Arduino Nano-golfvormgenerator

Nano + code = functiegenerator

Michael J. Bauer (Australië)

Dit is geen uitgewerkt DHZ-project compleet met print. Het aantrekkelijkste aspect van dit project is misschien wel de firmware en de codebibliotheek voor periferie, die veel nuttige technieken bevatten voor real-time embedded toepassingen, ontwikkeld door een professionele ingenieur met tientallen jaren ervaring in de industrie.

Golfvormgeneratoren, ook bekend als 'functiegeneratoren', zijn al sinds de eerste nummers van (toen nog) Elektuur populaire doe-het-zelfprojecten. Deze golfvormgenerator is gebaseerd op de Arduino Nano en daarom meer een low-end type wat betreft complexiteit en bouwkosten. Toch is die 8-bit AVR microcontroller verbazend krachtig, zodat de generator van pas kan komen bij het testen van audio-apparatur en niet te snelle digitale schakelingen.

Het geheel is ondergebracht in een kleine kunststof behuizing (130×70×40 mm), waardoor het apparaat goed draagbaar is. Het wordt gevoed met 5 V via USB. Alle onderdelen die nodig zijn om deze generator te bouwen zijn verkrijgbaar tegen belachelijk lage prijzen.

Bedrijfsmodi

Wave-modus

Golfvormen:	sinus, driehoek, blok, zaagtand en ruis
Frequentie:	laag 1...100 Hz (12 stappen); hoog 80 Hz...8 kHz (18 stappen) of continu 50 Hz...5 kHz
Ruis:	gefilterd bij 50, 100, 200, 400, 800, 1600, 3200 Hz of ongefilterd (wit)
Uitgang:	AC- of DC-gekoppeld
Uitgangsniveau:	100 mV_{PP}, 200 mV_{PP}, 500 mV_{PP}, 1 V_{PP}, 2 V_{PP} of 4 V_{PP}
DAC-resolutie:	8 bit

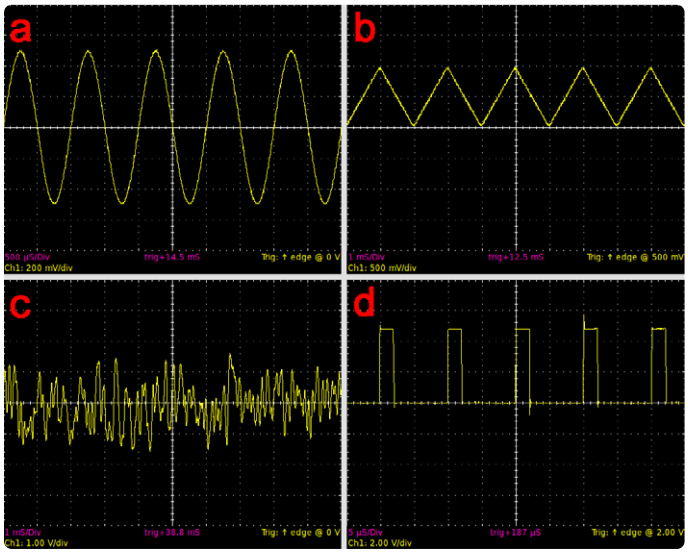
Pulse-modus

Frequentie:	laag 1...1000 Hz (16 stappen); hoog 1 kHz...4 MHz (16 stappen)
Duty cycle:	1...99% variabel (256 stappen)
Uitgangsniveau:	3,3 V of 5 V; 20 mA source/sink (bij 5 V)

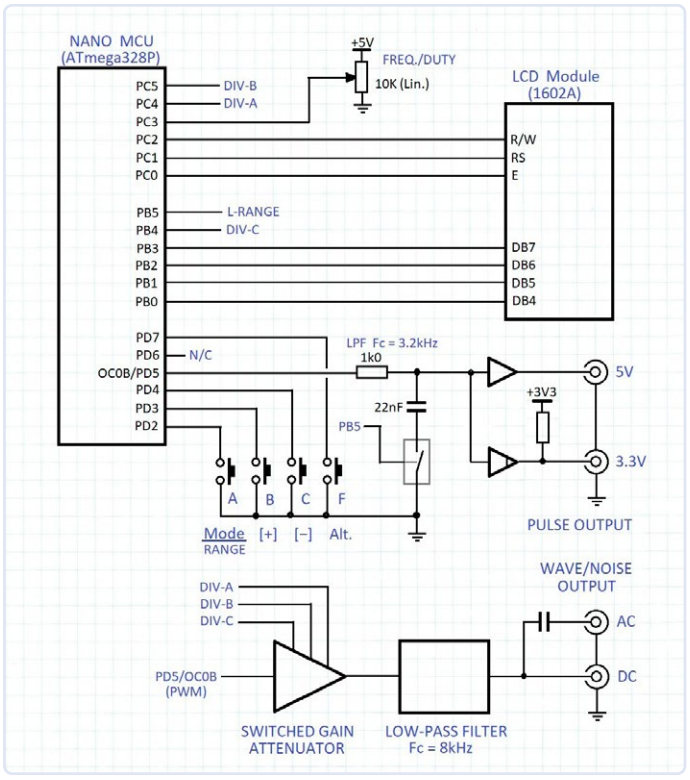
Het ontwerp

Het project is gebaseerd op het Arduino Nano v3-board plus een 2×16 (twee regels van 16 karakters) LCD-module (type 1602A) met LED-achtergrondverlichting, vier drukknoppen en een potentiometer. De potmeter regelt de signaalfrequentie of de duty cycle, afhankelijk van de gekozen uitvoermodus. De uitvoer-connectoren zijn RCA tulpconnectoren voor paneelmontage.

Het ontwerp maakt gebruik van een ATmega328P on-chip timer om in de Pulse-modus golfvormen met variabele duty cycle te genereren op een uitgangspen (zie het kader **Bedrijfsmodi**). In de Wave-modus wordt een 32 kHz PWM-uitgang gebruikt om signalen te genereren binnen het audiobereik (tot 8 kHz) waarbij uit diverse golfvormen (sinus, driehoek, blok en zaagtand) kan worden gekozen. Deze signalen worden geproduceerd door een wavetable-oscillatoralgoritme in de firmware. De 8-bit DAC maakt een amplituderesolutie mogelijk van 0,4% volle schaal. De 32 kHz bemonsteringsfrequentie wordt uit het audio-uitgangssignaal verwijderd met behulp van een analoog laagdoorlaatfilter met een afsnijfrequentie van 8 kHz, gebouwd rond een opamp (de helft van een MCP602). Omdat dit een derde-orde filter is, heeft het een –18 dB/octaaf karakteristiek. **Figuur 1** geeft screenshots van diverse door deze generator geproduceerde golfvormen.



Figuur 1. Screenshots van enkele golfvormen: a) sinus 1 kHz, 1 V_{PP}, AC uit; b) driehoek 500 Hz, 1 V_{PP}, DC uit; c) ruis, ongefilterd, 4 V_{PP}, AC uit; d) puls, 100 kHz, duty cycle = 20%, 5 V uit.



Figuur 2. Principeschema van de functiegenerator op basis van een Arduino Nano.

Vereenvoudigd schema

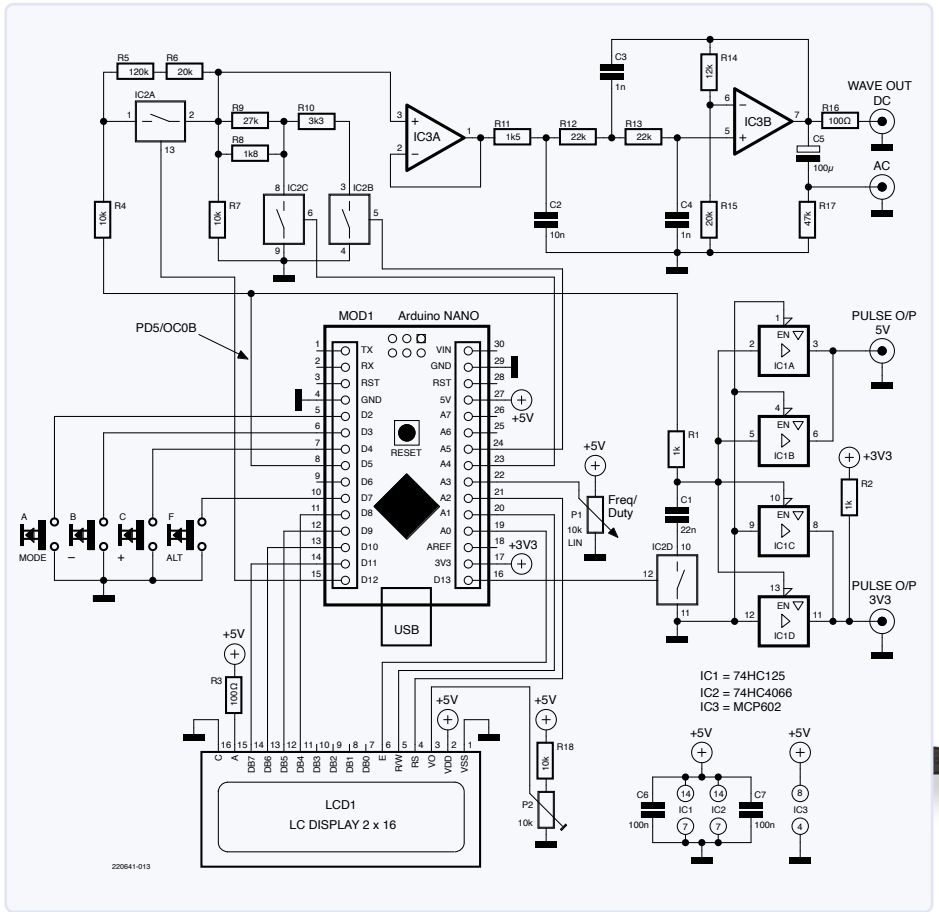
Figuur 2 toont een vereenvoudigd functioneel schema van de generator. In de Wave-modus wordt de uitgang ingesteld op één van zes niveaus met behulp van een geschakelde verzwakker tussen de PWM-uitgangspen en het opampgebaseerde laagdoorlaatfilter. De verzwakker gebruikt drie analoge schakelaars (74HC4066) die door de microcontroller worden aangestuurd, gevolgd door een opampbuffer (een halve MCP602).

In de laagfrequente Pulse-modus gebruikt de generator dezelfde 32kHz-samplefrequentie net als in de Wave-modus om pulsen met variabele duty cycle te genereren. Een eenvoudig eerste-orde RC-filter met een afsnijfrequentie van 3,2 kHz verwijdert de meeste resten van het 32kHz-signaal. Een buffer (74HC125) maakt van het analoge signaal een digitaal signaal, waardoor alle nog overgebleven 32kHz-residuen worden geëlimineerd. Het signaal heeft nu steile op- en neergaande flanken en kan digitale schakelingen direct aansturen.

In de hoogfrequente Pulse-modus genereert de MCU-timermodule rechtstreeks een signaal met variabele frequentie en duty cycle. Het RC-filter wordt uitgeschakeld door het openen van een schakelaar (een kwart van de 74HC4066). Het uitgangssignaal wordt gebufferd met een paar parallelgeschakelde poorten (uit een 74HC125 quad tri-state buffer). Dit biedt een 'high-current' drivemogelijkheid (20 mA sink of source) voor de 5V-pulsuitgang. De twee andere poorten gebruiken een 1-kΩ pull-up weerstand om een 3,3V-uitgangsniveau te verkrijgen. De pull-up oplossing leidt tot suboptimale stijgtijden, wat duidelijker wordt zijn bij hogere frequenties. Het uitgewerkte schema is te zien in **figuur 3**.

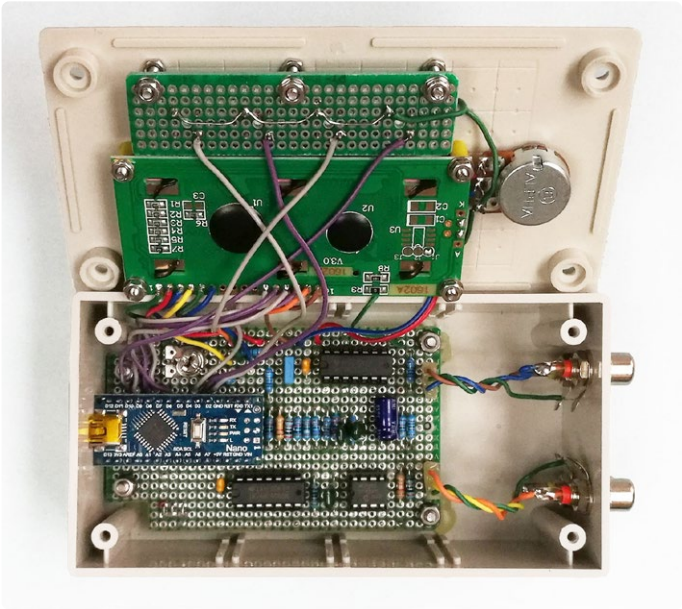
Bedieningspaneel

De gewenste uitgangsgolfvorm wordt geselecteerd door met de *MODE*-knop door de beschikbare opties te scrollen (zie **figuur 4**). In de modi sinus, driehoek, zaagtand en blokgolf kan de *FREQ/DUTY*-potmeter worden gebruikt om de uitgangsfrequentie (Fo) in te stellen. In de noise-modus wordt de potmeter gebruikt om de afsnijfrequentie van het ruisfilter in te stellen. In de sinus-, driehoek-, zaagtand-, blokgolf- en pulsmodus kunnen de knoppen *FREQ+* en *FREQ–* worden gebruikt om omhoog of omlaag te scrollen door een reeks vooraf ingestelde frequenties. Deze methode om de signaalfrequentie te selec-

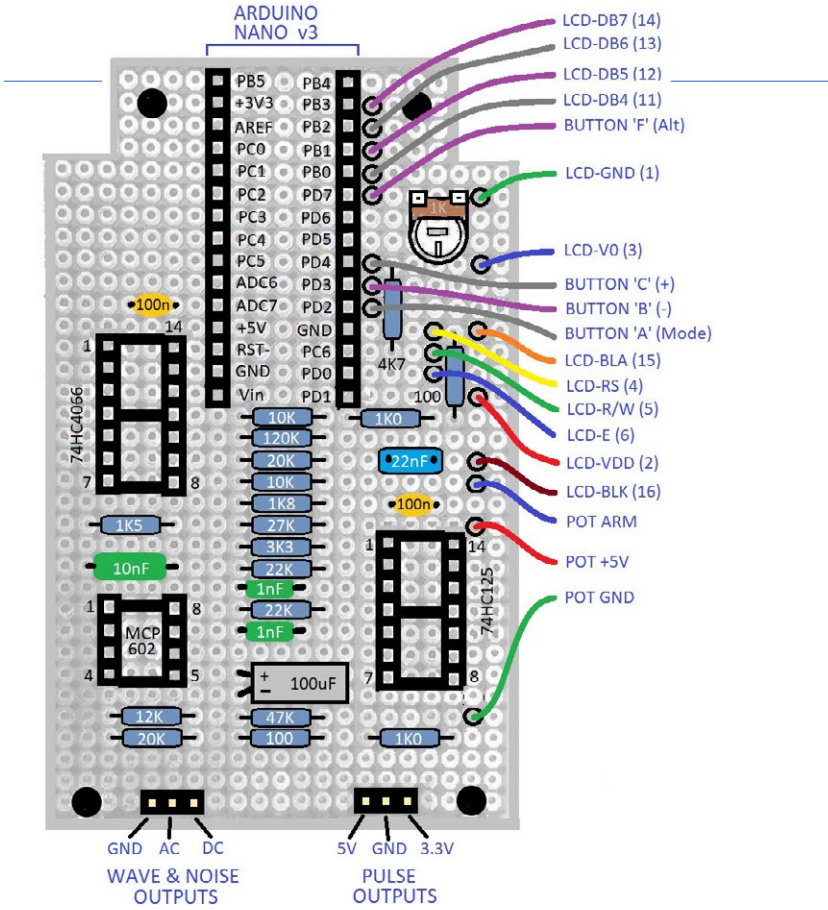


Figuur 3. Het uitgewerkte schema is niet veel ingewikkelder dan het principeschema van figuur2.

Figuur 4. Voorzijde van het prototype van de auteur met display, knoppen en potentiometer.



Figuur 5. Alle elektronica is gemonteerd in een kunststof behuizing. Het Arduino-board is tegen de zijkant geplaatst, zodat de USB-poort van buitenaf toegankelijk is.



Figuur 6. Onderdelenzijde van de gaatjesprint met de preciese positie van de componenten en voetjes.

teren geeft een nauwkeuriger instelling dan met een potmeter mogelijk zou zijn. Bovendien zijn de vaste frequentiewaarden zo gekozen dat het uitgangssignaal geen aliasing-artefacten vertoont.

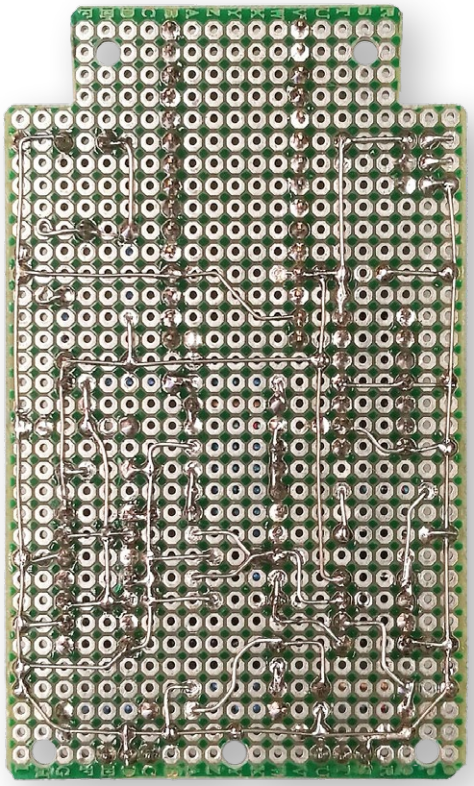
In de Pulse-modus wordt de pulsbreedte ingesteld met de potentiometer. De waarde wordt gekwantificeerd om een 'afgeronde' fijninstelling te verkrijgen, en is dus niet continu. De resolutie is afhankelijk van de frequentie. Bij hoge frequenties zijn de pulsbreedtes grof en bij lagere frequenties zijn ze fijner. De pulsbreedte wordt getoond als een percentage van de periode.

Als de *ALT*-toets ingedrukt wordt gehouden, worden alternatieve functies voor de andere drie toetsen toegankelijk. Om het frequentiebereik te selecteren, moet je de *ALT*-toets ingedrukt houden en op de *RANGE*-toets drukken om tussen het hoge en lage bereik te wisselen. Om het niveau van het uitgangssignaal in alle modi behalve Pulse in te stellen, houd je de ALT-toets ingedrukt en druk je op *LEVEL+* of *LEVEL–* om door de beschikbare niveaus te scrollen.

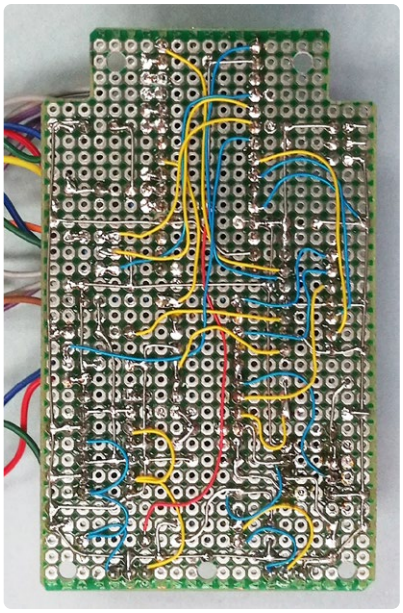
Bouw

Zoals al opgemerkt is voor deze generator geen print ontworpen. In plaats daarvan is de elektronica gebouwd op gaatjesprint met 'losse' soldeereilandjes (raster 0,1") zoals te zien in **figuur 5**. Het gebruik van een dubbelzijdige doorgemetalliseerde print is voor ons doel ideaal. Ik heb een stukje van 55 × 90 mm (met 21 × 34 eilandjes) gebruikt. Als de behuizing in elke hoek houders heeft voor schroeven om de deksel te bevestigen, moet je twee hoeken van het bord verwijderen zodat de korte kant precies tegen een smalle zijde van de behuizing past. De USB-aansluiting van de Arduino Nano moet dan aan die kant komen.

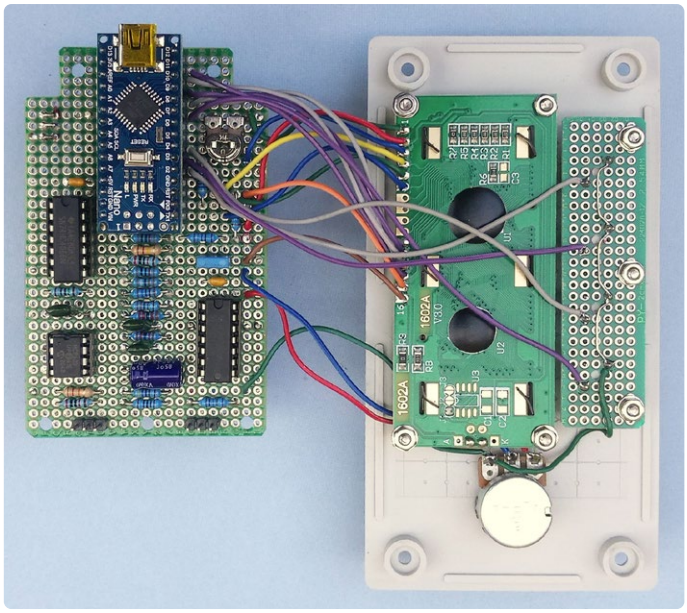
De print wordt op de bodem van de behuizing gemonteerd op vier afstandhouders van 6 mm. Het is een goed idee om die afstandhouders in de behuizing te lijmen, zodat ze op hun plaats blijven als de schroeven worden aangebracht. Vier drukknoppen zijn gesoldeerd op een apart stukje gaatjesprint, dat in de deksel is gemonteerd, onder de LCD-module. Voordat je de knoppen vastsoldeert, moet je gaatjes van 2 mm in de print boren precies in het midden van de plaats van elke knop.



Figuur 7. Onderkant van de gaatjesprint met de basisbedrading (vertind koperdraad).



Figuur 8. Onderkant van de gaatjesprint met alle bedrading.



Figuur 9. Het complete prototype met Arduino Nano, aangesloten display, knoppen en een potmeter, klaar voor een eerste test.

In **figuur 6** zie je waar de componenten aan de bovenzijde van de print zijn gesoldeerd. De bedrading aan de onderkant van de print is wat ingewikkelder: **figuur 7** toont de basisbedrading met blank vertind koperdraad rond de omtrek van de print, over de buitenste soldeereilandjes en op ruime afstand van de montagegaten. Dit is de GND-rail. Maak alle verbindingen met GND met blank vertind koperdraad, zoals aangegeven in het schema. Een draaddikte vergelijkbaar met die van de aansluitdraden van weerstanden (0,5 mm) verdient de voorkeur. De meeste verbindingen naar V_{CC} (+5 V) kunnen ook met blank draad worden gemaakt. De rest van de verbindingen kan worden gemaakt met korte stukjes dun geïsoleerd draad, zoals te zien in **figuur 8**. Verschillende kleuren kunnen helpen het overzicht te bewaren.

Nadat de print goed is gecontroleerd aan de hand van het schema, sluit je de LCD-module, de drukknoppen en de potmeter op de print aan. De gemakkelijkste en meest betrouwbare methode is om de draden rechtstreeks op de print te solderen. Plaats vervolgens de Arduino Nano-module en de IC's in hun voetjes. Stel de instelpot voor het LCD-contrast in op ongeveer de helft. Monteer de print in de behuizing zoals getoond in **figuur 9**.

Nu is het tijd voor een 'rooktest.' Sluit de Arduino Nano aan op een $5V_{DC}$-voeding via de USB-poort. De achtergrondverlichting van het display zou moeten oplichten. Als de DC-spanningen correct zijn en je nergens rookontwikkeling ziet, zou je board klaar moeten zijn om de firmware te ontvangen.

Installatie van de firmware

De firmware werd ontwikkeld met behulp van *Microchip Studio for AVR and SAM Devices* (voorheen Atmel Studio). De primaire reden om deze IDE te gebruiken en niet de Arduino IDE te kiezen is dat het hardware-ontwerp van de golfvormgenerator niet compatibel is met beschikbare Arduino-codebibliotheken. In het bijzonder lijkt de 1602A LCD-interfaceregeling (MCU I/O-pintoewijzing) niet te worden ondersteund door een Arduino-bibliotheek.

Het programmeren van de target-ATmega328P lukt zonder Microchip Studio en zonder hardware-programmeertool. Een alternatieve manier wordt beschreven in het kader **Eigen Firmware met Microchip Studio**. De Arduino Nano heeft een on-board USB/serieel-interface en een interne AVR-bootloader. Het Windows-programma AVRDUDE [1] communiceert met de bootloader via USB om firmware in het flash-geheugen van MCU te programmeren. Open na het aansluiten van je Arduino Nano op een PC het hulpprogramma *Windows Apparaatbeheer* en klik op *Poorten (COM & LPT)*. je zou de Nano USB/serieel-interface moeten zien staan. Noteer het bijbehorende COM-poortnummer en gebruik dit nummer bij het bewerken van de AVRDUDE-opdrachtregel. Het programmeren gebeurt met de Windows-opdrachtregel. Een enkele opdrachtregel volstaat. De vereiste AVRDUDE-opdrachtregel is echter vrij lang, en moet worden aangepast aan je specifieke PC software-installatie en COM-poorttoewijzing.

Kopieer en plak (of typ) de onderstaande opdrachtregel in een tekstverwerker zoals Notepad. Zorg ervoor dat je alle returns (line feeds) verwijdert, zodat het commando één enkele regel vormt. Het bestand *Program Nano.bat* dat deze commando's bevat zit in het archiefbestand dat gratis kan worden gedownload van de Elektor-projectpagina bij dit artikel [2]. Selecteer *Word Wrap* in het menu *Format* van Notepad om het leesbaarder te maken. Dit is dan die opdrachtregel:

```
avrdude.exe -C avrdude.conf" -p atmega328p -c arduino
-P COM4 -b 115200
-U flash:w:nano-wave-gen-v1.5.hex:i
```

Sommige van de vet gemarkeerde velden zullen moeten worden bewerkt om je PC-setup en de te programmeren firmware-versie aan te passen: vervang **COM4** door de werkelijke COM-poort waarmee je Arduino Nano is verbonden, zoals je die hebt gevonden met behulp van Windows Apparaatbeheer. Vervelend genoeg kan de toegewezen COM-poort veranderen wanneer de USB-kabel wordt losgenomen en opnieuw wordt aangesloten. Dit komt omdat een USB/serieel-verbinding wordt gemaakt via een 'virtuele seriële poort' – geen fysieke poort. De naam van het firmware-bestand uit [2] zal iets zijn als *nano-wave-gen-v1.5.hex*. Bewerk de AVRDUDE-opdrachtregel zodanig dat de hex-bestandsnaam (vet) dezelfde is als die van het gedownloade bestand. Sla het bewerkte opdrachtbestand op als *Program Nano.bat*.

Opmerking: sommige goedkope Arduino Nano-klonen gebruiken een niet-standaard baudrate voor de seriële bootloader, bijvoorbeeld 57.600 baud. Als AVRDUDE een foutmelding geeft, probeer dan een andere baudrate te gebruiken. Vervang dan '115200' door '57600' in de opdrachtregel.

Werkingsprincipe

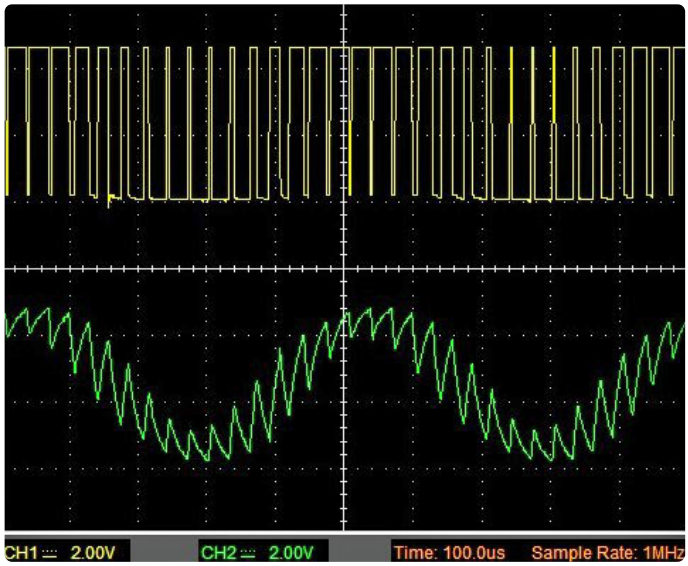
De gebruikte techniek heet 'wavetable synthesis.' Een wavetable is een array van gegevens die een volledige cyclus (één periode) van een golfvorm representeert. De array-elementen bevatten de signaalamplitudes van elk sample. De MCU voert die samples uit met een vaste frequentie, de 'sample rate.' Deze sample rate moet (veel) hoger zijn dan de maximale frequentiecomponent in het uitgangssignaal. Als vuistregel zouden we een factor vier moeten aanhouden. Om bijvoorbeeld audiosignalen tot 10 kHz te genereren, is een bemonsteringsfrequentie van ≥40 kHz nodig. Vanwege de kloksnelheid van de MCU van 16 MHz, hebben we gemakshalve een lagere bemonsteringsfrequentie van 32 kHz gekozen.

De firmware gebruikt de on-chip timer/counter-module TC0 van de ATmega328P in PWM-modus om willekeurige golfvormen te genereren in het hoorbare en niet-hoorbare frequentiebereik. De wavetable bevat de duty cycle van het PWM-uitgangssignaal. Dit nog steeds digitale PWM-signaal wordt door een laagdoorlaatfilter gestuurd dat het PWM-signaal middelt tot een analoog spanningsniveau dat de in de wavetable opgeslagen golfvorm vertegenwoordigt, zij het met een fasevertraging.

De afsnijfrequentie van 8 kHz van dit filter verwijdert de 32 kHz PWM-frequentie zoveel mogelijk, zodat alleen de gewenste audiofrequente componenten in het uitgangssignaal overblijven. De uitgang van het PWM-signaal in combinatie met het laagdoorlaatfilter werkt als DAC, waardoor de kosten van een aparte DAC-chip worden vermeden. **Figuur 10** toont het werkingsprincipe in de vorm van het PWM-signaal van een wavetable met sinusdata in de bovenste helft en de resulterende golfvorm na het laagdoorlaatfilter in de onderste helft.

Timerfuncties in de firmware

Niet alleen het kant-en-klare hex-bestand maar ook de volledige broncode is te vinden in het firmware-archiefbestand op [2]. Als je geïnteresseerd bent in hoe dit stukje software een Arduino Nano verandert in een golfvormgenerator, werp dan eens een blik op de code.



Figuur 10. Het bovenste scherm toont het PWM-signaal dat met een sinus van 2 kHz correspondeert. Na laagdoorlaatfiltering is de golfvorm in het onderste scherm het resultaat.

De `TC0_setup()`-functie initialiseert timer/counter TC0 in 'dual slope' PWM-modus (ook bekend onder de naam fasecorrecte modus) om een pulsgolfvorm met variabele duty cycle te genereren op pin OC0B (= PD5). Het Output Compare-register A van timer TC0 wordt gebruikt om een periodieke interrupt te genereren met een sample rate van 32 kHz, zodat het interval tussen twee IRQ's 31,25 µs bedraagt. De interrupt-serviceroutine (ISR) moet elke 31,25 microseconden een sample ophalen, verwerken en uitvoeren. Dat is een hele uitdaging voor een low-end 8-bit MCU, maar met een klokfrequentie van 16 MHz kunnen maximaal 16 instructies worden afgewerkt in één microseconde. In de ISR kunnen dus maximaal 16 × 31,25 = 500 instructies worden uitgevoerd. Dat is genoeg rekenkracht en het is niet nodig assembler te gebruiken.

Natuurlijk kan de MCU niet al zijn tijd in de ISR doorbrengen, anders zou er niets anders gedaan worden in de hoofd-programmalus op de achtergrond, dus moeten we ervoor zorgen dat de maximale uitvoeringstijd van de ISR veel korter is dan het IRQ-interval. Een goede keuze is om de helft van het IRQ-interval vast te leggen voor de uitvoering van de ISR.

De prescaler van de timerklok is uitgeschakeld (N = 1) om de maximale kloksnelheid te krijgen. Met een kloksnelheid van 16 MHz wordt register OCR0A ingesteld op 249 voor een timerperiode van 250 klokcycli (in beide richtingen – omhoog en omlaag), zodat de PWM-uitgangsfrequentie precies 32 kHz is. Het Output Compare-register B (OCR0B) wordt gebruikt om een PWM-signaal op pin PD5/OC0B te genereren. De PWM duty cycle wordt proportioneel gevarieerd met de sample-amplitude, en elke sample wordt elke 31,25 µs bijgewerkt door het Output Compare-register A van de timer (ISR).

Wanneer het tellerregister (TMR0) zijn maximale tellerstand bereikt (OCR0A = 250), wordt de telrichting omgekeerd en wordt er een IRQ-vlag (OCA) geze. Tegelijkertijd wordt de PWM-uitgangspen, PD5/OC0B, hoog gemaakt. Wanneer TMR0 gelijk is aan de OCR0B-waarde, wordt het niveau op pin PD5/OC0B automatisch laag gemaakt. De duty cycle van de PWM-puls is dus evenredig met de OCR0B-waarde. De duty cycle kan uiteraard niet groter zijn dan de periode, dus de maximale duty cycle (100 %) wordt verkregen door 249 in OCR0B te schrijven. Voor een meer gedetailleerde uitleg over het genereren van PWM-golfvormen met een AVR-microcontroller wordt verwezen naar de ATmega328P-datasheet [3], en wel het gedeelte over de 8-bit timer/counter TC0.

Wavetable- algoritme

Het wavetable-algoritme werkt door samples op te halen uit een wavetable ('golfvormtabel') met een vaste sample rate van 32 kHz. De frequentie van het uitgangssignaal wordt bepaald door de 'afstand' (fasehoek) tussen de punten van de wavetable. Hoe korter die afstand, hoe lager de uitgangsfrequentie. Wanneer de tabel-index van het volgende op te halen sample voorbij het einde van de tabel komt, wordt de index bijgewerkt zodat deze omklapt naar een punt binnen de tabel met behoud van de juiste sample-afstand.

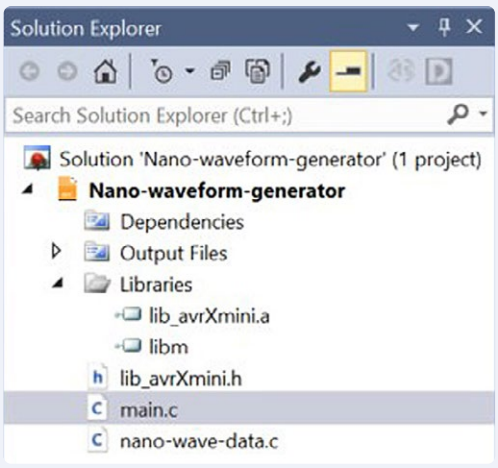
Laten we de afstand tussen de monsterpunten de 'fasehoekstap' (Phase Angle Step) noemen. Voor de meeste praktische doeleinden moet deze een reëel getal zijn, dus een getal met zowel een geheel als een breukdeel. Zwevende-komma berekeningen zouden het werk gemakkelijk maken, maar deze zijn te traag voor de ATmega328P omdat er geen floating-point hardware is geïntegreerd. Daarom worden in plaats daarvan vaste-komma berekeningen gebruikt.

Eigen firmware met Microchip Studio

Om de golfvormgenerator-firmware aan je eigen wensen aan te passen, moet je eerst Microchip Studio for AVR and SAM Devices (IDE) downloaden en installeren op je Windows PC of Mac [4]. De in [4] genoemde tutorial gaat waarschijnlijk uit van een 'targetplatform' met een Microchip- of Atmel-ontwikkelboard met een hardware-programmeertool, bijvoorbeeld Atmel AVRISP mk2 of een soortgelijk apparaat. Maar deze hulpmiddelen zijn niet werkelijk nodig om firmware in de Arduino Nano-MCU te laden. Je kunt een 'software programmeertool' in Microchip Studio creëren (zie **Optie 2** verderop).

Nadat je het firmware-archief van [2] hebt gedownload en uitgepakt, kopieer je alle bestanden naar een nieuwe map met de naam *Nano-waveform-generator* op een lokale schijf, bij voorkeur in de projectmap die Microchip Studio in je bibliotheek *Documenten* heeft aangemaakt. Sluit nu je Arduino Nano aan op een USB-poort van je PC. Open dan Microchip Studio en kies *Open Project...* op de startpagina. Navigeer naar de projectmap die je hebt aangemaakt en klik op de bestandsnaam *Nano-waveform-generator.atsln*.

Zoek naar en en klik dan aan de rechterkant van het IDE-venster op het tabblad *Solution Explorer* (zie **figuur 11**). Vouw de lijst *Libraries* uit en zorg ervoor dat de bibliotheek *lib_avrXmini.a* aanwezig is. Dit bibliotheekbestand bevat voorbereide functies ter ondersteuning van diverse periferie die vaak wordt gebruikt in ATmega328P MCU-toepassingen.



Figuur 11. Screenshot van het Solution Explorer-paneel.

Om een bronbestand te bewerken als het nog niet geopend is in de editor, klik je op de bestandsnaam in het *Solution Explorer*-paneel. Om het bestand open te houden in de editor, klik je op het punaise-pictogram in het bestandstabblad rechtsboven in het editorvenster. Het bestandstabblad wordt verplaatst naar de linkerkant van het venster en blijft geopend. Blader gerust door het headerbestand van de bibliotheek, *lib_avrXmini.h*, maar wijzig het niet, tenzij je wijzigingen in de bibliotheek wilt doorvoeren. De C-broncode is vrij om elke functie in de bibliotheek te wijzigen of meer functies toe te voegen.

Als je klaar bent met je wijzigingen, bouw dan je oplossing, repareer eventuele compilatiefouten en bouw het opnieuw. Volg dan de onderstaande instructies om de firmware over te zetten naar je Arduino Nano. Er zijn twee opties; in beide gevallen wordt aangenomen dat de AVRDUDE-distributiebestanden te vinden zijn in een map met de naam *Nano Wave Gen* op de lokale schijf van je PC.

Optie 1

Gebruik de Windows-opdrachtregel zoals eerder. Je kunt dezelfde methode gebruiken als beschreven onder het kopje *Installatie van de firmware*. Je hoeft slechts het gegenereerde hex-bestand te kopiëren naar de map *Nano Wave Gen*. Telkens wanneer je op *Build Solution* klikt, zal Microchip Studio het hex-bestand in de projectmap verplaatsen naar een submap met de naam *Debug*. Als je projectmap *Nano-waveform-generator* heet, wordt het hex-bestand weggeschreven in de submap *Nano-waveform-generator-Debug*. Hernoem voor het programmeren het gegenereerde hex-bestand zodat het identiek is aan de bestandsnaam in het Windows-opdrachtbestand *Program Nano.bat*.

Optie 2

Maak een 'programmeertool' in Microchip Studio. Klik in het menu *Tools* → *External tools*. Je zou een dialoogvenster moeten zien dat vraagt om enkele parameters: onder *Title* schrijf je *Program Nano* of wat voor naam je ook wilt. Bij *Command* schrijf je *C:\Nano Wave Gen\avrdude.exe*. Bij *Arguments* schrijf je (alles op één regel):

```
-C "C:\Nano Wave Gen\avrdude.conf" -p atmega328p -c arduino -P COM4 -b 115200 -U flash:w:"$(ProjectDir)Debug\$(TargetName).hex":i
```

Vervang COM4 door de werkelijke COM-poort waar je Nano-board op is aangesloten, en vergeet niet dat sommige Arduino Nano-klonen alleen lagere baudrates ondersteunen. Vink het vakje *Use output window* aan. Klik op *OK*. Klaar.

Je zou nu een nieuwe optie in het *Tools*-menu moeten zien met de naam *Program Nano*. Nadat de code is gebouwd, kan deze in het Nano-board worden geladen door gewoon op het item *Program Nano* in het *Tools*-menu te klikken.

Fixed-point berekeningen gebruiken snelle integer-bewerkingen met integer words (of long words) die uit twee 'bitvelden' bestaan die de 'hele' en 'fractionele' gedeelten van een getal voorstellen. Een 16-bit fixed-point getal kan bestaan uit een integer-deel van 8 bits en een fractioneel deel van 8 bits. Voor getallen zonder teken loopt het bereik van 0,0 tot 255,99 (ongeveer het decimale equivalent). Voor getallen met teken zou het bereik van –127,99 tot +127,99 lopen. De resolutie is 1/256 ≈ 0,004, wat goed genoeg is voor toepassingen met lage precisie. Een 32-bit fixed-point getal is vaak samengesteld uit een 16-bit geheel gedeelte en een 16-bit breukdeel. Voor getallen met teken is het bereik ruwweg –32.000 tot +32.000. De resolutie (nauwkeurigheid) is in dit geval ruwweg 1/65.000 ≈ 0,00001, wat meer dan genoeg is voor onze golfvormgenerator. De nauwkeurigheid van de uitgangs-signaalfrequentie wordt alleen beperkt door de klok van de MCU.
In onze wavetable-oscillator worden de Phase Angle Step en Phase Angle (golf-samplepunt) voorgesteld door 32-bits fixed-point getallen. Het gehele gedeelte van de fasehoek wordt gebruikt als array-index om een samplepunt uit de tabel op te halen. Het verband tussen de 'Phase Angle Step' en de oscillatorfrequentie is:

$$\text{PhaseStep} = \text{OscFreq} \times (\text{TableSize} / \text{SampleRate})$$

waarbij PhaseStep de afstand is tussen de samplepunten in de wavetable, OscFreq de gewenste uitgangsfrequentie (Hz), TableSize het totaal aantal samples in de wavetable, en SampleRate de bemonsteringssnelheid (32 kHz).
Hoewel een wavetable een willekeurige grootte mag hebben, moet de omvang van de tabel binnen de beperkingen van het programmageheugen overeenkomen met de resolutie van het uitgangssignaal. Als vuistregel geldt dat de grootte ongeveer gelijk moet zijn aan de maximale waarde van de samples in de tabel. Als de uitgangssamples bijvoorbeeld 8-bit getallen zonder teken zijn, zou een geschikte tabelgrootte 256 samples zijn. Een grotere tabel biedt hetzelfde voordeel als interpolatie tussen samplepunten, namelijk minder vervorming van de golfvorm.

Aliasing

Als je enige kennis van digitale signaalverwerking hebt, ben je wellicht bekend met de term 'aliasing' en de oorzaken en gevolgen daarvan. Aliasing is de vervorming van een bemonsterde golfvorm die optreedt wanneer een frequentiecomponent in de analoge ingangsgolfvorm te hoog is in verhouding tot de bemonsteringssnelheid. De vervorming manifesteert zich als ongewenste frequentiecomponenten in het bemonsterde signaal.
Een voorbeeld: indien een signaal een frequentiecomponent van 16 kHz bevat en wordt bemonsterd met 20 kS/s, zullen er nieuwe en ongewenste componenten zijn in het (gereconstrueerde) uitgangssignaal bij de som- en verschilfrequenties, dat wil zeggen bij 4 kHz en 36 kHz. De 4kHz-component ligt ruim binnen het hoorbare bereik en zou een merkbaar artefact zijn. Zuivere sinusgolven met frequenties van minder dan de helft van de bemonsteringsfrequentie hebben geen last van aliasing. Golfvormen die rijk zijn aan hogere-orde harmonischen (zoals blok-, puls- en zaagtandgolven) zijn zeer gevoelig voor aliasing. Evenzo treedt aliasing op bij wavetable-oscillatoren wanneer een golfvorm, gerepresenteerd door samplepunten van een tabel, frequentiecomponenten bevat boven de helft van de samplefrequentie (behalve wanneer de verhouding tussen uitgangsfrequentie en samplefrequentie een rationaal getal is). Een zorgvuldige keuze van de grootte van de wavetable ten opzichte van de samplefrequentie kan aliasing bij bepaalde uitgangsfrequenties elimineren. In deze golfvormgenerator bevat de tabel 640 samples en is gekozen voor vaste uitgangsfrequenties om aliasing-effecten te voorkomen.

Ten slotte

Zoals gezegd is dit geen compleet uitgeserkt DHZ-project, maar meer een demonstratie van wat er technisch mogelijk is met een goedkope Arduino Nano. Voel je vrij om de code en de schakeling aan te passen aan je eigen behoeften of je erdoor voor eigen projecten te laten inspireren.

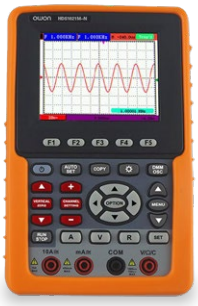
220641-03

Over de auteur

Michael Bauer is een door de wol geverfde ingenieur met belangstelling voor elektronische muziektechnologie (onder andere blaasinstrumenten en DSP-geluidssynthese). Hij woont in Victoria, Australië.

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 of naar de auteur via mjbauer@iprimus.com.au.



Gerelateerde producten

- **OWON HDS1021M-N 1-ch Oscilloscope (20 MHz) + Multimeter**
  www.elektor.nl/18778
- **Robert Sontheimer, Arduino & Co. — Measure, Control, and Hack**
  www.elektor.nl/20243

WEBLINKS

[1] Projectpagina op Elektor Labs: www.elektormagazine.com/labs/nano-waveform-generator

elektor TV

Elektor LabTalk (Engelse Show)

In onze laatste livestream hebben we de Raspberry Pi 5 onder de loep genomen en bekeken hoe je je Raspberry Pi-projecten kunt verbeteren met onze experts!

Ook de nieuwe mogelijkheden en inzichten van de Raspberry Pi 5 werden besproken in deze livestream op het Elektor TV YouTube-kanaal. Met zijn 2,4GHz quad-core Arm Cortex-A76 CPU is de raspberry Pi 5 tot 3 keer sneller dan zijn voorganger.

Bekijk nu de laatste aflevering van Elektor LabTalk: www.elektormagazine.nl/labtalk13

Heb je de livestream gemist? Geen zorgen. Bekijk de hoogtepunten van de livestream en extra content over Raspberry Pi nu op de YouTube-afspeellijst van Elektor TV.

www.elektormagazine.nl/pi-playlist

Blijf op de hoogte en volg Elektor TV:
www.youtube.com/@ElektorTV