

Synthesizer met een AT90S8515 Atmel microcontroller

deel 1

door Kruyf PAoWV

Wim PAoWV beschrijft in dit artikel hoe hij met behulp van een 90S851 microcontroller een synthesizer gebouwd heeft. Naast nabouwen kunt u de hier beschreven technieken ook toepassen voor andere frequenties en microcontrollers.



Samenvatting: Een 90S8515 microcontroller volstaat om een signaal te maken, digitaal instelbaar met slechts twee bedieningsorganen, met een resolutie van 0,1 Hz nauwkeurig instelbaar en afleesbaar in het frequentiebereik 0,1 Hz tot 350 kHz. Er kunnen eveneens andere golfvormen op logische niveaus tussen 0 en 5 V dan een sinus worden gekozen, zoals driehoek, zaagtand oplopend en dalend, een blokgolf en een benaderde diracpuls, waar hieronder wordt verstaan een puls die gedurende 1/256 van de periode hoog is en de rest van de tijd laag. Een per dB instelbare verzwakker met aflezing tussen 0 en 82 dBm voor het sinusvormige signaal completeert het geheel.

Principe van de synthesizer

Normaal kun je een kristalfrequentie delen, dan krijg je frequenties van $1/2$ $1/3$.. $1/n$ etc. van de kristalfrequentie. Wil je bijvoorbeeld een kristal iken aan de hand van DCF77, op 77,5 kHz en is de kristalfrequentie geen geheel veelvoud van 77,5 kHz, dan moet je mengen met frequenties die je uit het kristal opdeelt. Zo zijn voor het maken van 77,5 kHz uit 1 MHz zelfs twee mengtrappen nodig. Dat is niet nodig bij een synthesizer, die

een sinustabel in het geheugen van de microcontroller, zie figuur 1. De tabel heeft voor de gehele periode van de sinus 256 monsters opgeslagen die elk 8 bits zijn. Het gemiddelde ligt op 127,5 zodat de piek van de sinus op 255 ligt en het minimum op 0 en de amplitude dus 127,5 is. De sinus is dus benaderd met een nauwkeurigheid in amplitude beter dan 1% op die monstertmomenten. Omdat er 360 graden in een sinus zitten is het faseverschil (de afstand) tussen twee monsters dus $360/256$ graden. Iets minder dan 1,5 graad dus.

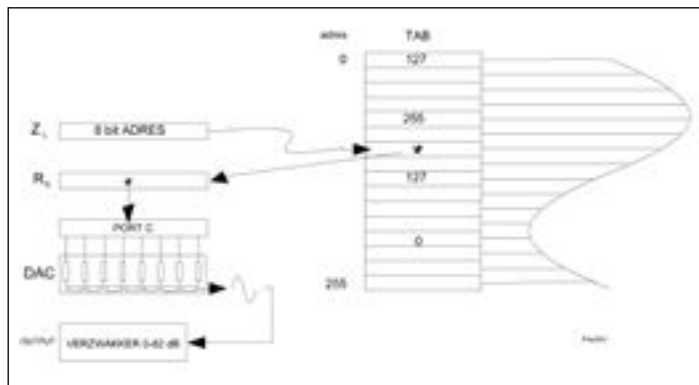
In het programmeergeheugen maken we een zo kort mogelijke loop (lus), met instructies die allemaal altijd even lang duren. Liever geen conditional branches dus, die soms 1 en soms 2 instructietijden duren.

In de loop wordt een monster uit de sinustabel gehaald, het wordt aan een 8 bits uitgangsport aangeboden, waar op de output er een analoge waarde van wordt gemaakt met een digital to analog converter (DAC). De pointer, die het monster in de tabel aanwijst, wordt verhoogd naar het volgende monster en we beginnen weer van voor af aan in de loop.

Dat wil zeggen dat als de loop 256 keer doorlopen is de gehele sinus uit de port is gekomen in de vorm van de 256 monsters van elk 8 bits en het hele verhaal weer opnieuw begint met de volgende sinus. Als we als pointer een byte nemen, dan springt dit bij verhogen als het de waarde 255 bereikt heeft vanzelf weer terug op 0, en begint dus de volgende periode van de sinus.

Frequentie opwekken

Als je dit doet blijkt dit bij de gebruikte 90S8515 een loop-tijd van 7 machine cycles te kosten en bij een klokfrequentie



Figuur 1

maakt vrijwel elke gewenste frequentie uit het basiskristal.

Om de werking begrijpelijk te maken beginnen we met

van 8 MHz geeft dat dan dat de monsterfrequentie $f_s/8$ MHz is, dat is dus de snelheid waarmee de monsters op de uitgangsport verschijnen. Een complete sinus wordt dus doorlopen in $1/256 * 8/7$ MHz oftewel met een frequentie van 4464,3 Hz. Een voorbeeld van een dergelijke lus in assemblercode staat hieronder, tijdsduur is in machine cycles.

```
main1: lpm      3 cycles: (ZL) → r0
out PORTC,r0  1 cycle: monster naar output
inc ZL       1 cycle: verhoogpointer met 1
rjmp main1   2 cycles: terug naar begin
totaal 7 cycles:  $f_s=8/7$  MHz
```

Toelichting:

- De instructie lpm laadt wat de pointer ZL aanwijst in r0, ZL wijst een monster aan in de sinustabel.
 - De instructie out zet die r0 inhoud over naar de 8 outputpennen.
 - De instructie inc ZL verhoogt de Z pointer met 1 zodat die naar het volgende monster wijst in de tabel TAB met de sinusmonsters, en als die ZL=255 was wordt die weer 0.
 - De rjmp instructie gaat weer naar het begin van de loop.
- Bij een 8 MHz kristalklok duren die instructies de tijden die erachter aangegeven staan, een cycle is dan 1/8 microseconde. Totaal dus 7 cycles, de hele loop om een monster af te geven kost daardoor 7/8 microseconde en 256 monsters van een hele sinus kosten dus 224 microseconde, oftewel de sinusfrequentie is de reeds genoemde 4464,3 Hz.

Meer dan één frequentie

We hebben nu maar één frequentie en dat willen we kunnen wijzigen, anders hebben we er nog niet veel aan. Dat wijzigen kan als eerste experiment door de pointer ZL niet een te verhogen per loopdoorgang maar bijvoorbeeld 2 of 3.

De frequentiestappen die je dan krijgt zijn echter veel te grof, namelijk ook 4464,3 Hz, want de tabel wordt dan ook 2 of 3 keer zo snel doorlopen en tevens slaan we steeds 1 of 2 monsters van de sinus over.

Daarom breiden we ZL uit in de breedte met een extra register `accu2`. Iedere keer als we de loop doorlopen verhogen we `accu2`, dat zet geen zoden aan de dijk want ZL die de monsters aanwijst blijft hetzelfde. Maar iedere keer als `accu2` van 255 naar 0 teruggaat wordt nu ZL een verhoogd, dat kan middels het carry bit die dat bijhoudt of een register (in dit geval `accu2`) 255 passeert. De loop wordt dan wel trager, die ziet er dan als volgt uit.

```
main1: lpm    3 cycles (TAB+n) → R0
out PORTC,r0 1 cycle
inc accu2    1 cycle
adic ZL,0    1 cycle
rjmp main1   2 cycle 8 cycles: fs=1MHz
```

Per loopdoorgang wordt nu `accu2` verhoogd en uitsluitend als `accu2` van 255 naar 0 gaat wordt ZL verhoogd, want de instructie `adic ZL,0` telt steeds 0 bij ZL op en de carry. Alleen als die carry 1 is wordt ZL dus verhoogd.

De tijd om de lus eenmaal te doorlopen is nu 1 microseconde dus de monsterwijzigingsfrequentie f_s uit de port is nu 1 MHz. De sinusfrequentie zou dan ruim 3906 Hz worden, ware het niet dat ZL steeds hetzelfde monster aanwijst tot `accu2` de waarde 255 heeft bereikt. Je krijgt dus 256 keer hetzelfde monster en dan 256 keer het volgende monster etc.

Daardoor is de uitgangsfrequentie nu 15,26 Hz, namelijk 1 MHz/256 monsters / 256 increments per monster. Dus de sample- of loopfrequentie 1 MHz / 2^{16} . Merk op dat die macht 16 het aantal bits is in de 8-bit registers ZL en `accu2` samen.

Verhogen we nu `accu2` per loopdoorgang niet met 1 maar met n , waarbij n een vast willekeurig geheel getal is dat kleiner of gelijk is aan 256, dan krijgen we vaker een carry naar ZL en wel n keer zo vaak, dus de uitgangsfrequentie zal dan n maal 15,25 Hz worden.

Komen we boven de 256 met n dan hebben we een sinus die minder dan 256 monsters bevat, maar dat is geen bezwaar, omdat een sinus met slechts iets meer dan 2 monsters volgens de informatietheorie middels een laagdoorlatend filter volledig vervormingsvrij te herstellen is.

Waarom dat zo is, zal verderop nog nader verduidelijkt worden. Als we het ge-

tal n opbergen in `inc3` en `inc2` dan kan n gekozen worden tussen 0 en 65535/2. Bij de hoogste waarde is er dan namelijk twee keer per sinus een carry en tevens een verhoging van ZL met 128, en dan blijven er dus twee monsters over per sinus.

De loop in het programma ziet er nu als volgt uit:

```
main1: lpm    3 cycles (TAB+n) → R0
out PORTC,r0 1 cycle
add accu2,inc2 1 cycle
adc ZL,inc3    1 cycle
rjmp main1    2 cycle
          totaal 8 cycles: fs=1MHz
```

We zien hier dat nu niet `accu2` altijd één wordt opgehoogd, maar samen met ZL met een vrij te kiezen getal dat groter dan 255 kan zijn mits kleiner dan 32768 en dat is opgeborgen in `inc2` en `inc3`. Dat getal bepaalt de frequentie van het uitgangssignaal.

Heeft `inc3` de waarde 0 en `inc2` de waarde 1 dan hebben we de laagste frequentie en die is de reeds berekende 15,25 Hz. Geven we de waarde van het 2 bytes getal `inc3 inc2` een waarde tussen 1 en 32767 (hoger mag niet want dan heeft de sinus niet meer dan 2 monsters) dan komt er een evenzoveel maal hogere frequentie uit dan 15,25 Hz.

In formule is de uitgangsfrequentie bij de instelling n van `inc3 inc2` dan $n \cdot \text{monsterfreq} / 2^{16}$. Theoretisch kunnen we dus maximaal een sinus eruit halen met een freq van 500 kHz, maar in de praktijk is dat, omdat laagdoorlatende filters niet zo steil gemaakt kunnen worden, wat minder.

Kleinere stapgrootte

Nu vinden we die stapgrootte van ruim 15Hz nog te groot en daarom slepen we er een derde registerset bij, waardoor de uiteindelijke loop van deze synthesizer ontstaat die er dan uitziet als:

```
main1: lpm    3 cycles (TAB+n) → R0
out PORTC,r0 1 cycle
add accu1,inc1 1 cycle
adc accu2,inc2 1 cycle
adc ZL,inc3    1 cycle
rjmp main1    2 cycle
          totaal 9 cycles: fs=8/9MHz
```

Je kunt nu ook het belang zien om een zo snel mogelijke loop te maken. De lustijd bepaalt namelijk direct de monsterfrequentie f_s en we hebben minimaal 2 monsters nodig per sinus, zodat de theoretische bovengrens tot waar de synthesizer gebruikt kan worden de halve f_s is. Hier dus met een stapgrootte van minder dan 0,1 Hz, wordt die 444 kHz. Immers de helft van f_s die 8/9 MHz is.

De monsterfrequentie is nu door die extra instructie gedaald tot 8/9 MHz. Dat

is bijna 889 kHz. De stapgrootte van de frequentie is nu de monsterfrequentie/ 2^{24} dus 0,0529 Hz en nog wat.

Je ziet dus ook dat als je kleinere stapgrootte wilt, de maximale frequentie daalt omdat de samplefrequentie door de loopvergroting met de uitgebreidere optelling daalt.

We hebben nu dus een synthesizer in handen die alle frequenties die een veelvoud zijn van die stapgrootte kan maken tot het theoretische maximum van de halve monsterfrequentie 444 kHz. Dat is wel aardig maar niet precies wat we willen, we willen namelijk een synthesizer die een iets grotere maar wel decimale stap maakt, namelijk een stap die dan uitkomt op 0,1 Hz.

Dat betekent dan dat we per stap van 0,1 Hz het 3 byte register ZL `accu3 accu2` niet met een increment 1 moeten verhogen, maar met een iets hoger getal. Dat getal is 0,1 Hz gedeeld door het zojuist berekende minimum increment van 0,05298 Hz.

Die berekening moet nauwkeurig gebeuren anders wordt bij grotere incrementen de fout groter dan de gewenste 0,1 Hz instelresolutie. Het getal is 1,8874 en nog meer decimalen.

Die vermenigvuldiging van de ingestelde frequentie met dit getal moet na elke wijziging van de frequentie-instelling in de synthesizer gebeuren om het 3 byte getal `inc3 inc2 inc1` te berekenen waarmee ZL `accu2 accu1` elke loopdoorgang moet worden verhoogd.

De processor kan maar een ding gelijktijdig, dus als je aan de instelling draait valt de output even weg om de nieuwe `inc` getallen te berekenen, uit de nieuwe ingestelde frequentie vermenigvuldigd met 1,88 en nog een aantal decimalen. Lang duurt dat niet, ik heb dat bij deze synthesizer gemeten en het blijkt rond de 100 microseconde te zijn. Ook wordt de frequentiedisplay dan bijgewerkt naar de nieuwe waarde.

Dat is van belang om te weten want de instelling wordt met een draaiknop gewijzigd en die heeft 25 instellingen per omwenteling en het is wel gewenst dat die snelheid die je met de hand draait kan worden bijgehouden.

Overigens is het wel zo, dat die discontinuïteit in de output bij verdraaien van de knop, het met deze synthesizer niet mogelijk maakt om bijvoorbeeld het houdbereik van een fase-lock loop te meten als dat groter is dan het vangbereik, zoals bij hogere orde loops het geval kan zijn.

Je kunt stellen dat we steeds bij elke loopdoorgang de fase uitrekenen die de gewenste sinus moet hebben en het bij die fase behorende monster op de uitgang zetten. De bytes `accu1 accu2` en ZL met een komma achter ZL gedacht, bevatten dus met een hoge nauwkeurigheid de fase van het gewenste signaal in eenheden van

360/256 graden. Dat wordt elk volgende monstertijd verhoogd met het 24 bits getal in de 3 inc registers, het faseverschil of fase-toename van de gewenste uitgangssinus per programmaloopdoorgang. ZL accu2 accu1 zijn dus samen de faseaccumulator die elk moment de fase van de gewenste sinus aangeven in binaire vorm.

plaatsen (Jmp1 in figuur 12) zijn die tabellen te activeren, als de jumper afwezig is worden ze overgeslagen in het keuzemenu op het LCD display.

Digitaal naar Analooq conversie

Er zijn diverse methoden om een binair getal om te zetten in een analoge waarde.

Een DAC van het type R-2R kan uit een handje weerstanden van allemaal dezelfde waarde worden gemaakt, in dit geval 33 kΩ voor 2R en twee stuks parallel voor R. Het zijn in mijn probeersel gewone 5% handelsweerstanden. De waarde is niet kritisch, ik had gewoon een patroonbandje 33 kΩ in een bakje liggen. Die weerstanden moeten wel zo goed mogelijk gelijk zijn en dat leek me wel het geval als ze van zo'n patroonbandje naast elkaar worden afgeplukt. Meting leert dat een dergelijke DAC nog net op het oog op een scope de volle amplitude haalt als beurtelings wordt geschakeld tussen 0 en 255, dus een blok van 444 kHz wordt aangeboden. Het is dus door parasitaire capaciteiten net niet te traag.

Theorie van het R-2R netwerk

Het schema van een dergelijk netwerk staat in figuur 2.

Elk van de dwarstakken 2R hangt aan een van de bits van de digitale byte, dat qua grootte de spanning aangeeft. 255 = hoogste outputspanning en 0 = 0 V. Het is een lineair netwerk, dat wil zeggen dat we de uitgangsspanning kunnen berekenen door de invloed van elke pen apart te bekijken, terwijl de andere pennen 0 zijn en aan massa liggen. De uitgangsspanning is dan de som van de invloeden van elke berekende penspanning afzonderlijk.

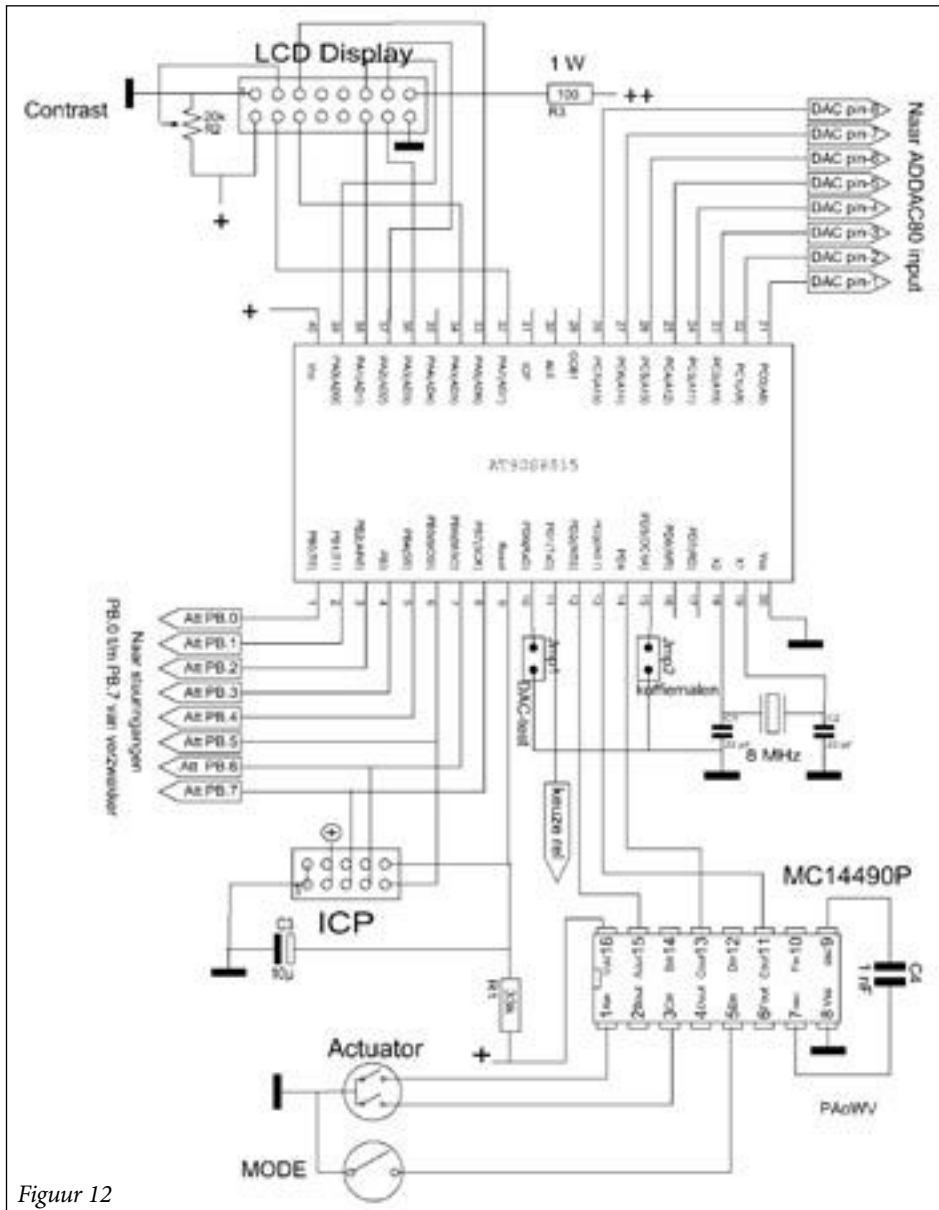
Het netwerk heeft aan het begin bij het minst significante bit naar links een afsluiting van 2R naar aarde, aan de rechterkant is de output en die moet een belasting vormen van 2R.

Is dat het geval, dan kan gemakkelijk worden ingezien, dat vanuit elke dwarstak 2R naar links kijkend een weerstand 2R wordt gezien en naar rechts kijkend eveneens een weerstand 2R.

De verklaring aan de hand van figuur 2 is als volgt: De weerstand 2R, die je naar rechts ziet, wordt met de weerstand 2R van de dwarstak samen R. Die staat in serie met een langstak R om een sectie naar links weer een impedantie van 2R naar rechts kijkend te geven.

Kijk je nu naar de spanningen: Een spanning U op een dwarstak ingang geeft stroom die gaat door de dwarstakweerstand van 2R en splitst zich vervolgens in 2R die hij naar links ziet en 2R die hij naar rechts ziet. Op het knooppunt van de ladder waar de dwarstak op aangesloten is staat dan dus U/3.

Naar rechts zie je daar 2R bestaande uit de serie tak R en de impedantie die je naar rechts ziet van R juist links van het volgende sectiepunt, die ook R is. Er gaat dus stroom lopen naar rechts die een spanning op het volgende sectiepunt geeft die de helft is van het sectiepunt waarvan we uitgaan.



Figuur 12

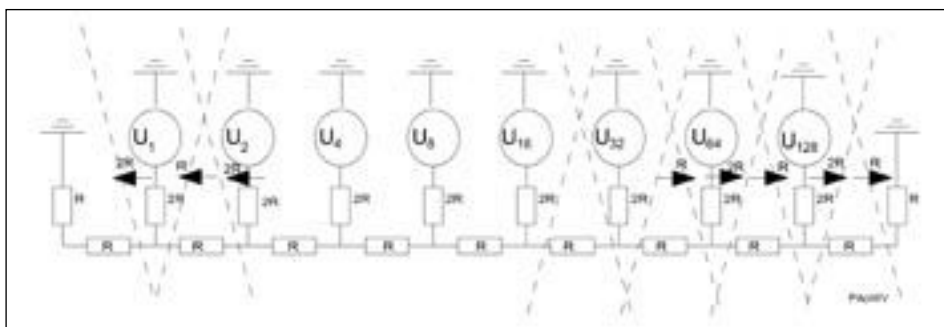
Andere golfvormen

Nu we de theorie weten, zal het duidelijk zijn dat we ook andere golfvormen kunnen opwekken, die we in andere tabellen van 256 monsters kunnen opbergen. Daarbij is gekozen voor een driehoek een zaagtand omhoog een zaagtand omlaag, een symmetrische blok golf en blok golf die slechts een monster per periode hoog is en de andere 255 monsters laag, dat is een benadering van een zogenaamde Dirac puls.

Ik heb ook nog 9 tabellen opgenomen die de monsters beurtelings 0 en 255 maken en beurtelings 0 en 128 etc., maar die zijn slechts gebruikt om de DAC op de uitgangsport te testen, waar we het nu even over zullen hebben. Door een jumper te

Een R-2R netwerk, een officieel DAC IC en met rate multipliers.

Rate multipliers bijvoorbeeld SN74167, laten zoveel pulsen door van een aangeboden kloksignaal als een binair of BCD getal aangeeft. De gelijkspanningscomponent van de doorgelaten pulsen is een maat voor het analoge signaal. Dat kunnen we dus krijgen als we de uitgangspulsen in een RC-netwerkje stoppen. Rate multipliers heb ik toegepast om een PSK31 en een QPSK signaal te maken met een microcontroller. Maar voor de synthesizer zijn ze niet geschikt, omdat de verhouding hoogste frequentie (8 MHz) en hoogste synthesizer frequentie (340 kHz) daarvoor te laag is.



Figuur 2

Conclusie: elk sectiepunt geeft een bijdrage aan de uitgangsspanning die steeds gehalveerd wordt als het voedende punt een stap naar links gaat. Dat is dus een bijdrage in overeenstemming met de binaire waarde van het bit.

Een R-2R netwerk is geprobeerd en met een audiovervormingsmeter is de vervorming gemeten tussen 18 Hz en 28100 Hz, dat is het bereik voor de grondfrequentie dat de meter accepteert.

We komen op de volgende vervormingspercentages voor dit netwerkje, zie figuur 3.

Frequentie vervorming

Frequentie (Hz)	Vervormingspercentage (%)
18 Hz	0,82%
50 Hz	0,78%
100 Hz	0,73%
250 Hz	0,61%
500 Hz	0,49%
1000 Hz	0,42%
2500 Hz	0,37%
5000 Hz	0,35%
10000 Hz	0,43%
20000 Hz	0,37%
28100 Hz	0,53%

Figuur 3: Vervormingspercentage van de sinus uit het R-2R netwerk.

Dat er enige vervorming moet optreden kan verduidelijkt worden door het feit vast te stellen dat alle spanningen van de sinus in gehele getallen tussen 0 en 255 worden benaderd. Gemiddeld zal de amplitude van fouts spanning dus 0,5 zijn terwijl de amplitude van de sinus 128 is. Dat is dus 0,4 procent fout, dat is de zogenaamde quantiseringsfout, die hier dus leidt tot quantiseringsvervorming, die zich vaak uit in ruis en daarom ook quantiseringsruis genoemd wordt. Die ruis ligt dus bij 8 bit monsters op 0,4% en dat is 48 dB onder het signaalniveau en die zakt bij verzwakken van het signaal in een verzwakker gewoon mee naar beneden.

Een dergelijk netwerk van handelsweerstand heeft, zoals blijkt uit de vervormingsmeting, fouten, die de signaal stoorverhouding iets verslechtert en bovendien moet het worden afgesloten met

een opamp, omdat het nauwkeurig belast moet worden. Met 33 kΩ in dit geval en daarachter komt dan de 50 Ω verzwakker. Om aan die bezwaren tegemoet te komen is het veel eenvoudiger een DAC in de vorm van een IC te kopen, dat op basis van een R-2R principe werkt. De ADDAC80, die dank zij Harry PAoLQ hier in de junkbox belandde, is 12 bits waarvan we er slechts 8 gebruiken en de opamp is ingebouwd. Bovendien kunnen ze een spanning afgeven symmetrisch rond 0. Wel zo prettig, want voor heel lage frequenties heb je dan geen scheidingscondensator nodig. Hoewel voor niet sinusvormige output is in praktische toepassingen te prefereren een spanningsminimum 0 en max 5V. Daarom zijn maatregelen genomen om automatisch om te schakelen, afhankelijk van de gekozen signaalvorm. Vervelend is wel dat die ADDAC80 een voeding van + en - 12 tot 15 volt nodig heeft.

Daar moet dus apart voor worden gezorgd. Er zijn 5V naar ±12 volt 1 W converters in de handel (CD technologies HPR1004), maar de prijs daarvan ligt rond de € 20,-, bij een Nederlandse leverancier. Een dergelijke converter kan de voeding van het IC wel voor zijn rekening nemen. Doorgaans zal het dus goedkoper zijn om zelf voor de vereiste + en - 12 volt te zorgen door een wat uitgebreidere voeding toe te passen, wellicht kan een MAX232 er oneigenlijk voor gebruikt worden.

Signaalspectrum

Een sinusvormig signaal heeft geen harmonischen, zie figuur 4.

Als we een periodieke puls nemen in figuur 5 getoond, dan heeft die in tegenstelling tot een sinus harmonischen op veelvouden van de grondfrequentie.

Onze monsterreeks van 888 kHz grond-

frequentie heeft bij constante amplitude dus harmonischen op $n \cdot 888$ kHz met n een geheel getal.

De harmonischen zwakken af in amplitude naarmate de frequentie hoger wordt, maar als de monsterpuls smaller worden treedt dat afzakken pas bij hogere frequenties op. Dan zijn er dus meer harmonischen. Theoretisch is bij een oneindig smalle monsterpuls het spectrum van harmonischen dan ook oneindig breed en zijn alle harmonischen even sterk. Wordt de monsterpuls breder bij dezelfde energieinhoud (dus lagere amplitude), dan zwakt het spectrum af bij hogere frequenties, zie figuur 6.

De amplitude van de harmonische op frequentie f met c een constante, is dan evenredig met $\sin(cf)/cf$, die gaat regelmatig door 0. Bij het groeien van f neemt de amplitude van de frequentiecomponenten verder af door de f in de noemer.

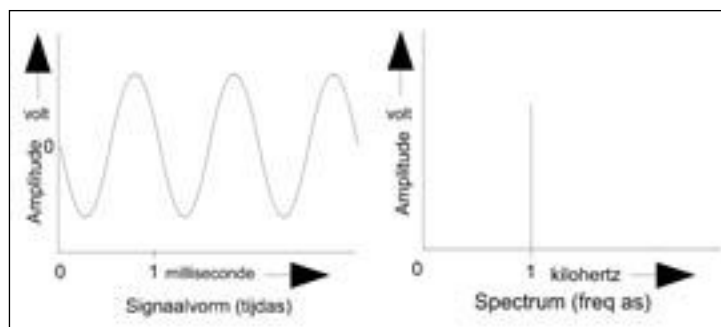
Bekend is dat een symmetrische blok-golf geen even harmonischen heeft, alleen oneven met amplitudes 1/3 1/5 1/7 etc. van de grondgolf. Dat komt omdat een symmetrische blok de $\sin(cf)/cf$ juist zijn nulpunten heeft op de even harmonischen. Zijn maxima er dus midden tussen. De sinus is daar dan 1 of -1 en de f in de noemer zorgt voor die 1/3 1/5 1/7 etc. van de amplitude van harmonischen op frequenties 3f, 5f, 7f etc. Het minteken is een fasedraaiing van 180 graden en is dus niet bepalend voor de amplitude van de harmonische.

Bemonsteren signaal

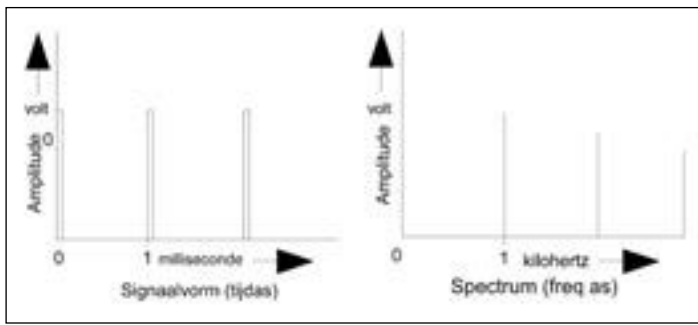
Zitten we op de halve monsterfrequentie, dan hebben we twee monsters per sinus, dat is dus een symmetrisch blok, die heeft geen even harmonischen dus de monsterfrequentie valt in een 0-punt van $\sin(\pi \cdot f/fs)/(\pi \cdot f/fs)$. Evenzo alle harmonischen van de monsterfrequentie. Leuk, daar hebben we dan dus alvast geen last van, maar dat betekent wel dat als we de synthesizer gebruiken tot 0,4 maal de monsterfrequentie (350 kHz) de amplitude hierdoor al verminderd is tot precies $\sin(\pi \cdot f/fs)/(\pi \cdot f/fs) = 0,76$, dus zijn we daar al bijna 2,5 dB kwijt. Als je het weet kun je er rekening mee houden. Een gewaarschuwd mens telt immers voor 6 dB.

Bemonsteren we een gelijkspanning dan hebben alle monsters dezelfde amplitude, namelijk de hoogte van de gelijkspanning.

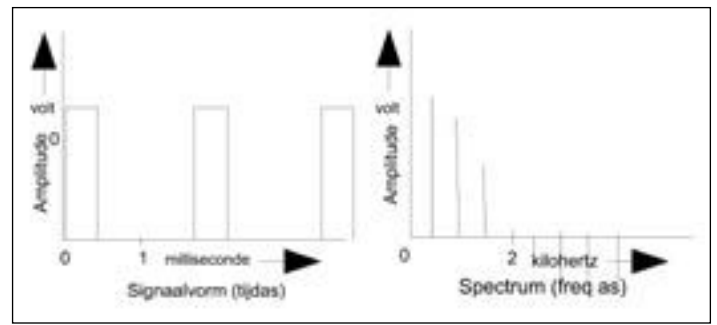
Gaan we in fig. 7 een sinus met hoekfrequentie



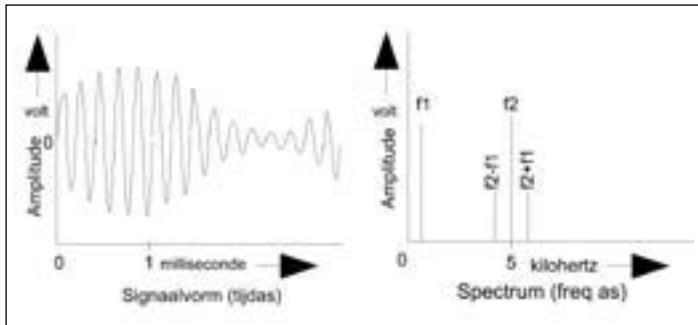
Figuur 4



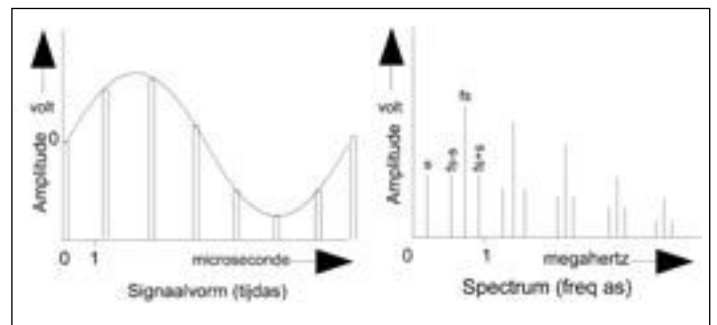
Figuur 5



Figuur 6



Figuur 7



Figuur 8

f_1 in amplitude wijzigen met een sinusvormig verloop met hoekfrequentie f_2 , dan kunnen we dat schrijven als een amplitude $1 + \cos(f_2.t)$. De cosinus is een 90 graden verschoven sinus. Die amplitude van een laagfrequentperiode begint dus bij 2 daalt naar 0 en eindigt weer bij 2, omdat de cos evenals de sin op en neer gaat tussen 1 en -1 .

De draaggolf nemen we $1 + \cos(f_1.t)$. Door gaans is f_1 veel groter dan f_2 , maar dat hoeft helemaal niet. Die constante waarde 1 zorgt ervoor dat de minimum spanning 0 is en de maximum spanning 2. Er is dus een gelijkstroomcomponent ter waarde 1 in dat signaal. We hebben dan amplitude-modulatie van draaggolf op hoekfrequentie f_1 met modulerende hoekfrequentie f_2 , en wel 100% modulatie diepte. Het signaal is dus $[1 + \cos(f_2.t)] * [1 + \cos(f_1.t)]$.

In de wiskundeles op school hebben we geleerd dat $\cos(a).\cos(b) = 0,5.\cos(a+b) + 0,5.\cos(a-b)$. Toch niet voor niks geleerd dus.

Vullen we dat in dan krijgen we voor het signaal

$$1 + \cos(f_1.t) + \cos(f_2.t) + 0,5.\cos(f_1+f_2)t + 0,5.\cos((f_1-f_2)t)$$

dan vinden we dus, zoals bekend uit de theorie voor het zendexamen, dat een 100% amplitudegemoduleerd signaal bestaat uit de ongemoduleerde draaggolf f_1 en twee zijbanden van halve amplitude die f_2 lager en hoger liggen dan f_1 . In dit geval, omdat de draaggolf een gelijkstroomcomponent 1 had tevens die gelijkstroomcomponent en de zijband daarvan en dat is het modulerende signaal $\cos(f_2.t)$ zelf met volle amplitude omdat de andere zijband van de gelijkstroomcomponent gespiegeld wordt van de negatieve naar de positieve

frequentie-as.

Nu gaan we in figuur 8 geen sinusvormig signaal in amplitude moduleren maar een pulsreeks van de sample frequentie f_s , die bij ons 888 kHz is.

Die pulsreeks bestaat uit sinussen op 888 kHz als grondfrequentie en veelvouden daarvan en een gelijkstroomcomponent, die de gemiddelde waarde van al die monsters is. Halveren we de amplitude van de monsters, dan halveren alle spectrumcomponenten in amplitude, inclusief de gelijkstroomcomponent.

Gaan we dus de pulsreeks amplitude moduleren, met freq s dan krijgen we dus dat AL die harmonischen en de gelijkstroomcomponent amplitude gemoduleerd worden en dus krijgen de gelijkstroomcomponent en alle harmonischen dan zijbanden die s hoger en s lager liggen dan die harmonischen.

Omdat de pulsreeks een gelijkstroomcomponent heeft (de gemiddelde waarde van de pulsen) wordt die ook amplitude gemoduleerd en dat is het modulerende signaal zelf met door de spiegeling al dubbele amplitude als de zijbanden van de eerste spectrumcomponent op 888 kHz hebben.

Die monsters met frequentie 888 kHz uit de synthesizer zijn inderdaad amplitude gemoduleerd met de gewenste ingestelde uitgangsfrequentie s van de synthesizer. Die monsters gaan immers in amplitude op en neer tussen 0 en 255.

Je krijgt dus als totaal spectrum de (gewenste) frequenties en voorts de sample frequentie van 888 kHz en alle veelvouden daarvan die elk zijbanden hebben die s hoger en s lager liggen dan die frequentie.

Nu is het meest bedreigende de dichtst bijliggende frequentie bij de gewenste s , die de ingestelde synthesizerfrequentie is. Als we s wensen, dan is die stoorfrequentie f_s , de onderzijband van de sample frequentie dus. Die moeten we kwijt en daarom gebruiken we een laagdoorlatend filter, die s wel doorlaat en f_s niet. Je kunt niet verder gaan dan dat f_s en s elkaar dicht genaderd zijn. Dit is het geval als s bijna op de halve sample frequentie ligt. Dus bijna op 444 kHz, nog net iets eronder, de onderzijband is dan genaderd tot f_s -s en is dan dus net iets hoger dan 444 kHz.

Reconstructie sinus

Hiermee zie je dus dat het theoretische maximum als je een oneindig steil laagdoorlatend filter gebruikt de halve sample frequentie is. Met een laagdoorlatend filter kun je dus de gewenste component op s eruit krijgen, zonder storing, een zuivere sinus dus, en belangrijk op te merken dat je dan slechts 2 monsters per sinus hebt en die dus volstaan.

Hiermee is dus de wellicht raadselachtige theorie, die het bemonsteringstheorema van Nyquist wordt genoemd, dat je een sinus volledig kunt herstellen met een laagdoorlatend filter, als je maar tenminste ietsje meer dan 2 monsters per sinus hebt, verduidelijkt. Neem je minder dan 2 monsters per sinus dan valt de zijband van f_s in je doorlaatband en dat heet dan vouwoverspraak in het Nederlands en aliasing in het Engels. Het filter wordt ook wel het reconstructiefilter genoemd. Bij het zendexamen wil de examencommissie dat je die naam kent.

De frequentie die je maximaal kunt bereiken is hoger naarmate de lus in de microprocessor korter duurt. Helaas duurt hij

wat langer als we extra instructies moeten opnemen in de lus om de stapgrootte van de synthesizer te verkleinen. Hier is de theoretisch maximale frequentie dus 444 kHz, bij een stapgrootte die een fractie is van een Hertz.

Een oneindig steil filter is niet mogelijk, maar als we gaan rekenen dan blijkt dat we met 4 spoelen en 5 condensatoren een negende orde Chebyshev filter kunnen maken, dat tot 340 kHz nog net niet noemenswaardig gaat dempen en dat de onderzijband op $888-340=548$ kHz, ruim 50 dB dempt.

Schone sisusvorm

Na dit verhaal ligt het voor de hand te veronderstellen dat je dan tot 340 kHz een volkomen schoon sinusvormig signaal kunt maken. Dat is iets te optimistisch gedacht. De monsters uit de tabel hebben namelijk een afgeronde waarde voor de amplitude op het naastliggende gehele getal voor elke waarde tussen 0 en 255.

De maximale fout die de amplitude kan hebben is dus ongeveer $1/512$ deel van de piekwaarde en dat is $1/256$ van de amplitude van het signaal.

Het afrondingssignaal heeft de vorm van een blok golfje gesuperponeerd op de sinus met een amplitude van 0,25% van de volle amplitude. De frequentie van dat blokje is de halve sample frequentie (444 kHz) en wordt dus nog eens flink onderdrukt in het filter.

Het is echter heel goed mogelijk, omdat de fase van het gewenste signaal op de monstermomenten in de lus door optelling steeds wordt berekend, dat de fouten van naburige monsters niet alterneren maar

gelijk zijn, in dat geval heb je dus een lagere frequentie van het storende blok golfje.

Afhankelijk van de ingestelde frequentie kun je dus in principe overall storing verwachten in de grootorde van 0,4% van de amplitude van het gewenste signaal, dat is een stoorniveau van -48 dB. Dezelfde orde van grootte als het filter levert aan demping van de onderzijband van de sample frequentie, bij het bereiken van de maximale frequentie van 340 kHz.

De ongewenste fs-f component op 548 kHz is dan al door de Si omhullende van het spectrum 6,35-2,21 is ruim 4 dB zwakker dan de gewenste component en daar komt de sperdemping van het filter dan bij, zodat de aliassing dan nog steeds verwaarloosbaar is en tevens significant geringer is dan de quantiseringsruis op -48 dB.

Het steeds optellen van de fase kan een volgend monster net wel of net niet halen, dat geeft ook een geringe fasejitter die bij 8 bits in dezelfde orde van grootte ligt, omdat de onzekerheid $1/256$ van 360 graden is.

Een en ander is een stoorniveau, dat voldoende laag is om de synthesizer binnen de machtigingsvoorwaarden als stuurzender voor de langegolfband te kunnen gebruiken.

Frequentiebereik

Nu hebben we gezien dat we niet alleen sinussen maar ook blokken, korte pulsen, driehoeken en zaagtanden uit de synthesizer kunnen halen. Die hebben harmonischen en het is dus duidelijk dat je geen hogere frequentie mag kiezen voor een andere golfvorm dan de sinus, dan dat je hoogste gewenste harmonische er nog doorkan en dus beneden 340 kHz ligt.

Boven 170 kHz komt zelfs de tweede har-

monische er niet door, boven 110 kHz de derde niet etc., zodat je met de uitgangsfrequentie voor die golfvormen je niet kan permitteren veel hoger dan 20 à 30 kHz te gaan.

Daar komt nog bij, dat de fasekarakteristiek van het filter in de doorlaat niet lineair is en dus de harmonischen niet in de goede fase er doorkomen, wat ook weer signaalvorming geeft. Maar voor lagere frequenties speelt dat geen rol, omdat de harmonischen dan ook vrij dicht bij elkaar liggen.

Voor sinusoutput kun je wel ongestoord tot de maximum frequentie van 340 kHz doorgaan.

Kijk je nu naar de pseudo diracpuls die een monster hoog is en 255 monsters laag, dan kun je inzien dat het hoge monster nooit mag worden overgeslagen, want dan mis je een puls. Dat betekent dus dat de frequentie voor de diracpuls principieel beperkt is tot 3472 Hz als je er geen wil missen uit een periodiek pseudo diracsignaal. De zaagtand en driehoek kun je wel een stuk verder gaan, maar hoger dan 20 kHz geeft teveel zuster Buitenhuis-effecten.

Bovendien willen we die van een sinus verschillende golfvormen bij voorkeur tussen 0 en 5 volt, met een gelijkstroomcomponent dus, hebben voor testsignaaltoepassingen en is demping ervan met een verzwakker daarvoor niet interessant. Ze behoeven dus een aparte behandeling. Daarin is voorzien, met een relais wordt de output van de DAC omgeschakeld naar een aparte BNC connector op het voorpaneel, de signaalbron heeft een Ri van 0 en is kortsluitvast. Lijkt in tegenspraak maar alles heeft zijn grenzen. Het betreffende deelschema staat in figuur 13.

(wordt vervolgd)

Voorpagina CQ-PA nr. 2

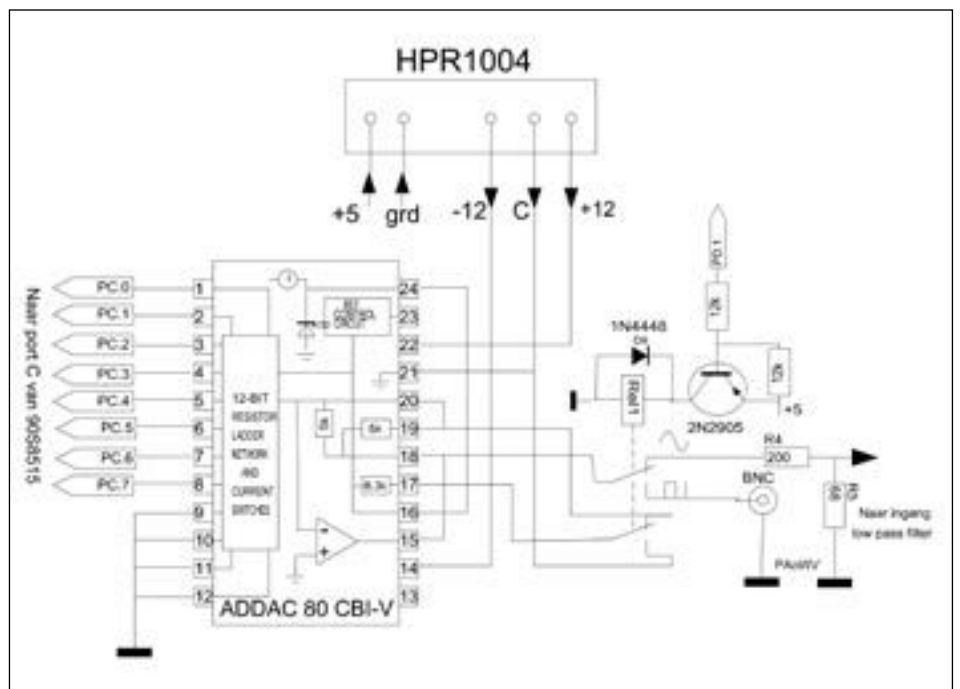
Op de voorpagina van CQ-PA nr. 2 stond een circa 90 jaar oude microfoon afgebeeld.

Niemand heeft dit antwoord gedaan, hoewel Henk Vrolijk PAoHPV er wel dichtbij zat.

Hij vermoedde dat dit een soort weergever (soort luidspreker) was.

Het voorwerp is afkomstig uit de collectie van Wim van Ulden PA3DMT.

Ook in deze maand hebben we weer zo'n raadsel voorwerp. Deze keer hebben we het niet zo moeilijk gemaakt en hopen dan ook op veel goede antwoorden.



Figuur 13