

RIJKSUNIVERSITEIT GENT
Faculteit van Letteren en Wijsbegeerte
Akademiejaar 1992-1993

Een Onzichtbaar Muziekinstrument

Boekdeel 3
Hoofdstuk 3
Hoofdstuk 4



Proefschrift ingediend tot het behalen van
de graad van Doktor in de
Kunstgeschiedenis en Oudheidkunde,
richting Muzikologie,

door Godfried-Willem RAES

Promotor: Prof.Dr.Ferdinand DE HEN

BOEKDEEL 3**HOOFDSTUK 3****Een digitale implementatie: "A Book of moves"**

3.1.- Algemeen opzet	3- p.1
3.1.1. - Hardware versus software	
3.1.2 - De signaalverwerking	p.2
3.1.3. - Data extraktie & data acquisition	p.4
3.1.3.1. - De digitale weg	p.4
3.1.3.2. - Analoge komputer voor data-extraktie	p.8
3.2.3.2.1. - bewegingshoeveelheidsinformatie	p.8
3.2.3.2.2. - bewegingssnelheidsinformatie	p.12
3.1.4. - Gegevensverwerking: multitasking dedicated computer	p.14
3.1.5. - Midi-interface	p.17
3.1.6. - Evaluatie en blokschema van het gehele instrument	p.18
3.2.- Software bespreking	
3.2.1.1.- mapping	p.21
3.2.1.2.- multitasking Basic (MT-Basic)	p.21
3.2.2.- Kode:	
3.2.2.1.- deklaratie van de variabelen	p.23
3.2.2.2.- initializatie	p.25
3.2.2.3.- definitie van de midi-functie	p.26
3.2.2.4.- initializatie midi & LCD-scherm	p.27
3.2.2.5.- definitie van de synthesizerfuncties	p.28
3.2.2.6.- kommando voor tijd en ritme	p.29
3.2.2.7.- test van de opstelling	p.29
3.2.2.8.- Hoofdmenu-keuze: makrokompositie	p.30
3.2.2.9.- De modules van 'A Book of Moves'	p.33
3.2.2.9.1. - Open	p.33
3.2.2.9.2. - Topoi	p.36
3.2.2.9.3. - Minor	p.41
3.2.2.9.4. - Beat	p.43
3.2.2.9.5. - Rising	p.45
3.2.2.9.6. - Sforte	p.48
3.2.2.9.7. - Lead	p.49
3.2.2.9.8. - Canvas	p.51
3.2.2.9.9. - Close	p.53
3.2.2.9.10.- Prime-Time	p.55
3.2.2.9.11.- Call	p.56
3.2.2.9.12.- Spooky	p.58
3.2.2.9.13.- Rec-Play	p.59
3.2.2.9.14.- Lock-Unlock	p.61
3.2.2.9.15.- Hammers	p.63
3.2.2.10.- De taken	p.64
3.2.2.10.1 - Taak 1	p.64
3.2.2.10.2 - Taak 2	p.65
3.2.2.10.3 - Taak 3	p.66
3.2.2.10.4 - Taak 4	p.67
3.2.2.10.5 - Taken 5-11	p.68
3.2.2.10.6 - Taken 12-17	p.69
3.2.2.10.7 - Taak 18	p.70
3.2.2.11.- Opzoekingstabellen en data	p.71
3.2.2.12.- Subroutines	p.71
3.2.3.: Memory Map	p.72
3.3.- Artistieke bespreking van "A Book of Moves"	p.73
	-p.78

HOOFDSTUK 4 :**Perspektieven en verdere onderzoeksmogelijkheden:**

4.1.- Experimenteel onderzoek: een verdere wegverkenning	4 - p.1
4.1.1.: - Digitale signaalverwerking	p.1
4.1.2.: - Verhoging van de draaggolffrequentie	p.2
4.1.3.: - Mikrogolfsysteem	p.3
4.1.4.: - Parallel-computing & Neurale netwerken	p.4
4.1.5.: - Analoge signaalverwerkingsmogelijkheden	p.5
4.2.- Het muzikale naakt	p.8
4.2.1.: figuurlijk	p.8
4.2.2.: letterlijk	p.9
4.3.- De automatizering van de muziek	p.11
4.3.1.- interpretatie zonder techniek	p.12
4.3.1.1.- Kodering	p.14
4.3.1.2.- Verklanking	p.16
4.3.1.2.1.: - elektronische instrumenten	p.16
4.3.1.2.2.: - orgel	p.18
4.3.1.2.3.: - klavichord en klavecimbel	p.19
4.3.1.2.4.: - piano	p.19
4.3.1.2.5.: - slaginstrumenten	p.22
4.3.1.2.6.: - blaasinstrumenten	p.23
4.3.1.2.7.: - strijkinstrumenten	p.24
4.3.1.2.8.: - tokkelinstrumenten	p.24
4.3.1.2.9.: - nieuwe akoestische instrumenten	p.24
4.3.2.- De uitvoerder als interpreet	p.26
4.3.3.- Improvisatie versus kompositie	p.28
4.4.- Een toekomst voor de organologie?	p.30

HOOFDSTUK 3:

Een digitale implementatie:

"A Book of Moves"

3.1: Algemeen opzet

3.1.1: Hardware versus Software

Na de ervaringen met de laatste analoge versies van 'Holosound' en de mislukking van de eerste poging tot digitalisering van het onzichtbare muziekinstrument, beslisten we de boeg helemaal om te gooien en de richting van de informatisering van het projekt in te gaan.

Vanuit elektrotechnisch standpunt gezien komt het hele probleem voor de bouw van ons instrument dan neer op een klassiek probleem van signaalverwerking, waarbij we dan volgende stappen kunnen onderscheiden:

1.- signaal verwerving

Dit is het nivo van de gebruikte transducers. De eerste omzetting dus van het fenomeen waarover we gegevens in elektrische vorm willen (beweging en motoriek) naar die elektrische vorm.

2.- signaal verwerking

Dit is de techniek nodig om het signaal in een bruikbare vorm te krijgen. Deze stap omvat dus in eerste plaats de versterking, linearisering van de deficiënties van de gebruikte transducer en alle stappen nodig om een optimale signaal-ruisverhouding te ^{bc} komen.

3.- data-extractie en data-verwerving

Dit is het geheel van technieken die we dienen toe te passen om uit het signaal informatie (data) af te leiden. In het Engelse vakjargon heet dit 'data-acquisition'.

4.- data verwerking

Dit is het geheel van de technieken van algoritmische aard die we dienen toe te passen om op grond van de verworven informatie, nieuwe informatie te genereren in de meestal meer gebalde vorm waarin we haar uiteindelijk nodig hebben.

5.- data output

Dit is het interface dat ervoor zorgt dat de informatie in een technisch geschikt formaat wordt omgezet dat door de aan te sluiten apparatuur kan worden begrepen.

De eerste stap was en is gezet: we beschikken over de nodige transducers en bruikbare relevante signalen.

De tweede stap kon ook als in zekere mate gezet beschouwd worden. Daartoe behoort immers het gehele opzet van de analoge komputer zoals we die in het vorige hoofdstuk uitvoerig hebben beschreven. De vraag of het signaal reeds in voldoende mate verwerkt was om te kunnen worden ingepast in het nieuwe opzet diende uiteraard te worden gesteld. Immers aan een signaal dat uiteindelijk voor rechtstreekse beluistering was bedoeld moeten andere eisen worden gesteld dan aan een signaal dat voor verdere verwerking wordt gebruikt en dus eigenlijk binnen de machine opgesloten blijft.

De laatste stap was in feite ook reeds gezet. We wisten immers dat de output het MIDI-protokol zou dienen te volgen, en

een universeel bruikbaar komputer MIDI-interface ontwierpen we reeds in 1984, kort na de publikatie van de officiële technische MIDI-specificaties (1983). Het was een interface dat op eender welke 8-bit Centronics printerpoort van eender welke komputer kon aangesloten worden. De besturing geschiedde op precies dezelfde wijze als die voor een standaard printer. We hebben er geen ogenblik aan getwijfeld dat deze stap in digitale hardware gerealiseerd diende te worden, ondanks het feit dat het ook via software wel haalbaar en mogelijk is.

Volledigheidshalve heb ik het opbouwschema van dit interface -waarvan terloops gezegd zo'n honderd exemplaren in omloop zijn onder diverse musici/komponisten over de gehele wereld evenals onder studenten en oud-studenten van mijn cursus experimentele muziek aan het Konservatorium, verder in dit hoofdstuk toegevoegd. Het interface kent nogal wat succes omdat het uitermate eenvoudig te programmeren is, wat niet gezegd kan worden van het door Roland voor de IBM-PC onder de naam MPU401 op de markt gebrachte onding. Ons ontwerp is niet veel meer dan een toepassing van de 6402- UART chip van Intersil.

Voor alle andere stappen (2 , 3 en 4) was een van de grootste dilemma's waarvoor we ons geplaatst zagen, de keuze tussen hard-ware en software implementaties enerzijds, en binnen de hardware-implementaties, tussen analoog en digitaal. Het is immers zo dat we voor heel wat problemen die te maken hebben met signaalverwerking en data-extractie, de techniek de vrije keuze toelaat tussen deze drie technische aanpakmogelijkheden. Met de ons vandaag ter beschikking staande technologie kunnen immers zowat alle signaalverwerkingsproblemen die geen al te strenge eisen stellen inzake snelheid, zowel via hardware als via software aangepakt en opgelost worden. Wanneer echter uiterste eisen aan snelheid en precisie gesteld worden, zal men steeds van hardware oplossingen gebruik maken. Om dit duidelijk te maken, wilden we eerst per stap alle mogelijkheden onderling vergelijken en een vergelijkende kosten en baten analyse ervan maken.

3.1.2: De signaalverwerking

Hier stonden we voor de keuze onze analoge komputer te behouden, ofwel te vervangen door een geheel digitale implementatie in hardware of software.

De signalen zoals die, na voorversterking, beschikbaar zijn als input zijn echter betrekkelijk hoogfrequent. (35kHz tot 220kHz). Willen we hierop bewerkingen uitvoeren via digitale technieken dan is eerst en vooral een omzetting naar een digitaal formaat onontbeerlijk. Precies hier zit ook de moeilijkheid: immers, om signalen in het hier aangegeven frequentiegebied om te zetten, dienen we hen te bemonsteren met een frequentie die minstens het dubbele bedraagt van de hoogste frequentiecomponenten in het om te zetten signaal. Dit is immers de konsekwentie van het beroemde Nyquist theorema. Veronderstellend dat we zouden afzien van alle harmonischen van ons signaal, dan zouden we dus een sampling rate van ca. 440.000 samples per sekonde moeten kunnen realizeren. Dit is ongeveer het tienvoudige van wat voor koerante muzikale toepassingen in digitale audio gangbaar is.

Hiertegen zou men kunnen opwerpen dat we -gezien de eerdere meetgegevens inzake resolutie- de omzetting zouden kunnen

beperken tot 12, 10 of zelfs 8 bits in plaats van de gebruikelijke 16-bits voor digitale audio, maar zelfs in dit geval vallen de noodzakelijke specificaties van de nodige ADC's buiten datgene wat vandaag koerant is.

Lange tijd heb ik gewerkt aan een ontwerp waarbij de TMS320C25 digitale signaal processor (DSP) van Texas zou gebruikt worden. De Amerikaanse komponist Larry Wendt had me enige tijd tevoren (1987) enkele proefexemplaren van de mikroprocessor en de volledige dokumentatie van Texas Instruments kado gedaan, zodat ik dus reeds een absoluut minimum aan ervaring had op dit vlak. Hij werkte op dat ogenblik aan een DSP-board voor de Studio voor Elektronische muziek aan de *University of California at Berkeley*, campus San José, onder leiding van Allen Strange, met de bedoeling er een real-time algoritmische soundprocessor mee op te bouwen.

Maar toen ik ook dokumentatie verkreeg met betrekking tot de reeds beschikbare toepassingen met deze DSP-chip werkend op een insteekkaart in een IBM-AT compatibele komputer (de Model 25 Digital Signal Processor card, geproduceerd door Dalance Spry, Rochester, NY) en de juiste specificaties ervan, sloeg de angst me om het hart. Ik voelde dat ik er wel niet in zou slagen datgene wat de industrie als top-produkten op technologisch vlak aanbood, zo maar eventjes te overtreffen... Dit DSP-board beschikt over 1 ADC kanaal, 12 bit en met een maximale sampling rate van 110kHz. Multiplexing was mogelijk, maar dan werd uiteraard de sampling rate gedeeld door het aantal kanalen. Ik had minstens drie kanalen nodig, aangezien er in de Holosound opstelling 3 ultrasoontransducers als ontvangers geschakeld zijn. Dit zou de maximale sampling rate dus hebben teruggebracht tot 36kHz, of, veel te laag om te voldoen aan de eisen van het Nyquist theorema. Bovendien, werd dit board aangeboden voor programmering uitsluitend in TMS320C25 assembler. Routines voor de noodzakelijke Fourier transformaties (256 punten) zou ik weliswaar kunnen overnemen uit de standaard applicatiebibliotheek, maar ook dan bleef het een weinig bemoedigend perspektief zonder veel uitzicht...

Cfr.: Texas Instruments, "Digital Signal Processing Applications - TMS320 Family" , 1986, "TMS32010 & TMS32020 User's Guide" , "TMS32020 Assembly Language Programmer's Guide".

Heel devoot en nederig besloot ik de reeds gerealiseerde print en de bijhorende DSP- en periferie chips terzijde te schuiven en zocht een andere uitweg, of beter, ik besloot het opzet van de analoge komputer in principe te behouden.

Later pas ben ik -kwestie van toch niet het gevoel te hebben een nederlaag te hebben geleden- gaan narekenen welke 'winst' er te behalen zou zijn door gebruik te maken van zuiver digitale technieken in deze fase van het ontwerp, en toen bleek er eigenlijk helemaal geen winst aan vast te zitten: het oplossend vermogen van de DSP-processor -dan wel uitgerust met een 'anno 1992' snelle ADC-converter- rekening houdend met dit van het signaal waarvan we uitgaan, bleek helemaal niet groter uit te vallen dan de hooguit 7 á 8 bits waartoe de analoge komputer sowieso ook al in staat was.

3.1.3: Data-extractie en data-acquisition

3.1.3.1.: -De digitale weg

Het probleem om uit de -ingevolge onze vorige beslissing- analoge en laagfrequentie signalen voortgebracht door de analoge komputer een maximum aan relevante informatie af te zonderen, stelde ons eens te meer voor de keuze analoog of digitaal. Hier was de vraag echter een heel stuk realistischer dan bij de vorige stap. De bandbreedte van de signalen zoals die de analoge komputer verlaten valt immers in het gebied 0 - 1000Hz, wat de eisen inzake sampling rate aanzienlijk gemakkelijker maakt. Een sampling rate van 2kHz is immers ruim voldoende, en iets waaraan gemakkelijk wordt voldaan door eenvoudige en betrekkelijk goedkope 8-10 bits ADC converter chips in de handel.

Vol goede moed, en met de vastberadenheid het ditmaal wel waar te maken, bouwden we een 3 kanaals ADC konverter, grotendeels volgens de lijnen van wat we in het vorige hoofdstuk beschreven sub 2.5., die we echter dit keer rechtstreeks aansloten op de bus van een van onze IBM-AT klonen.

We gebruikten 3 afzonderlijke en goedkope 8-bit ADC-converter chips (type ZN427 van Ferranti) volgens een schakeling grotendeels overgenomen uit de application notes van Tekelec Airtronic. Hiermee is een bemonsteringsfrequentie van minstens 10kHz haalbaar. Ruimschoot voldoende voor ons doel dus.

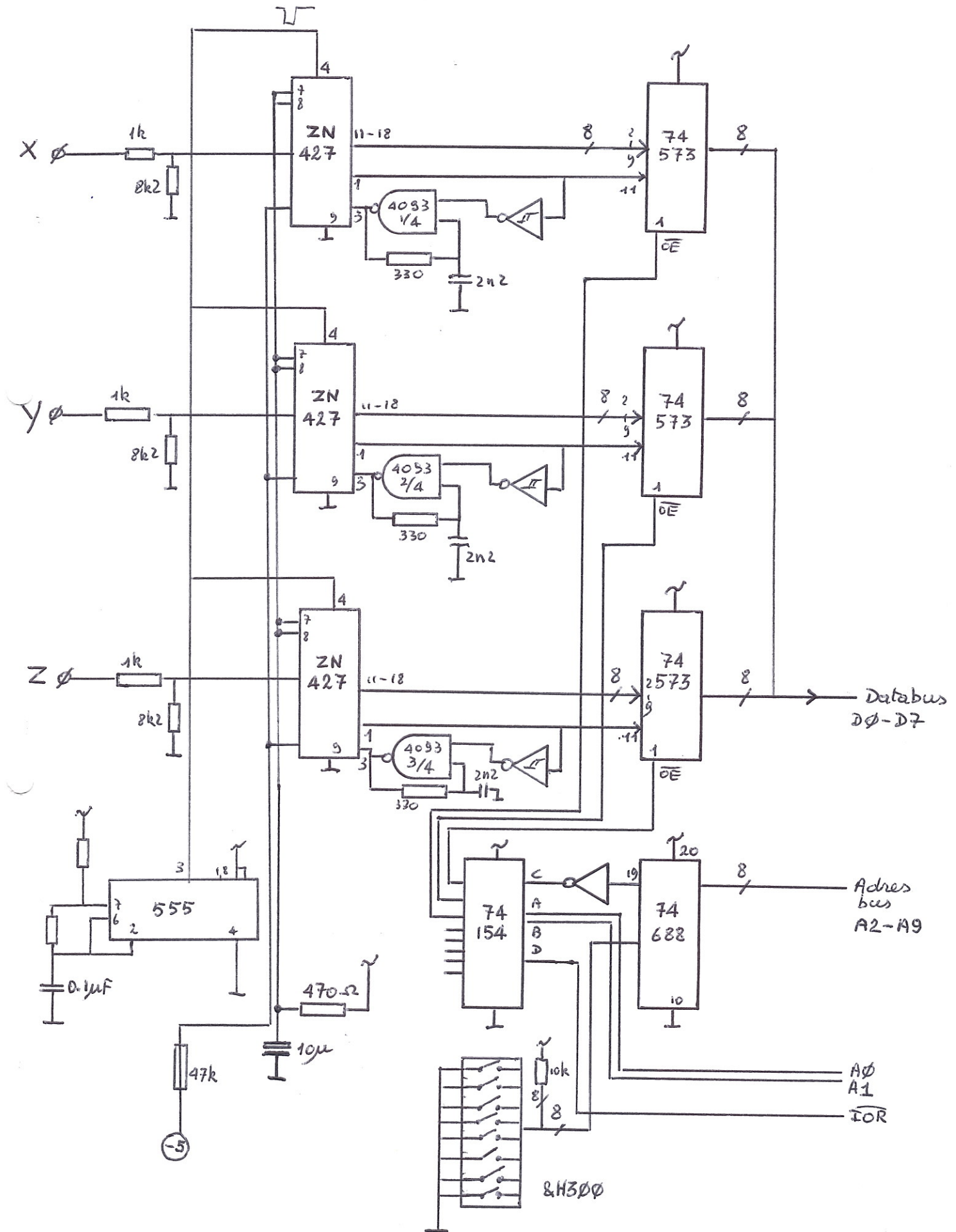
Vandaag (01/1993) zou het beter zijn de UVC3130 8-bit video flash omzetterchip te gebruiken, maar die was begin 1992 nog niet vlot beschikbaar op de onderdelenmarkt. Deze chip kan signalen bemonsteren tot 15Mhz en zou dus ook kunnen worden toegepast voor de vorige stap in combinatie dan met een klassieke 80486 50MHz processor PC... Bovendien is van deze ADC zelfs een type met een resolutie van 10bits verkrijgbaar!

Voor een typische toepassing, zie G.PELTZ, "AD-DA-omzetters", 1992, p. 666-, en p.90 e.v.

Het bijzondere aan ons ontwerp is dat de ADC-chips vrijlopend gebruikt worden en hun meest recente data-byte dumpen in een latch (de 74HCT573 chips). De komputer heeft dus steeds de meest recente gegevens onmiddellijk voor verwerking ter beschikking, zonder dat hij specifieke konversie-instructies dient te geven, en dan wachten op het konversierezultaat. We ontwierpen de schakeling voor aansluiting op de bus van de IBM-PC. Een technische verbetering zou erin kunnen bestaan hier in plaats van latches, zgn. dual-ported RAM-chips te gebruiken.

De uitlezing van de X,Y,Z signalen kan heel eenvoudig gebeuren als volgt:

```
X% = INP(&H300)
Y% = INP(&H301)
Z% = INP(&H302)
```



Na opbouw van deze eenvoudige ADC schakeling beschikten we over de mogelijkheid onze transducersignalen rechtstreeks als een reeks samples in de komputer in te lezen.

De amplitude van het afgeleverde signaal is een maat voor de hoeveelheid bewegend oppervlak van de speler. Dus wilden we deze amplitude uit het signaal kunnen afleiden. Deze kan van elk kanaal eenvoudig worden bepaald wanneer we N monsters van dat kanaal nemen en meten, en daarvan dan het gemiddelde berekenen. Immers in momentane waarden van de golfvorm van het signaal zijn we wat deze parameter aangaat niet geïnteresseerd. De amplitude van een signaal is wiskundig gezien equivalent met de oppervlakte van haar curve. In hardware termen, is het equivalent met een afgevlakte gelijkrichting, waarbij de integratietijd (RC-tijd) evenredig is met de waarde van de afvlakkondensator.

Ter wille van de leesbaarheid van de code doen we hier alsof het 8-bit ingangssignaal uitgelezen wordt als een numerieke waarde met de 0-as als symmetrielij, anders gezegd als decimale waarden tussen -127 en +127. In werkelijkheid echter bewegen de waarden zich uitsluitend tussen 0 en 255.

In Basic (PDF of QB4.5)) ziet de pseudo-code voor een procedure voor het verkrijgen van bewegingshoeveelheid in elk van de drie vektoren er dan (sterk vereenvoudigd) uit als volgt:

Over te dragen variabelen naar deze procedure zijn:

Integratietijd: 0.01s - 0.2s

Overgedragen variabelen zijn:

X , Y , Z : Amplitudes voor de 3 vektoren.

Over te dragen (of vast te leggen) konstanten zijn:

Adr1 , Adr2 , Adr3 : de hardwarebepaalde I/O adressen van de ADC-omzetter.

SUB GetAmplitude:

DEFINT A-S

DEFDBL T, I

T = TIMER : N = 0 : X = 0 : Y = 0 : Z = 0

DO

N = N + 1

X = INP(Adr1) : Y = INP(Adr2) : Z = INP(Adr3)

' bereken de permanente som der ingangssignalen

X = X + ABS(INP(Adr1))

Y = Y + ABS(INP(Adr2))

Z = Z + ABS(INP(Adr3))

LOOP UNTIL TIMER - T >= Integratietijd

X = X / N : Y = Y / N : Z = Z / N

END SUB

Puur theoretisch gezien zou hier gebruik gemaakt moeten worden van de gebruikelijke differentiaal-vergelijking voor een RC-integrator :

$$V(t) = 1/RC \int_{-\infty}^t V_i(\tau) \cdot e^{-(t-\tau)/RC} \cdot d\tau$$

Ook wanneer deze vergelijking finitistisch wordt uitgewerkt (wat we deden in het programma 'Holosimi' gebruikt voor de curves besproken aan het eind van hoofdstuk 2, en waarvan de code aan de appendix werd toegevoegd, is de winst aan precisie niet van dien aard dat de vertraging van het programma die erdoor wordt veroorzaakt geïnteresserd kan worden. Wanneer middelen beschikbaar komen om het wiskundig korrekte algoritme gebruikt in Holosimi ook in real-time toe te passen, mag dat uiteraard niet nagelaten worden.

De volgende parameter waarvoor we gezien ons opzet uiteraard belangstelling hebben en die we ook uit het ingangssignaal wilden kunnen afleiden, was de bewegingssnelheid. Zoals we vroeger hebben aangetoond, is deze een vektoriele functie van de hoogste dopplerfrequentie aanwezig in het signaal. Hier nu kwam de aap uit de mouw en stootten we op -zoals uiteindelijk bleek- onoverkomelijke wiskundige, theoretische én praktische problemen...

Intuitief gezien lijkt het op het eerste gezicht erg eenvoudig te zijn uit een signaal waarin we auditief wel degelijk een toonhoogteverloop kunnen waarnemen, dit verloop langs technische weg te rekonstrueren. Deze intuïtie blijkt evenwel ongegrond, niet in de laatste plaats omwille van de hoge complexiteit van het toonhoogtewaarnemingsverschijnsel van het menselijk oor.

Cfr.: LEMAN, Marc "Een Model van Toonsemantiek", 1991, p.99-117.

Leman geeft een goed overzicht van de stand van zaken en de problemen inzake toonhoogte-waarneming en extraktie. De kontekst waarin hij het probleem behandelt is daarbij signaaltechnisch gesproken heel wat 'eenvoudiger', aangezien de muzikale signalen waarover hij het heeft in hoge mate als periodiek beschouwd kunnen worden, terwijl wij eigenlijk slechts met a-periodieke ruisbanden te maken krijgen.

Informatie over de frequentie van een signaal op grond van een reeks monsters van dat signaal kan theoretisch worden bekomen door het toepassen van wiskundig (voor ons althans) relatief complexe diskrete F.F.T. (Fast Fourier Transforms), een techniek die onder meer toelaat van een signaal de spektrale samenstelling te bepalen. Voor een reeks samples geldt dan:

$$X(w) = \sum_{n=-\infty}^{\infty} x(n) \cdot e^{-j \cdot w \cdot n}$$

Voor de implementatie van een volledige FFT in real-time is het hier toegepaste computersysteem en de gebruikte ADC evenwel lang niet snel genoeg. Daarvoor zou immers een speciaal DSP-systeem gebruikt moeten worden. Rekening houdend met het feit dat de hoogste grond-frequentie componenten in het holosoundsignaal zelden groter zijn dan ca. 400Hz, en met het feit dat de sampling rate van de ADC ongeveer rond 800 a 900 samples per seconde ligt, bleek het -overeenkomstig het teorema van Nyquist- mogelijk iets over frequentie en spektrale samenstelling van het signaal te berekenen uit X samples door het aantal keer dat de signaalkurve van richting verandert te tellen. Voor zulk algoritme zijn slechts vergelijkingen van gehele getallen nodig waardoor het erg snel door het computersysteem afgehandeld kan worden. De interpretatie van de bekomen informatie bleek echter zeer problematisch.

Een tweede, en wellicht fundamenteeler probleem, is gelegen in het feit dat de FFT-analyse staat of valt met de periodiciteit van het te analyseren signaal. Ons signaal voldoet beslist niet aan deze voorwaarde.

Ontgoocheld door de weinig bemoedigende resultaten van deze pogingen tot software implementatie, (zowel naar verwerkingssnelheid als naar relevantie van de bekomen data voor de bewegingsinput) hebben we toen besloten ook hier uit te

kijken naar een oplossing via een 'kunstgreep' met een analoge computer.

Verder onderzoek op het gebied van de real-time signaal analyse is hier zeker aangewezen.

3.1.3.2: Analoge computer voor data-extractie

Bij het ontwerpen van deze eigenlijk tweede analoge computer (om in de terminologie van ons vorige hoofdstuk te blijven, gaat het dan eigenlijk over de toevoeging van enkele extra rekenblokken aan de bestaande blokkendoos) zijn we uitgegaan van de vereisten en mogelijkheden van de apparatuur aan dewelke de informatie diende afgeleverd te worden. Het probleem dat hier -omwille van de goede orde van de uiteenzetting- pas in de volgende paragraaf sub 3.1.4 wordt te berde gebracht, was dan uiteraard reeds opgelost.

Omdat de analoog naar digitaal konverter in het computersysteem waarmee we verder wilden gaan werken (een NEC ADC7004C met een resolutie van 10bit) alleen ingangssignalen verwerkt tussen 0 en +4V, was voor de aansluiting van de analoge Holosound-computerblokkendoos alvast een minimale signaal-konditionering noodzakelijk: de uitgangsspanningen mogen alleen positief zijn en mogen de 4V niet overschrijden. De uitgangssignalen van de eerste analoge computer zijn echter van nature uit en gezien hun oorspronkelijk doel, audio-signalen met een nominaal nivo van +6dB (1.55Vrms).

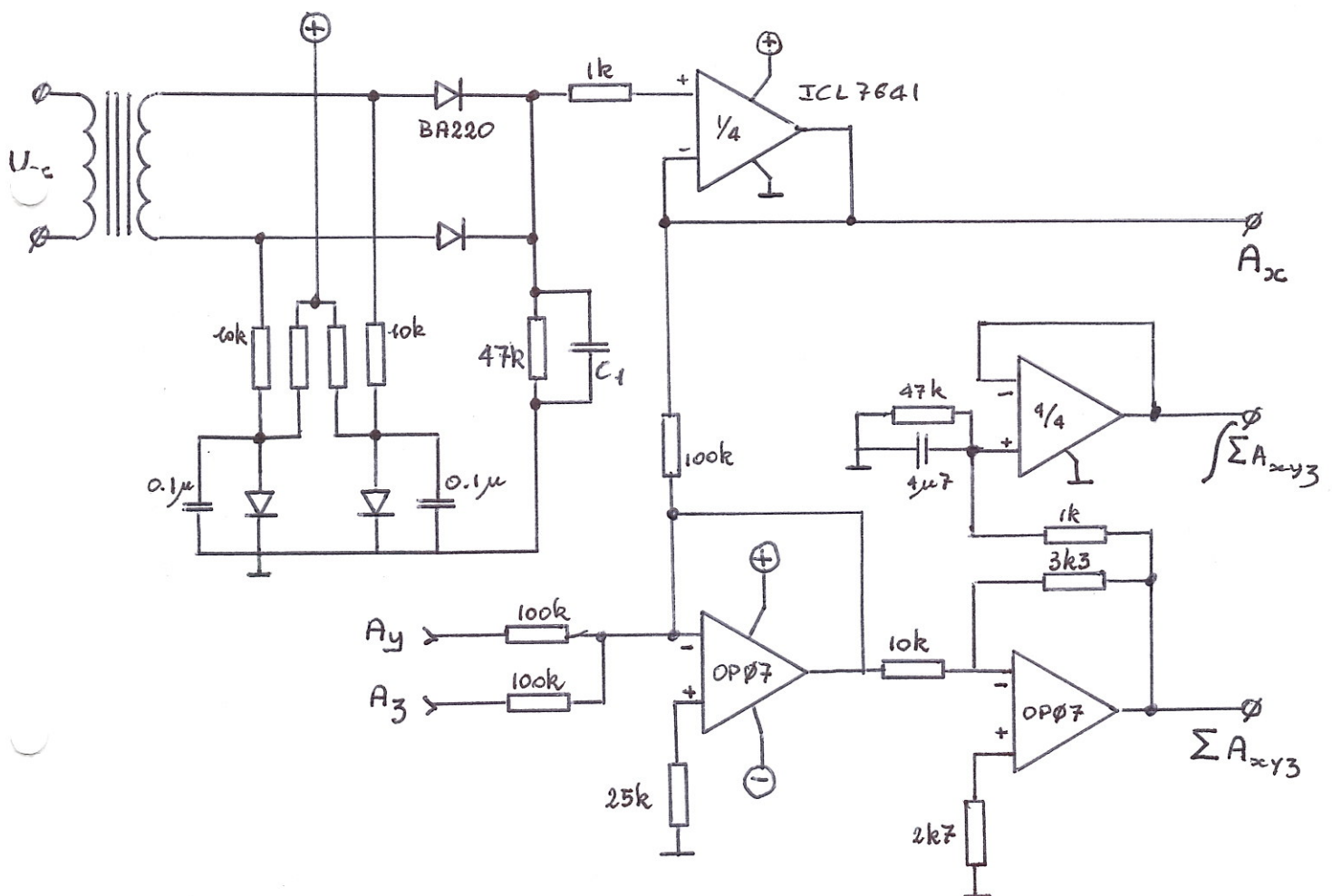
Voorts hebben we gebruik gemaakt van de eigenschap van de toegepaste ADC-konverter, dat hij over 8 onafhankelijke ingangskanalen beschikt. Dit bracht ons op het idee elke bewegingsparameter via een afzonderlijk kanaal aan de omzetter aan te bieden. Dus, 3 kanalen voor bewegings-intensiteitsinformatie, 3 voor de bewegingssnelheidsinformatie en dan hadden we nog twee kanalen op overschot, waarvoor we eveneens een bestemming vonden: deze twee kanalen zouden gebruikt worden voor de vektoriele som van de bewegings- en snelheidsinformatie der drie kanalen.

3.1.3.2.1: Bewegingshoeveelheid-data

Wat we aanvankelijk via software hadden 'opgelost', bleek via hardware een flink stuk sneller te realiseren te zijn.

Om de amplitude informatie te extraheren uit het signaal van de eerste analoge computer, maakten we gebruik van wat in het technisch jargon een *true-RMS convertor* heet, onmiddellijk gevolgd door een integrator. Teoretisch gezien dus eigenlijk een gewone amplitudedemodulator zoals we die al bespraken sub 2.3.2.6.1.

Het schema van deze schakeling, zowat de eenvoudigste van alle technische mogelijkheden, kwam er dan uit te zien als volgt:



De gebruikte diodes dienen nauwkeurig gekoppelde exemplaren te zijn, en worden tegen elkaar gemonteerd zodat ze zeker dezelfde temperatuur hebben. Alle weerstanden zijn 1% metaalfilm. De ingang is gebalanceerd uitgevoerd. In ons prototype hebben we audiotransformatoren toegepast, wat het voordeel van een perfecte blokkering van DC-offsetspanningen oplevert, en bovendien -door de 1/f karakteristiek van een transformator die sterk opvalt wanneer de frekwenties beneden de 30Hz zakken- de gevoeligheid voor trage bewegingen van het gehele lichaam aanzienlijk vermindert. Uiteraard brengt dit de wiskundige precisie van de schakeling in het gedrang, maar voor ons opzet was dat geen probleem, aangezien hiermee in de fase van de data-verwerking rekening kon gehouden worden. Naar 'betekenis' van het verkregen signaal is het een zuivere informatiewinst. Trage bewegingen van het gehele lichaam van de speler zijn immers expressief minder van belang dan snelle bewegingen van kleinere lichaamsdelen, onder meer omdat ze veelal ontsnappen aan een bewuste controle.

De RC-tijdkonstanten voor de integratietijd van de kanalen x,y,z werd gekozen op 15ms, ($47k\Omega * 320nF = 15ms$) wat overeenkomt met een laagdoorlaatfilter met -3dB punt rond 10Hz. Deze waarde werd bepaald op grond van experimenteel onderzoek in combinatie met de verder beschreven software. De uitgangssignalen worden na deze bewerkingen inverterend gemengd, en nogmaals geïnverteerd en geïntegreerd waardoor we op de uitgang ΣA_{xyz} het geïntegreerd resultaat krijgen van de bewerking

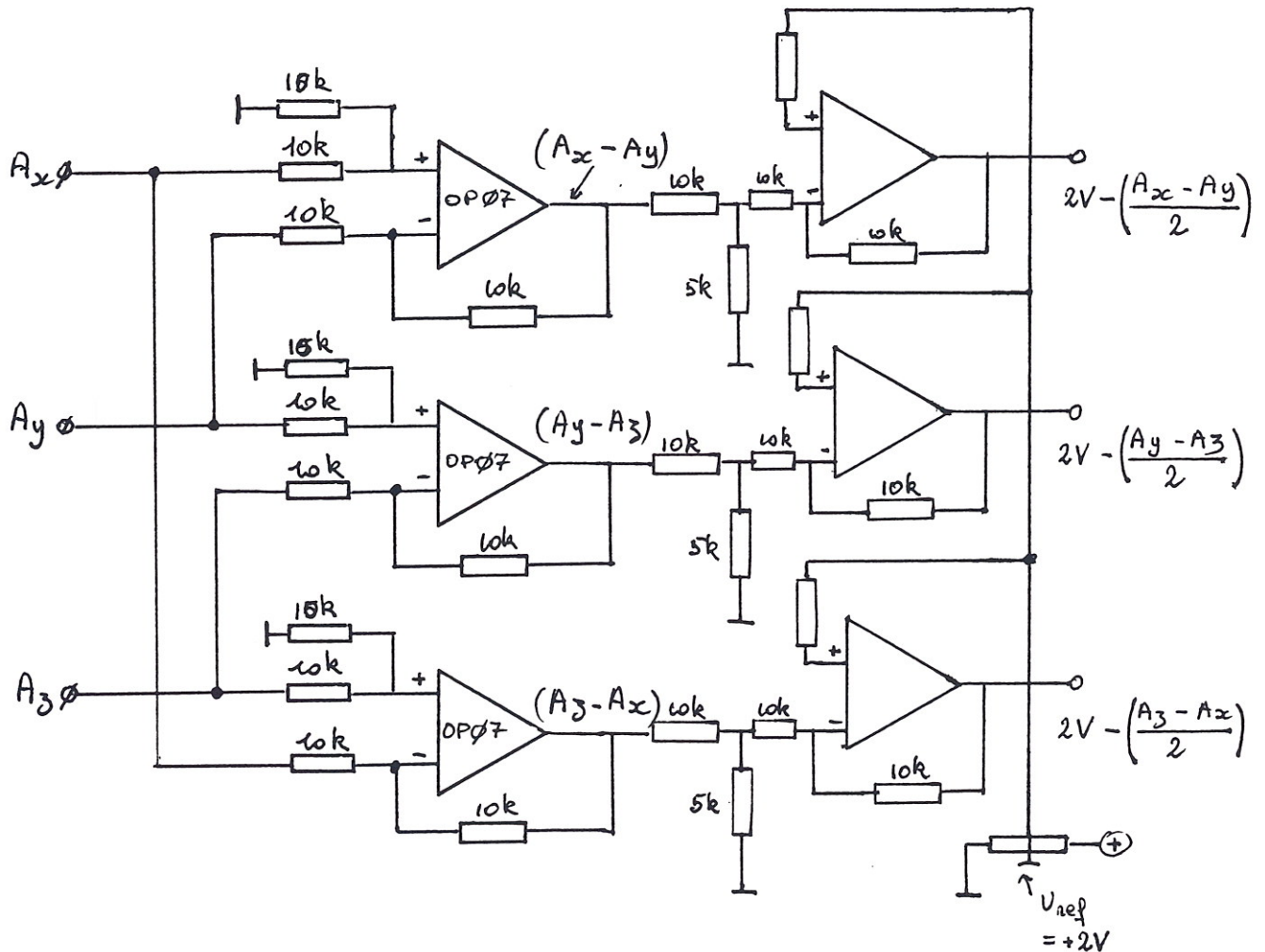
$$A_x + A_y + A_z$$

De RC-tijdkonstanten voor de integratietijd van het amplitude-som-kanaal werd bepaald op 220ms ($47k\Omega * 4700nF = 220ms$).

Dit is equivalent met een laagdoorlaatfilter met een afsnijfrequentie $f_c = 1 / (2 * \pi * R * C) = 0.723Hz$. Met deze eveneens experimenteel bepaalde waarden, bleek het mogelijk zoiets als de omhullende van de totale bewegingshoeveelheidskurve te verkrijgen. Het signaal kan vrij goed worden gebruikt om kwazi rechtstreeks toegepast te worden op een muzikale parameter zoals de globale dynamiek van de muzikale gebeurtenissen.

Een extra toegevoegde analoge rekenschakeling laat ons toe, in plaats van de A_x , A_y , A_z signalen, de onderlinge verschilsignalen te berekenen en als informatie voor verdere verwerking te gebruiken. Het is een gelijkaardige configuratie als die welke we gebruikten bij de vermenigvuldigers die we bespraken sub 2.3.3 en 2.3.3.3.7.

De uitgangssignalen hier -uitgedrukt in functie van A_x tot A_z - zijn $\{2 - [(A_x - A_y)/2]\}$, $\{2 - [(A_y - A_z)/2]\}$, $\{2 - [(A_z - A_x)/2]\}$, en moeten na de verdere analoge digitaal conversie gelezen worden als in binair 'two's-complement' gekodeerde getallen. Het nulpunt is hier immers verschoven naar 2Volt, het middelpunt van het 0-4V bereik.



Voor beide schakelingen is de keuze van het gebruikte type operationele versterkers vooral in de laatste trap van overlevingsbelang voor de ernavolgende schakeling. De ICL7641 op-amp wordt gevoed uit een enkele spanning van 5 Volt. Het is een type dat ook in die omstandigheden zijn uitgangsspanning volledig naar 0V en +5V kan laten gaan en bovendien vrij is van hinderlijke latch-up verschijnselen wanneer het ingangssignaal tegen het plafond aangaat. Hierdoor kan ons uitgangssignaal nooit groter noch kleiner zijn dan de maximale waarden die toelaatbaar zijn voor de analoog-digitaal konverter.

Door deze opamp keuze hebben we dus eigenlijk meteen een absolute begrenzer/limiter geïmplementeerd.

3.1.3.2.2: Bewegingssnelheidsdata

Zoals we eerder uitvoerig hebben aangetoond, is de bewegingssnelheid in het signaal aanwezig in de vorm van de frekwentiebandbreedte van het signaal. Puur theoretisch gezien zouden we hier dus een analoge omzetter moeten bouwen die de bandbreedte van een aperiodisch signaal omzet in een analoge spanning. Analytisch wiskundig gezien is dit een behoorlijk zwaar probleem, dat we hebben proberen omzeilen als volgt:

Stel dat we het signaal eerst ontdoen van zijn amplitudemodulatie. Dit kan eenvoudig gebeuren via een komparator-schakeling die men een nuldoorgangsdetektor noemt. Hoe meer nul-doorgangen het signaal nu heeft, hoe meer hoge frekwenties erin aanwezig zullen zijn. Stel nu dat we elke puls eenzelfde lading in een condensator laten opslaan en dat we de spanning over de condensator zouden integreren, dan moeten we de klus min of meer hebben geklaard. Immers de spanning op de condensator zal een functie zijn van de mate waarin hogere frekwenties in het signaal zijn terug te vinden.

Na wat opzoekingswerk in het overweldigende aanbod aan geïntegreerde schakelingen van diverse fabrikanten, viel ons oog op een verzameling chips die als tacho-konverters bekend staan. Het opzet van de interne schakelingen bleek goed overeen te komen met wat we nodig hadden, hoewel de chip-functie aangeduid wordt als *Frequency to voltage converter* en wij eigenlijk geen periodieke signalen ter verwerking konden aanbieden, bleek het probleemloos te functioneren.

We kozen voor de LM2907 chip van National Electronics. Volgens de databoeken is de uitgangsspanning van deze LM2907 chip een eenvoudige functie van de ingangsfrekwentie en wel als volgt:

$$V_{out} = F_{in} * V_{cc} * R1 * C1$$

Omdat we toch reeds een perfecte en alleen in het analoog gedeelte aangewende 5V voedingsspanning hadden gemaakt voor de voeding van de opamps in de bewegingshoeveelheidsschakeling (dit is overigens de laagste toelaatbare spanning voor de betrokken chips), en omdat het frekwentiegebied dat in deze toepassing voor ons van belang is beperkt kan worden tot 0 - 350 Hz, (overeenkomstig bewegingssnelheden tot circa 3 m/s wanneer een draaggolffrekwentie van 40 a 50kHz gebruikt wordt) berekenden we de waarden voor R1 en C1 als 22nF en 100kΩ. De hoogste meetbare ingangsfrekwentie wordt bepaald door:

$$f_{max} = 12 / (C1 * V_{cc})$$

De toegepaste schakeling wordt voorafgegaan door een eenvoudig π -low-pass LC-filter (82mH en 100nF) met een afsnijfrekwentie van 1200Hz. Hierdoor worden alle eventueel resterende 40kHz signalen buiten de tacho-schakeling gehouden. Zonder deze voorzorgsmaatregel bleef de uitgangsspanning van de frekwentie naar spanningsomzetter toch nog hangen op V_{cc} !

Over R1 staat een condensator die bepalend is voor de stabilisatietijd van de omzetting en tevens voor de grootte van de ingangsrimpelspanning aanwezig in het uitgangssignaal.

Deze rimpelspanningskomponent (die na de analoog-digitaal omzetting uiteraard zal verschijnen als een ruissignaal

-d.w.z. een signaal dat geen drager is van relevante informatie) wordt berekend als:

$$V_{rpp} = (V_{CC}/2) * (C1/C2) * (1 - ((V_{CC} * f_i * C1)/I2))$$

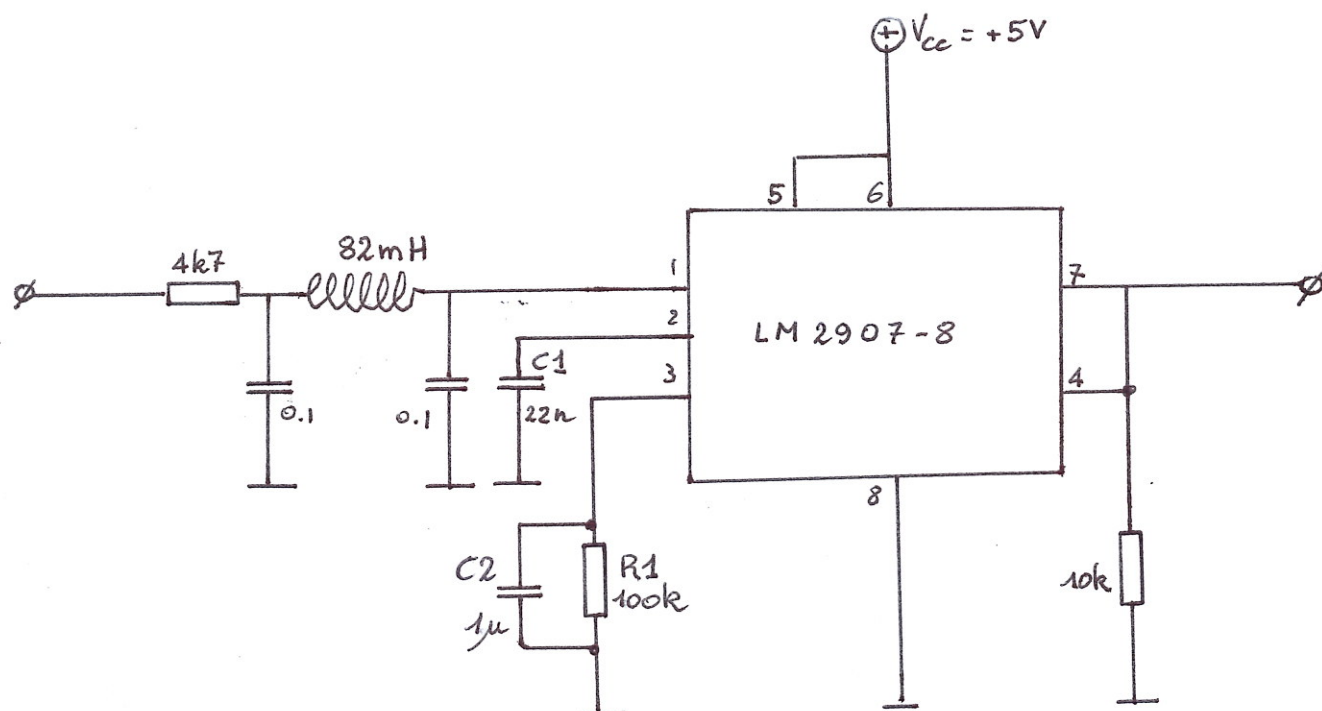
De rimpelspanning wordt 0 Volt vanzodra de ingangsfrequentie naar haar hoogste waarde gaat. Dit komt ons goed uit, want het betekent dat de resolutie voor hoge bewegingssnelheid groter zal zijn dan voor kleine. De term $1 - ((V_{CC} * f_i * C1)/I2)$ wordt dan immers 0. Voor de laagste voor ons relevante waarden van f_i krijgen we een rimpelspanning van:

$$\text{stel } f_i = 10\text{Hz} \quad U_{r-pp} = 15\text{mV}$$

Voor een 7 bit resolutie van de ADC convertors moet de rimpelspanning kleiner zijn dan $4V / 127 = 31\text{mV}$. Bovenstaande waarde zou dus marginaal voldoende zijn om zelfs een 8-bit resolutie niet in het gedrang te brengen. Het oplossend vermogen van dit onderdeel van de analoge komputer samen met de verder gebruikte AD-software beperkt dus de laagste meetbare frequentie, en dus bewegingssnelheid, tot ongeveer 5 Hz wat overeenkomt met een beweging van ca. 5cm/s.

Overigens moet opgemerkt worden dat ^{Ter herinnering: $v_b = f_{diff} * v_{geluid} / f_c$} alhoewel het lijkt dat we eenvoudig een resolutie van 8 bits voor de laagste frequenties en zelfs van 10 en meer bits voor de hoogste zouden kunnen halen, dit toch geen zin zou hebben omdat de precisie van de frequentie naar spanning omzeters in de data sheets voor deze chips toch slechts bepaald wordt op 1%, wat op zich reeds zelfs het LSB van een 7-bit konversie onbetrouwbaar maakt! Bovendien introduceert ook de aperiodiciteit van de signalen waarmee wij hier werken, tegenover die waarvoor de specificaties van de chip worden gegeven een onzekerheidsfactor.

Het schema van de uiteindelijk gebouwde en geteste schakeling kwam er uit te zien als volgt:



Ook de uitgangssignalen van deze schakeling worden aangeboden aan 3 afzonderlijke ingangskanalen van de analoog-digitaal converters op de 'dedicated single board computer' waarover we het nu gaan hebben.

3.1.4: Data-verwerking

Bij onze eerste pogingen om de informatieverwerking te doen in software, maakten we gebruik van een van onze IBM-AT klonen. Nu is dit soort machine niet bepaald erg handelbaar wanneer het erop aan komt ze te gebruiken in een implementatie van een realistisch muziekinstrument. Ze zijn betrekkelijk groot, lomp, kwetsbaar en... maken lawaai op de koop toe.

De lawaaibronnen zijn enerzijds de ventilator in het voedingsblok en anderzijds, de kontinu draaiende motor van de harddisk. De ventilator kan -althans rekening houdend met de gangbare omgevingstemperaturen in onze kontreien- zonder enig gevaar uitgeschakeld worden, de harddisk evenwel niet.

Bovendien veroorzaakt de ventilator in het voedingsblok een ontoelaatbare verstoring van het ultrasoon veld, dit dan door de beweging van de schoepen. Aangezien we voor vroegere automatiseringsprojecten reeds vaker voor dergelijke problemen waren geplaatst, dachten we ook hier onmiddellijk in de richting van een 'microcontroller'. Dit is in wezen een klein zelfstandig werkend computersysteem, met een processor, RAM en ROM geheugen, enkele I/O poorten en een seriële poort via dewelke middels een 'host'-komputer data en programmas in het systeem kunnen worden geladen. Het zijn dus eigenlijk volwaardige computersystemen maar dan zonder beeldscherm, zonder klavier en zonder enig perifeer toestel. Hoewel men dat zich zelden realiseert is het eigenlijk het meest toegepaste en het in de grootste aantallen industrieel gefabriceerd soort komputer.

De types waar we vroeger reeds veel ervaring hadden mee opgedaan waren resp. de 8052 van Intel en het Slicer-board (met een 80186 16-bit Intel processor) dat werkt onder CP/M of een primitief soort DOS. Het eerste type (de 8052) leek ons ondanks de vele argumenten die ervoor pleiten, in deze toepassing te beperkt. Tegenover het voordeel van de gemakkelijke programmeerbaarheid stond het nadeel van de trage snelheid vooral voortkomend uit het feit dat de Basic programmas niet konden gekompileerd worden. De Slicer -hoewel ik er via een genereuze gift van een sponsor aan Stichting Logos zo'n 30 exemplaren van in voorraad heb en had- kon in een komputertaal naar keuze worden geprogrammeerd, maar door de kompleksiteit van het operating system (meer bepaald de interrupt handlers) bleek het erg onbetrouwbaar voor dit soort taak waar het toch in eerste plaats op real-time signaalverwerking aankomt. Bovendien was het systeem behept met enkele storende 'bugs'.

Waarom ik eigenlijk behoefte had in deze faze, was aan een komputer die ofwel een parallelle in plaats van een seriële architectuur bood (wat me zou toelaten elke parameter afzonderlijk te verwerken), ofwel aan eentje dat over intrinsieke multitasking mogelijkheden zou beschikken. Na wat zoeken, vond ik ook daadwerkelijk zo'n ding...

De komputer die ik hiervoor dan uiteindelijk aanschafte en inzette, was een 'Multitrax' multitasking microcontroller: dit is een gehele komputer die uit niets meer dan 1 enkele printplaat bestaat en noch scherm, noch klavier heeft. De kommunikatie met de programmeur gebeurt via een terminal. Hiertoe wordt de RS232 poort waarover het board beschikt, verbonden met een terminal of eender welke 'gewone' komputer, waarop dan een kommunikatieprogramma loopt. Voor PC's van de IBM-familie beveel ik daarvoor het programma 'Procomm' sterk aan. Wanneer het programma is geschreven en getest, kan het -na compilatie- op het board zelf in een EPROM worden opgeslagen, waarna het systeem het programma zonder nood aan terminal of externe komputer, volkomen autonoom kan draaien.

Het leuke van dit komputersysteem is dat het ingebouwde besturingssysteem, werkelijke multi-tasking toelaat. Dit wil zeggen, dat het systeem verschillende taken tegelijkertijd kan afhandelen. Bovendien is het systeem rechtstreeks in Basic te programmeren: daartoe werd een ROM-basic ingebouwd, inclusief een compiler voor de omzetting naar machinetaal. De ontwikkelde software loopt alleen na compilatie op het systeem, zodat alle nadelen verbonden aan Basic-interpreters voor dit systeem niet geldig zijn.

De erin toegepaste mikroprocessor is een snelle 8-bits CMOS 64180 gefabriceerd door het Japanse Hitachi, wat eigenlijk een meer geavanceerde uitgave is van de vanouds bekende Z80 processor, een chip waarmee we reeds in de jaren '70 hadden gewerkt. Naar geheugen is het een voor hedendaagse normen erg bescheiden systeem: 32kByte RAM en 96kByte ROM. Meer ROM dan RAM dus, iets wat logisch is gezien de industriële bedoeling van het ontwerp.

Naast deze aspecten -en dit punt was natuurlijk niet aan onze aandacht ontsnapt- kon het board ook eenvoudig worden uitgebreid met een 8-kanaals ADC (Analog Digital Converter) met een resolutie van 10-bits, een ingebouwde real-time klok, 48 digitale I/O bits, een frekwentieteller, een tweede vrij programmeerbare RS232-poort, een LCD-scherm poort en een eenvoudige toetsenborddekodeer voor 16 toetsen... Deze uitbreidingen moesten er wel zelf bijgebouwd worden, maar de verschafta dokumentatie was erg goed en bovendien werden de uitbreidingen reeds zowel in het operating system als in de technische Basic implementatie voorzien.

Technische informatie m.b.t. het beschreven komputersysteem:

'Multitrax' - producent: Control Technology, Box 4605, Mountain View, CA 94040, USA.

Distributie: SINTEC Company, P.O. Box 410, Frenchtown NJ 08825 USA (Tel.:800 526-5960).

Wij besloten het board van volgende extra 'features' te voorzien met het oog op gebruik in het onzichtbare muziekinstrument:

-AD-konverter met de NEC D7004C chip

Deze chip leek ons ideaal geschikt voor ons doel. Hij is uitgevoerd in CMOS-technologie en het gevolgde AD-konversie procede is dat van de geleidelijke stapsgewijze benadering (Successive Approximation Register -SAR). Via multiplexing beschikt hij over 8 analoge ingangskanalen. Multiplexing is hier het om beurten aftasten van de verschillende kanalen. De

konversietijd bedraagt slechts 100 μ s. De ingangsspanningen mogen in geen geval negatief zijn en ook niet de voedingsspanning (+5V) overschrijden. Daarom beperkten wij deze in ons hiervoor beschreven analoog komputerontwerp tot het bereik 0-4V.

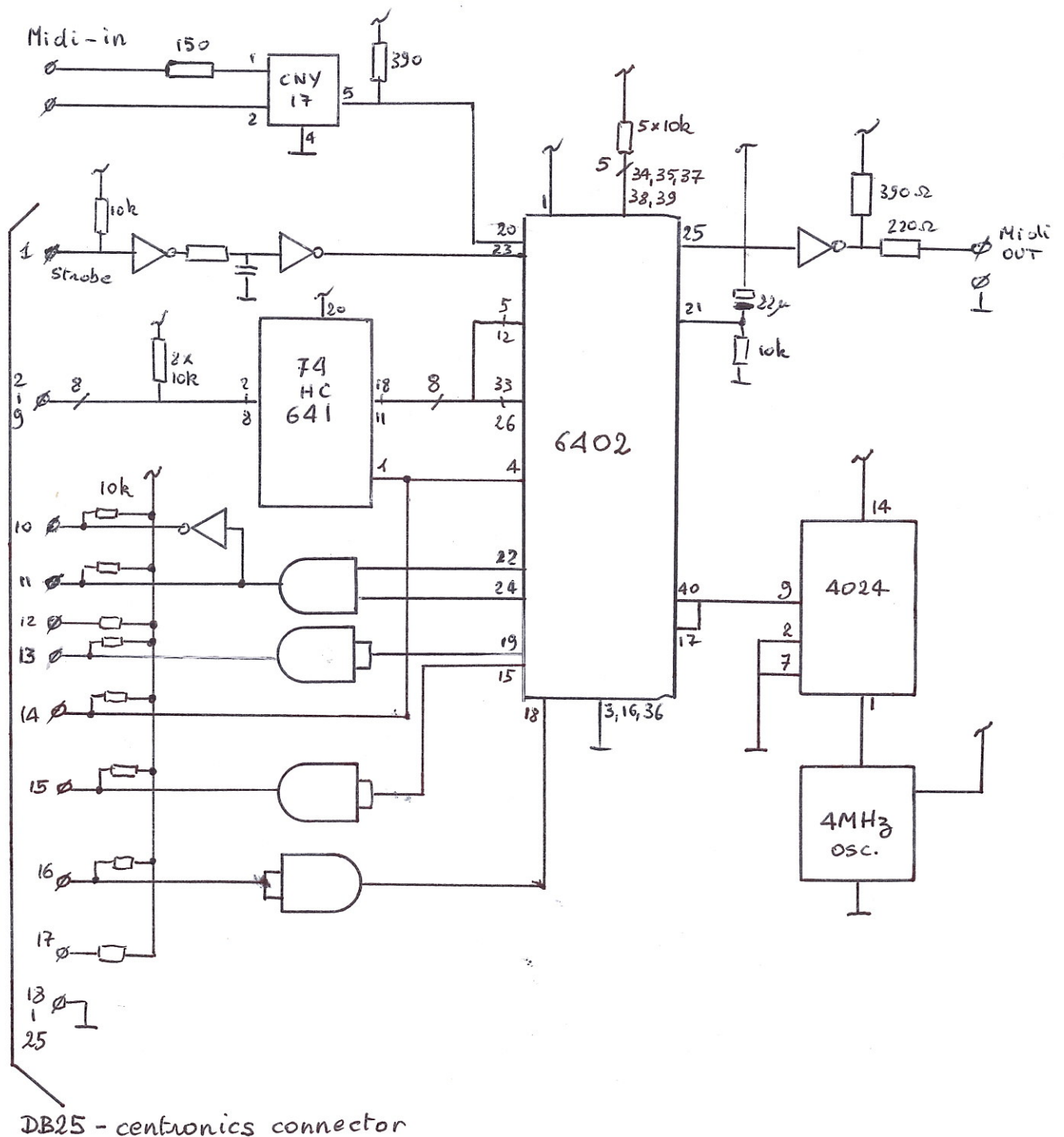
- Een 1-regelig LCD-display
Dit om toch enige feedback naar de gebruiker mogelijk te maken. Vergeten we immers niet dat het de bedoeling was een volstrekt zelfstandig opererend instrument op te zetten en dus niet iets wat alleen middels een bureelcomputer of via een ingewikkelde labo-opstelling aan de praat te krijgen was. Dit is immers de enige mogelijkheid om wanneer onverhoopt eens iets niet helemaal naar behoren zou functioneren, via software meldingen naar de gebruiker te sturen. Wij pasten een display toe van de firma Varitronics Ltd. Op de aansluitproblemen daarvan gaan we hier niet verder ingaan omdat dit niets te maken heeft met het instrument als dusdanig of met enige esthetische konsekwentie.
- Een midi-poort
Over zulk interface beschikten we reeds, maar er dienden stappen ondernomen te worden om de op het board geïmplementeerde printerpoort -die slechts 7-bits gebruikte- om te bouwen naar een volwaardige universele 8-bit poort. Meer hierover sub. 3.2.5.
- Een precieze timer
Gezien we het systeem voor muzikale doelen wilden gebruiken was het essentieel over een precieze en van de kloksnelheid van de mikroprocessor zelf onafhankelijke bron van tijd-informatie te kunnen beschikken. Hiervoor gebruikten we de MM58274 van National Semiconductor.
- Een miniatuur toetsenbord
We besloten het board van een minimaal 'input-device' te voorzien, in de vorm van een toetsenbord met 16 toetsen, een beetje zoals we dat kennen van de elektronische rekenmachines. 16 toetsen achtte ik voldoende ook voor het inbrengen van hexadecimale kodes.

3.1.5: Data-output

MIDI INTERFACE

Hiervoor gebruikten we het uit 1984 daterende ontwerp voor een MIDI-interface uitgaand van de signalen beschikbaar op een Centronics printerpoort, zoals we die op bijna alle computers terugvinden.

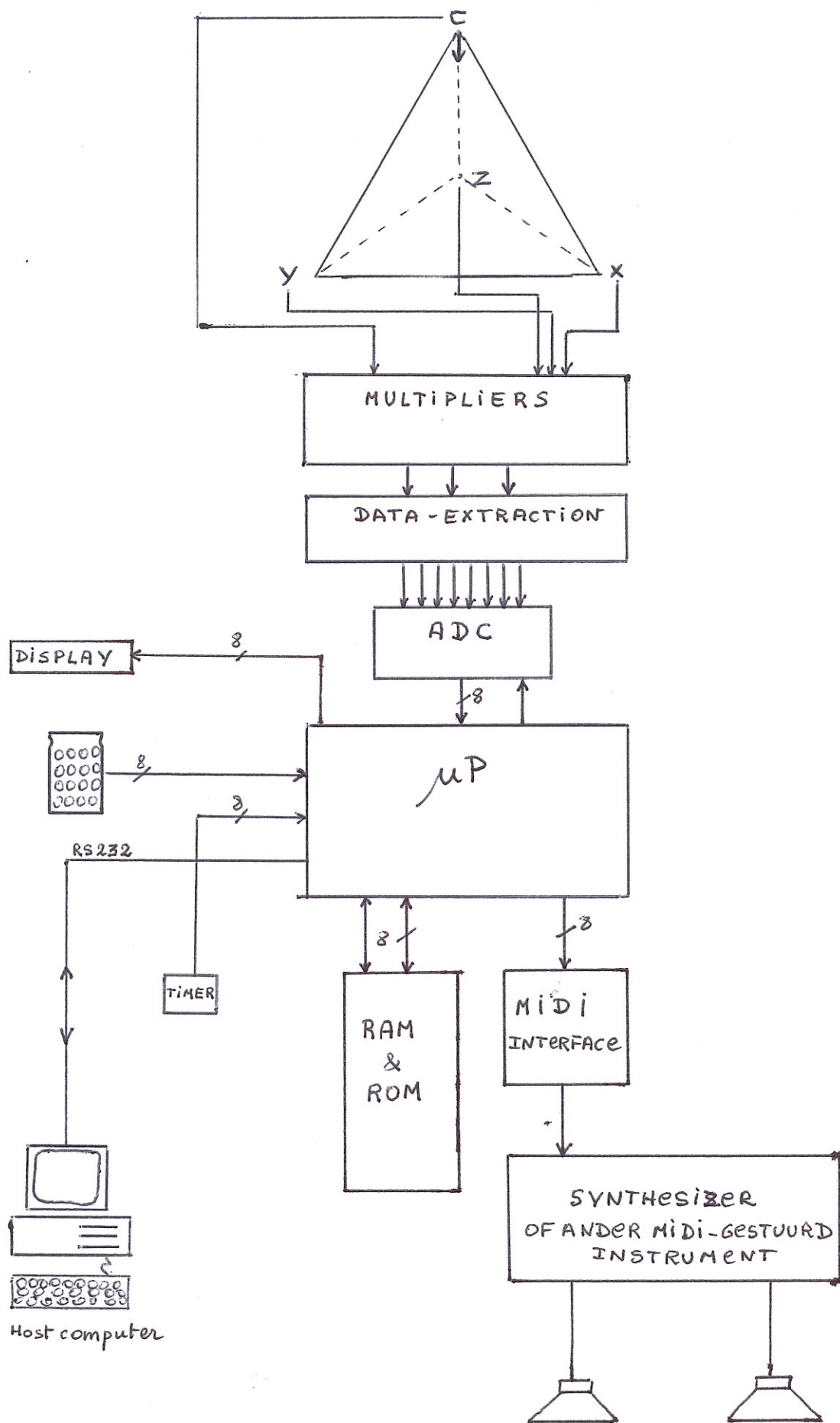
Hoewel het interface bidirectioneel is, wordt het in deze toepassing slechts als output-device gebruikt. De besturingswijze blijkt duidelijk uit de hiernavolgende software.



3.1.6: Evaluatie & blokschema van de gehele hardware

Hiermee is de gehele hardware van het instrument -of althans het eerste redelijk goed werkend prototype ervan- beschreven en verklaard.

Volgend schema vat samen hoe alle beschreven blokjes met elkaar worden verbonden. Het is belangrijk dit voor ogen te houden voor een goed begrip van de hierna beschreven software, waarzonder het instrument niet kan funktioneren.



Uiteraard kan software niet meer dan een academisch bestaan kennen, wanneer er geen hardware gespecificeerd wordt waarop die software dient te lopen. Niettemin, kan men opteren voor een minimale hardwarematige signaalverwerking en zoveel mogelijk via software-technieken afhandelen. De weg dus die wij hier niet hebben bewandeld. Het meest aangehaalde voordeel van een dergelijke softwarematige aanpak, zou zijn dat dit een grotere flexibiliteit mogelijk zou maken. Het zou immers mogelijk zijn via aanpassingen van de software of via geheel nieuwe software, totaal andere machines te koncipieren en te doen functioneren.

Uit o.m. mijn hiervoor geschetste persoonlijke ervaring terzake evenwel, meen ik te mogen stellen dat dit een fabeltje is of toch minstens gerelativeerd dient te worden. De ontwikkelingstijd nodig voor het schrijven van een werkend stuk signaalverwerkingssoftware is beslist in dezelfde orde van grootte dan de tijd nodig voor het berekenen, ontwerpen en bouwen van een modern stuk hardware. Het fabeltje blijft evenwel hardnekkig standhouden, vooral m.i. omdat beide aanpakken een grondig verschillende technische kunde en competentie veronderstellen. Het zijn onderscheiden specialisaties geworden die naarmate de specialisatie verder doorgedreven raakte, verder en verder uiteen zijn gegroeid. Het aantal software ontwikkelaars dat vandaag nog op de hoogte is van digitale hardware, is (helaas) uiterst klein. Dit legt onvermijdelijk een hypoteek op de vinding van optimale oplossingen voor gestelde problemen in het domein van de informatieverwerking in het algemeen.

Vooraf in het domein dat ons bezighoudt, de muziek, constateren we de laatste jaren bij de al te weinigen die zich überhaupt nog met de 'konstruktieve' aspecten van het muziekmaken bezighouden, een alsmaar sterkere verschuiving naar software-ontwikkeling ten nadele van de hardware. De hardware wordt al te vaak bekeken als de 'loodgieterij' van de hogere heren. Er wordt op neergekeken. Hier zien we een vervreemding ontstaan die we ook zien bij musici die van hun eigen instrument vervreemd raken. Hoeveel pianisten slagen er vandaag nog in een piano open te maken en er herstellingen aan uit te voeren? Welke pianist kan nog een piano stemmen of weet zelfs hoe dat theoretisch zou moeten gedaan worden?

Verder nog constateren we, binnen de software, een evolutie van echt programmeerwerk, naar louter gebruik van zuivere toepassingssoftware. De muzikus lijkt in deze ontwikkeling de o.i. noodzakelijke greep op zijn klinkende materie en vooral ook op de wijze waarop hij daarmee zal omgaan te verliezen. Hij wordt -extreem gesteld- veeleer een slachtoffer van de technologische ontwikkeling, dan een bevoordeligde ervan.

3.2. Software

3.2.1.1: 'Mapping'

Uiteraard doet al deze hardware helemaal niets, zolang er geen software wordt geschreven om op een relevante manier de aangeboden informatie te verwerken en tot muzikaal aanvaardbare resultaten te transformeren.

Nu is het precies op het vlak van deze software dat de uiteindelijke bepaling van het instrument tot stand zal kunnen komen. Taak van de software zal er moeten in bestaan uit de beschikbare data op grond van beslissingsregels muzikaal relevante en samenhangende informatie af te leveren. Het centrale begrip hierbij is dat van de 'mapping'.

'Mapping' noemen we de wijze van afbeelden van informatie uit een eerste domein naar informatie uit een ander domein. De Nederlandse vertaling 'inkaartbrenging' is niet echt geschikt, zodat we ons aan de Engelse terminologie zullen houden.

De 16 verschillende programmabrokken die samen het 'Book of Moves' vormen zijn niets anders dan 16 verschillende 'mappings' in deze zin. Elke mapping doet in menig opzicht, een nieuw instrument ontstaan. Ook de instructie van het via een mapping geïmplementeerd instrument, hangt volledig af van de aard van die mapping. We hopen dit in de hierna volgende kode en de erbij horende kommentaar duidelijk te maken.

Vooraf willen we er wel nog op wijzen, dat de hier gegeven kode gegeven is bij wijze van voorbeeld. Het is de kode zoals wij die voor de definiering van enkele praktische instrumenten voor eigen gebruik ontwikkelden. De muzikus die zich van ons instrument zou wensen te bedienen, dient zich -via het vastleggen van een voor zijn voorkeur geschikte mapping- dit instrument eerst te bepalen. Hij zal daartoe dus ook een eigen stukje software moeten schrijven. Moeilijk is dit echter niet.

3.2.1.2.: Multitasking Basic

Het BASIC dat voor deze processor en dit komputerboard werd geïmplementeerd is niet helemaal gelijk te stellen met het 'plain-vanilla' Basic zoals we dat kennen van simpele home-komputers. Het beschikt immers over multi-tasking mogelijkheden die op eenvoudige machines niet voorhanden zijn.

Typische instructies die betrekking hebben op deze multi-tasking mogelijkheden en waarmee de lezer allicht niet vertrouwd is zijn:

```
START
SUSPEND
EXIT
TASK
```

De START instructie wordt gebruikt om een taak van start te laten gaan. Elke gedefinieerde taak heeft een nummer evenals een periodiciteit, waarmee tevens de prioriteit tegenover andere taken kan worden vastgelegd. De syntax is dan ook:

```
START x,y.
```

waarbij x het taaknummer is en y de periodiciteit ervan.

Met het kommando EXIT kan een taak verlaten worden.

Voor het overige is dit Basic echter helemaal niet bijzonder geavanceerd. Klassieke structuren zoals IF-THEN-ELSE (het ELSE deel werd niet voorzien) en de Booleaanse negatie NOT werden helaas niet geïmplementeerd. Ook onze favoriete DO ... LOOP UNTIL... structuur moesten we ontberen evenals het programmeren in modulaire structuren zoals we dat kennen in Microsoft Quick-Basic. Het is evenmin mogelijk zelf de ROM-library uit te breiden, althans niet zonder eerst het hele Basic zelf te gaan herschrijven, iets waartoe we ons niet geroepen voelden.

Ook de mogelijkheden tot 'nesting' zijn erg beperkt. Om die redenen konden we dan ook de (door de 'schoolmeesters' van de informatika) verfoeilijk geachte GOTO's niet steeds vermijden in ons programma.

Daar waar meer geavanceerde Basics (Quick-Basic, Visual Basic of PDF-Basic van Microsoft of GFA-Basic voor IBM-PC's bijvoorbeeld) de programmeur toelaten procedures te definiëren, hebben we dit hier via 'user functions' moeten doen.

Gosub's zijn in dit stukje software dan ook nagenoeg niet aan te treffen. Dit komt -zo hopen we althans- de leesbaarheid zeer ten goede.

De subroutine voor het spelen van samenklanken opgebouwd uit 8 noten kon echter onmogelijk als een functie herschreven worden omdat functies in dit Basic de overdracht van array-variabelen niet toelaten.

Gezien het technische karakter van dit soort komputer -een industriële mikrocontroller- beschikt het geïmplementeerde Basic over enkele Booleaanse functies geschikt voor gegevensmanipulatie op het nivo der individuele bits in hexadecimale getallen. Functies dus, die zeer dicht staan bij machinecode en uiteindelijk bij de hardware zelf. We hebben er dankbaar gebruik van gemaakt. Het is niet in het minst door deze mogelijkheid dat dit Basic zo'n goede prestaties kan halen met betrekking tot uitvoeringssnelheid. De functies zijn:

BAND
BOR
BXOR

Het zijn de zuiver binaire equivalenten van de logische functies waarover elk gewoon Basic beschikt: AND, OR, XOR. De gebruikelijke waarheidstabellen zijn hier uiteraard van toepassing.

Zoals bij zowat alle professionele talen, eist ook dit Basic dat variabelen worden gedeclareerd op voorhand als geheel getal (INTEGER), reëel getal (REAL) of tekenreeks (STRING). Andere mogelijkheden zijn er niet.

3.2.2: Kode: "A BOOK OF MOVES"

We hebben de diverse programmasegmenten die volgen in ruime mate voorzien van commentaren teneinde de precieze werking duidelijk te maken. We hopen dat het programma zodoende ook voor de niet in Basic onderlegde lezer goed te volgen is. Alleen regels die aanvangen met een getal zijn eigenlijke programmaregels. Al het overige is commentaar.

Kode geschreven voor het Multitrax board met een 'extended-MTBasic' (multitasking Basic), EPROM-chip. De kode loopt slechts na compilatie. De wijze waarop dit dient te gebeuren, is in de appendix toegevoegd. De hier weergegeven listing volgt de versie met nummer MT42 d.d. 19.08.1992). Deze versie werd gebruikt in een twintigtal publieke en internationale voorstellingen en lezingen. Ze mag dus als getest worden beschouwd. Gezien het feit dat aan dit projekt doorlopend wordt voortgewerkt, komt dit op het ogenblik dat dit gel wordt, niet noodzakelijk overeen met de meest recente staat van het programma. Deze opmerking is overigens ook van kracht met betrekking tot de gebruikte hardware zelf. De historiek van het programma hebben we als dokumentatie eveneens in de appendix ondergebracht.

3.2.2.1.- Deklaraties

In volgende lijst, uitmondend in de eerste twee programmaregels, worden alle verder in het programma gebruikte variabelen gedeclareerd en hun gebruik binnen het programma vastgelegd. Dit terwille van de leesbaarheid en het onderhoud van de software.

A
naam van de variabele gebruikt voor de ingelezen som der bewegingsamplitudes van de 3 kanalen x,y,z. Hierin staan de laatst gelezen inhoud van ADC(3)

C
variabele gebruikt voor de ruimtelijke plaats waar een beweging werd gedetekteerd. (memo: CASE)

D
dummy-variabele die gebruikt wordt in de 'valse' funkties waarmee U(), UPAR(),P() en andere worden aanroepen. De waarde van D heeft zelf geen enkele betekenis.

E
Deze variabele wordt alleen gebruikt om de volgorde waarin de diverse algoritmische mappings worden uitgevoerd in op te slaan. Deze variabele wordt door de gebruiker via het toetsenbordje ingebracht. Er zijn slechts 16 mogelijke waarden: 0-F hexadecimaal.

I
Algemene teller voor gebruik in lussen. Eveneens toegepast als variabele voor het opslaan van een vorige toestand in taak 5 evenals als variabele voor het tonaal-centrum in bepaalde algoritmes.

J
Algemene teller variabele in lussen

L
Teller variabele, meestal gebruikt voor tijdsbepalende routines

via de SYSTIC-instructie.

M

Variabele gebruikt als kanaal-teller in de midi-kommandos.

T

Tijdsvariabele gebruikt in de functie S()

X,Y,Z

Variabelen gebruikt in de taken waarin de ADC waarden worden uitgelezen voor de drie bewegingsvektoren. X heeft steeds betrekking op een parameter van het x-kanaal (links), Y op het Y-kanaal (rechts) en Z op het Z-kanaal (boven). Deze variabelen worden door de multitasking ververst (updated) gedurende de procedures.

XH,YH,ZH

Deze variabelen worden gebruikt als houd-variabelen wanneer een verversing van de inhoud ongewenst is. Bvb. wanneer na een IF konditie een bewerking wordt uitgevoerd waarin die variabele wordt gebruikt.

Dit is wellicht niet duidelijk voor wie niet vertrouwd is met multitasking. Daarom een klein voorbeeld. Onderstel dat we een zin formuleren zoals: IF XH < 64 THEN XH=XH+63. In een 'normaal' programma mag aangenomen worden dat de waarde van XH wanneer aan de initiële konditie XH < 64 volstaan is, niet groter kan zijn dan 127. Indien evenwel XH behoort tot de klasse der variabelen die op dat ogenblik in een taak worden ververst, dan is het mogelijk dat eerst de voorwaarde XH < 64 volstaan is, vervolgens XH ververst wordt en de waarde 120 krijgt, waarna ingevolge XH=XH+63 aan XH uiteindelijk de waarde 183 zou verkrijgen. Dit was uiteraard niet de bedoeling van onze voorwaardelijke bepaling. Vandaar dus de noodzaak om tussenvariabelen te gebruiken in dergelijke gevallen.

Deze variabelen worden ook gebruikt voor de opslag van de vorige waarde van X, Y, Z in vergelijkingsalgoritmen.

XF,YF,ZF

Deze variabelen worden gebruikt voor de bewegingssnelheden in de drie vektoren. Via deze variabelen worden dus de uitgangen van de analoge tacho-komputer uitgelezen.

U

Dit is de naam van een gedefinieerde functie. Namen van functies moeten in MT-Basic gedeclareerd worden alsof het variabelen waren.

UPAR

Naam voor de functie die gebruikt wordt om parameter-veranderingen in de Proteus synthesizer te bewerkstelligen. Dit is niet in alle versies van dit programma geïmplementeerd.

P

Naam van de functie waarmee voorinstellingen en panning van de synthesizer audiokanalen wordt ingesteld.

S

Naam van de functie die gebruikt wordt in de tijdsbepalende routine.

Aanvankelijk werd dit gedaan door gebruik te maken van het SUSPEND kommando, maar de implementatie daarvan in MT-Basic heeft een merkwaardige Bug die het onbruikbaar maakt...

B

Variabele gebruikt voor de opslag van een Midi-byte zoals gebruikt in de midi-uitstuurfunctie U()

B0,B1,B2,B3,B4,B5,B6,B7

Deze variabelen worden gebruikt voor de opslag van de registratiegevens voor de Proteus synthesizer. Zij worden

gebruikt in de P functie.

Alle hiervoor opgesomde variabelen worden bepaald als gehele getallen:

1 INTEGER A,B,B0,B1,B2,B3,B4,B5,B6,B7,C,D,E,I,J,L,M,P,S,T,U,X,
Y,Z,XH,YH,ZH,XF,YF,ZF

N(7)

Deze gedimensioneerde variabele wordt gebruikt om de namen van de via MIDI uit te sturen noten op de kanalen 0 tot 7 op te slaan.

V(7)

Deze gedimensioneerde variabele wordt gebruikt om de aanslagsterkte van de via midi te spelen noten op de kanalen 0-7 op te slaan.

NO(7)

Deze gedimensioneerde variabele wordt gebruikt om de waarde van de vorige noot op het overeenkomstige kanaal op te slaan en vast te houden.

Ook alle gedimensioneerde variabelen worden als gehele getallen gedefinieerd:

2 INTEGER N(7),V(7),NO(7)

3.2.2.2.-Initialisatie van hardware en software

Het eerste wat nu dient te gebeuren is het toewijzen van ASCII-waarden aan de 16 schakelaars van het matrix-toetsenbord. We kozen deze waarden zo dat wanneer de hexadecimale konstante \$30 ervan afgetrokken wordt, we het decimaal equivalent van de eerste 16 hexadecimale getallen (\$0-\$F) overhouden. MT-Basic heeft een ingebouwde functie KEYPAD waarmee de toetsenmatrix uitgelezen kan worden. Wanneer de KEYPAD functie een 0 waarde oplevert, werd er geen enkele toets ingedrukt. In het omgekeerde geval levert deze functie ons de ASCII waarde op van de toets zoals we haar tijdens de initialisatie middels KEYSTR definieerden. De KEYPAD functie maakt gebruik van een hardware interrupt. Het is dan ook van groot belang deze interrupt niet buiten werking te stellen, wat via software hier wel mogelijk is.

Een opmerking wat dit betreft: wanneer een toets ingedrukt wordt gehouden, dan valt het hele processorgebeuren voor die tijd stil en herbegint pas wanneer de toets terug wordt losgelaten.

De MIDI-poort dient eveneens via software geïnitieerd te worden, aangezien deze poort hardwarematig gebruik maakt van een programmeerbare PIO chip, de 8255. De werkingsmodus wordt ingesteld via I/O adres 195. De voor ons gewenste configuratie is als volgt:

- Poort A - als uitgang voor de midi-databytes
- Poort B - als uitgang voor het Strobe-sigitaal
- Poort C - als ingang voor het lezen van de toestand

cfr. INTEL-databoek m.b.t. de 82C55-chip voor details met betrekking tot de programmatie van deze chip.

De volgende stap in de initialisatie is het invullen van de opzoekingstabellen waarvan het programma gebruik maakt. Daarna volgen nog enkele utilitaire instructies, o.m. een om de toevalsgenerator te herbronnen, en vervolgens instructies voor het juist zetten van de datum en het uur van de 'real-time' klok.

Deze vijf stappen worden uitgevoerd door volgende zin:

```
3 KEYSTR "0123456789:;<=>?"
  OUT $C3,$81 : OUT $C1,1 : OUT $C0,$FF
  GOSUB 900
  RANDOMIZE
  STIME 00,00,00,2:
  SDATE 01,01,93
```

De toevalsnummegerator wordt alleen gebruikt voor de panning van het geluid in de module 'Open'. Het juist zetten van de datum en het uur is in principe voor dit programma van geen enkel belang... Op een punt na: in geval de daggrens overschreden wordt, begint de SYSTIC instructie die van de tellerstand van de real-time klok afhankelijk is, terug van 0 af te tellen. Zou dit nu gebeuren in de loop van een muzikale uitvoering dan zou dit aanleiding kunnen geven tot een schijnbaar 'hangende' komputer. Er wordt immers doorlopend met tijdsverschillen gewerkt. Om dit euvel te vermijden, stellen we de klok bij het begin van het stuk in op het uur 00h00'00".

3.2.2.3.- Definitie van de midi-functie

Volgende functie, luisterend naar de naam U, (memo-tip: Uit) omvat al het nodige om ons midi-interface aan de praat te krijgen. De over te dragen parameter is B, het te verzenden midi-byte (7-of 8 bits). (Cfr. sub. 3.2.1.5)

```
4 DEF U(B)
5 IF B<$80 THEN OUT $C1,0
6 IF B>$7F THEN OUT $C1,1: B=B-$80
7 IF BAND(INP($C2),2) THEN GOTO 7
8 OUT $C0,B+$80
  OUT $C0,B
  OUT $C0,B+$80
  OUT $C0,$FF
9 FNEED
```

3.2.2.4.- Initialisatie van de Midi-poort en het LCD scherm

De initialisatie van de midi-poort maakt gebruik van de zopas gedefinieerde functie. Voor de initialisatie van het LCD-schermpje is een kommando voorzien in MT-basic: INITLCD.

De betekenis van de parameters bij het INITLCD kommando is als volgt: Het eerste cijfer dient 0 te zijn bij gebruik van een display dat lettertekens heeft die opgebouwd zijn uit een matrix van 5 op 7 punten. Het tweede cijfer bepaalt hoe de looper zich op het scherm zal gedragen: 0= meegaand met de verschijnende lettertekens, 1= terplaatse blijvend terwijl de tekst naar rechts doorschuift, 2= zoals op een gewoon computerscherm- de tekst schuift naar links door. Voor het derde cijfer gelden dan volgende betekenissen: 0-3: scherm onleesbaar, 4= looper onzichtbaar, geen geknipper, 5= geen looper, wel geknipper, 6=wel een looper geen geknipper, 7=loper zichtbaar en knipperend. De initialisatie instructie INITLCD plaatst tevens de looper op positie 1, wat ook mogelijk is via het kommando OUTLCD \$80. Er zijn slechts 16 plaatsen voor letters op het schermpje dat wij dit instrument meegaven. De posities lopen van \$80 tot \$8F. Het kommando OUTLCD 1 wordt gebruikt om het scherm te wissen.

```
10 D = U($FF)
11 INITLCD 0,0,5
    OUTLCD 1
```

```
12 STIME 0,0,0,2
    SDATE 01,01,93
```

Het instellen van tijd en datum is op zich voor dit programma van geen enkel belang. We namen deze kode alleen op omdat de overgang van tijd 24u00 naar 0u00 een interrupt genereert die ook de verder vaak gebruikte SYSTIC instructie in de war stuurt. Het getal dat door SYSTIC wordt geretourneerd is een 32 bit waarde in 'two's complement'. Bij de dagovergang springt deze teller van positief naar negatief, waardoor de gebruikte delta-t waarden een fout opleveren.

3.2.2.5.- Definitie van de synthesizerfuncties

De hier gedefinieerde functies worden gebruikt om de registratie van de Proteus-synthesizer module te programmeren. Elke registratie kan (in deze implementatie) uit 8 verschillende stemmen bestaan. (Het maximum voor dit type synthesizer is echter 16 en mits wat kunstgrepen zelfs 32).

Voor een goed begrip van de uitgestuurde hexadecimale codes, dient het technisch handboek van de Proteus synthesizer gekonsulteed te worden, uiteraard na kennisname van de Midi-specificaties zelf. (cfr. bibliografie).

Terloops weze opgemerkt dat dit Basic de prefix \$ gebruikt voor de aanduiding van hexadecimale getallen, waar andere Basic-versies eerder &H gebruiken en het \$ teken reserveren voor strings.

```
20 DEF P(B0,B1,B2,B3,B4,B5,B6,B7)
```

Selekteer de presets via de overgedragen variabelen:

```
21 D = U($C0)+U(B0)+U($C1)+U(B1)+U($C2)+U(B2)+U($C3)+U(B3)
```

```
22 D = U($C4)+U(B4)+U($C5)+U(B5)+U($C6)+U(B6)+U($C7)+U(B7)
```

Reset de pitch-bend controllers en stel de master volume regelaars op maximum: (= &7F)

```
24 FOR M = 0 TO 7
```

```
    D = U($E0+M) + U(0) + U(64) + U($B0+M) + U(7) + U($7F)
```

```
NEXT M
```

Nadat de registratie ingesteld is, programmeren we eveneens de ruimtelijke panning van de muzikale output. 0 komt overeen met volledig links in het stereo-beeld, \$7F met volledig rechts, en 64 is dan het midden van het stereobeeld.

```
26 D = U($B0) + U($A) + U(0) + U($B1) + U($A) + U(64) + U($B2) +
    + U($A) + U($7F) + U($B3) + U($A) + U(20)
```

```
27    D = U($B4) + U($A) + U(64) + U($B5) + U($A) + U(107) +
    + U($B6) + U($A) + U(47) + U($B7) + U($A) + U(80)
```

```
29 FNEND
```

De nu volgende midi-synthesizer functie wordt gebruikt om enkele muzikale parameters te kunnen wijzigen. We doopten deze functie UPAR. Zij verloopt in twee stappen:

1.- stel de synthesizer in op de registerinstelling waarbinnen parameters veranderd dienen te worden. Dit verloopt middels een 'channel-program-change message'.

2. stuur het parameterwijzigingskommando uit via midi-kanaal 0

```
30 DEF UPAR(B1,B4,B2,B3)
```

De variabele B1 bevat het midi-kanaal, terwijl B4 het volgnummer van de gewenste voorinstelling (preset) bevat. (0-127)

```
31 D=U($C0+B1)+U(B4)
```

De variabele B2 bevat het nummer van de gewenste parameter, B3 de data. Voor deze instellingen zijn 14-bit getallen nodig, die we dus eerst dienen te splitsen in MSB en LSB als volgt:

B2 te splitsen in MSB=B4 and LSB=B5

B3 te splitsen in MSB=B6 and LSB=B7

```
32 B4 = (B2 / $80)
```

```
    B5 = B2 - (B4 * $80)
```

```
    B6 = (B3 / $80)
```

```
    B7 = B3 - (B6 * $80)
```

```
33 D = U($F0) + U($18) + U(4) + U(0) + U(3) + U(B5) + U(B4) +
    + U(B7) + U(B6) + U($F7)
```

```
34 FNEND
```

3.2.2.6.: - Kommando voor tijd en ritme

Deze funkties werden geschreven ter vervanging van de binnen MT-Basic voorziene kommandos SUSPEND and WAIT. Het oplossend vermogen van de SYSTIC funktie is een honderdste van een sekonde.

```
40 DEF S(B7)
```

```
41 T = SYSTIC
```

```
42 IF SYSTIC < (T + B7) THEN GOTO 42
```

```
43 FNEND
```

3.2.2.7.: - Opstellingstest

Volgende programmaregels worden gebruikt om een optimale opstelling van de hardware, meer bepaald de precieze plaatsing van de zender en de drie ontvangers, mogelijk te maken, vooral in funktie van de ruimteakoestiek van de plaats waar het instrument wordt opgesteld voor gebruik. Via deze regels worden de waarden van de analoog-digitaal konverter uitgestuurd via de 9600 Baud RS232 verbinding naar het scherm van de host-komputer of terminal. Deze laatste is dus alleen in dit geval ook werkelijk nodig. De opstellings- en afregelingsroutine kan worden gestopt door eender welke toets op het matrix-toetsenbord in te drukken.

```
44 FOR I=0 TO 6
```

```
    PRINT ADC(I); "    ";
```

```
    NEXT I
```

```
45 D = S(50)
```

```
    PRINT " "
```

```
46 IF KEYPAD = 0 THEN GOTO 44
```


3.2.2.8.- Keuze-menu voor de verschillende modules

In de volgende programmaregels wordt een vrije keuze van de te implementeren mappings door de gebruiker mogelijk gemaakt. Voor de teatrale voorstelling van de "A Book of Moves" implementatie van ons instrument ligt deze volgorde weliswaar vast als kompositorisch gegeven, maar dit werd niet vastgelegd op het nivo van de hardware of de software.

De selectie van de geïmplementeerde algoritmen en mappings gebeurt via het matrix-toetsenbord en is niet bepaald erg gebruikersvriendelijk. Alleen hexadecimale getallen kunnen immers ingebracht worden. Omdat we geen 'enter' toets voorzagen, is bovendien een foutief ingetoetst cijfer, niet te corrigeren. Gekoppeld aan de keuze van een module in dit programma, impliceert een bepaalde keuze ook een bepaalde instelling van de synthesizer. Het is in deze implementatie dus niet mogelijk -althans niet zonder kleine wijzigingen in dit programma aan te brengen- de instrumentatie los te koppelen van de kompositorische algoritmieken.

De eerste regel maakt de keyboard-buffer leeg:

```
50 E = KEYPAD
   IF KEYPAD >< 0 THEN GOTO 50
```

Vervolgens schakelen we het schermje aan en wissen wat er eventueel nog op te zien was.

We vragen de gebruiker zijn keuzecijfer in te brengen:

```
51 DEVICE 6
   OUTLCD 1
   PRINT "Move? ";
```

Wacht nu tot er een toets ingedrukt wordt:

```
52 E = KEYPAD
   IF E = 0 THEN GOTO 52
```

Bereken nu de waarde van de variabele E (cfr. sub.3.2.2.2.).

Druk de waarde van de keuze af op het scherm.

Stuur eventuele output van het board terug naar de terminal uitgang.

```
53 E = E - $30
   OUTLCD $80
   PRINT "Move= ";E
   DEVICE 0
54 IF E = 0 THEN STOP
```

Op grond van de gemaakte keuze wordt nu de synthesizer ingesteld en springt het programma -via goto's- naar de betreffende programmamodules. Omdat de regelnummers niet direkt de leesbaarheid van de code bevorderen, hebben we de namen van de modules als labels toegevoegd:

Open:

55 IF E = 1 THEN D = P(26,26,26,26,26,26,26,26)
GOTO 70

Topoi:

56 IF E = 2 THEN D=P(5,35,19,25,32,45,79,65)
GOTO 80

Minor:

57 IF E = 3 THEN D = P(10,10,10,10,10,10,10,10)
GOTO 120

Beat:

58 IF E = 4 THEN D = P(8,18,75,89,52,58,63,67)
GOTO 140

Rising:

59 IF E = 5 THEN D = P(12,10,6,16,66,36,40,60)
GOTO 160

Sforte:

60 IF E = 6 THEN D = P(76,76,76,76,76,76,76,76)
GOTO 185

Lead:

61 IF E = 7 THEN D = P(60,61,15,35,50,45,125,56)
GOTO 200

Canvas:

62 IF E = 8 THEN D = P(57,74,93,126,84,9,14,106)
GOTO 220

Close:

63 IF E = 9 THEN D = P(6,10,12,16,26,36,40,47)
GOTO 250

Prime-time:

64 IF E = 10 THEN D = P(85,52,50,44,54,63,32,85)
GOTO 280

Call:

65 IF E = 11 THEN D = P(11,11,11,11,11,11,11,11)
GOTO 300

Spooky:

66 IF E = 12 THEN D = P(9,29,4,9,9,9,9,9)
GOTO 350

Rec-Play:

67 IF E = 13 THEN D = P(55,55,55,88,55,55,55,55)
GOTO 400

Lock-Unlock:

68 IF E = 14 THEN D = P(19,13,58,102,112,92,67,67)
GOTO 450

Hammers:

69 IF E = 15 THEN D = P(36,36,36,22,115,50,8,8)
GOTO 475

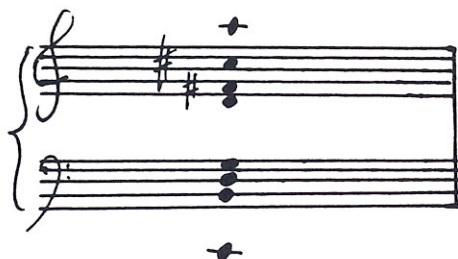
3.2.2.9.: De modules van 'A Book of Moves'

De hierna volgende stukjes programma vormen elk voor zich een verschillende mogelijkheid om het instrument te programmeren. Zij leggen, afhankelijk van het geval tevens in meer of mindere mate kompositorische en esthetische samenhangen vast tussen de sonore gebeurtenissen onderling. De controle gaat in elk geval steeds uit van de bewegende uitvoerder. De mate waarin het muzikale resultaat door de bespeler beïnvloed of bepaald kan worden is voor elke module verschillend.

In deze modules wordt uitgebreid gebruik gemaakt van de multi-tasking mogelijkheden van dit computersysteem. De kode voor de taken (tasks) die in de modules gebruikt wordt vormt het onderwerp van een afzonderlijke paragraaf.

3.2.2.9.1.: " Open "

In deze module wordt de bewegingsinput tot een strikt minimum beperkt. Het algoritme speelt een enkel breed en complex akkoord bestaande uit 8 noten:



De uitvoerings-instructie is teatraal en retorisch gedacht: bij het begin van de compositie 'A Book of Moves' zit de uitvoerder samengeballd in het centrum van de denkbeeldige tetraeder gevormd door de hoekpunten van het Holosound-systeem. De achtklank wordt alsmaar opnieuw aangeslagen.

Pas na een tijdje gaat de uitvoerder bewegen. Telkens wanneer hij beweegt, stopt het akkoord. Wanneer hij niet meer beweegt, herbegint de dreun. Echter, bij elke herhaling van dit gebeuren, verstillt het akkoord een stap. De beweger dient evenwel alsmaar bewegingsenergie te investeren om het geluid nog tot zwijgen te brengen. De beweging gaat dus in een diskontinu crescendo, terwijl de klank in decrescendo gaat. De module eindigt wanneer de uitvoerder het akkoord definitief het zwijgen heeft opgelegd.

Het instrument wordt hier dus in hoofdzaak aangewend als een 'trigger' systeem via de bewegingsdetektie. Technisch gezien dus eigenlijk bijzonder triviaal, echter volkomen ingegeven door overwegingen van kompositorische en teatraal-retorische aard. Het richt de aandacht van het publiek naar het allereenvoudigste kwazi binaire verband tussen klank en beweging.

J stelt het geluidsvolume bij het begin in op maximum.
 I stelt de gevoeligheid van de bewegingsdetectie bij de
 aanvang van de module in.
 De START instructie vraagt de multitasker om taak 1 met
 een periodiciteit van 200ms uit te voeren. De
 bewegingsparameters worden dus vijfmaal per seconde
 ververst.

```
70 J=$7F
    I=3
    START 1,20
```

Ingangspunt van de programmalus:

De lus speelt 8 noten na elkaar, wat gezien de
 uitvoeringssnelheid klinkt als een akkoord. De waarden
 van de noten worden gelezen uit een opzoekingstabel,
 vastgelegd aan het eind van het programma, (Een 'look-
 up table') en overgebracht naar de gedimensioneerde
 variabele N(M) gebruikt in de eerder gedefinieerde
 midi-functies.

Voor elke afzonderlijke noot uit het akkoord wordt een
 verschillend midi-kanal gebruikt. De panning geeft aan
 het klinkend resultaat een frenetiek karakter: immers
 zij wisselt na elke herhaling gestuurd door een
 toevalsfunctie.

```
71 FOR M = 0 TO 7
    D = U($B0+M) + U(10) + U(64 + (RND / 512))
    D = U($90+M) + U(PEEK($DF00 + M)) + U(J)
  NEXT M
  U(BXOR(BOR(RND,$FF80),$FF80))
```

Gebruik nu de gedefinieerde tijdfunctie voor de
 vastlegging van het speeltempo:

```
72 D = S(50)
```

Wanneer de som van de gedetekteerde bewegingsamplitudes
 in de drie kanalen kleiner is dan de ingestelde
 gevoeligheid (I), herhaal het akkoord:

```
73 IF X + Y + Z < I THEN GOTO 71
```

Verminder bij elke doorgang van de lus het volume (J)
 met 5-midi eenheden. Laat de waarde van J echter niet
 negatief worden:

```
74 J = J - 5
    IF J <= 0 THEN J = 0
    GOTO 78
```

Indien de gedetekteerde bewegingshoeveelheid groter was
 dan de ingestelde gevoeligheid, verminder ook deze
 gevoeligheid (I wordt in stapjes van 2 eenheden
 vergroot). Dit maakt het voor de speler alsmear
 lastiger om het geluid te doen stoppen.

Gebruik de tijdsfunctie om een rust van minstens een
 seconde in te brengen:

```
75 I = I + 2: D = S(100)
```

Wanneer de bewegingsamplitude de gevoeligheid overtreft, worden geen verdere akkoorden tot klinken gebracht:

```
76 IF X + Y + Z > I THEN GOTO 76
```

Het algoritme kan ook onderbroken worden door een willekeurige toets op het toetsenbordje in te drukken. Dit is voorzien voor het geval dat de speler er niet zou in slagen het akkoord tot stilte te brengen...

```
77 IF KEYPAD = 0 THEN GOTO 71
```

Schakel taak 1 voor de multitasker weer uit (Cancel), en spring weer naar het begin van de lus:

```
78 CANCEL 1  
GOTO 50
```

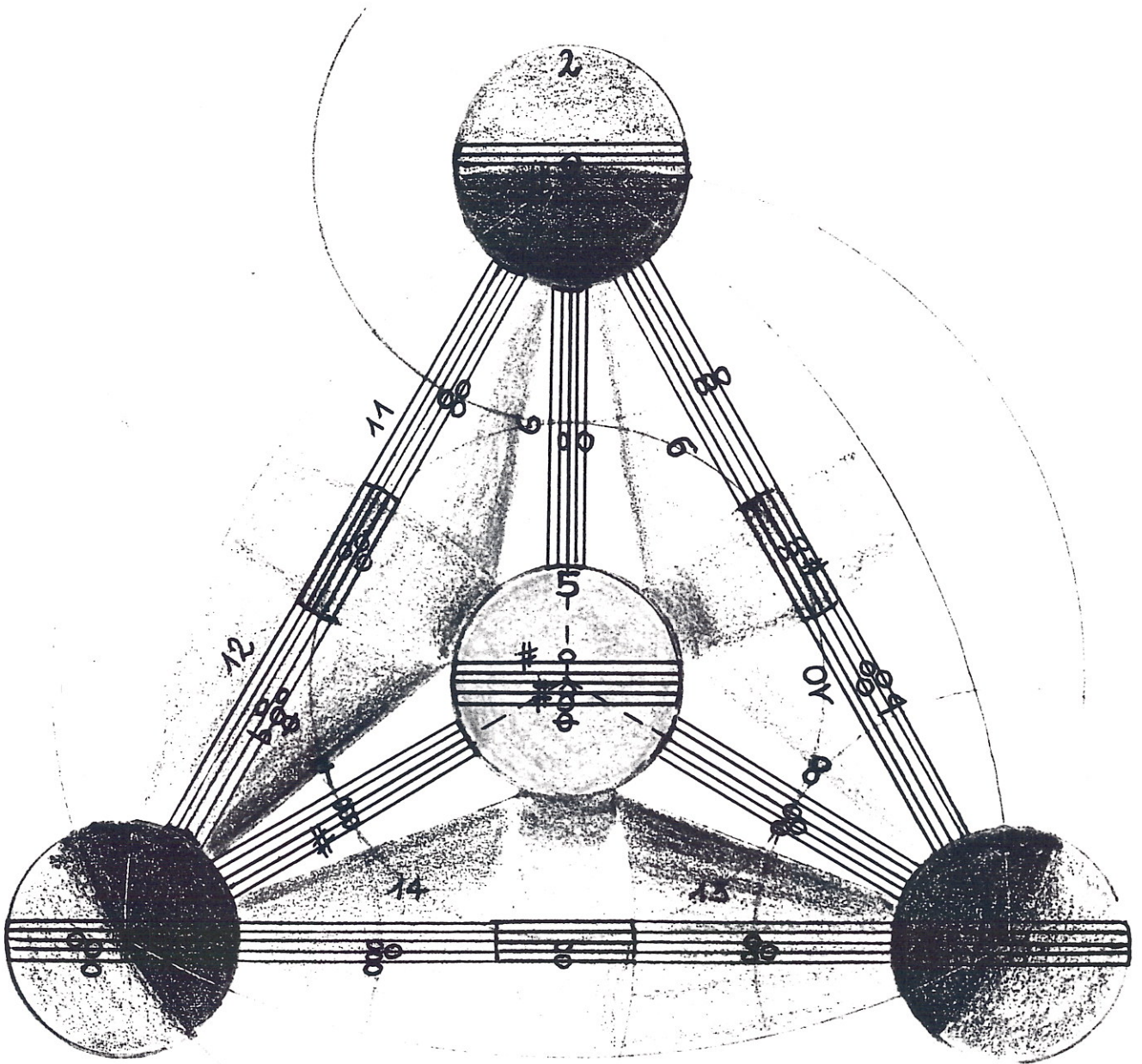
3.2.2.9.2.: " Topoi "

Dit algoritme vormt een goede demonstratie van de mogelijkheid van ons systeem om via bewegingsdetectie aan plaatsbepaling te doen. De kode probeert om aan de hand van de informatie van de analoge komputer, te achterhalen waar de speler zich binnen het veld bevindt wanneer hij beweegt. Men zou een soort topografische analyse van de data kunnen noemen. het

Op grond van de berekende plaats waar de speler zich bevindt, worden dan andere akkoordsamenstellingen tot klinken gebracht. Deze akkoorden zijn ook hier weer op voorhand ondergebracht in een 'look-up' tabel. Hun samenstelling en onderling harmonisch verband behoort tot het strikt kompositorisch aspekt van het stuk. Wel werden de akkoorden zo gekozen, dat de mapping van de bewegingsplaatsen op hun al dan niet klinken, een spelen van melodische en harmonische muzikale zinnen mogelijk maakt.

De eerste analoge komputer dient wel ingesteld te worden voor de ring-vermenigvuldiger patch. (Onderlinge vermenigvuldiging der signalen). Anders klopt de onderliggende rekenkundige analyse niet meer. De software in deze module laat toe 15 verschillende plaatsen te onderscheiden. De achterliggende logika laat zich wellicht het best aan de hand van een tekening illustreren. Wel moet men zich hierbij voor ogen houden dat de in de tekening uiteraard scherp getekende grenslijnen en ingekeurde vlakafbakeningen, in de praktijk niet zo heel precies te bepalen zijn. Een en ander hangt sterk af van de akoestische eigenschappen van de ruimte waarbinnen wordt gespeeld. Korrekties zijn overigens wel aan te brengen via enkele regelaars in het analoog-komputergedeelte. Een opstellingshulproutine werd overigens ook voorzien op het nivo van deze software.

De module maakt gebruik van gegevens verkregen via taak 18, (zie verder) die tienmaal per seconde wordt aanroepen. De analyse van de bewegingsinformatie maakt gebruik van de vektoriele bewegingssnelheden XF,YF,ZF.



TOPOI

De taak wordt tienmaal per seconde in de agenda van de multitasker opgenomen:

80 START 18,10

Ingangspunt van de programmalus:

Wanneer niets verandert in vergelijking met de voordien waargenomen bewegingssituatie, gebeurt er ook in de muziek niets:

81 IF (XF = XH) AND (YF = YH) AND (ZF = ZH) THEN GOTO 81

Ook wanneer de amplitudes 0 zijn, gebeurt niets. Dit is niet helemaal redundant, omdat de frekwentiedata iets trager reageren op de werkelijkheid dan die van de amplitude.

82 IF BOR(BOR(X,Y),Z) = 0 THEN GOTO 81

Hier volgt dan de analyse van de plaats van de beweger: Wanneer geen beweging gedetekteerd wordt zetten we de case-variabele gelijk aan 1.

83 IF A = 0 THEN C = 1
GOTO 110

Een beweging in het middelpunt van de denkbeeldige tetraeder, geeft aanleiding tot geval 2:

84 IF BOR(YF,ZF) = 0 THEN C = 2
GOTO 99

Beweging in de linker-hoek:

85 IF BOR(XF,YF) = 0 THEN C = 3
GOTO 99

Beweging in de rechterhoek:

86 IF BOR(XF,ZF) = 0 THEN C = 4
GOTO 99

Beweging die als identiek gezien wordt vanuit de drie vektoren. Dit is het geval wanneer de speler zich in het centrum van de opstelling 'opblaast' of 'ineenkrimpt'.

87 IF BAND(XF,YF) = ZH THEN C = 5
GOTO 99

Beweging op de centrale as:

88 IF YF = ZF THEN C = 6
GOTO 99

Beweging op de linker X-Z as:

89 IF XF=YF THEN C=7: GOTO 99

Beweging op de rechter Z-Y as:

```
90 IF XF=ZF THEN C=8: GOTO 99
```

Bepaling van de tusseninliggende overgangsposities:

```
91 IF (XF < YF AND XF < ZF) AND (YF < ZF) THEN C = 9
    GOTO 99
92 IF (XF < YF AND XF < ZF) AND (YF > ZF) THEN C = 10
    GOTO 99
93 IF (YF < XF AND YF < ZF) AND (XF < ZF) THEN C = 11
    GOTO 99
94 IF (YF < XF AND YF < ZF) AND (XF > ZF) THEN C = 12
    GOTO 99
95 IF (ZF < XF AND ZF < YF) AND (XF < YF) THEN C = 13
    GOTO 99
96 IF (ZF < XF AND ZF < YF) AND (XF > YF) THEN C = 14
    GOTO 99
```

Indien geen van de vorige gevallen gevonden werd, voert het programma de analyse opnieuw uit:

```
97 GOTO 81
```

De C-variabele is nu in elk geval bepaald. Op grond hiervan zal de mapping van de plaats op de muzikale samenklanken gebeuren. De look-up tabel start op geheugenadres \$DF00:

```
99 FOR M = 0 TO 7
```

Lees per kanaal de van het geval afhankelijke noot, en wanneer geen noot wordt opgegeven, zoek dan de opgegeven noot voor de volgende muzikale stem of midi-kanaal.

```
100 N(M) = PEEK($DF00 + M + (C * 8))
```

Indien een noot in de tabel werd gevonden, transponeer haar dan naar een bruikbare oktaaflijging en transpositie. De konstante 36 schuift de noten 3 oktaven naar omhoog. Vanaf deze ligging hebben we voorts de uiteindelijke oktaaflijging afhankelijk gemaakt van de globale bewegingsamplitude door bij de numerieke waarde voor de noot ($12*(A/30)$) op te tellen.

```
101 IF N(M) THEN N(M)=N(M)+36+(12*(A/30))
```

Wanneer de beweging gelokaliseerd werd in 'overgangszones', en wanneer er geen noot in de tabel gegeven wordt, laat de te spelen noot afhangen van de vektoriele bewegingssnelheid op het moment nu:

```
102 IF (C > 8 AND M = 0) AND (N(M) = 0) THEN N(M) = 60 + XF
103 IF (C > 8 AND M = 1) AND (N(M) = 0) THEN N(M) = 60 + YF
104 IF (C > 8 AND M = 2) AND (N(M) = 0) THEN N(M) = 60 + ZF
```

Wanneer een noot lager zou klinken dan MIDI-waarde 36, schrap ze:

```
105 IF N(M) < 36 THEN N(M) = 0
    V(M) = 0
```

Wanneer er een noot staat, geef haar dan een geluidssterkte afgeleid van de momentane bewegingsparameters:

```
106 IF N(M) THEN V(M) = A
```

Indien de gevonden noten verschillend zijn van de voordien in dezelfde stem gespeelde, schakel dan de vorige noot uit en de nieuwe aan:

```
107 IF N(M) >< NO(M) THEN
    D = U($80+M) + U(NO(M)) + U(0) + U($90+M) + U(N(M))+U(V(M))
```

Sla de zopas gespeelde noten op in de gedimensioneerde variabele NO():

```
108 NO(M) = N(M)
```

Sluit de kanaalteller-lus:

```
109 NEXT M
```

Merk op dat in deze module nergens een tijd- of ritme instructie voorkomt. Deze muzikale parameters volgen hier vooral op grond van de opgeven periodiciteitsparameter bij het opstarten van de gebruikte taak.

Sla ook de vorige bewegingstoestandsparemeters op in variabelen:

```
110 XH = XF
    YH = YF
    ZH = ZF
```

Ga na of er zich een exit-koniditie voordoet (wanneer er een toets gedrukt wordt op het toetsenbordje). Zoniet, ga terug naar het begin van de lus.

```
111 IF KEYPAD = 0 THEN GOTO 81
```

Indien er wel een toets werd ingedrukt, beëindig deze module en ga terug naar het hoofdmenu.

```
112 CANCEL 18
    GOTO 50
```

3.2.2.9.3.: " Minor"

Deze module is een software implementatie van een onzichtbare piano. De via de bewegingen gespeelde noten cirkelen centripetaal rond een a priori vastgelegd tonaal centrum. De gebruikte noten maken gebruik van een kleine tertstoonladder. De oktaafpositie waarin de noten worden gespeeld is afhankelijk gemaakt van de globale bewegingshoeveelheid.

De intervallische afwijking tegenover het tonaal centrum is afhankelijk gemaakt van de vektoriele bewegingssnelheden. (De XF,YF,ZF parameters). Wanneer de uitvoerder redelijk symmetrisch beweegt ten opzichte van de drie ontvangers op de hoekpunten van de denkbeeldige tetraeder, klinkt de noot die overeenstemt met het tonaal centrum van de muziek via midi-kanal 7.

De intervallen ontleend aan de mineur toonladder zijn ingeschreven in een look-up tabel in het geheugenadresgebied dat uitgelezen wordt met PEEK(\$DFF0) tot PEEK(\$DFFF). Het tessituurbereik is twee oktaven en een kleine terts.

Muzikaal gesproken klinkt deze module eerder 'romantisch'.

Taak 1, de enige die in multitasking gebruikt wordt in deze module, levert de bewegingsparameters in de variabelen:

XF,YF,ZF	bereik 0 -127 (7 bits)
A	bereik 0 -127
X,Y,Z	bereik 0 -127

Reset alle variabelen en start taak 1 tienmaal per seconde:

```
120 XF = 0
    YF = 0
    ZF = 0
    START 1,10
```

Ingangspunt van de programmalus:

```
121 IF A = 0 THEN GOTO 121
```

De variabele I in volgende formule bepaalt de oktaafligging van de klinken noten. Het bereik van I loopt van 24 tot 72. De oktaafpositie zal afhangen van de bewegingssnelheden. Een konstante bepaalt een minimum waarde voor de geluidsterkte, om de dynamiek in deze rustig klinkende module niet te groot te maken:

```
122 I = 36 + (12 * ((XF + YF + ZF) / 92))
```

Wanneer de bewegingsamplitudes in de X,Y,Z vektoren groter zijn dan nul, voer dan de mapping uit van de bewegingssnelheden naar de muzieknoten:

```
123 IF X THEN D = U($90) + U(I+PEEK($DFF0+(XF/8))) + U(X)
124 IF Z THEN D = U($91) + U(I+PEEK($DFF0+(ZF/8))) + U(Z)
125 IF Y THEN D = U($92) + U(I+PEEK($DFF0+(YF/8))) + U(Y)
```

Als de speler zich in het centrum van de tetraeder bevindt, speel dan ook de noot die het tonaal centrum vormt. De konditie met ZF werd toegevoegd om deze noot toch niet teveel te laten klinken.

```

126 IF ZF > 24 AND BAND(XF,YF) = ZF THEN
                                D = U($97) + U(I) + U(A / 2)
    GOTO 130

```

Wanneer hij zich ook helemaal in het centrum bevindt, speel dan ook het onderoktaaf van het tonaal centrum:

```

127 IF YF = ZF THEN
    D = U($93) + U(I+12) + U(A/2) + U($96) + U(I-12) + U(A/2)

```

Beweegt hij links van het centrum, voeg dan aan het tonaal centrum de bovenkwint toe:

```

128 IF XF = YF THEN D = U($94) + U(I+7) + U(A)

```

Beweegt hij rechts, speel dan een kwint omlaag:

```

129 IF XF = ZF THEN D = U($95) + U(I - 7) + U(A)

```

De ritmische structuur is afhankelijk gemaakt van de bewegingsamplitude: hoe groter deze wordt, hoe sneller de noten elkaar zullen opvolgen:

```

130 D = S(46 -(A / 3))

```

Test de exit-konditie:

```

131 IF KEYPAD = 0 THEN GOTO 121

```

Indien die konditie is voldaan, schakel taak 1 uit en spring terug naar het hoofdmenu.

```

132 CANCEL 1
    GOTO 50

```

3.2.2.9.4.: " Beat "

In deze algoritmische module worden de bewegingsparameters gemapt op een denkbeeldig uitgebreid slagwerkinstrumentarium. Om dit effectief te laten zijn moet het uiterst responsief zijn. De spel-instructie schrijft bruuske karate-achtige bewegingen voor: kort en met een duidelijk gemarkeerd begin en einde. Geen enkele 'elegantie' is hier toegestaan. Zoniet onstaat er sonore rommel.

Start taak 2 met een interval van 80ms:

140 START 2,8

De slaginstrumenten worden konditioneel gespeeld:

141 IF C = 0 THEN GOTO 155

Links in het stereobeeld komt een eerste slagwerkgroep:

142 IF XF AND X THEN D = U(\$90) + U(XF * 2) + U(\$7F)

Rechts plaatsen we de tabla:

143 IF YF AND Y THEN D = U(\$92) + U(YF * 2) + U(\$7F)

Centraal staat een tweede slagwerkensemble:

144 IF ZF AND Z THEN D=U(\$91)+U(ZF*2)+U(\$7F)

Een bijzondere konditie bepaalt wanneer de Shinto-gong gespeeld wordt:

145 IF X + Y + Z > 36 AND XF + YF + ZF > 90 THEN
D = U(\$96) + U((XF + YF + ZF + 24) / 2) + U(A)
GOTO 150

Rammelaars en Japanse bellen samen:

146 IF X AND Y THEN
D = U(\$93) + U(XF+YF) + U(X) + U(\$94) + U(YF*2) + U(Y)

Rammelaars en gestemde glissando-trommels:

147 IF X AND Z THEN
D = U(\$93) + U(XF + ZF) + U(X) + U(\$95) + U(ZF * 2) + U(Z)

Japanse bellen en glissando-trommels samen:

148 IF Y AND Z THEN
D = U(\$94) + U(YF + ZF) + U(Y) + U(\$95) + U(ZF * 2) + U(Z)

Laat de grote Miya Daiko trommel klinken:

150 IF (X + Y + Z) > 120 THEN
D = U(\$97) + U((X + Y + Z) / 3) + U(\$7F)

De volgende instructie maakt de tijdsduur van de gesampelde slaggeluiden afhankelijk van de piek-amplitude van de geregistreeerde beweging:

```
151 D = S(1 + L)
```

Voor het merendeel der gebruikte slaginstrumenten is het niet nodig de aangeslagen noten ook uit te schakelen. Ze sterven vanzelf uit. Sommige instrumenten moeten terwille van de muzikale transparantie, toch gedempt worden:

```
154 D=U($B4)+U($7B)+U(0)+U($B5)+U($7B)+U(0)+U($B6)+U($7B)+U(0)
```

Test de exit-konditie. Indien voldaan, schakel taak 2 uit en ga terug naar het hoofdmenu:

```
155 IF KEYPAD = 0 THEN GOTO 141
156 CANCEL 2
    GOTO 50
```

Deze implementatie staat of valt met de gebruikte synthesizer evenals met de instrumentatie waarmee de module wordt aanroepen. De Proteus-synthesizer waarvoor ze gedacht is, beschikt over een ruim arsenaal aan ingebouwde 16-bit samples van etnische slaginstrumenten. Deze samples werden in ruime mate op voorhand 'ge-edit' in functie van hun gebruik in deze module. Deze editing is evenwel niet mogelijk binnen het opzet van het hier voorgestelde programma. Daarvoor schreven we een afzonderlijk programma in Microsoft PDF-Basic dat evenwel buiten het opzet van deze studie valt. De naamgeving der instrumenten is overgenomen uit de handleiding bij de betreffende synthesizermodule.

3.2.2.9.5.: " Rising "

Dit algoritme bestaat uit een vanuit de bewegingsinformatie gestuurde langzame en extreme opwaartse akkoordprogressie. De bewegingsparameters worden hier o.m. gebruikt als informatie voor de toonwielkontrolle (PMD, pitch-modulation device) van de gebruikte synthesizer. Hierdoor kan de toonhoogte na de aanslag van elk afzonderlijk akkoord vanuit kleine bewegingen gemoduleerd en gebogen worden, wat een dramatisch en spannend effect geeft.

De interaktiviteit is in dit algoritme gesteund op een eenvoudige topologische analyse van de bewegingsvektoren. Wanneer de ruimtelijke plaats zich wat de beweging betreft herhaalt of konstant blijft, worden de toonbuigingen waarvan we spraken uitgevoerd. Vanzodra de uitvoerder op een iets andere plek bewegingen uitvoert, zal de akkoordprogressie een stapje omhoog klimmen.

Taak 1 levert ons hier volgende parameters:

A bereik 0 -127
X,Y,Z bereik 0 -127
XF,YF,ZF worden niet gebruikt.

Reset de teller voor de progressie. Start taak 3 om de 30ms. Reset de vorig-geval variabele op 1:

```
160 J = 0
    I = 1
    START 1,3
```

Ingangspunt van de programmalus:

```
161 IF X + Y + Z = 0 THEN C = 1
    GOTO 170
```

Wanneer de beweging in het centrum plaatsvindt, verhoog de J teller met een waarde evenredig met de amplitude van de gedetekteerde beweging. Hierdoor wordt de stijgende progressie mogelijk gemaakt:

```
162 IF A AND A / 3 = (X + Y + Z) / 9 THEN C = 15
    J = J + 1 + (A / 3)
    GOTO 170
```

Voer de plaatsanalyses uit en bepaal een waarde voor C in functie daarvan:

```
163 IF X = Y THEN C = 16
    GOTO 170
164 IF Y = Z THEN C = 17
    GOTO 170
165 IF X = Z THEN C = 18
    GOTO 170
166 IF X > Y AND X > Z THEN C = 19
    GOTO 170
167 IF Y > X AND Y > Z THEN C = 20
    GOTO 170
```



```
168 IF Z > X AND Z > Y THEN C = 21
```

Ingangspunt van de programmasprong na de plaatsbepaling:

```
170 IF C=I THEN GOTO 177
```

Wanneer het geval (C) gelijk is aan het vorige, realiseer dan de toonhoogtebuigingsmogelijkheid voor de bespeler via een sprong naar 530; is dat niet het geval, speel dan de noten van het nieuwe akkoord in de progressie:

De M-lus loopt alle muzikale stemmen of midi-kanalen af die hier gebruikt worden (8):

```
171 FOR M=0 TO 7
```

De noten worden opgezocht in een speciaal segment van de look-up tabel in het RAM geheugen:

```
172 N(M) = PEEK($DF00 + (C * 8) + M)
```

Wanneer er geen noot staat, zet dan ook het volume op nul. Dit heeft te maken met de 'stommititeit' van midi waarbij noot 0 overeenstemt met een werkelijk opgewekte toonhoogte, een hele diepe Do.

```
IF N(M) = 0 THEN V(M) = 0
GOTO 175
```

De konstante 36 bepaalt in welke oktaaflijging de akkoordprogressie een aanvang zal nemen. Het volume wordt bewegingsafhankelijk opgeslagen in de gedimensioneerde variabele V(). De IF konditie verhindert overflow. Het midi-bereik is immers slechts 7-bits, of decimaal 0-127:

```
173 N(M) = N(M) + 36 + J
V(M) = 63 + ((A) / 2)
IF N(M) > $7F THEN N(M) = $7F
```

Aankomstpunt van de programmasprong. De vorige noot wordt uitgeschakeld als er een nieuwe te spelen is en de nieuwe wordt aangeslagen:

```
174 IF N(M) AND N(M) >< NO(M) THEN
D = U($80+M) + U(NO(M)) + U(0) + U(144+M) + U(N(M)) +U(V(M))
```

De pas gespeelde noot wordt in de geheugenvariabele NO() opgeslagen voor gebruik in de volgende lusdoorgang.

```
175 NO(M) = N(M)
```

De kanaal-lus wordt terug afgesloten:

```
176 NEXT M
```

Ingangspunt van de instructies die betrekking hebben op de tijdsstructuur van de progressie. Kijk op de interne klok en sla de waarde op in T:

```
177 T = SYSTIC
```

Uitvoering van de pich-bend (toonhoogtemodulatie) via de PMD midi-sekwens. Ook hier wordt een kanaaltellerlus gebruikt met M:

```
178 FOR M = 0 TO 7
179 IF N(M) THEN D = U($E0 + M) + U(0) + U(A)
180 NEXT M
```

Hoe groter de bewegingsamplitude hoe sneller de progressie vooruitgaat en hoe responsiever het instrument gaat reageren:

```
181 IF SYSTIC < T + 64 - (A / 2) THEN GOTO 178
```

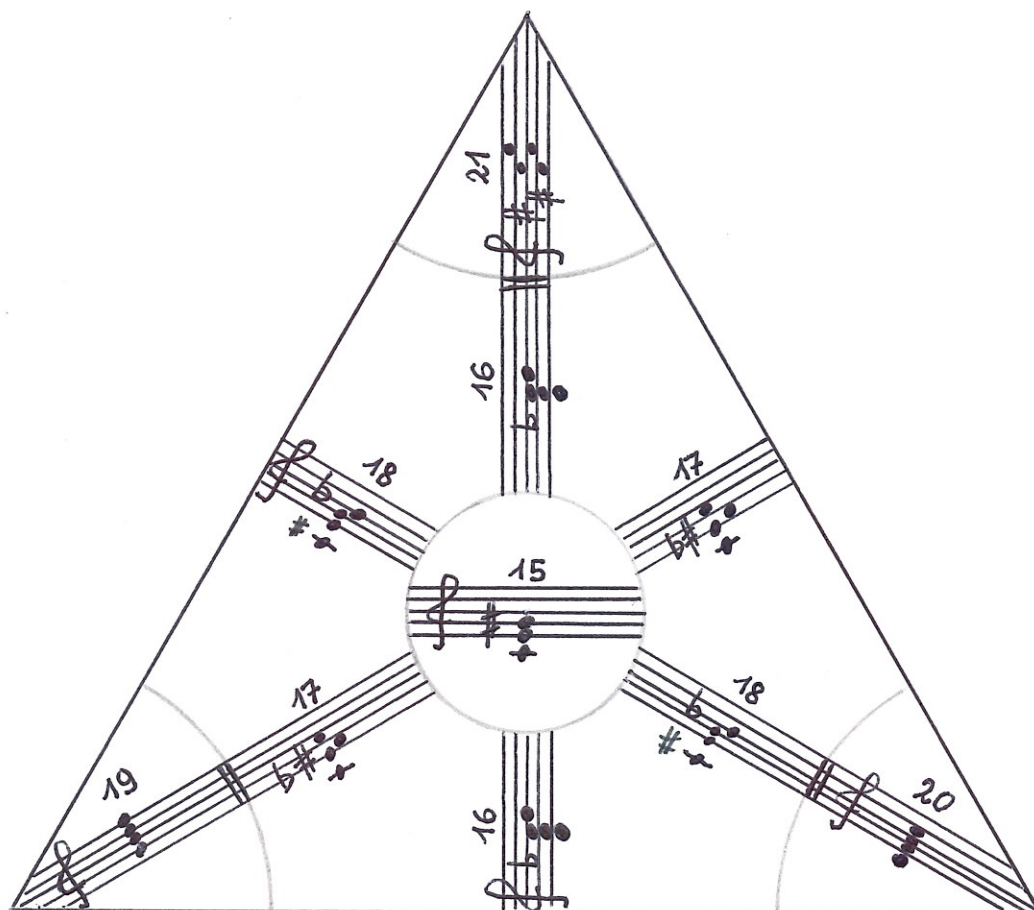
Sla het nummer van het vorig geval op in de variabele I en test de interne exit-konditie voor variabele J. De module eindigt dus op grond van haar intern muzikaal verloop. Niettemin werd toch een mogelijkheid ingebouwd haar te kunnen onderbreken via het kleine toetsenbord door het indrukken van een willekeurige toets:

```
182 I = C
    IF J < 75 AND KEYPAD = 0 THEN GOTO 161
```

Exit-punt van de module, waarbij taak 1 wordt opgeheven en terug wordt gekeerd naar de hoofdmenu-routine:

```
183 CANCEL 1
    GOTO 50
```

De initiële mapping van de akkoorden ziet eruit als volgt:



3.2.2.9.6.: " Sforte "

In dit algoritme wordt een imaginaire klavierspeler geïmplementeerd. Er wordt gebruik gemaakt van de verschillen (delta-waarden) tussen zes opeenvolgend genomen samples van de bewegingsinformatie.

Er wordt gebruik gemaakt van taak 4, een taak die in de variabelen X, Y, Z waarden tussen -30 en +30 retourneert. Deze variabelen bevatten dus de versnellingsinformatie met betrekking tot de beweging van de speler.

Indien de bekomen intervallen te klein worden om muzikaal bruikbaar te zijn, moet de tijd tussen het nemen van de individuele samples vergroot worden. In het omgekeerde geval kan ofwel de tijd verkleind worden, ofwel kan de waarde van de deler vergroot worden. Voorts wordt nog gebruik gemaakt van de globale amplitude-variabele A die hier echter diskontinu werkt: hij is 0 ofwel iets tussen 64 en 127.

Start taak 4 elke 40ms:

185 START 4,4

De waarden van de amplitude en van de te spelen noten worden gedurende deze module doorlopend ververst, iets wat alleen bij gebruik van multitasking tot de programmeermogelijkheden behoort. Hier begint de programmalus:

186 IF A = 0 THEN GOTO 186

Volgende noten worden op het linkerkanaal uitgestuurd:

188 IF X THEN D = U(\$90) + U(64+XF)+U(A)+U(\$95)+U(64-XF)+U(A)

Volgende klinken rechts:

189 IF Y THEN D = U(\$92) + U(64+YF)+U(A)+U(\$97)+U(64-YF)+U(A)

De volgende worden in het midden van het geluidsbeeld geplaatst:

190 IF Z THEN D = U(\$91) + U(64+ZF)+U(A)+U(\$96)+U(64-ZF)+U(A)

Bijzondere kondities met betrekking tot te spelen noten waardoor een bepaald tonaal centrum wordt gerealiseerd:

191 IF A > 48 AND X + Y + Z = 0 THEN D = U(\$93) + U(64)+U(120)

192 IF X + Y + Z > 64 THEN D = U(\$94) + U(40) + U(A)

Tempobepalende routine:

193 D = S(16)

Test de exit konditie om de module te verlaten:

194 IF KEYPAD = 0 THEN GOTO 186

Wanneer aan de exit konditie is voldaan, keer terug naar het hoofdkeuzemenu:

195 CANCEL 4
GOTO 50

3.2.2.9.7.: " Lead "

Deze module beoogt een eenvoudige implementatie te zijn van een enigszins bizar samengesteld groot symfonisch orkest. Het wordt gespeeld door het denkbeeldig aanwezige orkest te dirigeren.

De samenklanken zijn 'modernistisch', dissonant, dreigend en met abrupte inzetten. Samenklanken worden aangehouden door de beweging te stoppen.

Er wordt gebruik gemaakt van taak 3, een taak die volgende zeven variabelen retourneert:

A	bereik: 0 - 127
X,Y,Z	bereik: 0 - 127
XF,YF,ZF	bereik: 0 - 127

Start taak 18 om de 80ms:

```
200 START 18,8
```

Startpunt van de programmalus:

```
201 IF X + Y + Z = 0 THEN GOTO 201
```

De mapping gebeurt op een uiterst direkte wijze. Het is door dit 'gebrek aan muzikale samenhang' op het harmonische vlak dat het kompleks en dreigend karakter tot stand komt. De noot N(0) klinkt links, N(2) rechts, en N(1) in het centrum:

```
202 IF X THEN N(0) = X
203 IF Y THEN N(2) = Y
204 IF Z THEN N(1) = Z
```

Wanneer de noten te laag zouden uitvallen, transponeer ze dan drie oktaven omhoog:

```
205 IF N(0) AND N(0) < 48 THEN N(0) = N(0)+36
206 IF N(1) AND N(1) < 48 THEN N(1) = N(1) + 36
207 IF N(2) AND N(2) < 48 THEN N(2) = N(2) + 36
```

Bepaal een minimum geluidssterkte voor de A variabele. Bescherm evenwel A tevens voor overflow-kondities:

```
208 IF A THEN A = A + 48
A = BXOR(BOR(A,$FF80),$FF80)
```

Speel de noten uit de mapping, waarbij telkens de vroegere noot samen met de nieuwe noot wordt gespeeld. Dit verhoogt het idee van muzikale samenhang aanzienlijk. Merk op dat de A variabele gedurende de loop van deze instructie kan veranderen, wat een rijk geschakeerde dynamiek oplevert:

```
209 FOR M = 0 TO 5
D = U($90 +M) + U(N(M)) + U(A)
NEXT M
```

Onthoud de zopas gespeelde nieuwe noten en roep de tijdsroutine aan:

```
210 N(3) = N(0)
    N(4) = N(1)
    N(5) = N(2)
    D = S(15)
```

Test het toetsenbordje op een exit konditie:

```
211 L = KEYPAD
    IF L = 0 THEN GOTO 201
```

Indien echter toets 7 werd ingedrukt, verander dan de bron van bewegingsinformatie van taak 18 naar taak 1 en ga door met de muziek:

```
212 IF L = $37 THEN CANCEL 18
    START 1,8
    GOTO 201
```

Invalspunt voor de beëindiging van de module, waarbij tevens het slotakkoord wordt gespeeld:
C - G# - Eb - F - D

```
214 D=U($90) + U(72) + U($7F) + U($91) + U(80) + U($7F)
215 D=U($92)+U(87)+U($7F)+U($93)+U(89)+U($7F)+U($94)+U(98)+U($7F)
```

Haal de timerwaarde op ter bepaling van de duur van het slotakkoord:

```
216 T = SYSTIC
```

Tijdens de duur van dit akkoord staat de dirigent hopeloos te zwaaien en verliest de controle over zijn orkest. Alleen de globale dynamiek kan hij nog beïnvloeden. Dit gebeurt via de midi-parameter voor 'master'-volume:

```
217 FOR M = 0 TO 5
    D = U($B0 + M) + U(7) + U(64 + (A / 2))
    NEXT M
```

Dit akkoord wordt 10 seconden aangehouden:

```
218 IF SYSTIC < (T + 1000) THEN GOTO 217
```

Indien nu de bediener van het komputerboard een vergissing zou maken en een verkeerde toets zou indrukken, zou een taak niet worden opgeheven. Daarom schakelden we hier beide taken uit hoewel een van beide steeds overbodig is. De compiler schein hiertegen echter geen bezwaar te maken.

```
219 CANCEL 1
    CANCEL 18
    Keer terug naar het hoofdmenu:
    GOTO 50
```

3.2.2.9.8.: " Canvas "

Deze mapping herschept ons onzichtbaar instrument in het schilderdoek van een sonore 'action-painter'.

Het algoritme is bijzonder responsief en maakt gebruik van toonladders, toonhoogtebuidingen en glissandi evenals hoog-plastische volumekontrolle.

Taak 1 wordt gebruikt en retourneert:

X,Y,Z	0-127
A	0-127
XF,YF,ZF	0-127

Taak 1 wordt om de 50ms uitgevoerd:

220 START 1,5

Ingangspunt van de programmalus:

Klank links:

221 IF X > 36 AND X < 96 THEN D = U(\$90) + U(X+30) + U(\$7F)

Klank rechts:

222 IF Y > 36 AND Y < 96 THEN D = U(\$92) + U(Y + 30) + U(\$7F)

Klank in het midden:

223 IF Z > 36 AND Z < 96 THEN D = U(\$91) +U(Z + 30) + U(\$7F)

Extra klanken en geluiden (er worden immers 'niet-muzikale' geluiden gebruikt zoals kolkend water, wind en storm...), waarbij diverse parameters uit de beweging op diverse klankparameters gemapt worden:

224 IF X AND X < 36 THEN D = U(\$93) + U(X + 36) + U(BOR(A,64))

225 IF X > 64 THEN D = U(\$94) + U(X) + U(\$7F)

226 IF Y > 64 THEN D = U(\$95) + U(Y) + U(\$7F)

227 IF Y AND Y < 36 THEN D = U(\$96) + U(Y+36) + U(BOR(A,64))

228 IF Z > 64 THEN D = U(\$97) + U(Z) + U(\$7F)

Tijdsbepalende routine waarbij eerst op de systeemklok wordt gekeken en de tijd opgeslagen in de variabele L.

229 L = SYSTIC

Via de volgende instructies worden de glissandi en toonhoogtemodulaties uitgevoerd:

230 D = U(\$E0) + U(0) + U(X) + U(\$E2) + U(0) + U(Y)

D = U(\$E1) + U(0) + U(Z) + U(\$E4)

D = U(0) + U(X) + U(\$E5) + U(0) + U(Y)

Test de tijd-verstreken konditie gekoppeld aan de bewegingsamplitude:

```
232 IF SYSTIC < L + 25 AND A THEN GOTO 230
```

Test de exit voorwaarde:

```
233 IF KEYPAD = 0 THEN GOTO 221
```

Verlaat het algoritme en keer terug naar het hoofdmenu:

```
234 CANCEL 1  
GOTO 50
```

3.2.2.9.9.: " Close "

Deze module vormt de antipode of het komplement van de eerste. Ze wordt gebruikt ter afsluiting van het Book of Moves en is dus muzikaal gezien een finale.

Hier echter worden akkoorden gespeeld dan en dan alleen wanneer er beweging gedetekteerd wordt. De akkoorden doorlopen een gradueel decrescendo dat alleen helemaal aan het einde weer even in een crescendo overgaat. Parallel met dit -klassiek retorisch- dynamisch verloop, volgen de toonhoogtes een dalend patroon tot de finale unisono in de acht stemmen (lage C) is bereikt.

Dit algoritme maakt slechts van de totale bewegingsinformatie gebruik, via taak 1.

Vul de gedimensioneerde variabelen (arrays) met nieuwe noten uit de look-up tabel die hier begint op het RAM-geheugenadres \$DF00. Het beginakkoord is identiek aan het openingsakkoord van het stuk.

```
250 FOR M = 0 TO 7
    N(M) = PEEK($DF00 + M) + 24
NEXT M
```

Set (J=beginvolume) en reset (I=teller) de variabelen en start taak 1 om de 30ms:

```
251 J = $7F
    I = 0
    START 1,3
```

Ingangspunt van de programmalus:

```
252 IF A < 32 THEN GOTO 252
```

Indien dit niet het geval is, speel de noten. Dit maakt een minimum bewegingsdrempel noodzakelijk waaronder er niets gebeurt.

```
253 FOR M = 0 TO 7
    D = U($90 +M) + U(N(M)) + U(J +(I * 8))
NEXT M
```

Bepaling van de aanhoudtjidsduur voor het gespeelde akkoord:

```
254 D = S(16 + I + I + I)
```

Ga door met het spelen van hetzelfde akkoord zolang beweging gedetekteerd wordt:

```
255 IF A > 16 THEN GOTO 253
```

Wanneer echter geen beweging werd waargenomen, verminder dan het geluidsvolume via de J variabele (decrescendo). Begrens echter de laagste waarde van J tot 12 (muzikaal ppp).


```
256 J = J - 1
    IF J < 12 THEN J = 12
```

Zoek een willekeurig random getal tussen 0 en 7 dat kan gebruikt worden voor een willekeurige kanaaltoewijzing van de noten behorende tot het akkoord. (Dit implementeert een soort 'Klangfarbenmelodie'). Indien de laagste noot op sommige van de kanalen reeds werd bereikt en indien niet alle kanalen daar reeds zijn aangekomen, dan, als de willekeurig gekozen stem reeds op de laagste noot was aangekomen, zoek dan voort tot een stem is gevonden die nog niet op die laagste noot is aangekomen:

```
257 L = BXOR(BOR(RND,$FFF8),$FFF8)
    IF I AND (I < 8) AND (N(L) <= 36) THEN GOTO 257
```

Reset de I-teller waarin het aantal stemmen is opgeslagen die reeds hun laagste noot hebben bereikt. In de stemmen waarvoor dat niet het geval is, wordt de nootwaarde met een interval tussen een halve toon en een grote terts vermindert. Indien dit de noot te laag zou maken, begrenst haar tot de laagste noot.

```
258 I = 0
    N(L) = N(L) - (BXOR(BOR(RND,$FFFC),$FFFC) + 1)
    IF N(L) < 36 THEN N(L) = 36
```

Tel het aantal stemmen waarin reeds de lage Do bereikt is:

```
259 FOR M = 0 TO 7
260     IF N(M) <= 36 THEN I = I + 1
261 NEXT M
```

Ga terug naar het begin van de lus wanneer aan de exit voorwaarde nog niet is voldaan:

```
262 IF I < 8 THEN GOTO 252
```

Exit instructieregel van de hoofdprogrammalus, speel het slotakkoord (8 x unisono DO).

```
263 FOR M = 0 TO 7
    D = U($90+M) + U(36) + U($7F)
NEXT M
```

Deze module kan niet door ingrepen via het toetsenbord worden onderbroken. Wanneer het slotakkoord is bereikt stopt ze vanzelf en heft haar informatieverwervende taak op. Op het akkoord wordt een orgelpunt geplaatst:

```
264 CANCEL 1
    D = S(1000)
```

Keer terug naar het hoofdkeuzemenu:

```
265 GOTO 50
```

3.2.2.9.10.: " Prime Time "

Deze algoritmische module verloopt volledig polyritmisch. De zeven muzikale stemmen klinken alleen wanneer bewegingen gedetecteerd worden.

Voor het spelen van de noten wordt gebruik gemaakt van bijzondere taken in de multitasker. Dit maakt de polyritmie erg eenvoudig realiseerbaar.

Start alle taken (nrs. 5-10) waardoor noten worden uitgestuurd in de diverse stemmen. De metrische verhoudingen voor de periodiciteit in de individuele stemmen zijn afgeleid van priemgetallen ($\times 10$). Elke taak vraagt een variabele voor de te spelen noot evenals een voor de aanslagsterkte van die noot.

Voor de informatieverwerving wordt gebruik gemaakt van taak 4. Deze taak wordt in het 'agenda' van de multitasker geplaatst om de 50ms. De taak levert de variabelen XF,YF,ZF met het bereik -30 tot +30, evenals de amplitude variabele 0-127, diskontinu.

```
280 START 5,20
    START 6,33
    START 7,50
    START 8,70
    START 9,110
    START 10,130
    START 11,170
    START 4,5
```

Ingangspunt van de programmalus:

De noten (array-variabele N()) worden toegewezen in functie van de versnellingswaarden van de beweging en wel in termen van symmetrische afwijkingswaarden naar boven en naar onder ten opzichte van de centrale do van het klavier.

```
281 N(0) = 60 + XF
    N(1) = 60 + YF
    N(2) = 60 + ZF
    N(3) = 60 - XF
    N(4) = 60 - YF
    N(5) = 60 - ZF
    N(6) = 60 + (A / 2)
```

Voor de bepaling van de individuele volumes der stemmen wordt de vektoriele amplitudeinformatie rechtstreeks gebruikt.

```
282 V(0) = X :      V(3) = X
    V(1) = Y :      V(4) = Y
    V(2) = Z :      V(5) = Z
    V(6) = A
```

Test de voorwaarde -het indrukken van een toets- voor het verlaten van deze module:

```
283 IF KEYPAD=0 THEN GOTO 281
```

Verlaat de module , hef alle taken op en keer terug naar het hoofdmenu:

```
284 CANCEL 5: CANCEL 6: CANCEL 7: CANCEL 8: CANCEL 9
    CANCEL 10: CANCEL 11: CANCEL 4: GOTO 50
```

3.2.2.9.11.: " Call "

Deze module is muzikaal gezien driestemmig ontworpen. De stemmen cirkelen rond een tonaal centrum dat echter verschuift telkens de uitvoerder een beweging maakt die als zijnde in het centrum van de denkbeeldige tetraeder wordt waargenomen. De toonhoogte is een direkte mapping vanuit de vektoriele bewegingsnelheden.

Bij het begin wordt het tonaal centrum gelijk gesteld aan de centrale Do van het pianoklavier (60). De module reageert aanvankelijk aarzelend en traag, door de tijdsintervalvariabele L op 1.6s te bepalen. De geheugen-variabelen XH,YH,ZH worden gereset:

```
300 I = 36: L = 160
    XH = 0: YH = 0: ZH = 0
    Ingangspunt van de programmalus:
    Haal data voor het X-kanaal en beperk het bereik van
    XF:
```

```
301 XF = I + ADC(4)/24
    Haal ook een amplitude parameter voor X en map deze in
    het bereik 0 of 37 tot 87:
```

```
302 V(0) = ADC(0) / 20
    IF V(0) THEN V(0) = 36 + V(0)
    Wanneer de gevonden noot verschillend is van de vorige
    gespeelde noot binnen deze stem, speel haar:
```

```
303 IF XH >< XF THEN D = U($90) + U(XF) + U(V(0))

    Herhaal nu deze procedure van het Y kanaal en laat deze
    noten klinken door het rechteraudiokanaal:
```

```
305 YF = I + ADC(5) / 24
306 V(1) = ADC(1) / 20
    IF V(1) THEN V(1) = 36 + V(1)
307 IF YH >< YF THEN D = U($92) + U(YF) + U(V(1))
```

Herhaal de procedure voor het Z-kanaal en map deze noten in het centrum van het stereobeeld:

```
309 ZF = I + ADC(6) / 24
310 V(2) = ADC(2) / 20
    IF V(2) THEN V(2) = 36 + V(2)
311 IF ZH >< ZF THEN D = U($91) + U(ZF) + U(V(2))
```

Schakel alle eerder gespeelde noten uit:

```
312 D = U($81)+U(ZH)+U(0)+U($82)+U(YH)+U(0)+U($80)+U(XH)+U(0)
```

Sla de gespeelde noten op in het geheugen. Haal een nieuwe aktuele waarde voor de bewegingsamplitude op, en afhankelijk van het resultaat, wordt het tonaal centrum verschoven. De konditie wordt bepaald als XH=YH=ZH. Het bereik voor de variabele I is beperkt tot het trajekt 47 tot 72.

```

313 XH = XF
    YH = YF
    ZH = ZF
    IF XH = YH OR YH = ZH THEN I = 36 + (512 - ADC(3)) / 20

```

In functie van het tijdsverloop wordt de instrumentatie van de verschillende stemmen gewijzigd. De waarden van de hier gebruikte parameters staan uiteraard geheel in functie van de specificaties van de gebruikte Proteus module en de daarin opgeslagen en door ons op voorhand gewijzigde en aangepaste samples.

```

314 IF L = 1 THEN GOTO 320

```

Harmonium, Ney Fluit, Shofar

```

315 IF L = 50 THEN D = P(27,5,11,6,6,6,6,6)

```

Harmonium, Musette, Accordion

```

316 IF L = 10 THEN D = P(27,41,1,31,45,45,45,45)

```

Driestemmig harmonium:

```

317 IF L = 2 THEN D = P(27,27,27,27,27,27,27,27)

```

Tijdsbepalende routine, waarbij een gradueel accellerando wordt doorgevoerd, aan het einde waarvan de maximaal haalbare responsiviteit van dit onzichtbaar instrument bereikt wordt:

```

318 L = L - 1
    IF L < 1 THEN L = 1
320 D = S(L)

```

Test de exit konditie:

```

321 IF KEYPAD = 0 THEN GOTO 301

```

Indien deze exit konditie voldaan is, keer terug naar het hoofdmenu:

```

322 GOTO 50

```

3.2.2.9.12.: " Spooky "

In deze module wordt de vektoriele bewegingsinformatie gemapt op driestemmige nootinformatie. De amplitude informatie bepaalt de muzikale geluidssterkte. De noten worden gespeeld door gebruik te maken van de taken 6,7,8. Er wordt geen gebruik gemaakt van multitasking voor de informatieverwerking, wel voor het spelen van de verschillende stemmen.

Start de noot-speel taken voor de drie stemmen, met voor elke stem een andere periodiciteit:

```
350 START 5,59
    START 6,67
    START 7,71
```

Ingangspunt van de programmalus:

```
351 FOR M = 0 TO 2
```

Haal de nootinformatie uit de analoog-digitaal konverterkanalen 4,5,6 (Xf,Yf,Zf) voor de bewegingssnelheid.

Haal tevens de amplitudes (als 5-bit waarden) vanuit de AD kanalen 0,1,2 (X,Y,Z):

```
352 N(M) = ADC(M + 4) / 8: V(M) = ADC(M) / 8
```

Bepaal een minimale geluidssterkte:

```
353 IF V(M) THEN V(M) = BOR(V(M),48)
```

Filter al te lage noten weg:

```
354 IF N(M) < 24 THEN N(M) = 0 : V(M) = 0
```

Sluit de stemmenteller:

```
355 NEXT M
```

Test de exit konditie:

```
356 IF KEYPAD=0 THEN GOTO 351
```

Indien voldaan, schrap al de taken, en keer terug naar het hoofdselektiemenu:

```
357 CANCEL 5
    CANCEL 6
    CANCEL 7
    GOTO 50
```

3.2.2.9.13.: " Record-Playback "

Deze module is een poging om een soort muzikale lijntekening mogelijk te maken. De speler 'tekent' een melodische lijn in de lucht. De bewegingslijn wordt als een reeks samples opgeslagen en, nadien weergegeven als een melodische lijn. Het is een muzikale toepassing van wat we in het simulatieprogramma 'Holosimi', uitvoerig besproken in hoofdstuk 2, deden om de kurves om te zetten in muzikale notatie.

Van elk informatiekanaal worden 32 samples genomen. De waarden worden via POKE instructies in een stukje 'kladblok-geheugen' (scratchpad memory) weggeschreven. Aan 128 bytes hebben we genoeg. De geheugenadressen legden we tussen \$DE00 en \$DEFF, het bovenste topje van het voor de gebruiker beschikbare geheugen. Aan een procedure met rechtstreekse benadering van de geheugencellen werd de voorkeur gegeven boven het gebruik van variabelen, vanwege de veel grotere uitvoeringssnelheid van de verkregen kode.

Ingang van de hoofdlus voor opname en weergave:

```
400 L = -1
    I = 0
    D = U($93) + U(36) + U(120)
```

Ingangspunt van de opname-lus:

J is de teller waarmee de analoog-digitaal konverter kanalen worden afgelopen. De teller loopt als volgt: ZF, YF, XF, Amplitude.

```
401 FOR J = 6 TO 3 STEP -1
```

Haal een 10-bit waarde op van de transducers en sla 7-bits van die informatie op in variabele M.

```
402 M = ADC(J) / 8
    IF J = 3 AND M THEN M = BOR(M,24)
403 IF M < 24 THEN M=0
```

Ga naar een volgende geheugencel (L-teller), en schrijf de informatie weg naar een daarvoor via USERMEM \$200 gereserveerd stuk van het werkgeheugen:

```
404 L = L + 1
    POKE $DE00 + L,M
```

Sluit de kanaallus:

```
405 NEXT J
```

Voeg een 100ms wachttijd in tussen het nemen van de verschillende samples:

```
406 D=S(10)
```

Indien voldoende waarden (32) voor de bewegingssnelheid werden verzameld, verlaat dan de opnamelus. Zoniet verhoog te teller met een eenheid en keer terug naar

```

        het begin van de lus:
407 IF I<31 THEN I=I+1
      GOTO 401

```

Hierna volgt dan de weergave-lus voor de muzikale verklanking van de opgenomen beweging. De weergave verloopt in slow-motion op grond van een kompositorische beslissing.

```
410 D = U($93) + U(41) + U(110)
```

De I teller wordt gebruikt om alle geheugencellen af te lopen en uit te lezen:

```
411 FOR I=0 TO L-1 STEP 4
```

De inhoud van de geheugencellen wordt gedeeld door 4 om de afbeelding van beweging op melodie zo te vereenvoudigen dat het verband voor de kijker-luisteraar eenvoudig valt in te zien.

```

412 N(0) = (PEEK($DE00 + I)) / 4
      N(1) = (PEEK($DE01 + I)) / 4
      N(2) = (PEEK($DE02 + I)) / 4
      A = ((PEEK($DE03 + I)) / 3) * 3

```

De noten worden gespeeld, indien hun waarde niet 0 zijn:

```

413 IF N(0) THEN N(0) = 36 + N(0): D = U($90) + U(N(0)) + U(A)
414 IF N(1) THEN N(1) = 36 + N(1): D = U($92) + U(N(1)) + U(A)
415 IF N(2) THEN N(2) = 48 + N(2): D = U($91) + U(N(2)) + U(A)

```

Het afspeeltempo wordt door volgende instructie vastgelegd en bepaald op 100ms.

```
419 D = S(10)
```

De tellerlus voor de geheugenplaatsen wordt gesloten:

```
420 NEXT I
```

Alvorens een nieuwe reeks samples te nemen van een nieuwe bewegingsboog, worden de gespeelde noten uitgeschakeld:

```

421 FOR M = 0 TO 3
      D = U($B0 + M) + U($7B) + U(0) + S(10)
      NEXT M

```

Test de exit konditie. Indien geen toets werd ingedrukt, wordt terug gesprongen naar het begin van de programmalus (opname).

```
422 IF KEYPAD = 0 THEN GOTO 400
```

Indien aan de exit-voorwaarde is voldaan, keer dan terug naar het openingsmenu:

```
423 GOTO 50
```

3.2.2.9.14.: " Lock & Unlock "

In deze tweeledige module wordt een zevenstemmig eenvoudig slagwerkensemble geïmplementeerd. Elke stem speelt in een net ietsje verschillend tempo, waardoor we geleidelijke faseverschuivingen te horen krijgen. De gespeelde toonhoogtes evenals de dynamiek is gestuurd door de bewegingen.

In het tweede gedeelte van de module (Unlock) wordt dit patroon doorbroken en komen de stemmen metrisch gezien van elkaar los te staan. De metriek wordt hier een functie van de bewegingsparameters.

3.2.2.9.14.1: "lock...

De noten worden per stem in een eigen tempo gespeeld door taken toegewezen aan de multitasker. N(6)-V(6) is het MIDI-informatiekoppel (F#) voor de enkele trommelslag die het samenvallen van de fazecyclus markeert (taak 11).

Vervolgens worden de taken opgestart. De taken 5,6,7 besturen de muzikale stemmen 0,1,2 als afzonderlijke midikanalen.

```
450 N(6)=30
    V(6)=$7F
    START 11,10300
    START 5,100
    START 6,101
    START 7,102
```

De taken 8,9,10 sturen de muzikale stemmen 3,4,5 via de overeenkomstige midikanalen.

Taak 1 wordt gebruikt als bron van bewegingsinformatie en retourneert XF,YF,ZF, X,Y,Z.

```
451 START 8,103
    START 9,104
    START 10,105
    START 1,4
```

De bewegingssnelheidsinformatie vervat in XF,YF,ZF bepaalt de toonhoogtes. Het bereik voor de kanalen 0-2 loopt van 60 tot 92. Voor de kanalen 3 tot 5 loopt het van 28 tot 60.

De bewegingshoeveelheidsinformatie wordt rechtstreeks gebruikt ter controle van de aanslagsterkte van de instrumenten.

```
452 N(0) = 60 + ((XF - 64) / 2)
    V(0) = X
    N(1) = 60 + ((YF - 64) / 2)
    V(1) = Y
    N(2) = 60 + ((ZF - 64) / 2)
    V(2) = Z
453 N(3) = 60 - ((XF - 64) / 2)
    V(3) = X
    N(4) = 60 - ((YF - 64) / 2)
    V(4) = Y
    N(5) = 60 - ((ZF - 64) / 2)
    V(5) = Z
```


Test de exit konditie om wanneer dat gewenst is naar het tweede luik van deze module over te gaan:

```
454 IF KEYPAD = 0 THEN GOTO 452
      Schakel de eerder gebruikte taken waarmee de noten
      gespeeld worden uit:

455 CANCEL 5
      CANCEL 6
      CANCEL 7
      CANCEL 8
      CANCEL 9
      CANCEL 10
      CANCEL 1
      CANCEL 11
```

3.2.2.9.14.2: "...Unlock "

In dit tweede luik wordt het tempo waarin de noten worden gespeeld in de diverse stemmen een functie van de bewegingshoeveelheid. Hierbij wordt taak 3 gebruikt als informatiebron voor de bewegingsdata.

Deze taak levert $V(0), V(1), V(2)$, als aanslagsterktevariabelen afgeleid van de X, Y, Z , amplitudekanalen. De noten, $N(0), N(1), N(2)$ worden kwa oktaaflijging bepaald op grond van de snelheidsinformatie van de parameters XF, YF, ZF . De precieze noot echter binnen dit oktaaf laten we afhangen van de bewegingsversnellingsinformatie. De globale amplitude bepaalt het tijdsinterval tussen de noten.

Eerst wordt de instrumentatie naar de Proteus doorgegeven:

```
460 D = P(38,67,92,22,34,50,8,8)
      Vervolgens worden de taken waarmee de noten worden
      gespeeld opgestart evenals de taak die de informatie
      ophaalt:

461 START 3,10
      START 12,20
      START 13,20
      START 14,20
      START 15,30
      START 16,30
      START 17,30
```

Test voor de exit-konditie van de module, indien voldaan, schakel alle lopende taken terug uit.

```
462 IF KEYPAD = 0 THEN GOTO 462
463 CANCEL 3
      CANCEL 12
      CANCEL 13
      CANCEL 14
      CANCEL 15
      CANCEL 16
      CANCEL 17
      GOTO 50
```

3.2.2.9.15.: " Hammers "

Deze mapping is een poging om een maximum aan relevante bewegingsinformatie zo rechtstreeks mogelijk op overeenkomstige muzikale parameters af te beelden.

Daarom vertaalt het de vektoriele bewegingshoeveelheid in geluidssterkte, de vektoriele snelheid in de oktaafligging en de vektoriele versnelling van de beweging in de noten binnen elk oktaaf. De versnellingsinformatie wordt verworven via taak 3.

Start de informatieverwerkingstaak (3).

Start de taken waarmee de noten worden uitgestuurd over de gebruikte midikanalen (3-stemmig met een extra toegevoegd klankeffekt).

```
475 START 3,5
    START 5,11
    START 6,12
    START 7,13
    START 9,19
```

Wanneer de drie noten van elkaar verschillend zijn, wordt het effect gespeeld:

```
476 IF N(1) >< N(0) AND N(1) >< N(2) AND N(0) >< N(2) THEN
    N(4) = N(1)
    V(4) = V(1)/2
```

Test voor de exit-konditie:

```
477 IF KEYPAD = 0 THEN GOTO 476
```

Schakel alle taken uit en spring terug naar het hoofdselektiemenu.

```
478 CANCEL 3
    CANCEL 5
    CANCEL 6
    CANCEL 7
    CANCEL 9
    GOTO 50
```

3.2.2.10: De Taken

De hier volgende kodesegmenten illustreren overduidelijk de mogelijkheden van multitasking. De hier beschreven code wordt gestart via het START [n,m] kommando. [n] is daarbij het volgnummer van de taak, en [m] de periodiciteit waarmee ze dient te worden uitgevoerd. Deze periodiciteit wordt uitgedrukt in hondersten van een seconde.

De eerste taken worden gebruikt om informatie te verwerven vanuit de analoog-digitaal konverterkanalen. De konversietijd voor het hier toegepaste type konverter wordt door de fabrikant gespecificeerd als 100 μ s. In theorie zou dus een sampling-rate van 10ks/s haalbaar moeten zijn. Deze snelheid moet echter worden gedeeld door het aantal kanalen dat we willen uitlezen ($10ks/s / 7 = 1428s/s$) en verder is er nog de tijd die de softwarecode zelf nodig heeft tussen de verschillende omzettingen. In de praktijk bleek voor dit projekt de hoogste haalbare sampling rate rond 900s/s te liggen. Het teorema van Nyquist volgend, betekent dit dat de hoogste signaalfrequentie die nog zinvol te meten valt hooguit 450Hz bedraagt. Gezien het feit dat we het grootste deel van de rekenkundige bewerkingen eigenlijk reeds op het nivo van de analoge computerschakelingen konden uitvoeren, is dit hier ruimschoots voldoende. De specificaties van de filters in de analoge sectie zijn zo berekend, dat in het aan de AD-konverters aangeboden signaal geen frekwentiekomponenten meer aanwezig zijn die de Nyquist grens te boven gaan.

3.2.2.10.1.: Taak 1

500 TASK 1

Deze taak leest de beschikbare informatie van de analoog-digitaal konverterkanalen 0-6. (7 kanalen).

De geretourneerde variabelen zijn X,Y,Z en de waarden ervan vallen binnen het bereik 0-127. (7-bit) Deze waarden komen overeen met de geïntegreerde amplitude van de overeenkomstige kanalen. Alleen de hoogste 7-bits worden dus gebruikt. (De omzettingen zelf grijpen plaats met een resolutie van 10-bit). Dit werd gedaan omdat de laagste bits op grond van de precisie in de analoge computerschakelingen sowieso beperkt is tot niet veel beter dan 1% terwijl ook de signaal-ruisverhouding deze lage bits volstrekt onbetrouwbaar en irrelevant maakt. Bovendien is een oplossend vermogen ruim voldoende voor de mogelijkheden die het MIDI-systeem kan bieden en, last but not least, betwijfelen we of de mens tot een grotere lichaamscontrole in staat is. Maar, dus zelfs indien dat toch het geval ware, zou een grotere resolutie gezien de beperkingen van onze hardware, geen aarde aan de dijk brengen.

Verder levert deze taak ook de variabelen XF.YF en ZF op binnen hetzelfde bereik: 0 - 127. Deze variabelen zijn evenredig met de vektoriele bewegingssnelheid. De overdrachtfunctie en vooral het verloop ervan (lineair, exponentieel, logaritmisch) hangt uiteraard af van de

ingestelde rekenkundige functie op de analoge komputer. Als laatste variabele levert deze taak ons nog een waarde voor de globale bewegingshoeveelheid, geïntegreerd over een langer tijdsinterval dat voor de vektoriele bewegingshoeveelheden X,Y,Z. Ook A heeft een resolutie van 7 bits en valt in het bereik 0-127.

```
501 XF = ADC(4) / 8
    X = ADC(0) / 8
502 YF = ADC(5) / 8
    Y = ADC(1) / 8
503 ZF = ADC(6) / 8
    Z = ADC(2) / 8
504 A = ADC(3) / 8
506 EXIT
```

3.2.2.10.2.: Taak 2

Deze taakkode wordt gebruikt voor de implementatie van het onzichtbaar slagwerk in module 4 ("Beat").

De variabelen die erdoor geretourneerd worden zijn:

L = waarin de piek-amplitude (grootste waarde van twee lezingen) wordt opgeslagen.

X,Y,Z zoals in taak 1 (0 -127)

C is een logische variabele en is 0 (vals) wanneer de taak niet beeindigd werd, -1 (waar) wanneer ze werd afgewerkt.

Deze constructie is ook weer typisch voor multitasking, waarbij het immers mogelijk is dat de taak-kode onderbroken wordt voor het uitvoeren van een andere opdracht. Wanneer die andere code gegevens uit de taak gebruikt, kan het van belang zijn te weten in welke staat van afwerking deze gegevens verkeerden.

XF,YF,ZF, zoals in taak 1 maar met een beperkter bereik van 0-63, en dus met een resolutie van slechts 6-bits.

A zoals in taak 1.

```
510 TASK 2
```

Reset alle variabelen:

```
511 C = 0
    A = ADC(3) / 8
    X = ADC(0) / 8
    Y = ADC(1) / 8
    Z = ADC(2) / 8
```

Indien geen beweging gedetekteerd werd, verlaat dan de taak zonder enige variabele te wijzigen:

```
513 IF (X + Y + Z = 0) OR (A = 0) THEN GOTO 518
```

Lees de bewegingssnelheidsvariabelen. Konverteer via deling door 16 tot een 6 bits getal (0-64).

```
514 XF = ADC(4) / 16
    YF = ADC(5) / 16
    ZF = ADC(6) / 16
516 L = ADC(3) / 8
    IF L > A THEN A = L
517 C=1
518 EXIT
```

3.2.2.10.3: Taak 3

Deze taak retourneert rechtstreeks MIDI noten en aanslagsterkten via gedimensioneerde variabelen. In V(0), V(1), V(2) worden de bewegingshoeveelheden opgeslagen. De bewegingssnelheden worden in de oktaaflijging ondergebracht en de noten worden afgeleid van de hier berekende vektoriele versnellingen van de bewegingen. De uiteindelijke notenwaarden worden geretourneerd in N(0),N(1),N(2).

Op grond van onze ervaring met dit instrument levert een dergelijke mapping een erg muzikaal aanvoelend reageren van het instrument op de beweging van de uitvoerder. Talloze musici aan wie we hebben gevraagd het uit te proberen deelden deze mening. Een inherente logica voor deze mapping lijkt ons echter moeilijk te geven. Daarom nemen we dat het aan de gebruiker overgelaten moet worden de mapping naar keuze ook anders te organiseren en bijvoorbeeld de versnelling op de oktaaflijging (of een ander interval indien gewenst) en de snelheid op de noot af te beelden.

Taak 3 wordt gebruikt in de modules "Hammers" en "...unlock".

520 TASK 3

X,Y,Z worden hier gebruikt voor de eerste inlezing van de snelheidsinformatie. Merk op dat we hier de volle 10 bit waarde opslaan.

```
521 X = ADC(4)
    Y = ADC(5)
    Z = ADC(6)
```

De volumes worden rechtstreeks ingelezen in de betreffende Midi-variabelen:

```
522 V(0) = ADC(0) / 8
    V(1) = ADC(1) / 8
    V(2) = ADC(2) / 8
```

De globale amplitude wordt afgeknot gebruikt:

```
527 A=ADC(3) / 3
    IF A > 127 THEN A = 127
```

De booleaanse somfunctie wordt gebruikt ter bepaling van de uit te sturen aanslagsterkte:

```
530 V(0) = BOR(V(0),A)
    V(1) = BOR(V(1),A)
    V(2) = BOR(V(2),A)
```

Bepaal de tессituurligging van de te spelen noten in grove stappen:

```
532 N(0) = (X / 210) * 24
    N(1) = (Y / 210) * 24
    N(2) = (Z / 210) * 24
```

In de volgende berekeningen wordt de versnellingsinformatie berekend op grond van een nieuwe

opmeting van de snelheid. Het verschil tussen X en ADC(4) is de versnelling. Omwille van de responsiviteit van het algoritme worden de metingen voorwaardelijk uitgevoerd, alleen wanneer er reeds eerder een beweging was vastgesteld.

```
533 IF N(0) THEN N(0) = N(0) + (ADC(4)-X)/32
534 IF N(1) THEN N(1) = N(1) + (ADC(5)-Y)/32
535 IF N(2) THEN N(2) = N(2) + (ADC(6)-Z)/32
539 EXIT
```

3.2.2.10.4: Taak 4

Wanneer de uitlezingen van de kanalen 4,5,6 (XF,YF,ZF) waarden opleveren evenredig met de Dopplerfrequenties en dus met de vektoriele bewegingssnelheden, dan levert het verschil tussen twee opeenvolgende waarden van deze parameter ons een waarde proportioneel met de versnelling (positief of negatief).

De door deze taak geretourneerde variabelen zijn:

- X,Y,Z bereik: 0 - 127 kanaal amplitudes
- XF,YF,ZF bereik: - 30 to 30 vektoriele versnellingen
- XH,YH,ZH bereik: 0 - 63 vektoriele snelheden
- A bereik: in diskontinue stapjes.

Taak 4 wordt gebruikt in de modules 'Sforte' en 'Prime Time'.

```
540 TASK 4
```

Lees de waarden van kanalen 0.1.2, en draag de 7-bit waarden over in de amplitudevariabelen X,Y,Z.
Lees de waarden van de kanalen 4,5,6 en draag de 6-bit waarden over in de snelheidsvariabelen XH,YH,ZH.

```
541 X = ADC(0) / 8
    Y = ADC(1) / 8
    Z = ADC(2) / 8
    XH = ADC(4) / 16
    YH = ADC(5) / 16
    ZH = ADC(6) / 16
```

Lees de globale amplitude, en voeg er door afknotting, een duidelijk sforzando effect aan toe. Dit aspect wordt dus een eigenschap van de aanslag van het geïmplementeerde instrument.

```
543 A = ADC(3) / 4
    IF A > 64 THEN A = 127
    Lees nieuwe snelheidinformatie in als 6-bit waarden,
    bereken de versnelling en sla het resultaat op in
    XF,YF,ZF
```

```
544 IF X THEN XF = XH - (ADC(4) / 16)
545 IF Y THEN YF = YH - (ADC(5) / 16)
546 IF Z THEN ZF = ZH - (ADC(6) / 16)
```

```
547 EXIT
```

3.2.2.10.5.: Taken 5 tot 11

De reeks taken van 5 tot en met 11 worden uitsluitend gebruikt voor het spelen van MIDI-noten door per taak een andere muzikale stem. Aangezien de taken in het 'agenda' van de multitasker kunnen opgenomen worden met een verschillende periodiciteit van uitvoering, wordt het implementeren van ingewikkelde muzikaal ritmische patronen een fluitje van een cent. Aangezien we zeven taken definieerden, kunnen we zo iets zevenstemmig uitvoeren.

Vanzelfsprekend dienen de parameters elders een waarde toegekend te krijgen: zowel noten en volumes moeten toegewezen worden voor elke afzonderlijke taak.

Eerste stem:

```
550 TASK 5
551 D = U($90) + U(N(0)) + U(V(0))
552 EXIT
```

Tweede stem:

```
560 TASK 6
561 D = U($91) + U(N(1)) + U(V(1))
562 EXIT
```

Derde stem:

```
570 TASK 7
571 D = U($92) + U(N(2)) + U(V(2))
572 EXIT
```

Vierde stem:

```
580 TASK 8
581 D = U($93) + U(N(3)) + U(V(3))
582 EXIT
```

Vijfde stem:

```
590 TASK 9
591 D = U($94) + U(N(4)) + U(V(4))
592 EXIT
```

Zesde stem:

```
600 TASK 10
601 D = U($95) + U(N(5)) + U(V(5))
602 EXIT
```

Zevende stem:

```
610 TASK 11
611 D = U($96) + U(N(6)) + U(V(6))
612 EXIT
```

3.2.2.10.6.: Taken 12 tot 17

De taken in de reeks 12 tot en met 17 spelen net zoals de vorige reeks noten via de midi-kanalen, maar hier schrapt elke taak zichzelf uit de agenda van de multitasker wanneer ze beëindigd wordt. De SUSPEND-tijd is een lineaire functie van de geïntegreerde som der bewegingsamplitudes van de X,Y,Z kanalen, dus de A-waarde.

De parameters voor deze takenreeks worden normaal gezien ingevuld door taak 4.

Eerste stem: (X-vektor)

```
620 TASK 12
621 IF N(0) THEN D = U($90) + U(N(0)) + U(V(0))
    SUSPEND 32 - (A / 4)
622 EXIT
```

Tweede stem: (Y-vektor)

```
630 TASK 13
631 IF N(1) THEN D = U($92) + U(N(1)) + U(V(1))
    SUSPEND 32 - (A / 4)
632 EXIT
```

Derde stem: (Z-vektor)

```
640 TASK 14
641 IF N(2) THEN D = U($91) + U(N(2)) + U(V(2))
    SUSPEND 32 - (A / 4)
642 EXIT
```

Vierde stem:

De te spelen noot is het gemiddelde tussen N(0) en N(1).

```
650 TASK 15
651 IF N(0) THEN D = U($93) + U((N(0) + N(1)) / 2) + U(V(0))
    SUSPEND 32 - (A / 4)
652 EXIT
```

Vijfde stem:

De te spelen noot is het gemiddelde tussen N(1) en N(2).

```
660 TASK 16
661 IF N(1) THEN D = U($95) + U((N(1) + N(2)) / 2) + U(V(1))
    SUSPEND 32 - (A / 4)
662 EXIT
```

Zesde stem:

De te spelen noot is het gemiddelde tussen N(0) en N(2).

```
670 TASK 17
671 IF N(2) THEN D = U($94) + U((N(0) + N(2)) / 2) + U(V(2))
    SUSPEND 32 - (A / 4)
672 EXIT
```


3.2.2.10.7.: Taak 18

Deze taak is eigenlijk een vroegere versie van wat eerst taak 1 was. In principe vervult ze precies dezelfde opdracht. Het eigenaardige -zoals bleek uit de praktijk- is echter dat ze toch een heel verschillend esthetisch resultaat blijkt op te leveren. Aangezien ik dit resultaat eigenlijk heel goed vond behield ik dit stuk kode in het programma. De taak wordt gebruikt in de modules "Lead" en "Topoi". In de module "Lead" worden beide achtereenvolgens gebruikt en leveren daar een muzikale variatie op.

800 TASK 18

De ADC-kommandos zijn hier kwazi in assembler gekodeerd. Er wordt gebruik gemaakt van OUT en IN kommandos, die alleen op byte nivo functioneren.

De syntax is bepaald door de hardware uiteraard:

OUT \$91, <kommando>

kommando: moet 3 zijn, omdat daarmee de klokfrequentie van de AD-converter wordt aangepast aan de hardware.

OUT \$90, <kanaal kommando>

kanaal-kommando: het volgnummer van het te bemonsteren kanaal.

<Var> = INP(\$91)

de 8-bit waarde van de AD-converter kan worden ingelezen in Var.

Om een 7-bit waarde te verkrijgen delen we het resultaat uiteraard door 2.

De geretourneerde waarden zijn identisch met die van taak 1: dus X,Y,Z, (0-127), XF,YF,ZF, (0-127), A(0-127).

```
801 OUT $91,3: OUT $90,0: X = INP($91)/2
      OUT $91,3: OUT $90,1: Y = INP($91)/2
802 OUT $91,3: OUT $90,2: Z = INP($91)/2
      OUT $91,3: OUT $90,4: XF = INP($91)/2
803 OUT $91,3: OUT $90,5: YF = INP($91)/2
      OUT $91,3: OUT $90,6: ZF = INP($91)/2
```

Voor module 2 ("Topoi") moet de snelheidsinformatie omgezet worden naar een andere schaal:

```
804 IF E = 2 THEN XF = XF / 9 : YF = YF / 9 : ZF = ZF / 9
```

De globale amplitude wordt ingelezen en omgezet in een diskontinue variabele. Delen door 16 en vermenigvuldigen met 8 levert dit resultaat wanneer men er zich tenminste rekenschap van geeft dat alle bewerkingen ingevolge onze deklaraties uitgevoerd worden met als resultaat gehele getallen.

```
805 OUT $91,3 : OUT $90,3 : A = (INP($91) / 16) * 8
806 EXIT
```

3.2.2.11.: Opzoekingstabel voor muzikale gegevens

De hier volgende data-statements bevatten eigenlijk niets anders dan de 'a priori kennis' van het instrument. Het zijn de tabellen met definities van het openingsakkoord, de harmonische progressies gebruikt voor "Topoi" , de structuur van de mineur toonladder gebruikt in 'Minor' enz...

De data zien eruit als 16-bit getallen. Ze moeten dan ook bij gebruik gesplitst worden in MSB en LSB. Dit werd gedaan om geheugen te sparen.

```

900 DATA 4047,5055,6266,7381,0,0,0,0
901 DATA 6,0,0,1100,1100,7,2,0
902 DATA 0,900,500,12,3224,2820,1609,1204
903 DATA 12,0,0,400,1200,9,6,0
904 DATA 0,1100,700,4,0,9,5,20
905 DATA 0,0,1100,307,700,4,2,0
906 DATA 12,9,5,300,0,1100,700,5
907 DATA 0,200,900,705,12,12,4,800
908 DATA 211,5,8,0,9,1206,0,300
909 DATA 4,100,700,10,7,5,0,211
910 DATA 12,200,500,9,104,6,9,0
915 DATA 2,305,709,1112,1415,1719,2324,2627

```

3.2.2.12: Subroutines:

Deze subroutine wordt gebruikt om de in de data-statements opgesloten informatie terug te ontbinden naar enkelvoudige bytes die rechtstreeks in het geheugen geplaatst worden via POKE statements. Dit geheugen (\$DF00-\$DFFF) dient wel voorafgaand aan de compilatie van het programma vrijgemaakt te worden via het systeemkommando USERMEM \$200.

```

916 FOR M = 0 TO $FE STEP 2
917 IF M < $B0 OR M > $EF THEN READ I:
          POKE ($DF00 + M),I / 100
          POKE ($DF01 + M),I - ((I / 100) * 100)
918 NEXT M
919 I = 0 : RETURN

```

```

USERMEM $200
NOERR
END

```

```

CODE ENDS
[EOF]

```

3.2.3: Geheugenstructuur van het Multitraxboard na compilatie van het "A Book of Moves" programma:

Opgemerkt moet worden dat we het computersysteempje werkelijk tot op zijn uiterste grenzen hebben gebruikt.

De geheugenindeling van het board ziet eruit als volgt:

	Start Addr	End Addr	Hex Size	Dec Size		
MTOS-MTBasic	0000	7FFF	7FFF	32767	ROM	0 - 7FFF
Source code:	8000	C2BE	42BF	17087	RAM	8000 - C2BE
Compiled code:	4500	BFE2	7AE3	31459	EPROM	
Comp code free:	BFE3	BFFF	001D	29	EPROM	
Src/Vars free:	C2BF	C870	05B2	1458	RAM	C2BF - C870
Variables:	C871	DDFF	158F	5519	RAM	C871 - DDFF
User space:	DE00	DFFF	0200	512	RAM	DE00 - DFFF

De geheugenallokatie in absolute hardware adres-blokken is:

MTOS	00000	07FFF	0 - 32767	
RAM1	08000	0FFFF	32768 - 65535	Source code
RAM2	10000	17FFF	65536 - 98303	Compiled Code
EPROM	18000	1FFFF	98304 - 131071	Eprom Source code

Hoewel het Multitrax board zelf een EPROM-programmer aan boord heeft, bleek het onmogelijk om een zelfstartend systeem in een EPROM geprogrammeerd te krijgen. De instructies USERMEM \$200 en NOERR voor de compiler, kunnen alleen via een externe EPROM-programmer en debugger in de ROM geschoven worden. Dit feit is nergens gedocumenteerd in de manuals van het Multitrax board.

3.3.: Artistieke bespreking en presentatie:

"A Book of Moves" is de artistieke voorstelling geworden van een voorlopig eindpunt in een lang proces van zoeken naar een instrument dat rechtstreeks vanuit de bewegingsmotoriek zou worden gestuurd. Men zou zich hierbij uiteraard de vraag kunnen stellen wat het instrument zou opleveren los van deze of enige artistieke presentatie. Welnu, wij geloven dat dit onmogelijk is. Meer nog, principieel gezien is ons instrument in deze niet eens fundamenteel verschillend van enig traditioneel instrument. Een traditioneel instrument sluit in zich immers eveneens een heel arsenaal aan artistieke en esthetische keuzes van de maker, van de koper en van de bespeler. Zoals we in ons eerste hoofdstuk aantoonde, is het muziekinstrument principieel als expressief werktuig artistiek en esthetisch waardegeladen. Het 'bevat' een beeld van de muziek waarvoor het werd gekoncipieerd.

Ook een traditioneel instrument kan niet zinnig worden bestudeerd, laat staan beoordeeld, wanneer het niet wordt bespeeld. In de bespeling en de bespeelbaarheid ligt uiteindelijk de enige justifikatie van een muziekinstrument.

Zo ook dus, geldt dit alles voor het hier voorgesteld instrument. Daarom ook zou het geen zin hebben, alleen te volstaan met een technische beschrijving van de hardware. In dit geval ware zo iets nog veel onzinniger dan in het geval van traditionele instrumenten. Immers voor dit instrument bestaat geen standaard muzikaal repertoire. Het betracht immers nieuwe speelwijzen mogelijk te maken.

Nu willen we anderzijds ook geenszins de justifikatie van het door ons hier voorlopig voorgestelde instrument uitsluitend afhankelijk maken van het ene stuk wat we er zelf voor koncipieerden. Niets zegt immers dat dit ook maar ergens het best mogelijke maakbare stuk zou zijn gebruikmakend van dit instrument. "A Book of Moves" moet dan ook gezien worden als een artistiek en demonstratief voorbeeld. Het beoogt het instrument in werking te tonen en te laten horen.

We hopen en geloven echter dat de hier aangeduide weg een mogelijke richting voor de toekomstige instrumentenbouw zou kunnen aanwijzen. Dat er in zo'n proces nog ontelbare verbeteringen aan onze voorstellen en ontwerpen mogelijk zijn weten we wellicht zelf beter dan wie ook. Misschien ook hebben we ons op enige belangrijke punten fundamenteel vergist...

Mogelijke perspectieven voor verdere verbeteringen en research hebben we, voorzover we dat zelf kunnen zien, aangegeven in het volgende hoofdstuk.

Omdat na al deze kode-beschrijvingen en beschrijvingen van de meest diverse technisch-elektronische details -waarzonder evenwel niets zou functioneren en we in de filosofie in de slechtste betekenis van het woord zouden blijven zweven- de lezer wellicht het muzikale spoor bijster is geraakt, hernemen we hier heel beknopt en in meer muzikaal-artistieke termen de 16 verschillende modules waaruit "A Book of Moves" is opgebouwd. We geven onze beschrijving hier tevens in de volgorde waarin we ze bij de vele concertuitvoeringen die we er reeds van mochten verzorgen ook presenteerden.

Open

Een ouverture gebaseerd op een als trage mokerslagen steeds maar herhaald 8-tonig breed en kompleks akkoord.

Bij de aanvang van het stuk zit de speler in het midden van het podium ineengehurkt op de grond. Zij draagt een pakje in hoogglanzend metaalachtig tekstiel.

Na honderdachtentwintig *ff*-aanslagen van het akkoord -waarbij de tonen tussen de luidsprekers over en weer springen- zwaait zij een arm in de lucht. De akkoorden worden onderbroken, maar gaan van zodra de beweging stopt weer door. Ze zijn nu echter een nauwelijks hoorbare gradatie zachter geworden.

De speler maakt langzaam meer en meer bewegingen daarbij verduidelijkend dat zij de klank wég wil. De klank wordt alsmaar stiller. Zij moet echter alsmaar grotere inspanningen leveren om dit verstillingsproces te laten doorgaan. Wanneer de bewegingen een hoogtepunt bereiken, wordt het helemaal stil. De ouverture is afgelopen.

In klassieke zin komt deze ouverture volledig overeen met de opening van een klassieke redevoering: het onderwerp, in dit geval de beweging-klank transformatie- wordt zo duidelijk mogelijk afgebakend. Het probleem wordt gesteld in een tot zijn meest eenvoudige gedaante gereduceerde vorm. Daarom noemen we deze ouverture een retorische formule.

Call

Nadat het probleem in alle klaarheid en vereenvoudiging is gesteld, wordt nu de algemene geldigheid ervan betoogd.

In de kompositie 'Call' worden gradueel alsmaar meer muzikale eigenschappen aan de bewegingen van de speler gekoppeld. De volledige werking van het onzichtbare instrument wordt gedemonstreerd.

De module opent met de klanken van Shofars vanuit de verte. De koppeling aan de zwaai-bewegingen van de speler zijn aanvankelijk erg vaag. De instrumenten komen echter dichterbij en veranderen ook geheel van klank. Ze worden uiteindelijk gemetamorfozeerd tot een enkel akkordeon, waarbij de speler ook werkelijk een fiktieve akkordeon gaat bespelen. Hier kan de speler heel getrouw gaan improviseren. De klanken sluiten met grote direktheid aan bij de uitgevoerde bewegingen. Gedurende het gehele stuk blijft de speler hier in het midden van het speelveld.

Topoi

In 'Topoi' wordt de muzikale ruimte -het formaat- van het muziekinstrument afgetast. Daartoe wordt via verplaatsingen van de speler over de scene, aan de diverse plaatsen verschillende harmonische konstellaties toegekend. Er kunnen plaatsafhankelijke akkoordprogressies worden gespeeld. Het tonaal centrum (hoewel het niet om tonale muziek gaat in traditionele zin- reden waarom we dan ook deze term eerder dan 'tonika' gebruiken-) bevindt zich in het centrum van de opstelling. Hoe verder de speler zich van dit centrum verwijderd hoe verder ook de tot klinken gebrachte harmoniën van dit tonaal centrum gaan afwijken.

Rising

In deze formeel gezien uiterst eenvoudige kompositie worden we muzikaal gekonfronteerd met een graduele akkoordenprogressie van laag naar hoog. Naar heel erg hoog, dat wel... De wijze waarop de progressie vooruitloopt is echter geheel afhankelijk van de bewegingen van de bespeler. Hij speelt dus eigenlijk een vastgelegde partituur waar uitsluitend een soort becijferde bas op voorgeschreven staat. De grootte van de intervallen evenals de metrische geleiding kan hij volledig zelf interpreteren.

Retorisch gezien krijgen we te maken met een figuur waarin een alsmar grotere spanning wordt opgebouwd.

Spooky

De schrille ijle en hoge klanken waarmee 'Rising' wordt afgesloten, monden uit in een totaal 'andere' muzikale klankenwereld. Hier krijgen we moeilijk thuis te brengen klanken en geluiden afkomstig van niet-toonhoogte gerichte muziekinstrumenten. Alle harmonie lijkt opgeheven. Ook van enige melodische samenhang is hier geen sprake meer. De klankenwereld is zuiver vloeibaar en timbraal. De bewegingen zijn spookachtig en zwevend. De klanken worden bijzonder plastisch en op continue wijze door de bewegingen gestuurd. De speler verdwijnt in de klankennevel van de scene.

Minor

Een nieuwe speler verschijnt ten tonele. Hij speelt een onzichtbaar klavierinstrument. De toonsoort is duidelijk mineur. Hij heeft de melodische patronen letterlijk in de hand. De muziek wordt door haar centrum aangezogen. De melodische patronen cirkelen rond het tonaal centrum. De metriek is rustig maar niet regelmatig. Het dynamisch bereik is beperkt tot het bereik *mp* tot *mf*. De sfeer is vriendelijk. De bewegingen van de speler zijn soepel en beperken zich grotendeels tot de bovenste ledematen.

Beat

In deze kompositie staat op de scene een bijzonder groot aantal slaginstrumenten van het meest diverse pluimage opgesteld: westerse drumkits, symfonische slaginstrumenten, latijns-amerikaans slagwerk, afrikaanse trommels, gamelan-instrumenten en nog meer dergelijk exotika. Als een parodie op nogal wat solistische slagwerk-literatuur uit de seriele en post-seriele periode, springt de speler van het ene instrument naar het andere waarbij hij zich vaak in de meest onmogelijke bochten dient te wringen om bepaalde klanken te kunnen spelen. De klanken zijn volledig op de bewegingen afgebeeld. Van de bespeler van het onzichtbaar instrumentarium wordt een soort muzikaal karate verwacht. De bewegingen moeten abrupt zijn, zonder de minste elegantie. Immers ook waar 'normaal' slagwerk wordt gespeeld, eindigen de armbewegingen van de speler toch in een abrupte botsing met het klankopwekkend gedeelte van het slaginstrument...

Sforte

Zoals de naam van het stukje al laat uitschijnen, levert het hier bespeelde instrument een uitgesproken sforzando element. De dynamische nuances van de bewegingen van de bespeler worden daarbij dus sterk overdreven. Het melodisch verloop is uitgesproken centripetaal en gericht naar een tonaal centrum.

Anders dan in 'minor' echter, is de sfeer hier geenszins vriendelijk. Er worden talloze smalle klusters gespeeld, waarbij de breedte afhangt van de energie, de inspanning, van de bespeler. Hoe groter de geïnvesteerde energie, hoe groter ook de afwijking van de akkoorden en klusters van het tonaal centrum.

Lead

In deze kompositie wordt de bespeler tot dirigent van een betrekkelijk groot 'symfonisch' orkest. De toonspraak is uitgesproken modernistisch en vormt een 'karikatuur' van de orkestratie zoals we die zouden kunnen aantreffen bij komponisten zoals Xenakis, of, bij ons dan, Luc Brewaews. De dirigent kan met armen -en desgewenst ook met de benen- orkestrale groepen doen inzetten, samenklanken aanhouden en de gehele metriek bepalen. De muzikale samenhang berust op een bijna triviaal maar uitermate effectief middel: de overlapping van de vorige met de volgende samenklank. Dit is het enige 'kompositorische' gegeven dat hier a priori vastligt, afgezien dan van de orkestratie zelf. De orkestbehandeling is volledig acht-stemmig.

Canvas

In dit stuk wordt gespeeld met in hoofdzaak 'niet-muzikale' geluiden. Geluiden die zich in elk geval moeilijk laten inpassen in de klassieke traditie van de Westeuropese muziek waarbij in eerste plaats de diskontinue organisatie van de muzikale parameters toonhoogte en tijd centraal heet te staan. Hier krijgen we het omgekeerde: de muzikale parameters toonhoogte -voorzover die al duidelijk bepaalbaar zou zijn in de hier gebruikte klanken-, geluidsterkte en tijdstrukturering worden hier als een plastische massa, een sonore brei, behandeld. Een klankmoeras zo men wil. De uitvoerder speelt en manipuleert de klankmassa een beetje zoals een 'action painter' zijn schilderdoek te lijf zou kunnen gaan.

Rec-Play

Dit stukje heeft een uitgesproken demonstratief karakter. Het laat de speler toe een melodisch verloop min of meer 'in de stille ruimte' te tekenen. Die ruimtelijke tekening wordt als bewegingspatroon opgenomen, en vervolgens als een muzikaal melodisch verloop weergegeven. De opname en weergave verloopt cyclisch waarbij de afzonderlijke cycli door een gongslag en een bel-cimbaal worden aangegeven. De melodieën zijn driestemmig. Voor de instrumentatie worden Nej-fluiten gebruikt.

Prime Time

Deze slagwerkkompositie heeft als genererende kompositorische cel de overlapping van zuiver periodieke metra die in een irrationale verhouding tot elkaar staan. Priemgetallen bepalen de onderlinge verhoudingen der periodetijden. Is de resulterende ritmische structuur hier het vaststaande kompositorisch gegeven, dan ligt de interpretatieve vrijheid van de uitvoerder hier geheel op het vlak van de bepaling van de juiste noten (er worden slaginstrumenten voorgeschreven waarvan de toonhoogte bepaalbaar is) evenals van de dynamiek, die bepaald wordt door de inspanning van de speler.

Hammers

In dit stukje wordt een imaginair en hypotetisch cimbalon op de scene geplaatst. Toonhoogte is hier een functie van de versnelling van de beweging.

Lock ... Unlock

Dit stukje vormt eigenlijk een diptiek met twee kontrasterende panelen. Het eerste luik -Lock- berust op cyclische slagwerkpatronen waarin heel trage faseverschuivingen optreden. De herhalingstijden zijn in de kompositie vastgelegd, en verschillen slechts een fractie van elkaar. Een 'klassieke' kompositorische truuk dus die ook aan de basis ligt van heel wat zogenaamde minimal muziek.

In het tweede luik -unlock- wordt dit repetitief karakter volledig doorbroken en controleert de speler de periodetijden van de metrische cycli zelf.

Close

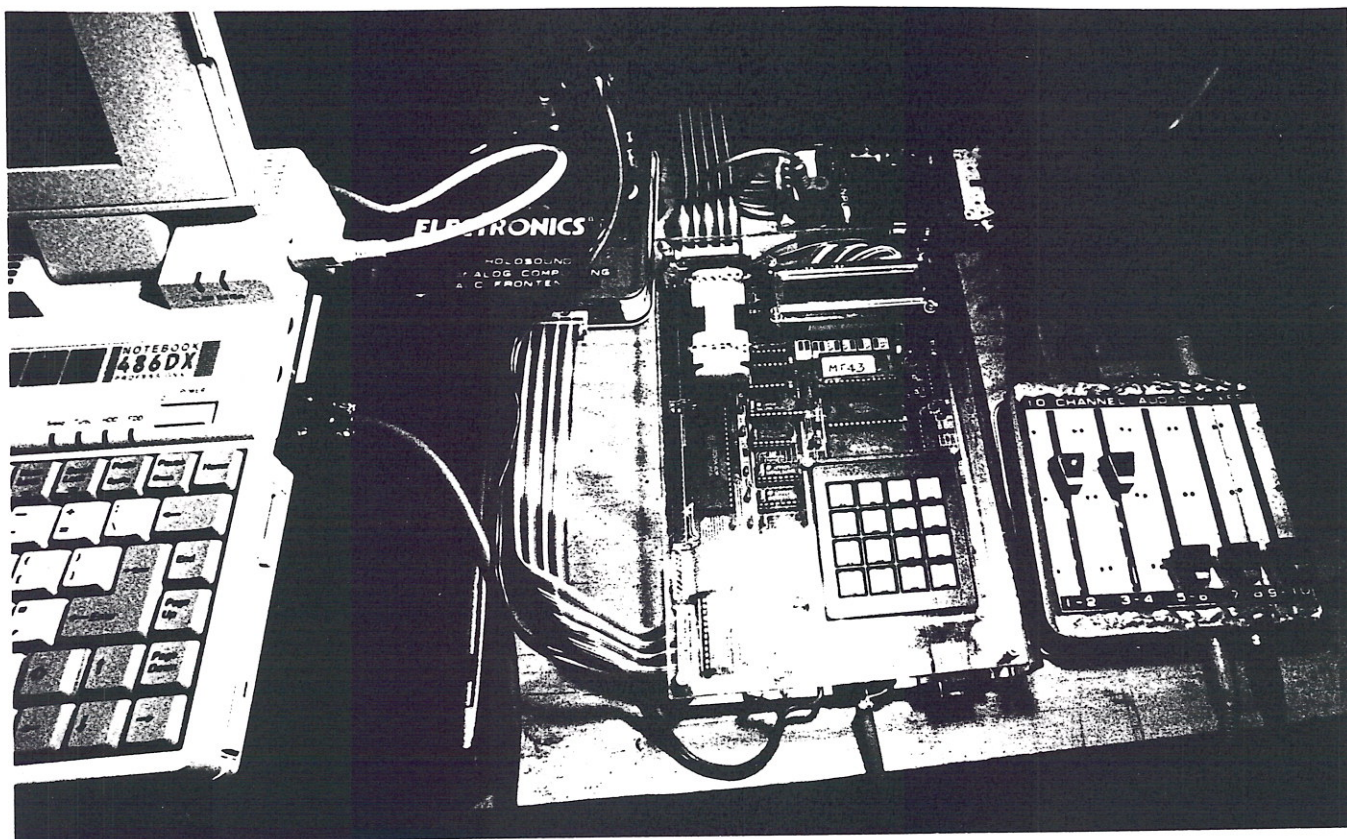
Dit is het sluitstuk van 'A Book of Moves'. Het vormt een antipode van de ouverture 'Open'. Het muzikale verloop is nochtans geen parametrisch omgekeerde imitatie van de ouverture, hoewel ze net zoals die ouverture, een zuiver retorische opbouw kent. Zij is immers opgebouwd als een grote dalende akkoorden progressie eindigend in een finaal achtstemmig unisono. Het verloop van deze progressie -in het bijzonder de ritmische structuur en de intervalgrootte van de melodische sprongen in de 8 stemmen- wordt volledig door de speler zelf bepaald.

Hoewel 'A Book of Moves' voor vele toeschouwers/luisteraars wellicht zou kunnen overkomen als een toch minstens met dans verwante produktie, is dit toch een fundamentele misvatting. Zoals allicht duidelijk zal zijn uit deze studie gaat het wel degelijk in elk onderdeel van het stuk om een bespeling van instrumenten. Dat deze misvatting kan ontstaan is evident, aangezien het publiek nu eenmaal niet gewoon is concerten bij te wonen waar de uitvoerders eigenlijk niet veel meer lijken te doen dan wat min of meer expressief over de scene te bewegen.

We kunnen er echter niet genoeg de nadruk op leggen, dat een dergelijke implementatie van een onzichtbaar instrument zelfs nauwelijks voor toepassingen binnen de wereld van de dans bruikbaar is, en wel omdat dansers vanuit hun vorming en opleiding, steeds de muziek achternahollen. Hun bewegingen staan in functie van de muziek. Zij genereren deze niet. Dansers produceren gekonfronteerd met dit instrument -en uiteraard hebben we dit bij verschillende gelegenheden uitgeprobeerd- slechts de meest triviale sonore resultaten. Zij weten niet om te gaan met het bewust hanteren van motoriek in functie van de klankproduktie. De geproduceerde klanken worden bij hen grotendeels accidenteel, louter sekundair gevolg. Ze slagen er niet in hun beweging in functie van een klankresultaat te formuleren. Ook de aangeleerde 'elegantie' die de danstechniek toch kenmerkt, staat een behoorlijk muzikaal resultaat ernstig in de weg. Hierdoor immers worden bewegingen steeds soepel en afgerond uitgevoerd, terwijl het precies de motorische diskontinuiteiten zijn die de grootste muzikale en expressieve profilering toelaten.

In de enscenering van 'A Book of Moves' wordt normaal gezien een volledig uitgeschreven toneelbelichting voorzien. Deze strekt er uitsluitend toe, de muzikale sfeer te versterken. Zij vormt hier een ondersteunend retorisch element. Het stuk kan overigens ook uitgevoerd worden zonder deze belichtingen.

Een laatste verwarring die bij sommige luisteraars soms kan ontstaan, is dat gedacht wordt dat het hier gaat om een soort bewegingsgestuurde sequencer. Dat dit principieel onjuist is, zal wel duidelijk zijn uit ons verhaal.



HOOFDSTUK 4: Perspektieven

4.1.: Experimenteel onderzoek: een verdere wegverkenning

Het zou van een grenzeloze overmoed getuigen te beweren de problemen voor de instrumentenbouw van de toekomst, of zelfs maar van een mogelijke toekomst, met de realisatie van de hardware en software voor 'A Book of Moves', te hebben opgelost. Het grootste werk ligt beslist nog in de toekomst. Hooguit hebben we een weg aangegeven waarvan de bewandeling tot de realistische mogelijkheden zou kunnen behoren.

In het volle besef daarvan, willen we hier, in enkele paragrafen, een overzicht geven van de technische mogelijkheden die verder kunnen worden onderzocht en ontwikkeld. Een onderzoek als het onderhavige, is uiteraard nooit echt volledig afgesloten en de weergave ervan in deze studie is dan ook niet meer dan een momentopname. Daarom zijn op het ogenblik dat deze studie wordt gelezen wellicht reeds stappen naar verdergaand onderzoek gezet zodat deze weergave wellicht op enkele punten reeds door de werkelijkheid is achterhaald.

We zullen beginnen met de kleinste stappen vertrekkend van en aansluitend bij het hier gepresenteerde onderzoek, om uit te komen bij reeds heel wat verder af liggende mogelijkheden die wellicht ook de onze überhaupt te buiten gaan.

4.1.1.: Digitale signaalverwerking

- DSP-processors voor het uitvoeren van FFT's en andere signaal-analyses.

In ons opzet hebben we, gemotiveerd door de argumenten aangehaald in hoofdstuk 3, het gebruik van digitale signaalverwerkingsprocessor-chips voor toepassing in ons projekt voor een onzichtbaar instrument afgewezen. Hun toepassing, binnen de beperkingen gesteld door het gehele opzet van het systeem, werd door ons als een 'overkill' gevoeld. Gezien echter het feit dat digitale elektronika om intrinsieke redenen wel steeds goedkoper zal blijven dan de hoge precisie analoge componenten die wij uiteindelijk hebben voorgesteld en gebruikt, en dat bovendien de economische kloof dag na dag nog vergroot door de snelle prijsdalingen voor digitale processor- en geheugen-chips, zal in een niet erg verre toekomst, een volledig digitale technologie onze analoge komputer wellicht toch kunnen vervangen.

Een voordeel daarvan zou onder meer zijn, de veel flexibeler programmeringsmogelijkheid. Immers, van het hier voorgestelde systeem werden in de loop der jaren door ons inmiddels reeds zo'n 20 verschillende hardware-versies gebouwd... De totale kostprijs daarvan

is uiteraard ook niet niks. Wanneer deze schakelingen middels voldoende snelle en krachtige DSP-processoren gebouwd waren geweest, kon dit tijdrovende doorlopend herbouwen van projekten en experimentele schakelingen geheel achterwege blijven en vervangen worden door een wellicht even arbeidsintensieve, maar geen materiaal-kosten met zich brengende, update van de software. Uiteraard zal ook dan de hardware van tijd tot tijd wijzigingen ondergaan, maar gezien de digitale natuur ervan, is dit beslist een heel stuk economischer.

4.1.2.: Hogere frekwentiegebieden

- Hogere draaggolf-frekwentie voor het ultrasoon-systeem

Binnen de door ons voorgestelde hardware, bleek het niet erg interessant van veel hogere draaggolffrekquenties voor het ultrasoonsignaal gebruik te maken. Nochtans moet toegegeven worden dat een verhoging van de draaggolffrekwentie naar het bereik 200 tot 400KHz grote voordelen met zich zou kunnen brengen. Immers, de Dopplerfrekwenties waarvan gebruik wordt gemaakt in ons opzet, komen dan enkele oktaven hoger te liggen. Het is op technische gronden een vaststaand feit, dat frekwenties een korte meettijd vergen naarmate ze hoger zijn. Het direkte gevolg van dit feit is dat het onzichtbare instrument veel responsiever zou kunnen worden gemaakt bij toepassing van deze hogere frekwenties.

Dat de noodzakelijke meettijd voor de meting van heel lage frekwenties groter is dan voor hoge frekwenties valt eenvoudig in te zien wanneer we ons voor ogen houden dat een frekwentiemeting berust op een meting van de periodetijd van een golf. Daartoe moeten minstens toch enkele periodes kunnen worden gemeten. Het meten van deze tijden voor trage trillingsverschijnselen duurt dan uiteraard -wil men een gelijke precisie kunnen behalen- langer dan het geval is bij snelle trillingsverschijnselen.

Vooraf onze algoritmes voor de berekening van de bewegingsversnelling -die in onze huidige versie eigenlijk welhaast primair kunnen worden genoemd aangezien ze uitsluitend berusten op de berekening van het verschil tussen snelheidsmetingen op slechts twee tijdstippen-, zouden daarbij veel te winnen hebben.

Vergeet we immers niet dat de motorische versnelling beter dan menige andere parameter van de menselijke expressieve beweging, expressief relevante informatie insluit.

Een probleem dat echter eerst en vooral zal dienen opgelost te worden -en dat los staat van de eisen met zich gebracht door de hogere snelheden voor de analoge en digitale hardware- heeft te maken met de intrinsiek veel sterkere directionaliteit van geluiden in dit erg hoge frekwentiegebied. De kommerciële gangbare transducers voor 200kHz hebben een afstraalkarakteristiek waarvan de openingshoek de 5

graden zelden te boven gaat.

De akoestiek leert ons dat een geluid directiever wordt afgestraald naarmate de frekwentie ervan hoger komt te liggen. Dit feit zal iedereen wel al hebben opgemerkt: bas-geluiden zijn erg moeilijk precies te lokaliseren, terwijl hoge geluiden een veel betere lokalisatie van de geluidsbron toelaten.

Een laatste -maar vandaag niet zo moeilijk op te lossen- probleem is, dat bij hogere draaggolffrekquenties ook evenredig snellere analogo-digitaal konvertors gebruikt dienen te worden. Volgens het Nyquist-teorema immers, hoort bij een te verwerken signaal van 400kHz reeds een bemonsteringsfrekwentie van minimaal 800 kilosamples per seconde. Dankzij de recente opkomst van de zeer snelle chips voor digitale beeldverwerking (ontwikkeld met het oog op de invoering van HDTV) is dit echter niet zo'n bezwaar, wanneer we ons dan tenminste willen beperken tot een resolutie van niet meer dan 8 bits.

4.1.3: Mikrogolftechnieken

In ons onderzoek werden de mogelijkheden die het gebruik van elektromagnetische radar technologie met zich brengt, ver van volledig onderzocht. Wij hebben het onderzoeksgebied verlaten omwille van:

- 1.- de lage signaal/ruisverhoudingen die we met onze middelen en in ons beperkt laboratorium slechts konden bereiken.
- 2.- de veel grotere direktionaliteit van de elektromagnetische signalen in het gebied 10-20GHZ.

Door deze laatste eigenschap, klonken de door ons onder gebruikmaking van dit soort radartechnologie geproduceerde geluiden veel minder controleerbaar. Het instrument kreeg teveel het karakter van een positioneel systeem, het bezwaar dat we ook al naar voor brachten tegen bijna alle optische (infrarood) implementaties van een onzichtbaar instrument.

Hiermee willen we evenwel niet de mogelijkheden verbonden aan het gebruik van deze technologie uitsluiten. Er zijn immers ook enorme principiële voordelen aan verbonden: door het elektromagnetisch karakter van de golven, planten deze zich voort aan de lichtsnelheid, zoals men weet aanzienlijk sneller dan die van het geluid. Dit maakt in principe een veel sneller reagerend instrument mogelijk.

Experimenten zouden kunnen worden opgezet met over een beperkte hoek (60 graden in ons opzet) aftastende systemen. De afbuiging van de golven zou daarbij dan langs radio-elektrische weg dienen te

verlopen en dus niet mechanisch zoals gebruikelijk bij klassieke radartoepassingen. Op het vlak van de hardware zou geëxperimenteerd moeten worden met meer geavanceerde technieken voor signaaldiskriminatie binnen een ruiskontekst. Daarover bestaat een uitermate uitgebreide maar wiskundig erg complexe technische literatuur.

Last but not least, dienen experimenten opgezet te worden met de meest geavanceerde types van Gunn-diodes of andere onderdelen waarmee deze golven kunnen worden opgewekt en uitgezonden. Zoals we ook elders reeds betoogden, bleken dergelijke onderdelen voor ons onverkrijgbaar te zijn.

4.1.4.: Parallele verwerking

Gezien het 3-dimensionele opzet van het Holo-sound concept, ligt het nogal voor de hand, voor elke bewegingsdimensie (en dus voor elk transducersignaal) een verschillend systeem te bouwen. De drie systemen (drie Multitrax MT-boards zoals beschreven in hoofdstuk 3 bijvoorbeeld) zouden dan hun werk parallel kunnen doen in real-time. Dit zou dan uiteraard reeds een snelheidswinst met een faktor 3 betekenen. We kunnen echter nog veel verder gaan, en ook processoren inschakelen voor de berekening van de muzikale parameters en de instrumentale stemmen. Ook die zouden we parallel kunnen laten werken. In een minimalistisch opzet is een echte 'bus-sharing' hierbij nog niet eens noodzakelijk! De interactie tussen de vier of meer komputersystemen kan volledig met hardware interrupts worden afgehandeld.

Naarmate de muzikale eisen echter hoger gesteld worden, en de algoritmes voor de syntax, en -in een verder stadium- ook voor de klankopwekking zelf geraffineerder worden, zal de noodzaak zich opdringen van een gemeenschappelijke komputerbuss gebruik te maken voor de kommunikatie tussen de verschillende parallel verlopende processen. Dit soort technologie is op dit ogenblik trouwens volop in ontwikkeling, onder meer omdat zij een noodzakelijke voorwaarde is voor 'real-world' werk met software-matige technologieën zoals onder meer bvb. de neurale netwerken.

Het is overigens een vergissing te menen dat de nodige hardware voor parallel verwerkende systemen volledig buiten de mogelijkheden van een individuele onderzoeker zou vallen, wat nogal eens wordt beweerd. Microprocessoren zijn vandaag de dag reeds gezakt tot prijsniveaus beneden de 40fr. per processor...

4.1.5: Analoge signaalverwerking

Ook aan onze voorstellen op het vlak van de signaalverwerking met analoge computers zou nog heel wat kunnen verbeterd worden.

Zo kan bijvoorbeeld worden gedacht aan een vorm van diskrete Fourier transformatie uitgevoerd in analoge techniek, waarbij gebruik zou worden gemaakt van vertraginglijnen (deze kunnen gezien worden als een soort geheugen voor analoge signalen) die op meerdere programmeerbare plaatsen kunnen worden afgetakt. Konkreet gesproken denken we hier aan de mogelijke toepassing van de zogenaamde 'emmertjes-geheugen' chips die voorzien zijn van aftakkingen op gespecificeerde tijdsafstanden.

In het Engels worden deze onderdelen steeds aangeduid met de benaming 'Bucket Brigade Delay Line'. De werking van deze analoge geheugen chips berust op een kaskade van miniatuur condensatoren die in de maat van een centrale klok hun lading doorgeven aan een volgende condensator. Elke condensatorlading is een geheugenelement voor een analoge momentane spanning. De chips kunnen verkregen worden o.m. bij Panasonic onder de typenummers MN3005-3101. Zij omvatten, voor deze types, 1024 tot 4096 geheugenelementen. De haalbare signaal/ruisverhoudingen liggen tussen 75dB en 85dB.

De tijdsafstanden zijn enerzijds bepaald door de hardware en anderzijds ook programmeerbaar via sturing van de klokfrequentie van de analoge vertraginglijn. Digitale programmering van de vertraginglijn behoort dus tot de mogelijkheden.

Binnen zo'n opzet moet het mogelijk zijn signalen in het frekwentiedomein op te zetten naar het tijdsdomein en vice versa en dit dus zonder gebruik te maken van digitale technieken.

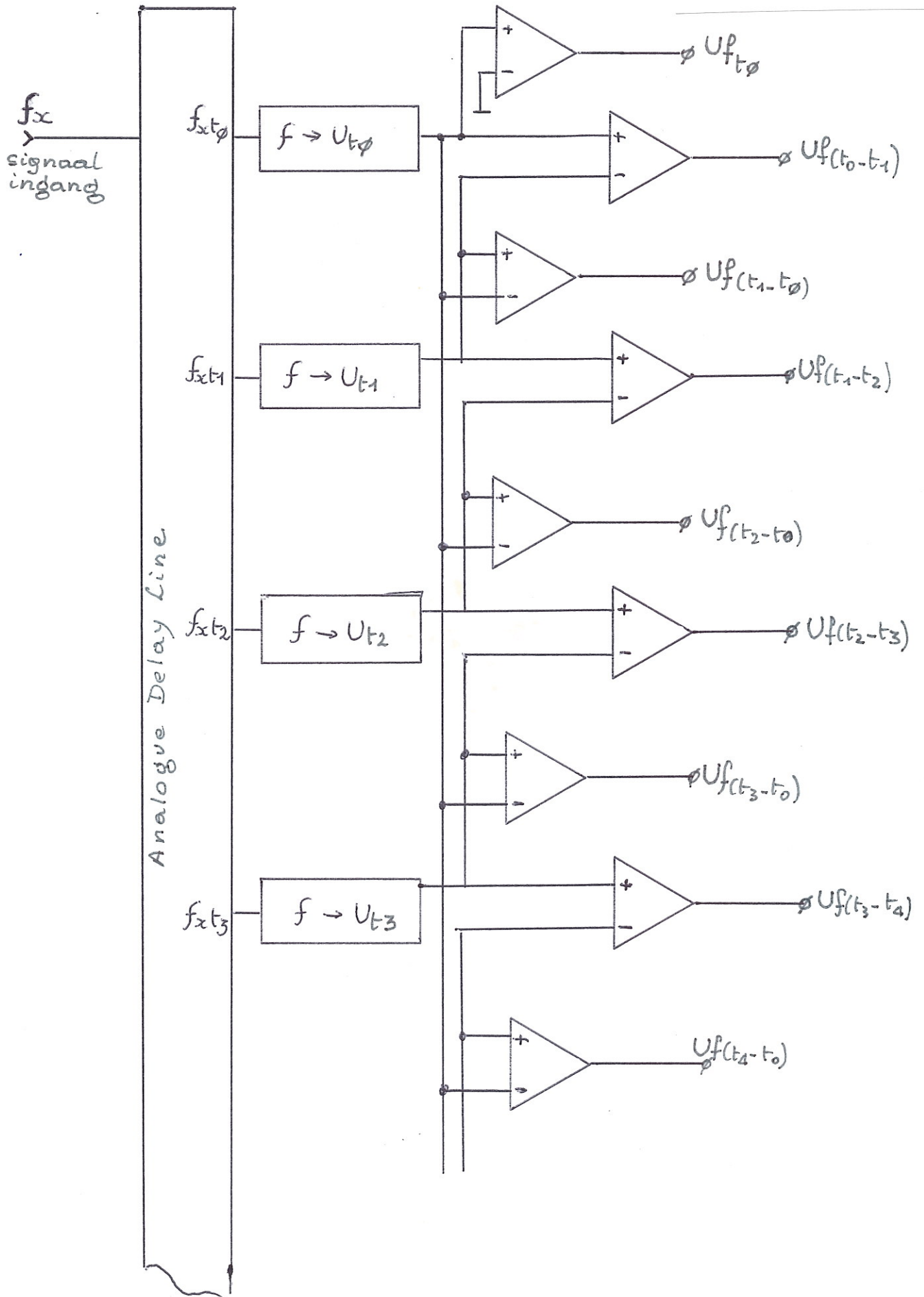
In een verder stadium valt dan te overwegen om de gehele analoge computer in al zijn functies en patches digitaal programmeerbaar te maken. Voor de patches kan dit eenvoudig door gebruik te maken van digitaal stuurbare analoge schakelaars (CMOS-switches). Ook programmeerbaarheid van de integratietijdkonstanten kan langs die weg gebeuren.

Om het systeem gebruikersvriendelijk te maken, moeten ook spanningsgestuurde versterkers op in- en uitgangen toegepast worden, waarmee dan functies zoals 'autoranging' geïmplementeerd kunnen worden. Het instrument kan dan gebruikt worden zonder dat ook maar enige afregeling van de hardware de gebruiker ten laste wordt gelegd.

De achterliggende idee komt tamelijk uitvoerig aan bod in volgende technische publikatie van Analog Devices: SCHEINGOLD, Daniel H.(Red.), 'Analog-digital conversion notes' , p.67-71.

Schematisch gezien zou zoiets er kunnen uitzien zoals op volgende vluchtige schets, waarbij alle verbindingen dan als programmeerbaar moeten worden gezien en de differentieelversterkers als willekeurig programmeerbare analoge rekenblokken. Elke uitgang kan naar een ingang van de (digitale) komputer worden gebracht voor digitale signaalverwerking. Aangezien hier ook versnellingsinformatie reeds in analoge vorm beschikbaar kan zijn, kunnen de taken van deze komputer nu op de instrumentale definitie en dus ook op de muziek zelf gekoncentreerd worden.

Praktisch kan een dergelijke schakeling integraal opgebouwd worden als een insteekkaart voor een EISA-bus P.C. Een ontwerp -bedoeld om te worden gebruikt in een 80486 - 50MHz laptop komputer- is in voorbereiding.



4.2.: De Muzikale Naaktheid

4.2.1.: de ongemedieerde naaktheid

Het onzichtbaar muziekinstrument plaatst de uitvoerende muzikus in zijn blootje.

Het instrument als objekt fungeert bij de normale muziekbeoefening uiteraard eerst en vooral als hulpmiddel voor die muziekbeoefening, als medium waardoorheen de muzikale uitdrukking kan plaatsvinden. Anderzijds echter kan het evenzeer fungeren als een soort psychologisch scherm, waarmee een zekere afstand tussen muzikus en publiek kan worden geschapen.

Het instrument als objekt introduceert via zijn bemiddeling een zekere mate van objektivering van de expressie van de muzikus. Het scherm dat wordt opgetrokken tussen de onmiddellijkheid van de uitdrukking van de muzikus, de direktie van diens motorische impuls en dus van diens lichamelijkheid, kleedt in deze zin de muzikus. Zonder instrument staat hij figuurlijk in zijn blootje.

Dat dit idee ook met een beleefde psychologische werkelijkheid overeenstemt, blijkt uit talloze gedragingen van podiumkunstenaars die niet zijn ingegeven door technisch instrumentale noodzaak, maar precies door een zekere angst om naakt -ongemedieerd dus- voor het publiek te verschijnen. Zo is de noodzaak voor vele instrument-loze publieke uitvoerders en sprekers om over een mikrofoon te beschikken, bijna steeds ingegeven door de angst om 'naakt' voor een publiek te verschijnen. De mikrofoon is een hou-vast. De bijna onzichtbare 'dasspeld'-mikrofoons en dergelijke, en dit zeker wanneer het gaat om een draadloze uitvoering, hebben dan ook zelden of nooit sukses bij niet echte 'teaterbeesten'. Deze laatsten, geen musici maar acteurs, zijn dan ook gekenmerkt door een vorm van psychologisch 'exhibitionisme' die precies voorwaarde is voor hun acteurschap: de bereidheid volstrekt ongemedieerd en zonder schermen voor een publiek te staan.

Hoewel 'volstrekt onwetenschappelijk', in die zin dan dat we er hoegenaamd geen onderzoek over hebben verricht, meen ik doorheen mijn toch vrij lange ervaring in het werken met musici en musici in spe, een zeker verband te kunnen zien tussen de karakteriële en psychologische ingesteldheid wat dit aspect betreft en de keuze van het soort muziekinstrument. Zo valt het me telkens weer op dat 'typische' organisten (d.w.z. organisten die niet ook klavecimbel enzomeer willen spelen), de meest 'geklede' karakters hebben. Van alle traditionele westerse instrumenten is het orgel inderdaad het sterkst gemedieerde en meest indirekte muziekinstrument. Zangers vertonen, aan het andere eind van dit spektrum, de grootste naaktheid. Ze zijn extravert, eerder grillig, egocentrisch en teatraal. Maar zelfs zangers hebben vaak nood aan een scherm, en hierbij komt ons onwillekeurig, het karikaturale zakdoekje van Pavarotti voor de geest... Dit zakdoekje fungeert als een minimaal schaamlapje en ook al is dit bij Pavarotti karikaturaal, een parallel ervan vinden we bij heel wat zangers in de vorm van de letterlijke

vaak overdreven kostumering waarmee zij ten tonele verschijnen.

In de mate waarin in deze analyse een zekere waarheid zou schuilgaan, is deze van grote betekenis voor het ontwerp van een onzichtbaar muziekinstrument en voor de penetratiekansen van zo'n ontwerp in het praktische muziekleven. Immers, het opzet zal alleen aantrekkingskracht uitoefenen op de meest 'exhibitionistische' musici onder ons. Anderzijds kan het ook een bevrijdend effect hebben en het musiceren 'eerlijker' en ongemedieerder maken. Immers het kan mogelijkheden openen voor mensen die noch begiftigd zijn met iets wat men als een goede stem zou kunnen omschrijven, noch met de technisch-ambachtelijke kunde van een of ander solistisch instrument, om toch op een volwaardige manier muzikaal expressief op het podium te komen.

4.2.2.: de fysieke naaktheid

Is het onzichtbare instrument niets verhullend en zet het de muzikus figuurlijk in zijn blootje, dan blijkt op grond van de analyse van de technische werking van het instrument, het nog het best te werken, wanneer deze muzikus ook letterlijk in zijn blootje speelt.

Het heeft me niet veel onderzoek gekost om hier via vergelijkende metingen en auditieve evaluatie achter te komen. De redenen daarvoor liggen trouwens nogal voor de hand. Hoe meer immers het bewegend lichaam reflektierend het is voor geluid, hoe beter het echosignaal door de apparatuur kan worden ontvangen en hoe groter ook de resolutie zal zijn van de gegenereerde parametrische bewegingsinformatie. De naakte huid is de eenvoudigste manier om het lichaam zo reflektierend mogelijk te maken. Alleen een natte of glad ge-oliede naakte huid doet het nóg beter...

Precies hierin ligt de reden waarom in de muziekteaterversie van 'Holosound', de uitvoerder halfnaakt ten tonele verschijnt. Alleen de benen zijn gehuld in een speciale broek gemaakt uit een bijzonder glad soort textiel. Het transducersysteem is -gezien de opstelling ervan in een imaginaire geometrische tetraeder- het meest gevoelig in het zwaartepunt van de ruimtelijke figuur gevormd door de transducer-punten. Dit zwaartepunt valt grosso modo samen met het bovenlichaam van de uitvoerder, dat deel van het lichaam dus waarin motorische expressie precies de grootste differentiatie kan vertonen.

Voor de artistieke realisatie van 'A Book of Moves' echter -en dit heeft in eerste plaats te maken met cultureel-maatschappelijke factoren op grond waarvan naaktheid nu eenmaal niet steeds als naaktheid (vizuele 'stilte', maximale teatrale eerlijkheid) wordt geïdentificeerd- hebben we gebruik gemaakt van kostumes in metaalachtig glad textiel. Het ziet eruit als een soort astronautenpakjes uit een of andere science fiction film.

In het ontwerp van mijn kollegas en vrienden Michel Waisvisz en Joel Ryan is dit probleem geheel ondervangen precies omdat zij niet perse vanuit de muzikale naaktheid vertrokken in het ontwerp voor 'The Hands'.

Hun instrument is immers, zoals de kommerciele tegenhanger, de 'Power Glove', fysisch en objekt: het IS een 'kledingsstuk'. Immers, het wordt als 'handschoenen' gedragen en gespeeld. Bovendien is het via (dikke) kabels -een navelstreng- met de apparatuur verbonden.

4.3.: De automatizering van de muziek

Wie kent niet de droom van de student waarin hij -liefst 's nachts- door een muze wordt bezocht, die hem dan alle kennis waarover hij zou willen en moeten beschikken in de hersenen inplant. Of, de droom van de pil die hem bij inname alwetend zou maken?

Ook musici blijven van dergelijke dromen natuurlijk niet gespeend. Ook zij zouden maar wat graag over een tovermiddelje kunnen beschikken waarbij zij automatisch elke hen voorgezette partituur, zonder de minste technische moeilijkheden zouden kunnen vertolken. Of, in het geval van de komponist, het middelje waarmee ze over de mogelijkheid zouden kunnen beschikken hun muzikale concepten gewoonweg gekoncentreerd te denken, waarbij ze al naar gelang het geval, automatisch ook zouden worden verklankt op een voor derden hoorbare wijze, of zouden worden genoteerd, op een voor derden leesbare (en speelbare) wijze.

De werkelijkheid is echter anders ingericht en maakt vele jaren oefening en studie noodzakelijk om tot intellectuele en technische beheersing van wat dan ook te kunnen komen.

Hoewel de mens het werktuig uitvond om de beperkingen van zijn lichamelijke mechanika te kunnen overstijgen, blijft dit werktuig in wezen toch nooit meer dan een proteze, omdat het nooit tot een werkelijk verlengstuk van het eigen lichaam kan worden. Anders gesteld, het werktuig is en blijft een vreemd lichaam waartegenover in meer of mindere mate toch steeds afstotingsverschijnselen optreden. In hoge mate heeft dit te maken met het feit dat het werktuig tegenover zijn gebruiker een veel geringere volgzaamheid, of nog, een grotere weerspanningheid vertoont dan het eigen als ongemedieerd ervaren lichaam. Hoe dichter het werktuig bij de gebruiker staat en hoe groter de vertrouwdheid van de gebruiker met het werktuig, hoe minder uiteraard deze weerspanningheid als een bron van frustratie ervaren wordt.

De automaat is vanuit deze optiek de proteze die zich van de drager heeft losgemaakt en een eigen leven gaat leiden. De fascinatie van de mens voor de automaat, vooral dan wanneer deze taken uitvoert die normaal gezien niet alleen door mensen worden verricht, maar die bovendien behoren tot het gebied van de menselijke expressie, is reeds zo oud als de technische mogelijkheden van de mens zelf. De wil dergelijke automaten te bezitten heeft te maken met de drang over middelen te kunnen beschikken die eigen deficienties teniet kunnen doen.

Het overweldigend succes van grammofoonplaten en geluidsopname, en vooral weergave, sedert het begin van de twintigste eeuw, mag dan ook niet alleen worden verklaard uit de angst voor de stilte en de ontdekking van de eigen leegte, zoals dat o.i. ten onrechte o.m. door Adorno werd voorgesteld.

In plaats van deze automaten te zien als negatief en hebbende de plaats ingenomen van live-musici, een vizie die uitsluitend is ingegeven vanuit een conservatieve weemoed naar een bepaald verleden van een bepaalde maatschappelijke klasse, kan men er ook iets erg positiefs in ontdekken en eraan werken dat precies dit aspekt kan ontwikkeld worden. Daarvoor echter is het nodig de macht over de automaat de nemen, in extremis, door hem naar eigen en maatschappelijke behoeften te bouwen. Omdat in deze tijd dit bouwen een ontwikkeling doormaakte waarbij het meer en meer een kwestie van software wordt, is dit meer dan de laatste eeuw het geval was, in principe ook een realistische mogelijkheid geworden. Immers, een ambachtelijke werkplaats is in zo'n toekomst nauwelijks nog nodig. Alleen een komputer-sowieso vandaag reeds aanwezig in een meerderheid van de Westeuropese gezinnen- en een stel gezonde en tot logisch denken in staat zijnde hersenen volstaan om een komputerprogramma te schrijven en zo een instrument te bouwen.

Is de beweegreden voor een dergelijk grijpen naar techniek om los te komen van de beperkingen van de menselijke techniciteit, niets meer dan luiheid? Als luiheid daarbij betekent meer plaats voor spel, dan kan dat best zo zijn. Muziek kan toch beter en met meer plezier gespeeld worden wanneer de speler boven de techniek verheven is.

In een interessante en ook controversiele voordracht (15 februari 1985, Logos Koncertstudio te Gent) verdedigde de Nederlandse komponist Leon VAN NOORDEN de originele stelling dat de automatizering van de muziek een aanvang nam vanaf het ogenblik dat de mens begon muziek te maken voor anderen dan voor hemzelf. Als eerste faze in de automatizering van de muziek ziet hij dan historisch, de invoering van muzikale standaardizatie: het toonsysteem, ritmiek, muzieknotatie, muzikale stijl... De produktieketen voor muziek werd alsmaar langer (en daarmee de lijst voorschriften en standaarden) en moet, volgens hem uitmonden in de algehele automatizering van de muziek.

4.3.1: Interpretatie zonder techniek?

Zonder in de 'science fiction' van de zogeheten 'virtuele werkelijkheid' te vervallen, lijkt het ons inderdaad mogelijk een toekomstbeeld voor de muzikale uitvoerder te schetsen, waarbij deze zich in de allereerste plaats zou kunnen toeleggen op de interpretatie, of, noem het de retoriek, van de muzikale uitvoering. De ambachtelijke beheersing van het muzikale werktuig, resultaat van een leerproces waarvoor heel wat geduldig oefenen noodzakelijk is (er moeten immers een heleboel senso-motorische automatismen aangeleerd worden) kan gebruik makend van de hier beschreven technologie, en mits we ons beperken tot het gebruik van mechaniseerbare muziekinstrumenten, overbodig worden gemaakt.

Reeds in 1929 voorspelde de wat dit betreft op zijn minst vizionaire dirigent Leopold Stokowski iets in die richting:

It is only a few years before we shall have entirely new methods of tone production by electrical means...

Thus will begin a new era in music...

One wonderful feature of the new electrical instruments is, or will be, the practical absence of technical difficulty in playing them. There will be no long hours of practice every day, for electricity will do all the mechanical part. The performer will give musicianship, interpretation, variety of tone-colour and tone-volume and all the non-material side of music'.

(geciteerd in Percy A.Scholes, 'The Oxford Companion to Music', p.321)

Toen Stokowski deze woorden schreef, bestond er evenwel nog nauwelijks een technologische grond voor zijn bewering. Immers de komputertechnologie, vandaag toch beschouwd als een conditio sine qua non voor automatiseringsprojecten in welke sektor dan ook, moest nog uitgevonden worden.

Hiermee willen wij, noch Stokowski, zoals blijkt het uit citaat, echter geenszins gezegd hebben dat met behulp van dergelijke apparatuur waardevolle muzikale interpretaties mogelijk zouden zijn door personen zonder enige muzikale of algemeen kulturele vorming! Immers, een van de belangrijkste voorwaarden om tot een kultureel-maatschappelijk waardevol geachte interpretatie te komen van een muzikaal kunstwerk, is de aanwezigheid van een interpretatieve vizie op dat kunstwerk door de uitvoerder. Zo'n vizie vergt een ruime kulturele vorming, een specifiek muzikaal analytisch inzicht, een zeker empatisch vermogen en een deel expressieve en/of retorische competentie.

Deze laatstgenoemde elementen -door Stokowski eenvoudig samengevat in '*musicianship*'- kunnen met de hier beschreven technologie beslist niet overbodig worden gemaakt, waarmee dan niet gezegd wil zijn dat het überhaupt niet ooit zou mogelijk zijn hen overbodig te maken... Daarvoor zouden we evenwel eerst een 'grote stap voorwaarts' moeten kunnen zetten op het gebied van wat met een door modieus en spekulatief gebruik uitgeholde term 'artificiële intelligentie' genoemd wordt, een onderwerp dat we welbewust buiten deze studie hebben willen houden.

Alleen het zuiver ambachtelijke aspekt van het musiceren, dat deel dus dat alleen door technische inoefening kan verkregen worden, kan tot op zekere en beperkte hoogte, overbodig worden gemaakt zonder iets aan de potentiële artistieke waarde van een vertolking in te boeten. Ook Manfred Clynes, kwam na onderzoek op het gebied van de mikrostructuur van de muzikale expressie, tot een in algoritmes omzetbare theorie (een maakbare machine dus) waarvan hij stelt:

"In an age of personal computers, the programs developed can give access to creative interpretation to all so inclined without the need for physical musical skills."

CLYNES, Manfred, "Expressive microstructure in music linked to living qualities" , Sydney 1983, p.154.

Dit neemt uiteraard niet weg dat er allicht steeds wel musici -of wat dat betreft mensen in het algemeen- zullen zijn die er juist een groot genoeg aan beleven zich het ambachtelijke eigen te maken. Ook een doelgerichte zelfdiscipline kan immers lustverschaffend zijn. Zo heeft het bestaan van computers, tekstverwerkers en programmas zoals Ventura's 'Page Maker', niet meteen alle kaligrafen uitgeroeid.

Maar, gesteld dat het mogelijk wordt het ambachtelijke aspekt aan een automaat over te laten, kunnen we even onderzoeken hoe een en ander dan praktisch in het werk zou moeten gaan:

4.3.1.1.- Kodering:

De muzikale te vertolken partituur dient eerst in een door de komputer te verwerken formaat gekodeerd te worden. Deze technologie is sedert een tiental jaren vlot beschikbaar en kan onder andere via het Midi-formaat in een gestandaardizeerde vorm gerealiseerd worden. Het Midi-formaat, in deze gestandaardizeerde vorm, laat toe volgende muzikale parameters met een hoge graad aan precisie vast te leggen:

1. de metrische indeling
(rezolutie minstens 14 bit per tijds-eenheid)
2. de toonhoogte (de noot in een chromatische ladder)
(rezolutie 7-bit)
3. de aanslagsterkte van die toonhoogte of noot
(rezolutie 7-bit)
4. de toonhoogte afwijking t.o.v. die noot (Pitch-bend)
(rezolutie maximaal 14-bit)
5. het eventueel amplitudeverloop van die noot (Volume)
(rezolutie 7-bit)
6. de steminformatie (polyfonie) (Channel)
(rezolutie 4-bit)
(in principe kan maximaal 16-stemmig gewerkt worden, tenzij gebruik wordt gemaakt van dubbel uitgevoerde Midi-interfaces, waarbij het aantal stemmen telkens met 16 eenheden kan uitgebreid worden).

Deze parameters volstaan om muzikale eigenschappen zoals toonhoogte, ritme, frasering, dynamiek, tempo met grote precisie vast te leggen.

Wat middels Midi slechts node mogelijk is, heeft betrekking op het timbrale aspekt van de muziek. Weliswaar hebben bijna alle Midi-synthesizers, of het nu gaat om FM-synthese, PCM-synthese, additieve synthese of samplers, mogelijkheden tot parametrische programmering van de klankkleur-synthese-algoritmes, maar daarvoor dienen zogenaamde 'system exclusives' (machine specifieke midi-codes) gebruikt te worden, iets wat onvermijdelijk leidt tot een verstoring van de ritmische en/of metrische structuur van de muziek wanneer men er in real-time

gebruik wil van maken. Dit is gevolg van de inherente traagheid van het Midi-kommunikatieprotokol zelf. (De Baud-rate). Een tweede aspekt dat daarbij van belang is, is dat er geen enkele standaard bestaat met betrekking tot klankkleur tussen de diverse MIDI-apparatuur onderling. Dus, zelfs al zou Midi snel genoeg zijn voor een nauwkeurige klankkleurkontrolle in real-time, dan nog zou het ons fundamenteel nog niet zoveel verder helpen, omdat het ons zou overleveren aan de grillen van een enkele fabrikant en een enkel type synthesizer.

Wanneer we echter genoeg nemen met per stem selekteerbare en eventueel op voorhand programmeerbare klankkleuren, laat MIDI toe 7-bits te benutten voor wat we om een orgel-term te gebruiken, 'registratie' zouden kunnen noemen. Met deze 7 bits kan dan telkens uit een uit 128 elementen bestaande verzameling gekozen worden.

4.3.1.2.- Verklanking:

4.3.1.2.1.- Elektronische instrumenten

Voor de verklanking van de gekodeerde partituur ligt het gebruik van elektronische muziekinstrumenten -synthesizers of samplers- uiteraard het meest voor de hand. Deze oplossing wordt evenwel in de ('ernstige') concertpraktijk door het publiek zelden aanvaard, zeker wanneer het gaat om bestaande en oorspronkelijk voor akoestische muziekinstrumenten geschreven muziek. Walter (Wendy) Carlos' 'Switched on Bach' heeft nooit de echte Barok-liefhebber kunnen overtuigen... Ook de tegenwoordig tamelijk gesofistikeerde instrumenten die gebruik maken van meerdere samples per klank/noot (vb. EMU-emulator, Akai S1000 e.d.), worden door het klassieke concertpubliek als 'elektronische' namaak afgewezen. We kunnen het daarin overigens geen ongelijk geven. Het probleem van de elektronische klanksynthese is immers tot op de dag van vandaag nog steeds niet afdoende opgelost en we beschikken nog niet over werkelijk werkende matematische modellen die ons zouden toelaten een akoestisch geluid in zijn tijdsverloop als een algoritme vast te leggen.

De euforie van de jaren 50, -toen de o.i. 'platonische' additieve syntese opgang maakte, sloeg over in de euforie van de jaren '70 toen de FM-synthese werd gepropageerd, maar beide syntese modellen bleken toch niet de echte sonore rijkdom van de akoestische klank te kunnen evenaren. In de jaren '80 dacht men dan het probleem te kunnen omzeilen door dan maar rechtstreeks akoestische geluiden te gebruiken via de zogenaamde samplers, maar ook dit ging de luisteraar -die kritischer bleek dan men had gedacht- danig op de zenuwen werken dat men stilaan ging inzien dat op dit gebied nog bergen werk te verzetten zijn.

Het op dit ogenblik enige valabele alternatief is dus de werkelijke akoestische instrumenten zelf te gebruiken voor de muzikale output. Nu stelt dit uiteraard het probleem van de mechaniseerbaarheid van reeds bestaande en ingeburgerde akoestische muziekinstrumenten.

Op dit gebied werden de laatste jaren door vele fabrikanten en vooral ook door vele komponisten met kompetentie op het gebied van hardware en software, grote sprongen voorwaarts gemaakt.

Opdat een muziekinstrument, rekening houdend met de huidige stand van de technologie, volwaardig mechaniseerbaar zou zijn moeten alle handelingen die een uitvoerder stelt en die mechanisch ingrijpen op het instrument (alle handelingen die deel uitmaken van de rechtstreeks klankbepalende bespeeling) via een technische konstruktie kunnen geschieden die zich bovendien leent voor elektrische bediening of sturing.

De sturing van elektrische toestellen vanuit elektronika is een, wat dit toepassingsgebied betreft athans, kwazi volstrekt opgelost gebied en kan dan ook, mits ontworpen volgens de regels der kunst, probleemloos gebeuren.

De omzetting van de elektrische stromen naar mechanische energie is -hoewel er een veelheid aan keuzemogelijkheden voorhanden is- iets moeilijker te realiseren, vooral omwille van de precisie-eisen te stellen aan de bouw van de mechanische onderdelen: veren, hefboompjes, vilten kussentjes, sluitringen, geleidestangen, schroefdraden, las- en hardsoldeerwerk.... Hier komt, zolang er geen sprake kan zijn van massaproductie, onvermijdelijk een deel traditioneel ambacht bij kijken.

Op grond van deze overwegingen menen we volgende tweeledige definitie voor de mechaniseerbaarheid van instrumenten te kunnen naar voor schuiven:

a. Vanuit de techniek: een instrument is volwaardig mechaniseerbaar wanneer alle mechanische input-parameters vanuit de normale uitvoerder op minstens equivalente wijze in een mechanisme kunnen worden geïmplementeerd.

b. Vanuit het resultaat: een instrument is volwaardig mechaniseerbaar, wanneer tussen de mechanische bespeling en de automatische bespeling via het mechanisme, door de erudiete luisteraar geen verschil kan worden waargenomen.

4.3.1.2.2.: Orgel

Het vanuit het organologisch werkingsprincipe eigenlijk eenvoudigst perfect te mechaniseren muziekinstrument is het klassieke pijporgel.

Immers, de toetsen hebben geen aanslaggevoeligheid en er is geen overdracht van vingerdrukmotoriek naar muzikale dynamiek mogelijk. Ook de klankkleuren worden via de registers op diskontinue wijze geselecteerd. Vele laat-romantische orgels hebben overigens reeds een hetzij pneumatische, hetzij (hoewel veel zeldzamer) elektrische traktuur.

We gaan hier dan wel even voorbij aan de mogelijkheden geboden door orgels met een zuiver mechanische traktuur, om toetsen 'half' in te drukken en registerschuiven heel geleidelijk open te trekken. Dit zijn immers eigenlijk 'fouten' van het instrument en de te bereiken klankeffekten behoren beslist niet tot de intentie van de oorspronkelijke bouwer. Ook gaan we voorbij aan de enkele en uiterst zeldzame instrumenten waarin een zekere luchtdrukmodulatie via het 'toucher' kon gebeuren.

Hoewel bovendien bijna alle moderne orgels tegenwoordig beschikken over een geëlektroniseerd registratiepaneel dat bovendien programmeerbaar is, zijn er -bij mijn weten- geen grote orgels die voorzien zijn van door elektromagneten stuurbare toetsen en pedalen waarbij deze sturing door andere elektrische signalen dan deze die van de klavieren afkomstig zijn, mogelijk is. Het feit dat orgels zich in meerderheid in kerken bevinden en dat ze bijna steeds tot het kollektief patrimonium behoren, zal wel niet vreemd zijn aan het gebrek aan experiment op dit gebied. De enige orgels waarvoor dergelijke systemen wel degelijk bestaan, zijn stuk voor stuk omgebouwde draaiorgeltjes en/of orchestrions, niet direkt de geschikte instrumenten om de competitie op interpretatief vlak aan te gaan met de organisten van dienst...

De Duitse instrumentenbouwer en experimenteel komponist Martin Riches heeft rond 1980 een klein pijporgel gebouwd, dat uitsluitend via een klein computersysteem (een Sinclair ZX81) bespeeld kon worden. Dit instrument is echter ook weer veel te beperkt om als een eerlijke partner in de 'strijd' te worden geworpen.

Het akkordeon, in menig opzicht verwant aan harmonium en zo aan het orgel, maakte reeds vroeg een volledige mechanisering door. Het is als pneumatisch stuurbaar instrument terug te vinden op de meerderheid van alle orchestrions uit het begin van de 20e eeuw. Ook elektrifikatie stelt hier geen enkel probleem. Een oplossing voor ons probleem is het echter niet door de welhaast onbestaande aanvaarding van het instrument door de 'erudiete' muziekkultuur.

4.3.1.2.3.: Klavichord & Klavecimbel

Ook klavichord en klavecimbel lenen zich in principe uitstekend voor mechanisatie. Het klavichord stelt daarbij wel iets meer problemen dan het klavecimbel, omdat hier de toetsen wel degelijk een drukgevoeligheid hebben die interpretatief door de uitvoerder wordt gebruikt.

Het klavichord is echter omwille van zijn uiterst beperkte dynamiek en geluidsterkte, in onze helaas geluidsverontreinigde tijd, een niet bepaald erg populair instrument. De mechanisatie ervan - ik heb zelf een aantal pogingen in die richting gewaagd - is ook problematisch omwille van de zeer strenge eisen die gesteld moeten worden aan de geruisloosheid van de mechanische onderdelen.

Het beste mechanisme waarmee de tangenten kunnen worden vervangen bleek bij mijn experimenten de elektromagneet te zijn. Een kant van de magneet wordt daarbij als een 'mes' of tangent uitgevoerd, die de snaar aantrekt. Uiteraard moeten dan wel ferromagnetische snaren worden gebruikt. Ook losse ijzeren tangenten beweeglijk geplaatst binnen elektromagneten kunnen worden gebruikt. Met deze laatste elementen is de 'Bebung' beter te realiseren.

Automatische klavecimbels anderzijds, zijn perfect bouwbaar, maar ikzelf heb er nog nooit een ontmoet. Het is ook weer niet zo'n courant instrument in de werkplaats van de experimentele komponist of instrumentenbouwer...

4.3.1.2.4.: Piano

Zo belanden we dan willens nillens bij het onvermijdelijke *piece de resistance* van de westerse muziek: de piano.

Inzake mechanisatie kan dit instrument op een lange geschiedenis terugblikken. De eerste 'player-pianos' werden geproduceerd rond 1890 en kenden een stormachtige ontwikkeling. Denken we maar even aan de pianorollen uit het begin van deze eeuw waarmee een reeds vrij getrouwe opname en weergave mogelijk was. De perfectie werd er evenwel nooit mee bereikt. Ervaren musici horen steeds het verschil tussen een rollenpiano-reproductie en de rechtstreeks vertolkende pianist, ook wanneer hetzelfde instrument wordt gebruikt. Dit hing, rekening houdend met de vroeger ter beschikking staande technologie, vooral samen met de te geringe resolutie op het vlak van de aanslagsterkte evenals op dat van de ritmiek in deze pianolas. De eerste automatische piano's werkten pneumatisch hoewel het type dat voor opname en weergave geschikt was, al heel vroeg vaak ook werd ge-elektrificeerd.

Wat juist de redenen waren is niet zo eenvoudig te achterhalen, maar pas sedert het begin van de jaren '80, valt een plotse herleving van de belangstelling voor het idee waar te nemen. Een van de belangrijke factoren daarin lijkt ons wel de 'ontdekking' te zijn van het werk van Conlon Nancarrow.

De vroegste publieke presentatie van het werk van Nancarrow in Europa greep wellicht zelfs plaats in Gent, bij Stichting Logos, in 1977, lang voordat o.a. György Ligeti de komponist in Duitsland ging propageren. Overigens is die 'ontdekking' in eerste plaats het werk van de Amerikaanse komponisten John Cage, Peter Garland, Gordon

Mumma, Charles Amirkhanian en James Tenney. De 'Studies for Player Piano' werden door Peter Garland uitgegeven in het midden van de jaren '70 in de reeks 'Soundings'. Het eerste publiek gebruik van Nancarrow's werk dateert reeds van de vroege jaren '60, waarin het door Merce Cunningham -via Cage's bemiddeling- gebruikt werd voor zijn moderne dansvoorstellingen.

Ref.: o.a.: MUMMA,Gordon, " Briefly about Nancarrow" in 'Soundings' book nr.4, p.1-6, California 1977, ed. Peter Garland.

De eerste plaatopnames verschenen bij Columbia (MS7222) en zijn al meer dan 20jaar niet meer verkrijgbaar. Wel zijn er nog exemplaren van alle vroege plaatuitgaven in het archief van Stichting Logos te Gent aanwezig.

Nancarrow schreef zowat al zijn muziek voor pneumatische player-pianos en het probleem dat zich daarbij natuurlijk stelde wanneer men deze muziek buiten zijn eigen woning in Mexico op concertpodia wilde ten gehore brengen anders dan via opnames, was geschikte instrumenten te vinden. Ook wanneer deze gevonden waren, groeide de sleet op de rollen uit tot een bezwaar tegen deze oplossing.

Begin van de jaren '80, zoals gezegd, beleefden we een hernieuwde belangstelling die ertoe leidde dat het pneumatische mechanisme werd ge-elektrificeerd en tevens voorzien van een zuiver elektronische aansturingmogelijkheid. In de eerste types (o.a. door de intussen ter ziele gegane firma 'Maranz' gefabriceerd) geschiedde dit via een opname van controlekodes op een gewone klankcassette. (Een 'Data-cassette' eigenlijk, zoals in de pionierstijd van de microprocessors voor home-computers en ontwikkelingssystemen gebruikelijk was).

Niet lang daarna, met de aanvaarding van de MIDI standaard in 1983, volgde de 'Midifikatie' van het instrument. In deze vorm wordt het thans o.a. gefabriceerd door de Amerikaanse bouwer Trimpin en door het Japanse Yamaha, dat de rechten van Maranz overnam. Een uiterst belangrijke vooruitgang is dat in bijna alle hedendaagse ontwerpen, de player-piano wordt uitgevoerd als een 'Vorsetzer', m.a.w. een toestel dat bovenop de toetsen van eender welke piano kan worden geplaatst en dat deze dus ook op de meest traditionele wijze bespeelt. Hoewel het woord 'vooruitgang' hier zeker moet worden gerelativeerd, vooral omdat de alleroudste vormen van pneumatische player-piano's in de 19e eeuw ook volgens dit concept werden gebouwd. Het grote voordeel van het 'Vorsetzer' mechanisme is dat het op een piano naar keus kan worden gezet, waardoor de kwaliteit van het bereikte klankresultaat een functie wordt van de gebruikte piano, daar waar bij vroegere als player-piano ontworpen instrumenten de kwaliteit van deze laatste bijna steeds bedenkelijk was.

Sedert een tiental jaren gaat de belangstelling voor dit mechanisme alsmaar in stijgende lijn. Niet alleen werd een groot deel van Nancarrow's muziek intussen in MIDI-code omgezet (o.a. door Clarenz BARLOW), maar heel wat komponisten schreven nieuw werk voor het mechanisme, waarbij ze zich niet langer gehinderd wisten door de zeer beperkte technische mogelijkheden van een pianist van vlees en bloed.

Vooral het mechanisme van Trimpin is bij machte de nuanceringsmogelijkheden van een echte pianist te overtreffen. Het Yamaha instrument heeft iets minder mogelijkheden en is in het ontwerp teveel gedacht als een toestel om bestaande muziek -zonder veel extremen kwa snelheid noch kwa dynamiek- mee te vertolken. We konstateerden zelf o.a. een enorm probleem, wanneer grote klusters moeten worden gespeeld: de voeding is er niet tegen bestand en geeft de geest...

Enkele voorbeelden van hedendaagse komponisten die erg veel voor player-piano schreven en schrijven: Clarenz Barlow, Trimpin, Alec Bernstein, Marc Carney, Richard Teitelbaum, Eberhard Blum, Martin Riches, Peter Garland, Tom Johnson, Alistair Ridell...

Vooral ook bij komponisten die gebruik maken van algoritmische kompositietechnieken, is het instrument erg in trek. Immers het is niet omdat men elektronika voor kompositie wil gebruiken dat men ook in elektronische muziek geïnteresseerd zou moeten zijn. Vroeger was dergelijke koppeling voor velen eerder een noodzakelijk kwaad dan wel een bewuste keuze van de komponisten.

Ook de hedendaagse mogelijkheid om kompositieprogrammas rechtstreeks muzieknotatie te laten produceren, heeft de komponist van het elektronische en syntetische geluid verlost. Op dit gebied, dat overigens een der belangrijkste inhoudten vormt van het door ons op het Konservatorium gedoceerde kompositievak, hebben we zelf ook veel onderzoek verricht dat onder meer uitmondde in een aantal grotere komputerprogrammas voor automatische muzieknotatie (DARMS.EXE), voor experimentele fuga-kompositie (FUGA1-4), voor experimentele koralen, voor polyritmische muziek (Shifts, Primes...) enz...

Deze stelling wordt ten andere eveneens gekonfirmeerd door het grote sukses waarvan de minder 'elektronisch' klinkende samplers thans genieten.

Kortom, wanneer we de stand van zaken inzake mechanizatie van bestaande akoestische instrumenten kritisch bekijken, kunnen we er niet onderuit te erkennen dat deze slechts voor de piano een zeer grote graad van perfectie heeft bereikt.

Cfr. TEITELBAUM, Richard "Das Digitale Piano" , in: HATTINGER, Gottfried (Red.), "Ars Electronica", Linz, 1986 p.164-166 & 203-205.

De firma Bösendorfer heeft vorig jaar (1992) een volledig nieuwe automatische piano op de markt gebracht, waarbij de dynamische resolutie werd opgedreven tot 10 bits. Het betreft hier echter geen 'Vorsetzer'-mechanisme maar wel een vleugelpiano waarbij het hele mechanisme vast werd ingebouwd. De nuanceringsmogelijkheden zijn verbluffend en overtreffen in ruime mate wat menselijk mogelijk is. Ook Yamaha heeft een dergelijk instrument (met geringere specificaties) in zijn assortiment.

Volledigheidshalve willen we de lezer echter toch niet de mededeling van de stand van zaken voor andere instrumenten onthouden.

4.3.1.2.5.: Slaginstrumenten

Deze instrumenten zijn over het algemeen bijzonder eenvoudig en met grote perfectie te mechaniseren. Dat het niettemin niet zo erg vaak voorkomt, heeft o.i. alles te maken met de erg sekundaire rol die deze instrumentengroep binnen onze muziekkultuur te spelen krijgt. Er bestaat omzeggens geen solistische literatuur, tenzij dan, voor telkens wisselende instrumentale combinaties, waardoor een mechanisering uiteraard niet erg lonend is. Komponisten zitten dan ook niet echt verlangend uit te kijken naar een ultiem elektronisch stuurbaar akoestisch slagwerk-arsenaal. Een tweede reden daarvoor is natuurlijk dat het ook erg onpraktisch zou zijn, gezien de ruimtelijke eisen die het zou stellen.

Het enige slagwerkinstrument waarvoor er zelfs van een traditie in elektrifikatie kan gesproken worden, is de beiaard! De vroegere mechanische programma-trommels werden en worden in de meeste beiaarden vervangen door elektrische systemen die veel eenvoudiger -uiteraard via een klein computersysteempje- te programmeren zijn.

In de vroegere orchestrions is geautomatiseerd slagwerk een essentieel deel van het automatisch orkest. Orchestrions werken evenwel bijna allemaal uitsluitend pneumatisch.

Enkele verdienstelijke komponisten en instrumentenbouwers die zich met geautomatiseerde akoestische slaginstrumenten hebben beziggehouden zijn:

Stephan Von Huene die verschillende volautomatische trommels, washboards, cimbalen, een tapdanser e.d.m. bouwde.

Martin Riches, die een stel elektronisch gestuurde woodblocks bouwde, inclusief een komputerinterface.

Alec Bernstein, die naast automatische trommels, ook automatische vibrafoons, xylofoons en marimbas bouwde met een zeer hoge graad van universele stuurbaarheid. In tegenstelling tot de meeste hedendaagse bouwers, maakt hij echter geen gebruik van MIDI-sturing, maar integendeel van een op technische gronden absoluut superieur parallel 8-bit digitaal kommunikatie-formaat.

4.3.1.2.6.: Blaasinstrumenten

Deze instrumentenkategorie valt heel wat moeilijker te automatiseren dan de hoger vermeldde groepen. Niettemin werd er toch door verschillende bouwers aan gewerkt.

De eerder vermelde Martin Riches bouwde voor Eberhard Blum een automatisch spelende dwarsfluit. Dit instrument werd en wordt o.a. gebruikt voor de vertolking van minimalistisch werk van de Amerikaanse maar in Parijs levende komponist Tom Johnson.

Van alle blaasinstrumenten is het klankopwekkingsmechanisme van de dwarsfluit wellicht het eenvoudigst mechanisch na te maken. Het volstaat een stuurbare luchtstroom onder een stuurbare hoek op het klankgat te richten. Voor de rietblaasinstrumenten en zeker voor de koperblaasinstrumenten is een en ander heel wat moeilijker te realizeren. Hier zou immers zoiets als syntetische lippen gebouwd dienen te worden. Voor zover wij konden nagaan zijn van iets dergelijks nog geen mechanische modellen gefabriceerd.

Zelf hebben we ons nog niet lang geleden gewaagd aan een experimenteel projekt voor de bouw van een elektronisch stuurbare en volautomatische akoestische saxofoon. Dit projekt, onder de naam 'Autosax', hebben we wat uitvoeriger toegelicht in de appendix bij deze studie. In de toekomst willen we wel deze 'Autosax' gaan gebruiken als klankopwekker in het kader van ons onzichtbaar instrument. 'Autosax' kan immers makkelijk MIDI-informatie verwerken.

4.3.1.2.7.: Strijkinstrumenten

De mechanika nodig voor de bespeling van strijkinstrumenten is bijzonder complex en eist een uiterste precisie. Van alle instrumenten is het dan ook zowat het laatste type waarvoor ooit mechanizaties werden ontworpen. In de pre-elektronische tijd, werden er wel reeds enkele pogingen gedaan, maar meer dan rariteiten -ondanks de mechanische complexiteit van de tuigen- waren het toch niet. Ook vandaag zijn er slechts aanzetten tot elektrische sturing van strijkinstrumenten bekend, meer bedoeld als demonstraties voor de mogelijkheden van stappenmotoren en komputersturing dan voor werkelijk muzikale toepassingen.

4.3.1.2.8.: Tokkelinstrumenten

Deze instrumenten zijn weliswaar aanzienlijk eenvoudiger mechaniseerbaar, maar -en dit heeft wellicht veel te maken met enerzijds hun geringe geluidsproduktie, anderzijds het feit dat zij, met uitzondering van de harp, niet in 'het symfonisch orkest' werden opgenomen-, ernstig bedoelde en elektronisch stuurbare exemplaren zijn ons ook in deze groep niet bekend. Aanzetten tot elektrische mechanizatie zijn wel regelmatig terug te vinden als elementen van klankskulpturen. (o.a. bij Frédérique LE JUNTER).

4.3.1.2.9.: Nieuwe akoestische instrumenten

Uit dit beknopt overzicht blijkt dat wanneer we voor de verklanking traditionele instrumenten willen gebruiken, eigenlijk alleen de klavierinstrumenten, in het bijzonder de piano, in aanmerking genomen kunnen worden. Alleen voor deze groep bestaat er nu reeds een elektrisch mechanisme dat een controle over het instrument mogelijk maakt met dezelfde, zoniet betere, specificaties als die van een menselijke bespeler.

Logischer ware het uiteraard, het probleem van de onbevredigende elektronische klanken en het probleem van de mechaniseerbaarheid van de bestaande akoestische instrumenten te omzeilen door geheel nieuwe a priori elektrisch stuurbare akoestische klankbronnen en mechanismen te gebruiken. Het is de weg die we o.m. voor het projekt 'Hex' zelf een heel eind hebben bewandeld. Een diepgaande bespreking van de problematiek verbonden met het opnemen van 3-dimensionele voorwerpen in elektronische trillingskringen en de methodes voor de sturing ervan valt echter buiten het opzet van deze studie. Een kommentaar bij ons eigen 'Hex' projekt, voegden we niettemin -ter informatie- toe in de appendix.

Anderen die op dit gebied van de automatische instrumentenbouw belangrijk experimenteel werk leveren zijn o.a. Jacques REMUS, die allerlei snaarinstrumenten, slaginstrumenten en blaasinstrumenten voor automatische bespeling bouwt, zonder dat ze echter aanspraak kunnen maken op enige meer algemene muzikale aanwendingsmogelijkheid.

Het klinkend resultaat van zijn nieuwsoortige akoestische machines is gedokumenteerd op de CD 'Concertomatique/Mecamusique', IPOT001, ed. :L'Ecart, Paris, 1992.

Verder hebben we nog de reeds vermelde Stephan Von Huene, die echter vooral bedrijvig is op het gebied wat meestal als 'sound sculpture' wordt omschreven. Daardoor hebben ook zijn instrumenten geen bredere toepassingsmogelijkheden dan wat hij er zelf in voorzag. Ook moet opgemerkt, dat het grootste deel van Von Huene's werk gebruik maakt van pneumatische sturing.

Een komponist die bij zijn instrumentenbouwkundig werk vertrok van hetgeen reeds bestond op het gebied van de elektrifikatie van de player-piano, is de Australiër Alistair RIDDELL. Net zoals de Amerikaan Alec BERSTEIN, ging ook hij oorspronkelijk uit van de Maranz-Superscope player-piano. In plaats van het instrument in traditionele zin te verbeteren, breidde hij het echter uit met allerlei volstrekt non-konventionele speeltechnieken. Zijn gemodificeerde player-piano's omvatten dan ook mechanismen om de snaren in de klankkast rechtstreeks te bespelen en zelfs, om lange aangehouden tonen voort te brengen.

Cfr. BURT, Warren, "Experimental Music in Australia using live electronics" , London 1991, p.164.

JENKINS, John "22 Contemporary Australian Composers", Melbourne, 1988, p.149-156.

4.3.2.: De uitvoerder als loutere interpreet

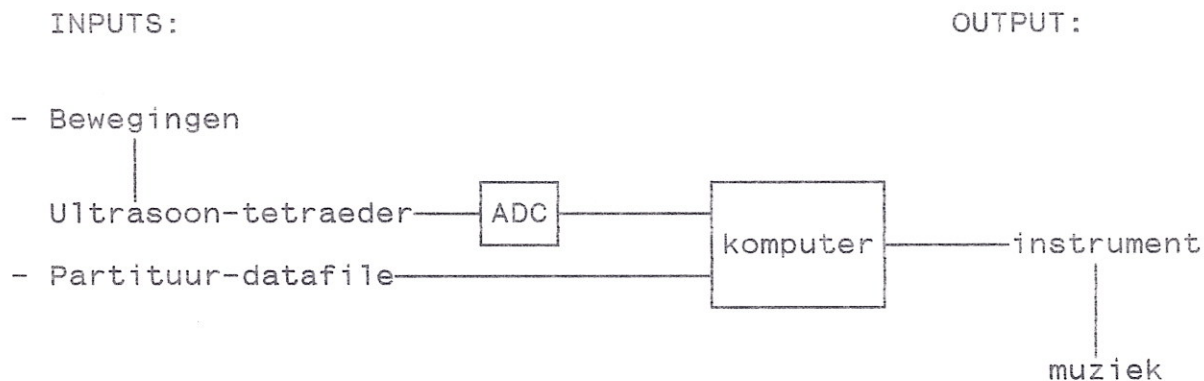
Om nu de uitvoerder van het ambachtelijk aspekt van het muziekmaken volledig te ontlasten is in de eerste plaats een motorische 'controller' nodig die interpretatief relevante bewegingen van de uitvoerder, 'mapt' op uitvoeringstechnische aspekten van de partituurvertolking.

Hier valt nog behoorlijk wat research en experimenteel werk te verrichten inzake precies relevante onderzoek van uit motoriek afleidbare elektrische signalen voor het klankresultaat. Het technisch materiaal nodig om dergelijk onderzoek te verrichten is o.i. voorhanden: druksensoren, kameras, ultrasoondetektoren, spierspanningsopnemers e.d.m. Idealiter zou dergelijk onderzoek interdisciplinair moeten worden gevoerd met medewerking van gespecialiseerde medici, electrotechnische ingenieurs en een terzake competent muzikoloog.

Afgezien van het nog te verrichten onderzoekswerk op dit vlak, kan als 'controller' ook het hier beschreven 'Holosound' principe worden ingezet. Uiteraard is daarvoor dan een brok voor deze toepassing specifieke software nodig waarmee de noodzakelijke 'mapping' van beweging naar muzikale interpretatie kan geschieden.

Een eerste aanzet daartoe, hebben we zelf reeds gezet in de vorm van het komputerprogramma 'Player', dat evenwel specifiek werd geschreven voor de vertolking van mijn eigen eveneens via een komputer-expert-systeem gekomponeerde koraal- en fuga-boeken. Omdat deze orientatie niet op algemene toepasbaarheid kan bogen ga ik er hier niet verder op in. De resultaten van dit werk zijn overigens wel publiek beschikbaar: vele fugas werden door diverse radio-instituten opgenomen en uitgezonden, terwijl ikzelf regelmatig deze experimentele komposities via genoemd player-programma op concerten naar voor breng.

Een mogelijk gebruik van het Holosound systeem als controller bij het interpreteren van bestaande muziekstukken gaat uit van volgende blokschematische opbouw:



In de software moeten elementen van patroonherkenning ('pattern recognition') ingebouwd worden zodat de muzikale frazering als een lichaamsbeweging kan worden gerealiseerd: bvb. het maken van boog- en golfbewegingen met de armen. De metriek kan worden afgeleid uit de meer diskontinue elementen van de lichaamsbeweging. Dit sluit nauw aan bij zowat door alle musici en amateurs gebruikelijke bewegingen in verband met muziek: met de voet wordt de maat getikt, terwijl de armen het melodisch verloop lijken te 'volgen'. Verder kan de bewegingshoeveelheid een maat zijn voor de dynamiek van het te vertolken muziekstuk.

Ook de hantering van een dergelijk 'muziek-interpretatie-instrument' zal natuurlijk tot op zekere hoogte geleerd dienen te worden, ook al zouden we het in de implementatie nauw laten aansluiten bij 'normaal' of 'intuitief' muzikaal-motorisch gedrag. Alleen van de nauwkeurige tekst-lezing en de technische reproductie kan het ons wel degelijk verlossen.

4.3.3.: Improvisatie versus kompositie

In deze studie hebben we het 'Holosound' systeem in de eerste plaats beschreven als een onzichtbaar of non-impact instrument. Een werktuig dus, voor gebruik in publieke muziekuitvoeringen.

Tijdens de vele experimenten met het tuig, bleek het ook erg geschikt als kompositorisch werktuig. Niet in het minst omwille van de grote onmiddellijkheid en de haalbare complexiteit van het klankresultaat.

Het is uiteraard geen nieuwtje wanneer we hier zouden verdedigen dat het instrument toelaat op grond van 'improvisaties' te komen tot muzikale composities, als betrof het een bijzondere verdienste. Immers, elk beheersbaar instrument laat in principe toe het resultaat vast te leggen en als 'kompositie' te presenteren: hetzij via een muzieknotatie waarbij de tekens verwijzen naar klanken, hetzij via een notatiesysteem dat de te stellen handelingen weergeeft die tot een bepaald akoestisch resultaat aanleiding geven, hetzij in ultimo, via een gewone geluidsopname.

Ook het maken van composities in klassiek notenschrift via het inspelen van de muziek op elektronische instrumenten (via MIDI) behoort reeds lang tot de verworvenheden van de kompositorische techniek. Dat ook het hier voorgestelde instrument dit toelaat ligt voor de hand. Het volstaat daartoe de midi-uitgang(en) aan te sluiten op een komputer met MIDI-interface waarop een meerkanaals sequencer programma loopt ('CakePro' doet het prima). De in real time ingelezen sporen (tracks) kunnen dan als MIDI-file worden opgeslagen en achteraf -al dan niet na invoering van gebeurlijke korrekties- opnieuw ingelezen in een programma voor notenschrift. Zelf gebruik ik voor dit doel bijna uitsluitend 'The Noteprocessor', een programma gepubliceerd door Thoughtprocessors lopend op computers van de IBM-AT familie.

Het bijzondere in dit geval is echter, dat er steeds een deel a priori-kompositie -en nog wel op algoritmische basis- dient te gebeuren op voorhand, waarmee de kompositorische syntax grotendeels wordt vastgelegd. Immers deze muzikaal syntaktische algoritmes zijn een werkingvoorwaarde voor dit instrument. Hierdoor vervaagt de grens tussen 'improvisatie' en 'kompositie'. Het kompositorische gaat namelijk gesubstantieerd deel uitmaken van de instrumentale definitie. Het instrument wordt voor een deel de kompositie die erop of ermee wordt vertolkt.

Ook dit is op zich genomen nu ook weer niet zo nieuw als het moge klinken: de orchestriërs die het leven opvrolijkten in vroegere tijden deden dat eigenlijk ook reeds in zekere mate. Maar, bij de vroegere mechanische, pneumatische, elektrische en elektronische speelautomaten kon eens het programma (de rol) in het instrument gebracht was, hooguit het tempo en de geluidsterkte ingesteld worden. Bovendien, en dit is het meest wezenlijke onderscheid, konden de composities uitsluitend als

lineaire vormen in de tijd worden opgevat en niet als een verzameling syntactisch samenhangende structuren die kunnen worden toegepast op in principe willekeurige muzikale parameters. Anders dan bij speelautomaten het geval is, kunnen implementaties van het holosound systeem gezien worden als in real-time werkende expertsystemen die een werkelijke interpretatieve zowel als kompositorische muziekschepping mogelijk maken. De balans tussen interpretatie van het in het instrument vastgelegde enerzijds en compositie van nieuwe klankstructuren binnen de algoritmisch vastgelegde syntax anderzijds hangt volledig af van het programma waarmee het instrument wordt uitgerust. Een zo grote vrijheid qua keuzemogelijkheden werd o.i. nooit eerder door speelautomaten geboden.

Thans werk ik aan een orkestpartituur die berust op een verdere uitwerking van het algoritme uit '*A Book of Moves*' waarin een dirigent wordt geïmplementeerd ('*Lead*'). Het surrealistische ervan is, dat de partituur dus gedirigeerd wordt nog voor ze zelfs maar bestaat...

Technische problemen zijn hieraan echter wel verbonden. Deze hebben in eerste plaats te maken met de intrinsieke traagheid van het MIDI-systeem zelf, en in tweede plaats met de beperkingen qua snelheid van de gebruikte personal computers. Het is weliswaar eenvoudig een brokje software te schrijven dat alle ontvangen data kanaal per kanaal naar disk wegschrijft, maar zulks leidt -behalve in de meest banale en triviale gevallen- niet tot een bruikbare muzieknotatie. Immers, muzikaal dikte veronderstelt een behoorlijk groot deel informatieverwerking en structurering, iets waarvoor behoorlijk complexe software nodig is die tot op de dag van vandaag nog niet is geschreven. De software waaraan op dit ogenblik gewerkt wordt, loopt niet in real-time. De partituur 'rolt' dus niet zomaar uit de printer naarmate het stuk wordt gespeeld...

4.4.- Een toekomst voor de organologie?

Wij bevinden ons thans welhaast op de grens van de eenentwintigste eeuw en kijken nu reeds terug op meer dan een eeuw instrumentenbouw waarin aanvankelijk elektrische (Cahill), dan elektronische instrumenten en in het laatste kwart van de eeuw software-matig gebouwde instrumenten de belangrijkste bijdragen vormden aan de verzameling muzikale werktuigen die de mens heeft voortgebracht.

Deze simpele en onmiskenbare vaststelling heeft voor de organologie, wanneer zij als wetenschap ernstig wil worden genomen, verstrekkende gevolgen. De organoloog die zich ook om het twintigste eeuwse arsenaal aan muzikale werktuigen wil -en hij is daar eigenlijk toe verplicht- bekommeren, zal zich een kennis van die 'nieuwe' technieken eigen dienen te maken. Daar waar vroeger een elementaire kennis van akoestiek, mechanika en warenkunde, ingebed in een goed inzicht in muziekgeschiedenis en/of etnomuzikologie volstonden, is thans ook een kennis van elektromechanika, analoge en digitale elektronika en... software essentieel geworden voor de studie van de twintigste eeuwse instrumenten.

Wanneer we vooruit kijken op wat komen kan, hebben we alle redenen om aan te nemen dat de instrumentenbouw in de volgende eeuw wellicht kwasi uitsluitend gebruik zal gaan maken van informatika. Het instrument van de toekomst zal gebouwd worden, geheel en al in software. De hardware zal tegen die tijd immers zo krachtig zijn, dat onafhankelijk van haar architectuur en bouw, de software de muzikale resultaten en de manier waarop die worden bereikt (de instructie) zal bepalen.

Tot voor de intrede van de informatika werd de fysische verschijningsvorm evenals de speelwijze van een muziekinstrument integraal gezien in functie van het mechanisme dat voor de klankopwekking zorg draagt. De ergonomie van de vroegere ontwerpen van muziekinstrumenten was zwaar gekompromiteerd door de kondities van de fysische klankopwekking. Een beiaard met 'gewone' piano-toetsen zou puur mechanisch gezien volstrekt onrealiseerbaar zijn. Laat staan, een viool die via piano-toetsen zou worden bespeeld. We weten wel dat dergelijke tuigen ooit zijn gebouwd (hierbij denken we dan bijvoorbeeld aan het Truchado instrument in de kollektie van het Brusselse Instrumentenmuseum), maar veel meer dan historische rariteiten en probeersels zijn het toch nooit geweest.

Cfr.: DE HEN, F.J., "The Truchado Instrument: a Geigenwerk?" in: RIPIN, E.M. (Red.) "Keyboard Instruments", 1977 (1971).

Zelfs voor alle vroege elektrische en elektronische instrumenten kan men vaststellen dat de werktuiglijke vormgeving een functie bleef van de methode van klankopwekking. Zo zijn de speelwijzen van instrumenten zoals de Theremin en de Ondes Martenot in eerste plaats een louter gevolg van de technologie waarop zij berusten en beslist niet van een vizie op een alternatieve ergonomische aanpak. De gevolgen van de gevolgde technologische weg bij de bouw op de instructie, op de speeltechniek, werden als gevolgen aanvaard en ingerekend. Zij vormden geen uitgangspunt bij het ontwerp.

Pas sedert de laatste jaren (hooguit sedert iets meer dan een decennium) biedt de elektronische en elektromechanische technologie ons de definitieve mogelijkheid om de menselijke omgang met het muzikale werktuig onafhankelijk te gaan ontwikkelen en ontwerpen van het mechanisme dat voor de uiteindelijke klankproduktie instaat. Thans is het perfect mogelijk een 'user-interface' te ontwikkelen onafhankelijk van, en zelf volstrekt zonder rekening te houden met, de precieze aard van de muzikale output.

De eerste stappen in deze richting waren de diverse pogingen van ontwerpers om interfaces te bouwen die zo nauw mogelijk aansloten bij de instructie die de musici reeds vertrouwd was: de Lyricon (een ge-elektronificeerde sopraansaxofoon), het midi-klavier, de Max Matthews viool, de MIDI-gitaar, de drumpads en de Air-drums... de lijst groeit nog elke dag.

De volgende stap, één waarvoor we in deze studie een eerste aanzet uitvoerig wilden demonstreren, gaat een heel stuk verder en behoorde nooit eerder tot de mogelijkheden van de mens: uitgaan van de menselijke motoriek zelf, in eerste instantie los van enige instructie, als uitgangspunt voor de bouw van een user-interface. Verder gaan in ergonomische perfectie, is nauwelijks voorstelbaar. Het werktuig wordt hier in zijn hantering bevrijd van zijn mechanische implicaties. Een uniek moment in de ontwikkeling van de technologie en de ergonomie.

Dat reeds op dit ogenblik het user-interface volledig los kan staan van het tuig dat voor de uiteindelijke verklanking instaat, wordt wellicht het duidelijkst wanneer we de mogelijkheid beschouwen om bvb. het in 'A Book of Moves' voorgestelde interface te gebruiken voor de aansturing van een moderne digitaal stuurbare akoestische player-piano. Dergelijke toestellen worden met een zeer hoge graad van perfectie -hun spel is thans in niets meer van dat van een echte pianist van vlees en bloed te onderscheiden- gefabriceerd door bedrijven zoals Yamaha en Marantz, evenals door individuele musici zoals Alec Bernstein en Trimpin (zie hoger). Gekoppeld aan bijvoorbeeld het door ons ontworpen interface, is het mogelijk bvb. écht piano te spelen, zonder ook maar ergens rekening te moeten houden met de instructie die normaal gezien en historisch aan de piano als muziekinstrument is verbonden. Het is weliswaar zo, dat dergelijke mechanisering niet met eenzelfde graad van perfectie voorhanden zijn voor andere traditionele instrumenten, maar dat is hier principieel eigenlijk niet van belang.

Overigens, maar daar zijn we in de huidige stand van de techniek en meer bepaald van de informatika, nog niet echt aan toe, hoeven we in de toekomst ook niet de weg te bewandelen van de mechanisering van traditionele instrumenten, maar kunnen we ook gebruik maken van de klank die zij voortbrengen.

De thans reeds bestaande en bijzonder populaire samplers zijn hiervan de voorboden. Door velen worden ze nu reeds beschouwd als finale realisaties, maar dat lijkt ons veel te ver te gaan. Immers alle bekende samplers maken gebruik van periodiciteit in de klankgeneratie: zij spelen een beperkt aantal digitale samples af onder parametrische beïnvloeding van

enkele soms programmeerbare muzikale aspecten (frekwentiemodulatie -vibrato-, amplitudemodulatie -tremolo-, pitch-bend -portamento en legato, ...). Een muziekinstrument dat voor zijn muzikale verklanking gebruik maakt van periodiek herhaalde samples, zal nooit eenzelfde klankrijkdom kunnen laten horen dan een werkelijk mechanisch-akoestische klankbron. Om dit te kunnen bereiken is nog heel wat werk aan de winkel inzake algoritmische implementaties van complexe trillende systemen met meerdere vrijheidsgraden. Hiervoor is nog menig probleem inzake de onderliggende kennis van complexe trillingssystemen onopgelost. De modellen waarover we beschikken zijn nog tezeer simplificaties, en dat wordt vooralsnog gehoord door de kritische luisteraar. Pas wanneer onze modellen een evenwaardige complexiteit -musici zouden dit het interne leven van een klank noemen- kunnen tot klinken brengen als een akoestische geluidsbron dat kan, zal volkomen syntetische klankopwekking een muzikaal volwaardig alternatief kunnen bieden.

Cfr. wat we terzake schreven in de parenteze 1.4.

De konsekwenties van deze stelling zijn ook voor de toekomst van de muziekkultuur van niet te onderschatten belang.

Enerzijds zullen instrumenten meer dan dat ooit in enig verleden het geval is geweest, een functie kunnen zijn van de voorkeuren op speeltechnisch zowel als muzikaal esthetisch en syntaktisch vlak, van de muzikmakers in de ruimste zin. De instrumentale verscheidenheid kan dus uitermate grote proporties aannemen, inzover zelfs, dan ieder muzikmaker uiteindelijk over een eigen van dat van anderen verschillend instrument zou kunnen beschikken. Bovendien houdt dit de mogelijkheid in, van een nog grotere en intrinsieke verwevenheid tussen kompositie enerzijds en instrument, waarbij de definitie van het instrument en de kompositie in laatste instantie zelfs kunnen gaan samenvallen.

Anderzijds kan diezelfde potentiële flexibiliteit van het software-instrument ook het gevaar van een extreme verarming van het instrumentarium in zich sluiten. Dit laatste geval zal zich voordoen waar er onvoldoende kennis en macht over de middelen bestaat bij diegenen die het instrumentarium zouden willen gebruiken om dit naar hun hand te zetten. In dit laatste en pessimistische geval, worden zij slachtoffers van de technologie en van de marktmechanismen die de modes noodzakelijk maken.

Precies omdat wij ons ervoor willen inzetten dit laatste scenario minstens ten dele te verijdelen, is het zo belangrijk en dringend deze kennis voor musici, muzikologen en organologen beschikbaar te stellen.

Wij hopen hiertoe een steentje te hebben bijgedragen.