



MEI/JUNI 2025 | ELEKTORMAGAZINE.NL

€ 9,95

elektor

MEDIA & LEARNING

DIGITALE

BONUS

EDITIE

695B

SINDS 1961

GRATIS
dankzij
MOUSER
ELECTRONICS

FOCUS OP

Meten & Testen

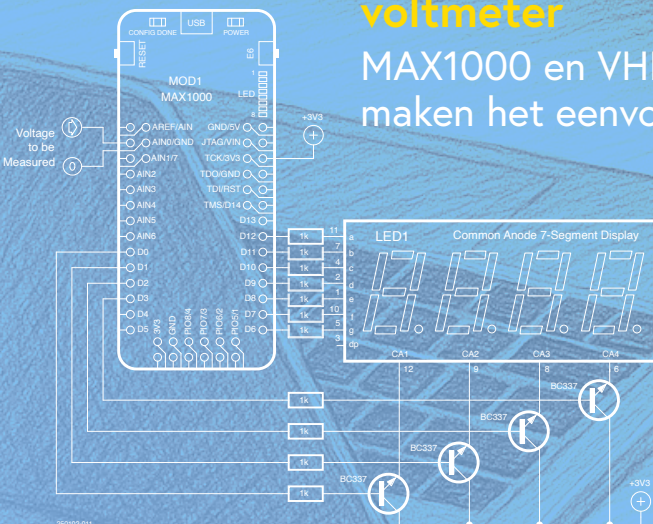
PWM-meting met een PIC

Meet de aan/uit-verhouding van een stuursignaal



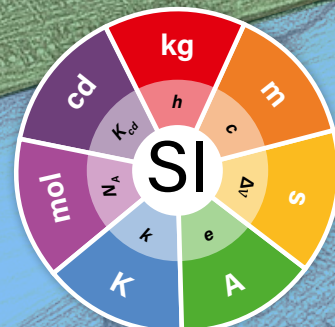
FPGA-gebaseerde voltmeter

MAX1000 en VHDLPlus maken het eenvoudig



Oscilloscoop/multimeter/generator

Fnirsi 2C53T met twee kanalen en 50 MHz bandbreedte



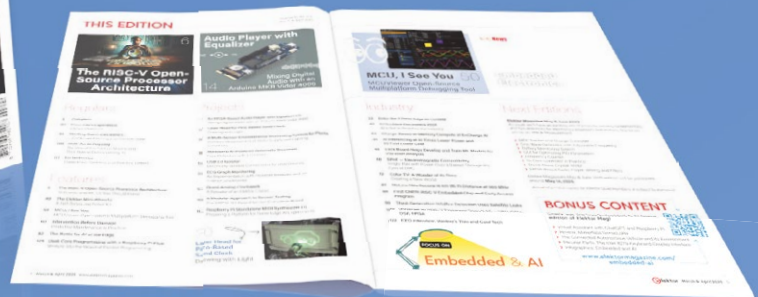
Infographics: Testen & Meten



Word lid van de Elektor Community



Neem **nu** een
lidmaatschap!



- ✓ Toegang tot het compleet web-archief t/m 1960!
- ✓ 8x Elektor Magazine (Print)
- ✓ 8x digitaal (PDF)
- ✓ 10% korting in de Elektor Store, en exclusieve aanbiedingen
- ✓ Toegang tot meer dan 5000 Gerberfiles



Ook verkrijgbaar

Het digitale
lidmaatschap!



- ✓ Toegang tot het compleet web-archief
- ✓ 10% korting in de Elektor Store
- ✓ 8x Elektor Magazine (PDF)
- ✓ Toegang tot meer dan 5000 Gerberfiles



www.elektormagazine.nl/abo

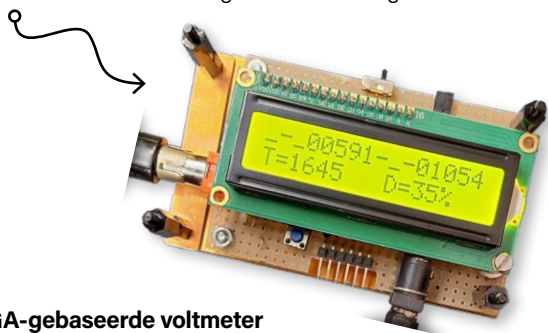


IN DIT NUMMER

3 Colofon

4 PWM-meting met een PIC

Meet de aan/uit-verhouding van een stuursignaal



12 FPGA-gebaseerde voltmeter

MAX1000 en VHDPlus maken het eenvoudig

20 Infographics: Testen en Meten

22 Oscilloscoop/multimeter/generator

Fnirsi 2C53T met twee kanalen en 50 MHz bandbreedte

De mei/juni-editie van ElektorMag 2025 is nu te koop bij de tijdschriftenwinkel en in de Elektor Store.



C. J. Abate

Content Director, Elektor

Testen en meten op uw werkbank

Test- en meettoepassingen zijn super belangrijk voor het debuggen, controleren en verbeteren van elektronische ontwerpen. In deze speciale editie laten we je praktische en toegankelijke tools zien die ingenieurs, studenten en makers meteen op hun werkbank kunnen gebruiken.

In dit nummer wordt een slimme aanpak van signaalanalyse beschreven in 'PWM-meting met een PIC'. Lees het artikel om te leren hoe je een PIC16F628 kunt gebruiken om PWM-pulsduur te meten en hoge/lage tijden en volledige periodegegevens nauwkeurig vast te leggen.

We duiken ook in de fascinerende wereld van FPGA-programmering. In 'FPGA-gebaseerde voltmeter' ontdek je hoe je een voltmeter bouwt met behulp van de VHDPlus-omgeving en de MAX1000 FPGA-kaart. Dit is een geweldig project voor iedereen die geïnteresseerd is in het combineren van theorie en praktijk in digitaal ontwerp.

Wil je je tools graag samenvoegen? Bekijk dan eens de 2C53T. We geven je een kijkje in dit compacte 3-in-1-instrument dat werkt als een oscilloscoop, multimeter en functiegenerator.

Deze bonusuitgave staat helemaal in het teken van het verbeteren van je elektronica-workflow. Of je nu signalen aan het afstemmen bent, circuits aan het debuggen bent of gewoon nieuwe testtools aan het ontdekken bent, wij hebben wat je zoekt.

Veel leesplezier en deel je creaties op het Elektor Labs-platform!

Ons team

Internationaal hoofdredacteur: Jens Nickel | **Content Director:** C.J. Abate | **Internationale redactie:** Hans Adams, Asma Adhimi, Roberto Armani, Eric Bogers, Jan Buiting, Rolf Gerstendorf (RG), Ton Giesberts, Saad Imtiaz, Alina Neacsu, Dr. Thomas Scherer, Jean-Francois Simon, Clemens Valens, Brian Tristram Williams | **Vaste medewerkers:** David Ashton, Stuart Cording, Tam Hanna, Ilse Joostens, Prof. Dr. Martin Ossmann, Alfred Rosenkränzer | **Vormgeving & Layout:** Harmen Heida, Sylvia Sopamena, Patrick Wielders | **Directeur:** Erik Jansen | **Technische vragen:** redactie@elektor.com

65^e jaargang nr. 537B
mei/juni 2025 bonus editie

Elektor verschijnt acht keer per jaar
en is een uitgave van

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

www.elektor.nl | www.elektormagazine.nl

Voor al uw vragen: service@elektor.nl

Lid worden: www.elektormagazine.nl/abo

Advertenties

Raoul Morreau
Tel. +31 (0)6 4403 9907 – raoul.morreau@elektor.com
www.elektormagazine.nl/adverteren

Auteursrecht

© Elektor International Media B.V. - 2025

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 Rijksop-trooiwet mogen de in Elektor opgenomen schakelingen slechts voor particuliere of wetenschappelijke doeleinden vervaardigd worden en niet in of voor een bedrijf. Het toepassen van de schakelingen geschiedt buiten de verantwoordelijkheid van de uitgever. De uitgever is niet verplicht ongevraagd ingezonden bijdragen, die hij niet voor publicatie aanvaardt, terug te zenden. Indien de uitgever een ingezonden bijdrage voor publicatie aanvaardt, is hij gerechtigd deze op zijn kosten te (doen) bewerken. De uitgever is tevens gerechtigd een bijdrage te (doen) vertalen en voor haar andere uitgaven en activiteiten te gebruiken tegen de daarvoor bij de uitgever gebruikelijke vergoeding.

PWM-meting met een PIC

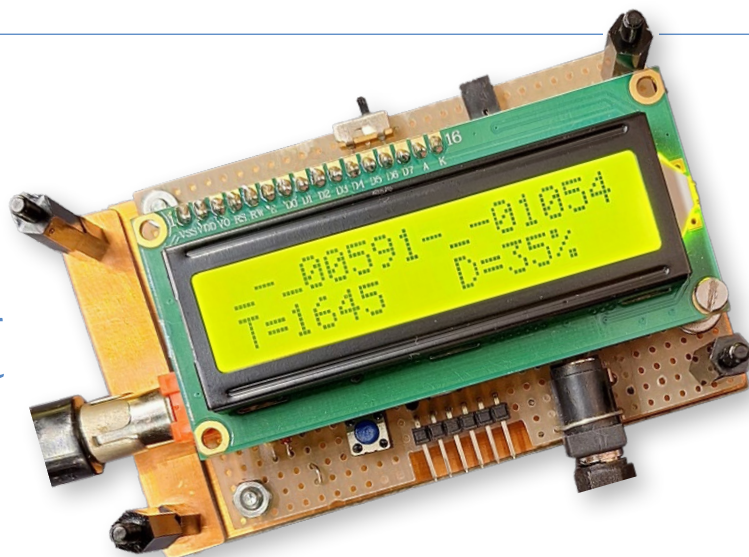
meet de aan/uit-verhouding van een stuursignaal

Giovanni Carrera (Italië)

Hoewel het programmeren van een Arduino eenvoudiger is dan dat van een PIC, was de laatste preciezer en nauwkeuriger bij het berekenen van de timing van een PWM-signaal, vanwege zijn architectuur. De uitdagingen die moeten worden aangegaan zijn beslist moeilijker, maar uiteindelijk zullen ze de ervaring verrijken van code-liefhebbers. Het hier gepresenteerde instrument kan tegelijkertijd de hoog- en laagintervallen meten en ook de periodeduur, dat alles met de resolutie van één microseconde voor pulsduren van 10 μ s tot 65.535 μ s bij PWM-signalen met een frequentie van 7,63 Hz tot 50 kHz.

De techniek om de pulsduur van een signaal in de tijd te variëren heet PWM (pulse width modulation, pulsbreedtemodulatie), een modulatievorm die zijn oorsprong vindt in de telecommunicatie maar ook toepassing heeft gevonden in de digitaal/analoge wereld om de gemiddelde spanning of stroom te variëren waarvan de waarde afhangt van de verhouding tussen de duur van de positieve puls en de gehele periode, een verhouding die duty cycle wordt genoemd.

PWM wordt gebruikt om de spanning te regelen in schakelende voedingen, voor motorbesturing, voor servo-actuators en zelfs voor de uitgang van sommige sensoren. Om deze redenen hebben de meeste moderne microcontrollers een of meer PWM-uitgangen. Sommige transducers, zoals akoestische afstandssensoren, hebben een digitale pulsuitgang waarvan de pulsduur evenredig is met de amplitude van



het signaal. Het doel van dit project is niet om PWM-signalen te genereren, maar om hun pulsduur nauwkeurig te meten.

In dit artikel beschrijf ik een versie die de PIC16F628 gebruikt om de duur van de hoge en lage delen van de pulsen te meten, evenals de periode (zie de kopfoto bij dit artikel). Als je ook de duty cycle als percentage wilt berekenen, heb je een PIC16F648 nodig, omdat het project een deling vereist. Het programma heeft slechts een paar extra instructies nodig, maar de code is langer dan 2 k woorden en de F628 heeft slechts 2 k flashgeheugen. Voor beide versies is voorzien in de sourcecode en gecompileerde HEX-bestanden [1]. De rest van de hardware blijft precies hetzelfde.

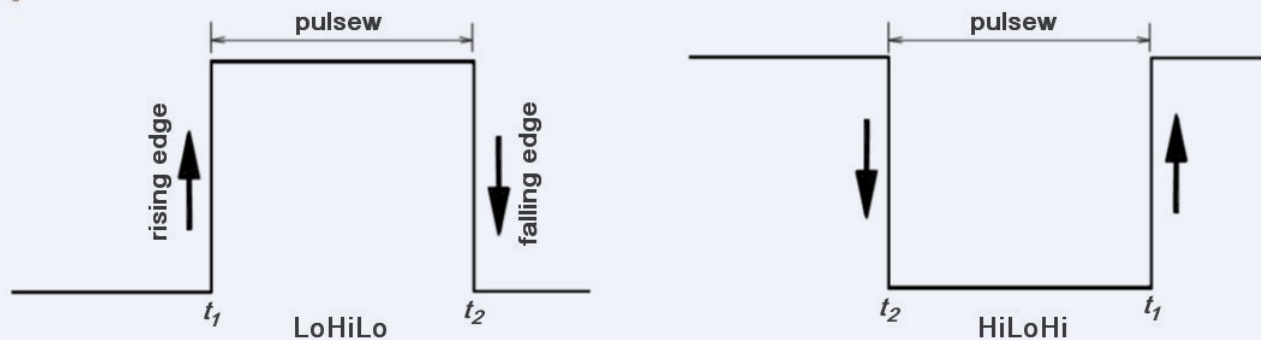
De PIC-microcontroller

Waarom deze keuze? In mijn geval had ik zoiets een aantal jaren geleden al gedaan, toen ik voornamelijk PIC's gebruikte in mijn projecten en toen overstapte op het ontwikkelen met de Arduino IDE en MicroPython.

In PIC-microcontrollers is de interne klok een kwart van de frequentie van de kristaloscillator en worden alle instructies in één klokperiode uitgevoerd, behalve programmavertakkingen. Met een 16 MHz-kristal is de interne klok 4 MHz, dus de uitvoeringstijd van een instructie is slechts 0,25 μ s.

De PIC16F628 is een Microchip-microcontroller met RISC-architectuur (Reduced Instruction Set Computer) die met 8-bit data werkt; zijn 35 instructies hebben echter elk een breedte van 14 bits. In ons geval heeft het flashgeheugen een omvang van 2 K woorden. Deze PIC biedt, naast zijn vele I/O-functies, ook event capture op de CCP1-pin (RB3), waar hij de inhoud van Timer1 opslaat in twee registers, CCPR1L en CCPR1H. Iets soortgelijks is er ook voor de ATmega328P van de Arduino UNO, maar de keuze voor een PIC leidt tot een veel eenvoudiger, compacter systeem met een veel geringer stroomverbruik.

Natuurlijk zullen degenen die gewend zijn om met Arduino te werken grotere moeilijkheden ondervinden bij het ontwerpen van een schakeling met een PIC. Het is niet zo eenvoudig om kant-en-klare printen te vinden of de vele bibliotheken die beschikbaar zijn voor Arduino om de gebruikte apparaten te sturen. Naar mijn mening kun je, wanneer je uitsluitend met Arduino werkt, niet erg diep in de materie duiken of zelfs maar precies weten wat je aan het doen bent.



Figuur 1. Er zijn twee verschillende capture modi beschikbaar, voor de pulstypes 'LoHiLo' (links) en 'HiLoHi' (rechts).

Veel Arduino-gebruikers doen niets anders dan projecten nabouwen aan de hand van Fritzing-figuurtjes en met plaatjes van de componenten die op breadboards zijn gerangschikt, en hebben geen idee hoe het allemaal werkt (terwijl het soms ook helemaal niet werkt door gammele of onjuiste aansluitingen).

De compiler die voor PIC's wordt gebruikt is minder gebruiksvriendelijk dan de Arduino IDE en de beschikbare bibliotheken zijn veel beperkter, er is bijvoorbeeld geen `pulsein()`-functie om de duur van een puls te meten. Om deze redenen moet de ontwerper zich verdiepen in de werking van microcontrollers, zoals ik ook heb gedaan voor dit project, waarbij ik de datasheet [2] moest bestuderen om de werking van de timers en CCP-capture te begrijpen.

De mikroPascal-compiler, die ik heb gebruikt om de programma's met de PIC's te ontwikkelen, is alleen gratis voor programma's waarvan de code niet meer dan 2 k woorden lang is (1 woord = 14 bit), wat in dit geval ook niet werd overschreden omdat dit de flash-omvang van de gebruikte PIC is terwijl het RAM-geheugen slechts 224 bytes omvat. Binnen deze beperkingen zijn de mogelijkheden beperkt: met bijvoorbeeld een float-deling wordt de geheugenlimiet gemakkelijk overschreden. Maar het compileren gaat veel sneller dan met Arduino en de geproduceerde code is zeker ook compacter en optimaler.

Het programma is geschreven in Pascal met behulp van een krachtige cross-compiler voor Windows PC's: mikroPascal PRO for PIC, versie 7.6.0. Een volledige versie kan worden gedownload van de MIKROE-website [3].

In de eerste versie van het programma is de code korter, zodat je het zonder problemen kunt compileren. Wie meer vertrouwd is met C of Basic kan deze compilers downloaden en het programma naar die taal vertalen. Als er geen wijzigingen in het programma worden aangebracht, hoeft je niet eens de compiler te downloaden, maar heb je alleen een PIC-programmer nodig die het reeds gecompileerde hex-bestand *PWMmeter.hex* gebruikt.

Er is ook een andere versie van het programma ontwikkeld die de duty cycle berekent, maar deze vereist vanwege de floating-point deling meer flashgeheugen en een PIC16F648. In dat geval moet je, als je de compilerlicentie niet hebt aangeschaft, de chip programmeren met het HEX-bestand dat bij dit project wordt geleverd.

Beide microcontrollers zijn geheel pin compatibel.

Meting van pulsduur

Sommige microcontrollers met een ingebouwde BASIC-interpret, zoals Stamp, Picaxe of BasicX, gebruiken een speciale routine (`PULSIN`) om de pulsduur te meten, maar kunnen geen korte pulsen meten omdat ze relatief traag zijn. Arduino-boards gebruiken de functie `pulsein(pin,`

level, timeout) om de duur van een hoge of lage puls met een resolutie van een microseconde te meten. Voor langere tijden is er de functie `pulseInLong()`, die de interrupt gebruikt en die tijden meet van 10 μ s tot 3 minuten. Beide functies retourneren een `unsigned long`.

Het programma gebruikt twee interrupts: Timer0 voor de sampling-tijd, terwijl de tweede interruptbron wordt gegenereerd bij CCP capture-gebeurtenissen (capture/compare/PWM) op de opgaande en neergaande flank van de puls waarbij de betreffende tijden worden geteld in Timer1.

Timing met Timer0

Timer0 genereert een interrupt die wordt gebruikt om metingen te bemonsteren en telkens 50 events op het LC-display weer te geven (overeenkomend met ongeveer 0,5 seconde). Timer0 is een 8-bit TMR0-teller, voorafgegaan door een 8-bit programmeerbare deler (de zogenoemde prescaler).

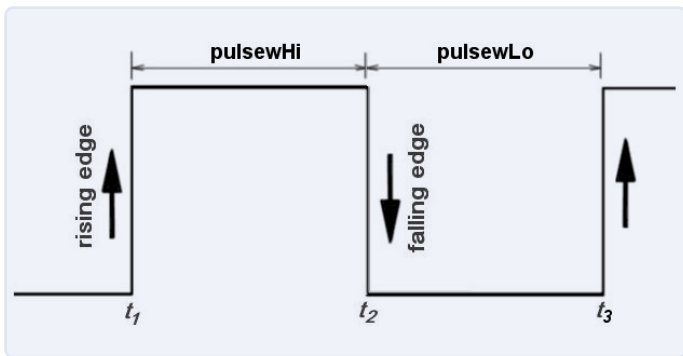
Het programma selecteert de interne klok, die een frequentie heeft die gelijk is aan een kwart van de kristalfrequentie: $16 \text{ MHz}/4 = 4 \text{ MHz}$. Als we de prescaler instellen op de hoogste waarde (1:256), zal Timer0 als ingang een frequentie hebben van $f_1 = 4 \text{ MHz}/256 = 15,625 \text{ kHz}$, met een periode van $64 \mu\text{s}$; wanneer we 100 in de teller laden zal dat resulteren in een interrupt elke 156 ($256 - 100$) pulsen met frequentie f_1 , wat overeenkomt met een periode van $156 \times 64 = 9984 \mu\text{s}$ (ongeveer 100 Hz).

De Timer0-interrupt wordt gegenereerd wanneer de timer/counter in het TMR0-register overloopt gaat van 0xFF naar 0x00, waardoor het T0IF-bit wordt gezet. Deze interrupt genereert de klok die wordt gebruikt om de metingen te bemonsteren en ze om de 50 events op het LC-display te presenteren, wat overeenkomt met ongeveer 0,5 s.

Event capture met Timer1

Het programma verandert na de opgaande flank op pin RB3 de capture-modus voor de neergaande flank, en omgekeerd. De pulsen kunnen van het type laag $\rightarrow$ hoog $\rightarrow$ laag zijn, wat we LoHiLo zullen noemen, of omgekeerd HiLoHi, zoals getoond in **figuur 1**. Het pulstype wordt geselecteerd door dipschakelaar 1: OFF = LoHiLo, ON = HiLoHi; de stand van de schakelaar wordt gelezen door het programma bij het opstarten. Verwijder deze, dat is een relikwie een oudere versie van het project en er per ongeluk in gelaten door de auteur. Nu is de functie geautomatiseerd via SW.

Voor LoHiLo-pulsen stelt het programma het register `CCP1CON := $05` in op opgaande flank. Wanneer deze gebeurtenis optreedt, slaat de interruptroutine de inhoud van het `CCPR1L`- en `CCPR1H`-registerpaar op, dat vervolgens naar t_1 wordt overgedragen. Vervolgens wordt het



Figuur 2. Inlezen van de t1-, t2- en t3-tijden om de benodigde berekeningen uit te voeren.

register `CCP1CON := $04`, waardoor de capture op dalende flank wordt ingesteld, bij welke gebeurtenis het registerpaar `CCPR1L` en `CCPR1H` opnieuw wordt gelezen, dat vervolgens wordt overgedragen naar t2, zoals te zien is in figuur 1. De `pulsew`-tijd wordt afgeleid uit het verschil $t_2 - t_1$.

Voor HiLoHi-pulsen gaat het ongeveer hetzelfde, maar je begint dan met de neergaande flank (t2) en eindigt met de opgaande flank (t1), in welk geval de `pulsew`-tijd gelijk is aan $t_1 - t_2$. Timer1 heeft een 16-bit teller, voorafgegaan door een 2-bit prescaler.

Ook hier gebruiken we de interne klok van 4 MHz. Met de prescaler op 1:1 bedraagt de resolutie 0,25 μ s, wat te veel is als we uitgaan van een normaal kristal, dus gebruiken we de verhouding 1:4 wat een resolutie van 1 μ s oplevert. Daarom meet het systeem pulsen met een duur tussen ongeveer 10 μ s en 65.535 μ s. Onder de 10 μ s werkt het niet, omdat er verschillende instructies in de interruptroutine zitten en dit enige uitvoeringstijd vergt. Als je langere tijden wilt meten, met een resolutie van 2 μ s, moet je de 1:8 prescaler-instelling gebruiken. In dat geval moeten de tijden met twee vermenigvuldigd worden na het converteren van de variabelen naar long integers.

De al genoemde nieuwe, grotere versie van het oorspronkelijke programma kan de twee tijdsduren tegelijkertijd meten; daarom kun je de PWM-periode en ook de duty cycle berekenen. In dat geval selecteert het programma de capture voor de opgaande flank, slaat de tijd op in t1, verhoogt de `firstcre`-vlag om deze flank te onderscheiden van de vorige, stelt vervolgens in op de neergaande flank en zet de tijd in t2, en tenslotte voor de volgende opgaande flank die in t3 wordt gezet, zoals te zien is in figuur 2. Bij de derde flank zet de interruptroutine de vlag `datok` om aan te geven dat de tijden t1, t2 en t3 zijn gemeten.

En dan wordt de berekening:

$$pulsewHi = t_2 - t_1, pulsewLo = t_3 - t_2, periode = pulsewHi + pulsewLo$$

De duty cycle vereist bewerkingen met `float`-variabelen (`real` in Pascal) die de flash-geheugenlimiet van de 16F628 overschrijden, maar we kunnen ze uitvoeren met een rekenmachine:

$$periode = T = pulsewHi + pulsewLo$$

$$duty\ cycle\ D = pulsewHi / T \times 100 [\%]$$

Vergelijking met Arduino `pulseIn()`

Ter vergelijking, en alleen voor een LoHiLo-puls, heb ik ook een klein programma geschreven voor de Arduino Uno:

```
// program pulseintest
#define pulsein 4
```



Onderdelenlijst

Weerstanden:

R1 = 10 k, 5%, 0,25 W
R2 = 100 Ω , 5%, 0,25 W
R3 = 100 Ω , 5%, 0,5 W
Rp1 = 22 k instelpotmeter

Condensatoren (keramisch):

C1,C2 = 22 p, 50 V
C3,C4,C5,C6 = 100 n, 25 V

Halfgeleiders:

U1 = PIC16F628A of PIC16F648A microcontroller
U2 = LM7805 5V-regelaar
D1,D2 = 1N4148 diode

Diversen:

Display = LCD 2x16
X1 = 16MHz-kristal
SW1 = drukknop (Reset)

```
void setup() {
  Serial.begin(115200);
  pinMode(pulsein, INPUT);
}

void loop() {
  unsigned long duration = pulseIn(pulsein, HIGH);
  Serial.print("Pulse High [us] = ");
  Serial.println(duration);
  delay(500);
}
```

Om de tijden te controleren gebruikte ik mijn eigen kristalgebaseerde tester met programmeerbare deler en blok golfuitgang als referentie; in tabel 1 zijn de resultaten vergeleken met die van mijn PIC-systeem.

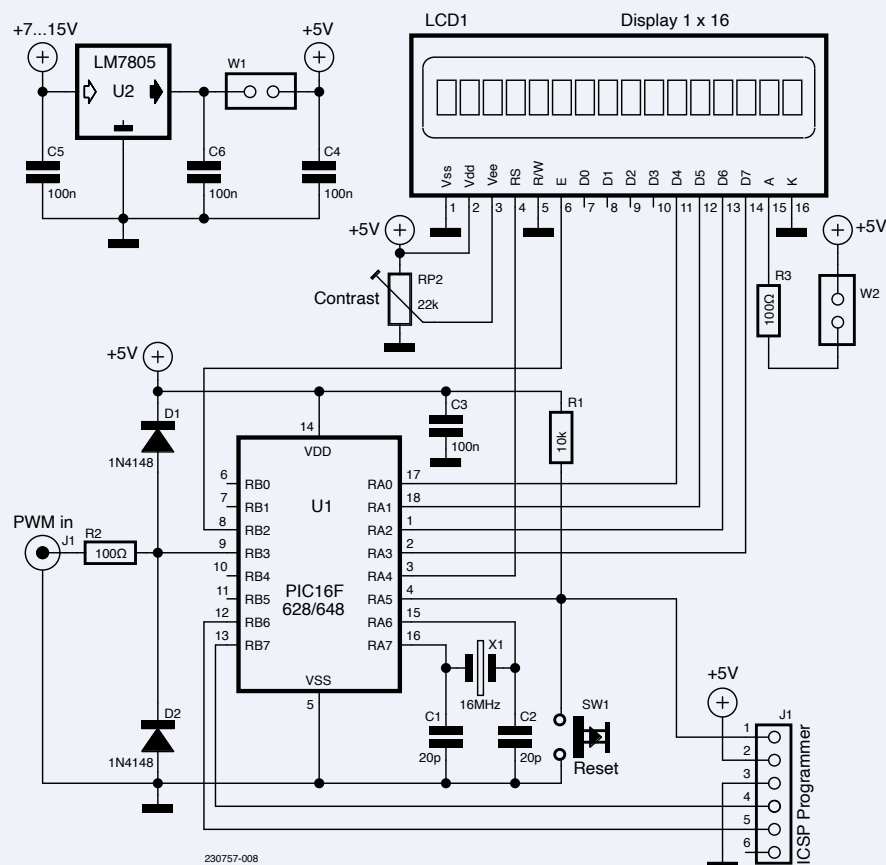
Table 1. Metingen.

ref. [μ s]	PIC [μ s]	Arduino [μ s]	Vershil % PIC	Vershil % Arduino
5		5...6		0...20
50	50	50...51	0	0...2
500	500	494...500	0	-0,8...0
5.000	5.000	4.966	0	-0,68
50.000	49.997	49.662	-0,006	-0,676

Het moge duidelijk zijn dat het PIC-systeem erg nauwkeurig is.

Het schema

Figuur 3 toont het schema van het systeem. Met behulp van een LM7805 (een 78L05 is ook goed) moet het systeem worden gevoed als een Arduino UNO, dat wil zeggen met een spanning van 7 V (optimaal) tot 12 V; het verbruikt echter, zonder achtergrondverlichting (W2 uit), minder dan 10 mA, waarvan een stroom van ongeveer 3 mA voor rekening van de regelaar komt. Een veel eenvoudiger mogelijkheid is het gebruik van drie gewone 1,5 V-batterijen. In dit geval is het LC-display

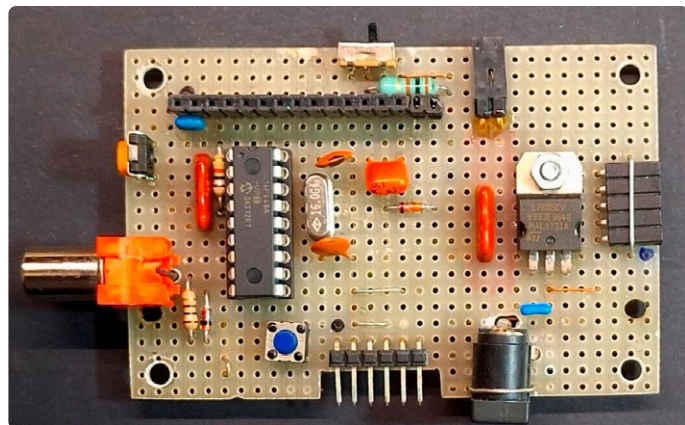


wellicht kritischer, omdat dit een nominale voedingsspanning van 5 V heeft, maar veel displays werken ook op 4,5 V.

Als display heb ik een gewoon LCD met twee regels van 16 karakters gebruikt, met een parallelle interface die compatibel is met HD44780. Jumper W1 wordt gebruikt om voedingsconflicten te voorkomen en moet worden verwijderd als je de chip programmeert met de PICKIT. Jumper (of schakelaar) W2 wordt gebruikt om de achtergrondverlichting uit te schakelen en stroom te sparen.

Het prototype

Figuur 4 toont de plaatsing van de componenten op het stuk gaatjesprint waarop het prototype is opgebouwd. Het LC-display is verwijderd om de eronder gemonteerde componenten te laten zien.



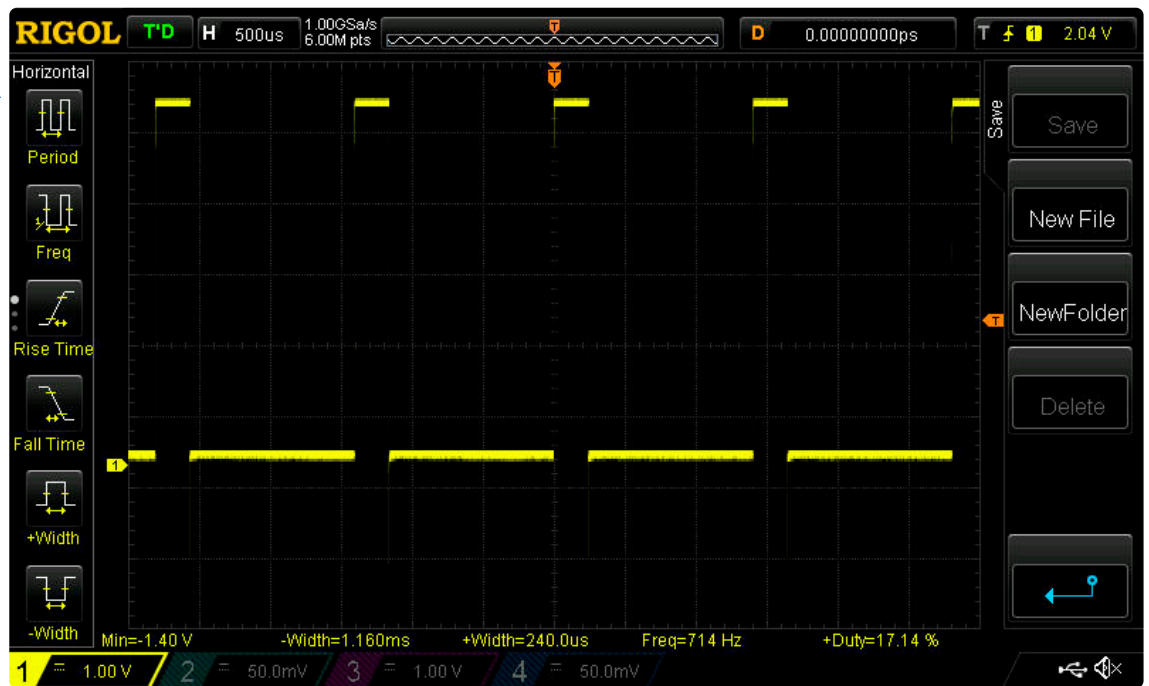
Figuur 4. Het voltooide prototype, opgebouwd op een stuk gaatjesprint dat is overgebleven van een eerder project. De 5-polige busstrip rechts is niet in gebruik.

Aan de bovenkant zie je een kleine schakelaar (W2) – deze schakelt de achtergrondverlichting van het display uit. Aan de linkerkant zit een cinch-connector voor de PWM-ingang en aan de onderkant zit een 6-pens connector voor het programmeren van de chip, die compatibel is met de PICkit in-circuit programmer. Als je een PIC-programmer met ZIF-voetjes hebt, is deze connector niet nodig, maar moet je de PIC uit zijn voetje halen om hem te programmeren. Daarnaast, nog steeds aan de onderkant, zie je de voedingsconnector. De print kan nog worden verkleind – het hier gebruikte exemplaar is gerecycled van een eerder project. De 5-polige busstrip rechts wordt niet gebruikt.

Het programma

Het grootste deel van het programma is hiervoor al beschreven. De uitvoer voor het LC-display is basaler dan de *LiquidCrystal*-bibliotheekfuncties van Arduino; de berichten moeten strings of arrays van tekens zijn, dus is het nodig om de variabelen om te zetten naar strings en alle voorloopspaties te verwijderen.

Het belangrijkste werk wordt gedaan door de interrupt-routine. In het eerste deel controleert deze of de interrupt wordt veroorzaakt door Timer0-overflow, in welk geval `INTCON`, `T0IF` = 1, en verhoogt het de `int0flg`-flag die wordt gebruikt voor de timing. Het laadt dan de waarde 100 opnieuw in de teller zodat een interrupt wordt getriggert na 156 pulsen en reset de interrupt `INTCON`- en `T0IF`-vlaggen op 0. Het tweede deel, dat complexer is, betreft het vastleggen van de opgaande en neergaande flanken en het opslaan van de inhoud van de twee capture-registers, `CCPR1L` en `CCPR1H`, in de variabelen `t1`, `t2` en `t3`, wat overeenkomt met de eerste opgaande flank, de neergaande flank en de tweede opgaande flank. Als al deze drie tijden zijn vastgelegd, wordt de vlag `datok` gezet, die door het programma wordt gebruikt om de puls- en periodeduur te berekenen.



Figuur 5. Oscilloscoop-screenshot van de meting van figuur 6.

De tijden, uitgedrukt in μs , met betrekking tot het hoge deel van het PWM-signaal, aangegeven met '—', en het lage deel, aangegeven met '—', worden afgedrukt op de eerste regel. De PWM-periodeluur, dus de som van de twee tijden, wordt afgedrukt op de tweede regel.

Als er geen signaal is, geeft het scherm niets weer. De volledige code is afgedrukt in **listing 1**. Dit programma gebruikt 1.236 woorden flashgeheugen (65%) en 74 bytes RAM (37%); het compileren duurde 187 ms.

Programmaversie met berekening van de duty cycle

Hiervoor heb je een PIC16F648A nodig, die lijkt op de F628, maar een flashgeheugen van 4 k woorden heeft. Zoals al opgemerkt moet je voor het compileren van deze versie de programmalicentie kopen, die de limiet van 2 k woorden opheft. Maar het reeds gecompileerde hex-bestand voor de PIC16F648 zit in het download-pakket bij dit artikel en kan meteen in de PIC worden geladen.

Het programma print `pulswHi` [μs], `pulswLo` [μs], `period` [μs] en `dutyc` [%] naar het display.

Helaas gaf de floating-point deling me geheugenproblemen bij het compileren, omdat zelfs de PIC16F648 niet erg geschikt is. Ik heb dit probleem opgelost door de procentuele duty cycle te converteren naar een geheel getal, waarbij het decimale deel verloren gaat.

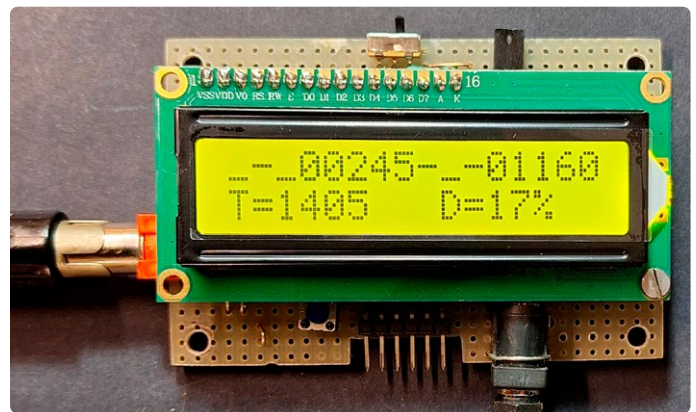
Vergelijkende tests

Figuur 5 en **figuur 6** tonen een vergelijking tussen metingen met een oscilloscoop met het PIC16F648 systeem. Hier is een TTL-pulsgenerator gebruikt. Je ziet dat de metingen zonder meer vergelijkbaar zijn.

PIC16F648 PWM-meter programmalisting

Het programma lijkt grotendeels op het vorige, met uitzondering van de `LCDprint`-routine. Er zijn instructies voor het berekenen en printen van de duty cycle toegevoegd. De gewijzigde routine staat in **listing 2**. De gewijzigde code gebruikt 2266 woorden flashgeheugen (55%) en 92 bytes RAM (62%); de compilatie nam 47 ms in beslag. ◀

230757-03



Figuur 6. Uitlezingen van de 'hoge' pulsduur (245 μs), 'lage' pulsduur (1160 μs), totale periodeluur T (1405 μs) en de duty cycle (17%).



Listing 2. PIC16F648 PWMmeter (gedeelte).

```
procedure LCDprint; // print measures on LCD
begin
  Lcd_Cmd(_LCD_CLEAR); // lcd clear
  WordToStrWithZeros(pulswHi, texdH); // microseconds
  ltrim(texdH); // trims the leading spaces
  WordToStrWithZeros(pulswLo, texdL);
  ltrim(texdL); // trims the leading spaces
  Lcd_Out(1, 1, '—' + texdH + '—' + texdL);
  LongWordToStr(period, texPer);
  ltrim(texPer); // trims the leading spaces
  //FloatToStr_FixLen(dutyc, texd, 5);
  Lcd_Out(2, 1, 'T=' + texPer);
  dutyc:= real(pulswHi)/real(period)*100.0;
  IntToStr(integer(dutyc), texd);
  ltrim(texd);
  Lcd_Out(2, 10, 'D=' + texd + '%');
  LCDOut:= false;
end;
```



Over de auteur

Giovanni Carrera heeft een graad in Elektrotechniek. Als professor aan de faculteit Scheepsbouwkunde in Genua (Italië) doceerde hij talloze cursussen, zoals scheepsautomatisering en de simulatie van scheepsvootstuwingsystemen. Carrera ging eind jaren 1970 aan de slag met de 6502-CPU en stapte daarna over op andere processoren. Tegenwoordig houdt hij zich bezig met het ontwerpen en ontwikkelen van analoge en digitale elektronische schakelingen, waarover hij veel heeft geschreven op zijn blogs (ArduPicLab en GnssRtkLab) en in verschillende tijdschriften.

Vragen of opmerkingen?

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



Gerelateerde producten

- > Tam Hanna, *Microcontroller Basics with PIC*, Elektor, 2020
www.elektor.nl/19188
- > Microchip MPLAB PICKIT 5 In-Circuit Debugger/
Programmer
www.elektor.nl/20665



WEBLINKS

- [1] Software-downloads bij dit artikel: <https://elektormagazine.nl/230757-03>
- [2] PIC16F627A/628A/648A datasheet: <https://ww1.microchip.com/downloads/en/DeviceDoc/40044G.pdf>
- [3] MIKROE-website: <https://mikroe.com/mikropascal-pic>



Listing 1. PIC16F628 PWMmeter.

```
program PWMmeter;
// PIC16F628 measures PWM (resolution= 1 usec)
// display the value on LCD every 0.5 sec
// timer#0 for interrupt every 9.984 msec (approx 100Hz)
// the LCD display has 2 rows x 16
// by G. Carrera 23/12/2023
// I/O configuration:
// PortB.3 = pulse input (in)
// PortA.0..3,= LCD data (4 bit mode) (out)
// PortB.2 = LCD E (out)
// PortA.4 = LCD RS (out)
// xtal = 16.00 MHz
var
  t1,t2,t3: word;
  int0flg,datok,LCDout,firstcre: boolean;
  i,t1l,t1h,t2l,t2h,t3l,t3h: byte;
  pulsewLo,pulsewHi: word;
  texdL: string[5];
  texdH: string[5];
  period: longint;
  texPer: string[10];
  dutyc: real;
  texd: string[6];
```

(Vervolg op volgende pagina)

```

// Lcd module connections
var LCD_RS : sbit at RA4_bit;
var LCD_EN : sbit at RB2_bit;
var LCD_D4 : sbit at RA0_bit;
var LCD_D5 : sbit at RA1_bit;
var LCD_D6 : sbit at RA2_bit;
var LCD_D7 : sbit at RA3_bit;
var LCD_RS_Direction : sbit at TRISA4_bit;
var LCD_EN_Direction : sbit at TRISB2_bit;
var LCD_D4_Direction : sbit at TRISA0_bit;
var LCD_D5_Direction : sbit at TRISA1_bit;
var LCD_D6_Direction : sbit at TRISA2_bit;
var LCD_D7_Direction : sbit at TRISA3_bit;
// End Lcd module connections
procedure interrupt;
// read capture times between rising and falling edge or vice versa
begin
  if TestBit(INTCON, T0IF) then // timer#0 interrupt
    begin
      ClearBit(INTCON, T0IF); // reset Timer#0 interrupt flag
      TMR0:= 100;
      SetBit(INTCON, T0IE); // Enables Timer#0 interrupt
      int0flg:= true;
    end;
  if TestBit(PIR1, CCP1IF) then // capture interrupt
    begin
      if TestBit(CCP1CON, CCP1M0) then // rising edge ($05)
        begin
          if firstcre then // already captured the first rising edge t1
            begin
              t3l := CCPR1L; // load final time to t3
              t3h := CCPR1H;
              firstcre:= false;
              datok:= true;
            end
          else
            begin
              t1l := CCPR1L; // load initial time to t1
              t1h := CCPR1H;
              firstcre:= true; // the first rising edge is captured now
            end;
          CCP1CON := $04; // change to falling edge on CCP1
          ClearBit(PIR1, CCP1IF); //Clear CCP1 Int. flag (also due to changing)
        end
      else // falling edge
        begin
          t2l := CCPR1L; // load intermediate time to t2
          t2h := CCPR1H;
          CCP1CON := $05; // change to rising edge on CCP1
          ClearBit(PIR1, CCP1IF); //Clear CCP1 Int. flag (also due to changing)
        end;
    end;
end;
end;
procedure IOinit; // initialize I/O
begin
  TRISB:= %00111011; // PORTB bit2 : output
  TRISA:= %10100000; // PORTA bits 0..4 are outputs
  CMCON:= $07; // comparators off (RA0..RA4 usable as digital IO)
  Lcd_Init(); // Initialize LCD
  Lcd_Cmd(_LCD_CURSOR_OFF); // Turn off cursor

```

(Vervolg op volgende pagina)

```

CCP1CON := $05; // rising edge on CCP1
T1CON := $21; // 1:4 prescaler, int. clk, enable t1
OPTION_REG:= $07; // set prescaler of Timer#0 to 256
ClearBit(OPTION_REG,7); // enable port B pull-ups
SetBit(PIE1,CCP1IE); // Enables the CCP1 interrupt
SetBit(INTCON, PEIE); // Enables peripheral interrupt
TMR0:= 100; // T0 overflow every 156 counts
SetBit(INTCON, T0IE); // Enables Timer#0 interrupt
ClearBit(INTCON, T0IF); // reset Timer#0 interrupt flag
SetBit(INTCON, GIE); // Enables global interrupt
int0flg:= false;
Lcd_Out(1, 1, 'PWMmeter-231223'); // Print text to LCD (row,column,text)
Delay_ms(2000);
Lcd_Cmd(_LCD_CLEAR); // lcd clear
i:=0;
firstcre:= false;
end;
procedure LCDprint; // print measures on LCD
begin
  Lcd_Cmd(_LCD_CLEAR); // lcd clear
  WordToStrWithZeros(pulsewHi, texdH); // microseconds
  ltrim(texdH); // trims the leading spaces
  WordToStrWithZeros(pulsewLo, texdL);
  ltrim(texdL); // trims the leading spaces
  Lcd_Out(1, 1, '_-' + texdH + '_-' + texdL);
  LongWordToStr(period, texPer);
  ltrim(texPer); // trims the leading spaces
  //FloatToStr_FixLen(dutyc, texd, 5);
  Lcd_Out(2, 1, 'T=' + texPer);
  LCDout:= false;
end;
begin // main program
  IOinit; // Initialize
  LCDout:= false;
  while true Do // endless loop
    begin
      if int0flg then
        begin
          int0flg:= false;
          i:=i+1;
          if i= 50 then // about 0.5 seconds
            begin
              LCDout:= true;
              i:= 0;
            end;
          end;
        if datok then
          begin
            datok:= false;
            t1:= (t1h shl 8)+t1l;
            t2:= (t2h shl 8)+t2l;
            t3:= (t3h shl 8)+t3l;
            pulsewHi:= t2-t1;
            pulsewLo:= t3-t2;
            period:= longint(pulsewHi) + longint(pulsewLo);
            if LCDout then LCDprint;
          end;
        end;
      end;
    end
  end
end

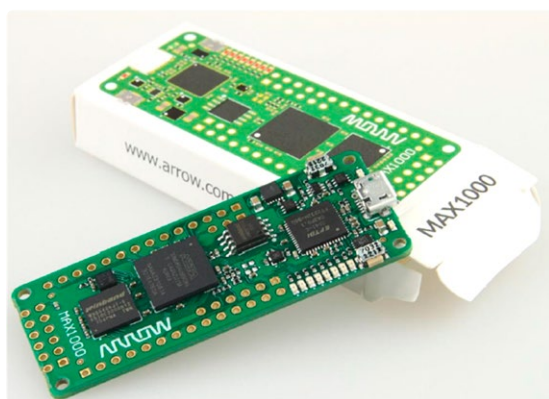
```

FPGA- gebaseerde Voltmeter

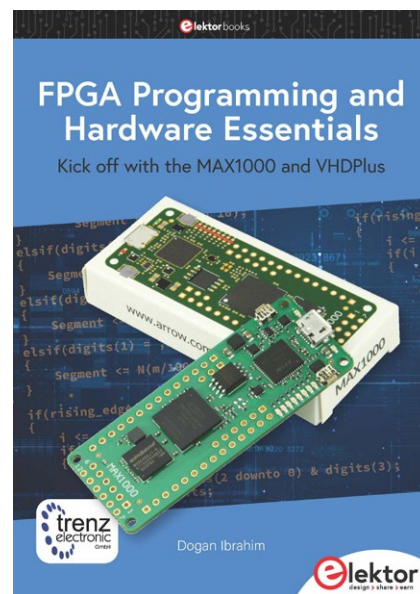
MAX1000 en VHDPlus
maken het eenvoudig

Dogan Ibrahim (Verenigd Koninkrijk)

Staat u te popelen om FPGA-programmering onder de knie te krijgen? Het Elektor-boek *FPGA Programming and Hardware Essentials* is de perfecte manier om in de wereld van field-programmable gate arrays te duiken. Laten we als snelstart-voorbeeld eens kijken naar een voltmeter die is ontworpen met de krachtige VHDPlus-programmeer-omgeving en is geïmplementeerd op de MAX1000 FPGA.



Figuur 1. De MAX1000
FPGA.



Noot van de redactie: dit artikel is een gedeelte van het Elektor-boek *FPGA Programming and Hardware Essentials*. Het is opgemaakt en licht bewerkt zodat het past bij de conventies en paginaopmaak van *Elektor Magazine*. Auteur en redactie helpen je graag bij eventuele vragen. Contactgegevens staan in het kader **Vragen of opmerkingen**.

Een FPGA is een enkele, configureerbare geïntegreerde schakeling die kan worden geprogrammeerd om een verscheidenheid aan taken uit te voeren die normaal worden geassocieerd met afzonderlijke chips. Als je dacht dat het thuis programmeren van een FPGA moeilijk was en alleen weggelegd voor professionals, dan moet je vooral verder lezen!

Kennismaking met de MAX1000 FPGA

De FPGA die in dit boek wordt gebruikt, is in principe het MAX1000 development board van Arrow Electronics (**figuur 1**). Het is ontworpen om u op weg te helpen met een FPGA. Omdat het board in andere hoofdstukken van het boek wordt beschreven, moeten de basisfuncties ervan hier aan het begin van dit artikel worden samengevat:

- Intel MAX10-chip
- Arrow USB Programmer 2
- 64 Mbit SDRAM
- 64 Mbit flashgeheugen
- 1× 12-MHz MEMS-oscillator
- 1× optionele MEMS-oscillator met de gewenste frequentie



- 8× rode LED voor de gebruiker
- 2× on-board indicatie-LED
- 2× drukknop voor de gebruiker
- 1× 3-assige versnellingsmeter
- 1× 12-polige Pmod-header (alleen soldeereilandjes)
- 1× JTAG-header voor de gebruiker (alleen soldeereilandjes)
- 1× I/O-header voor de gebruiker (alleen soldeereilandjes)
- mini-USB type-B connector

En als software: VHDPlus IDE

Er kunnen verschillende softwarepakketten worden gebruikt om FPGA-toepassingen te ontwikkelen op het MAX1000 FPGA-ontwikkelboard. In het boek leer je zowel de basisfuncties als de installatie van de VHDPlus IDE voor het programmeren van je MAX1000 kennen.

VHDPlus IDE is een superset van VHDL, die het programmeren eenvoudiger maakt door de mogelijkheden uit te breiden en de syntaxis te vereenvoudigen. VHDPlus is geen compleet andere taal, maar breidt de mogelijkheden van VHDL uit. Dus alles wat je met VHDL kunt doen, is ook mogelijk met VHDPlus. VHDPlus is een moderne benadering die FPGA-programmeren sneller en vriendelijker maakt, vooral voor beginners. VHDPlus IDE ondersteunt de open CRUVI-standaard. Simulatie van een ontworpen FPGA-programma kan lang duren. De simulatie-assistent van VHDPlus IDE helpt om je programma's te simuleren en eventuele fouten snel te herstellen. VHDPlus IDE integreert belangrijke functies van Quartus en is beschikbaar voor zowel Windows- als Linux-platformen. VHDPlus IDE ondersteunt ook C++, inclusief debugger.

Installatie van de VHDPlus IDE

Zo kun je de VHDPlus IDE naar je Windows-computer downloaden:

- Ga naar de VHDPlus-website [1].
- Klik op stap 1 om het MAX 10 device support bestand te downloaden (**figuur 2**). Het bestand wordt gedownload naar je Downloads-map. Sla het op in een map.
- Scroll omlaag naar stap 2 en download Quartus Prime Lite for Windows. Het bestand wordt in je Downloads-map opgeslagen.
- Klik op het bestand om het Quartus Lite-programma te installeren.
- Wanneer je het Quartus Lite-programma opstart, kan de volgende foutmelding verschijnen: You successfully installed the Quartus Prime software

Updates door de auteur

Sinds de publicatie van het boek en in reactie op feedback van lezers hebben we twee updates van de auteur ontvangen, voornamelijk met betrekking tot het gebruikte type FPGA.

1. Zie pagina 34, tweede bullet:

Er wordt een venster geopend (figuur 3.6) waarin u het gebruikte development board kunt selecteren.

- Dit venster heet Connect and compile.
- Linksboven zie je de tekst "FPGA:" gevolgd door een keuzelijst.
- Kies MAX1000 uit de lijst voor de 8 kLE-variant (de standaardoptie).
- Kies MAX1000 16k in de lijst voor de 16 kLE-variant.

Belangrijk: keuze van de verkeerde variant zal resulteren in een JTAG-fout wanneer geprobeerd wordt de gecompileerde code te downloaden naar de FPGA.

2. De 8 kLE-versie is te herkennen aan de markering 10M08 op de grote chip op de print, terwijl de 16 kLE-versie de markering 10M16 draagt.

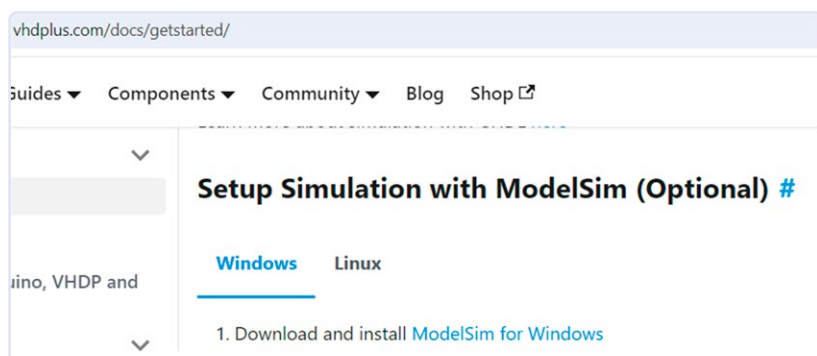
but did not install any devices. Do you want to launch the device installer to add devices? Installeer het device support-bestand .qdz zoals hieronder beschreven.

- Ga naar het Start-menu van Windows en zoek het programma Device Installer (Quartus Prime 18.1).
- De Device Installer vraagt naar de map waarin je het .qdz-bestand hebt opgeslagen. Selecteer die map.

Figuur 2. Download het device support-bestand voor MAX 10-IC's.

The screenshot shows the VHDPlus website with the 'Install VHDPlus IDE' section. It has tabs for Windows, Linux, and MacOS. Under '1. Download device support¹ for your hardware:', there is a table with columns 'Hardware' and 'Download'. The table lists three items: 'Core Max10, Core Max10 Ultra, MAX1000, ...' with a link to 'MAX 10 device support', 'CYC1000, ...' with a link to 'Cyclone 10 device support', and 'CYC5000, ...' with a link to 'Cyclone 5 device support'. Below this, it says '2. Download and install Quartus Prime Lite for Windows²' and '3. Download and install VHDPlus IDE:'. At the bottom, there is a button for 'VHDPlus-0.11.1.8-x64.msi'.

Hardware	Download
Core Max10, Core Max10 Ultra, MAX1000, ...	MAX 10 device support
CYC1000, ...	Cyclone 10 device support
CYC5000, ...	Cyclone 5 device support



Figuur 3. Download de ModelSim-simulator.

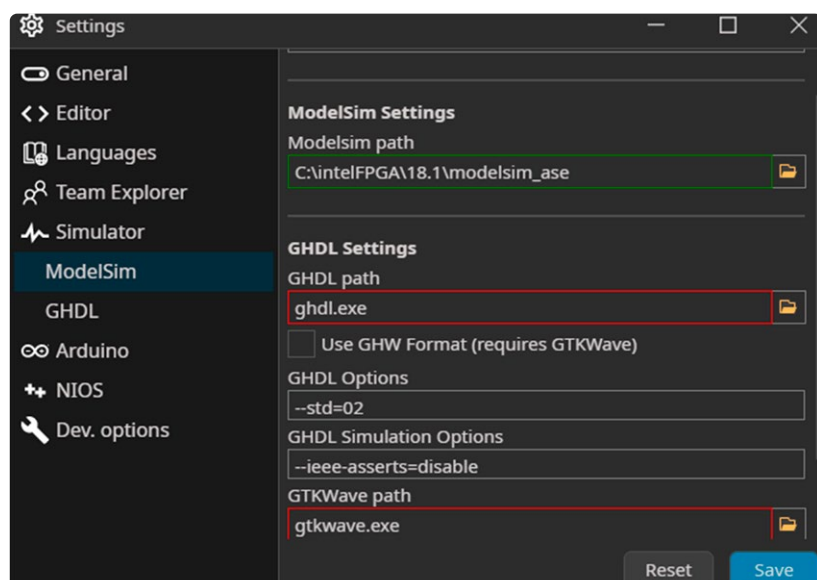
- > Het programma zoekt naar alle .qdz-bestanden. Kies het device dat je wilt installeren. Het .qdz-bestand is slechts een installatiepakket totdat je het installeert.
- > Klik op stap 3 om VHDPlus te downloaden. Het bestand wordt opgeslagen in je Downloads-map.
- > Klik op het bestand om VHDPlus te installeren.

Je moet de drivers voor uw programmer installeren om je FPGA te kunnen programmeren. Dat gaat als volgt:

- > Download de Arrow USB Programmer-driver voor je besturingssysteem van [2].
- > Unzip het gedownloadte bestand en voer het installatieprogramma uit om de installatie te voltooien.

Voor direct compileren en programmeren met VHDPlus IDE is een verbinding met Quartus nodig. Als Quartus in de standaardmap is geïnstalleerd, zijn er geen verdere stappen nodig. Pas anders het Quartus-pad aan in VHDPlus IDE door naar Extras → Settings →

Figuur 4. Specificeer het pad naar de simulator in VHDPlus IDE.



General te gaan. Zodra Quartus is gedetecteerd, wordt het kader groen.

Setup van de ModelSim-simulatie

Volg deze stappen om de ModelSim-simulator te installeren:

- > Ga naar de VHDPlus website [1].
- > Scroll omlaag en klik op de download van ModelSim for Windows (figuur 3). Het bestand wordt opgeslagen in je Downloads-map.
- > Installeer ModelSim door te dubbelklikken op het gedownloadte bestand.
- > Geef in VHDPlus IDE het pad op naar de modelsim_ase-map bij Extras → Settings → Simulator (figuur 4).
- > Voer de simulatie uit door met de rechtermuisknop op een VHDL-bestand te klikken.

Analoog/digitaal-omzetter (ADC)

ADC's zijn belangrijke schakelingen in de meeste op microcontrollers gebaseerde toepassingen. Dit komt omdat de meeste fysieke sensoren een analoge uitgangsspanning leveren. Een analoge temperatuursensor kan bijvoorbeeld een uitgangsspanning geven die recht evenredig is met de gemeten temperatuur. Laten we eens kijken naar de analoge ingangen van de FPGA van de MAX1000 en een project ontwikkelen dat daarvan gebruik maakt.

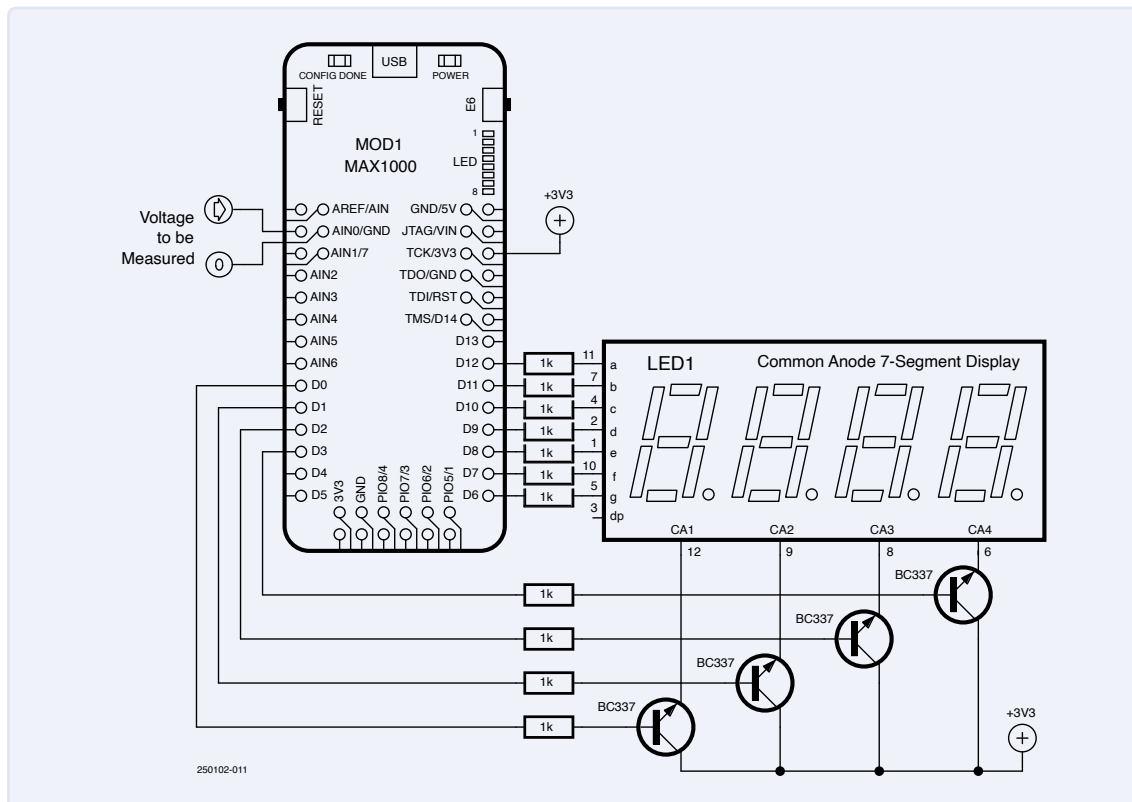
De MAX1000 FPGA heeft negen analoge ingangen (AIN en Ao...A7). Elke analoge ingang is 12 bit breed, wat overeenkomt met 4096 stapjes. Met de standaard referentiespanning van +3,3 V is de resolutie van een ADC-ingang $3,3 \text{ V} / 4096 = 0,8 \text{ mV}$. Daarom kunnen analoge ingangsspanningen worden gemeten met een resolutie van 0,8 mV. Als de ingangsspanning bijvoorbeeld 0,8 mV bedraagt, zal de uitgang van

Tabel 1. MAX1000 ADC-kanalen.

Analoge pin	Kanaal
AIN0	1
AIN1	6
AIN2	5
AIN3	7
AIN4	3
AIN5	0
AIN6	2
AIN7	4
AIN	8



Figuur 5. Blokschema van het voltmeter-project.



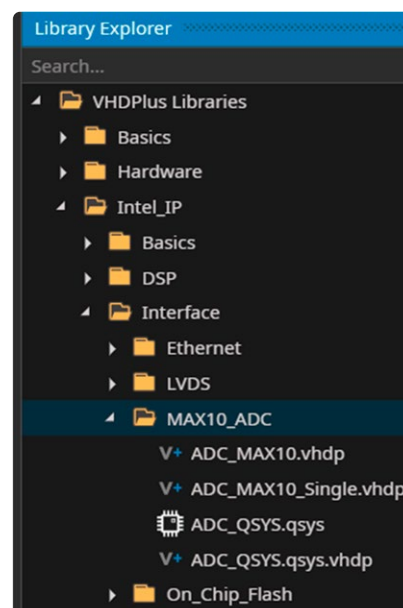
Figuur 6. Schema van het FPGA-gebaseerde voltmeterproject.

de ADC binair 0000 00000001 zijn. Als de analoge ingangsspanning 0,16 mV is, is de ADC-uitgang 0000 00000010. En als de analoge ingangsspanning 2,4 mV is, zal de ADC-uitgang 0000 00000011 zijn, enzovoort. De analoge ingangen hebben de vorm van kanalen. **Tabel 1** toont de kanaalnummers van elke analoge ingang.

Project 'voltmeter'

De hardware

Laten we als eerste project een uiterst eenvoudige voltmeterschakeling bouwen die de op een van de ADC-ingangen aangelegde spanning meet en (in mV) weergeeft op een 7-segment-display. De maximale ingangsspanning is +3,3 V (3300 mV). **Figuur 5** toont het blokschema van het project. Het schema is getekend in **figuur 6**, waar de te meten spanning wordt aangelegd op ADC-ingang AIN0 (kanaal 1).

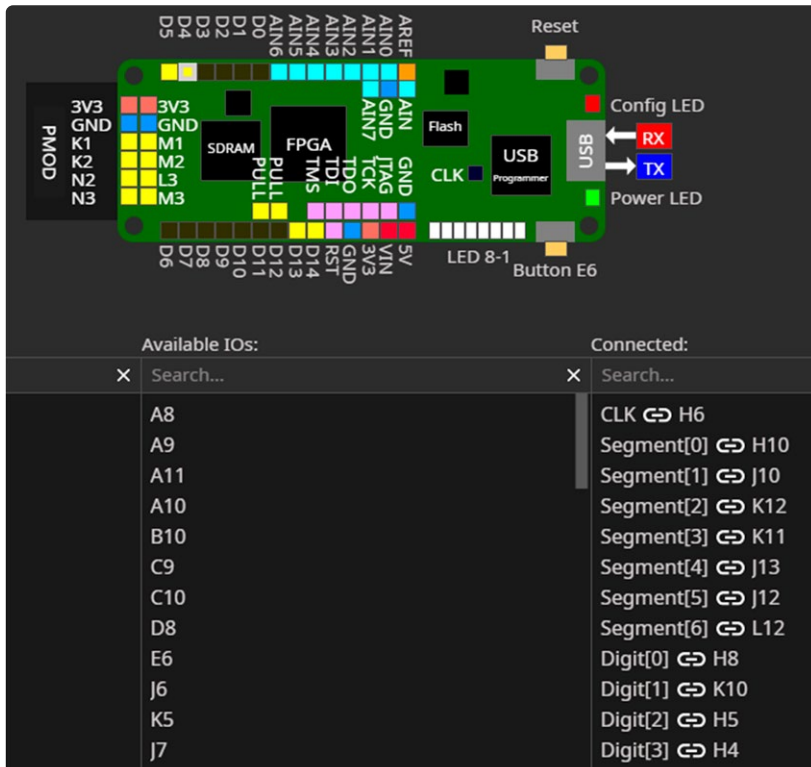


Figuur 7. Voeg de ADC-bibliotheek aan het project toe.

```

Output
AutoScroll [x] [icon]
Added library ADC_MAX10.vhdp to active project
Added library ADC_MAX10_Single.vhdp to active project
Added library ADC_QSYS.qsys to active project
Added library ADC_QSYS.qsys.vhdp to active project

```



Figuur 9.
Bedrading van de FPGA.

◀ Figuur 8. Bevestiging dat de ADC-bibliotheek is toegevoegd aan het project.

De software

We hebben de ADC-bibliotheek nodig om de ADC-ingangen te kunnen gebruiken. De stappen zijn als volgt:

- Open de IDE en geef je programma een naam (bijvoorbeeld *Voltmeter*).
- Klik linksonder in het scherm op *Library Explorer*.
- Selecteer (figuur 7) *Intel_IP* → *Interface* → *MAX10_ADC*.
- Rechtsklik op *MAX10_ADC* en selecteer vervolgens *Add to active project*. De ADC-bibliotheek is nu toegevoegd aan het project. Onderaan het scherm moet je nu een bericht zien waarin staat dat de *ADC_MAX10_ADC*-bibliotheek is toegevoegd aan het actieve project (figuur 8).
- Nu kun je code gaan schrijven met de *ADC_MAX10_Single* component uit de ADC-bibliotheek.

Figuur 9 toont het bedradingsschema, dat in principe de aansluiting van het 7-segment-display betreft (wat in meer detail wordt besproken in een ander hoofdstuk in het boek).

De code van het programma *Voltmeter* staat in **listing 1**. Het 7-segment LED-gedeelte van het programma is hetzelfde als wat we in hoofdstuk 5 van het boek hebben gezien. Daarnaast hebben we de signalen *temp*, *ADCdata* en *ADCchannels* gedefinieerd. Het signaal *temp* loopt van 0 tot 3300, wat de maximale spanning is die de ADC levert. *ADCdata* is de



Listing 1. De VHDL-code die wordt gebruikt om de voltmeter op een FPGA te laten draaien.

```

Main
(
    Segment: OUT STD_LOGIC_VECTOR(6 downto 0);
    Digit: OUT STD_LOGIC_VECTOR(3 downto 0);
)

{
    TYPE MyArray is an array (0 to 9) of STD_LOGIC_VECTOR(1 to 7);
    SIGNAL N: MyArray;
    SIGNAL m: integer range 0 to 9999 := 0;
    SIGNAL temp: integer range 0 to 3300 := 0;
    SIGNAL digits: STD_LOGIC_VECTOR(3 downto 0) := "0001";
    SIGNAL i: integer range 0 to 36000 := 0;

```



```
SIGNAL ADCdata: natural range 0 to 4095 := 0;
SIGNAL ADCchannels: natural range 0 to 8 := 0;

NewComponent ADC_MAX10_Single
(
    Channel => ADCchannels,
    Data => ADCdata
);

N(0) <= "1000000";
N(1) <= "11111001";
N(2) <= "0100100";
N(3) <= "0110000";
N(4) <= "0011001";
N(5) <= "0010010";
N(6) <= "0000010";
N(7) <= "1111000";
N(8) <= "0000000";
N(9) <= "0010000";

Process()
{
    if(digits(0) = '1')
    {
        Segment <= N(m mod 10);
    }
    elsif(digits(3) = '1')
    {
        Segment <= N(m/10 mod 10);
    }
    elsif(digits(2) = '1')
    {
        Segment <= N(m/100 mod 10);
    }
    elsif(digits(1) = '1')
    {
        Segment <= N(m/1000 mod 10);
    }
    if(rising_edge(CLK))
    {
        i <= i + 1;
        if(i = 36000)
        {
            i <= 0;
            digits <= digits(2 downto 0) & digits(3);
            Digit <= digits;
        }
        ADCchannels <= 1;           -- Read channel 1 (AIN0)
        temp <= ADCdata;           -- As raw data
        m <= temp * 3300 / 4095;   -- in millivolts
    }
}
}
```

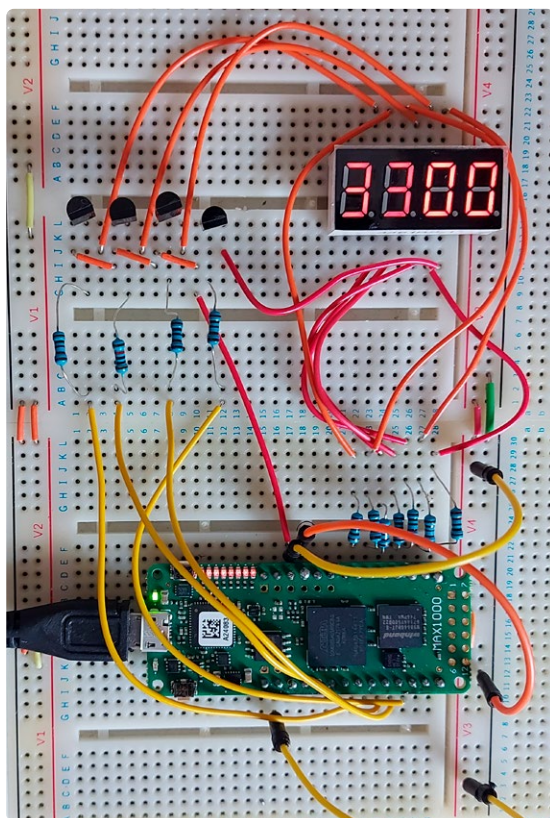


ruwe ADC-data die een waarde kan aannemen tussen 0 en 4095 waarbij 4095 overeenkomt met binair 1111 1111111, `ADCchannels` is het kanaalnummer en kan waarden aannemen van 0 tot 8.

Een nieuwe component met de naam `ADC_MAX10_Single` is gedefinieerd in het programma, en wordt gebruikt om toegang te krijgen tot de enkele ADC-module in de ADC-bibliotheek. Deze component heeft twee variabelen in de bibliotheek: het kanaalnummer (`Channel`) en de kanaalgegevens (`Data`).

Tijdens de opgaande flank van de klok wordt kanaal 1 geselecteerd, wat overeenkomt met ADC-ingang AIN0. De ruwe ADC-gegevens worden gekopieerd naar het signaal `temp`. Deze waarde wordt dan vermenigvuldigd met 3300 en gedeeld door 4095 zodat de ruwe waarde wordt omgezet in 'echte' millivolts. Door deze waarde te kopiëren naar `m` wordt de ADC-waarde weergegeven, dat wil zeggen de voltmeter-uitlezing op het 7-segment-display.

Figuur 10. Het voltmeter-project is hier op breadboard opgebouwd.



Uitproberen!

Figuur 10 tenslotte toont het project zoals het door de auteur werd gebouwd voor publicatie in het boek. Hoewel beperkt in mogelijkheden, is deze kleine voltmeter een prima introductie in de fascinerende wereld van FPGA programmeren met goedkope en gemakkelijk toegankelijke hardware en software. Wat wordt jouw volgende project? 

250102-03

Vragen of opmerkingen?

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



Over de auteur

Prof. dr. Dogan Ibrahim heeft een BSc in elektrotechniek, een MSc in automatisering en regeltechniek en een PhD in digitale signaalverwerking. Dogan heeft in veel industriële organisaties gewerkt voordat hij terugkeerde naar het academische leven. Hij is de auteur van meer dan 70 technische boeken en heeft meer dan 200 technische artikelen gepubliceerd over microcontrollers, microprocessoren en aanverwante onderwerpen. Hij is gecertificeerd elektronicus en Fellow van de Institution of the Engineering Technology. Hij is ook een gecertificeerde Arduino-professional.



Gerelateerde producten

- **Dogan Ibrahim, MAX1000 FPGA Programming Bundle (Elektor 2024)**
www.elektor.nl/21082
- **Dogan Ibrahim, FPGA Programming and Hardware Essentials (Elektor 2024)**
Boek: www.elektor.nl/21054
E-book: www.elektor.nl/21055

WEBLINKS

[1] VHDPlus-website: <https://vhdplus.com/docs/getstarted/#install-vhdplus-ide>

[2] Driver voor Arrow USB Programmer (Trenz Electronic): <https://tinyurl.com/Arrow-USB-Programmer>

WORD LID VAN ONZE COMMUNITY

GRATIS
DOWNLOAD



Abonneer je vandaag nog op elektormagazine.nl/ezine-24



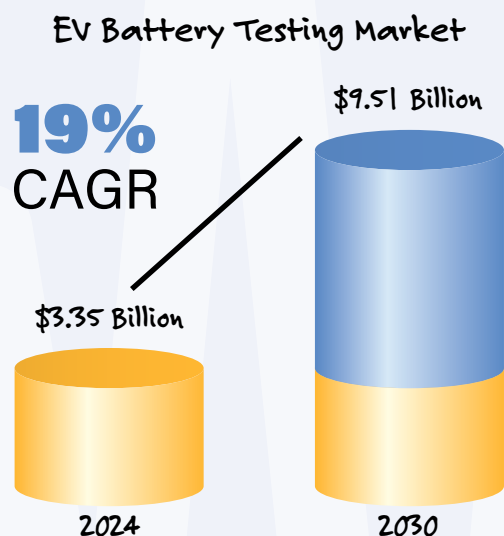
EV Batterijen testen: de belangrijkste groeimotor

De markt voor testapparatuur voor elektrische voertuigen (EV) kent een snel stijgende vraag naar gespecialiseerde testtools zoals batterijtesters en thermische beheersystemen, die kritieke aspecten van een EV beoordelen [1]. Vooral het segment Batterij Elektrische Voertuigen (BEV) domineert en stimuleert de groei van EV-testapparatuur door de unieke testbehoeften. Met ongeveer 40 miljoen EV's op de weg in 2023 [2] vragen BEV om geavanceerde tests van batterijen, aandrijflijnen en laadinterfaces om prestaties, efficiëntie en veiligheid te kunnen garanderen. Ontwikkelingen in batterijtechnologieën waaronder solid-state en de vooruitgang in hoge capaciteit lithium-ion chemie vereisen gespecialiseerde protocollen om de efficiëntie ervan te valideren. Naar verwachting zullen prestatietesten als marktsegment het snelst groeien tegen het jaar 2033.

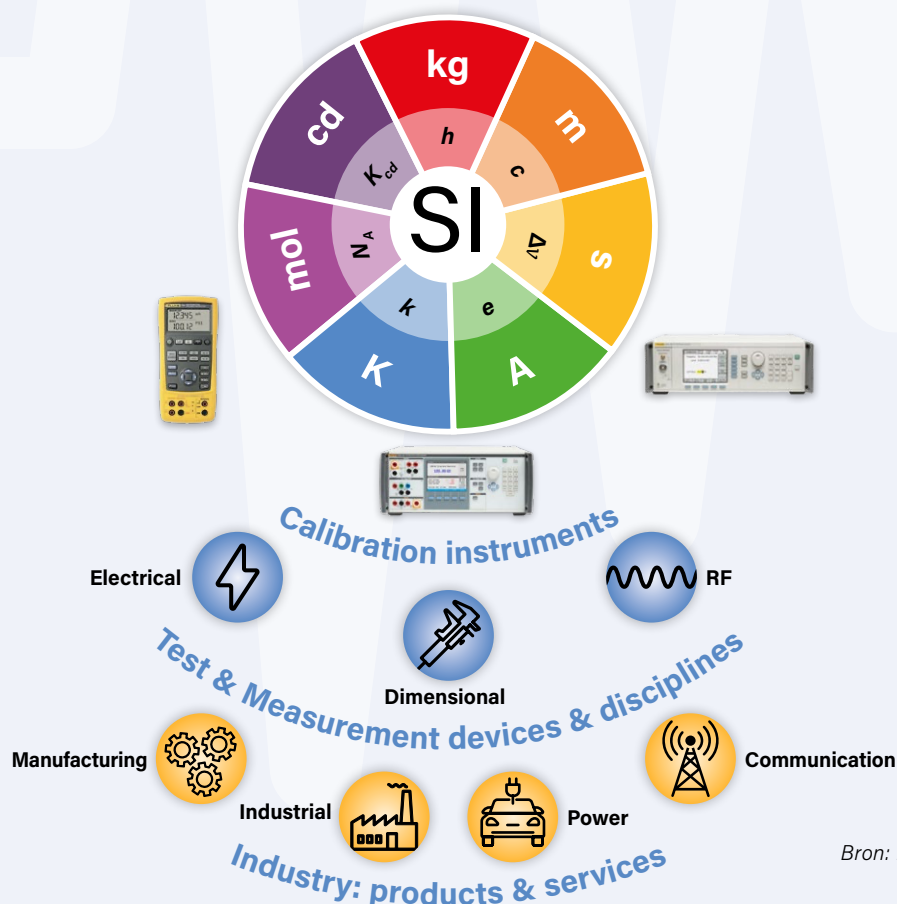
Belangrijkste spelers:

- › TÜV SÜD
- › Bureau Veritas
- › SGS Societe Generale De Surveillance SA
- › TÜV Rheinland AG Group
- › Dekra
- › Applus+

Organisaties zoals TÜV SÜD zijn toonaangevend in deze sector en bieden hun expertise op het gebied van lithium-ion batterijen aan met diensten zoals cyclische- en verouderingstests, snellaadprofielen en elektrochemische impedantiespectroscopie om optimale batterijprestaties te garanderen [3].



Bron: Onderzoek en Markten [3]



Het Kalibratie-domein

De toenemende vraag naar kalibratiediensten is een gevolg van steeds strengere kwaliteitsnormen in de precisieproductie, met een verwachte samengestelde jaarlijkse groei (CAGR) van 4,5% tussen 2025 en 2033 [4]. Metrologie zorgt voor nauwkeurigheid in zowel de pre- als de postproductie, waardoor de eisen met betrekking tot nauwkeurige toleranties blijvend kunnen worden nageleefd. Ontwikkelingen in kalibratiestandaarden vergroten echter de complexiteit en dagen serviceproviders uit om voorop te blijven lopen [5]. Tegelijkertijd kan de opkomst van zelfkalibrerende apparatuur in combinatie met softwareoplossingen traditionele diensten uitdagen waardoor aanbieders zich moeten blijven aanpassen.

Bron: Fluke [6]



Oscilloscopen blijven leidend

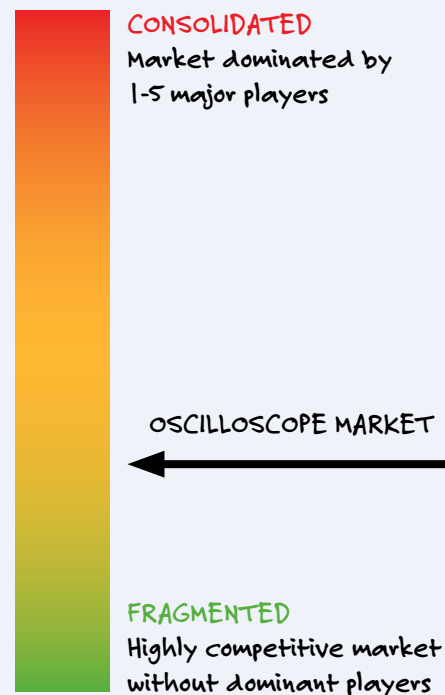
De markt voor General Purpose Test Equipment (GPTE) maakt een gestage groei door, gedreven door de toenemende vraag naar precisie meetinstrumenten in verschillende bedrijfstakken. Volgens Technavio [7] zal de GPTE-markt met

\$2.06 miljard groeien tussen 2023 en 2028. Met het grootste marktaandeel binnen GPTE zal naar verwachting de markt voor oscilloscopen alleen al groeien van \$3.74 miljard in 2025 naar \$5.49 miljard in 2030 bij een CAGR van 7,99% [8].

Markt leiders in oscilloscopen

- › Tektronix
- › Keysight Technologies
- › Rohde & Schwarz
- › Teledyne LeCroy
- › Yokogawa Test & Measurement Corporation

Market Concentration



Bron: Mordor Intelligence [8]

250108-03

WEBLINKS

- [1] Towards Automotive, "Electric Vehicle Test Equipment Market Size, Trends and Developments," September 2024: <https://tinyurl.com/EV-market>
- [2] IEA, "Trends in Electric Cars," 2024: <https://www.iea.org/reports/global-ev-outlook-2024/trends-in-electric-cars>
- [3] Research and Markets, "EV Battery Testing Market," December 2024: <https://www.researchandmarkets.com/reports/6036229>
- [4] IMARC Group, "Calibration Services Market," 2024: <https://www.imarcgroup.com/calibration-services-market>
- [5] MarketsandMarkets, "Calibration Services Market," 2023: <https://tinyurl.com/mam-calibration-market>
- [6] Fluke, "Why is Calibration Important?": <https://www.fluke.com/en-us/learn/blog/calibration/why-is-calibration-important>
- [7] Technavio, "General Purpose Test Equipment (GPTE) Market," Aug 2024: <https://tinyurl.com/technavio-GPTE-market>
- [8] Mordor Intelligence, "Oscilloscope Market," 2024: <https://www.mordorintelligence.com/industry-reports/oscilloscope-market>

Oscilloscoop/ multimeter/ generator

Fnirsi 2C53T met twee kanalen en
50 MHz bandbreedte

Harry Baggen (Nederland)

Er lijkt geen einde te komen aan de constante stroom van nieuwe producten die de Chinese fabrikant Fnirsi uitbrengt. Dit keer kijken we naar de 2C53T, een compact 3-in-1 meetinstrument dat een oscilloscoop, een multimeter en een functiegenerator combineert. Het ziet er precies zo uit als de 2C23T, die ik ongeveer een jaar geleden heb besproken. Dus, wat is er nieuw?

De Fnirsi 2C53T [1] is een compact 3-in-1 meetinstrument dat een oscilloscoop, een multimeter en een functiegenerator combineert (figuur 1).

Er lijkt geen einde te komen aan de constante stroom van nieuwe producten die de Chinese fabrikant Fnirsi uitbrengt. Soms zijn dat verbeterde versies van apparaten die al eerder zijn uitgebracht en dat is het geval bij de 2C53T die hier besproken wordt. Het ding lijkt namelijk aan de buitenkant als twee druppels water op de die ik een klein jaar geleden onder de loop heb genomen [2].

Even het geheugen oprissen

Zoals ik al opmerkte, is de behuizing precies hetzelfde gebleven, alleen is de opdruk bij enkele drukknoppen wat veranderd. Voor degenen die de voorganger niet kennen, hier de belangrijkste fysieke kenmerken van zowel de 2C23T als de 2C53T. Het is een vrij kleine behuizing van 17x9x3,5 cm met een 2,8-inch groot kleuren-LCD dat een redelijk helder en contrastrijk beeld levert. De behuizing is op de hoeken voorzien van een blauw rubberachtig materiaal



Figuur 1. De Fnirsi 2C53T is compact en vervangt drie aparte meetapparaten.

en voelt vrij stevig aan. De voeding wordt verzorgd door een ingebouwde lithium-accu van 3 Ah, goed voor zo'n 6 uur gebruik. Aan de voorzijde zitten 4 banaanstekerbussen voor de multimeter en aan de bovenzijde van het apparaat vinden we 3 BNC-bussen: twee voor de (tweekanaals) oscilloscoop en eentje voor de functiegenerator. Alles wordt bediend met behulp van 15 druktoetsen. Via een USB-C-connector aan de zijkant kan de interne lithium-accu worden opgeladen en kan verbinding worden gemaakt met een PC voor het updaten van de firmware en voor het downloaden van screenshots.

Veel toebehoren meegeleverd

De 2C53T wordt nu geleverd in een handig opberg-etui waarin ook plaats is voor de bijbehorende set meetsnoeren voor de multimeter, twee 100-MHz oscilloscoop-probes (bij de 2C23T was dat er maar eentje), een BNC-snoer met krokodilklampen voor de functiegenerator, een USB-C-kabel en een beknopte handleiding



Figuur 2. Bij de 2C53T krijgt u twee oscilloscoop-probes en een handig opberg-etui.

(figuur 2). Zoals bij de meeste apparaten van Fnrirs is alles keurig afgewerkt en netjes verpakt. Zelfs het bijbehorende etui is stevig en praktisch ingedeeld.

De verschillen tussen de Fnrirs 2C53T en 2C23T

Bij de 2C53T zijn voornamelijk de specificaties en mogelijkheden van de oscilloscoopfunctie flink verbeterd en uitgebreid. Zo is de ingangsbandbreedte verhoogd van 10 naar 50 MHz en de sampling-rate van 50 naar 250 Msamples/s (figuur 3). Dat is wel uitzonderlijk voor een meetapparaat in deze prijsklasse! Verder is er nu een X/Y-modus en zijn er 8 wiskundige functies waaruit de gebruiker kan kiezen. Ook is er een FFT-analyse, maar die is bij dit soort apparaten niet echt bruikbaar.

Een zogenaamde Persistence-functie maakt het mogelijk om het signaal een bepaalde tijd te laten nalichten, zoals dat bij oude scopes met een beeldbuis het geval was. Het aantal meetwaarden dat u in beeld kunt laten verschijnen, is uitgebreid tot 14 stuks per kanaal. Tot slot zijn alle instelmogelijkheden voor de oscilloscoop nu ondergebracht in een apart menu, dat is een stuk overzichtelijker.

Multimeter-updates

Bij de multimeter is er niet veel veranderd. Het aantal digits is vergroot van 4 (9999) naar 4½ (19999) en het display heeft een gewijzigde opzet, er is nu nog maar één analoge schaal en dat is wat strakker en overzichtelijker (figuur 4). Verder zijn de specificaties en mogelijkheden hetzelfde gebleven: een basisnauwkeurigheid van 0,5%, mogelijkheid om gelijk- en wisselspanningen en -stromen meten, plus weerstand, capaciteit en temperatuur. Een doorgangs- en diodetester zijn er uiteraard ook nog.

Zoals bij de meeste Fnrirs-multimeters detecteert de meter automatisch of er een gelijkspanning, een wisselspanning of weerstand op de ingangen wordt aangesloten, maar u kunt ook handmatig omschakelen naar een bepaalde functie.

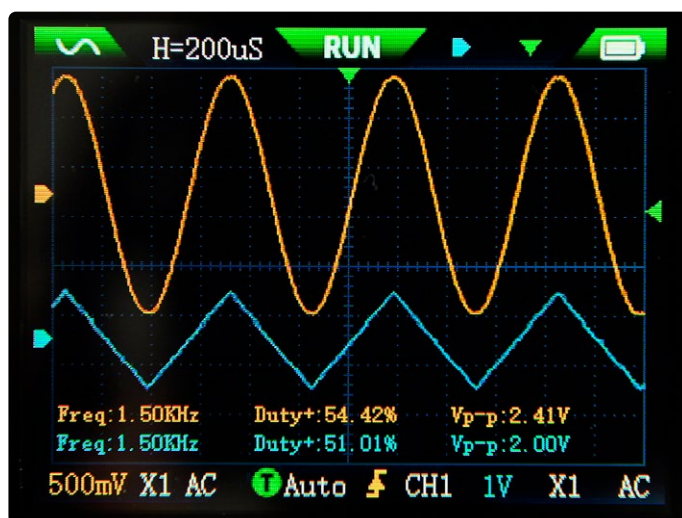
Wijzigingen bij de functiegenerator

Bij de ingebouwde functiegenerator is ook het een en ander gewijzigd. Het aantal beschikbare golfvormen is uitgebreid van 6 naar 13 stuks. Het uitgangssignaal heeft nu een maximale amplitude van 3 V piek-piek (was 3,3 V), maar het slechte nieuws is dat het frequentiebereik nog maar tot 50 kHz loopt (bij de 2C23T was dat 1 MHz). Nou ja – voor niets gaat de zon op.

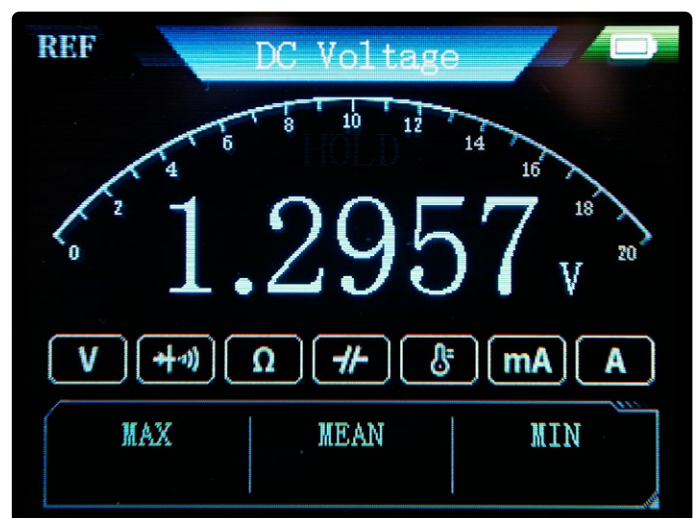
Aan de slag met de Fnrirs 2C53T

Als u al eens hebt gewerkt met een 2C23T, dan voelt u zich meteen thuis op de 2C53T. Afgezien van enkele kleinigheden is de bediening hetzelfde gebleven. Na het inschakelen van de meter verschijnt een menu met een aantal plaatjes waar u een meetinstrument of het instellingenmenu kunt selecteren. Het is overigens mogelijk om de 2C53T te laten opstarten met een bepaald meetinstrument. De oscilloscoop is toch wel het belangrijkste instrument in dit apparaat, vooral vanwege de grote bandbreedte en hoge sampling-rate. Helaas is het display niet groter geworden, dus het staat nog steeds erg vol, vooral als er veel meetwaarden worden getoond. De oscilloscoop reageert snel op signaalveranderingen en de bediening is wat overzichtelijker geworden door het gebruik van een apart instellingenmenu. Ook het veranderen van tijdbasis, gevoeligheid, triggerniveau en de positie van het signaal op het scherm is duidelijker: door op de Select-toets te drukken kiest u welke van deze parameters met de cursortoetsen kunnen worden ingesteld. Welke zijn gekozen, wordt boven in het scherm aangegeven.

Of de autosetup-functie een update heeft gehad, is moeilijk te beoordelen. Maar ik had wel de indruk dat de auto-functie wat sneller werkte dan bij de voorganger. De BNC-bussen aan de bovenzijde zitten nog steeds erg dicht bij elkaar, waardoor probes met een kunststof mantel rond de bus niet passen. Maar gelukkig worden hier twee geschikte probes meegeleverd. De bandbreedte van de oscilloscoop loopt inderdaad tot minstens 50 MHz. Alleen nam



Figuur 3. De oscilloscoop is nu bruikbaar tot 50 MHz en lijkt ook sneller te reageren dan bij de 2C23T.



Figuur 4. Het display van de multimeter is aangepast, er is nu nog maar één analoge schaal en dat is ook voldoende.

de amplitude van het signaal boven 10 MHz en tot 50 MHz enigszins toe, waardoor de gemeten waarde ook wat afweek. Dat wordt waarschijnlijk veroorzaakt door een niet geheel lineaire ingangstrap. Maar als u dat weet, kunt u er rekening mee houden.

De multimeter nader bekeken

De multimeter heeft net zoals de 2C23T een autodetectiefunctie die beneden 0,7 V niet werkt. Ook hier geldt weer dat u daar zelf gemakkelijk op kunt inspelen en handmatig kunt omschakelen als u vermoedt dat de getoonde waarde niet klopt of een weerstandswaarde in plaats van een spanning wordt getoond.

Het display van de multimeter vind ik wel een vooruitgang vergeleken met de 2C23T, met maar één analoge schaal boven de meetwaarde. Onderaan worden nog de minimale, maximale en gemiddelde meetwaarde getoond. Ook hier heb ik de nauwkeurigheid van de multimeter getest bij een aantal waarden voor spanning, stroom, weerstand en capaciteit, en dat was identiek aan de 2C23T. De nauwkeurigheid is zeker een factor 2 beter dan de gespecificeerde waarde, heel netjes!

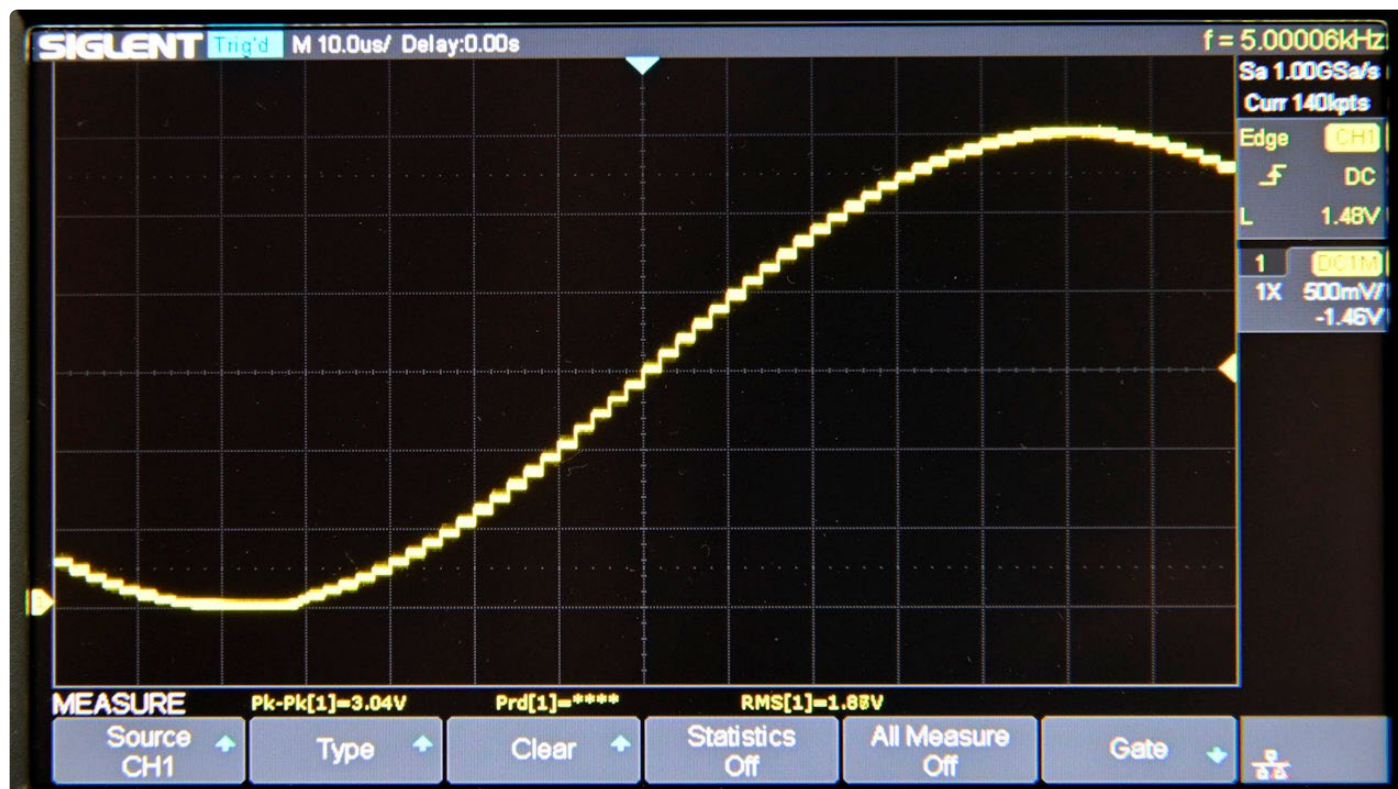
Ik vraag me wel af waarom het aantal digits is uitgebreid, want bij een basisnauwkeurigheid van 0,5% heeft men echt niet zoveel digits nodig. Het kan eventueel wel handig zijn voor vergelijkende metingen. Voor de twee stroombereiken heeft de meter weer twee interne zekeringen die u kunt vervangen door de behuizing open te schroeven.

Tenslotte is het jammer dat de vier ingangsbussen van de multimeter – net als bij zijn voorganger – niet de gebruikelijke onderlinge afstand van 19 mm hebben, maar net iets minder. Waarschijnlijk zou Fnirsi anders de behuizing iets breder hebben moeten maken. De functiegenerator heeft weliswaar meer signaaltypes waaruit u kunt kiezen, maar de kwaliteit van de signalen is niet verbeterd. Bij de sinus ziet u al op het scherm dat deze aan de onderzijde wat is afgeplat (**figuur 5**). Een beetje inzoomen toont meteen de vrij grove kwantiseringsschakels waaruit het signaal is opgebouwd, dus dit is alleen geschikt voor algemene testdoeleinden waarbij de signaalkwaliteit niet zo belangrijk is. De beperking van frequentiebereik tot 50 kHz is opvallend, dat is een erg groot verschil met de 2C23T.

De bediening van de functiegenerator hetzelfde gebleven. De frequentie kan op de hertz nauwkeurig worden ingesteld. De maximale uitgangsspanning bedraagt nu 3 V piek-piek onbelast. De uitgangsimpedantie is wel aangepast en bedraagt ongeveer 50 Ω . Houd er rekening mee dat het uitgangssignaal compleet positief is ten opzichte van massa, er is geen uitgangsschakelaar of symmetrische voeding aanwezig.

Conclusie

De Fnirsi 2C53T is weliswaar een paar tientjes duurder dan zijn voorganger 2C23T, maar daarvoor krijgt u wel enkele flinke verbeteringen in het oscilloscoopgedeelte. Vooral een bandbreedte van



Figuur 5. De signalen van de functiegenerator zijn nogal grof opgebouwd; we zien duidelijk de afzonderlijke kwantiseringsschakels.

50 MHz en een bemonsteringssnelheid van 250 Msamples/s zijn een belangrijke upgrade.

Ook al is de multimeter nauwelijks veranderd en is het frequentiebereik van de functiegenerator veel kleiner dan voorheen, alleen al vanwege de oscilloscoop-eigenschappen zou ik voor de 2C53T kiezen. Bovendien krijgt u er voor deze prijs ook twee oscilloscoop-probes en een handig opberg-etui bij. Dat is toch mooi meegenomen! 

250137-03



Gerelateerde producten

- **FNIRSI 2C53T (3-in-1) 2-ch Oscilloscope (50 MHz) + Multimeter + Signal Generator**
www.elektor.nl/21112
- **FNIRSI 2C53P (3-in-1) 2-ch Oscilloscope (50 MHz) + Multimeter + Signal Generator**
www.elektor.nl/20917
- **FNIRSI 2C23T (3-in-1) 2-ch Oscilloscope (10 MHz) + Multimeter + Signal Generator**
www.elektor.nl/20717

WEBLINKS

[1] Fnirsi 2C53T: <https://www.elektor.nl/products/fnirsi-2c53t-upgrade-3-in-1-2-ch-oscilloscope-50-mhz-multimeter-signal-generator>

[2] H. Baggen, "Fnirsi 2C23T oscilloscoop, multimeter & functiegenerator (review)", [elektormagazine.nl](https://www.elektormagazine.nl), april 2024:
<https://www.elektormagazine.nl/review/fnirsi-2c23t-oscilloscoop-multimeter-functiegenerator-review>

Start uw elektronica-innovaties met

ElektorLabs

- Gratis publicatie van uw projecten
- Support van experts
- Samenwerkingsmogelijkheden
- Toegang tot exclusieve bronnen
- Publicatie in Elektor Magazine mogelijk

Deel nu uw projecten!
www.elektormagazine.nl/e-labs



elektor
MEDIA & LEARNING

U ontwikkelt. Wij leveren.

De Nieuwste Producten voor Uw Nieuwste Ontwerpen™



nl.mouser.com/new



MOUSER
ELECTRONICS