

‘Revolution – J. Shaw and others’

Emulation proposal by Paul Klomp – May 31 2007

English summary – Dutch version below

From June 2006 through May 2007 klomp kunst & electro (aka. Paul Klomp) has done an in depth research of the technology that forms the functional basis of ‘Revolution’, an interactive installation by J. Shaw and others.

This research was to result in a precise description of hard- and software, function and user interaction, a description that can later be used as a basis for an emulation.

Why emulate?

Within the next 10 years one of the components of ‘Revolution’ (an analog rotation sensor, an 8086 XT PC, a Sony DLP1500 Laserdisc player and a handmade sample player/interface box) will probably break down beyond repair. PC or laserdisc player may be replaced, but the handmade hardware that interfaces audio- and sensor data can not be rebuilt. There is no schematic diagram, no documentation whatsoever. If this installation is to be shown to the public later, the current hardware must be replaced.

Description process:

Before setting up the installation for measurements and observation the contents of the laserdisc were digitized. Also the audio EPROM’s were read out and their data secured digitally.

During the research the hardware of ‘Revolution’ broke down. The motherboard of the PC appeared to be damaged by acid leakage from the backup battery, resulting in malfunction of the communication ports.

Running the software – that is also completely undocumented and only available in binary executable form – on any other PC would not work. After a long search it was found that the original motherboard had been manually patched, at the bottom side. Connecting these patched signals in another computer made it possible to run the software again, enabling further measurements and repair of the original PC.

Now many aspects of the behavior of the installation were measured and described: the relation between movement and image, the timing of image sequences, the playback frequencies of the audio channel related to the speed of movement, etc. Also output from the installation, both video and audio, was captured and stored. Combined, these media, tables and dependencies form a description of the behavior of the installation.

Functional emulation and test:

To test the validity of this description an emulation was made in ‘Pure Data’. The emulating (modern) PC gets its input form a new rotation sensor through a well documented interface board (Arduino, using RS232 or MIDI).

In the programming process it appeared that the installation showed images in a different order then stored on the laserdisc. Furthermore, the sound from the EPROM’s was barely useable because the player uses a non standard (not documented) hardware expansion algorithm.

concept emulatievoorstel 'Revolution'

In a test setup at Montevideo image and sound from the emulation were compared to the output of the original installation, showing minor timing discrepancies.

The emulating program was adapted and once more functionally compared to the original set up, this time using the data that was captured from the in tact installation.

The 'copy' appeared to be reasonably precise.

The emulation is not a full replacement of the original hardware: e.g. the images are not full screen, nor in the right video resolution. But it proves that there is an adequate description of the functionality of the installation and that image and sound data was stored correctly.

Future:

Computers develop rapidly. This implies that the emulation as it is can only be used a few years, since then the operating system on which it is based (winXP) will be no longer supported.

Next step is to investigate the possibilities to ensure operation of the software say 100 years from now. A solution could be building the whole project from source, using Linux, principally enabling a rebuild for any future hardware platform.

J Shaw e.a

Revolution

Emulatievoorstel

mei 2007

Inhoud

Inleiding	3
Concept Emulatievoorstel	4
Logboek	14
Emulatie – eerste opzet	31

Inleiding

Van juni 2006 tot maart 2007 is door NIM i.s.m. klomp kunst & electro onderzoek gedaan naar de installatie 'Revolution' van J.Shaw e.a..

'Revolution' is een interactieve video-installatie van Jeffrey Shaw, met bijdragen van Tjebbe van Tijen (co-author), Huib Nelissen (constructor), Gideon May (software developer), Charly Jungbauer (hardware developer) en Bas Bossinade (hardware developer).

De installatie maakt deel uit van de reizende tentoonstelling "Imago, fin de siècle in Dutch contemporary art" (1990).

Doel van het onderzoek:

Het zodanig omschrijven van de in de instalatie gebruikte hard- en software, opbouw, werking en interactie dat op basis van deze omschrijving een haalbaar voorstel voor emulatie van de gebruikte hardware kan worden geformuleerd.

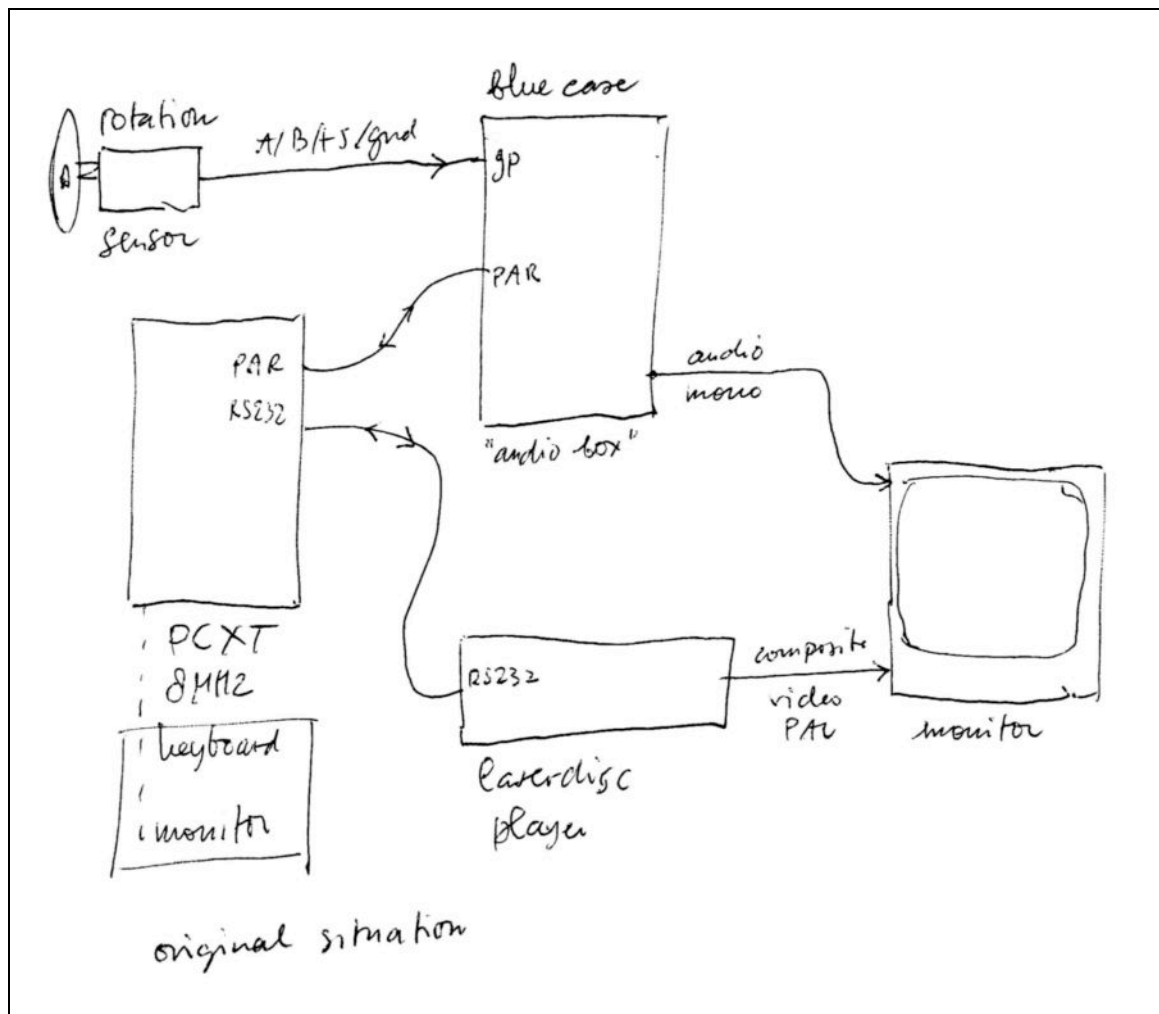
Omschrijving van de gebruikte techniek

De installatie bestaat uit een centraal gelagerde kolom die door de bezoeker kan worden rondgedraaid d.m.v. een hefboom.

Op deze kolom staat een monitor die videobeelden toont.

De draaiing van deze kolom wordt door een sleepwiel dat is verbonden aan een rotatiesensor vertaald in elektronische pulsen. Deze pulsen vormen het ingangssignaal van een computerprogramma dat

- via RS232 een laserdiskspeler bestuurt die de beelden genereert
- via de LPT poort een kastje controleert dat het geluid op variabele snelheid kan weergeven.



figuur 1, oorspronkelijke situatie

Probleemstelling

Tijdens een eerste bespreking van de problematiek kwamen aan de hand van de vraag "wat moet er gebeuren om 'Revolution' in de toekomst ten toon te stellen" de volgende punten aan de orde:

te verwachten problemen:

- levensduur monitor
- leverbaarheid CRT monitor
- vervagen inhoud eprom's
- slijtage elektronica, m.n. condensatoren
- slijtage beeldplaatspeler, elektronisch/mechanisch
- gevoeligheid beeldplaat

strategieën:

- a. niets doen, kan nog misschien 10 jaar tentoongesteld worden
- b. nauwkeurige inventarisatie van de werking van de installatie, inclusief dynamisch gedrag
 - vervolgens
 - geen verder onderhoud, de installatie kan zonodig gereconstrueerd worden
 - noodzakelijk onderhoud, ad hoc, beeldplaatspelers op voorraad houden
 - vervanging van slijtagegevoelige componenten: laserdiskspeler, eproms
 - vervanging van de functionele techniek door een simulatie

ethische/esthetische vragen: welke parameters zorgen voor 'look and feel'

- het mechaniek van de installatie (blijft onveranderd)
- vervanging van PC, kastje etc (zichtbaar in de sokkel) door nieuwe apparatuur, lichter bovendien
- monitor TFT/LCD of CRT
- beeldkwaliteit: teruggrijpen op origineel digitaal materiaal of op maximale kwaliteit captures van de bestaande beeldplaat
- geluidskwaliteit: zoeken naar origineel geluid of uitlezen uit de e(e)proms
- timing: seamless loop is essentieel, ook kwaliteit stills is met mpeg2 niet voldoende, dus: afspelen vanaf harddisk

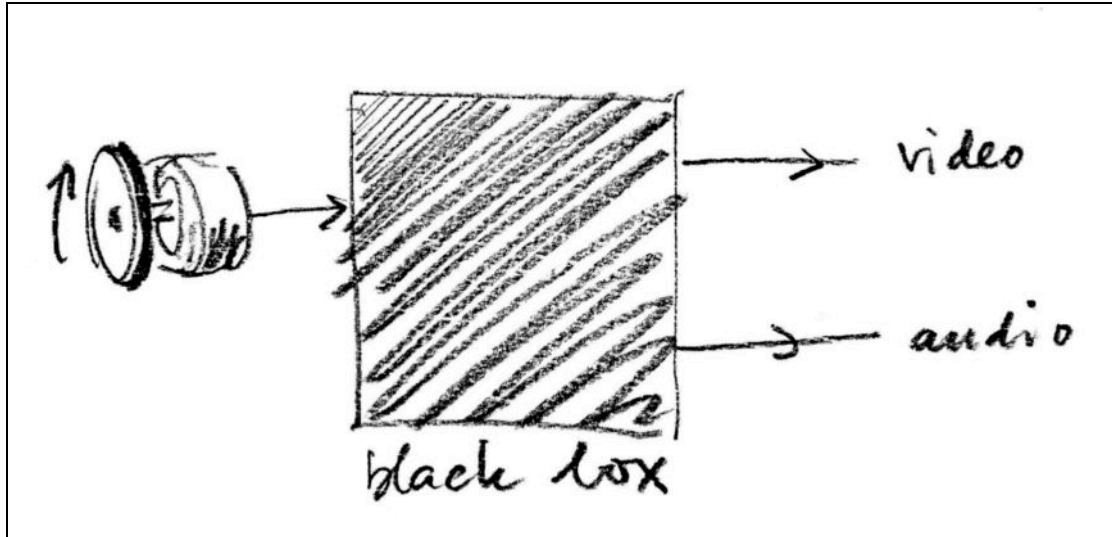
Op deze esthetische vragen van wordt in dit voorstel nauwelijks ingegaan, het onderzoek heeft zich toegespitst op het vinden van vervanging van de technische functionaliteit.

Vervanging van de monitor is daarbij buiten beschouwing gebleven.

Wellicht zal het door de voortgaande verbetering van LCD/TFT schermen mogelijk zijn contrastomvang en grijsweergave van een Sony Trinitron monitor te benaderen. Maar de cilindervorm van de Trinitron buis die het beeldmateriaal over een denkbeeldige cirkel laat bewegen zal moeilijk met een TFT scherm gerealiseerd kunnen worden. Ook is de kijkhoek van TFT/LCD nog steeds kleiner dan die van een CRT monitor waardoor voor kijkers op afstand de installatie een andere werking zal hebben.

Opzet van het onderzoek

Uitgaande van het 'black box'-principe (alleen de in- en uitgangssignalen worden bekeken) is het mogelijk een sluitende analyse van het gedrag van de toegepaste techniek te maken.



figuur 2, de "black box"

Het ingangssignaal van de elektronica wordt gevormd door verdraaiing van een sleepwielletje.

Uitgangssignalen zijn video en audio.

De analyse kan vergemakkelijkt worden door stiekem in de black box te kijken, dat wil zeggen gebruik te maken van meetbare toestanden binnen deze black box (zie ook figuur 1):

- de functionaliteit van de audio-box kan aan de hand van de interne opbouw geanalyseerd worden
- met een pulsgenerator in plaats van het sleepwielletje kunnen exact getimed bewegingen worden nagedaan
- met een LED display kunnen de signalen uit de PC ('PAR') waarmee de audio-afspeelfrequentie wordt bepaald discreet worden uitgelezen.
- met een oscilloscoop kan de samplefrequentie van de audio-player gemeten worden.

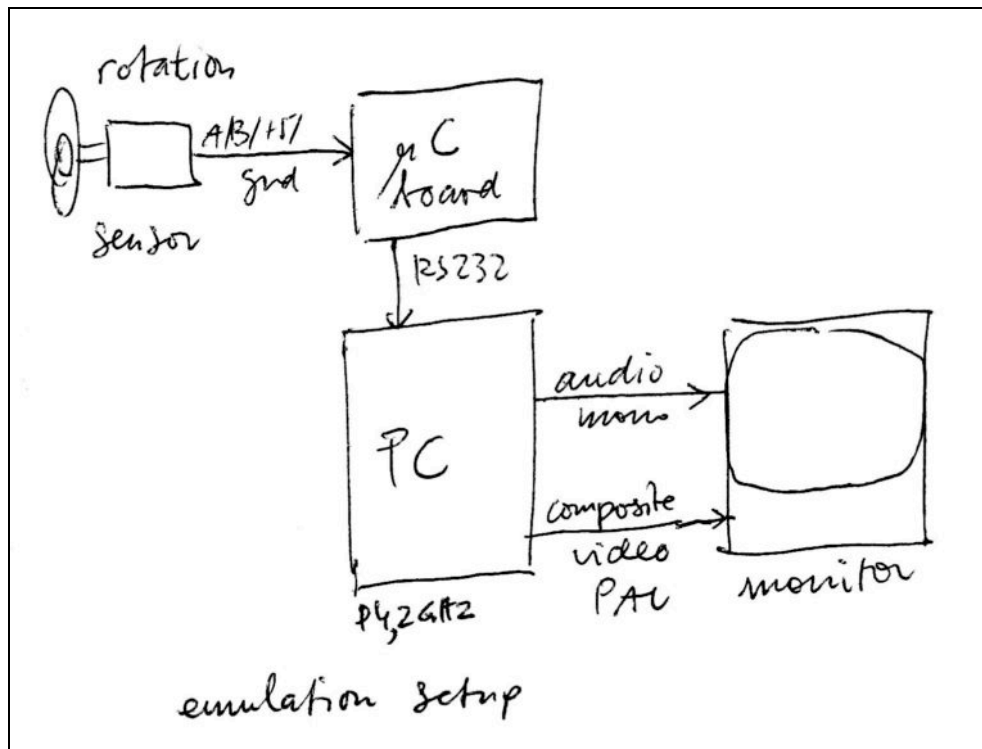
De precieze onderzoeksresultaten zijn in de 'logboek' sectie terug te vinden. Per sessie is aangegeven wat de onderzoeksdoelen waren, welke methoden gebruikt zijn, wat de resultaten zijn geweest en welk vervolgonderzoek nodig was.

Emulatievoorstel

Op basis van de in de 'logboek' sectie beschreven onderzoeksresultaten en de volgende uitgangspunten is een emulatievoorstel geformuleerd:

1. de installatie is volledig geanalyseerd en omschreven
 - a. video- en audiomateriaal zijn in digitale vorm voorhanden
 - b. de precieze reactie/interactie van de installatie is vastgelegd
 - c. de fysieke kenmerken van de installatie zijn vastgelegd
2. de hardware van de installatie zal geëmuleerd worden op basis van deze analyse; dit betreft rotatiesensor en aan het gedrag van de sensor gekoppelde beeld- en geluidsgeneratie.
3. de vervanging hoeft niet conform de huidige indeling naar functie, interfacing tussen de apparaten, protocollen, m.a.w. de huidige indeling in apparaten kan vervallen.

Of de hier verzamelde informatie voldoet om de installatie correct te emuleren kan worden onderzocht door een emulatie naast het oorspronkelijke werk op te stellen. Dan zal duidelijk worden of er essentiële informatie ontbreekt in de observaties, en of interactie, 'look & feel' werkelijk overeenkomen.



figuur 3, hardware voor emulatie

Voor een goede emulatie van de elektronica in 'Revolution' is een actuele PC (2006) met 2GByte RAM nodig. Op de harde schijf van deze PC zijn de beelden en de geluidsfragmenten opgeslagen. Het grote geheugen is nodig om zonder vertraging alle beelden te kunnen tonen.

De rotatiesensor moet op deze PC worden aangesloten, daarvoor is een I/O-interface nodig.

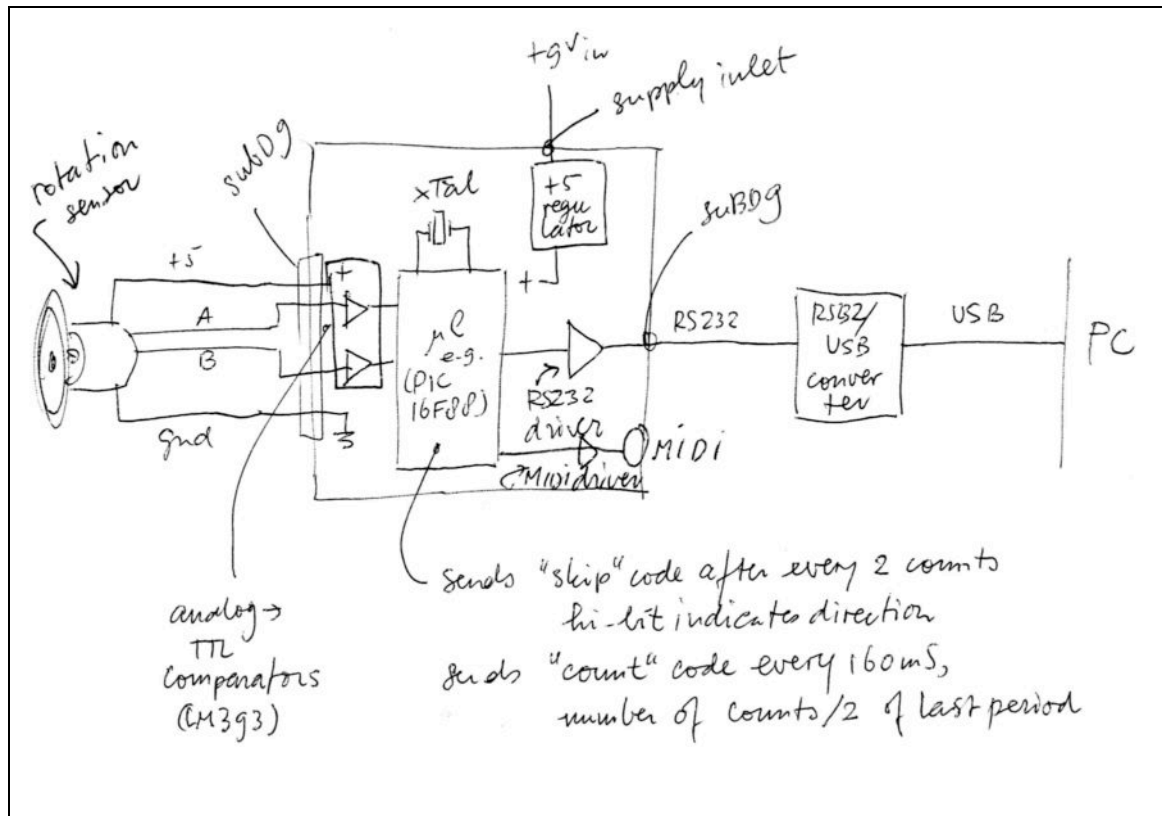
Om timingproblemen te voorkomen – er kan zeer snel achtereen een nieuw videoframe worden opgevraagd als de installatie snel wordt rondgedraaid - wordt een tijdkritische functie, namelijk het meten van het aantal rotatiepulsen tijdens iedere 160 mS, uitgevoerd door een microcontroller. Dit heeft als bijkomend voordeel dat de software voor het aansturen van beeld- en geluidsfragmenten via een standaard protocol (RS232, MIDI) met de sensor kan communiceren. Dit maakt het ontwerp in grote mate onafhankelijk van de in de PC gebruikte hardware en geeft veel vrijheid bij de ontwikkeling van de software die de geluids- en beeldbewerking real time moet uitvoeren.

Uitwerking emulatievoorstel

Hierna wordt in meer detail ingegaan op de in figuur 3 vermelde componenten:

- a. de I/O interface (microcontroller board)
- b. de PC, specificaties
- c. de PC, software:
 1. observaties gedrag
 2. voorstel voor testprogramma
- d. fasering verdere ontwikkeling
- e. begroting emulatie

a. de I/O Interface



figuur 4, voorstel I/O-interface

- rotatiesensor
 - o het vergt ingrijpende veranderingen aan de constructie van de installatie om een axiale rotatiesensor te gebruiken. Reden om de gebruikte 'sleewiel'-constructie te handhaven.
 - o rotatiesensors met 360 pulsen per omwenteling + richting indicatie zullen nog lang leverbaar zijn. Ze worden gebruikt in industriële productielijnen, robotica etc.. Omdat de pulsen door 2 gedeeld worden voordat ze geïnterpreteerd worden voldoet een 180 pulsen sensor ook.

concept emulatievoorstel 'Revolution'

- signaalniveaus: in de simulatie wordt uitgegaan van signalen op logisch niveau (0/5V).
- om TTL-niveau logische signalen te verkrijgen wordt de huidige sensor aangesloten op een level-converter of eventueel vervangen door aan TTL ge-interfaced model.
- voeding:
De voeding voor microcontroller en level-converters wordt in de interface gestabiliseerd om storing van buitenaf te voorkomen
- communicatie: RS232 of MIDI interface
 - RS232 is een standaard protocol dat tot op heden in veel industriële applicaties wordt gebruikt. Een datasnelheid van 19 KBaud is ruim voldoende om het maximale aantal rotatiepulsen (400/S) door te geven aan de hoofdcomputer.
 - MIDI is een nog steeds veel gebruikte standaard voor datacommunicatie. De snelheid voldoet voor het doorgeven van de pulsen.
- microcontroller: miniatuurcomputer die 2 taken uitvoert:
 - telt binnen een venster van 160 mS het aantal binnenkomende pulsen (en deelt dat door 2). Het resultaat van deze telling wordt elke 160mS verstuurd.
 - iedere tweede puls van de rotatiesensor genereert een uitgaande boodschap via RS232 die aangeeft
 1. dat er een puls is geweest
 2. wat de richting is van de laatste bewegingde programmering van deze bouwsteen kan in een hogere programmeertaal plaatsvinden waardoor de code ook later begrijpelijk is en aangepast kan worden aan een andere sensor, een andere microcontroller of een ander type interface.
- RS232/USB converter:
moderne Pc's hebben geen RS232 poort maar wel USB aansluitingen. Eventueel kan ook gebruik gemaakt worden van een USB microcontroller, de converter kan dan vervallen. Wel moet dan onderzocht worden of de driver voor zo'n interface in open source beschikbaar is

b. de PC, specificaties

- voldoende werkgeheugen (2 GByte)
- Microsoft Windows XP SP2 of linux
- P4, > 2 GHz processorsnelheid
- composite of S-Video PAL output t.b.v. CRT monitor
- RS232, MIDI of USB port
- bij voorkeur fanless, diskless (en batteryless!)

c. de PC, software

1. observaties gedrag software – zie logboek voor details

- video
 - o na iedere 4 of 10 pulsen van de interface (afhankelijk van de richting, er zijn 180 plaatjes fwd en 480 frames achteruit) wordt het volgende beeld getoond
 - o bij fwd/rev en rev/fwd overgangen wordt de nieuwe beeldsequentie gestart met beeld 1
 - o bij iedere richtingsovergang (fwd-pause-rev) is er 14 frames zwart video te zien voor het nieuwe beeld verschijnt.
 - o na 10 seconden zonder beweging wordt teruggesprongen naar het pauzebeeld.

- audio
 - o de telling van het aantal pulsen tijdens de laatste 160mS wordt volgens de tabel omgezet in een weergavefrequentie voor de volgende 160 mS
 - o de richtinginformatie van iedere puls bepaalt onmiddellijk de keuze van het gespeelde audio (fwd/rev)
 - o de audioweergavepointer verandert niet van plaats bij fwd/rev en rev/fwd overgangen, de hardware van het audiokastje selecteert alleen een andere eprom
 - o het uitgaand audio wordt op 3,3 KHz (en harmonischen) met een low-pass filter gefilterd

2. programma ontwikkeling, keuze programmeeromgeving

Om te onderzoeken of de vastgelegde gegevens voldoen voor het maken van een goede emulatie moet op basis van deze gegevens een computerprogramma geschreven worden.

Mijn voorstel is, de software ontwikkeling te faseren.

Eerst wordt een emulatie in Delphi geschreven.

Zo kan snel worden vastgesteld of de formele omschrijving van het werk voldoet.

Om deze software-emulatie met de installatie te kunnen verbinden moet de I/O-interface (microcontroller en level-aanpassing) ontwikkeld en getest worden.

Dezelfde interface kan voor de uiteindelijke emulatie gebruikt worden.

Het resultaat van het testprogramma kan, nu de originele hardware nog werkt, vergeleken worden met het origineel.

Dan wordt – op basis van de geverifieerde gegevens - het programma in een 'toekomstzekere' omgeving geschreven.

Voor de uiteindelijke programmeeromgeving zijn een aantal mogelijkheden:

- Borland Delphi
- Microsoft Visual C++, C# of VB (Visual Studio .NET)
- MAX/Msp/Jitter
- Pure Data.
- Mono (Novell Open Source .NET)

concept emulatievoorstel 'Revolution'

Gebruik van MAX/Msp, MS Visual Studio of Borland Delphi vraagt om vastleggen van gebruikte hardware, operating system etc.: geen van deze componenten kan later nog vervangen worden door een recentere versie of zonder ondersteuning door de fabrikant gereconstrueerd worden. Waarschijnlijk zullen hiermee ontwikkelde programma's op een PC uit 2016 niet meer werken.

Pure Data of Mono leveren uit oogpunt van houdbaarheid van de emulatie veruit het mooiste resultaat: van Pure Data, Mono maar ook van het gebruikte OS (linux) is source code beschikbaar. Dat geldt ook voor de C-compiler, de audio- en video-libraries en alle andere tools die nodig zijn.

Voor Mono moet nog onderzocht worden of alle functionaliteit die voor de emulatie nodig is inderdaad voorhanden is.

Een correcte emulatie in één van deze omgevingen kan in de toekomst op basis van de source code voor een andere computerarchitectuur, een ander processorplatform etc. gereconstrueerd worden. Op lange termijn (> 20 jaar) is de open source oplossing de enige die de zekerheid biedt dat zo'n reconstructie onafhankelijk van informatie of software derden kan worden uitgevoerd.

d. begroting

1. I/O-interface ontwikkeling, Delphi-emulatie, functietest

raming uren revolution werkzaamheden simulatie (herprogrammeren)

	A	B
Onderzoek/programmering		
tellen sensor omwentelingen per 360 draai (of 390)	1	
loggen en analyseren data naar laserdisk player (relatie teller/beeldpositie)	5	
meten gedrag schakeling, koppeling audio-frequenties aan digitale input waarden	6	
correleren sensor rpm -> audio-frequentie	4	
uitwerken schema elektronica	24	
hardware interfacing rotatiesensor -> USB	8	8
uitlezen rotatiesensor (serieel)	4	4
digitaliseren videomateriaal, omzetten naar 400 jpg's 720x576 (geen correctie)	4	
uitlezen eproms, omzetten naar wav files	4	
programmeren video-output	25	15
programmeren audio-loops	20	8
testen programma functionaliteit	16	4
Opbouwen/afbreken installatie voor uiteindelijke test	10	4
installeren PC	8	0
documentatie: programma source, interfacedoosje source + commentaar	12	4
totaal uren	151	47

Kolom A bevat de uren die bij aanvang van het project ingeschat waren voor een emulatie.

In kolom B staan de uren die nog aan het project besteed moeten worden om tot een sluitende 1^o fase emulatie + praktijktest te komen.

2. Pure Data of Mono applicatie

Op basis van een correcte omschrijving wordt het programmeren in PD op ca 40 uur geraamd.

Daarnaast moeten operating system, gebruikte tools etc. volledig in source worden vastgelegd en moet op basis van uitsluitend de vastgelegde informatie het systeem herbouwd worden. Naar schatting 60 uur werk.

Deze urenraming is een grove schatting, er zal met de uitvoerenden verder overlegd moeten worden om de precieze omvang van de werkzaamheden (m.n. de documentatie en het testen van de uiteindelijke versie) goed in kaart te brengen.

14/06/2006 aantekeningen n.a.v. opbouwen van de installatie

- Monitor Sony cube monitor (PVM2950?), composite PAL video in
- Netkabel (lange zware rubberen kabel)
- Sleepcontacten, custom made, deel van het frame
- Frame, duwstang, gebouwd door Huib Nelissen (Nelissen Decorbouw)
- PC - XT, ws. 8 MHz, geen HD
- Floppy bevat DOS boot, executable en tabel met parameters
- Kastje met custom elektronica
 - via LPT poort aan PC verbonden, afspeelsnelheid samples wordt d.m.v. D/A converter bepaald
 - ingang voor Sharp rotatiesensor type GP1R52C, 0/+5,0-doorgang,A,B (signaal gaat naar LM339 voor detectie). (? 1 puls per graad, 360/omw?)
 - geluidsrichting (keuze eprom) direct door rotatierichting bestuurd
 - bevat 2 e(e)proms: 27C512DC (AMD) en M5M27512P (Mitsubishi)
 - geluids-afspeelsnelheid door PC adhv delta waarde via printerpoort, D/A converter->opamp
- schrijver software: Gideon May
- maker hardware?
- na 10 seconden valt het beeld terug op steeds dezelfde still ('pauzebeeld')
- vanuit still bij beweging naar 1 van beide kanten een zwart tijd van ca 20 frames.
- bij omschakelen links-rechts of omgekeerd ca 6-10 frames zwart.
- 200 frames worden in ca 390 graden afgespeeld.

! meten ! omtrek sensorwiel + straal opname
! meten ! duur audio-fragment bij lage snelheid
! meten ! audio samplerate bij verschillende input snelheden - analyseer functie tabel
! onderzoek ! werking van de opstartparameters van het programma
! onderzoek ! gebruik van 'dosbox' PC emulator, data van seriële en LPT poort doorsluizen naar applicaties voor video en audio.

klomp's ad hoc oplossing:

- verbind de sensor met ser. poort (via microcontroller en eventueel RS232/USB adapter) van een solid state PC, die verder alle bewerkingen uitvoert: audio-scrub en video frames vertonen.
- beeldmateriaal: gedigitaliseerde frames van laserdisk of liever U-Matic tape (master).
- audio: gekopieerd uit de eproms.

nodig: geheugen van 200 x uncompressed frame 720x576 = 250MB video data, 128K audio.

linux systeem, usb boot. 3 processen, 1 voor video (elke graad 1 verder of terug), 1 voor audio, volgens analyse kastje gideon may, 1 voor uitlezen encoder en aansturen audio/video

10/7/2006: eerste onderzoek 'revolution'

Doel onderzoek:

- vaststellen functies interface kastje
- d.m.v. analyse controleren of 'black box' waarnemingen correct zijn

Werkwijze:

De apparatuur die deel uitmaakt van 'revolution' wordt op een werktafel geplaatst en waar nodig voor metingen met een oscilloscoop ontmanteld (m.n. het audio-kastje). Alles wordt aangesloten zoals in de installatie, m.u.v. audio, dit wordt via een externe versterker weergegeven.

1. rotatiesensor

- a. merk/type-aanduiding: Sharp GP-1R52C 360P/R
 - 1 (BLACK) . 0V
 - 2 (RED) . 5V
 - 3 (WHITE) . INDEX
 - 4 (YELLOW) . SIG A
 - 5 (GREEN) . SIG B
- b. werking: de sensor geeft een positie signaal in de vorm van 2 gelijkspanningsniveaus (SIG A en B) die ten opzicht van elkaar 90 graden in fase verschoven zijn. Deze signalen wisselen 360 x per omwenteling van polariteit, minimale waarde 1,2V, maximaal 3,2V
- c. uitlezing:

De signaalniveaus van SIG A en B worden ieder vergeleken met een instelbare referentiespanning.

Uit de tijdverhouding tussen deze signalen wordt een fwd/rev signaal afgeleid. Dit signaal is op TTL niveau voor de PC beschikbaar op pin 12 van de printer poort (/PE). Ook is het verbonden met de sample-player. Het pulssignaal wordt in het kastje door 2 gedeeld, dat levert 180 TTL pulsen (hoog->laag veranderingen) per omwenteling van het rubber wiel. Deze pulsen zijn door de PC te lezen op pin 10 van de printerpoort (/ACK).
- d. Berekening aantal pulsen per omwenteling van de installatie:
 - i. de straal van het slijtspoor van de rotatiesensor (op de stalen onderplaat) is gemeten op 28 cm. De lengte van dit spoor is $2 \cdot \pi \cdot 28 = 176$ cm.
 - ii. de diameter van het wiel op de sensor is 5,8 cm. De omtrek van dit wiel is $\pi \cdot 5,8 = 18,22$ cm
 - iii. hieruit volgt een totaal van $176/18,22 = 9,66$ omwentelingen per rondgang. Per omwenteling gaan er dus (ca.) 1738 pulsen naar de PC.

2. observaties t.a.v. de weergave van video door de installatie

- a. analyse van het beeldmateriaal
 - i. de 'animatie' bestaat uit 180 frames (720x576, PAL interlaced video)

logboek 'Revolution'

- ii. de 'molensteen' beslaat 478 frames – formaat idem
Het videomateriaal bleek op eigenaardige wijze op de beeldplaat te staan. Dit is gedaan om een langer stuk video dan de max. 400 frames die een beeldplaatspeler normaal 'seamless' kan lassen toch beeld voor beeld beschikbaar te maken. Het video is 'interleaved' op de beeldplaat gezet, in blokken van 48 frames:
de 48-frame-blokken 0 1 2 3 4 5 6 7 8 9 staan in de volgorde 0 1 9 2 8 3 7 4 6 5 op de plaat zodat er sprongen van maximaal 96 frames gemaakt worden.
- iii. het pauzebeeld: 1 frame
- b. zwart
 - i. bij omschakelen van voor- naar achteruit en andersom is er 12-14 frames zwart beeld
 - ii. bij omschakelen van pauzebeeld naar fwd/rev en omgekeerd: 10 frames zwart.
- c. snelheid van het afspelen
uit het gedrag van de installatie is te zien dat de beelden 1 voor 1 worden aangeroepen door de computer. Zeer snel draaien van de sensor (mogelijk omdat de installatie niet gemonteerd was tijdens dit onderzoek) geeft na ruim 10 omwentelingen weer het beginbeeld, waarbij duidelijk beelden gemist worden. Ook heel langzaam bewegen van de sensor heeft geen invloed op de telling van de pulsen.
 - i. snelheid afspelen forward (animatie): per 10 pulsen van de sensor springt het video 1 frame verder (ongeacht de tijd tussen de pulsen) -> 18 frames per omwenteling van het sensorwiel -> 180 frames in 10,39 omwentelingen (ruim 1 x rond duwen).
(deze '10' is één van de parameters waarmee het DOS programma wordt gestart, de parameter div fwd)
 - ii. afspelen reverse (molensteen): per 4 pulsen een nieuw beeld -> 45 frames per omwenteling van het sensorwiel -> 468 frames per rotatie.
(ook dit getal '4' komt voor in de programma-aanroep)
- d. conclusie
De computer telt het aantal pulsen dat binnenkomt. Afhankelijk van de richting krijgt na 10 of na 4 pulsen de laserdiskspeler een commando om een volgend beeld te tonen. Als het laatste beeld van een reeks vertoond is, wordt teruggesprongen naar het eerste, het videomateriaal wordt 'gelooped'.
Tegelijk is er een time-out functie actief in de software die na 10 seconden geen pulsen te hebben ontvangen het pauzebeeld oproept.

Dit levert de volgende - voorlopige - pseudocode op voor het koppelen van video beelden aan pulsen van de sensor:

logboek 'Revolution'

forever loop

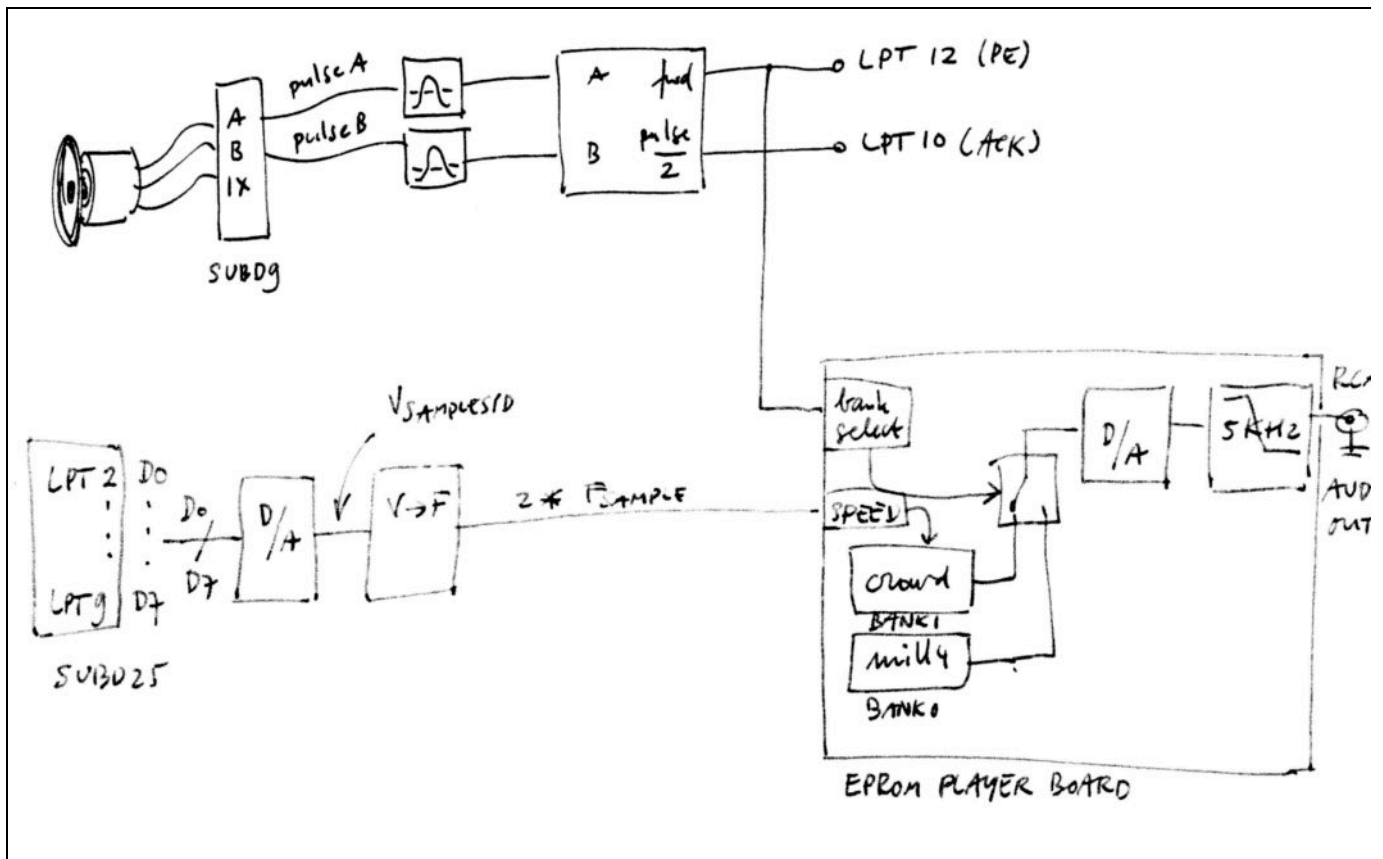
```
// bij veranderde draairichting eerste beeld van juiste reeks laten zien
if ouderichting != richting then
  pcount = 0 // reset puls teller
  if richting == fwd then
    fmod = 10 // deeltal voor de puls teller
    anim( 0) // spring naar frame 1 van animatie
  else
    fmod = 4
    steen( 0) // naar frame 1 van de molensteen
  end if
  ouderichting = richting
end if

// bij wel of geen puls van de sensor
if puls then
  timeout = 0 // reset pauzebeeld timer
  pcount++ // verhoog framecounter
else
  if timeout > 182 then
    pauze(0) // spring naar pauzebeeld
    timeout = 182 // en blijf daar (klopt niet, zie verderop!)
  end if
  puls = false
end if

// controleer of het juiste aantal pulsen is geweest, geef zonodig
commando voor
// volgend beeld
if pcount % fmod == 0 then
  if richting = fwd then
    anim( pcount / fmod)
  else
    steen( pcount / fmod)
  end if

// elke 1/18.2 seconden (BIOS tick van DOS PC)
if tick then
  timeout++
end if
```

end loop



figuur 5, de audio-box

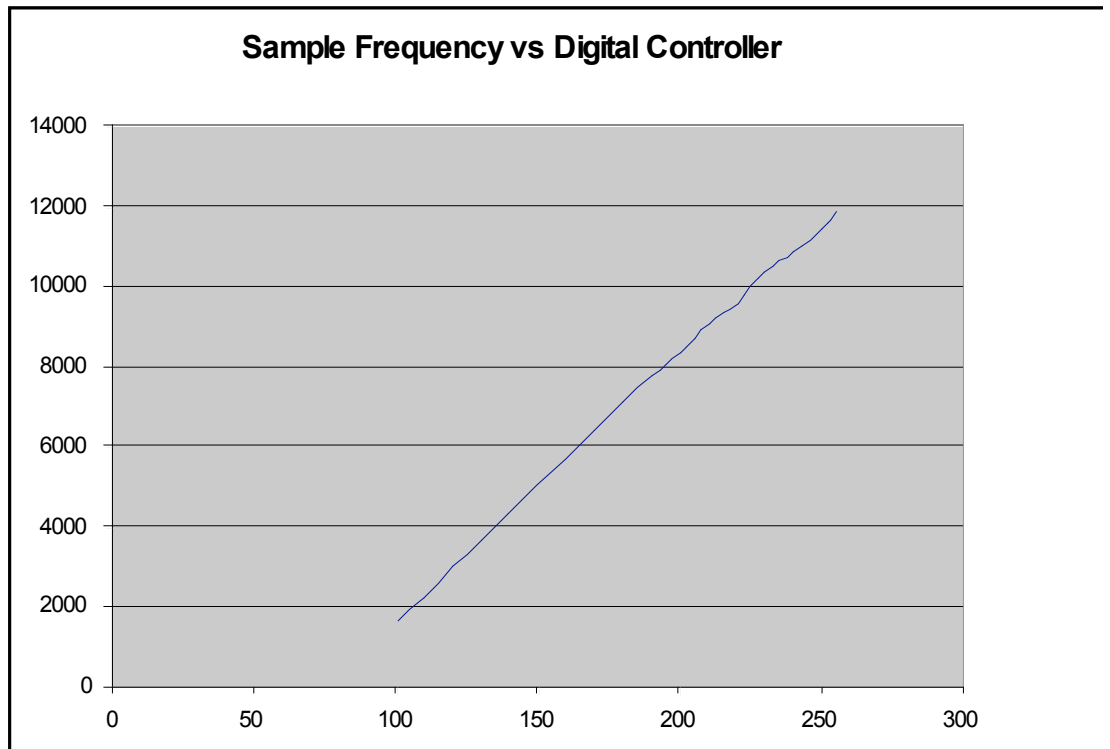
3. audio-observaties

- a. besturing van de afspeelsnelheid:

de PC kan via de printerpoort een 8-bits getal naar het audio-kastje sturen. Dit getal bepaalt de snelheid waarmee de geluidssamples worden afgespeeld. De audio player springt vanzelf terug naar het begin van de sample als het einde bereikt is.

Dmv een oscilloscoop en een testprogramma is vastgelegd, welke sampleclock bij welk getal hoort.

Hiervoor zijn getallen 100..255 gebruikt, dit zijn de waarden die in de installatie gebruikt worden (dat blijkt uit de inhoud van de tabel 'SOUND.TBL' die bij het starten van het programma geladen wordt).



figuur 6

- b. Deze grafiek geeft het verband tussen aangeboden getal en de gemeten afspeelfrequentie (Hz).
- c. relatie samplerate en sensor

Hoe sneller de sensor draait, des te sneller wordt het geluid afgespeeld, tot een bepaalde maximum afspeelsnelheid bereikt is (waarde 255, ca 12KHz)

 - i. onderzoek: de sensor wordt met zo'n snelheid gedraaid dat het geluid op maximale frequentie afspeelt. Dit levert (meting met oscilloscoop) een pulsduur van ca 5 mS op.
 - ii. veronderstelling: de software telt het aantal binnenkomende pulsen gedurende een bepaalde tijdseenheid. Gezien de beperkte mogelijkheden van de PC-XT is waarschijnlijk de BIOS tick als tijdseenheid gebruikt. Deze tick heeft een 'venster' van 55 mS. Tellen we het aantal 5 mS pulsen gedurende 55 mS dan levert dat per tick 11 pulsen op. Dit komt overeen met de tellerwaarde die in SOUND.TBL aan de maximale samplesnelheid is gekoppeld, als er 3 ticks lang wordt geteld (165 mS).
 - iii. vervolgonderzoek: de relatie tussen afspeelsnelheid en beweging is onvoldoende nauwkeurig vastgelegd. Voor een nauwkeurige meting moeten bekende frequenties worden aangeboden aan de PC (in plaats van het pulswiel) en aan de hand daarvan het precieze gedrag van de afspeler worden gemeten. Voor een voorlopige (principe-)test voldoet de voorhanden informatie wel.

- iv. observatie: de audio-player speelt altijd minimaal 170 mS geluid. Dit is ongeveer 3 x 55 mS, ook dit doet een relatie met de BIOS tick vermoeden.

4. analyse van de DOS bestanden

```
config.sys:
-----
buffers = 20
files = 20
device=x00.sys
-----
```

device x00.sys roept een 'fossil driver' aan.
RS232 access is onder DOS beperkt mogelijk en deze driver zorgt voor gestandaardiseerde communicatie met de seriele poort.
De applicatie heeft dus geen rechtstreekse toegang tot de seriele poort hardware.
Wel zijn de poort parameters (welke COM: poort, interrupt, adres) vast gecodeerd in het pserv.exe programma.

```
autoexec.bat:
-----
@echo off
cls
pserv 4 10 10
-----
```

'pserv 4 10 10' is nogal cryptisch, maar wordt duidelijk wanneer je het programma werkelijk draait, het meldt bij een aanroep zonder parameters:

```
Usage : D:\DATA\PROJECTS\REVOLU\BOOTFL\PSERV.EXE
<forward div> <reverse div> <timeout>
```

De parameters '4 10 10' staan dus voor resp. deeltal vooruit, deeltal achteruit, time-out.

De time-out van 10 seconden was al bij het observeren van de installatie geconstateerd.

De deeltallen geven waarschijnlijk aan na hoeveel pulsen voorwaarts of achterwaarts er een nieuw beeld moet verschijnen.

logboek 'Revolution'

Dan laadt het programma pserv nog de volgende tabel ('sound.tbl'):

0 0	208 22	255 44
100 1	213 23	255 45
105 2	218 24	255 46
110 3	224 25	255 47
115 4	229 26	255 48
120 5	234 27	255 49
125 6	239 28	255 50
131 7	244 29	255 51
136 8	249 30	255 52
141 9	255 31	255 53
146 10	255 32	255 54
151 11	255 33	255 55
156 12	255 34	255 56
162 13	255 35	255 57
167 14	255 36	255 58
172 15	255 37	255 59
177 16	255 38	255 60
182 17	255 39	255 61
187 18	255 40	255 62
193 19	255 41	255 63
198 20	255 42	
203 21	255 43	

logboek onderzoek 'Revolution'

De relatie tussen het aantal pulsen van de rotatiesensor in een bepaald interval (hoe groot is niet duidelijk) en de waarde die naar de LPT-poort gestuurd wordt om de afspeelsnelheid van het geluid te bepalen wordt hierin waarschijnlijk vastgelegd.

Hoe groot dit meetinterval moet verder onderzocht worden.

Gezien de beperkte timing mogelijkheden in DOS zal het interval waarschijnlijk in 1/18 seconden (55 mS) gedefinieerd zijn (BIOS ticks).

25/7/2006 audio, eeprom

Doel: vastleggen van de audio-informatie

Werkwijze: uitlezen van de Eprom-inhoud met behulp van een programmer.

Een eerste poging om de inhoud van de Eproms vast te leggen (jan 2006?) leverde slechts bij 1 van de 2 geheugen-IC's resultaat op.

De Eproms uit de audio-box blijken met aan andere programmer beide leesbaar, de audio data voor beide kanalen is nu veiliggesteld.

Bij sampelen van geluid uit het kastje blijkt een nieuw probleem voor de emulatie: de afspeelsnelheid verandert weliswaar, maar de hoog-af filtering vindt altijd plaats vanaf 3,3 KHz. Dit blijkt uit spectraalanalyse van de gesampled audio-output van het kastje.

Als gevolg moet bij emulatie ook real-time audio gefilterd worden of moet er een hardware filter tussen computer en audioversterker geplaatst worden.

20/9/2006 vervolgonderzoek

Doel: vaststellen exacte relatie rotatiesensor – sample snelheid

Werkwijze: De apparatuur die deel uitmaakt van 'revolution' wordt op een werktafel geplaatst en waar nodig voor metingen met een oscilloscoop ontmanteld (m.n. het audio-kastje).

Bij Montevideo wordt de elektronica die nodig is voor het functioneren.

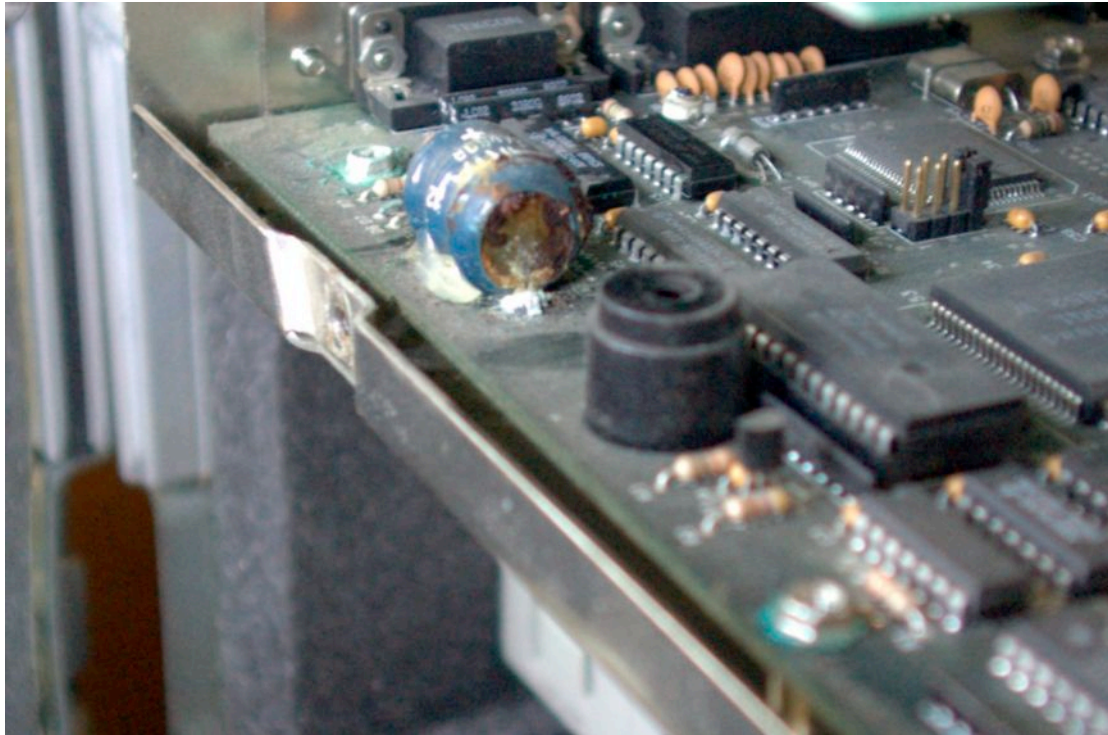
Eerst wordt gecontroleerd of alle elektronica nog werkt, dat blijkt het geval.

Dan, net voor het aansluiten van de signaalgenerator, ontstaat bij het meten van de voedingsspanning een sluiting in de voeding van het audio-kastje.

Hierna werkt de besturing van de audio-interface niet meer: de printerpoort van de Twinhead PC is uitgevallen. Of deze uitval door de sluiting veroorzaakt is is niet zeker. Gezien het vervolg kan er een samenloop van omstandigheden zijn geweest. Met een andere PC dan de originele Twinhead XT wordt de werking van de audio-interface, tot nu toe door ons beschouwd als het meest kwetsbare onderdeel van de installatie, gecontroleerd. De audio-box blijkt helemaal in orde.

Om te zoeken naar de oorzaak van de uitval van de printerpoort wordt de kast van de Twinhead computer geopend, tot onze schrik blijkt het moederbord van deze computer ernstig aangetast door weggelekte vloeistof uit de NiCad back-upbatterij (figuur 7).

We besluiten een vervangende PC te zoeken.



figuur 7

25/9/2006, test nieuwe besturingscomputer

De elektronica voor 'Revolution' wordt opnieuw opgebouwd en aangesloten. Met een Pentium-I 150MHz computer in plaats van de oorspronkelijke Twinhead SS200 PC wordt getest of de installatie werkt. Deze computer is gekozen omdat de BIOS het mogelijk maakt de printerpoort op verschillende hardware adressen te laten werken; het vermoeden bestaat dat het besturingsprogramma van de installatie een bepaald I/O-adres verwacht.

De besturing van de laserdisk werkt, dat blijkt uit het feit dat na 10 seconden het pauzebeeld verschijnt. Maar er is geen enkele reactie op binnenkomende pulsen, ongeacht verschillende instellingen van printerpoort-adres. We besluiten de installatie te transporteren naar klomp kunst & electro in Hengelo waar de opstelling langer kan blijven staan zodat rustig verder gezocht kan worden naar de oorzaak van de ontstane problemen.

1 en 2/10/2006 onderzoek Twinhead computer

1. aanmaak 720KB bootfloppy, MSDOS6.22/qbasic/debug
2. onderzoek met debug.exe: geen van de 2 fysiek aanwezige printerpoorten wordt door de BIOS herkend.
3. experiment: jumpers op de MONO/LPT kaart verwijderd. Nu ziet de BIOS afhankelijk van de jumpersetting een printerpoort op 0x3BC, 0x378 of 0x278. Besloten wordt met deze 'nieuwe' printerpoort de installatie weer op te bouwen en te testen.
4. Nu blijkt dat ook de RS232 poort is uitgevallen, waarschijnlijk is alle data uit de configuratie-RAM gewist door een zeer lage accuspanning.

5. Het moederbord van de Twinhead PC wordt uitgebouwd, zo goed mogelijk gereinigd (met alcohol en contactspray) en de lekkende batterij wordt vervangen. Na samenbouwen lukt het niet d.m.v. toetscombinatie in het BIOS setup menu te komen, speciale setup software voor de Twinhead computer is niet voorhanden, kortom: printer- en serieel poort worden niet gedetecteerd en blijven onbruikbaar.
6. Om te onderzoeken of de software specifieke kenmerken van de IBM-XT architectuur gebruikt wordt nogmaals een andere computer geprobeerd (Commodore PC20-III, een PC-XT uit dezelfde tijd als de Twinhead PC). Bijna alles werkt: de PC communiceert met de audio-box, de laserdisk wordt bestuurd, maar er is geen reactie op pulsen van het wielkje.
7. Om te controleren op mogelijk weggeetste printsporen wordt nogmaals het moederbord van de Twinhead uitgebouwd, en dan valt het kwartje: onderop het moederbord is een 'patch' aangebracht die het signaal van de rotatiesensor rechtstreeks doorsluisst naar een aansluiting op de ISA bus. Uit de aansluitgegevens van printer en ISA bus valt op te maken dat /ACK door middel van een op de print gesoldeerde draad verbonden is aan IRQ3.
8. Dmv dis-assembly van het programma PSERV.EXE wordt vastgesteld dat het programma een printerpoort verwacht op adres 0x378 ('hard-coded I/O access').
9. De extra printerpoort van de Twinhead PC ingesteld op dit adres. Op de parallel printer kaart wordt een patch gemaakt van pin 10 LPT naar IRQ3/ISA, en ziedaar: de installatie werkt weer!

4/10/2006 aanvullend onderzoek Revolution – relatie geluidssnelheid en verdraaiing

Bij het draaien van de installatie past de snelheid van het afgespeelde geluid zich aan aan de omwentelingssnelheid van de installatie.

Doel: dit onderzoek moet vastleggen wat precies de relatie is tussen de snelheid van draaien en de afspeelsnelheid van de samples.

Werkwijze: deze meting wordt in 2 stappen gedaan:

- Meetopstelling 1: eerst wordt vastgelegd welke stuurgetallen de PC bij verschillende snelheden van het wielkje genereert, een meting waarvoor alleen PC, functiegenerator en een speciaal hiervoor gemaakte interface nodig zijn.
- Meetopstelling 2: in deze opstelling wordt vastgelegd welke weergavesnelheid bij welke ingangswaarde van de audio-box hoort. Hiervoor zijn (een) PC en het audio-kastje nodig.

De tijdmetingen worden gedaan met behulp van een oscilloscoop; dit levert redelijke precisie op, naar schatting 5%.

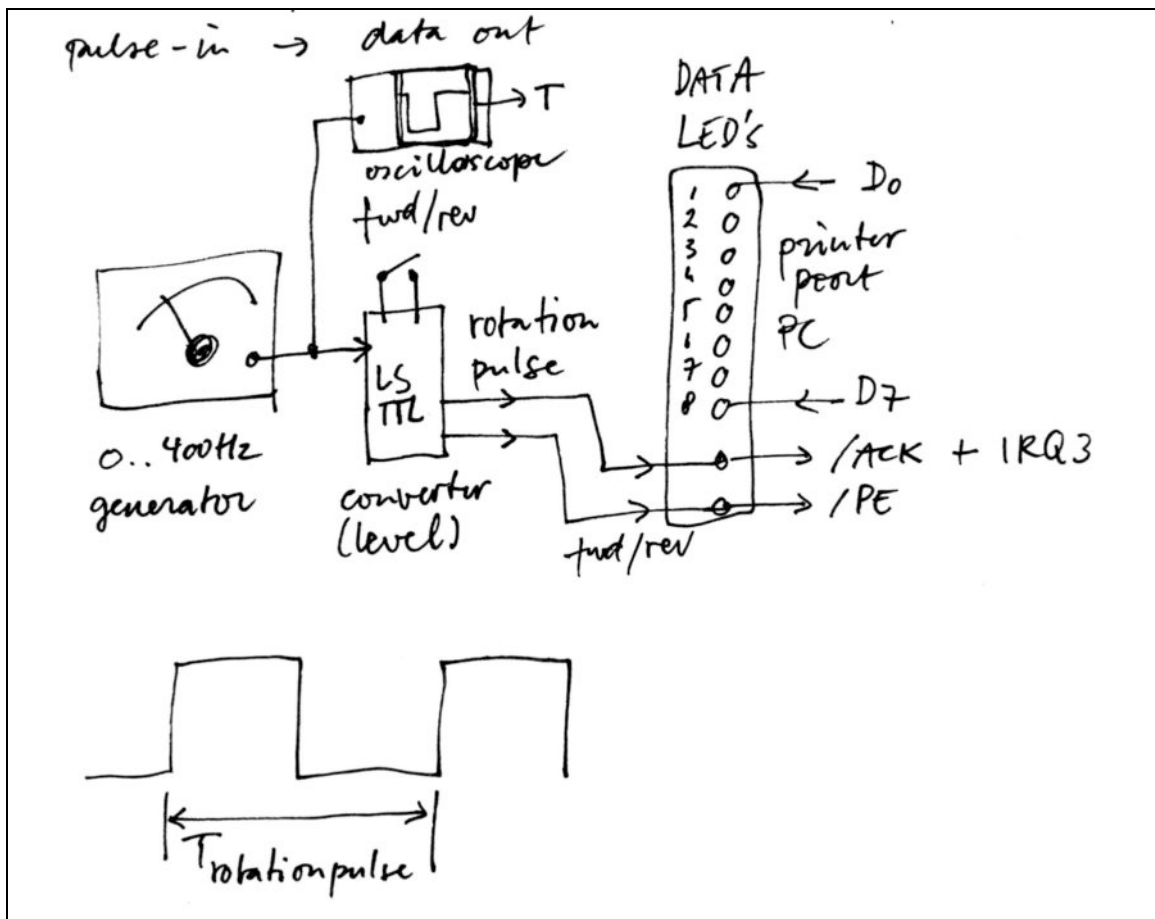
Observatie vooraf: voor- of achteruit draaien maakt voor de afspeelsnelheid van het geluid niet uit.

Doel 2:

Deze dag is nogmaals – en met succes - onderzocht of het mogelijk is, de Twinhead PC te vervangen door een modernere computer.

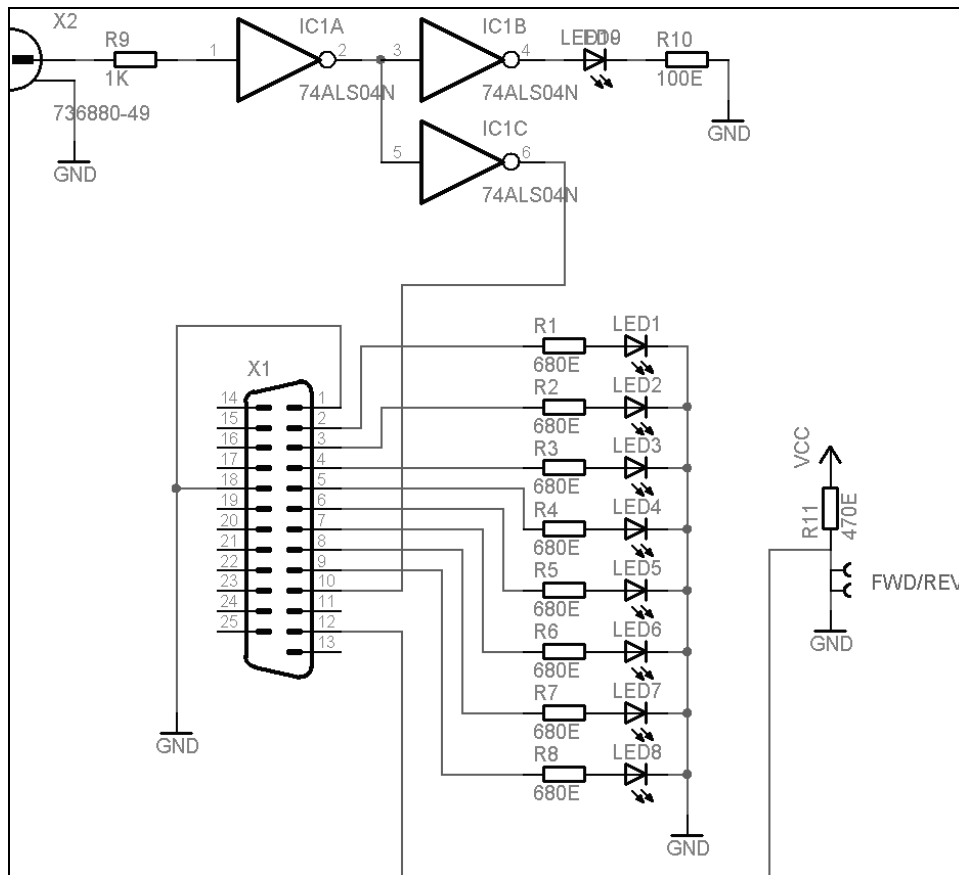
Meetopstelling 1 (figuur 8, 9):

Funciegenerator (Philips PM5167) -> LS-TTL interface -> SUBD25 PC LPT aansluiting p.10 (/ACK, pulse)
 LPT 2..9 aan LED via 680^E
 LPT 12 aan jumper 0/+5V (/PE, fwd/rev)
 Software: Diskette K90043-E-3: revolution boot disk



figuur 8

logboek 'Revolution'



figuur 9

Bij deze meting is het audio-kastje niet aangesloten, er wordt alleen gekeken naar de getallen die de PC bij verschillende snelheden van de rotatiesensor uitgeeft (de 8 databits op de printerpoort).

Doel: meten relatie puls-frequentie (uitgangssignaal wiel) en uitgangswaarden die de weergavesnelheid van de audio interface bepalen.

Meetmethode: de frequentie van de generator wordt langzaam opgevoerd tot de status van de datalijnen verandert. Als dit net is gebeurd, en de data stabiel is, wordt de pulstijd (T-rotationpulse) d.m.v. de oscilloscoop uitgelezen en genoteerd met daarnaast de nieuwe data waarde.

Resultaten:

- a. Bij zeer trage pulsen ($>160\text{ms}$): de pulsen worden ongeacht de pauzesituatie doorgeteld, dus wordt er na 4 (rev) of 10 (fwd) pulsen naar het eerste beeld van de sequentie gesprongen waarbij 160 ms lang de waarde 100 naar de audio interface gestuurd wordt. Als de pulsen voldoende traag zijn springt 10 seconden later de player terug naar pauzebeeld.
- b. Bij 'normale' pulsen: tabel relatie pulstijd – output getal op printerpoort naar audio box:

logboek 'Revolution'

T (mS)		DATA as read	binary	decimal	On for 160 mS, then off for 3 or 9 pulses
1000	Ms	00100110			
160		00100110	1100100	100	
80		10010110	1101001	105	
53		01110110	1101110	110	
40		11001110	1110011	115	
32		00011110	1111000	120	
27		10111110	1111101	125	
23		11000001	10000011	131	
20		00010001	10001000	136	
17.6		10110001	10001101	141	
15.8		01001001	10010010	146	
14.2		11101001	10010111	151	
13		00111001	10011100	156	
12.2		01000101	10100010	162	
11.2		11100101	10100111	167	
10.8		00110101	10101100	172	
10		10001101	10110001	177	
9.6		01101101	10110110	182	
8.8		11011101	10111011	187	
8.4		10000011	11000001	193	
8		01100011	11000110	198	
7.6		11010011	11001011	203	
7.4		00001011	11010000	208	
7		10101011	11010101	213	
6.6		01011011	11011010	218	
6.4		00000111	11100000	224	
6.2		10100111	11100101	229	
5.9		01010111	11101010	234	
5.7		11110111	11101111	239	
5.5		00101111	11110100	244	
5.4		10011111	11111001	249	
5.2		11111111	11111111	255	
2.5		11000100	100011	35	

De uit observatie verkregen DATA-waarden in deze tabel komen overeen met de waarden in SOUND.TBL. Daarmee is aangetoond dat deze tabel de vertaling van getelde pulsen -> aanstuurgetal voor de geluidssnelheid bepaalt.

Meetopstelling 2:

Meting: relatie waarde op printerpoort en audio-frequentie

Opstelling: Twinhead PC met daaraan verbonden de audio-box.

Software: eenvoudig QBASIC testprogramma.

Werkwijze: getallen uit de vorige reeks (DATA) invoeren en daarna de pulslengte van 1 sample meten (de tijd van één puls op adreslijn 0 van de Eprom).

Er worden 2 series metingen verricht om afleesfouten op het oscilloscoopscherm uit te middelen.

T-rotation-pulse (mS)	DATA LEDstate	DATA decimal	Measurements				result average
			series 1		series 2		
			Tsample mS	Fsample Hz	Tsample mS	Fsample Hz	Fsample Hz
>160	0 / 100 *						
160	1100100	100	0.6300	1587	0.6400	1563	1575
80	1101001	105	0.5200	1923	0.5300	1887	1905
53	1101110	110	0.4400	2273	0.4550	2198	2235
40	1110011	115	0.3850	2597	0.3950	2532	2565
32	1111000	120	0.3400	2941	0.3500	2857	2899
27	1111101	125	0.3000	3333	0.3150	3175	3254
23	10000011	131	0.2675	3738	0.2700	3704	3721
20	10001000	136	0.2450	4082	0.2500	4000	4041
17.6	10001101	141	0.2250	4444	0.2300	4348	4396
15.8	10010010	146	0.2100	4762	0.2150	4651	4707
14.2	10010111	151	0.1950	5128	0.1975	5063	5096
13	10011100	156	0.1825	5479	0.1875	5333	5406
12.2	10100010	162	0.1700	5882	0.1720	5814	5848
11.2	10100111	167	0.1600	6250	0.1630	6135	6192
10.8	10101100	172	0.1500	6667	0.1550	6452	6559
10	10110001	177	0.1400	7143	0.1480	6757	6950
9.6	10110110	182	0.1380	7246	0.1400	7143	7195
8.8	10111011	187	0.1320	7576	0.1350	7407	7492
8.4	11000001	193	0.1240	8065	0.1280	7813	7939
8	11000110	198	0.1200	8333	0.1230	8130	8232
7.6	11001011	203	0.1160	8621	0.1170	8547	8584
7.4	11010000	208	0.1110	9009	0.1140	8772	8890
7	11010101	213	0.1080	9259	0.1090	9174	9217
6.6	11011010	218	0.1040	9615	0.1060	9434	9525
6.4	11100000	224	0.0990	10101	0.1020	9804	9952
6.2	11100101	229	0.0960	10417	0.0980	10204	10310
5.9	11101010	234	0.0930	10753	0.0960	10417	10585
5.7	11101111	239	0.0900	11111	0.0930	10753	10932
5.5	11110100	244	0.0880	11364	0.0910	10989	11176
5.4	11111001	249	0.0850	11765	0.0880	11364	11564
5.2	11111111	255	0.0830	12048	0.0850	11765	11906

Combinatie van gegevens uit SOUND.TBL en het gemeten gedrag:

Bij 5,2 mS (192 Hz) wordt de maximale snelheid voor de audio weergave bereikt. De tabel geeft aan dat die maximale snelheid bereikt wordt bij het tellen van 31 pulsen. Dan wordt er $31 * 5.2 = 160\text{mS}$ lang gemeten. Dat klopt met de laagste snelheid die nog een stabiele waarde geeft: 160 mS oftewel 1 puls per meting. De pulsen van de rotatiesensor worden dus 160 mS lang geteld. Het resultaat van die telling wordt gebruikt om een nieuwe waarde voor de audio weergavesnelheid te kiezen, aan de hand van tabel SOUND.TBL

Doel 2: vervangende PC

Test: Twinhead XT vervangen door Pentium-I 150MHz (bj. ca 1996, ASUS P/155T moederbord).

Bij deze computer wordt alleen COM1 (adres 3F8, IRQ4) in de BIOS ingeschakeld, de gemodificeerde MONO/LPT kaart uit de Twinhead wordt gebruikt voor interfacing met signaalgenerator en LEDs.

Resultaat: installatie werkt ook op 'moderne' PC.

Voorwaarden vervangende PC:

- a. ISA slot (t.b.v. gemodificeerde LPT kaart)
- b. LPT poort uitschakelbaar
- c. COM poort instelbaar op adres 3F8, IRQ4

Te verwachten is, dat ook een recentere PC met ISA slot (anno 2006 nog in productie) na modificatie goed zal werken.

7/10/2006 aanvullend onderzoek Revolution – videosamples

Doel: vastleggen beeldinhoud

Werkwijze: digitaliseren output LDP

De output van de installatie wordt als DV video gedigitaliseerd. Door de installatie langzaam te triggeren wordt het videomateriaal in clusters van 4 gelijke frames uitgegeven. Daaruit is voor digitalisatie steeds het beste frame gekozen. Vooral op de plekken waar een 'seamless jump' heeft plaatsgevonden bleek het eerste frame na de jump veel drop-outs te bevatten.

12/10/2006 aanvullend onderzoek Revolution – audiosamples

Doel: conserveren audiosignaal

Werkwijze: digitaliseren audiosignalen bij 8 KHz sampling snelheid

Bij een eerste test naar real time filtering (mbv ESS library in "processing") blijkt de geluidskwaliteit van de eeprom data verschrikkelijk slecht.

Het bekijken van de audio samples in een sample-editor laat zien er eigenaardige sprongen in de sample zitten. Mogelijk is de bitvolgorde van de audio samples in

eproms niet normaal, m.a.w. er zijn datalijnen verwisseld tussen eprom en D/A converter. Géén van de bekende interpretaties voor 8-bit audio data (signed, unsigned, U-Law en A-Law) levert bruikbaar geluid op.

Vervolgonderzoek: een meting in het audio-kastje van 8 datalijn naar D/A converter kan een conversietabel opleveren. Dan kan er een programma geschreven worden om de audio data weer in de juiste bitvolgorde te krijgen.

14/10/2006 aanvullend onderzoek Revolution – audiosamples – 2

Doel: vaststellen data structuur audio in eproms

Werkwijze: onderzoek gesampled audio, datasheets AD converters, data analyse dmv processing programma

Nader onderzoek brengt aan het licht dat de D/A converter werkt volgens een oud telecom principe, "mu-Law". Dit blijkt uit het typenummer van de gebruikte converter, een Analog Devices AM6070DC.

Dit gegeven biedt aanknopingspunten voor de nominale sample snelheid: 8000Hz is de standaard weergave frequentie voor 8 bit mu-Law.

De aansluitingen van Eprom datalijnen naar de inputs van de converter zijn 1:1, er zijn geen bits van plaats verwisseld.

Wel worden er door de DA-converter mogelijk bits geïnverteerd vóór conversie. Dit is niet te achterhalen omdat er geen datasheet voor betreffende chip te vinden is.

Analyse van de raw files (de eprom inhoud) levert voor MILL4 en CROWD een verschillend beeld op, waarbij de CROWD sample een eigenaardige waardespreiding over de 8 bits vertoont. Mogelijk is de gebruikte eprom al data aan het verliezen, de hoogste bit, die de polariteit van het audiosignaal aangeeft, verandert half zo vaak van waarde als de lagere bits, de spreiding van verandering binnen de lagere bits is zeer gelijkmatig. Bij een 'normaal' audiosignaal veranderen hogere bits in de sample minder vaak van waarde dan lagere, dat is ook bij het eveneens mu-Law gecodeerde MILL4 sample het geval.

Conclusie: het is beter de output van de audio-box op goede kwaliteit te sampelen dan gebruik te maken van de eprom inhoud.

Daarvoor zijn 96KHz/16bits sampling settings gebruikt, het kastje kreeg als ingangswaarde voor de weergavesnelheid het getal 180 (ca. 7.2 KHz samplerate). Na digitalisatie blijkt er een sterkteverschil is van 3 dB tussen het CROWD en het MILL4 geluid, dit laatste is harder. De samples zijn na digitalisatie genormaliseerd, de software die de samples bij emulatie afspeelt moet dit sterkteverschil herstellen.

nov 2006 – mrt 2007: Vervolg werkzaamheden Revolution:

In deze periode is gewerkt aan het maken van een uitvoerbare emulatie van 'Revolution'. Na enige zijsporen en onderzoek naar mogelijke oplossingen resulteerde dit in een Pure Data programma, gecombineerd met een nieuwe rotatie-opnemer en een nieuwe interface voor deze opnemer (USB). De resultaten van deze uiteindelijke emulatie zijn ter afsluiting 1:1 vergeleken met die van de de originele hardware.

het programmeren van de emulatie

1. Processing test
 - a. In Java ('Processing') wordt een reeks beelden geladen waarnaar at random gesprongen kan worden. Bij het aantal beelden dat voor 'Revolution' nodig is wordt de laadtijd erg lang. Het geheugengebruik is zeer groot. Een andere benadering, nl een QuickTime movie beeld voor beeld toegankelijk te maken mislukt omdat de timing (in milliSeconden) geen herhaalbare framenuwkeurige access toestaat. Na een eerste verkenning besloten dat dit geen geschikte omgeving is, vooral als ook nog blijkt dat implementeren van het audio deel moeilijk of onmogelijk is: geen real time filtering, geen aanpassing van de afspeelsnelheid terwijl een sample speelt.
2. Delphi implementatie
 - a. Video

Op basis van een eerder ontwikkeld programma, nl een imagelist waarin de revolution beelden geladen worden die dan random toegankelijk zijn wordt de basis voor de video weergave geïmplementeerd.

De weergave werkt, er zijn wat problemen met het oprekken van de beelden naar full screen. Ivm het beperkte geheugen van de ontwikkel PC (1Gbyte) moeten de plaatjes in verkleinde vorm gebruikt worden waarbij kwaliteitsvelies ontstaat. Afgezien hiervan biedt deze oplossing de mogelijkheid de beelden met de nodige snelheid te tonen, voldoende voor een eerste test van het geobserveerde gedrag.
 - b. Audio

Windows Media Player, standaard media interface voor Delphi, ondersteunt geen weergave van audio met variabele snelheid en ook geen real time filtering, die nodig is om het hardware 3 KHz hoog-af filter te emuleren.

Er zijn 2 opties om dit voor elkaar te krijgen
 - i. Directsound (DX7 of hoger). Dit vraagt om een DirectX interface voor Delphi. Die blijkt te bestaan maar is slecht gedocumenteerd, niet actueel en vrij ondoordringelijk.
 - ii. FMOD: een game audio interface; platform onafhankelijk en goed gedocumenteerd stelsel van functies voor real time geluidsweergave, m.n. voor 3D geluid bij games ontwikkeld. FMOD is bovendien rechtenvrij voor niet commerciële toepassingen. Systeemonafhankelijk bovendien.

De installatie van FMOD blijkt vrij eenvoudig. Maar bij het programmeren komt aan het licht dat het een complexe omgeving is die veel experimenten vraagt om de vrij eenvoudige opgave het

Revolution geluid juist weer te geven (een continue loop, waarvan de weergavesnelheid gevarieerd kan worden, en een low-pass filter voor het resultaat) op te lossen.

Ik besluit dat de inspanning een nieuwe toolbox te leren beter besteed kan worden aan onderzoek naar de mogelijkheden van een oplossing in Max/Msp of Pure Data, uiteindelijk doel van de emulatie.

3. voorlopige Pure Data implementatie

a. video

in PD blijkt het vrij gemakkelijk te zijn een Quicktime video file beeld voor beeld aan te spreken. Daarmee is het video deel te realiseren.

Een eenvoudige patch wordt geschreven waarmee duidelijk wordt dat de weergave van een Quicktime video vlot verloopt (op een dual core processor / OpenGL1.5 graphics board)

b. audio

Ook weergave van geluid met een variabele samplerate blijkt goed te doen in PD. Het bijbehorende low-pass filter is gemakkelijk te implementeren.

c. implementatie algoritme

De analyse van de installatie is in pseudo pascal geschreven, gericht op een 'lineair' programma. Voor de objectgeoriënteerde structuur van Pure Data lijkt deze benadering minder geschikt.

Het ontwikkelen komt een stap verder met de ontdekking dat er meerdere video renderers in een patch kunnen zijn. Door het aan- en uitzetten ervan kan gekozen worden tussen weergave van één van beide beeldsequenties, van het pauzebeeld of een zwart beeld (dat laatste door alle renderers uit te zetten).

De verschillende video bronnen bevatten nu hun eigen weergever waartussen geschakeld kan worden.

d. opnemer: nu weergave van beeld en geluid in principe werken, wordt het tijd het gedrag van deze objecten te koppelen aan een rotatieopnemer. Gekozen wordt voor een 180 ppr opnemer van Kübler die TTL informatie uitgeeft (digitale signalen, 0 / 5V), als vervanging van de analoge opnemer die in de originele installatie gebruikt is. Het voordeel van een TTL interface is, dat de ingangsversterkers die in Revolution gebruikt zijn kunnen vervallen. Om de pulsen van de opnemer te interpreteren wordt een Arduino microcontroller board ingezet waarop een vrij eenvoudig programma draait (zie bijlage). Om niet afhankelijk te zijn van een specifieke hardware interface (zoals RS232 of USB) wordt een MIDI protocol gebruikt. Dit protocol wordt door de PD patch gelijktijdig vanaf een RS232 poort als vanaf een MIDI poort geïnterpreteerd, het maakt voor het programma niet uit of de data via MIDI of serieel binnenkomt.

De microcontroller deelt de pulstrein die bij draaiing van het wielje binnenkomt door 2 en geeft bij iedere tweede puls een signaal met richtinginformatie aan de host computer. Daarnaast is er een timer actief in de controller die na iedere 160 mS het aantal in dit interval getelde pulsen verstuurt. Hieruit kan de PD patch de snelheid van het wielje en zodoende de snelheid van het ronddraaien van de installatie bepalen.

e. Interpretatie van de opnemer data in Pure Data

- i. de fwd en reverse pulsen worden respectievelijk door 10 en door 4 gedeeld en besturen dan een beeldteller die steeds een volgend frame van de actieve Quicktime movie toont. Wanneer het laatste beeld getoond is wordt vanzelf naar frame 1 gesprongen.
 - ii. op iedere binnenkomende puls wordt een pauze-teller teruggezet op 0 seconden. Weet deze teller de 10 seconden te bereiken dan wordt na 14 frames (500mS) zwart het pauzebeeld getoond. Ook worden dan de weergave pointers van beide beeldreeksen op 1 gezet.
 - iii. de richtinginformatie die in iedere puls zit bepaalt onmiddellijk welk geluidskanaal wordt weergegeven
 - iv. bij verandering van de richting worden beide framepointers op 1 gezet, en volgt een zwart beeld gedurende 500 mS. De telling van binnenkomende pulsen loopt intussen door en bepaalt het getoonde frame.
 - v. het aantal pulsen gedurende de laatste 160 mS wordt gebruikt om op basis de geobserveerde/uitgelezen tabellen de weergavefrequentie van het geluid te bepalen. Een lineaire benadering zou mogelijk zijn, maar de sprongetjes in weergavesnelheid zijn karakteristiek voor de installatie.
- f. Eigenaardigheden
- i. De audiosamples die voor de documentatie gemaakt zijn blijken sterk verschillend van lengte (> 5%), terwijl vaststaat dat de originele samples voor de beide draairichtingen even lang zijn, ze zitten in een even grote Eprom. Ik neem aan dat er iets fout is gegaan bij het bepalen van de loop-points (dit is op het gehoor gebeurd) en gebruik de beide samples met hun feitelijke lengte. Dit betekent dat de 1:1 relatie in sample positie vervalt. Dit is niet hoorbaar omdat het molensteen-geluid vrijwel geen herkenningspunten biedt.
 - ii. De seriële interface van PD 0.38 blijkt niet te werken. De file 'comport.dll' moet met een andere overschreven worden. Binary install voor PD 0.38 + deze file zijn in de documentatie aanwezig.
 - iii. Het molensteen-geluid maakt soms klikjes bij snel veranderen van de weergavesnelheid. Op een tragere PC (1.4GHz laptop) wordt dit erger. Waarschijnlijk heeft dit proces de laagste prioriteit in PD, omdat het helemaal links staat. Dit moet verder onderzocht worden.

presentatie Montevideo, 8 maart 2007

4. Presentatie Montevideo, 8 maart 2007
Vrijdag 8 maart wordt het onderzoek bij Montevideo gepresenteerd aan een aantal geïnteresseerden.. Behalve een verslag van het onderzoek is er een proef installatie waarbij origineel en emulatie naast elkaar staan opgesteld zodat ze vergeleken kunnen worden.
Bij het koppelen van de beide opnemers blijkt dat de analyse in een groot

aantal opzichten correct was: beelden verspringen in beide installaties even snel, de toonhoogte van het geluid volgt op dezelfde manier de bewegingen van het wielkje. Maar ook blijken er een paar verschillen tussen de emulatie en het origineel:

- a. De tijd dat het beeld zwart blijft na een wisseling van bron is te lang ingeschat. De emulatie reageert daardoor te traag op richtingswisselingen.
- b. De laserdisc krijgt pas weer opdrachten naar een volgend frame te gaan nadat hij gemeld heeft dat frame 1 klaar staat. Dat betekent dat de beeld pointer pas na het verstrijken van de zwart tijd verhoogd mag worden; de emulatie slaat vaak het eerste beeld over.

Ook worden er die dag nieuwe samples van het geluid gemaakt om het probleem met de loop lengte aan te pakken. Bij nadere analyse blijkt dat de weergavefrequentie van het kastje niet helemaal stabiel is. Dit is de oorzaak van verschil in lengte tussen de beide audioloops.

aanpassingen n.a.v. de vergelijking emulatie-origineel

5. aanpassingen

- a. de zwart tijd wordt teruggebracht naar 300 mS (ca. 8 video frames)
- b. de originele software bleek pas na wisseling van beeldreeks (zwart) weer gevoelig voor inkomende pulsen. In de emulatie werden de pulsen ook in dit interval geïnterpreteerd en werd het beeld aan deze telling aangepast. Het programma wordt zo veranderd dat na afloop van een zwart moment de frame pointers van de beide beeldreeksen op 1 gezet worden.

tweede (uiteindelijke) vergelijking

6. controle PD emulatie, 28 maart 2007

In het Montevideo lab worden nogmaals origineel en emulatie naast elkaar opgebouwd. Nu blijkt dat bij gekoppelde opnemers de installaties zich vrijwel synchroon gedragen.

- a. Er is een fase verschil bij de beeldweergave, het beeld van het origineel komt ca 2 pulsen eerder dan dat van de emulatie.
- b. Ook komt aan het licht dat de gebruikte PC (Dell4600, P4/2.8GHz) in PD een wat grotere audio buffer moet krijgen, nl 100mS in plaats van de default 70mS. Zonder deze aanpassing kraakt soms de geluidswaergave. Deze processorsnelheid is het minimum voor deze installatie.
- c. Bij zeer snel draaien van de opnemers blijkt dat emulatie en origineel niet meer parallel lopen, maar dit probleem treedt pas op bij snelheden die bij normaal gebruik van de installatie nooit zullen voorkomen.

7. opmerkingen

- a. PD serial: comport.dll van PD0.38 moet na installatie van PD vervangen worden door bijgaande comport.dll

- b. De draairichting van de rotatieopnemer wordt tbv de koppeling met het origineel nog verkeerd om geïnterpreteerd.

vervolg

8. vragen:

- a. Is de GEM implementatie optimaal in termen van resource gebruik
- b. Full screen video, op PAL formaat in goede kwaliteit vanuit Pure Data, hoe moet dat
- c. Is Quicktime (het formaat van de gebruikte filmpjes) een eigen formaat van Apple, zijn er open source interpreters voor (in MSwindows wordt de gratis Apple player gebruikt).
- d. Is er een oplossing voor de audio klikjes
- e. Kan er een live CD gemaakt worden met de PD patched, video en audio materiaal, bv gebaseerd op pure:dyme

bijlagen

9. documentatie

- a. Arduino code
- b. Arduino schema
- c. Arduino IDE install versie 007 *)
- d. Kübler opnemer documentatie
- e. PD patch
- f. PD install binary versie 0.38 *)
- g. Correcte comport.dll + readme voor installatie *)
- h. PD code geprint
- i. Linux live image + pd + code (todo!)

*) zie CDROM

a. arduino code

```

--- start code 'revolution.pde'

// paul klomp
// january 2007
// arduino rotary encoder interface

// sends MIDI commands over serial (9600Bd)

// send aftertouch for ever pulse received, channel
pressure 0 for fwd, 1 for rev tick
// pitch control channel 0 every 160 milliseconds, pulse
count as 14bit data, low 7 bits first

// input pins for rotary encoder signals
#define pulseA 2
#define pulseB 3

// midi command definitions
#define MIDI_pitch 0xe0 // pitch channel 0
#define MIDI_aftertouch 0xd0 // aftertouch channel 0

//#define MIDI

long time, p_time;
int count = 0;
int pulsecount = 0;
int BState;
int fwd = 0;
int rev = 0;

void setup()
{
  pinMode(pulseA, INPUT);
  pinMode(pulseB, INPUT);
  attachInterrupt(0, counter, RISING);
#ifdef MIDI
  Serial.begin(31250); // for real MIDI
#else
  Serial.begin( 38400); // for MIDI over serial
#endif
}

void loop()
{
  time = millis();
  if (time - p_time >= 160)
  {

```

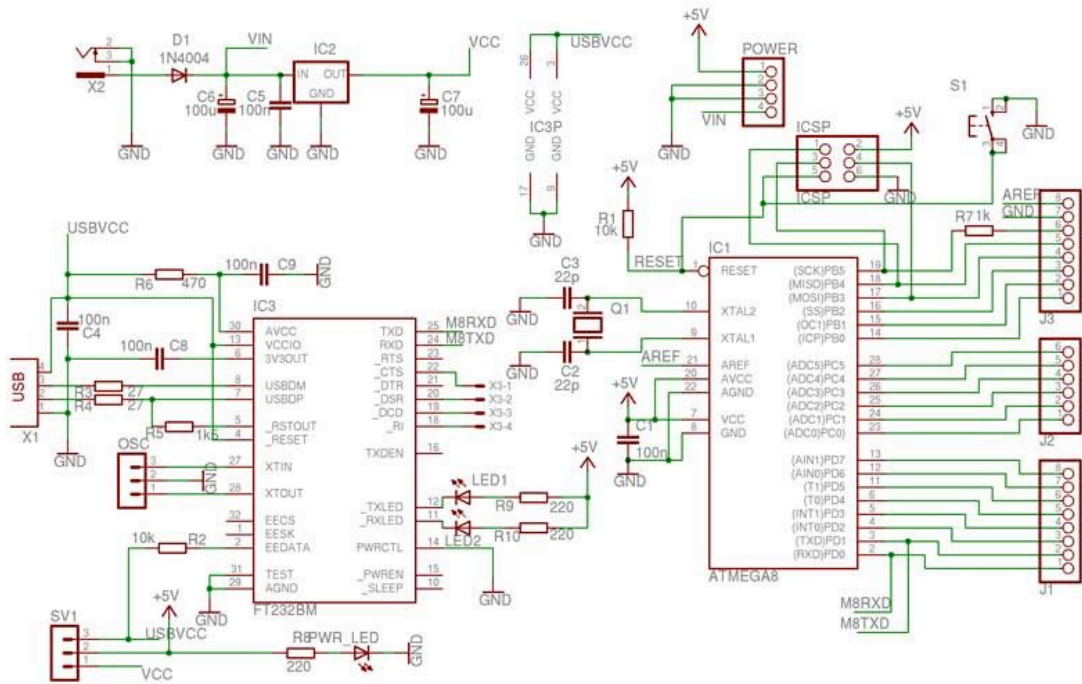
emulatie – eerste opzet

```
p_time = time;
Serial.print( MIDI_pitch, BYTE);
Serial.print( count & 0x7f, BYTE);
Serial.print( (count >> 7) & 0x7f, BYTE);
count = 0;
}
if (fwd > 0)
{
  fwd--;
  Serial.print( MIDI_aftertouch, BYTE);
  Serial.print( 0, BYTE);
}
if (rev > 0)
{
  rev--;
  Serial.print( MIDI_aftertouch, BYTE);
  Serial.print( 1, BYTE);
}
}

void counter()
{
  BState = digitalRead( pulseB);
  pulsecount++;
  if (pulsecount & 0x01) // every second count, we need
180 ticks per revolution
  {
    count++;
    if (BState == HIGH) fwd++;
    else rev++;
  }
}

--- end code 'revolution.pde'
```


b. arduino schema




c. Kübler opnemer documentatie

New

Rotary Measuring Technology

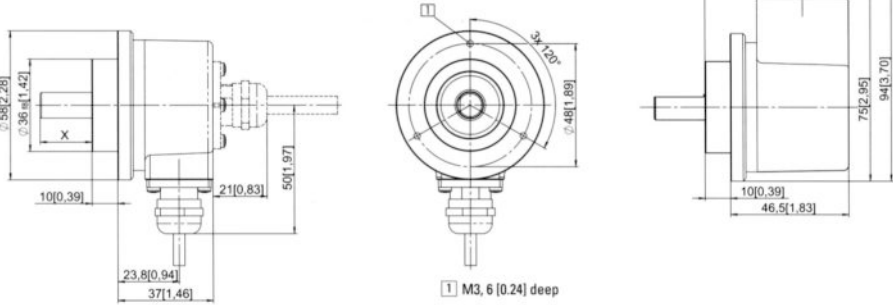
Incremental shaft encoder



Universal, compact Type 5000

Clamping bracket

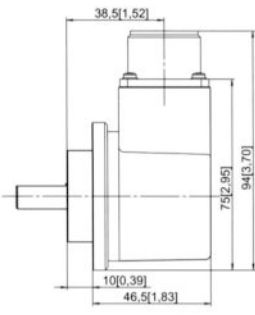
ø 58 mm
M12, M23 connector and cable versions
(Bracket type 7 and 8)



1) M3, 6 [0.24] deep

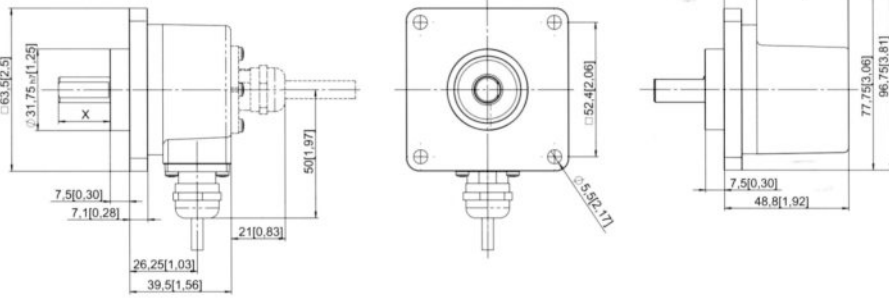
Clamping bracket

ø 58 mm
MIL-connector version



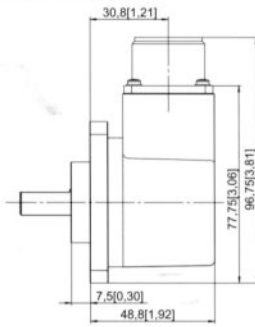
Rectangular bracket

□ 63.5 mm [2.5 inch]
M12, M23 connector and cable versions
(Bracket type C and D)



Rectangular bracket

□ 63.5 mm [2.5 inch]
MIL-connector version



Shaft versions

Order code for shaft	Shaft	length X
1	ø 6 mm	10 mm
2	ø 1/4 "	7/8" mm
3	ø 10 mm	20 mm
4	ø 3/8 "	7/8" mm
5	ø 12 mm	20 mm
6	ø 8 mm	15 mm

70 www.kuebler.com

1/2006

Rotary Measuring Technology Incremental shaft encoder

Universal, compact Type 5000

Terminal assignment:

Signal:	0 V GND	+U _B	0 V Sens	+U _B Sens	A	\bar{A}	B	\bar{B}	Z	\bar{Z}	Shield
M23, 12 pin connector, Pin:	10	12	11	2	5	6	8	1	3	4	– ¹⁾
M12, 8 pin connector, Pin:	1	2			3	4	5	6	7	8	– ¹⁾
MIL (MS styled), 10 pin con. Pin:	F	D		E	A	G	B	H	C	I	J ¹⁾
MIL (MS styled), 7pin con. Pin:	F	D		E	A	–	B	–	C	–	G ¹⁾
Cable colour:	WH	BN	GY PK	RD BU	GN	YE	GY	PK	BU	RD	Shield

¹⁾ Shield is attached to connector housing

Insulate unused outputs before initial startup

Top view of mating side, male contact base:

Type	8 pin M12 connector	12 pin M23 connector	MIL connector 10 pin
View			
Order code:	8.5000.XXX3.XXXX 8.5000.XXX4.XXXX	8.5000.XXX7.XXXX 8.5000.XXX8.XXXX	8.5000.XXXY.XXXX
Corresponding mating connector:	05.CMB-8181-0	8.0000.5012.0000	8.0000.5062.0000

Order code

8.5000.XXXX.XXXX

Range

Bracket

- 5 = Synchronous bracket, metric, ø 50,8, IP 67
- 6 = Synchronous bracket, metric, ø 50,8, IP 65
- 7 = Clamping bracket, metric, ø 58, IP 67
- 8 = Clamping bracket, metric, ø 58, IP 65**
- A = Synchronous bracket, ø 58, IP 67
- B = Synchronous bracket, ø 58, IP 65**
- C = Rectangular bracket 2.5", IP 67
- D = Rectangular bracket 2.5", IP 65

Shaft (ø x L)

- 1 = ø 6 mm x 10 mm**
- 2 = ø 1/4 inch x 7/8"
- 3 = ø 10 mm x 20 mm**
- 4 = ø 3/8 inch x 7/8"
- 5 = ø 12 mm x 20 mm
- 6 = ø 8 mm x 15 mm

Preferred types are indicated in bold

Pulse rate

(e.g.. 100 pulses => 0100)

Type of connection

- 1 = Cable axial (1 m PVC cable)
- 2 = Cable radial (1 m PVC cable)**
- 3 = Connector axial 8 pin M12
- 4 = Connector radial 8 pin M12**
- 7 = Connector axial 12 pin M23
- 8 = Connector radial 12 pin M23**
- W = Connector 7 pin MIL-specified
- Y = Connector 10 pin MIL-specified

Note: all connector versions without mating connector

Output circuit and supply voltage

- 1 = RS 422 (with inverted signal)
5 ... 30 V supply voltage
- 2 = Push-pull (7272 with inverted signal)
5 ... 30 V supply voltage
- 4 = RS 422 (with inverted signal)
5 V supply voltage**
- 5 = Push-pull (with inverted signal)
10 ... 30 V supply voltage

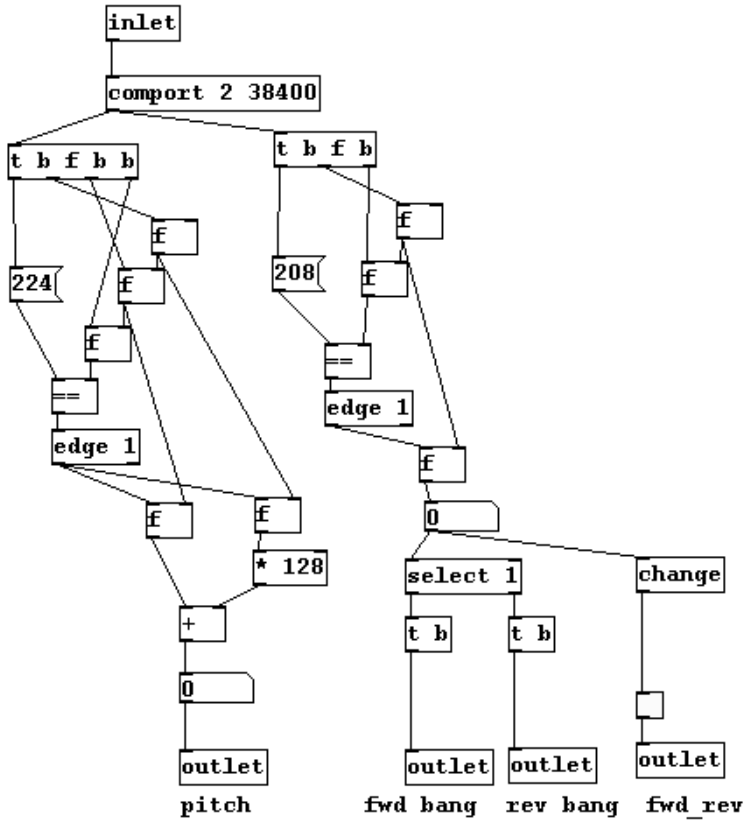
Accessories:

Cables and connectors see Connection Technology section in catalogue Sensor Technology

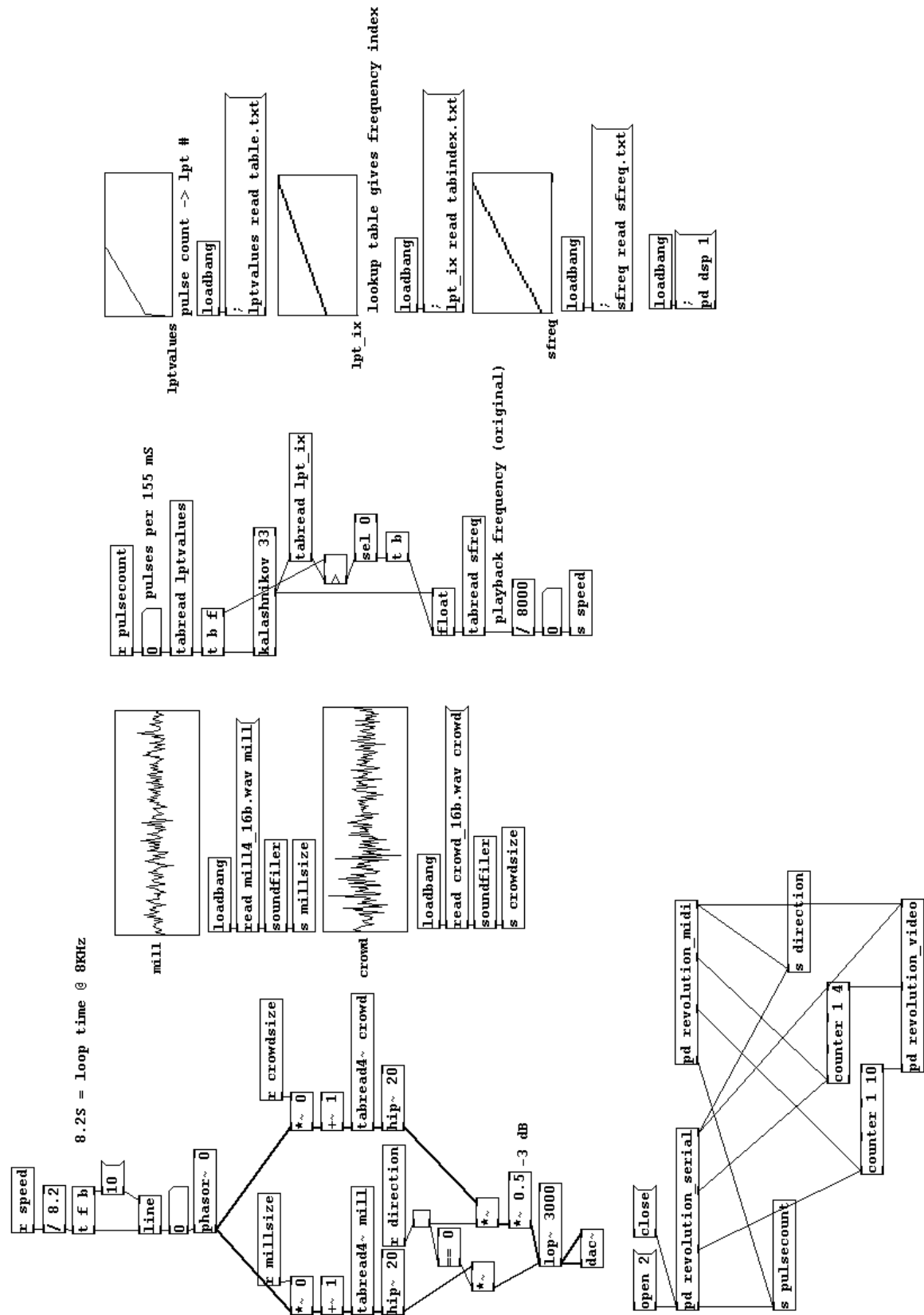
Corresponding mating connector:
M12: 05.CMB-8181-0
M23: 8.0000.5012.0000
MIL-connector
10-pin: 8.0000.5062.0000

revolution_serial.pd:

receive MIDI commands over serial line - 0xe0 for pitch msg
 data: number of ticks during last 160 ms is sent to 'pitch'
 - 0xd0 for after pressure data value "1" bangs fwd data
 value "0" bangs reverse - actual direction in fwd_rev



revolution_mov_mv.pd



video.pd

