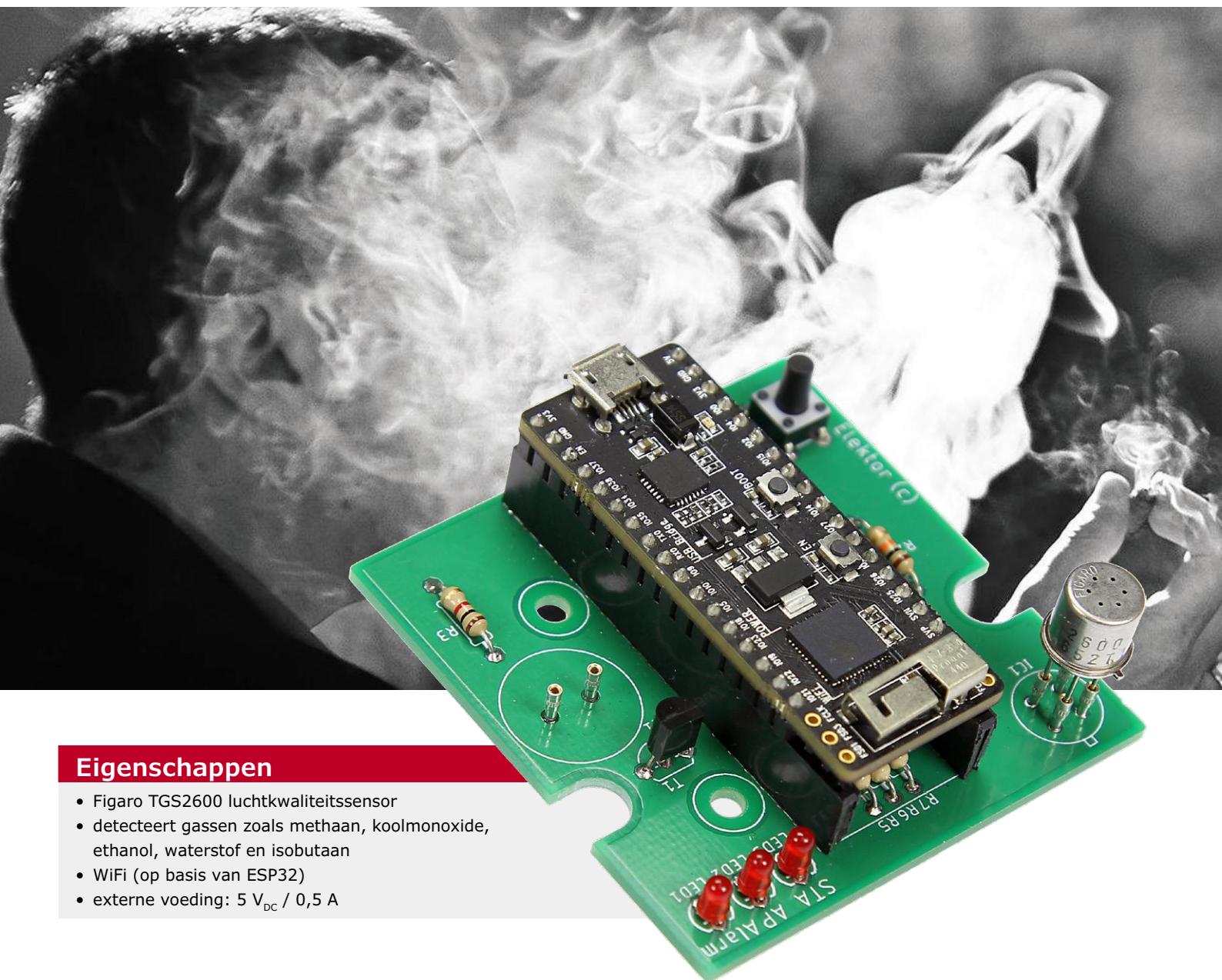


Luchtvervuilingsmonitor

rookt daar iemand op het toilet?

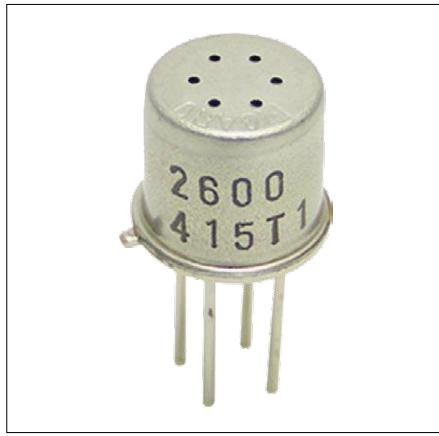
Bera Somnath (India) en Roy Aarts (Elektor Labs)

De kwaliteit van de lucht die we inademen zou ons allemaal zeer ter harte moeten gaan. We weten allemaal dat fijnstof in de lucht astmapatiënten de hele nacht wakker houdt, maar ook de aanwezigheid van bepaalde gassen kan erg schadelijk zijn. Dit project meet de aanwezigheid van vervuilende gassen in de lucht en laat een alarm klinken als de concentratie een bepaald niveau overschrijdt, bijvoorbeeld wanneer iemand een sigaret opsteekt...



Eigenschappen

- Figaro TGS2600 luchtkwaliteitssensor
- detecteert gassen zoals methaan, koolmonoxide, ethanol, waterstof en isobutaan
- WiFi (op basis van ESP32)
- externe voeding: 5 V_{DC} / 0,5 A



Figuur 1. De TGS2600-sensor van Figaro detecteert gassen zoals methaan, koolmonoxide, ethanol, waterstof en isobutaan.

De sensor die we in dit project gebruiken, een TGS2600 van de Japanse specialist in gassensoren Figaro, reageert bij lage concentraties van diverse gassen zoals methaan, koolmonoxide, ethanol, waterstof en isobutaan. Die laatste stof wordt vaak gebruikt als drijfgas in spuitbussen. De sensor kan trouwens geen onderscheid maken tussen deze gassen, hij detecteert alleen het effect ervan op de schone lucht. Verder is de TGS2600 niet echt een gas-sensor, het is eerder een gas-in-luchtsensor: hij werkt niet goed als het zuurstofniveau te laag is. Omdat de TGS2600 reageert op veranderingen in de luchtkwaliteit, is hij vooral goed in het detecteren van bijvoorbeeld sigarettenrook. Hoewel de TGS2600 een heleboel gassen detecteert, zijn er toch een paar waar hij niet zo blij mee is. Siliconendamp en sterk corrosieve omgevingen kunt u beter vermijden, net als water en vooral zee-water (maar ja, tegen zee-water is vrijwel niets bestand. Hetzelfde geldt voor kat-tenpis, maar dat is een ander verhaal).

Over de sensor...

De TGS2600 zit in een kleine metalen behuizing (**figuur 1**), en is gebaseerd op een resistief element van tindioxide (SnO_2). Om te kunnen reageren op veranderingen in gasconcentraties heeft hij verwarming nodig; gelukkig is een geschikte verwarming ingebouwd in de sensor. De sensor heeft na het inschakelen dan ook een paar seconden opwarm-tijd nodig voordat hij goed werkt. Als de sensor voor het eerst wordt ingeschakeld na een inactieve periode van enkele weken of langer, bedraagt de opstarttijd een dikke 15...20 minuten.

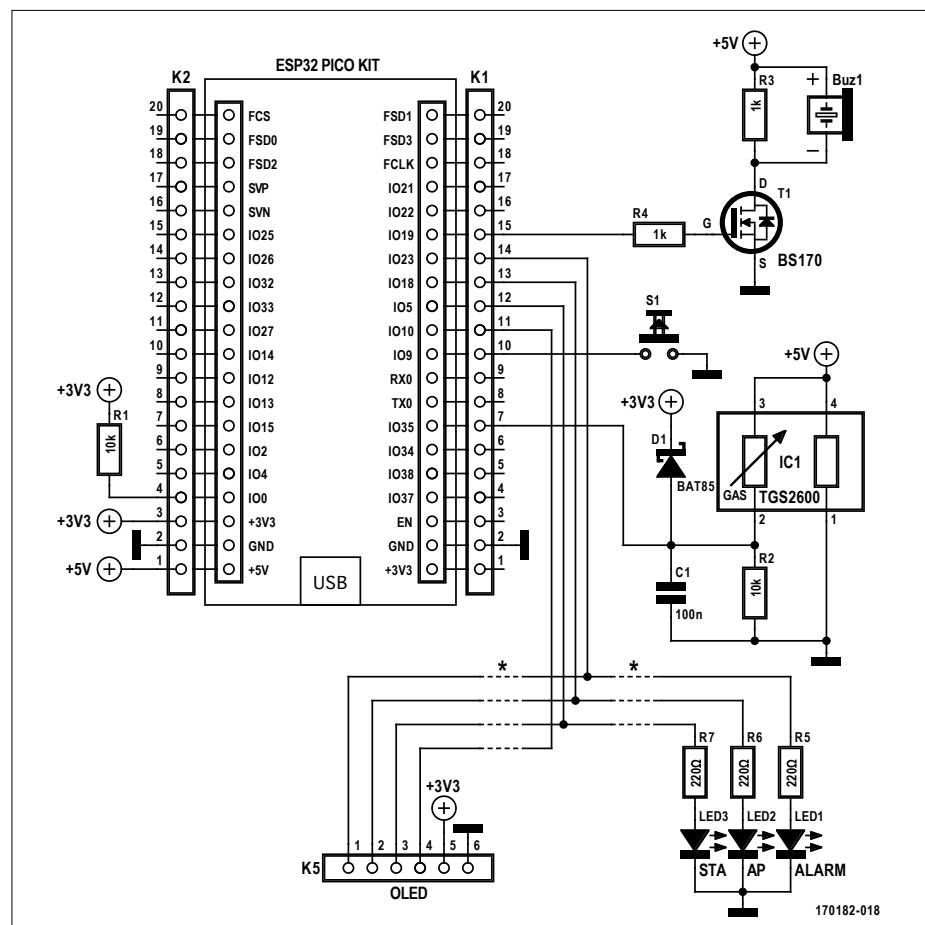
Het weerstandselement (R_{GAS} in IC1) in de sensor is eenvoudigweg opgenomen in de ‘hoge’ kant van een resistieve spanningsdeler. De weerstandswaarde wordt beïnvloed door de samenstelling van de omgevingslucht; het uitgangssignaal is de spanning over de ‘lage’ kant van de spanningsdeler, R_2 in het schema, die extern moet worden aangesloten. De weerstand van de sensor in schone lucht noemen we R_0 en dat is de referentiewaarde. De waarde van R_{GAS} is omgekeerd evenredig met de luchtverontreiniging. De omgevingstemperatuur en de relatieve luchtvochtigheid beïnvloeden de gevoeligheid van de sensor enigszins, maar zolang de sensor wordt gebruikt binnen een temperatuurbereik van +10...50 °C, is temperatuurcompensatie niet nodig.

...de bewakingsschakeling...

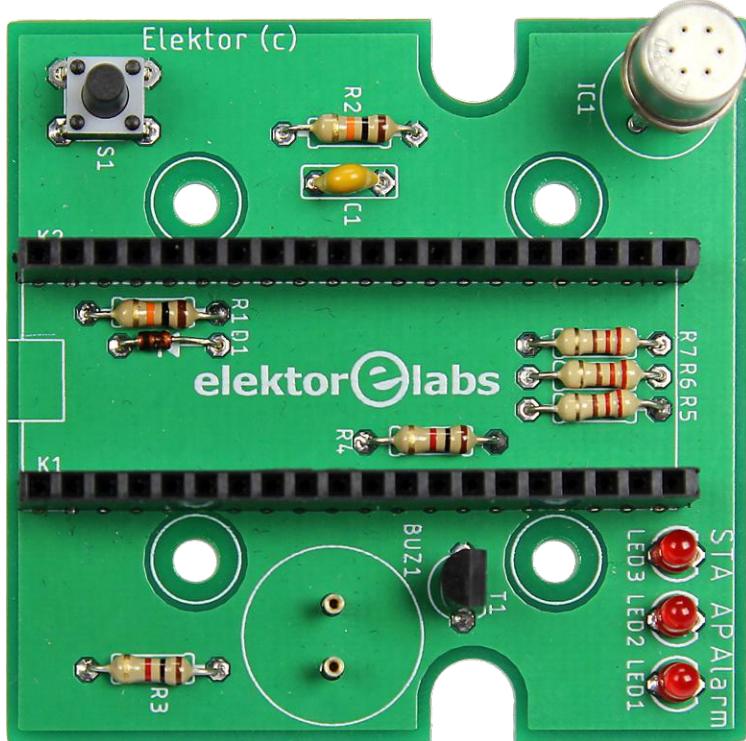
De TGS2600 is gemakkelijk te gebruiken in een schakeling: een microcontroller met een geïntegreerde analoog/digitaal-omzetter (ADC) is alles wat u nodig hebt (**figuur 2**). Omdat we de data beschikbaar wilden maken in een net-



werk, hebben we gekozen voor een op de ESP32 gebaseerde microcontroller-module. We hebben eigenlijk twee versies van dit project: één die op een stuk



Figuur 2. Omdat al het moeilijke werk eigenlijk wordt gedaan door de TGS2600, is het schema van de luchtkwaliteitsdetector heel eenvoudig.



Figuur 3. Sommige onderdelen zitten hier verborgen onder de ESP32-module. De TGS2600-sensor is het 'blikje' in de rechter bovenhoek van de print.

gaatjesprint is opgebouwd en eentje op een echte print. De eerste versie bestaat uit een ESP32 Pico Kit-module, een 0,96" OLED-display, een drukknop, een zoemer en, natuurlijk, de sensor. De versie op een print is bijna identiek, met uitzondering van het OLED-display, omdat deze versie in plaats daarvan gebruik maakt van een internetpagina om de meetwaarden te visualiseren. Er zijn drie LED's toegevoegd om de status van de draadloze netwerkverbinding weer te geven (**figuur 3**).

De printversie is ontwikkeld in het Elektor-lab en past in een goedkoop Hammond-kasje van het type 1593k (**figuur 4**).

...en de software

De software hoeft in feite niet meer te doen dan de spanning over R2 te meten. Als de waarde van R2 bekend is (10 kΩ in het schema) kan de waarde van R_{GAS} worden berekend. En als R_o bekend is, is de verhouding R_{GAS}/R_o een maat voor de gemeten luchtverontreiniging. Het

Weblinks

- [1] U8g2lib grafische bibliotheek: <https://github.com/olikraus/u8g2>
- [2] ESP32 Arduino-core: <https://github.com/espressif/arduino-esp32>
- [3] SPIFFS upload-tool: <https://github.com/me-no-dev/arduino-esp32fs-plugin>



ONDERDELENLIJST

Weerstanden:

R1,R2 = 10 k
R3,R4 = 1 k
R5,R6,R7 = 220 Ω

Condensatoren:

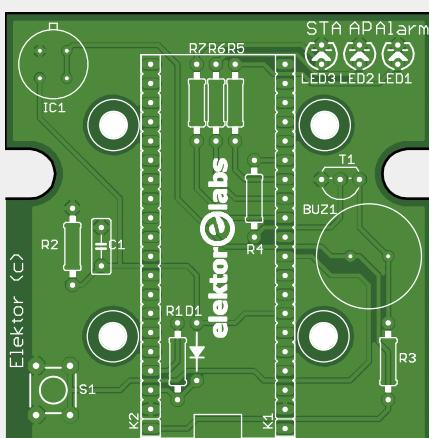
C1 = 100 n

Halfgeleiders:

D1 = BAT85
IC1 = TGS2600
LED1..3 = LED, 3 mm, rood
T1 = BS170

Diversen:

BUZ1 = PS1420P02CT
K1,K2 = 17-polig busstrip, steek 0,1" (voor de
ESP32 Pico Kit)
S1 = druktoets voor printmontage, 6x6 mm



Print 170182-1 (Elektor Shop)
ESP32 Pico Kit (Elektor Shop)

ThingSpeak

Luchtkwaliteitsmetingen zijn geschikt om te worden overgedragen naar het open IoT-platform ThingSpeak. U moet daarvoor een kanaal aanmaken met één dataveld. U krijgt dan een Write API-key die u moet invoeren op de configuratiepagina van de ESP32. Als dat is gelukt, worden de metingen telkens na een meetinterval overgedragen. De verzonden waarde is de gemiddelde waarde berekend over het hele interval.

Om de data naar ThingSpeak te sturen, wordt eerst een HTTP-request met de sensorwaarde verzonden naar de website. ThingSpeak reageert dan met het aantal entries in het kanaal, of met nul (0) als het request is mislukt. Enkele redenen waarom een request kan mislukken, zijn een verkeerd ingevoerde API-key of een te snelle opeenvolging van requests (ThingSpeak begrenst het upload-interval tot 15 seconden voor niet-betalande gebruikers). Het minimum upload-interval van onze luchtverontreinigingsmonitor is één minuut.

zwakte punt is hier R_0 , omdat de waarde daarvan onbekend is. Om die te bepalen, moet een kalibratie worden uitgevoerd in een referentieomgeving waar de lucht verondersteld wordt 'schoon' te zijn. De berekende waarde van de luchtverontreiniging wordt ofwel getoond in een grafiek op het kleine OLED-display, of overgedragen naar het populaire IoT-platform ThingSpeak, waar de metingen kunnen worden gevisualiseerd als een grafiek over een langere meetperiode. Het OLED-display toont alleen de laatste 20 metingen.



De metingen worden ook vergeleken met een drempelwaarde. Als de spanning die met de luchtverontreiniging correspondeert onder die grenswaarde komt, begint de zoemer te werken. Volgens de datasheet van de sensor is een R_{GAS}/R_0 -verhouding van 0,85 een redelijke drempelwaarde voor het alarm. De alarmdrempel is op het display zichtbaar als een horizontale stippellijn. Als het alarm actief is, kan het worden gestopt door op de knop te drukken.

Het indrukken van de drukknop levert ook enige informatie over de momentele WiFi-verbinding en de drempelwaarde van het alarm. Daarnaast kunt u zien of ThingSpeak is ingeschakeld of niet.

Configuratie en kalibratie

Om het project nog leuker te maken, hebben we wat positieve complexiteit toegevoegd in de vorm van een webserver. Daarmee kunnen we onze luchtvervuilingsmonitor configureren met behulp van een handige internetpagina, die wordt gehost op de ESP32 zelf. Op die pagina kunnen de alarmdrempel instellen, evenals de ThingSpeak-configuratie,

Past altijd

Beide versies van dit project (de gaatjesprint-uitvoering met, en de 'echte' print zonder OLED-display) gebruiken dezelfde Arduino-sketch. Om tussen de beide versies te onderscheiden, gebruiken we een preprocessor-definitie. Deze staat in de eerste regel in de code: `#define USEDISPLAY`. Verwijder die regel of commenteer hem uit voor de PCB-versie. Een extra bibliotheek met de naam U8g2lib (te vinden op: <https://github.com/olikraus/u8g2>) is nodig om het display aan te sturen. Die wordt gebruikt om de tekst en de grafiek weer te geven op het display.

Om de software naar de ESP32 te kunnen overdragen, moet de ESP32 Arduino-core worden geïnstalleerd (te vinden op: <https://github.com/espressif/arduino-esp32>). Na het installeren daarvan moet ook het SPIFFS-uploadtool worden geïnstalleerd om de internetpagina te uploaden naar het bestandssysteem van de ESP32 (dit staat op: <https://github.com/me-no-dev/arduino-esp32fs-plugin>). Als alles goed is geconfigureerd kan de ESP32 Pico Kit worden geselecteerd in het Board-menu en kunnen de internetpagina en de sketch in de ESP32-module worden geladen.

zoals de API-key die nodig is om data te mogen uploaden, en de tijd tussen twee data-uploads (zie het tekstkader ThingSpeak).

Ook het WiFi-netwerk dat uw apparaat moet gebruiken, kan worden geconfigureerd op die internetpagina. Maar de belangrijkste functie van de pagina is de kalibratie van de sensor door de referentiewaarde voor de weerstand in schone lucht, R_0 , in te stellen. Om dit op de goede manier te doen, moet de sensor een tijdlang in een omgeving met schone lucht (dus niet in een studentenkamer :-)) ingeschakeld blijven totdat de weerstand zich stabiliseert. In de datasheet van de sensor wordt een periode van zeven (!) dagen genoemd. Druk op de knop 'Calibrate sensor' zodra deze periode is verstreken.

Access point-modus

Bij de eerste gebruiknaam moet de ESP32 in acces point-modus (AP) wor-

den gebracht. Activeer deze modus door de module in te schakelen, terwijl u de drukknop ingedrukt houdt; LED2, 'AP', zou dan moeten oplichten. Als dat het geval is, maak dan met een computer, smartphone of tablet verbinding met het netwerk 'ESP32' en navigeer naar IP-adres 192.168.4.1 om op de pagina van de luchtverontreinigingsmonitor te komen. Dan kunt u het WiFi-netwerk en de andere bovengenoemde parameters instellen.

De communicatie tussen de internetpagina en de ESP32-module gebeurt via asynchrone XMLHttpRequests op de webpagina. Als de pagina wordt geladen, wordt er een HTTP-request samengesteld en verzonden naar de ESP32. Als de ESP32 dit verzoek ontvangt, stuurt hij een antwoord met de huidige configuratie terug naar de internetpagina. Door op de knop Submit op de internetpagina te drukken, wordt een soortgelijke transactie gestart, maar dan om de door de gebruiker ingevoerde waarden naar de ESP32 te sturen. ▶

(170182-02)

@ WWW.ELEKTOR.COM

- Luchtvervuilingsmonitor, kale print
www.elektor.nl/air-pollution-monitor-170182-1
- ESP32 Pico Kit
www.elektor.nl/esp32-pico4
- TGS2600 Air Contaminants Sensor
www.elektor.nl/tgs2600-air-sensor
- 0,96" 128x64 OLED-display, 6-polig
www.elektor.nl/oled-display-for-arduino

I²C-bus datalezer

Via USB alles op het pc-scherm

Willem den Hollander (Zwitserland)

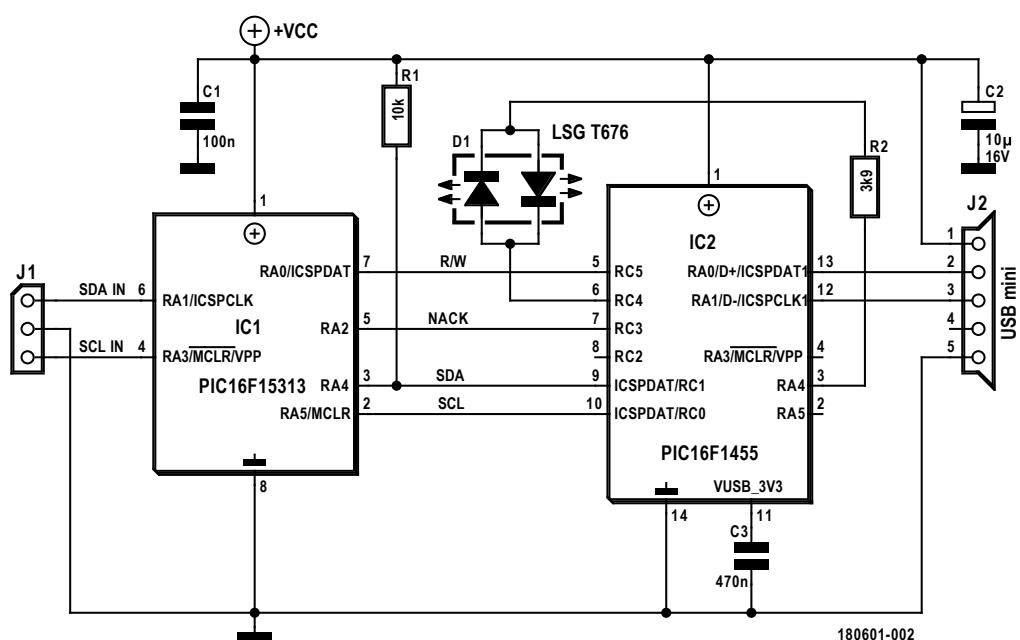
De Inter-Integrated Circuit-bus — oftewel I²C-bus — wordt regelmatig gebruikt voor communicatie tussen geïntegreerde schakelingen. Het is vaak nuttig om te weten welke gegevens over deze bus worden verzonden. De hier beschreven schakeling leest de I²C-gegevens en stuurt deze via de USB naar een pc die deze vervolgens in hexadecimale vorm op het scherm weergeeft. Optioneel kunnen de gegevens worden gefilterd om gegevens met specifieke adressen te selecteren.

De I²C-bus bestaat uit twee signaallijnen: de SCL-klok en de SDA-datalijn. De gegevens worden in serie doorgegeven. Een bericht bestaat uit twee of meer bytes, elk gevolgd door een bevestigingsbit ("Acknowledge"). Een startreeks geeft het begin van het bericht aan, een stopreeks sluit het af. Om alle informatie over te dragen, moeten we twee keer zoveel bytes over de USB versturen als de data zelf omvat. De klokfrequentie in de "standaard modus" van de I²C-bus is 100 kHz, wat resulteert in een datasnelheid van 20 kB per seconde. In de "fast mode" wordt dit 80 kB per seconde. In de hier beschreven toepassing wordt per milliseconde één pakket van 64 bytes via de

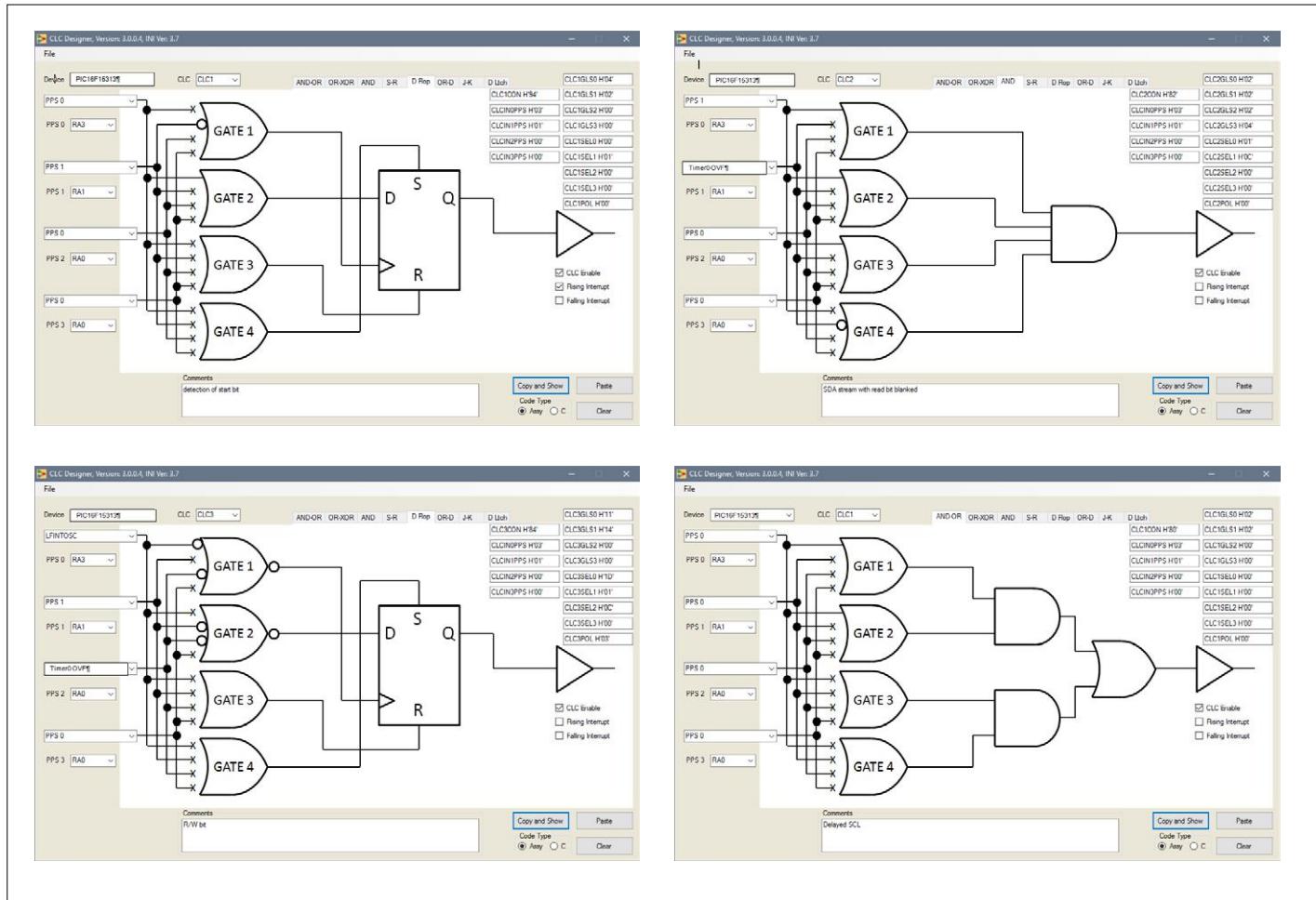
USB verzonden. Dit is meer dan voldoende voor het uitlezen van gegevens op de I²C-bus in de standaardmodus, maar niet genoeg voor de snelle modus. De I²C-bus datalezer kan echter nog steeds gebruikt worden voor het lezen van gegevens in de fast mode als er slechts af en toe gegevens aanwezig zijn, wat vaak het geval is.

De hardware

Figuur 1 toont het schema van de I²C-bus datalezer. We zien hier twee microcontrollers. De PIC16F1455 (U2) gebruiken we om de I²C-data uit te lezen en via USB naar de pc te verzen-



Figuur 1. Hier is het volledige schema van de I²C-bus datalezer afgebeeld.



Figuur 2. De CLC's in U1 zijn voor de I²C-bus datalezer op deze manier geconfigureerd.

den. De I²C-module van deze controller is geconfigureerd als slave-ontvanger om alle gegevens op elk willekeurig adres te verzamelen. Dit gebeurt echter alleen als het schrijfbit van de binnengekomen data aanwezig is. Als de leesbit is ingesteld, worden alle gegevens genegeerd. Bovendien kan de bevestigingsinformatie niet worden gelezen, aangezien de I²C-module deze zelf levert.

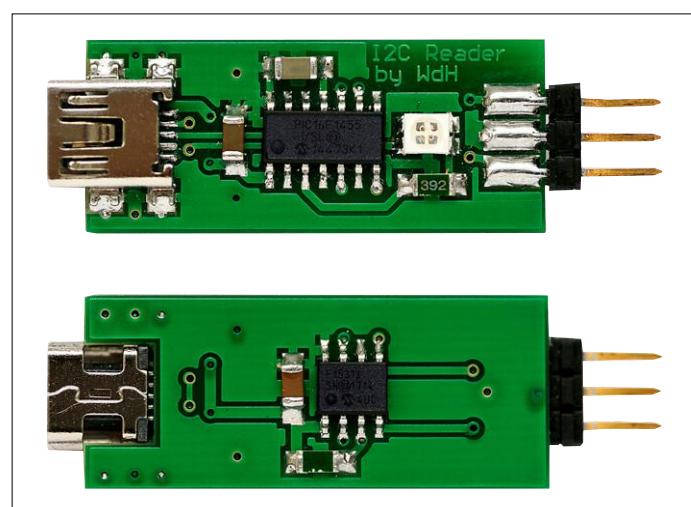
Om deze redenen hebben we een preprocessor ingezet, een PIC16F15313 (U1), die elk leesbit in een schrijfbit omzet. De leesbits en de ontvangstbevestiging worden uit de datastrook gehaald en separaat doorgegeven aan de PIC16F1455. Deze laatste neemt deze informatie weer op in de ontvangen I²C-gegevens en stuurt deze via de USB naar de PC.

U1 is een PIC die is uitgerust met vier configurerbare logische cellen (CLC). In **figuur 2** tonen we hoe ze voor dit project geconfigureerd zijn.

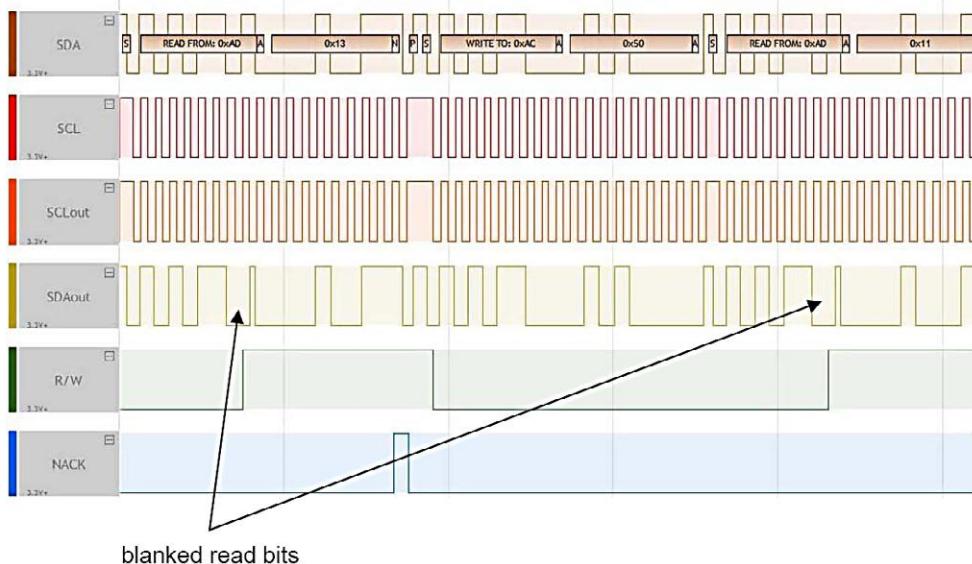
De PIC16F1455-microcontroller bevat de volledige USB-interface. De interne oscillator is nauwkeurig genoeg en een extern kristal is niet noodzakelijk. De microcontroller plaatst alle ontvangen bytes op de juiste geheugenplaatsen en na ontvangst van voldoende gegevens worden ze naar de pc gestuurd.

De firmware voor beide processoren is geschreven in assembler. Kritische timing is op deze manier nauwkeurig onder controle. De configuratie van de USB-module is ook in assembler geschreven. Dit deel van de firmware is oorspronkelijk geleverd door Microchip en na enige aanpassingen en debugging

is het nu al ongeveer twaalf jaar in gebruik. De schakeling heeft slechts weinig stroom nodig. Zijn voeding haalt hij uit de USB-aansluiting van de pc. Ik heb de schakeling opgebouwd met SMD-onderdelen op een kleine dubbelzijdige print met een afmeting van 14 bij 33 mm (zie **figuur 3**).



Figuur 3. De boven en onderzijde van de met SMD's opgebouwde print van de auteur.



Figuur 4. De signalen rond U1 met daarin aangegeven waar het lees-bit verwijderd is.

Firmware & software

Timer0 van U1 genereert een één-bit breed venster dat het lees-bit leegmaakt, zie ook **figuur 4**. In **Listing 1** tonen we hoe CLC2 het lees-bit op '0' zet. Het gewijzigde SDA-singaal, de uitgang van CLC2 – zie rechtsboven in figuur 2, is verbonden met uitgang RA4. RA4 is een open-drain-uitgang met een externe pull-up-werstand. Dit is een vereiste omdat U2 na ontvangst van een byte van de preprocessor altijd een bevestigingssignaal (ACK) terugstuurt. De ingang van U2 wordt dus gedurende één bit een uitgang. De uitgelezen informatie staat ter beschikking aan de uitgang van CLC3 (zie linksonder in figuur 2) en wordt naar uitgang RA0 geleid.

Omdat het SDA-singaal iets vertraagd wordt door de CLC, moet

om de timing correct te houden ook het SCL-singaal worden vertraagd. Dat doen we hier met CLC4. Een klein programma bestuurt de timer en leest het bevestigingsbit af op RA2.

Figuur 5 toont de grafische user-interface (GUI) van de bijbehorende pc-software. De statusregel onderaan geeft aan of de I²C-bus datalezer aangesloten en herkend is of niet. Met de knop "Start Acquisition" wordt het lezen van de I²C-data gestart. De led van de lezer licht dan groen op. De startknop wordt nu de stopfunctie. De I²C-busgegevens worden in het venster weergegeven, waar volgens de norm S staat voor Start, R voor Read ('lees'), W voor Write ('schrijf'), A voor Acknowledge ('bevestig'), N voor No Acknowledge ('geen bevestig') en een P voor stop. De databytes worden in hexadecimaal formaat weergegeven.

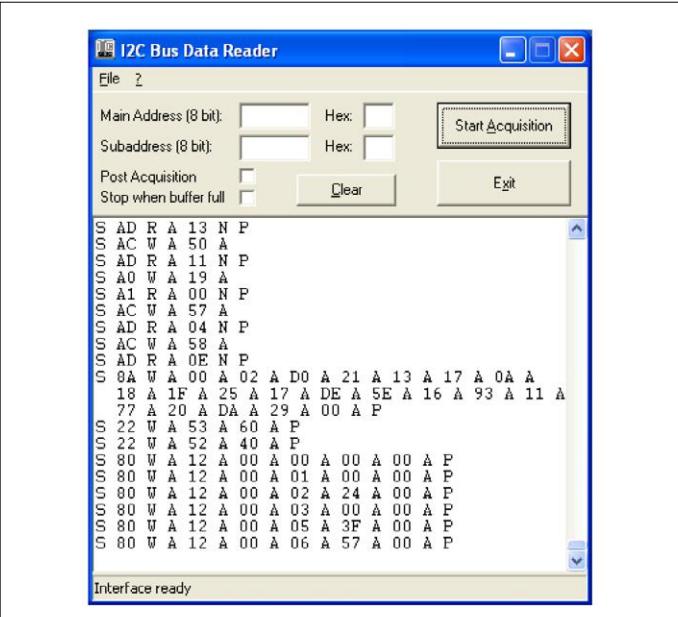
Listing 1.

Dit stukje broncode toont de configuratie van CLC2 (figuur 2b). Hier wordt het lees-bit uit de SDA data stream op nul gezet. Timer0 wordt gestart bij detectie van het start-bit.

```

banksel CLC2GLS0 ; TMR0 ANDed with SDA clears bit 8
movlw 2
movwf CLC2GLS0
movwf CLC2GLS1
movwf CLC2GLS2
movlw 4
movwf CLC2GLS3
movlw 1      ; SDA input signal
movwf CLC2SEL0
movlw 0ch     ; TMR0 overflow
movwf CLC2SEL1
clr f CLC2SEL2
clr f CLC2SEL3
clr f CLC2POL
movlw 82h
movwf CLC2CON

```



Figuur 5. De gebruikersinterface is heel overzichtelijk en voorzien van enkele handigheidjes.

Met de software is het mogelijk om te filteren in hexadecimale of binaire vorm op de eerste twee bytes. Wanneer deze bytes ingevuld zijn, zullen alleen gegevens die beginnen met deze bytes worden weergegeven. Als "Post Acquisition" is aangevinkt, start de data-acquisitie pas na ontvangst van de opgegeven bytes. Bij filteren met de binaire vorm is het ook mogelijk om met don't cares te werken, op de plaats van de don't care voer je dan een 'X' in. De grootte van de buffer is ongeveer 40 kbyte. Als de buffer vol dreigt te raken wordt een deel van de gegevens aan het begin van de buffer verwijderd. Het aanvinken van "Stop when buffer full" verhindert dit door de data-acquisitie op dat moment te stoppen. De groene led dooft zodra het acquireren stopt. Als er te veel gegevens aanwezig zijn, stopt de data-acquisitie en de led licht rood op. De software wordt dan automatisch weer in de startsituatie gebracht. De gegevens in het logvenster blijven staan tot ze gewist worden of totdat de buffer vol is en een deel wordt verwijderd. Via het menu "File" kun je de gegevens in het venster geheel of gedeeltelijk op de harde schijf opslaan of afdrukken.

Ik heb gemerkt dat het continue weergeven van de gegevens een zware belasting vormt voor het Windows-besturingssysteem. Daarom is het aan te raden om niet tegelijkertijd andere programma's op de pc uit te voeren.

In gebruik

Voordat je de I²C-bus datalezer kunt gebruiken, moet je het USB-stuurprogramma installeren. Wanneer je de lezer via de USB-kabel aan de pc koppelt, ziet de pc een nieuw apparaat. In de download bij dit artikel [1] is een .inf-bestand in de dri-

ver directory opgenomen. Selecteer dit bestand wanneer de pc om een stuurprogramma vraagt en het stuurprogramma wordt automatisch geïnstalleerd. De gebruikersinterface installeert door te dubbelklikken op het Setup.exe-programma dat eveneens in de download op [1] te vinden is en de aanwijzingen op het scherm te volgen.

Conclusie

Het bijzondere van de beschreven I²C-bus datalezer is, dat data gelezen kunnen worden zo lang het nodig is, voor een onbeperkte tijd. We kunnen zo fouten opsporen die slechts zelden optreden. Door de pc-software om te schrijven, is het mogelijk om ontvangen data op de harde schijf op te slaan, die we dan later kunnen analyseren. ▀

180601-01

Weblink

[1] Software en drivers: www.elektrormagazine.nl/180601



IN DE STORE

- Boek: Elementary Course BASCOM-AVR
www.elektor.nl/elementary-course-bascom-avr
- Boek: Make: AVR Programming
www.elektor.nl/make-avr-programming

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!

Automotive-stoorbronnen

en hoe je die elimineert

Tam Hanna (Slowakije)

Wie nog uit de tijd van de klassieke PLC's stamt en een schakeling voor een voertuig moet ontwerpen, komt al gauw in de 'automotive-val' terecht. Wat daar allemaal kan misgaan en welke risico's de ontwerper en zijn hardware lopen, daarover gaat dit artikel.

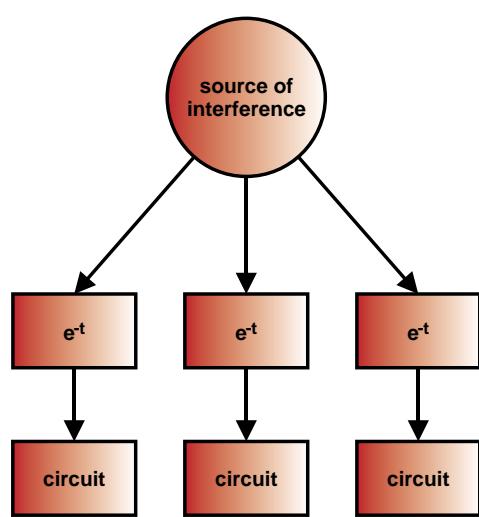
Voordat we in het diepe duiken eerst een woord vooraf: de auteur kent talloze in kleine series geproduceerde schakelingen die deels al jaren worden gebruikt en waarin nauwelijks of geen van de hier beschreven beschermingsmaatregelen zijn

toegepast. Dat ligt – onder andere – daaraan, dat een automotive-systeem – zoals in **figuur 1** – ongeveer kan worden voorgesteld als een netwerk van e^{-t} -structuren. Sommige schakelingen zitten 'dichter bij het vuur' en krijgen de grootste storingen te verduren, terwijl andere relatief gespaard worden. De voor een complete vrijwaring van problemen noodzakelijke maatregelen zijn echter niet eens zo duur en zijn in elk geval economisch gezien gerechtvaardigd.

De brute dynamo

Laten we maar beginnen met de ergste van alle storingen, namelijk die welke in de industrie in het Engels *load shed* wordt genoemd. Laten we als voorbeeld een quad – of vergelijkbaar – voertuig nemen, die als een gek door een hobbelige straat scheurt. Helaas is het niet zo best gesteld met de verbinding tussen dynamo en accu. De fabrikant heeft bezuinigd, de ontwerper heeft zitten slapen, of er heeft zich roest gevormd in de verbindingen. Hoe dan ook, bij een flinke kuil raakt de verbinding dynamo-accu kortstondig onderbroken. De dynamo wil, nee moet, ergens zijn overtollige energie kwijt. Toegegeven: dit fenomeen treedt gelukkig maar zelden op.

Hier wrekt zich iets wat in de automotive-sector vaker gebeurt: de concurrerende fabrikanten specificeren allerlei meer en minder relevante zaken en maken daarbij naar hartelust gebruik van eveneens concurrerende standaardisatie-instituten. Een ware jungle!



Figuur 1. Schakelingen zijn blootgesteld aan verschillende storingsbronnen.

Terug naar onze load shed. De ISO-groep kent twee specificaties: om te beginnen de sinds 2010 geldende norm ISO-7637-2, die bij enkele autofabrikanten (bijv. Ford) wordt toegepast en via [1] is te downloaden. Ten tweede de nu vigerende versie ISO-16750-2, die op dit moment echter alleen tegen betaling kan worden ingezien. Halfgeleiderfabrikant Vishay stelt onder [2] een kort overzicht beschikbaar, dat de actuele norm samenvat. Texas Instruments verraat in [3] de opbouw van pulsen, zoals te zien in **figuur 2**.

Per saldo kunnen we stellen dat een 12-V-systeem moet worden blootgesteld aan een 400 ms durende puls van maximaal 101 V, met een weerstand van minimaal $0,5 \Omega$ en maximaal 4Ω . Bij een 24-V-systeem is de maximale duur 350 ms, de spanning bedraagt 151...202 V, de inwendige weerstand ligt tussen 1 en 8Ω .

Vrij naar Adam Riese volgt daaruit dat een 12-V-systeem een momentaan vermogen van 20.400 W moet dissiperen en dat een 24-V-systeem zelfs 40.880 W moet kunnen weerstaan. Wat het allemaal nog erger maakt is de bijzonder korte stijgtijd (*rise time*) van ca. 10 ms. Schakelingen moeten zeer snel reageren, wat onderdelen als relais van meet af aan uitsluit. Wat het eveneens erger maakt is dat de nieuwe norm vereist dat een schakeling tien van zulke pulsen in een periode van 10 minuten moet kunnen verdragen. Onder de oude norm volstond het nog, dat het apparaat één zo'n puls overleeft. Niettemin is deze eis uit de nieuwe norm wel zinvol, omdat een voertuig met gecorrodeerde accuklemmen doorlopend load shed-storings kan produceren.

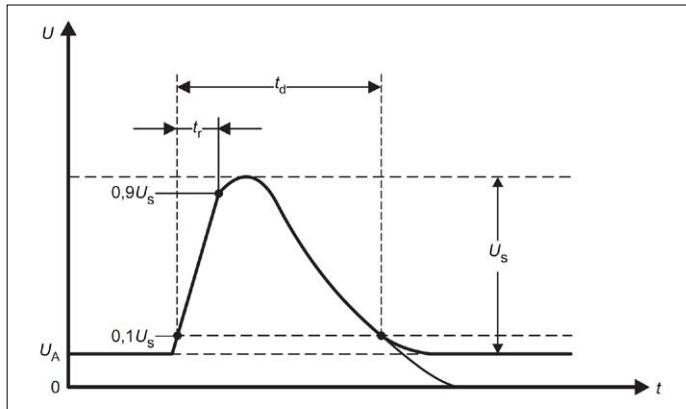
Een klassieke manier van zelfbescherming tegen zulke pulsen is – naast het voorschakelen van een massieve capaciteit en een weerstand – het invoegen van een zeer snel reagerende beschermingsschakelaar, die de te beschermen schakeling simpelweg loskoppelt van het boordnet (**figuur 3**).

Gelukkig is de halfgeleiderindustrie zich zeer goed bewust van dit probleem en biedt dan ook onderdelen aan die geschikt zijn om transiënten tegen te gaan. Met name de firma Littelfuse is in dit verband zeer succesvol. De als TVS (Transient Voltage Suppressor-diode) bekend zijnde en onder [4] beschreven dioden fungeren als een soort breekijzer dat in twijfelgevallen de spanning begrenst. Maar ook STM heeft een palet aan suppressordioden, door STM ‘Transil’ genoemd, onderverdeeld naar de diverse ISO-testpulsen, waardoor ontwikkelaars makkelijk geschikte componenten kunnen kiezen (**figuur 4**). Vishay noemt de ESD-dioden ‘Transzorb’.

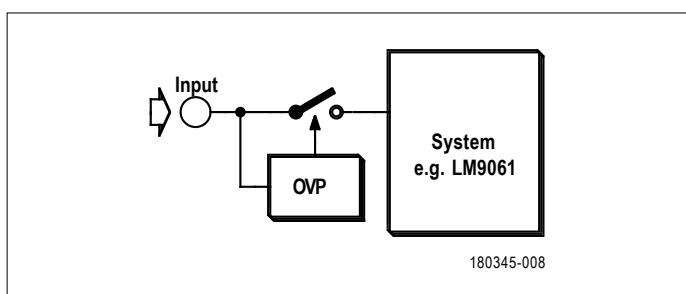
Let er wel op, dat transils in zowel uni- als bidirectionele uitvoering worden aangeboden. Unidirectionele transils gedragen zich net als zenerdiodes: komt de puls uit de verkeerde richting, dan wordt die gewoon doorgelaten. Bidirectionele transils hebben in beide richtingen een minimale spanning, voordat ze geactiveerd worden.

De auteur gebruikt graag de voor redelijk hoge spanningen geschikte schakelende regelaar LM2576HV, uit de ‘Simple Switcher’-familie van TI (die overigens op het vlak van koudestart-spanning niet uitblinkt), om het werk van de diode te verlichten door die op een voerspanning van bijvoorbeeld 35 V te zetten, waardoor het te dissiperen vermogen niet zo extreem hoog is.

TVS-diodes zijn overigens niet de enige componenten die geschikt zijn voor het bestrijden van de diverse soorten spikes. Varistoren (VDR) zijn bidirectionele, spanningsafhankelijke weerstanden, die in principe een diode-achtig gedrag verto-



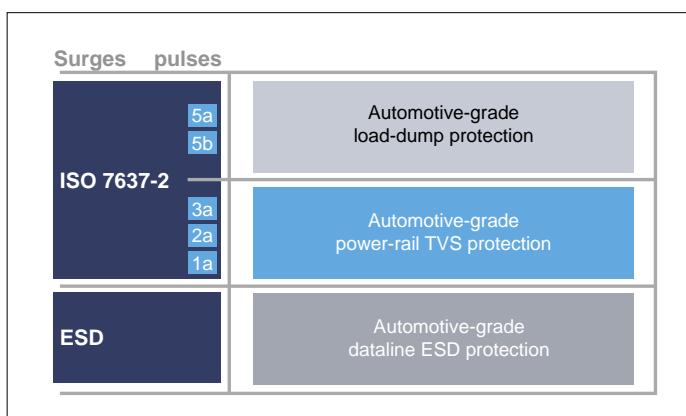
Figuur 2. Met zulke spikes is het slecht kersen eten (Bron: TI [3]).



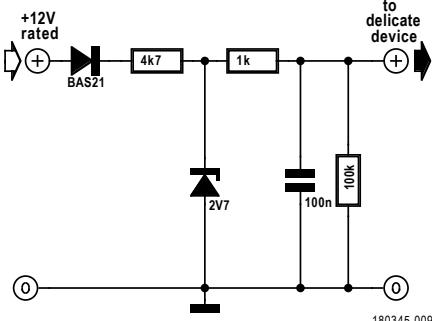
Figuur 3. Lafheid kan zich lonen!

nen. Ze onderscheiden zich van TVS door een wezenlijk hogere energiecapaciteit, maar schakelen langzamer en degraderen in de loop van de tijd (ook al spreekt Vishay in [6] over “Negligible wearout within SOA”). Problematisch in dit verband is de tendens tot doorslag bij verouderde Metal Oxide Varistors (MOV). De auteur zijn meerdere systemen bekend die dankzij een verouderde MOV te kampen hadden met één of meerdere kortgesloten spanningsrails.

In de praktijk wordt meestal een combinatie toegepast van een MOV en een TVS-diode met een wat hogere doorslagsspanning. De achterliggende gedachte is de combinatie van de snelle schakelkarakteristiek van de TVS-diode met de hogere ther-



Figuur 4. STM deelt zijn Transil-familie op basis van de ESD-bestendigheid in (Bron: STM [5]).



Figuur 5. Niet origineel, maar wel simpel: spanningsbegrenzing met een zenerdiode.

mische belastbaarheid van de MOV. Maar let erop, dat norm IEC 60950-1 de toepassing van MOV's voorschrijft en die van transils niet toestaat!

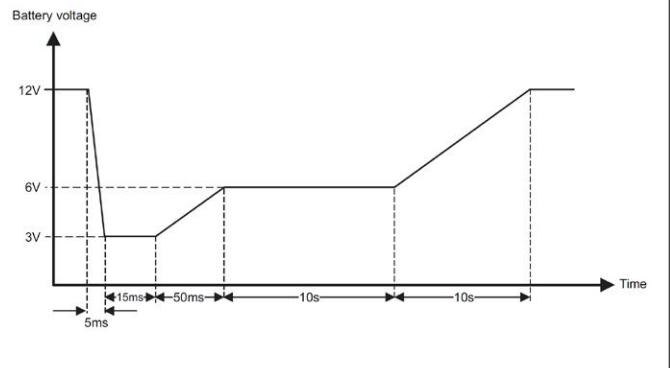
Venijnige spanningen

Naast de bescherming van de hoofd-spanningsbron moeten ook gevoelige onderdelen respectievelijk ingangen met gulle hand van spanningsbegrenzers worden voorzien. De klassieker is natuurlijk de niet bijzonder elegante, maar niettemin doeltreffende zenerdiode-schakeling van **figuur 5**. Bij deze overspanningsbeveiliging volstaat het ironisch genoeg, ervoor te zorgen dat de weerstand en de zenerdiode voldoende vermogen kunnen dissiperen, om eventueel optredende spanningspieken in warmte om te zetten. De weerstand moet de zenerstroom op een bepaalde waarde begrenzen, maar mag daarbij niet in rook opgaan.

Theoretisch gezien zou een 12-V-systeem op 14, maximaal 15 V moeten worden begrensd. Helaas ligt dat in de praktijk een beetje anders, omdat er een onvoorspelbare factor in het geding is, te weten de gebruiker (de 'wetware'). Een klassieker is dat iemand door nalatigheid of onwetendheid een accu heeft uitgeput en dan starthulp vanuit een ander voertuig gebruikt. Daarbij komt het verrassend vaak voor, dat aan een personenwagen met 12-V-boordspanning starthulp wordt verleend door een vrachtwagen (met zijn 24-V-boordspanning). In Angelsaksische landen is het populair om als starthulp de twee accu's in serie te schakelen. Hoe het ook zij, de beveiligingsschakeling mag in geen geval de 24-V-boordspanning als een te begrenzen puls beschouwen en die proberen kort te sluiten. Overigens zullen de meeste schakelende spanningsregelaars geen al te grote problemen hebben als kortstondig spanningen van 24 V aanliggen, als de schakeling tenminste een beetje snugger is ontworpen.

Wantrouw de spanning!

Het 'buiten gevecht stellen' van overspanningen is slechts het halve verhaal. Het zijn niet alleen stoorpulsen die storen. Een ander probleem zijn namelijk spanningsdips die door verschillende oorzaken kunnen ontstaan. Zoals het plotseling inschakelen van een 'zware' verbruiker, bijvoorbeeld door het starten van de verbrandingsmotor bij lage buitentemperatuur. Deze in het Engels als *cold crank* bekende situatie is gevreesd, omdat dan als gevolg van de door de koude slecht functionerende



Figuur 6. Cold cranking kan heel onaangenaam worden. (Bron: TI).

accu het in **figuur 6** geschatte verloop van de boordspanning kan optreden.

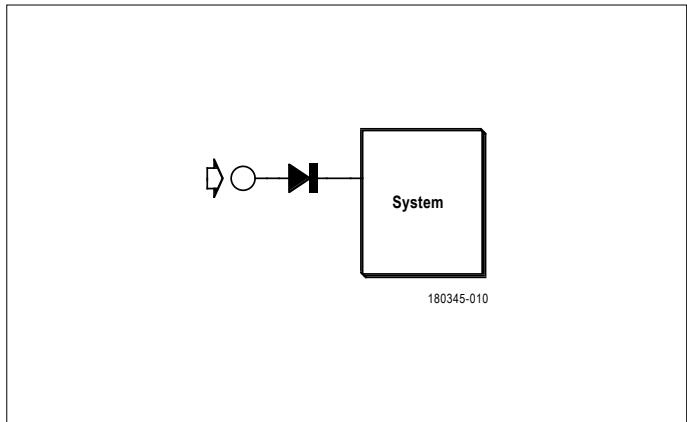
Op het eerste gezicht ben je als ontwerper geneigd cold cranking te negeren. Spanningsregelaars zoals de eerder genoemde LM2576 zijn ook nog gespecificeerd bij een ingangsspanning van 8 V. De ISO specificeert echter – afhankelijk van hoe essentieel het betreffende systeem is – spanningen tot aan 3 V. Voor het bestrijden van dit soort problemen zijn meerdere oplossingen denkbaar. Ten eerste natuurlijk het voorschakelen van een forse condensator, die bovendien bij snel stijgende spanningen de stijgcurve afzwakt. De tweede mogelijkheid is, het eigen systeem zo te ontwerpen dat het in geval van een cold crank simpelweg stopt en pas weer verder gaat als de spanning weer hoog genoeg is. Dit is echter niet zo'n fraaie oplossing, temeer omdat steeds meer microprocessors een operating systeem hebben met een lange opstarttijd, zeker wanneer het Unix-achtige systemen betreft.

Een andere, niet minder interessante, oplossing is de toepassing van de buck/boost-technologie. Het gaat hierbij om schakelende regelaars die zowel omhoog als omlaag kunnen werken. Texas Instruments geeft daarvoor in Application Note 728 [7] nadere aanwijzingen. En als de regelaar werkelijk met 3 V ingangsspanning uit de voeten kan, dan hoeven we ons over cold cranking geen zorgen meer te maken.

De strijd tegen de negatieve pulsen

Je kunt er als ontwerper natuurlijk om lachen als een gebruiker of monteur de accu onbedoeld verkeerd-om aansluit, maar het gebeurt in de praktijk toch. Gelukkig is het niet zo moeilijk om maatregelen tegen deze situatie te nemen door het simpelweg invoegen van een diode zoals in **figuur 7**. Maar we moeten er wel rekening mee houden dat zo'n beveiligingsdiode slechts in beperkte mate bescherming biedt. Een voertuig is namelijk niet alleen een netwerk van e^{-t} -structuren, maar ook een verzameling van inductieve verbruikers, die bij in- en uitschakelen allerhande spikes veroorzaken.

Naast de positieve spikes, waartegen TVS-dioden worden ingezet, komen ook spikes voor die spanningen tot wel -200 V (!) op de ingangen van elektronica zetten. Daarom moet de diode extreem zorgvuldig worden geselecteerd. De auteur heeft goede ervaringen opgedaan met de S5BC-13-F van Diodes Incorporated voor hoofdvoedingsschakelingen, terwijl voor beveiliging van verzwakkers (*attenuators*) en dergelijke de



Figuur 7. Zo eenvoudig kan het oplossen van het probleem zijn!

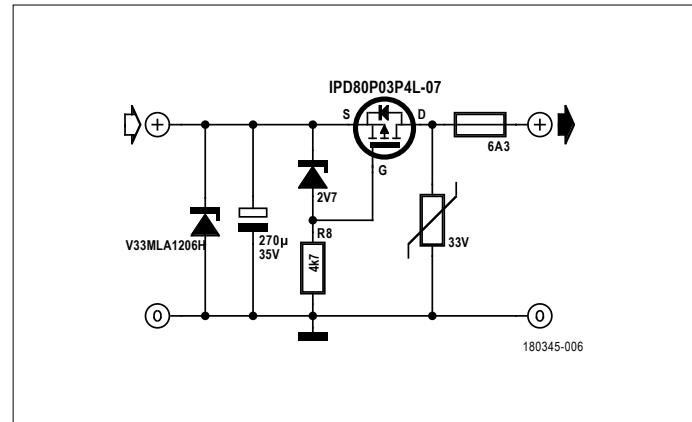
BAS20 beter geschikt is.

Klassieke dioden zijn al lang niet meer de enige oplossing voor dit soort toepassingen. Vanwege de relatief hoge doorlaatspanning zijn met name bij hogere stroomsterktes efficiëntere componenten gewenst. Hierin hebben ontwerpers van transistorschakelingen voor zowel P- als N-kanaal veldeffecttransistoren nieuwe toepassingen gevonden. **Figuur 8** laat schematisch zien hoe dergelijke schakelingen kunnen worden opgebouwd.

Tot slot

In dit artikel beperken we ons tot 12- en 24-V-systemen. Vanwege de steeds complexer wordende voertuigelektronica rijden tegenwoordig ook voertuigen rond met een 48-V-boordnet. De normen hiervoor zijn echter nog niet allemaal vastgelegd. In geval van twijfel kan men zich het beste tot de betreffende fabrikant wenden.

De automotive-sector is natuurlijk geen nichemarkt: het zal geen verbazing te wekken dat er een enorme hoeveelheid aan literatuur vorhanden is over elektronica in voertuigen. De in de inleiding genoemde ISO-normen vormen in elk geval de klassieke bronnen, maar zijn moeilijk verkrijgbaar. In Oostenrijk kunnen enige ervan bij het normalisatie-instituut worden ingezien, maar daarvoor moet men dus eerst een reis naar Wenen ondernemen. In Duitsland bestaat een veelheid aan norm-infopoints, meestal bij hogescholen, maar ook bij de



Figuur 8. Niet iedereen zal op het idee komen een MOSFET op deze manier in te zetten!

overheid en andere instituten. Een overzicht daarvan is onder [8] te vinden.

Een aardig alternatief voor de normbladen wordt beschikbaar gesteld door autofabrikant Ford [9]. Dit is een specificatielidocument dat eigenlijk is bedoeld voor leveranciers die componenten voor Ford willen produceren. Maar het is ook voor andere ontwerpers interessant, omdat het een op zichzelf staand document is, dat niet aldoor verwijst naar andere normen waardoor het niet meer nodig is die – ook – allemaal te lezen.

Zoals reeds in de inleiding gesteld, zijn er duizenden systemen die enige of zelfs alle hier besproken ontwerpregels botweg negeren en toch probleemloos functioneren. Maar wie zijn systeem met een TVS-diode en enkele andere onderdelen beschermt, hoeft zich in elk geval geen al te grote zorgen te maken. ▶

180345-03

Weblinks

- [1] ISO 7637-2: www.smd.ru/upload/medialibrary/a3d/iso_7637_2.pdf
- [2] Load dump protection: www.vishay.com/docs/49748/49748.pdf
- [3] Load Dump and Cranking Protection: www.ti.com/lit/an/snva681a/snva681a.pdf
- [4] Littelfuse Automotive TVS Diodes: www.littelfuse.com/~/media/electronics/application_notes/littelfuse_tvs_diode_automotive_circuit_protection_using_automotive_tvs_diodes_application_note.pdf.pdf
- [5] Automotive Protection Devices: www.st.com/content/st_com/en/products/protection-devices/automotive-protection-devices.html
- [6] Transient Suppressors, a Competitive Look: www.vishay.com/docs/88444/tvscmp.pdf
- [7] Output Voltage Regulation During Automotive Cold-Crank: www.ti.com/lit/an/snva728/snva728.pdf
- [8] Normen inzien: www.beuth.de/de/regelwerke/auslegestellen/
- [9] Electromagnetic Compatibility Specification: www.fordemc.com/docs/download/FMC1278.pdf

Storende ledlampen

Voldoen storende ledlampen wel aan de EMC-norm?

Johan Jongbloed (Nederland)

In april 2018 hebben we een oproep gedaan aan onze lezers om storende ledlampen voor onderzoek op te sturen naar de redactie. Hierop hebben we een redelijk aantal verschillende lampen mogen ontvangen. Om de ledlampen conform de Europese Norm EN 55015 te testen, heeft de EMC-EMF-commissie van de VERON (Vereniging van Experimenteel Radio Onderzoek in Nederland) aangeboden deze lampen aan een EMC pre-compliance test te onderwerpen.



Eerst iets over de EMC-richtlijn en normen. De EMC-richtlijn is opgenomen in de Telecom Wet — de normen zijn hierin niet opgenomen — en omschrijft “emissie” in bijlage I als volgt:

De opgewekte elektromagnetische verstoringen mogen het niveau niet overschrijden waardoor radio- en telecommunicatieapparatuur niet meer overeenkomstig hun bestemming kunnen functioneren.

Maar hoe moeten we dit interpreteren? Als een radio niet meer goed functioneert in de zin van de Richtlijn, dat wil zeggen wanneer er geen ontvangst van gewenste zenders mogelijk is, wordt deze duidelijk verstoord. Maar moet je hiervoor een minimale afstand tussen de stoerbron en radio aanhouden? De limieten en de meetopstellingen zijn in de jaren 70 ontwikkeld met in gedachte de enige op dat moment voorkomende stoerbronnen, zoals de collectormotor (stofzuiger en koffiemolen) en de ontstekingen van een verbrandingsmotor. In die tijd waren er geen computers, mobieljes, schakelende voedingen, ledverlichting of PV-(zonnenpaneel)systemen. Men ging er van uit dat de stoerbron bij de buren was, of in ieder geval ergens waar je zelf niet in staat bent om deze uit te schakelen. Daarom is er in de 70'er jaren gekozen voor een meetafstand van 10 meter tussen een apparaat en een stoerbron.

Is er sprake van voortschrijdend inzicht?

Niet echt. De tijden zijn veranderd, maar de normen zijn wat betreft de uitgangspunten in de jaren 70 blijven steken. Er zijn nu zoveel stoerbronnen onder één dak te vinden dat je als radioamateur bijna niet meer fatsoenlijk kan communiceren via de ether, zelfs niet met een goede buitenantenne. De

eigen stoerbronnen veroorzaken ongeveer 70% van de stoeringsklachten. Maar deze kun je zelf uitschakelen. Die van de buren zeker niet!

En daarbij komt dat de huidige stoerbronnen niet slechts een korte periode ingeschakeld zijn, zoals de stofzuiger of de koffiemolen. Deze nieuwe stoerbronnen staan vaak continu aan of worden eenvoudigweg niet uitgeschakeld.

Hebben de EMC-normcommissies dan niets gedaan?

De EMC-commissies hebben niet in de gaten gehad dat de radiocommunicatie zo snel zou worden verstoord als nu het geval is. Het is heel geleidelijk gebeurd zonder dat iemand iets direct is opgevallen. Het waren voornamelijk radiozendamateurs die als eerste aan de bel hebben getrokken. Zij werken vaak in of vlak boven het ruisniveau en merken daardoor als eerste dat er iets gaande is in het frequentiespectrum. Ook gaat men er in de EMC-regelgeving vanuit dat ieder product afzonderlijk moet voldoen aan de normen, terwijl er vaak meerdere (storende) producten tegelijk in gebruik zijn. Net als in een druk café waar veel mensen tegelijk praten, stijgt de achtergrondruis dan aanzienlijk.

Misschien wordt het langzaam tijd dan men gaat kijken naar een bijgestelde norm en dat bijvoorbeeld de gebruiks dichtheid per m² wordt bekeken en productlimieten met een aantal dB verlaagd moeten worden voor een enkel product.

De test: waarom EMC pre-compliance?

Voor ontwikkelaars van apparatuur is het (vooral financieel) aantrekkelijker om in het ontwikkel stadium eerst een pre-compliance test te laten uitvoeren. Een full compliant test of for-

mele test is vrij kostbaar (€ 1.000 tot € 4.000). Daarom voeren ontwikkelaars in het ontwikkelstadium eerst pre-compliance testen uit tegen veel lagere kosten. Zo kunnen EMC-problemen al vroeg in de ontwikkeling van een product opgespoord en opgelost worden nog voordat het apparaat full compliance getest wordt, waardoor de kans van slagen groter is.

Een EMC pre-compliance test zegt dus al vrij veel over het afstraalgedrag van een elektrisch apparaat en is in dit geval voldoende om iets zinnigs te kunnen zeggen over het afstraalgedrag van de ingestuurde ledlampen.

Vormen van afstraling

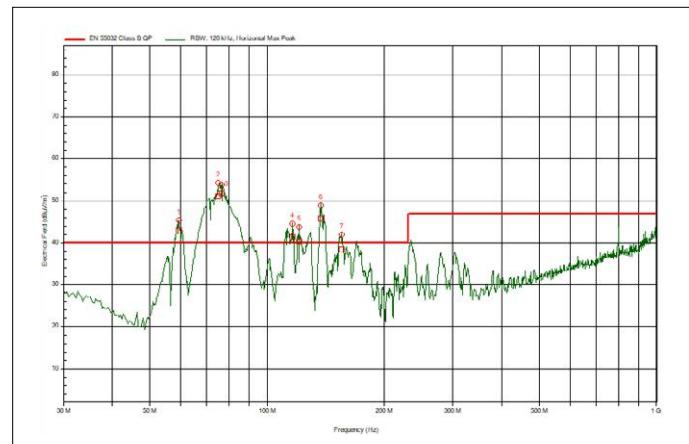
Bij het testen houden we rekening met twee vormen van afgestraald vermogen: *Conducted Emission (CE)* en *Radiated Emission (RE)*. De eerste is de elektromagnetische energie die wordt opgewekt in het inwendige van een apparaat en die via de 230V-bekabeling een weg naar buiten vindt. Het elektriciteitsnet in huis gedraagt zich hierbij als een lange draadantenne en kan deze signalen uitzenden. *Radiated Emission (RE)* is elektromagnetische energie die wordt opgewekt in het inwendige van een apparaat en die via de lucht wordt uitgezonden. Deze elektromagnetische energie wordt vaak door apparaten opgewekt die niet goed ontworpen zijn of die defecten vertonen. Apparaten die dergelijke energie uitstralen, kunnen problemen veroorzaken in andere apparatuur die zich in de directe omgeving bevindt.

In de norm EN 55015, die van belang is voor het afgestraalde vermogen, staat een alternatieve methode vermeld voor het meten van de *Radiated Emission*. Hierin wordt de emissie gemeten met behulp van een ontkoppelnetwerk. Deze meting noemen we in dit verslag ook een *Conducted emission*-meting.

Resultaten

Bij de CE-meting wordt het meetspectrum opgedeeld in twee gebieden: van 9 kHz tot 30 MHz en van 30 tot 300 MHz. Elke ‘frequentieband’ heeft zijn eigen meetopstelling. Bij de RE-meting wordt gemeten met een antenne op 3 of 10 m afstand van de (stoer)bron in het frequentiegebied van 30 tot 300 MHz. Omdat storing of hinder niet begint bij het overschrijden van de limieten maar al eerder, hebben we vier onderverdelingen gespecificeerd. Deze staan dus niet in een of andere norm, maar ze geven ons gewoon wat meer inzicht. In plaats van enkel de aanduiding of de lamp voldoet aan de norm of niet, geven we in 3 stappen aan hoeveel de emissie onder de norm ligt. **Tabel 1** laat de meetresultaten van de CE-metingen zien. Tot onze lichte verbazing lag de afstraling van alle geteste ledlampen bij de CE-meting onder de limiet. Bij de CE-meting in het bereik van 9 kHz tot 30 MHz kwam een groot deel wel erg dicht bij de limiet — slechts 0–5 dB er onder. Maar bij meer dan de helft van de ledlampen lag de afstraling meer dan 10 dB onder de limiet. Bij de CE-test in het 30–300MHz-bereik naderde slechts een klein gedeelte de limiet, maar overschreed geen enkele lamp de limiet.

In **figuur 1** tonen we de meting aan één van de twee ledlampen die bij de RE-meting *boven* de gestelde limiet uit kwamen. De rode lijn geeft de limiet aan, de groene het afgestraalde vermogen volgens de RE-meting. In dit geval ligt de piekoverschrijdingswaarde meer dan 10 dB boven de maximaal toegelaten waarde. Dit was echter de uitzondering die de regel bevestigde: slechts twee lampen overschreden de limiet en één kwam binnen 5 dB van de grenswaarde; de overige lampen



Figuur 1. RE-meting aan een ledlamp waarbij duidelijk is te zien dat de norm wordt overschreden.

Tabel 1. Meetresultaten van de ledlampen

CE 0,009 - 30 MHz	37,5%	0-5 dB onder de limiet
CE 0,009 - 30 MHz	6,25%	5-10 dB onder de limiet
CE 0,009 - 30 MHz	25%	10-15 dB onder de limiet
CE 0,009 - 30 MHz	31,25%	>15 dB onder de limiet
CE 30 - 300 MHz	6,25%	0-5 dB onder de limiet
CE 30 - 300 MHz	12,5 %	5-10 dB onder de limiet
CE 30 - 300 MHz	12,5 %	10-15 dB onder de limiet
CE 30 - 300 MHz	68,75%	>15 dB onder de limiet

gedroegen zich zo goed, dat ze niet eens de aanleiding gaven om er aan te meten. Een aantal lampen liet zelfs geen enkele toename in de ruisvloer zien.

Conclusie

Hoewel het hier om een EMC pre-compliantestest gaat, kunnen we met grote waarschijnlijkheid stellen dat het merendeel van de ledlampen aan de gestelde eisen voldoet. Maar waar komt die storing dan vandaan? Ligt het aan specifieke omstandigheden (denk aan versterkte antennewerking van 230V-kabels)? Ligt het aan het gebruik van meerdere ledlampen of stoerbronnen tegelijk? Ligt het aan gevoeligere omstandigheden of apparatuur? Is het onderzoek te beknopt? Wie het weet mag het zeggen (via redactie@elektor.nl). ▀

180608-01

Weblink

www.elektor.nl/180608-01



IN DE STORE

→ Boek: EMC for Product Designers

www.elektor.nl/emc-for-product-designers-5th-edition

Met de groeten van de tekenkamer het elektronica-tekenwerk en de cartoons van Elektor – puur handwerk!

Jan Buiting (redactie Retrotronica)

Vandaag de dag kun je er veilig vanuit gaan dat alle illustraties, opmaak en tekst die we op papier of op het beeldscherm zien, digitaal tot stand zijn gekomen. Dat betekent niet dat er geen mensen aan te pas zijn gekomen, maar dat ze met een computer zijn vervaardigd. Dankzij CAD is het vakmanschap voor het tekenen van print-layouts en (bedradings)schema's volledig verdwenen, samen met de tekentafel, maskeertape, vlakgom en tekenpennen in soorten en maten. Om niet uitsluitend nostalgische redenen vond ik sommige grafische Elektor-kunstwerkjes het waard om hier te reproduceren en te beschrijven in deze sterk grafisch georiënteerde aflevering van Retrotronica. Geniet van de plaatjes!

Na de zoveelste 'reorganisatie' van staf, functies en kantoorruimte bij Elektor afgelopen jaar, en het daarmee gepaard gaande uitmesten van kasten, liet collega Patrick Wielders me drie dikke zware ordners zien van de vroegere grafische afdeling van Elektor. Op miraculeuze wijze hadden deze alle verhuizingen, reorganisaties en inkrimpingen van de grafische afdeling gedurende de afgelopen 25 jaar overleefd, en zelfs de volledige vernietiging van het papieren archief in 2005.

Historisch

Kort na de geboorte in 1961 begon uitgeverij Elektuur speciaal voor het vervaardigen van schema's en bedravingsdiagrammen personeel in dienst te nemen. Deze jongens (en soms ook meisjes) waren *engineers* met een elektronica-achtergrond en intern opgeleid om de standaard met betrekking tot de educatieve aspecten van de schema's hoog te houden. Van aanvankelijk twee of drie in de jaren '60 groeide de afdeling naar tien man (m/v) in 1980, toen niet iedereen meer zo gelukkig was een eigen

bureau, laat staan een eigen tekentafel te hebben! Maar ik moet er meteen aan toevoegen dat onder die tien ook mensen waren voor de layout, een 'cover-ontwerper' en grafici voor illustraties zoals exploded views en ook (laten we eerlijk zijn) wat meer frivole zaken. Hún nalatenschap staat hier centraal.

De stijl van de Elektor-schema's kwam op natuurlijke tot stand maar werd omstreeks 1970 geformaliseerd toen de inmiddels beroemde schemasymbolen en zelfs printsporen en soldeereilandjes werden ontworpen en op afwrijfvelletten werden geprint – speciaal voor intern gebruik. In steen gebeiteld, zogezegd, en tot op de dag van vandaag afgunstig bekijken door de concurrentie. Elke technisch tekenaar werd geacht die afwrijfsymbolen te gebruiken om de door een labmedewerker aangeleverde op papier gekrabbelde schema's om te toveren in een schema in Elektor-stijl dat in het tijdschrift kon worden afgedrukt. De originele schema's werden gemaakt door de componentsymbolen en verbindinglijnen van de Letraset-vellen op

papier te wrijven. Nieuwe componenten werden ter plekke getekend en met de hand ingeplakt. Dit alles doorgaans op een schaal van 200%.

Labmedewerkers en tekstredacteuren werkten gezamenlijk aan een eerste versie van een tijdschriftartikel, en moesten het 'educatief verantwoorde' tekenwerk van de grafische afdeling verifiëren aan de hand van de ruwe schema's van naast de soldeerbout. Fouten werden gevonden (of juist niet), waarna een correctieslag volgde – enfin, dat kunt u zich wel voorstellen. Het hele proces was gebaseerd op de eigen lithografische productie en knip-en-plakwerk met hete bijenwas om alle papier op zijn plek te houden. Tijdens een reorganisatie in 1985 verlieten veel van de layout-medewerkers en typografen de grafische afdeling van Elektor en stapten over naar "Grafisch Bureau Sinke" (GBS), dat Elektor als belangrijkste klant had en binnen loopafstand gevestigd was. Elektor hield echter wel zijn eigen tekenaars en cover-ontwerpers in dienst.

Opleuken van de artikelen

Ik was blij te ontdekken dat de drie ordners honderden 200%-originelen bevatte van de meer artistieke tekeningen die gedurende de periode 1980–1990 (schat ik) zijn vervaardigd. Het doel hiervan

ESTD 2004

www.elektor.tv



Retrotronica is een rubriek over oude elektronica en legendarische Elektor-ontwerpen. Blijven, suggesties en vragen zijn meer dan welkom; stuur uw telex of telegram naar redactie@elektor.nl

was Elektor te verlevendigen en een wat luchtiger element toe te voegen aan de voor beginners vaak moeilijk te begrijpen materie. De schemasymbolen en een aantal 3D-tekeningen van componenten en elektronische schakelingen werden voor alle Elektor-publicaties gemaakt, dus voor de boeken en voor de tijdschriften in alle talen, maar de redacteur had het laatste woord. Het grootste deel van de handgemaakte tekeningen, de echte 'artist impressions', werd gebruikt voor het tijdschrift 'Elex', dat was gericht op jonge Duitse, Nederlandse – en later ook Franse – lezers en dat tussen 1980 en

1992 verscheen. Vanwege de wat speelse doelgroep moest de boodschap door de Elex-redactie op een wat ongedwongener manier worden gepresenteerd dan door de Elektor-redactie, die werd geacht zich op een 'hoogstaander' publiek te richten, afhankelijk van de betreffende taal-editie. Van de wat minder serieuze tekeningen die in een periode van ca. 20 jaar zijn geproduceerd, hadden de cartoons de meeste impact.

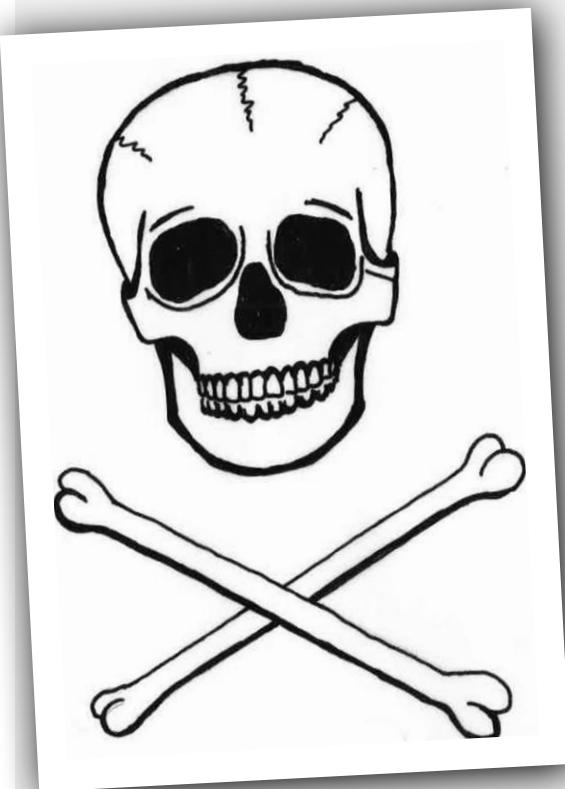
In een poging te achterhalen wie al dat moois heeft getekend, ben ik niet verder gekomen dan Laurent Martin voor de 3D-componenten en bouwtekenin-

gen, en Jaap Kuiper voor de cartoons, alles met de hand in schitterend zwart-wit getekend met een hele verzameling Rotring-pennen waarvan ik me het tikende geluid herinner als de tekenaar de inkt in de pen losschudde. In sommige tekeningen kun je nog de handtekeningen 'LM' of 'JAAP' ontwaren. Chapeau!



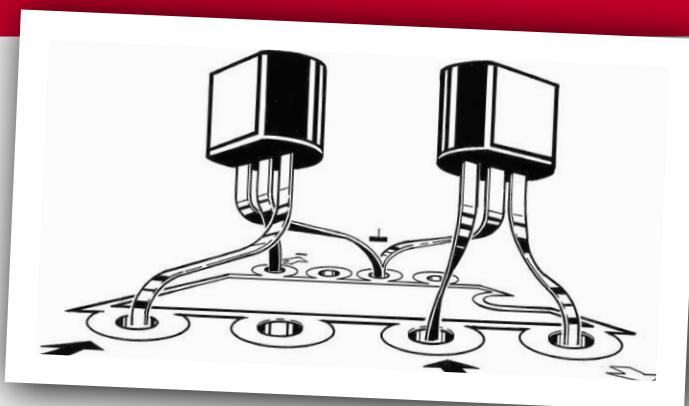
(180574-B-03)

De componenten

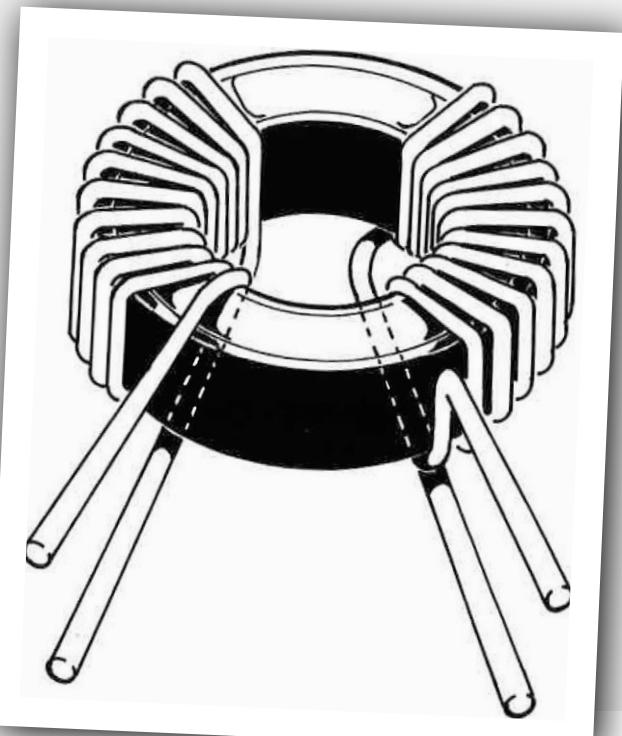


Regelmatig vond Elektor het nodig lezers te waarschuwen voor het risico van acuut levensgevaar of ander ongemak, als gevolg van in schakelingen voorkomende hoge spanningen.

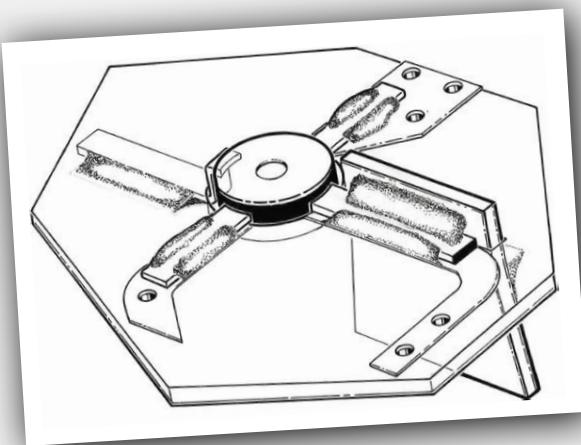
Spoelen en het zelf wikkelen daarvan was zwarte kunst voor veel Elektor-lezers die liever met DC te maken hadden, in tegenstelling tot radioamateurs, die dol waren op 'de L'. Ook nadat de ferrietkern zijn intrede had gedaan, bleef de wikkelaanwijzing belangrijk en dus werden tekeningen gemaakt om duidelijk te maken hoe het moest.



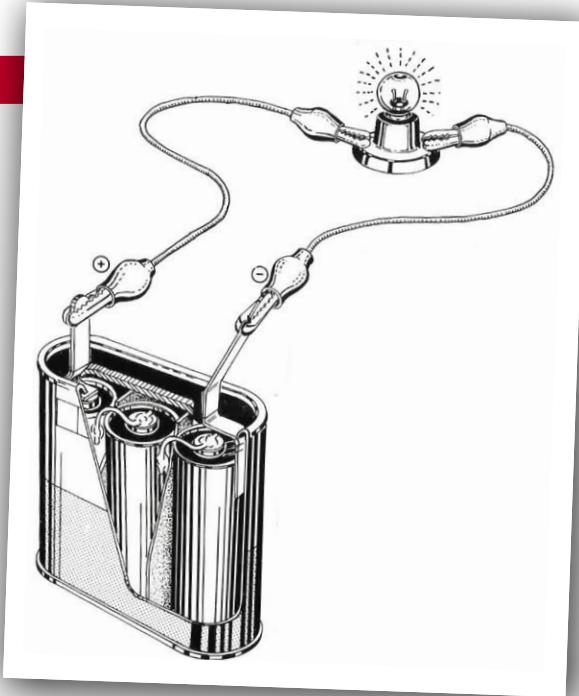
Elegant als een danspaartje, nietwaar? Hoe monter je twee TUN's (of TUP's) in wat wel wat lijkt op de footprint van een IC. Bedoeld als stroombron, stroomspiegel, of een armeluis-Darlington? Wie het weet mah het zeggen!



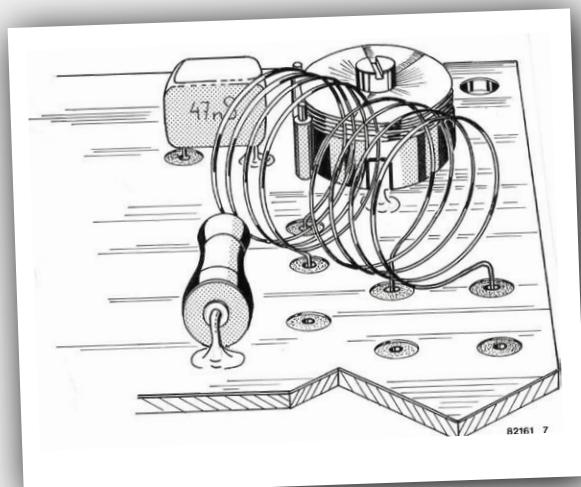
Bouwtekeningen in 3D



De BF9xx-serie dual-gate MOSFET's verbijsterde veel lezers, omdat ze verzonken in een gat in de print moesten worden gemonteerd. Om de verwarring nog groter te maken, schreef Elektor een trapeziumvormige 'draadloze' maket, ontkoppelcondensator voor, te monteren in een sleuf in de print, direct op de source-aansluiting. Help! Deze artist impression scheelde de redacteur pakweg 100 woorden, maar kon toch niet echt overtuigen.

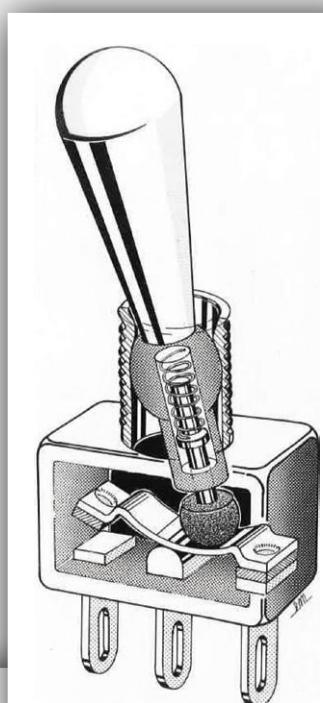


Een 'typisch' plaatje dat aan het begin van menige carrière in de elektronica moet hebben gestaan: een 6-V-lampje laten branden op een 4,5-V-platte batterij. Let op de krokodilleklemmen en de lampvoet! Deze afbeelding is gebruikt in hoofdstuk 1 van de immens populaire '300'-boekenreeks die door Elektor is uitgebracht in de jaren '70 en '80. Ook tegenwoordig nog zou elke CEO en CFO in de elektronica-industrie dit experiment zelf een keer moeten uitvoeren...

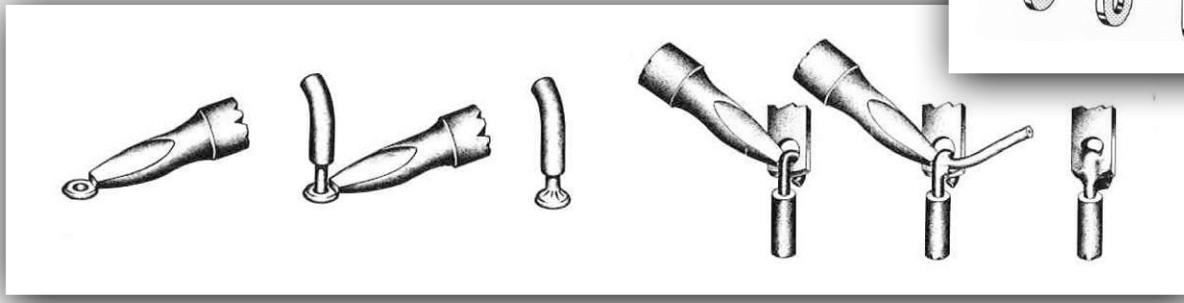


Als je aan een hobbyist iets moet uitleggen over 6 pF PTFE (Teflon) trimmers, gekoppelde spoelen en ontkoppelcondensatoren, dan zegt een plaatje meer dan 1000 woorden.

Het geheim van de wipschakelaar onthuld. Zoals u wellicht zelf wel eens hebt ontdekt bij het foutzoeken in eigen bouwsels: de contacten sluiten **niet** aan de kant waar de handle naartoe beweegt...
Hier ziet u waarom.

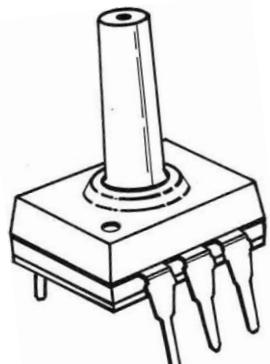
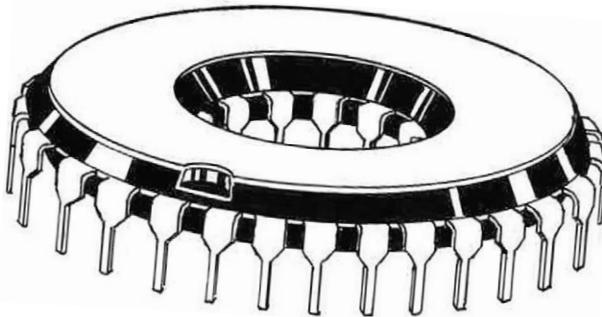


Een verzameling goede en slechte manieren om een draad in een soldeereiland of aan een soldeeroog te solderen (volgens Elektor). Wat zou de industrie hiervan hebben gevonden?



Waar of niet?

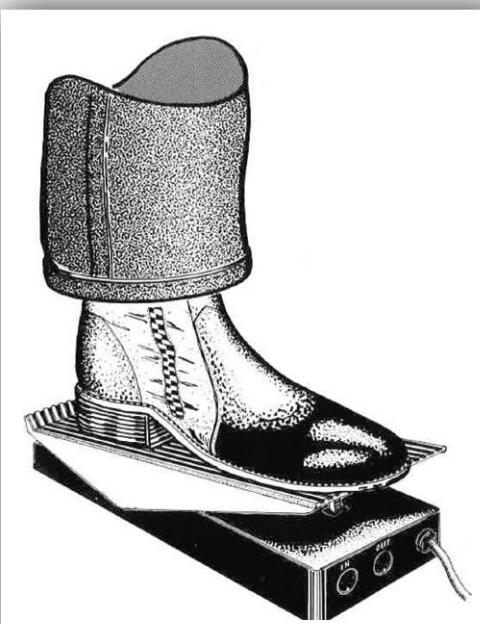
Toen surround sound werd geïntroduceerd, schreeuwden hobbyisten ernaar om hun eigen systemen te kunnen bouwen. Het was in veel gevallen onmogelijk om aan de IC's te komen, maar met wat vindingrijkheid en vooral veel fantasie kon men met wat duwen en trekken wel een 'sound' worden gefabriceert met een surround-achtig effect.



Elektor had plotseling de smaak van drucksensoren in IC-vorm te pakken gekregen, en dit kon wel eens één van de allereerste zijn geweest. In elk geval had het mondstuk bovenop het IC een hoge artistieke waarde. Later was deze tekening de inspiratiebron voor een fake-tekening van de 'CD4711 programmeerbare geurgenerator', één van Elektor's meest succesvolle grafischakelingen, opgenomen in de toen nog jaarlijks verschijnende Halfgeleidergids.

Het muziekwereldje

Schakelingen en projecten om de weergave van audio en muziekinstrumenten te ondersteunen, zijn decennialang een sterk, om niet te zeggen dominant element geweest in Elektor. Het naar een droog elektronica-tijdschrift trekken van 'buitenstaanders' vereiste plaatjes waarin ze zich konden herkennen. Dat wordt 'aas' genoemd. In het geval van Elektor heeft geen onderwerp zoveel 'buitenstaanders' de elektronica-hobby in getrokken als audioversterkers, in het bijzonder hiFi. Helaas kwamen hier naast audiophilen, ook nogal wat 'audioten' (audio-idioten) uit voort.



In de jaren '80 bouwde iedereen gitaar-effectapparaten, waarschijnlijk om indruk op het andere geslacht te maken en het volslagen gebrek aan eigen talent te camoufleren. Hier een voorbeeld van zo'n hip voetpedaal, naar ik vermoed een wah-wah pedaal. Opmerkelijk: geen broeken met van die wijde pijpen...



De cartoons

Het uitleggen van een cartoon is heiligschennis, maar dat is ook niet nodig want de meeste zijn voor de Nederlandse markt gemaakt. Disclaimer: elke gelijkenis tussen in de cartoons afgebeelde personen en de Elektor-redactie is puur toeval.

Hmm, hoe gaan we deze partituur op een elektronisch instrument spelen?

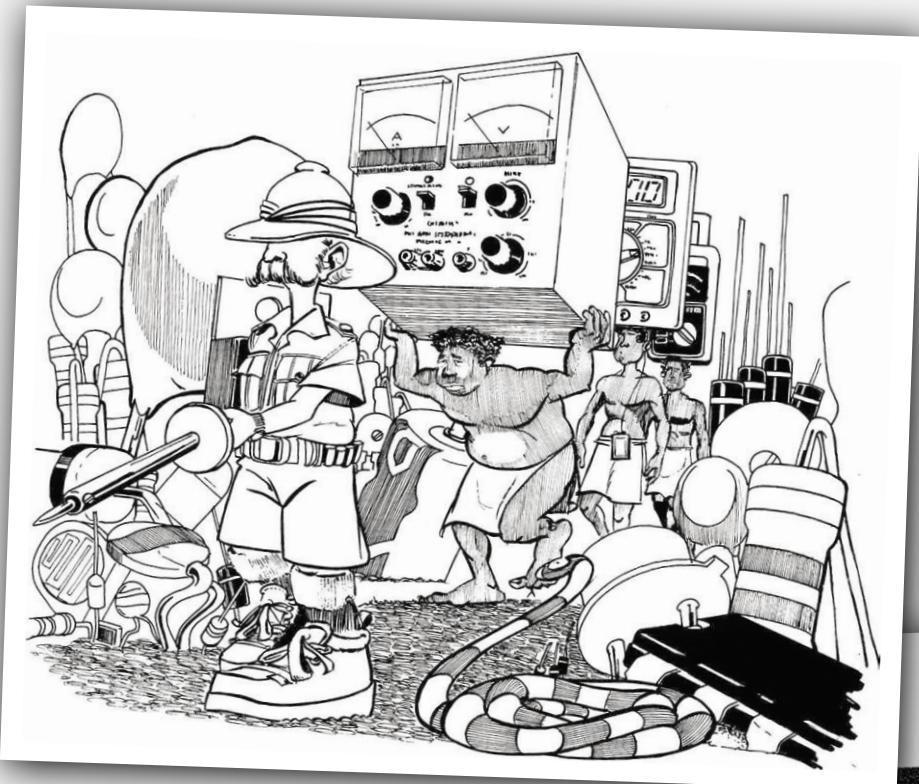


De Video Splitter!

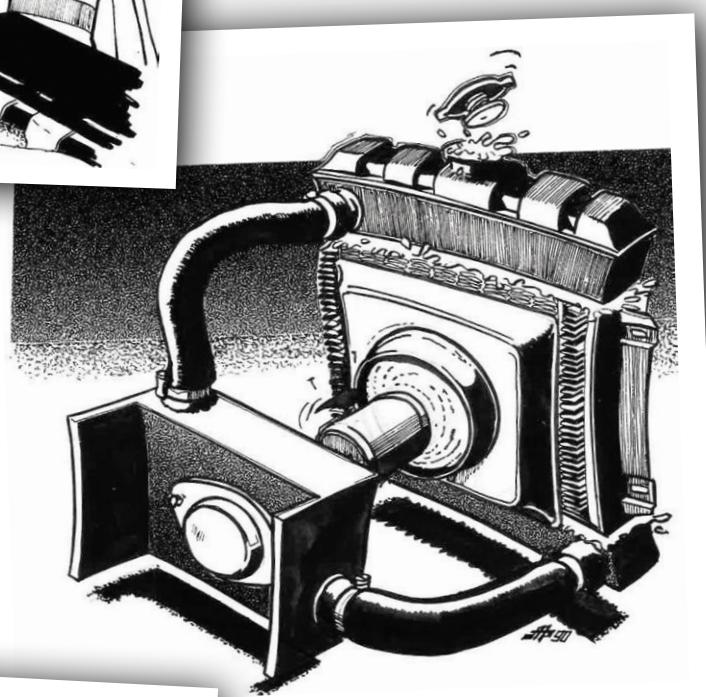


Looplichten!





Het moet gezegd: om elektronica te onderzoeken en de baas te worden, is moed, solderen en meetapparatuur vereist!



Koelen van een vermogenstransistor met behulp van een radiateur...



Bakken, braden en opblazen van componenten en apparaten is... soms leuk, maar eigenlijk verontrustend en onwenselijk. Zo is het bijvoorbeeld een minder goed idee om een multimeter in de ohmstand te zetten als je vette stromen aan het meten bent!



Hoe implementeer je ethische waarden concreet in je project?

Tessel Renzenbrink (Nederland)

Gelukkig is er steeds meer aandacht voor de ethische kant van technologie. Maar hoe maak je ethische waarden een praktisch onderdeel van je project? Die vraag stond centraal op een bijeenkomst in Amsterdam. Drie sprekers deelden hierover hun ervaringen. Gijs Boerwinkel van de Waag vertelde hoe je de ethische waarden kunt bepalen als je te maken hebt met een grote, gevarieerde gebruikersgroep. Tada-bestuurslid Douwe Schmidt ging in op het verankeren van de waarden. Het Tada-manifest over verantwoord gebruik van data werd opgenomen in het coalitieakkoord van de Amsterdamse gemeenteraad. En Mart van Santen, CTO van Greenhost, vertelde dat het bewaken van principes niet alleen betekent dat je kritisch naar je eigen bedrijf kijkt maar ook dat je met je ketenpartners in gesprek gaat.

Hoe bepaal je de ethische waarden?

Om je project te sturen met betrekking tot ethische waarden, moet je natuurlijk eerst bepalen wat die waarden zijn. Eén manier om dat te doen is luisteren, naar zo veel mogelijk mensen luisteren. Dat is een van de methoden die het Digitale Identiteitslab [1] heeft toegepast. Het Lab is een langlopend project van de Waag, een kenniscentrum dat burgers en experts betrekt bij het vormgeven van technologie.

Het Lab doet onderzoek naar digitale identiteit, vertelde Gijs Boerwinkel, community manager bij de Waag. Digitale identiteit speelt een steeds grotere rol in ons leven. Van het gebruik van DigiD om in te loggen bij onze zorgverzekeraar tot de tekstberichten die we sturen naar onze liefsten. Allemaal maakt het deel uit van onze digitale identiteit. Op dit moment hebben we daar weinig grip op. Onze digitale identiteit is versnipperd opgeslagen in databanken van overheden en bedrijven. We weten niet wat er gebeurt met de reisgegevens die worden vastgelegd via de OV-chipkaart, of met wie Facebook onze gegevens deelt. In opdracht van het ministerie van Binnenlandse Zaken en de Vereniging van Nederlandse Gemeenten doet het Lab onderzoek naar methoden en toepas-

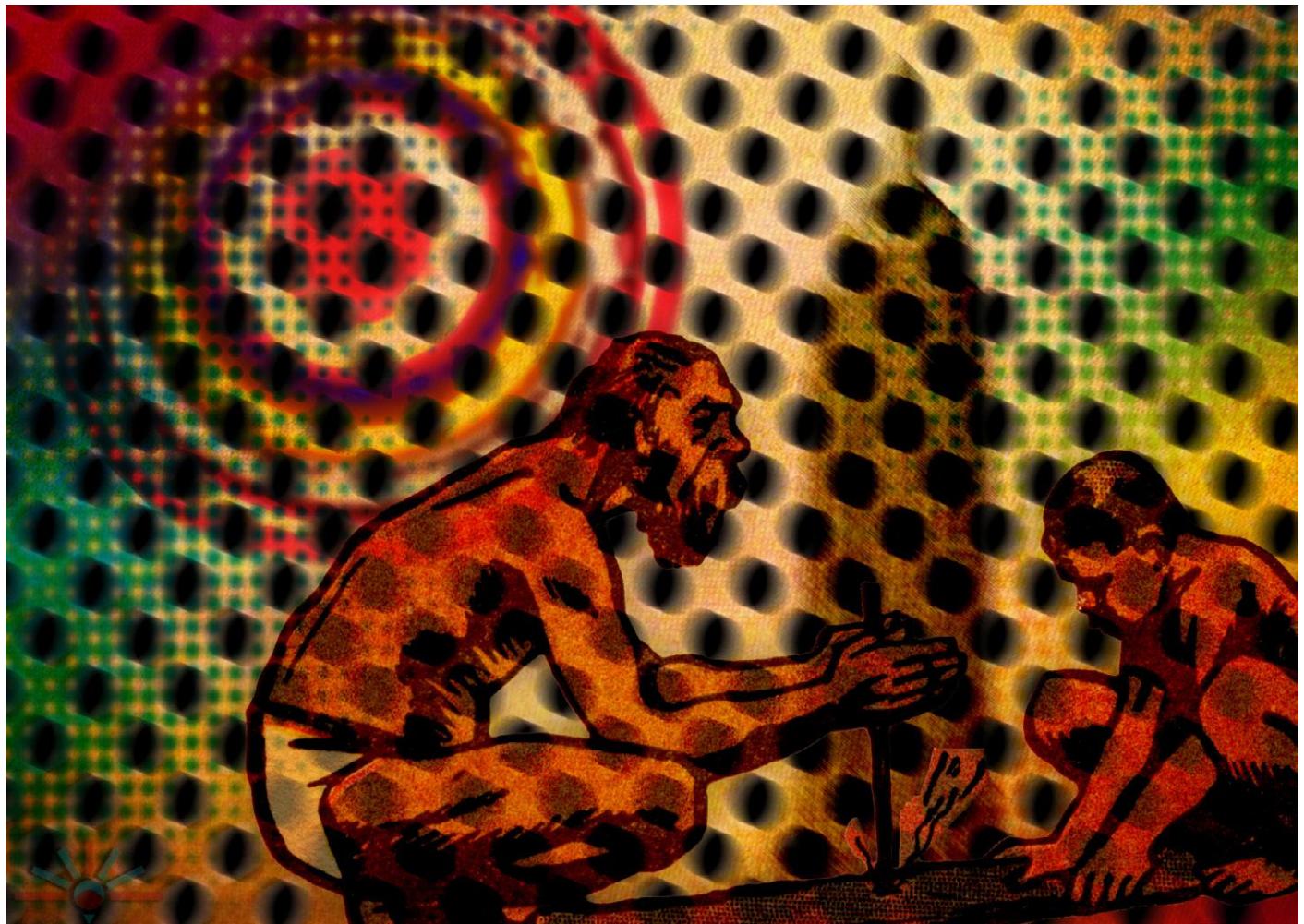
singen waarmee we meer grip krijgen op onze online identiteit. Denk bijvoorbeeld aan een app waarmee je online je identiteit kunt bewijzen zonder dat je een kopie van je paspoort hoeft af te geven. Het Lab test of dit soort apps voldoen aan de wensen van gebruikers. Maar daarvoor moet eerst achterhaald worden wat gebruikers willen. Hiertoe organiseerde het Lab bijeenkomsten in verschillende delen van het land. Niet alleen met experts en software-ontwikkelaars maar vooral met gewone burgers die de apps zullen gaan gebruiken. Ook gingen ze de straat op om voorbijgangers te interviewen. Met die input van een dwarsdoorsnede van de Nederlandse bevolking kon het Lab de waarden identificeren die burgers belangrijk vinden als het gaat om hun online-identiteit. De waarden die naar voren kwamen waren onder meer transparantie (inzicht in gegevens) en controle (zeggenschap over gegevens). Dat zijn voor de hand liggende waarden in deze context. Maar een andere veelgenoemde waarde was gebruiksgemak. Dat laatste zou in een expertmeeting met app-ontwikkelaars en privacy-activisten misschien wel over het hoofd worden gezien. Om blinde vlekken te voorkomen heeft het Lab zoveel mogelijk belanghebbenden betrokken bij het formuleren van de waarden. Terug-

komend op de vraag hoe je de ethische waarde voor een project bepaalt: het is zinvol daar niet alleen zelf over na te denken, maar ook naar anderen te luisteren.

Hoe implementeer je de waarden?

Hoe je die waarden vervolgens concretiseert, kwam aan de orde bij de besprekking van het project Tada [2]. Tada gaat over het gebruik van data in de stad. Met de opkomst van de smart city worden data steeds belangrijker. En dat is een goede zaak, stellen de initiatiefnemers van Tada. Data kunnen een bijdrage leveren aan een schonere, efficiëntere, meer democratische stad. Maar dan moet het gebruik van data wel gestoeld zijn op bepaalde ethische waarden.

Tada is een initiatief van zo'n vijftig organisaties waaronder de Amsterdam Economic Board en (alweer) de Waag, vertelde Douwe Schmidt, bestuurslid van Tada. Net als het Lab organiseerde Tada een aantal bijeenkomsten met belanghebbenden om de ethische waarden te bepalen. Dit mondde uit in een heus manifest met zes principes waaraan het gebruik van data in de stad moet voldoen. Eén van de principes is bijvoorbeeld inclusiviteit: het gebruik van data mag niet leiden tot uitsluiting van bepaalde groepen.



Illustratie: Dawn (A.K. Rockefeller). Creative Commons licentie BY-SA 2.0 / Flickr: <https://is.gd/Gbbqwe>.

De volgende stap is het implementeren van de waarden. ‘Hoe zorg je ervoor dat het manifest niet onderin de la verdwijnt?’, zei Schmidt. ‘Om ervoor te zorgen dat mensen ook daadwerkelijk iets met het manifest doen, zijn we een campagne gestart. We roepen iedereen op het manifest te ondertekenen: burgers, bedrijven, maatschappelijke organisaties enzovoort. Data in de stad gaat namelijk niet alleen de gemeente aan. Iedereen heeft er mee te maken. Daarnaast hebben we ook workshops georganiseerd voor gemeenteraadsleden.’ Met succes. Het Amsterdamse gemeentebestuur dat in 2018 aantrad, heeft het manifest opgenomen in het coalitieakkoord.

Hoe geef je uitvoering aan de waarden?

Mart van Santen, CTO en mede-oprichter van webhosting-provider Greenhost [3], vertelde hoe je ethische waarden een blijvend onderdeel van je project maakt. Greenhost heeft duurzaamheid tot een leidend principe van zijn bedrijfsvoering gemaakt. Internettechnologieën verbruiken veel elektriciteit. Om de milieu-impact zoveel mogelijk te beperken, kiest Greenhost voor groene elektriciteit en zuinige apparatuur. Dat is niet altijd makkelijk, want een server moet niet alleen duurzaam zijn maar ook aan bepaalde technische eisen voldoen. ‘Het is steeds zoeken naar een balans’, zei Van Santen. Een verantwoorde bedrijfsvoering beperkt zich niet alleen tot de eigen bedrijfsprocessen, je moet ook kritisch kijken naar de rest van de productieketen. Greenhost koos voor een datacenter dat claimde volledig op groene stroom te draaien. Maar op een gegeven moment bleek dat het datacenter deels werkte met sjoemelstroom: stroom die weliswaar legaal groen mag worden genoemd maar die niet bijdraagt aan het vergroten van de hernieuwbare elektriciteitsproductie. Greenhost sprak hen daarop aan. Het datacenter bleek al bezig te zijn met een verbeteringsslag door over te stappen op Europese windenergie. Daarnaast beloofden ze voor het aandeel elektriciteit dat Greenhost verbruikt Nederlandse windenergie in te kopen.

Een verantwoorde bedrijfsvoering beperkt zich niet alleen tot de eigen bedrijfsprocessen, je moet ook kritisch kijken naar de rest van de productieketen. Greenhost koos voor een datacenter dat claimde volledig op groene stroom te draaien. Maar op een gegeven moment bleek dat het datacenter deels werkte met sjoemelstroom: stroom die weliswaar legaal groen mag worden genoemd maar die niet bijdraagt aan het vergroten van de hernieuwbare elektriciteitsproductie. Greenhost sprak hen daarop aan. Het datacenter bleek al bezig te zijn met een verbeteringsslag door over te stappen op Europese windenergie. Daarnaast beloofden ze voor het aandeel elektriciteit dat Greenhost verbruikt Nederlandse windenergie in te kopen.

De bijeenkomst ‘Principes in technologie, hoe maak je ze concreet?’ [4] vond plaats op 12 december 2018 in Amsterdam. De organisatie was in handen van het Digitale Identiteitslab, Greenhost en Tada. ■

180572-B-01

Weblinks

- [1] Digitale identiteitslab: <https://digitaleidentiteit.waag.org/>
- [2] Tada: <https://tada.city/>
- [3] Greenhost: <https://greenhost.nl/>
- [4] Bijeenkomst ‘Principes in technologie’: <https://digitaleidentiteit.waag.org/event/principes-in-technologie-hoe-maak-je-ze-concreet/>



welkom in je **WEB STORE**

TIP VAN DE REDACTIE



MonoDAQ-U-X – Multifunctioneel USB-data-acquisitiesysteem

Met zijn galvanisch gescheiden USB-C-poort en robuuste aluminium behuizing vormt de MonoDAQ-U-X een stevig en veelzijdig apparaat voor data-acquisitie. Er kunnen meerdere apparaten worden aangesloten op dezelfde computer.

De MonoDAQ-U-X beschikt over acht poorten die individueel configurerbaar zijn als analoge of digitale input of output. De analoge inputs kunnen ook worden gegroepeerd om differentiële inputs te vormen. Naast deze acht configurerbare poorten is er een bidirectionele digitale poort beschikbaar die het 1-Wire-protocol ondersteunt. Ook een analoge output voor het exciteren van sensoren of het voeden van een subsysteem (maximaal 24 V, 1 W) is niet vergeten. Verder heeft de MonoDAQ-U-X een geïntegreerde temperatuur- en luchtvochtigheidssensor en een 6-assige IMU.

Clemens Valens (Elektor Labs)

www.elektor.nl/monodaq



Weller WT 1014 digitaal soldeerstation (90 W)



Het Weller WT 1014 soldeerstation bestaat uit de voedingsunit WT 1, de soldeerbout WSP 80 Robust en de standaard WSR 201. Het is stapelbaar en bespaart daardoor ruimte in de werkplaats. Dankzij de geïntegreerde gebruikssensor schakelt het soldeergereedschap automatisch uit.



Ledenprijs: € 350,10

www.elektor.nl/weller-wt-1014

PeakTech 8200 Measuring Accessories Set



Met deze uitgebreide set van meetaccessoires voor de elektronica beschikt u altijd over de meest geschikte meetadapter voor uw toepassing en uw digitale multimeter. Alle klemmen en kabels zijn gefabriceerd volgens de laatste veiligheidsstandaards, zodat de gebruiker in elke toepassing veilig kan werken. Het gebruikte materiaal is een buigzaam en slipvrij plastic dat bestand is tegen ruw gebruik.



Ledenprijs: € 40,46

www.elektor.nl/peaktech-8200

Elektor Bestsellers

1. Miniware TS100 soldeerbout
www.elektor.nl/miniware-ts100



2. Android App Development for Electronics Designers
www.elektor.nl/android-app-development
3. Mastech MS8911 slimme SMD-tester
www.elektor.nl/mastech-ms8911
4. Raspberry Pi 3A+
www.elektor.nl/rpi-3a-plus
5. PoE-HAT voor Raspberry Pi 3B+
www.elektor.nl/poe-hat-rpi-3-plus
6. Bakeey UM25C USB Multimeter Logger
www.elektor.nl/bakeey-um25c

JoyPi – Experimenteerkoffer voor de Raspberry Pi



De JoyPi is een experimenteerkoffer die is opgebouwd rondom een Raspberry Pi en is een ideale start voor wie wil beginnen met elektronica en programmeren. De geavanceerde opzet biedt een perfecte alles-in-één omgeving en maakt een einde aan het gepruts met kleine onderdeeltjes en de chaos van kabels op de werkbank. De JoyPi bevat een groot aantal sensoren en modules die zonder ervaring in de elektronica te gebruiken zijn.



Ledenprijs: € 224,10

www.elektor.nl/joypi



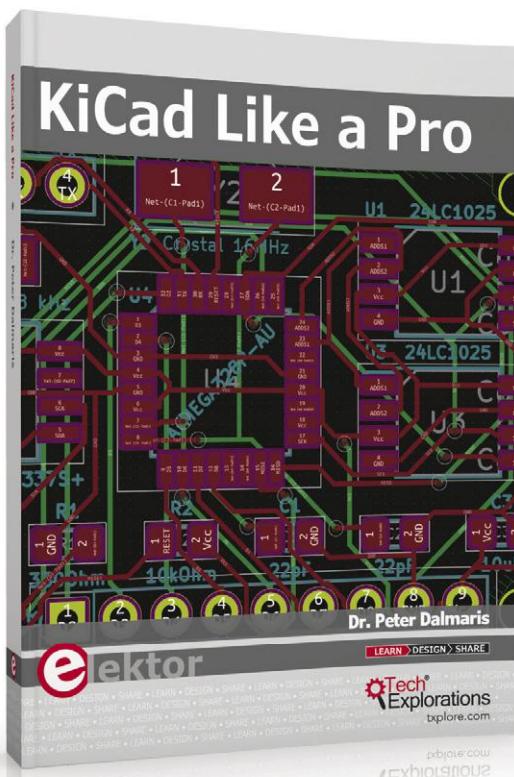
SHOPPING

BOEKEN

ONTWIKKEL TOOLS

PROJECTEN

CD/DVD

**nieuw**

KiCad Like a Pro

Dit Engelstalige boek leert u KiCad te gebruiken. Of u nu 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.

Zowel de basisfuncties als de meer geavanceerde mogelijkheden van KiCad komen aan de orde. Als u nog geen enkele ervaring hebt met het ontwerpen van printen, kunt u bij het inleidende project de basiskennis opdoen. Daarna kunt u doorgaan met de rest van de projecten. U ontwerpt een print voor een breadboard-voeding, een kleine Raspberry Pi HAT en een Arduino-kloon met chips voor een uitgebreid geheugen en een real-time klok.



Ledenprijs: € 35,96
www.elektor.nl/kicad-like-a-pro

JOY-iT DPS5005 programmeerbaar labvoeding



Deze labvoeding combineert analoge en digitale technologie in een geavanceerd ontwerp en biedt een opmerkelijke nauwkeurigheid. Hij levert maximaal 50 V of 5 A en kan exact worden ingesteld in stappen van 10 mV of 1 mA. De bediening is geoptimaliseerd om eenvoudig mee te kunnen werken en alle informatie wordt gedetailleerd weergegeven op het kleurendisplay.



Ledenprijs: € 38,66

www.elektor.nl/dps5005

Electronics for Space



De ruimtevaartindustrie groeit voortdurend en er zijn steeds nieuwe producten en diensten nodig. Er is innovatie nodig voor het ontwikkelen van deze industrietak. Het is tegenwoordig onmogelijk om alle ontwikkelingen op het gebied van de ruimtevaart te volgen. De markt groeit enorm en er zijn steeds meer activiteiten, vooral op het gebied van kleine satellieten. Dit boek praat u bij over de stand van zaken.



Ledenprijs: € 22,46

www.elektor.nl/electronics-for-space

NixieCron-M4 – viercijferige LED Nixie-klok



De NixieCron M4 werkt met een zeer nauwkeurige klokchip met temperatuurcompensatie met een afwijking van slechts enkele seconden per maand. Hij bevat ook een geluidsmodule, die dient als slagwerk en hij kondigt de tijd aan als u op een knop drukt. U kunt hem ook verbinden met de DS18B20-temperatuursensor en hem configureren met een draaknop.



Ledenprijs: € 134,10

www.elektor.nl/nixiecron-m4

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 getallen 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 (ge марkeerd door de dikke zwarte lijnen).

Een aantal cijfers is al aangegeven en deze bepalen de uit-

gangssituatie voor de puzzel.

Onder de inzenders met de goede oplossing verloten we **vijf Elektor-waardebonnen**, elk ter waarde van **50 Euro**.

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 18 maart 2019** naar: hexadoku@elektor.nl

De prijswinnaars

De juiste oplossing van de Hexadoku uit het januari/februari-nummer is: **6A258**.

De Elektor-waardebonnen van 50 Euro zijn gewonnen door Siegfried Keep (Duitsland), Mehmet Can Cakar (Turkije), Per Troelsen (Denemarken), Lucien Zadikian (Frankrijk) en Vladimir Koci (Tsjechië).

Allemaal van harte gefeliciteerd!

	0		E	9	3	D		4							
	3	2			6	9			B	7					
D	4			1				2			A	E			
	8		F	2	4	6	A		7			9			
		3	5	F	A	C	2	1		7					
1		F	3	E			6	9		C	0				
		B	6	7	9		8	A	E	3					
7	9	A	1					0	E		4	6			
9	6	7	A					2	4		C	1			
		C	7	8	6		4	9	A	0					
3		A	0	1		D	5		8		2				
		5	F	4	2	E	C	3		9					
1			9	5	F	2	B	3			6				
4	7			6			E				2	A			
	C	6			1	8			D	0					
	E		D	7	A	6			3						

3	8	5	A	C	9	E	F	7	1	6	0	D	2	B	4
6	7	F	1	A	D	B	4	8	3	2	9	E	5	0	C
9	4	E	B	2	0	1	7	D	5	A	C	F	6	3	8
0	C	D	2	3	5	6	8	E	F	4	B	7	1	9	A
7	A	C	8	6	B	D	E	9	4	F	2	1	3	5	0
2	5	0	E	4	3	8	9	1	B	C	6	A	7	D	F
F	9	B	D	1	C	7	0	3	E	5	A	6	4	8	2
1	6	3	4	F	2	5	A	0	7	D	8	9	E	C	B
B	E	1	9	0	8	4	5	2	C	7	F	3	D	A	6
A	F	2	0	D	1	3	C	4	6	B	E	5	8	7	9
4	D	7	C	9	6	A	2	5	8	1	3	B	0	F	E
5	3	8	6	7	E	F	B	A	9	0	D	2	C	4	1
8	1	9	F	5	7	C	D	6	A	E	4	0	B	2	3
C	0	6	5	8	A	9	1	B	2	3	7	4	F	E	D
D	B	A	7	E	4	2	3	F	0	8	1	C	9	6	5
E	2	4	3	B	F	0	6	C	D	9	5	8	A	1	7

Medewerkers van Elektor International Media en hun familieleden zijn van deelname uitgesloten.



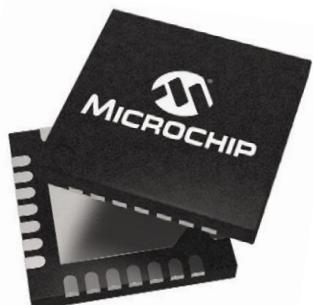
Ontdek uw voeding

Flexibiliteit bij het kiezen van de gewenste voedingsoplossing

Als toonaangevende leverancier met een uitgebreid assortiment voedings- en beheerproducten biedt Microchip u het vermogen, de flexibiliteit en het vertrouwen om de juiste oplossing te kiezen voor uw ontwerp.

Het beheren van de voeding van uw systeem is cruciaal om dit de gewenste prestaties te laten halen. Met ons programma voedingsbeheercomponenten kunt u nauwkeurig het actieve, reactieve en schijnbare vermogen, de effectieve spanning en stroom, de netfrequentie en de arbeidsfactor meten. U kunt een keus maken uit ons brede spectrum aan voedingsbeheercomponenten, waaronder gelijkspanningsomzetters en spanningsregelaars, MOSFET's en MOSFET stuurratten, spanningsbewakers en referentiebronnen, alsook voedingsmodulen waarmee u efficiënt een oplossing kunt ontwerpen die voldoet aan de voedingseisen van uw systeem.

Aan de hand van referentie-ontwerpen via evaluatiekaarten tot en met simulatiehulpmiddelen beperkt u de ontwikkeltijd en risico's met de omvangrijke ondersteuning die Microchip kan bieden.



Ontdek uw voeding op
www.microchip.com/PowerSolutions



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

Productronica Fast Forward is brought to you by

Meer info:
www.elektormagazine.nl/p-ffwd

