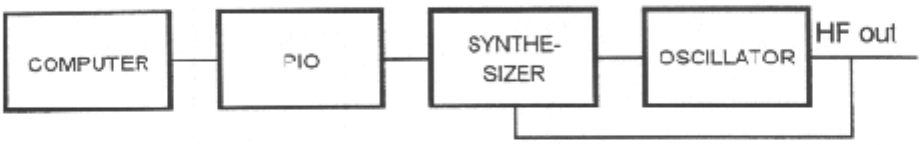
Programmeerbare universele synthesizer
(512 kHz tot 406,5875 MHz)

door Henk van den Hof, PE1BVJ.

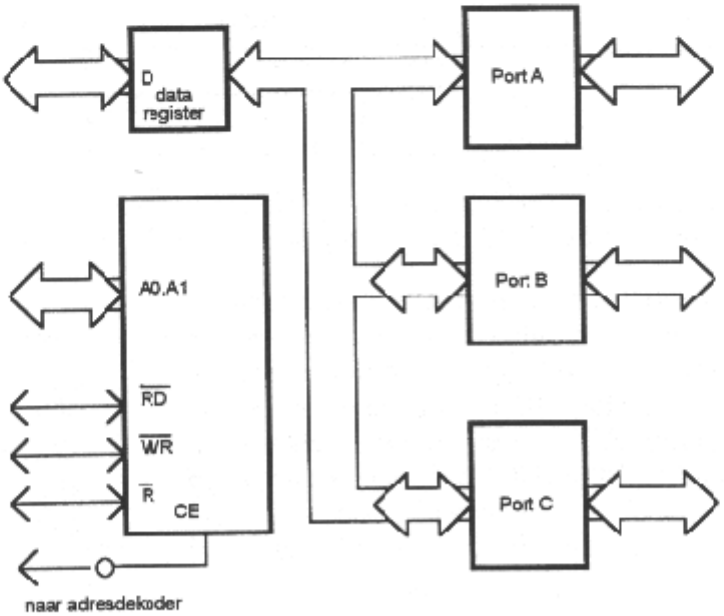
Tijdens de vele experimenten die ik doe, komt het vaak voor dat ik een stabiele oscillator nodig heb. Een oscillator is eenvoudig te bouwen, maar de stabiliteit laat vaak te wensen over. Zodoende ben ik eens in de boeken op zoek gegaan naar een synthesizer. Deze moest wel een groot frequentiebereik hebben om universeel inzetbaar te zijn. In een van de boeken vond ik een synthesizer IC, de SAA 1057 van Philips die programmeerbaar is. De besturing van het IC kan vanuit een microcontroller gebeuren. Nu heb ik weinig kaas gegeten van microcontrollers en aanverwante artikelen en vroeg mij dan ook af, of dit ook met mijn PC gedaan kon worden. Tijdens het snuffelen in de boeken kwam ik zo een schakeling tegen, die in een PC geplaatst kan worden en die de signalen naar de synthesizer kan sturen. Deze schakeling wordt een PIO ofwel Peripheral Input Output Interface genoemd. De PIO verstuurt de data van de PC naar de synthesizer. In het blokschema, figuur 1, is te zien, hoe het een en ander aan elkaar gekoppeld dient te worden.



figuur 1, blokschema

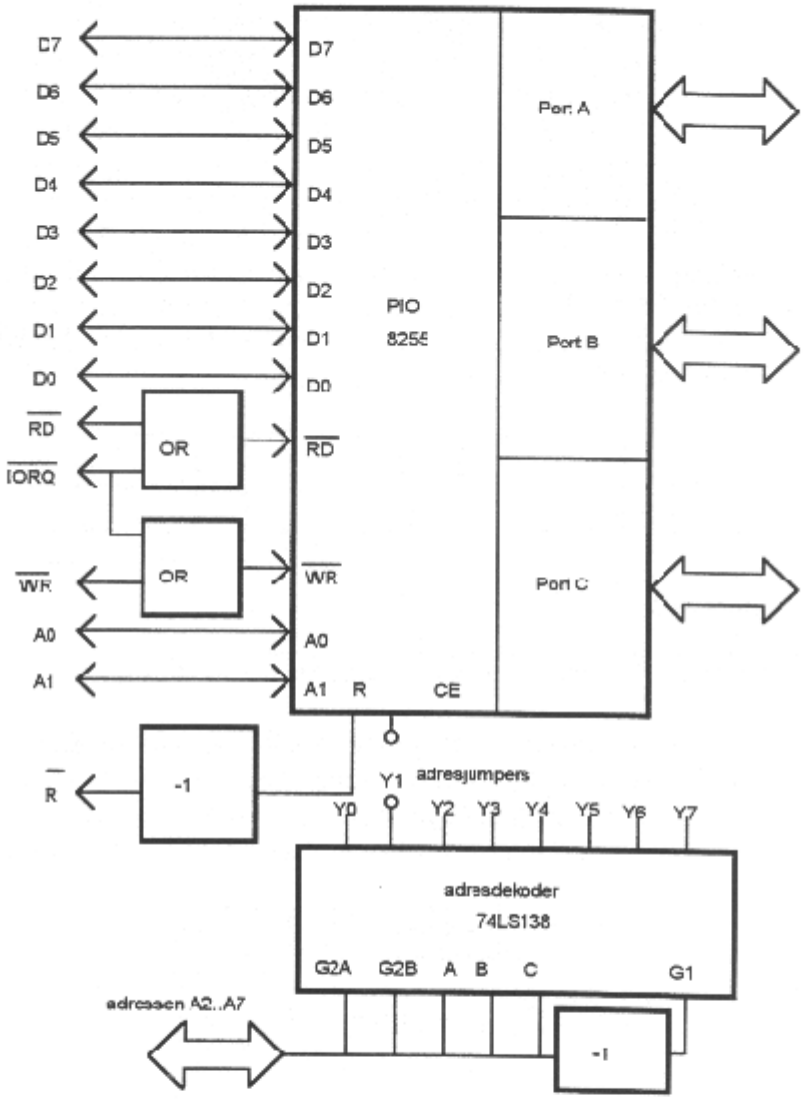
De PIO: interface tussen computer en synthesizer.

Er zijn tal van PIO's verkrijgbaar. In dit geval is gekozen voor de 8255, een PIO met drie 8 bit brede poorten. Dit IC heb ik gekozen omdat door zijn 24 uit- en ingangen eventuele verdere experimenten om het geheel tot een volwaardige transceiver te vervolmaken tot de mogelijkheden behoort. In figuur 2 ziet u het inwendige van het PIO IC.



Figuur 2, het inwendige van de 8255 PIO

Naar de kant van de computer zien we de 8 bit brede databus, twee adresbussen A0 en A1, en de stuurlijnen READ (RD), WRITE (WR) en RESET (R). Intern beschikt het IC over de nodige besturingslogica en de invoer- en uitvoerregisters voor de drie poorten A, B en C. Behalve het PIO IC is er nog een adresdecoder, de 74LS138, nodig om het PIO IC aan te kunnen sturen vanuit de PC. Figuur 3 toont het inwendige van de adresdecoder 74LS138.



Figuur 3

De adresdecoder decodeert 6 adresbits. Met de adresjumpers (Y0 t/m Y7) kan het gewenste adres gekozen worden. De laagste twee adreslijnen A0 en A1 gaan rechtstreeks naar de adresingangen A0 en A1 van het PIO IC. Als de computer het hardwarematig geprogrammeerde adres uitgeeft, schakelt de betreffende uitgang van de 74LS138 om naar 0 en wordt de PIO-chip select ingang (CE) geactiveerd. Als voorbeeld nemen we adres 200. In de tabel is te zien welk adres bij welk register hoort.

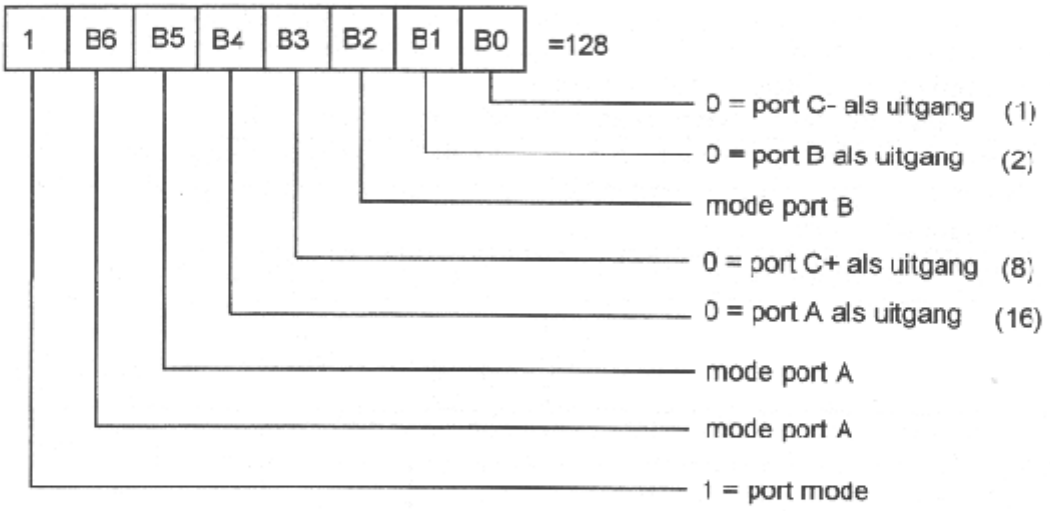
adres	register
200	A
201	B
202	C
203	stuurregister

Programmering van de PIO.

Voordat de PIO het datatransport van en naar de PC voor zijn rekening kan nemen, moet via een "control word" opgegeven worden, welke van de 3 poorten als ingang resp. uitgang geschakeld moet worden. Dit noemt men initialisering. Er moet dus een 8 bits commando ("woord") naar adres 203 gestuurd worden waarvan de afzonderlijke bits de volgende betekenis hebben:

B7=1	hiermee wordt de "port mode" van het "control word" geselekteerd
B6,B5	bepalen de mode van poort A
B4=0	schakelt poort A als uitgang
B3=0	schakelt de hoogste 4 bits van poort C als uitgang
B2	bepaalt de mode van poort B
B1=0	schakelt poort B als uitgang
B0=0	schakelt de laagste 4 bits van poort C als uitgang

In figuur 4 ziet u een beknopte weergave van de bits.



Figuur 4

Voor elke poort die als ingang (mode = 1) wordt gebruikt moet het erachter vermelde getal bij 128 worden opgeteld. De volgende listing in BASIC schakelt bij adres 200 alle poorten als uitgang:

<u>listing 1.0</u>

```
10    OUT 200,128
20    A=123              :REM A=0...255
30    OUT 200,A
```

Met het commando OUT 200,128 maken we eerst alle poorten uitgang. Met het commando OUT 200,A wordt een getal A naar poort A gestuurd. De waarde van A mag liggen tussen 0 en 255 en werd in regel 20 bepaald.

De volgende listing in BASIC laat zien hoe we een waarde via poort B kunnen inlezen:

listing 1.1

```
10    OUT 200,132        :REM B=INGANG
20    INP 202,B
30    PRINT B
```

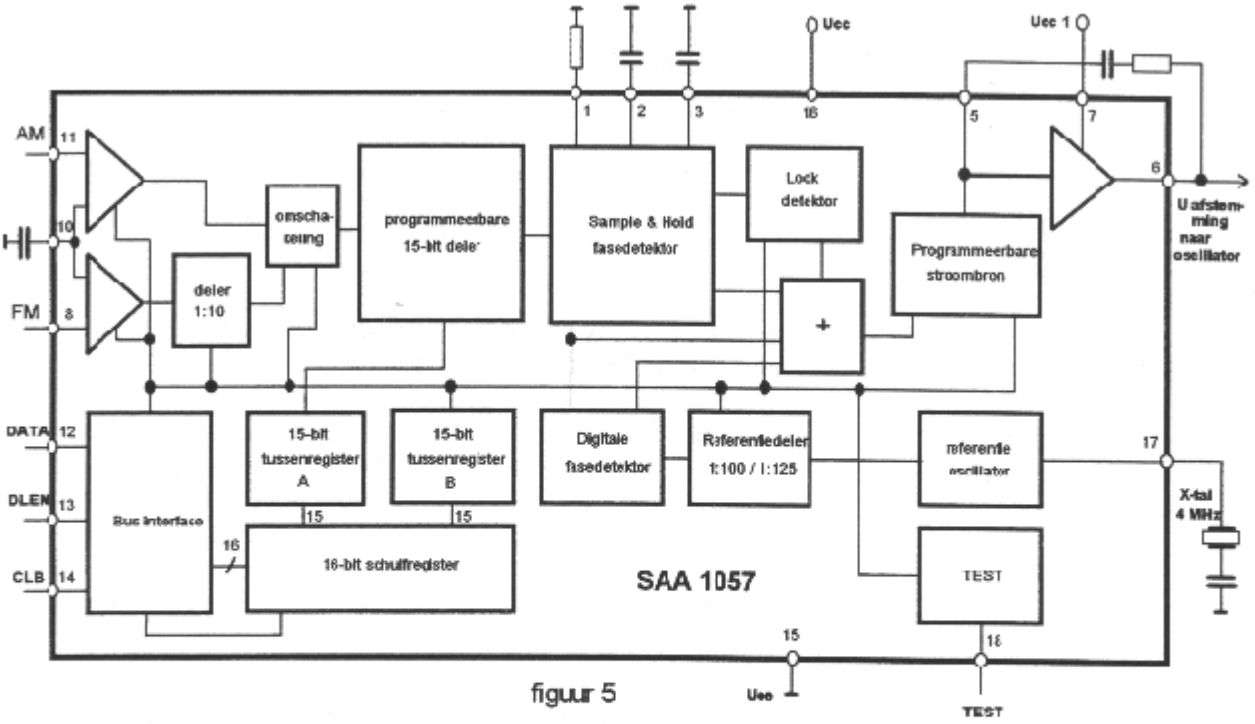
Met het commando INP 202,B wordt het woord dat op de ingangen van poort B staat ingelezen en aan de variabele B toegekend. Het commando PRINT B zet de aldus ingelezen waarde op het beeldscherm.

WAARSCHUWING: schakel niet plotseling een poort als uitgang als er nog spanningen op de aansluitingen staan (bijvoorbeeld door softwarematige poort instellingen). Ik spreek uit ervaring, de PIO zal dit niet overleven. De ingangsspanningen mogen tussen de 0 en 5 Volt liggen.

De programmeerbare synthesizer.

Er is gekozen voor het IC SAA 1057 van Philips, omdat deze een aantal voordelen boven andere synthesizer IC's heeft. Deze voordelen zijn:

1. Een stapgrootte van 1 kHz bij een lusfrequentie van 32 kHz;
2. Twee gescheiden regellussen: een digitale, die een snelle bijregeling mogelijk maakt en een analoge, die de oscillator bij kleine frequentie afwijkingen bijregelt;
3. Een groot frequentie bereik;
4. Een goede ingangsgevoeligheid (voor AM 30 mV en FM 10 mV)



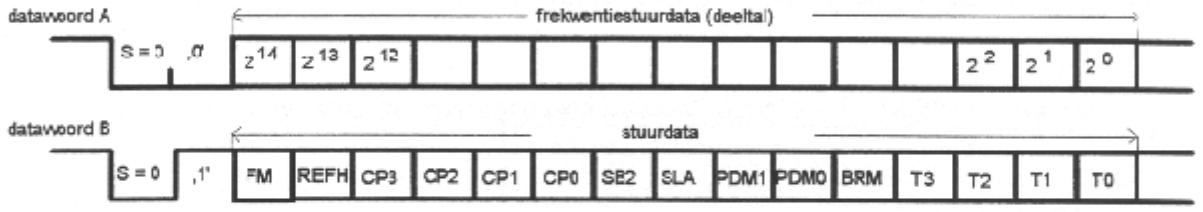
figuur 5

Figuur 5

Het bereik van de SAA 1057 ziet er als volgt uit:

bereik	stapgrootte	
	1,0 kHz	1,25 kHz
AM	512...32767 kHz	640...40958,75 kHz
FM	5,12...327,67 MHz	6,4...409,5875 MHz

Het IC wordt geprogrammeerd met stuur- en frequentiecommando's ("woorden") van elk 17 bit lang. In figuur 5 ziet u linksonder de digitale ingangsschakeling die de rest van het IC via de zogenaamde CBUS koppelt aan de drie buslijnen DLEN, DATA en CLB. De aansluiting DLEN fungeert als "portier" en laat alleen ingangssingalen door als DLEN=1. Op de DATA-lijn komt de betreffende informatie binnen. CLB is een positiefgaande puls die verzonden moet worden, terwijl de DATA-informatie geldig is (voor meer details zie onder). Het IC controleert het aantal door de computer verzonden bits en het formaat ervan. De data wordt parallel in een tussenregister overgenomen op een extra CLB-impuls (logisch 1) als tegelijkertijd DLEN=0. Voor de stuur- en frequentiecommando's zijn afzonderlijk 15-bit tussenregisters beschikbaar. Het frequentiewoord programmeert de 15-bit delerschakeling. In het midden van figuur 5 ziet u de beide fasedetectoren en rechts de uitgangsversterker plus laagdoorlaatfilter, waarvan de versterking digitaal ingesteld kan worden (zie de tabel bij stuurbits CP3...CP0).



de serieel naar het IC te verzenden data

Het IC controleert het aantal door de computer verzonden bits en het formaat ervan. De data wordt parallel in een tussenregister overgenomen op een extra CLB-impuls als tegelijkertijd DLEN=0. Voor de stuur- en frequentiecommando's zijn afzonderlijke 15-bit-tussenregisters beschikbaar. Het frequentiewoord programmeert de 15 bit-delerschakeling.

De afzonderlijke bits van een stuurcommando hebben de volgende betekenis:

stuurbit FM: met dit bit wordt omgeschakeld tussen de AM- en FM-ingang. De niet gebruikte ingangsversterker en voor zover van toepassing de ingangsdeler 1:10 worden uitgeschakeld (FM=0: AM; FM=1: FM=1).

stuurbit REFH: dit bit bepaalt de deelverhouding van de referentiedeler. Bij een kristalfrequentie van 4 MHz krijgen we bij REFH=0 in het AM bereik een referentie-frequentie van 32 kHz en een stapgrootte van 1 kHz. Bij REFH=1 is dit 40 kHz resp. 1,25 kHz. Bij FM liggen deze waarden een faktor 10 hoger.

stuurbits CP3...CP0: deze bits bepalen de versterkingsfactor van de programmeerbare stroomversterker en daarmee de lusversterking van de synthesizer (zie de tabel).

CP3	CP2	CP1	CP0	stroom (µA)
0	0	0	0	0,023
0	0	0	1	0,07
0	0	1	1	0,23
0	1	1	1	0,7
1	1	1	1	2,3

stuurbit SB2: met dit bit delen we de SAA 1057 mee of met de er na volgende 8 bits rekening moet worden gehouden (SB2=1) of dat deze moet worden genegeerd (als 0 beschouwd). Omdat het dataformaat moet blijven kloppen, moeten deze bits wel mee gezonden worden, ook al worden ze verder genegeerd.

stuurbit SLA: dit bit bepaalt hoe de frequentiedata uit het ingangs-schuifregister in de programmeerbare deler moet worden geladen. Bij SLA=1 gebeurt dit synchroon met de klokfrequentie.

stuurbits PDM1, PDM0: zoals eerder opgemerkt, beschikt het IC over een digitale en een analoge fasedetector. De digitale detector laat het synthesizer snel "locken". De analoge detector regelt daarna restfoutjes weg (zie tabel).

PDM1	PDM0	digitale fasedetektor
0	0	automatisch aan/uit
0	1	automatisch aan/uit
1	0	konstant aan
1	1	konstant uit

In het vierde geval (beide bits 1) werk dus alleen de analoge fasedetector. De inregeltijd wordt hierdoor langer, maar de faseruis is tegelijk een stuk minder.

stuurbit BRM: hiermee wordt de werking van de bus-interface van het IC ingesteld. Met BRM=0 is deze voortdurend ingeschakeld. Bij BRM=1 wordt deze na elke datatransmissie uitgeschakeld om het stroomverbruik te verminderen.

stuurbits T3..T0: hiermee bepalen we welk testsignaal er op de Test uitgang staat (zie tabel). Een voorbeeld om een "locked" led te laten branden vindt u in het bijvoegsel.

T3	T2	T1	T0	testsignaal
0	0	0	0	geen testsignaal
0	1	0	0	uitgang referentiedeler (32 resp. 40 kHz)
0	0	0	1	uitgang programmeerbare 15-bit-deler
0	1	0	1	lock-indikatie: 1=locked

stuurwoord

1	0	0	0	0	0	1	1	1	0	0	1	0	1	0	1
KB	FM	REF	CP3	CP2	CP1	CP0	SB2	SLA	PDM0/1		BRM	T3	T2	T1	T0

frekwentiewcord

0	0	0	0	0	1	0	1	1	1	0	0	0	1	0	1
KB											16	8	4	2	1

voorbeeld van een stuurkommando en frekwentiedata

Schema synthesizer

Zoals het schema van de synthesizer laat zien is er niets ingewikkeld aan. De spanning op pin 7 (V ref.) zou eigenlijk 30 Volt moeten zijn. Deze 30 Volt wordt gebruikt om de afstemming van de aangesloten oscillator een zo groot mogelijk bereik te geven. Sluit u in plaats van 30 Volt hier een spanning van 12 Volt aan, dan wordt het afstembereik verkleind. De DATA, CLB en DLEN aansluitingen verbinden het synthesizer-IC met de PIO. De aansluitpennen die u gebruikt zijn afhankelijk van de in regel 40 van de software bepaalde parameters (in de voorbeeld software zijn dat pin 1 (DATA), 2 (DLEN) en 3 (CLB) van de 25-polige D-connector.

De bouw van de synthesizer.

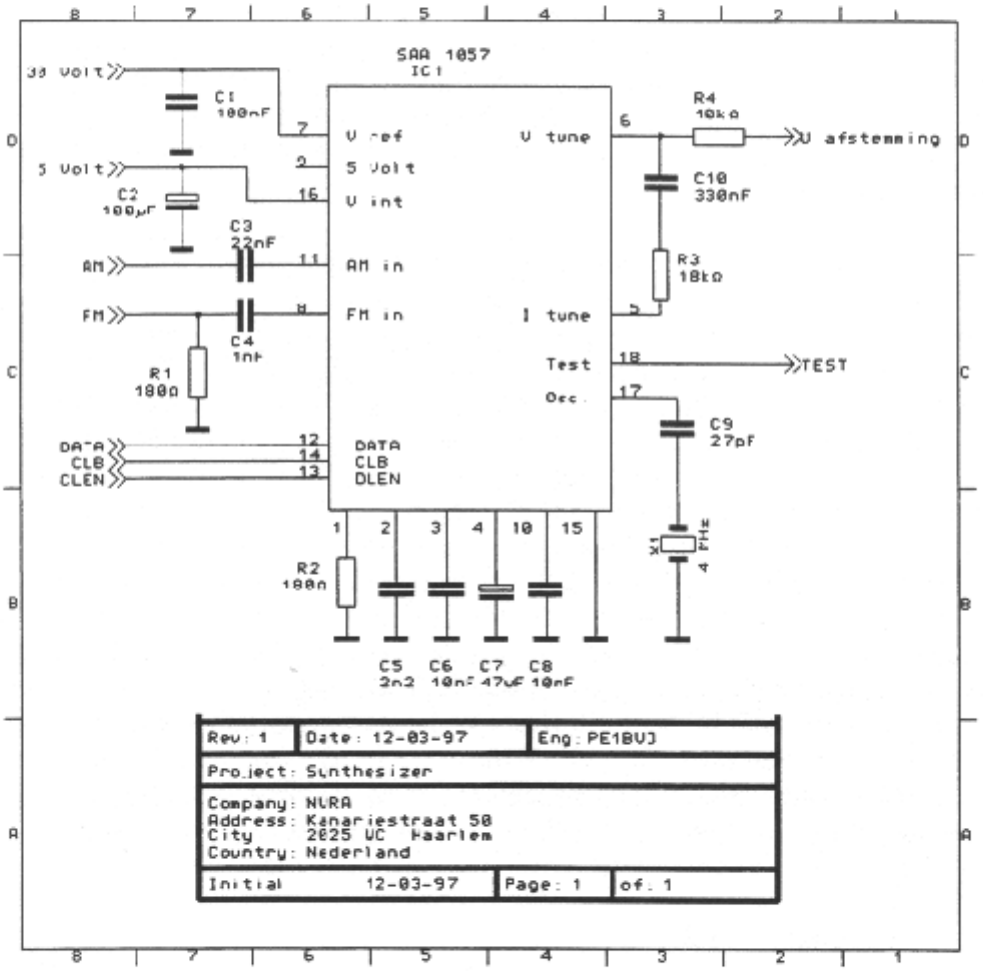
De print van de synthesizer is zodanig opgebouwd dat deze voor zowel de frequenties 512 kHz tot 40958,75 kHz (AM ingang) als voor 5,12 MHz tot 409,5875 MHz (FM ingang) te gebruiken is. In tegenstelling tot de PIO is de synthesizer op een enkelzijdige print gebouwd. Ook voor de synthesizer geldt dat, door de geringe hoeveelheid onderdelen, de bouw geen problemen zal opleveren. Het enige waar u op dient te letten is een draadbrug die gelegd dient te worden tussen pin 13 van IC1 en pin 1 van de 25-polige D-connector. Het geheel kan in een klein kastje gebouwd worden. U kunt daarbij zowel de AM- als FM-ingang voorzien van bijvoorbeeld een BNC chassisdeel. U kunt zelf bepalen of u een klein voedinkje wilt inbouwen of dat u gebruik wilt maken van een zgn. netsteker voeding. De voeding moet minimaal 12 Volt en 200 mA kunnen leveren. Zoals eerder beschreven bereikt u het grootste afstembereik, door op pin 7 van de SAA1057 30 Volt aan te sluiten. Bij mijn experimenten met de 135 MHz oscillator kon ik deze met gebruikmaking van 30 Volt ongeveer 50 MHz verstemmen. Bij gebruikmaking van 12 Volt was dit ongeveer 30 MHz. Belangrijk bleek de invloed van rimpelspanning op de voeding van de synthesizer. Bij een rimpelspanning van 50 mV was de regeling van de synthesizer soms instabiel. Nadat ik de rimpelspanning terug gebracht had tot 5 mV was dit effect verdwenen.

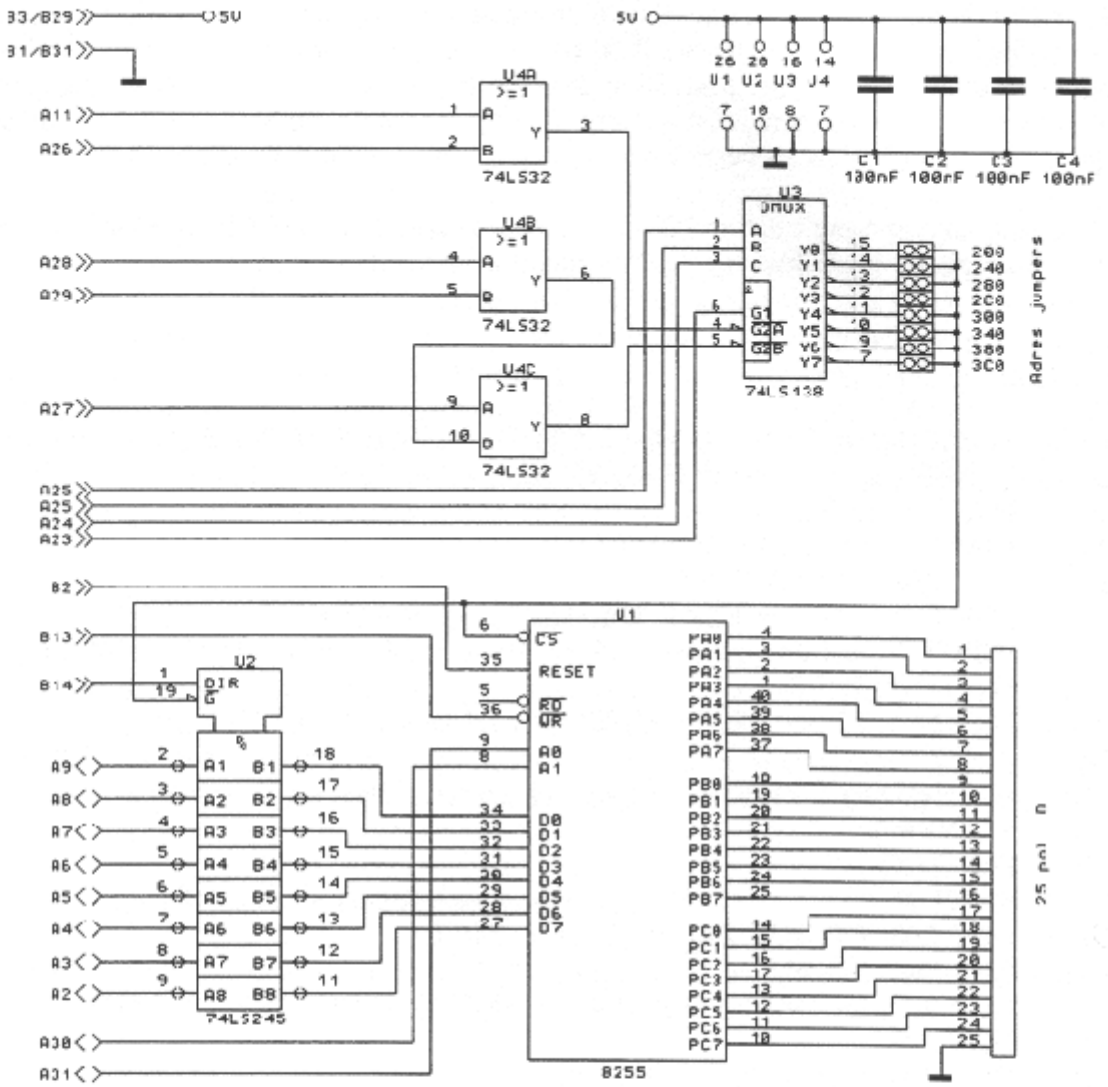
De bouw van de PIO.

Daar de PIO is opgebouwd met slechts een 4-tal IC's, 4 ontkoppel C's en een 25-polige D-connector zal het bouwen ervan geen problemen op hoeven te leveren. De PIO is opgebouwd op een dubbelzijdige print. Er zijn een aantal doorverbindingen (via's) aanwezig die als eerste met een zo dun mogelijk boortje geboord dienen te worden. Boor de rest van de gaten met een 0,8 mm boortje. Verbindt de doorverbindingen met een dun stukje draad door

naar beide printzijden en soldeer deze vast. Daarna kunnen de onderdelen op de print gezet worden. Plaats de IC's **NIET** in IC-voeten, er dienen nl. een aantal pootjes van IC's ook aan de componentzijde gesoldeerd te kunnen worden. Dit geldt ook voor de ontkoppel C's die op een aantal plaatsen aan weerszijden van de print gesoldeerd dienen te worden. Plaats als laatste de PC-I/O beugel op de 25-polige D-connector.

LET OP: Het grote kopervlak aan de componentzijde is het +5 Volt vlak. Het grote kopervlak aan de soldeerzijde is het massavlak. Klem de print dus niet vast met een tangetje waarvan de bekken geleidend zijn, als er spanning op de print staat.





Aansluitingen :8m AT slot

A1	= I/O CH CK	B1	= Ground
A2	= Data 7	B2	= Reset Drv
A3	= Data 6	B3	= +5 Volt
A4	= Data 5	B4	= IRQ 2
A5	= Data 4	B5	= -5 Volt
A6	= Data 3	B6	= DRQ 2
A7	= Data 2	B7	= -12 Volt
A8	= Data 1	B8	= Reserved /(inverted !)
A9	= Data 0	B9	= +12 Volt /(inverted !)
A10	= I/O CH RDY	B10	= Ground
A11	= AEN	B11	= MEMW (inverted !)
A12	= Adres 19	B12	= MEMR (inverted !)
A13	= Adres 18	B13	= IOW (inverted !)
A14	= Adres 17	B14	= IOR (inverted !)
A15	= Adres 16	B15	= DACK 3 (inverted !)
A16	= Adres 15	B16	= DRQ 3
A17	= Adres 14	B17	= DACK 1 (inverted !)
A18	= Adres 13	B18	= DRQ 1
A19	= Adres 12	B19	= DACK 0 (inverted !)
A20	= Adres 11	B20	= Clock
A21	= Adres 10	B21	= IRQ 7
A22	= Adres 9	B22	= IRQ 6
A23	= Adres 8	B23	= IRQ 5
A24	= Adres 7	B24	= IRQ 4
A25	= Adres 6	B25	= IRQ 3
A26	= Adres 5	B26	= DACK 2 (inverted !)
A27	= Adres 4	B27	= T/C
A28	= Adres 3	B28	= ALE
A29	= Adres 2	B29	= +5 Volt
A30	= Adres 1	B30	= Osc
A31	= Adres 0	B31	= Ground

Rev: 1	Date: 12-03-97	Eng: PE1BUJ
Project: PIO		
Company: NVRA Address: Kanariestraat 50 City 2025 VC Haarlem Country: Nederland		
Initial 12-03-97	Page: 1	of: 1

De software.

Zoals ik al schreef, is mijn kennis van microcontrollers en het programmeren niet zo groot. Toch heb ik aan de hand van de door de leveranciers aangeleverde datasheets en testsoftware de synthesizer werkend gekregen. Wellicht dat een beetje programmeur zich een kriek lacht om de onderstaande listing. Naar mijn mening kan de listing veel korter om het zelfde doel te bereiken. Ik nodig een ieder uit om vrijelijk te experimenteren met de software, deze (grafisch) te verfraaien en beschikbaar te stellen aan andere gebruikers.

De listing gaat er vanuit dat de PIO het adres 200 gebruikt. De jumper op de PIO-print dient dus op de plaats aangeduid met 200 geplaatst te zijn.

listing 1.2

```
10    REM * PROGRAMMA VOOR DE SYNTHESIZR MET DE SAA 1057 *
20    SCREEN 0, 0, 0: WIDTH 80, 25: KEY OFF: CLS
30    PORT = &H200
40    DIM M1(16): DIM M2(16): DL=4: C=2
50    OUT 203, 139: OUT 200, 0
60    FOR N = 1 TO 16: READ DA: M1(N) = DA: NEXT N
70    INPUT " Geef de gewenste frekwentie op in kHz (bijv. 145000 (=145 MHz)) "; FE
80    FO=FE+0: F=FO
90    FOR N=1 TO 16: REM OMZETTEN IN BINAIR GETAL
100   X=F/2
110   IF 2*INT(X) = F THEN M2(N) = 0: GOTO 130
120   M2(N) = 1
130   F=INT(X)
140   NEXT N
150   OUT 200, DL:OUT 200,DL+C:OUT 200, DL
160   FOR N=1 TO 16
170   OUT 200, DL+M1(N):OUT 200, DL+M1(N)+C:OUT 200, DL+M1(N)
180   NEXT N
190   OUT 200, 0:OUT 200,C:OUT 200,0
200   OUT 200,DL:OUT 200, DL+C:OUT 200,DL
210   FOR  N=16 TO 1 STEP -1
220   OUT 200, DL+M2(N):OUT 200,DL+M2(N)+C: OUT 200,DL+M2(N)
230   NEXT N
240   OUT 200,0:OUT200,C:OUT 200,0
250   GOTO 70
260   DATA 1,0,0: REM S, FM, REFH
270   DATA 0,0,0,1: REM CP3...CP0
280   DATA 1,1,0,0,1: REM SB2, SLA, PDM1, PDM0, BRM
290   DATA 0,1,0,1: REM T3...T0
```

In regel 30 wordt het adres van de PIO vastgelegd. In regel 40 worden twee matrices (M1 en M2) gedefinieerd. Met DL=4 en C=2 worden de binaire waarden van de ingangslijnen DLEN en CLB overeenkomstig de hardware vastgelegd. In regel 50 wordt de PIO geïnitialiseerd, poort A wordt als uitgang geschakeld en de poorten B en C als ingangen. In regel 80 kunt u het getal 0 desgewenst door de middenfrequentie vervangen (opgeven in kHz) zodat u uw oscillator als local-oscillator voor uw (zend)ontvanger kunt gebruiken. Bij ondermenging dient u tevens de + in een - te veranderen.

De opgegeven frequentie moet dan de ontvangstfrequentie zijn. De 16-voudige lus in regel 90 berekent het binaire getal door herhaalde deling door 2 en zet het resultaat in matrix M2(N). De oscillatorfrequentie wordt steeds door twee gedeeld. Blijft er na elke deling een rest over, dan wordt M2(N)=1 gemaakt, anders 0. De bits die nu in M2(N) staan, worden in omgekeerde volgorde (M2(16...M2(1) naar het synthesizer IC gestuurd. In regel 260 geeft het tweede getal (hier 0) aan welke ingang van de synthesizer gebruikt wordt (hier de AM-ingang). Wijzigt u de 0 in een 1, dan kunt u de FM ingang gebruiken. Het derde getal in regel 260 (hier 0) geeft aan welke deelverhouding van de referentiedeler wordt gebruikt die de stapgrootte bepaalt. In dit voorbeeld levert dit een stapgrootte op van 1 kHz. Maakt u van het derde getal in regel 260 een 1, dan wordt de stapgrootte 1,25 kHz. Bij gebruik van de FM ingang, liggen deze waarden een faktor 10 hoger (zie de tekst: stuurbit FM en stuurbit REFH). We hebben nu de data in de velden M1 en M2 serieel ter beschikking en kunnen deze naar de synthesizer sturen. Als we nog even terugbladeren en kijken naar de frequentie- en stuurdata, zien we dat elk met een startbit en een eigen identificatiebit beginnen. Nadat een 0 is verzonden (alle drie lijnen 0) maken we DLEN=1 met de instructie OUT 200,DL. Daarop volgt de CLB-puls met de instructie OUT 200,DL+C. Ter afsluiting van die puls volgt OUT 200,DL (zie regel 150). Met het commando GOTO 70 in regel 250 springt het programma terug en wacht op de invoer van een nieuwe frequentie.

Inbouw van de PIO in de computer.

Voordat u aan de inbouw van de PIO begint dient u de computer spanningloos te stellen en zich zelf goed te "aarden". Dit om te voorkomen dat door statische elektriciteit de computer en/of de PIO overlijdt. De PIO dient geplaatst te worden in een vrij slot in de computer. Druk de PIO voorzichtig, maar stevig, in het slot en schroef de PC-I/O beugel vast.

De PIO met de synthesizer verbinden.

De PIO kan met een standaard printerkabel (25-polig) aan de synthesizer verbonden worden.

Dankwoord.

Een speciale dank voor hun steun aan mijn experiment gaat uit naar de volgende bedrijven en personen.

ASWO onderdelen Hengelo, voor het (gratis) leveren van de onderdelen en (uitgebreide) datasheets.
Henk Wiedijk, voor de software, de beschrijving en ondersteuning.
Hans van Waardenburg (Philips Eindhoven), voor de laatste updates en datasheets van de SAA 1057.