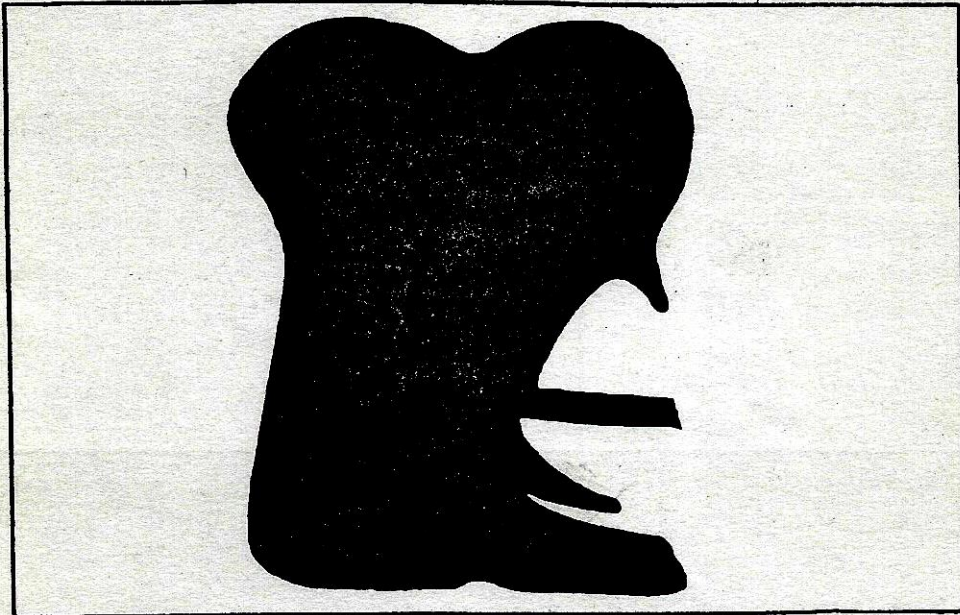


PRIJS F.2.50

Jaargang: **E E E E**

Nummer: **E E E**



# Acorntjesbrood

Redactie-adres:  
!!--> Calelaan 11-312 <--!!  
7522 MH Enschede

ACORNTJESBROOD IS HET ORGAAN VAN DE  
ATOM COMPUTER CLUB

## TWENTIE

NAMENS: de redactie

Rob Stekelenburg

Beste atomisten,

Deze keer vindt u het portret van een van uw redacteuren afgebeeld. Het is geproduceerd tijdens de voorlaatste bijeenkomst. Een samenwerkingsverband tussen een TV-camera van Maarten van Alphen, een videodigitizer van Gerrit



Hillebrand en een printer van onbekende oorsprong. Wie het spektakel gemist heeft mag de volgende clubavond niet overslaan.

In dit nummer vooral veel voer voor de hardware liefhebbers. Een 'hard' broodje dus, maar daarom niet minder lekker! Ik heb zo'n idee dat de landelijke HARDWARE-commissie tureluurs zou worden van de hoeveelheid materiaal dat binnen onze club bedacht en ook gemaakt wordt. Iemand die bijvoorbeeld om een 80-koloms kaart verlegen zit kan beter het gepubliceerde schema maken, dan wachten op de gebeurtenissen in den lande (trap ik nu tegen zere benen??....).

We hebben van dit nummer, met de bijdragen van anderen weer iets interessants proberen te maken. Een ding moet mij nog van het hart: ik denk dat een (te) groot deel van onze club bestaat uit consumenten en passieven. Bedenk dat we dit alles doen als hobby. Dus niet alleen commentaar leveren als iets niet geheel naar wens gaat, maar laat ook eens iets horen als iets goed is bevallen. Ik ben ervan overtuigd dat bijvoorbeeld de nieuwe leden genoeg vragen hebben. Laat wat van je horen!!!! Het moet toch mogelijk zijn om een 'lezers schrijven' rubriek in stand te houden zoals een respectabel maandblad betaamd.

As far as the Atom is concerned, the 'sealed box' is dead anyway. Five years after its inception, the Atom has been superseded twice by its inventors alone. But ... the big BUT! this does not mean that it no longer functions. It just means that to perform the more advanced and complex operations ingenuity will have to substitute for the cheque book.

The following ...

at.  
that  
way  
6502  
the p  
is t'  
pe  
m.  
RC

ACORN USER JANUARY 1986

## ALGEMENE INFORMATIE:

Bestuur:  
Voorzitter: K.v.d.Velde  
Secretaris: R.Overeem  
Penningmeester: D.t.Harmsel

Redactie:  
J.Biel  
G.Hillebrand  
R.Stekelenburg

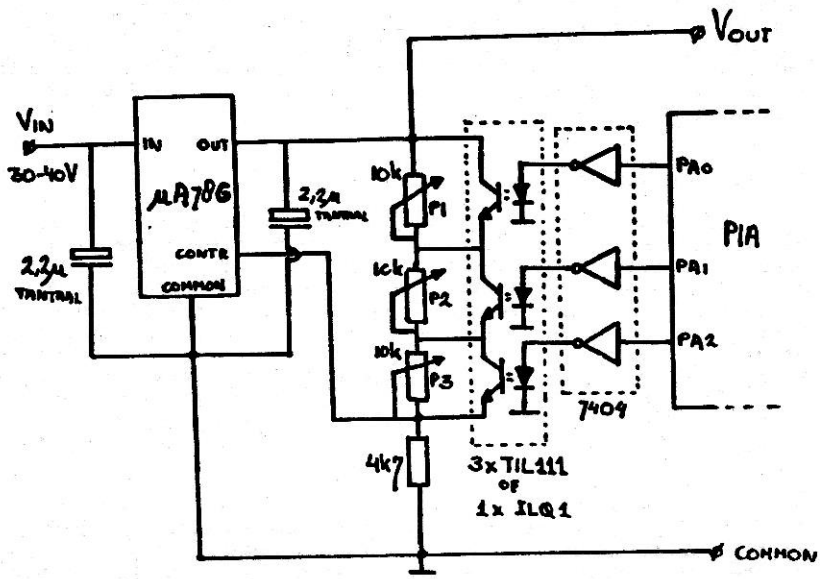
Programma-archief: P.Huisken  
Drukwerkarchief: R.Spel  
Epromprogr.dienst: R.Boers

Printendienst: D.t.Harmsel  
Ledenadministratie: D.t.Harmsel

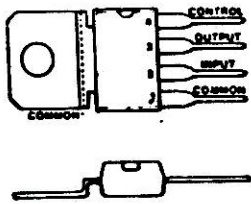
## Ledenlijst Atom Club Twente: Mei 1986

Naam:	Straat:	Plaats:	Telefoon:
Adena F.	Wierdensestraat 14	7443 AC Nijverdal	05486-16225
Alphen H.P. van	Korianderhof 40	7641 XK Wierden	05496-2264
Avest A.J. ter	Vondelstraat 47	7442 WN Nijverdal	05486-16565
Bakker J.H.	Fazantstraat 52	7481 BL Haaksbergen	05427-12913
Berendsen H.H.J.	Larenseweg 19	7475 PV Markelo	05476-1290
Bernink R.	In Den Vijfhoek 58	7571 DZ Oldenzaal	05410-20749
Biel J.	Calslaan 11-312	7522 MH Enschede	053-895018
Biezenbos F.J. van den	Burcht 75	7608 JC Almelo	05490-63539
Boers R.	Calslaan 7-210	7522 MH Enschede	D 053-895011
Boom H.H.	Maldengenstraat 9	7641 DW Wierden	05496-5187
Borgerink J.A.L.	Geert Grootestr. 9	7665 AV Albergen	05494-1612
Brand J.G.M.	Valeriusstraat 177	7604 CH Almelo	05490-25030
Brink R.J. van	Grotestraat 156	7443 BP Nijverdal	
Bruggen T. van	Spartastraat 67	7535 BL Enschede	
Buning W.H.G.	Kieskamp 1	7781 EB Gramsbergen	
Busschers G.H.v.	Krabbenbosweg 45	7555 EC Hengel	074-439048
Collet C.	Prins Bernardlaan 8	7622 BG Borne	074-661989
Coupre J.P.	Braakmansdijk 52	7462 LW Rijssen	05480-19477
Dam H. ten	Josef Israelstr. 13	7606 HM Almelo	05490-24703
Dijk T.E. van	Ferdinand Bolstr. 22	7545 CH Enschede	
Dijkstra J.K.	Julianastraat 13	7586 AT Overdinkel	05423-83698
Drijver G. den	Kotkampweg 188	7531 JK Enschede	053-354225
Droog E.	Hanenberglinden 111	7542 ET Enschede	053-769228
Eerden H. van	Hofkampstraat 42	7607 NJ Almelo	05490-24168
Egberink P.	Anjerstraat 52	7572 VR Oldenzaal	05410-20919
Elhorst W.	Leerlooierstraat 9	7447 XZ Hellendoorn	05486-54916
Guley F.	Boekeloseweg 134	7553 DR Hengelo	074-436452
Harmsel D. ter	R.v. Schevenstr. 37	7521 SC Enschede	053-359034
Hillebrand G.	Mendelssohnstr. 30	7557 BJ Hengelo	074-912931
Huiberts G.	Booggang 17	7552 JX Hengelo	074-433450
Huisken P.	Hogelandstraat 6	7573 CK Oldenzaal	05410-14713
Jansen B.	Geulstraat 56	7523 TV Enschede	053-355046
Jansenklomp P.	Drieturven 3	7552 TT Hengelo	074-427191
Jelsma S.	Lampertheimstraat 8	7641 DP Wierden	05490-67479
Jong B. de	Emmastraat 268a	7513 BJ Enschede	053-300463
Kamphuis B.	Rembrandtlaan 274	7545 ZS Enschede	053-322906
Klijnstra P.	Calslaan 7-304	7522 MH Enschede	053-895012
Krabbenbos J.	Oude Almeloseweg 123	7622 CD Borne	074-662939
Laarman J.	Potgieterstraat 13a	7442 XP Nijverdal	05486-17178
Leeuw H. de	Weleveldstraat 46	7557 JH Hengelo	074-421873
Mullink S.	Handellaan 20	7522 KN Enschede	053-350759
Oosterloo A.	Drentestraat 9	7543 DS Enschede	053-767099
Overeem R.	Molenstraat 108	7514 DW Enschede	
Passchier H.	Calslaan 46-31	7522 HG Enschede	D 074-434779
Poot G.B.	IJsselstraat 1	7555 AT Hengelo	
Pots J.B.A.	Van Hasseltstr. 18	7203 GG Zutphen	053-895005
Raven M.	Calslaan 3-202	7522 MH Enschede	05490-25151
Scherphof A.T.	Noorderstraat 57	7607 VT Almelo	05486-13129
Scholten J.A.	Rijnstraat 30	7442 ER Nijverdal	053-300728
Schuringa W.	Velweg 109	7533 XG Enschede	053-612772
Sessink W.	Janesrossstraat 35	7534 ZZ Glanerbrug	074-775963
Silvis J.M.	Else Mausstraat 62	7558 RB Hengelo	
Sinck A.	Grotestraat 66	7443 BK Nijverdal	
Slomp R.	Deurningerstraat 44	7514 BJ Enschede	
Snyder P.	Bakkersveenweg 17	7676 AM Westerhaar	074-437564
Spel R.	Oldenzaalsestr. 74II	7551 AL Hengelo	05410-11991
Stekelenburg R.	Pr. Hendrikstr. 122	7571 BX Oldenzaal	05490-24178
Stoop H.C.M.	De Grutto 27	7609 DC Almelo	05486-14234
Veen A. van de	Merelweg 26	7442 CB Nijverdal	05486-14234
Veenstra R.E.	Reviuslaan 57	8024 CD Zwolle	038-532984
Velde K. van der	Hengeveldebrink 108	7541 AX Enschede	053-761294
Veldhuis H.	Sperwerweg 29	7701 JV Dedemsvaart	05230-12605
Verhoeven W.L.A.	Witbreuksweg 377-403	7522 ZA Enschede	053-337026
Vink A.	Lankheethoek 13	7546 BP Enschede	053-775298
Wamelink H.J.	Pashegge 62	7103 BJ Winterswijk	05430-19601
Werken W.J.A.	Moellenbergstr. 47	7582 ZX Losser	
Wietmarschen H.L.M. van	Bachstraat 41	7651 MS Tubbergen	05493-2870
Wisse R.G.A.	--- Onbekend ---	---	---
Zieleman G.	Lochterseweg 41	7442 BN Nijverdal	05426-16992

**PROGRAMMEERBARE VOEDINGSSPANNING**

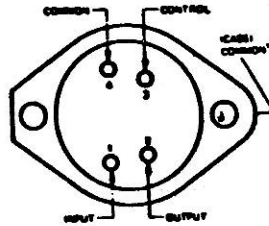


**μA78G  
POWER WATT PACKAGE  
CONNECTION DIAGRAMS**



ORDER INFORMATION	
TYPE	PART NO.
μA78GC	μA78GU1C

**TO-3 PACKAGE  
(TOP VIEW)**



ORDER INFORMATION	
TYPE	PART NO.
μA78G	μA78GKM
μA78GC	μA78GKC

ORS (registerselect) bepaalt, of het hoge, dan wel het lage register wordt aangesproken. Of het een lees- of schrijffactie betreft wordt bepaald door R/nw.

Verder moet uiteraard chipselect laag zijn als de 6502 access wil plegen.

Wie meer of alles wil weten over de werking van de 6845 kan natuurlijk de datasheet aanvragen bij de drukwerkarchivaris.

### 3: De karakter-eprom.

Hierin staat de geïnverteerde karakterset, een "0" komt dus overeen met een witte punt op het scherm. Als er een karakter tot videosignaal moet worden omgezet staat de code voor dit karakter (de ASCII-code zoals aanwezig in het videogeheugen) op de adreslijnen A4..A10 van de eprom. Alleen de laagste 7 bits van de waarde die in het videogeheugen staat worden gebruikt bij de adressering van de eprom. Het hoogste bitje dient voor het selecteren van de helderheid van het videosignaal.

De 6845 bepaalt de stand van de laagste vier adreslijnen van de eprom (het rij-adres) en selecteert daarmee, welke byte van de karaktermatrix gebruikt wordt. In principe zou de controller met deze 4 lijnen tot 16 kunnen tellen, maar de software zorgt ervoor dat hij maar tot 9 of 10 telt. De karaktermatrix beslaat zodoende 8 punten in de breedte bij 9 of 10 punten in de hoogte.

De hoogste twee adreslijnen van de eprom - A11 en A12 - worden ook door de 6845 aangestuurd, en wel met DE (display-enable) en Curs (cursor enable).

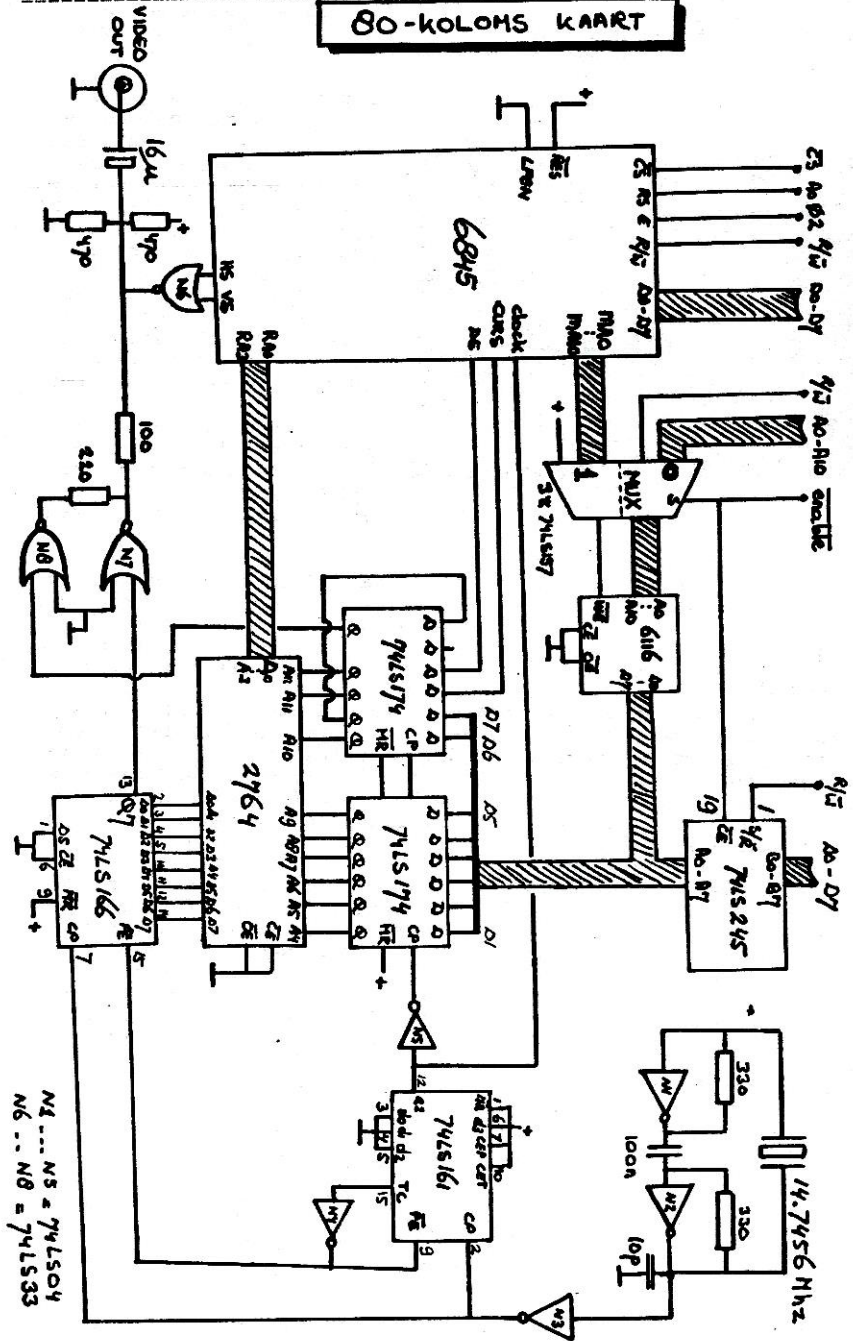
Met deze lijnen wordt de eprom in 4 blokken verdeeld:

DE	Curs	blok	adres in eprom
0	0	0	#0000 - #07FF
0	1	1	#0800 - #0FFF
1	0	2	#1000 - #17FF
1	1	3	#1800 - #1FFF

De blokken 0 en 1 zijn geheel leeg (#FF), omdat het niet de bedoeling is dat er iets buiten de "bladspiegel", d.w.z. wanneer display-enable laag is, gebeurt. Door het hoge bit in deze beide blokken "0" te maken zou het mogelijk zijn om een grijze border rond het display-venster te maken.

In het derde blok bevindt zich de data voor de karaktermatrixen. Per karakter zijn er 16 bytes gereserveerd (waarvan zoals gezegd er slechts 9 of 10 gebruikt worden). In het laatste blok staat de informatie voor de cursor. Uit eprom-programmeer-tijdwinst-oogpunt bekeken is het slim om alleen de bytes 8, 9 en 10 laag te maken, de rest hoeft dan niet door de programmer bewerkt te worden, vermits uw programmer zijn taak "intelligent" uitvoert. Op deze manier krijg je een underline-cursor.

# 80-KOLOMS KAART



=====  
 informatie, in de vorm van kleursignalen worden door de  
 digitizer in het geheel niet gebruikt.

-----  
 Het analoge gedeelte:  
 -----

Zoals nu bekend, het videosignaal bestaat uit synchronisatie- en beeldinformatie. Voordat we met het digitaliseren kunnen beginnen moeten we de 3 bestanddelen van een videobeeld van elkaar scheiden:

- verticale sync
- horizontale sync
- beeldinformatie

De twee torschakelingen in het analoge gedeelte scheiden de negatieve syncpuls van het beeldsignaal. Bij de schakeling voor het afscheiden van de verticale sync, wordt de frequentie met C5 zodanig verminderd, dat alleen het 50Hz signaal versterkt wordt. Na de schmitt-triggers staan de sync-puls ter beschikking aan de digitizer.

Omdat de spanningsnivo's van verschillende videosignalen niet duidelijk gedefinieerd zijn, is er gebruik gemaakt van een IC die zelf het zwartnivo van een videosignaal kan zoeken: de TBA 970. Dit IC vindt men veel in portable TV's en bevat een videoversterker met helderheidsregeling. De helderheid kan met P2 ingesteld worden. Het verschil tussen zwart en wit (contrast) kan met P1 ingesteld worden.

Het versterkte videosignaal wordt nu aangeboden aan een snelle 7bits A/D-converter opgebouwd uit de 4 LM319's. De referentie spanning van deze A/D-schakeling is vast ingesteld met het weerstandsnetwerk. Men kan nu met behulp van de helderheids- en contrast regeling het videosignaal in het werkgebied van de A/D converter brengen. Dit kan ook bewust foutief ingesteld worden om zodoende leuke effecten van het gedigitaliseerde videosignaal op de monitor te bereiken.

Het omklapnivo van de comperatoren is met behulp van een weerstandsnetwerk ingesteld. Het beeldsignaal is optimaal indien het zwartnivo net onder de omklapspanning van de onderste comperator ligt (ongeveer 1.8V) en het witnivo net over de omklapspanning van de bovenste comperator ligt (ongeveer 2.8V). Met een aangesloten hulpmonitor kan men het reeds gedigitaliseerde beeld bekijken en geeft, geloof me, prachtige effecten.

Na de comperatoren komt er een prioriteits-selector (IC14), die op zijn beurt een 3-bits helderheidsinformatie van een beeldpunt doorgeeft en gebruikt kan worden door de digitizer. Met de multiplexer (IC13) kunnen nu verschillende beeldsignaal-bronnen gekozen worden die elk direkt achter een comperator geschakeld is. Wil men het binairgecodeerde 3-bits woord lezen, dan moet men het beeld 3 keer digitaliseren en steeds een der 3 bits selekteren (grijsnivo 0, 1 en 2). Het gelijktijdig lezen van de 3-bits woorden is niet mogelijk. (wil men alle 3 bits informatie lezen en opslaan, dan zal de totale scan-tijd van een beeld nog maar ongeveer 0.8 seconde bedragen). Aan de ingang van het digitale gedeelte staan nu als ingangssignalen:

- digitale beeldinformatie
- horizontale sync
- verticale sync

-----  
 Het digitale gedeelte  
 -----

Nadat in het analoge gedeelte het beeldsignaal in

## De interface

Het computerblad MC had de totale interfacing met de computer verzorgt door normale TTL IC's voor het besturen van de verschillende functies in de digitizer en het uitdecoderen van alle besturingslogica. Ik heb dit alles vervangen door 1 IC: de 8255. Dit IC kost tegenwoordig nog maar een gulden of 7 en kan door middel van zijn 3 I/O poorten alle interfacing op zich nemen. Het uitdecoderen van de 8255 dient U nog wel zelf te doen. Let op, de 8255 is een positief actieve resetlijn nodig. Mogelijk is het om met een invertertje die U nog ergens hebt liggen, dit signaal te inverteren. De 8255 bevat 3 poorten. De functies voor de digitizer zijn als volgt:

- poort A beeldinformatie (16 bit data)
- poort B bit 0,1,2,3 : kolom nummer  
bit 4,5,6 : grijsnivo (mux select)
- poort C bit 0 : data available  
bit 1 : verticale sync ingang  
bit 4,5 : selectie beelddata woord

Ikzelf heb de 8255 uitgedecodeerd op BFD4. Voor iedere andere plaats dient de software aangepast te worden. Het resetten van de JK flip-flop, welke het 'data-available' signaal geeft indien de 16 bit informatie op poortA aangeboden wordt, gebeurt door een 74LS139 (IC21) die zodanig op de controllijnen is aangesloten dat deze een actief resetsignaal geeft bij het uitlezen van poort B (BFD5). Dit resetten dient zeer snel te gebeuren en is daarom 'hardware' uitgevoerd. Vergeet dus niet de A0, A1, NRDS en de CS aan deze 74LS139 aan te sluiten!

Let op: alle elco's in het schemadienantaal-C's te zijn.

Onderdelen:

R1 = 100K	R2,4,5 = 1K	R3,6,10,28 = 12K
R7 = 1K5	R8 = 27K	R9,12 = 2K7
R11 = 1K2	R13 = 750	R14,15,16,17,18,19 = 150
R20 = 1K8	R21,22,23,24,25,26,27 = 470	
R29 = 10K	R30 = 18K	
C1 = 4u7	C2,14,15 = 1n	C3,9 = 1u
C4,5 = 100n	C6 = 100p	C7,8,12,13 = 10u
C10 = 0.68u	C11 = 120p	
Z1 = 5,6 V		
T1 = BC309	T2,3,4 = BC238	
P1 = 5K	P2,4 = 10K	P3 = 50K
IC1 = 78L05	IC2 = 74LS629	IC3 = 74LS132
IC4 = 74LS395	IC5 = 74LS395	IC6 = 74LS395
IC7 = 74LS395	IC8 = 74LS76	IC9 = 74LS193
IC10 = 74LS193	IC11 = 74LS85	IC12 = 8255
IC13 = 74LS151	IC14 = 74LS148	IC15 = 74LS14
IC16 = LM319	IC17 = LM319	IC18 = LM319
IC19 = LM319	IC20 = TBA970	IC21 = 74LS139
IC22 = 74LS123		

Wat moet men nu allemaal afregelen ?

- P1 : contrast
- P2 : helderheid
- P3 : aftastfrequentie (beeld inkrimpen of uitrekken)
- P4 : verschuiven van het beeld naar rechts (nabootsen horz-sync)



printer en zal terzijner tijd voor deze digitizer aangepast moeten worden.

Voor de beeldbewerkingen is men wel GAGS en/of een RXBOX (of kloon) en/of GDOS1.5 nodig, let dus even op indien je een van de toolkits niet hebt.

Normale werkwijze:

-Regel contrast en helderheid met behulp van de hulpmonitor af

-Start het programma

-Regel de tastfrequentie EN de verschuiving naar rechts af

Dit alles net zo lang doen totdat het plaatje op het beeld redelijk (wat grootte betreft) overeen komt met het werkelijke beeld

Dit alles met SCAN (3 tot 7), 1 (normaal 1 bits Z/W beeld, klein plaatje).



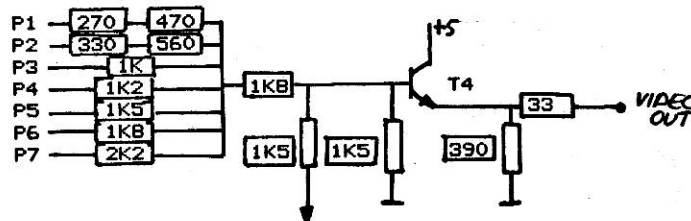
De totale kosten van de digitizer zijn ongeveer 70 gulden. Dit maakt zo'n ding erg aantrekkelijk voor avonden lang te computeren en tegelijkertijd TV kijken. De resultaten zijn erg goed te noemen, al moet ik zeggen dat met een camera betere resultaten te boeken zijn dan met b.v. een video-recorder, waarvan de sync-signalen toch al niet de steilsten ter wereld zijn. Een oplossing is b.v. een video-optimiser die de flauwe sync-signalen zeer sterke flanken geeft, toe te passen. Op dit moment is er in ieder geval geen gebrek meer aan plaatjes, zodat je altijd iets hebt om naar te kijken.

Niets op TV vanavond ?

Gerrit Hillebrand

## VIDEO-DIGITIZER

P.S. De onderdelen van de hulpmonitor trap zijn misschien niet overduidelijk, hier alsnog de trap in close-up:





```

530 JSR @C8BC      \ READ LARGE/SMALL IND.
540 JSR @C4E4
550 LDX @0
560 STX 4
570 LDA #16
580 ASLA
590 ASLA
600 ASLA
610 ASLA
620 STA PORTB      \ GREY VALUE SET
630 LDA @BUFFERH
640 STA MEMH+1
650 LDA @BUFFERL
660 STA MEML+1
670 LDY @0
680 STY MEMH
690 STY MEML
700 LDA @#10
710 STA PORTC
720 DEC PORTB
730:LOOP
740 INC PORTB      \ RESET HARDWARE & NEXT ROW
750:FRAME
760 LDA PORTC
770 AND @:0000 0010
780 BEQ FRAME      \ FRAME NOT YET OCCURED
790 LDX @#20        \ 40 LINES TO BE SKIPPED
800:SKIPREADY
810 LDA @1
820 BIT PORTC      \ DATA AVAILABLE ?
830 BEQ SKIPREADY  \ NO
840 LDA PORTB      \ RESET HARDWARE
850 DEX             \ YES, NEXT LINE
860 BNE SKIPREADY
870:LINELoop
880 ASL PORTC
890:WAITFORDATA
900 LDA @1
910 BIT PORTC      \ CHECK IF DATA AVAILABLE
920 BEQ WAITFORDATA \ NO
930 LDA PORTA      \ GET H.O VALUE
940 STA (MEMH),Y   \ STORE IN MEMORY
950 LSR PORTC
960 LDA PORTA      \ GET L.O. VALUE
970 STA (MEML),Y   \ STORE IN MEMORY
980 LDA PORTB      \ RESET HARDWARE
990 INY
1000 BNE LINELoop
1010 INC MEMH+1     \ NEW COLUMN, ADAPT POINTERS
1020 INC MEML+1
1030 BPL LOOP      \ ALL DONE ?
1040
1050 \ CONVERSION TO SCREEN
1060 LDA @BUFFERH
1070 STA MEMH+1
1080 LDA @BUFFERL
1090 STA MEML+1
1100 LDY @0
1110 STY GRAPH
1120 STY ROW

1720 PROC DRBORDER (W)
1730 FOR I=0 TO W-1
1740 CUBE 1,0,I,I,(255-2*I),(191-2*I)
1750 NEXT I
1760 PEND
1770
1780 PASS2;ASSEMBLE
1790 PASS1;ASSEMBLE
1800 ?#3FC=#40      ;REN #?#?#?#
1810
1820 INIT
1830 CLS
1840 P."VIDED DIGITIZER""
1850 P."ALL PICTURES WILL BE SAVED AS""
1860 P."-PICTURE#.PIC-, WHERE @=INTEGER""
1870 IN."ENTER BEGIN VALUE FOR # "T
1880 IN."SMALL OR LARGE PICTURES (1/0)"S
1890 IN."START VALUE OF GREY "A
1900 CLS
1910 P."OPTIONS: ""-B- BORDER (4)"
1920 P."-E- EXTENDED BORDER (8)"
1930 P."-U- SCROLL UP"
1940 P."-D- SCROLL DOWN"
1950 P."-L- SCROLL LEFT"
1960 P."-R- SCROLL RIGHT"
1970 P."-P- PRINTOUT (SMALL)"
1980 P."-O- PRINTOUT (LARGE)"
1990 P."-S- SAVE ON DISC"
2000 P."GREYVALUE CAN BE ALTERED DURING"
2010 P."SCANNING BY PRESSING 0-7"
2020 P."PRESS KEY TO CONTINUE"
2030 XIF $I=L-#9800
2040 ELSE L=#A000
2050 INKEY C
2060 $NONUM
2070 CLEAR4
2080 A=A+CH*0"
2090 WHILE 0=0
2100 SCAN (A-CH*0"),S
2110 PAUSE 30
2120 KEY C
2130 XIF C=CH"H"
2140 DO INKEY C
2150 CASE C OF
2160<CH"B"> DRBORDER (4)
2170 BORDER 0,0
2180<CH"E"> DRBORDER (8)
2190 BORDER 0,0
2200<CH"L"> SCROLL L;BLOCK 1,248,0,8,192
2210<CH"R"> SCROLL R;BLOCK 1,0,0,8,192
2220<CH"U"> SCROLL U;BLOCK 1,0,0,256,8
2230<CH"D"> SCROLL D;BLOCK 1,184,0,256,8
2240<CH"P"> SBUMP 1,10,5
2250<CH"B"> SBUMP 7,10,5
2260<CH"Q"> END
2270<CH"S"> @=0
2280 $OSCLI"SAVE",#34,"PICTURE",T,".PIC",#34,##0000," ",#L,
        #13
2290 T=T+1
2300 CEND

```

de string \$6\$12"ACORN ATOM"\$10\$13 afgedrukt en als laatste GDOS 1.5 aangelinkt.

De reset routine eindigt met een JMP #C2B2: zet de text-space pointer op #29 en voer een NEW uit.

Om deze 'korte' Reset-routine correct te laten verlopen is het noodzakelijk om te beginnen met een wachtlus (zie regel 300 t/m 370 in de hierna volgende listing van de RESET-MODIFIER). Dit om de 8255 even tijd te gunnen zich te stabiliseren na een reset. (zie ook AcornNieuws 2.1.43 e.v.: Bootstrap voor de Acorn Atom).

- Wordt behalve de Break- ook de CTRL-toets ingedrukt dan wordt naast het bovenstaande nog het volgende gedaan:
  - Het resetten van de vectoren op hun default waarden (behalve die voor DOS en eventueel VDU80 soft)
  - Het resetten van de Array adressen op #FF
  - COS op NOMON / LOCK-toets uit / cursor uit / ?#FE = 0

Wat toets-keuze betreft vertoont een aldus gewijzigde Atom duidelijk overeenkomst met de BBC en Electron. (It's all in the family).

De aldus ontstane nieuwe Reset-routine past precies op de plaats van de originele. Het lag ook in mijn bedoeling om ook nog SHIFT-BREAK in te bouwen, die dan \*!BOOT zou moeten uitvoeren, maar hier is dus extra ruimte in het F- of C-blok voor nodig. Die heb ik nog niet zodanig kunnen vinden zonder dat er rare c.q. ongedefinieerde dingen gebeuren.

Als we dan toch bezig zijn kunnen we in het C-blok gelijk de test op de aanwezigheid van de Floating-Point er uit slopen (wie heeft er nu nog steeds geen Dxxx). Dit kan door op de plaatsen #C550 t/m #C554 en #CA24 t/m #CA28 NOP-instructies te plaatsen

Hiervan hoeft U geen spectaculaire snelheidswinst in floating-point berekeningen te verwachten (aanroep en uitvoering van de testroutine is slechts 31 cycles: bij klokfrequentie van 1 MHz dus 31 us.), maar toch ....

Wilt U echt snelheidswinst behalen in het Dxxx blok dan kunt U beter de Autosync schakeling (op 1.8 MHz) van Peter Ehrlich uit Atom Nieuws 5.1.20 e.v. inbouwen (tevens ruisvrij plotten op 1.8 MHz).

Het bestuderen van een disas-lijst van het D-blok heeft mij namelijk geleerd dat het D-blok zo mogelijk een nog grotere berg spaghetti is dan het C-blok.

Het moge een ieder duidelijk zijn dat voor het implementeren van deze nieuwe Reset-routine de originele ROM (Cxxx + Fxxx) vervangen moet worden door een EPROM van het type 2764.

Zoals u misschien al reeds min of meer weet, heb ik destijds de ons zo vertrouwde word pack omgebouwd ten behoeve van het gebruik ervan op een 80-kolommenscherm. Ik had dat verwezenlijkt door de in deze toolkit ingebouwde read- en write-character routines te vervangen door overeenkomstige routines maar dan speciaal voor de toen actuele configuratie van de 80-kolommenkaart. Nu is het zo dat de hardware van mijn kaart in de loop der tijd een aantal keren is veranderd, zodat ook de word pack aangepast moest worden. Aangezien ik deze voornamelijk in een ram-uitvoering in de schakelkaart had zitten, was dat niet zo'n probleem. Veel ingrijpender werd het toen de systeemroutines voor de 80-kolommenkaart in belangrijke mate werden uitgebouwd. De behoefte deed zich toen al gauw voelen om een word pack te creëren die de bestaande vdu-routines kon gebruiken. Een faciliteit die de word pack bijvoorbeeld nog steeds moest missen, terwijl deze allang in de 80-kolommensoft was ingebouwd, was het zichtbaar maken van de cursorpositie rechtsboven in het scherm. In geval van een zgn. onafhankelijke word pack zou dit geen probleem geweest zijn, dus het werd tijd om in verband met mogelijke verdere ontwikkeling van de routines een dergelijke word pack te maken. En dat is dus inmiddels in beperkte mate verwezenlijkt, zodat deze nieuwe word pack in principe ook geschikt is voor andere 80-kolommenkaarten (clubkaart bijvoorbeeld).

Er zijn echter een aantal voorwaarden waaraan de read- en write-character routines moeten voldoen willen ze kunnen samenwerken met deze word pack.

Ten eerste mogen er natuurlijk geen geheugenconflicten optreden. De word pack in z'n oorspronkelijke vorm gaat vooral met de zero page uiterst ruig en onzorgvuldig om. Ik heb daar nauwelijks enige verbetering in kunnen aanbrengen. In ieder geval werken de 80-kolommenroutines zoals ze in het vorige Acorn-tjesbrood zijn beschreven samen met deze versie van de word pack.

Ten tweede wordt de schermgrootte intern bijgehouden. Deze moet voldoen aan het formaat van 25 regels van elk 80 karakters.

Ten derde zijn er een paar zaken, inherent aan deze benadering in het gebruik van de word pack, die op het eerste gezicht nogal vreemd overkomen. Als voorbeeld neem ik de cursortoetsen. Als deze normaal niet zijn voorzien van een auto repeat functie, dan zal dat in de word pack ook niet het geval zijn. Verder kun je het scherm laten scrollen door herhaald cursor down in te tikken. In de word pack zal de tekst (dus) ook scrollen, zodat het scherm geen goed beeld meer geeft van de werkelijke tekst. Druk in zo'n geval even op de escape-toets en het leed is alweer geleden.

Ten vierde moet gemeld worden dat de word pack weliswaar alle I/O-acties via de vectoren kan afhandelen, maar dat dat v.w.b. de actie van de copy-toets via een vervelende omweg moet geschieden. In de originele word pack wordt het indrukken van de copy-toets vertaald in het opleveren van het ascii-karakter EOT (#04). Op compatibiliteitsgronden kon ik daar geen andere code van maken, zodat ik gedwongen was om CTRL-D in te tikken i.p.v. COPY en in mijn routines moet dat zelfs een SHIFT-CTRL-D zijn, omdat de code

## =====

INFORMATION:  
 START: #2900, END: #3046, LENGTH: #0746 (= 1862)  
 #LINES: 72, LAST LINE: 720, #PAGES: 1  
 LONGEST LINE: 470 (= 58 CHAR'S)

```

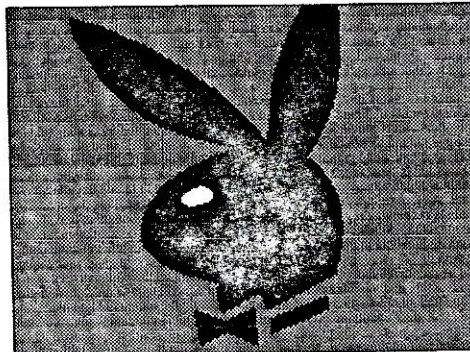
10 REN SQUEEZE.BAS
20 REN FOR GDOS 1.5
30 REN DOOR: G.B. FOOT
40 REN DATUM: 1986-02-23
50 A=#3400;REN START ADDRESS OF BUFFER
60 Z=#8000;REN END ADDRESS OF BUFFER
70 R=#2800
80 REN ALLOC. SPACE FOR FILE NAME:
90 DIM N(15)
100 INPUT "WHICH DRIVE?";R
110 $OSCLI"DIR",R,$13
120 F=?200F;REN NUMBER OF FILES ON DISK
130 IF F=0;P."DISK IS EMPTY";END
140 REN CHECK IF DISK HAS ALREADY BEEN SQUEEZED
150 I=0;E=3;H=0;REN I: FILE NUMBER, H: FLAG FOR HOLES
160 B=#2500+(F-1)*8;REN B: BASE ADDRESS FOR FILE I
170 S=B*64256+877;REN S: START SECTOR OF FILE I
180 L=B*5+(H*4<0);REN L: LENGTH OF FILE I IN BLOCKS
190 IF I=0;S=0;L=9;REN SPECIAL VALUES FOR 'FILE 0'
200 T=?*(B-8+6)*256+?(B-8*7)
210 REN T: START SECTOR OF NEXT FILE
220 X=T-(S+L);REN X: LENGTH OF HOLE
230 IF X=0;GOTO 270
240 P.X" SECTOR(S) UNUSED BETWEEN FILES'I' AND'I+1'
250 IF H=0;V=I+1
260 H=1
270 I=I+1
280 IF I<F;GOTO 160
290 IF H=0;P."SQUEEZED";END
300 INPUT "SHALL I CONTINUE?";R
310 IF ?R=CH"N";P."OK!";END
320 M=#FFFF;REN ADDRESS MASK
330 FOR I=V TO F
340 REN READ IN NAME OF FILE:
350 FOR J=0 TO 15
360 N?J=J?(#2000+16*(F+1-I))
370 NEXT J
380 Q=N*15
390 N?15=CH""
400 B=#2500+(F-1)*8
410 S=B*8*8
420 E=B*2*8
430 L=B*14*8
440 T=B*64256+877
450 $MON
460 $OSCLI"SET",Q,8*8*7F,$13
470 IF L>Z-A;P.QN," IS TOO LONG TO BE SQUEEZED";GOTO 660
480 $OSCLI"INFO",,,,,,$N,,,,,$13
490 $MON
500 $OSCLI"UNLOCK",,,,,,$N,,,,,$13
510 $OSCLI"LOAD",,,,,,$N,,,,,$A,$13
520 @=B
530 GOTO 610;REMOVE THIS LINE FOR MORE FILE INFO
540 P.QN:""
550 P."START" *S'
560 P."EXEC" *E'
570 P."END" *(S+L)'
580 P."LENGTH" *L'
590 P."TRACK" *(T/16)'
600 P."SECTOR" *(T/16)'
610 $OSCLI"SAVE",,,,,,$N,,,,,$A,*(A+L),BE,$13
620 IF Q=#80;$OSCLI"LOCK",,,,,,$N,,,,,$13
630 ?B-S*256;B?1=S/256
640 $USECT 0 1 B 2000
650 $USECT 0 1 B 2000
660 $MON
670 $OSCLI"INFO",,,,,,$N,,,,,$13
680 NEXT I
690 $OSCLI"SET",,$13
700 $KEEP
710
720 END
  
```



## ===== DISC-DISC COPY

INFORMATION:
START: #2900, END: #305E, LENGTH: #075E (= 1886)
#LINES: 72, LAST LINE: 720, #PAGES: 1
LONGEST LINE: 670 (= 86 CHAR'S)

10 PROGRAM DISC-DISC COPY
20
30 REM ONLY FOR DOUBLE DENSITY
40
50 PROC WISH
60 F=0;Z=0
70 PRINT#12"WHICH DRIVE CONTAINS THE SOURCE DISC [0/1/2/3] ?"
80 SIDE;S=K-48;@=2;P.S'
90 PRINT"- NUMBER OF TRACKS [40/80] ?"
100 TRACKS;V=(K-48)\*10;@=3;P.V''
110 PRINT"WHICH DRIVE CONTAINS THE TARGET DISC [0/1/2/3] ?"
120 SIDE;D=K-48;@=2;P.D'
130 PRINT"- NUMBER OF TRACKS [40/80] ?"
140 TRACKS;M=(K-48)\*10;@=3;P.M''
150 IF (S+@2) OR (S+@2 AND S=1) OR (S+@4 AND @=2) OR (S+@4)
;F=1
160 PEND
170
180 PROC SIDE
190 DO
200 INKEY K
210 UNTIL K&lt;52 AND K&gt;48
220 PEND
230
240 PROC TRACKS
250 DO
260 INKEY K
270 UNTIL K=52 OR K=56
280 PEND
290
300 PROC INIT
310 IF F;PRINT"INSERT TARGET DISC &amp; PRESS KEY";INKEY C
320 \*OSCLI "DIR",0,13
330 IF F;PRINT"INSERT SOURCE DISC &amp; PRESS KEY";INKEY C
340 \*OSCLI "DIR",S,13
350 M=0;N=0;T=1;U=1;X=0;Y=0
360 C=(V\*16)/88
370 E=?(#2506)\*256+?#2507
380 G=E+?#2505+1
390 B=(G/88)+1
400 IF B&gt;C;B=C
410 IF B&gt;640;Z=1
420 PEND
430
440 WISH
450 INIT
460 IF Z OR (V&lt;@ AND F);PRINT"UNABLE TO BACKUP !!!";END
470
480 @=0
490 PRINT"SECTORS TO COPY : #\*#G"
500 PRINT"- OUT OF : #\*#(V\*16)'"
510 IF F;PRINT"YOU MUST "B" TIMES CHANGE THE DISCS""
520
530 FOR I=1 TO B
540 \*OSCLI "DRIVE",S,13
550 \*OSCLI "#",V,13
560 XIF F;P.#11"INSERT SOURCE DISC &amp; PRESS KEY";INKEY C
570 ELSE XIF M&lt;0;?#BFC1=M;?#BFC2=T
580 ELSE
590 \*OSCLI "RSECT",M,132,&amp;T,132,&amp;#58,132,&amp;#2800,13
600 N=M+5;T=T+8;IF T&gt;16;T=T-16;N=N+1;\*OSCLI "RSECT",M,132,&amp;T,132,&amp;#1,132,&amp;#2700,13
610 \*OSCLI "DRIVE",D,13
620 \*OSCLI "#",M,13
630 XIF F;P.#11"INSERT TARGET DISC &amp; PRESS KEY";INKEY C
640 ELSE XIF M&lt;0;?#BFC1=M;?#BFC2=U
650 ELSE
660 \*OSCLI "WSECT",M,132,&amp;U,132,&amp;#58,132,&amp;#2800,13
670 N=N+5;U=U+8;IF U&gt;16;U=U-16;N=N+1;\*OSCLI "RSECT",M,132,&amp;U,132,&amp;#1,132,&amp;#2700,13
680 NEXT
690
700 PRINT"BACKUP PROCESS COMPLETED""
710 \*OSCLI "CAT",0,13
720 END



Herken  
U 't  
kereltje?

Tot slot nog twee voorbeelden om zelf uit te proberen:  
 Voorbeeld1 (voor de 'kop' van het appelmantje):  
 $%X=+1.764$  ;  $%U=+1.781$   
 $%Y=0$  ;  $%V=+0.013$   
 $E=200$

Voorbeeld2  
 $%X=+0.7445$  ;  $%U=+0.7459$   
 $%Y=+0.1121$  ;  $%V=+0.1132$   
 $E=1000$   
 (Duurt in BASIC vele dagen!!!!)

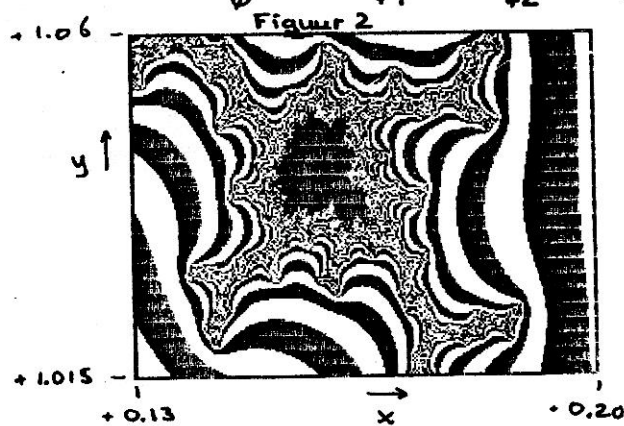
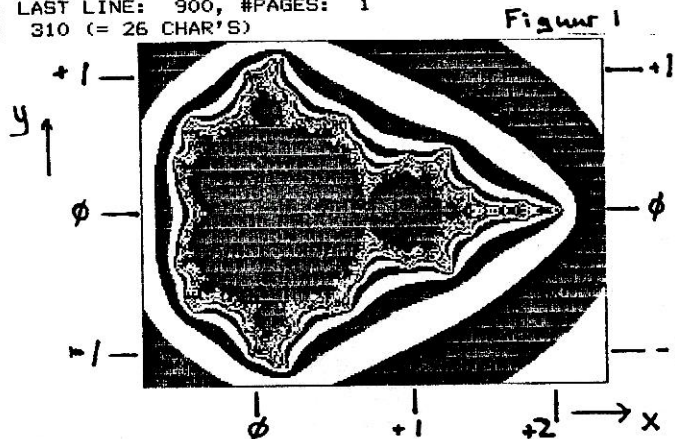
**VEEL PLEZIER EN GEDULD TOEGEWENST**

Bron: HC mein heim computer Nr. 4, April 1986

INFORMATION:  
 START: #2900, END: #2A6F, LENGTH: #016F (= 367)  
 #LINES: 28, LAST LINE: 900, #PAGES: 1  
 LONGEST LINE: 310 (= 26 CHAR'S)

```

10 CLEAR
20 X=255;Y=191
30 E=64
90 S=100
100 XI=.13;IU=.20
110 ZY=-1.06;ZV=-1.015
120 ZD=(ZU-ZX)/X
130 ZE=(ZV-ZY)/Y
140 FORM=0T0 Y
150 FORM=0T0X
160 ZF=ZX+X*ZD
170 ZG=ZY+Y*ZE
180 K=0
190 ZI=0
200 ZJ=0
210a K=K+1
220 ZN=XI*ZI;ZN=ZJ*ZJ
230 ZJ=2*ZJ*ZJ-ZG
240 ZI=ZN-ZN-ZF
250 IFK=E+1;G.b
260 FIF(ZN+ZN)<S;G.a
270 PLOT(K&1+14),N,N
280b NEXTN
290 NEXTM
300 P.#7;LINK#FFE3
310 *SAVE"FRACPIC"8000 9800
320 LINK#FFE3
900 END
  
```





```

410 GDS.200;GDS.120;GDS.190;GDS.120;GDS.180;GDS.120;GDS.190;GDS.120;GDS.210
420 GDS.400;GDS.230;GDS.120;GDS.210;GDS.120;GDS.200;GDS.120;GDS.210;GDS.120
430 GDS.250;PLAY3,4,3,0;GDS.400;GDS.260;GDS.120;GDS.250;GDS.120
440 GDS.240;GDS.120;GDS.250;GDS.120;F.C=1T02
450 PLAY2,3,6,L;PLAY3,4,3,0;GDS.320;GDS.120;GDS.300;GDS.120
460 PLAY2,4,1,L;PLAY3,4,3,L;GDS.290;GDS.120;GDS.300;GDS.120;N.;R.
470 PLAY3,2,6,L;GDS.160;PLAY3,3,8,L;GDS.160;PLAY3,3,3,L;GDS.160
480 PLAY3,3,6,L;GDS.150;PLAY3,3,6,0;GDS.150;PLAY3,3,6,L;R.
490 PLAY3,2,2,L;GDS.160;PLAY3,2,10,L;GDS.160;PLAY3,2,6,L;GDS.160
500 PLAY3,3,2,L;GDS.150;PLAY3,3,2,0;GDS.150;PLAY3,3,2,L;R.
510 PLAY3,2,3,L;GDS.160;PLAY3,2,11,L;GDS.160;PLAY3,2,7,L;GDS.160
520 PLAY3,3,3,L;GDS.150;PLAY3,2,3,0;GDS.150;PLAY3,3,3,L;R.
530 GDS.290;PLAY2,4,11,L;GDS.510;GDS.250;PLAY2,4,7,L;GDS.130;R.
540 PLAY2,4,10,L;GDS.120;PLAY2,4,2,L;GDS.120;PLAY2,4,10,L;GDS.120;R.
550 GDS.560;L=8;GDS.640;GDS.980;GDS.1160;GDS.1350;GDS.980;L=8;GDS.560;GDS.640;
GDS.980;G.1610
560F.F=1T02;IF F=2 L=5
570 GDS.410;GDS.330;GDS.400;GDS.300;GDS.130;GDS.330;GDS.130;F.B=1T03
580 PLAY3,4,7,0;PLAY2,3,3,L;GDS.280;GDS.150;GDS.300;GDS.150;GDS.320;GDS.120
590 PLAY3,4,7,L;PLAY2,5,10,L;GDS.300;GDS.130;PLAY2,5,3,L
600 GDS.280;GDS.130;IFB<>3 PLAY2,5,10,L;GDS.300;GDS.130;N.
610 PLAY2,5,9,L;GDS.270;GDS.130;PLAY3,4,7,0;PLAY2,4,3,L;GDS.250;GDS.140
620 PLAY2,4,3,0;N.;R.
630 GDS.300;PLAY2,4,6,L;GDS.470;R.
640F.G=1T02;F.E=1T02;IFE=2;L=6
650 GDS.250;PLAY2,5,1,L;GDS.130;GDS.260;PLAY2,5,2,L;GDS.130;PLAY3,3,1,L;F.D=1T
02
660 GDS.280;PLAY2,5,3,L;GDS.120;PLAY2,5,3,0;PLAY1,5,5,0;GDS.120;N.
670 GDS.300;PLAY3,3,3,L;GDS.120;GDS.280;GDS.120;GDS.260;GDS.120
680 GDS.250;GDS.120;GDS.230;PLAY2,4,7,L;PLAY3,3,5,L;GDS.130;IFE=2 G.700
690 PLAY2,4,5,L;GDS.130;PLAY3,3,5,0;N.
700 L=8;GDS.130;F.E=1T02;IFE=2 L=6
710 PLAY3,3,5,0;GDS.210;PLAY2,4,6,L;GDS.130;GDS.230;PLAY2,4,7,L;GDS.130
720 PLAY3,2,6,L;F.D=1T02;GDS.250;PLAY2,5,1,L;GDS.120;PLAY2,5,1,0
730 PLAY1,5,3,0;GDS.120;N.;PLAY3,3,1,L;GDS.260;GDS.120
740 GDS.250;GDS.120;GDS.230;GDS.120;GDS.210;GDS.120;GDS.200
750 PLAY2,4,11,L;PLAY3,3,3,L;GDS.130;IFE=2 G.770
760 PLAY2,3,3,L;GDS.130;PLAY3,3,3,0;N.
770 GDS.130;PLAY2,4,11,0;PLAY3,3,3,0;L=8;GDS.410;PLAY1,6,1,10;PLAY2,3,4,10
780 GDS.130;PLAY2,4,9,L;PLAY3,3,6,L;GDS.130;GDS.300;GDS.130;GDS.320
790 GDS.130;PLAY3,3,6,0;GDS.330;PLAY2,2,3,L;GDS.130;GDS.320
800 PLAY3,3,6,L;GDS.130;GDS.300;PLAY3,3,6,0;PLAY2,3,2,L;GDS.130
810 GDS.290;PLAY2,3,7,L;PLAY3,3,4,L;GDS.130;GDS.300;PLAY2,3,1,L
820 PLAY3,3,4,0;GDS.130;GDS.250;PLAY3,3,3,L;PLAY2,3,6,L;GDS.130
830 GDS.260;PLAY3,3,3,0;PLAY2,3,2,L;GDS.130;GDS.230;PLAY2,3,7,L
840;PLAY3,3,4,L;GDS.130;GDS.210;F.E=1T02;PLAY2,3,6,L;PLAY3,3,3,L
850 GDS.120;PLAY2,3,6,0;PLAY3,3,3,0;GDS.120;N.;PLAY3,3,3,L
860 PLAY2,3,11,L;GDS.200;GDS.160;GDS.210;GDS.160;GDS.200;GDS.160
870;PLAY3,3,3,0;PLAY2,3,11,0;GDS.210;GDS.160;GDS.200;GDS.160;GDS.210
880 GDS.160;PLAY3,3,3,L;PLAY2,3,11,L;GDS.200;GDS.160;GDS.210
890 GDS.160;GDS.200;GDS.160;GDS.190;GDS.150;GDS.200;GDS.150
900 GDS.190;PLAY2,3,6,L;PLAY3,2,6,L;GDS.140
910 PLAY3,2,6,0;N.G;R.
920 GDS.300;PLAY2,4,6,L;GDS.130;GDS.320;PLAY2,4,7,L
930 GDS.130;GDS.340;PLAY2,5,8,L;GDS.470;GDS.130;GDS.300
940 PLAY2,4,6,L;GDS.130;GDS.320;PLAY2,4,7,L;GDS.130;GDS.340
950 PLAY2,5,8,L;GDS.470;GDS.320;PLAY2,4,7,L;GDS.130;GDS.300
960 PLAY2,4,6,L;GDS.130;GDS.290;PLAY2,4,11,L;GDS.130
970 GDS.270;PLAY2,4,10,L;GDS.490;R.
980L=13;F.E=1T02;GDS.920;GDS.290;PLAY2,4,11,L;GDS.130;GDS.300

```

```

1570 PLAY2,5,10,L;GOS.350;GOS.490;GOS.120;PLAY3,3,2,0;GOS.120;F.E=1T02;PLAY3,3,
2,L;GOS.120;PLAY3,3,2,0;GOS.120;N.;R.
1580 GOS.1540;GOS.350;GOS.150;GOS.340;PLAY2,5,3,L;GOS.470;GOS.120;PLAY3,3,6,0;G
OS.120;F.E=1T02;PLAY3,3,6,L;GOS.120;PLAY3,3,6,0;GOS.120;N.
1590F.E=1T04;GOS.340;GOS.150;GOS.320;PLAY2,5,11,L;PLAY3,3,3,L;GOS.120;PLAY1,5,7
,0;PLAY2,5,11,0;PLAY3,3,3,0;GOS.150;N.;R.
1600 PLAY2,4,3,L;GOS.120;PLAY2,4,8,L;GOS.120;PLAY2,4,3,L;GOS.120;R.
1610L=13;GOS.340;PLAY2,5,8,L;PLAY3,3,6,L;GOS.120;PLAY3,3,6,0;GOS.120;PLAY1,6,8,
0;PLAY3,3,6,L;GOS.120;PLAY3,3,6,0;GOS.340;GOS.120;GOS.1540
1620F.E=1T04;GOS.350;GOS.150;GOS.340;PLAY2,5,3,L;PLAY3,3,6,L;GOS.120;PLAY1,6,8,
0;PLAY2,5,3,0;PLAY3,3,6,0;GOS.150;N.
1630 GOS.320;PLAY2,5,11,L;GOS.510;GOS.120;PLAY3,3,3,0;GOS.120;PLAY3,3,3,L;GOS.1
20;PLAY3,3,3,0;GOS.120;GOS.360;PLAY3,3,3,L;GOS.130;GOS.1580
1640L=10;GOS.300;PLAY3,3,3,0;PLAY2,3,6,L;GOS.120;PLAY2,4,3,L
1650 GOS.120;PLAY2,4,8,L;GOS.120;PLAY2,4,3,L;GOS.120;L=6;PLAY2,3,6,L
1660 GOS.250;GOS.150;GOS.300;GOS.150;GOS.340;PLAY2,4,3,L
1670 GOS.120;PLAY2,4,8,L;PLAY1,6,8,0;GOS.120;GOS.340
1680 PLAY2,4,3,L;GOS.120;F.E=1T02;PLAY2,3,6,L;GOS.250;GOS.150;GOS.300
1690 GOS.150;GOS.340;GOS.1600;PLAY2,3,6,L;GOS.120;GOS.1600
1700N.;F.E=1T02;GOS.350;PLAY2,3,6,L;GOS.120;GOS.340
1710 PLAY2,4,3,L;GOS.120;GOS.320;PLAY2,4,8,L;GOS.120;GOS.340
1720 PLAY2,4,3,L;GOS.120;N.
1730 PLAY2,3,6,L;GOS.350;GOS.120;GOS.540;PLAY2,3,6,L;GOS.120;GOS.540;GOS.150;F.
E=1T02;GOS.350
1740 PLAY2,3,6,L;GOS.150;GOS.340;GOS.150;PLAY2,4,3,L;GOS.120;GOS.350
1750 PLAY2,4,8,L;GOS.150;GOS.340;GOS.150;PLAY2,4,3,L;GOS.120;N.
1760 GOS.320;GOS.370;GOS.120;PLAY2,4,3,L;GOS.120;PLAY2,3,3,L;GOS.120;PLAY2,4,3,
L;GOS.120;GOS.360;PLAY2,3,7,L;GOS.120;PLAY2,4,3,L;GOS.120
1770L=13;GOS.1580;F.E=1T03
1780 GOS.380;GOS.340;GOS.130;GOS.380;GOS.270;PLAY2,5,3,L
1790 GOS.130;GOS.380;GOS.340;GOS.130
1800 N.;GOS.380;GOS.130;GOS.130;GOS.130
1810GENIE;END

```

---

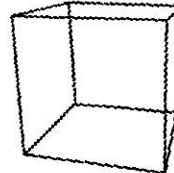
**GENIE-TURKA**


---

```

10 REM KALEIDOSKOOP
20 CLEAR 1;REM 64*64, COLOUR
25 X= 63;Y= 63
30 FOR W=3 TO 50
40 FOR I=1 TO X/2
50 FOR J=0 TO Y/2
60 K=I+J
70 COLOUR(J*3/I+W)
80 PLOT 13,I,K;PLOT 13,K,I;PLOT 13,(X-I),(Y-K)
90 PLOT 13,(X-K),(Y-I)
100 PLOT 13,K,(Y-I);PLOT 13,(X-I),K
110 PLOT 13,I,(Y-K);PLOT 13,(X-K),I
120 NEXT J
130 NEXT I
140 NEXT W
150 END

```

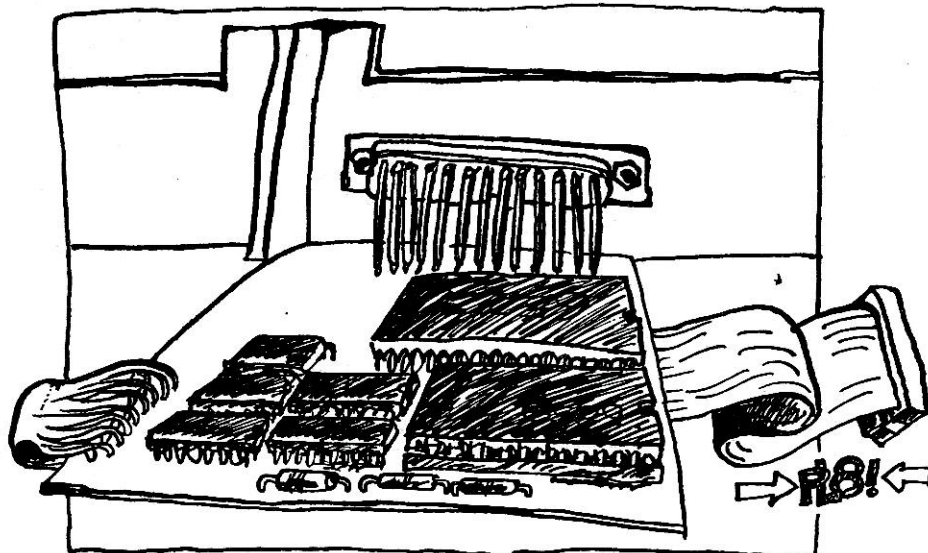


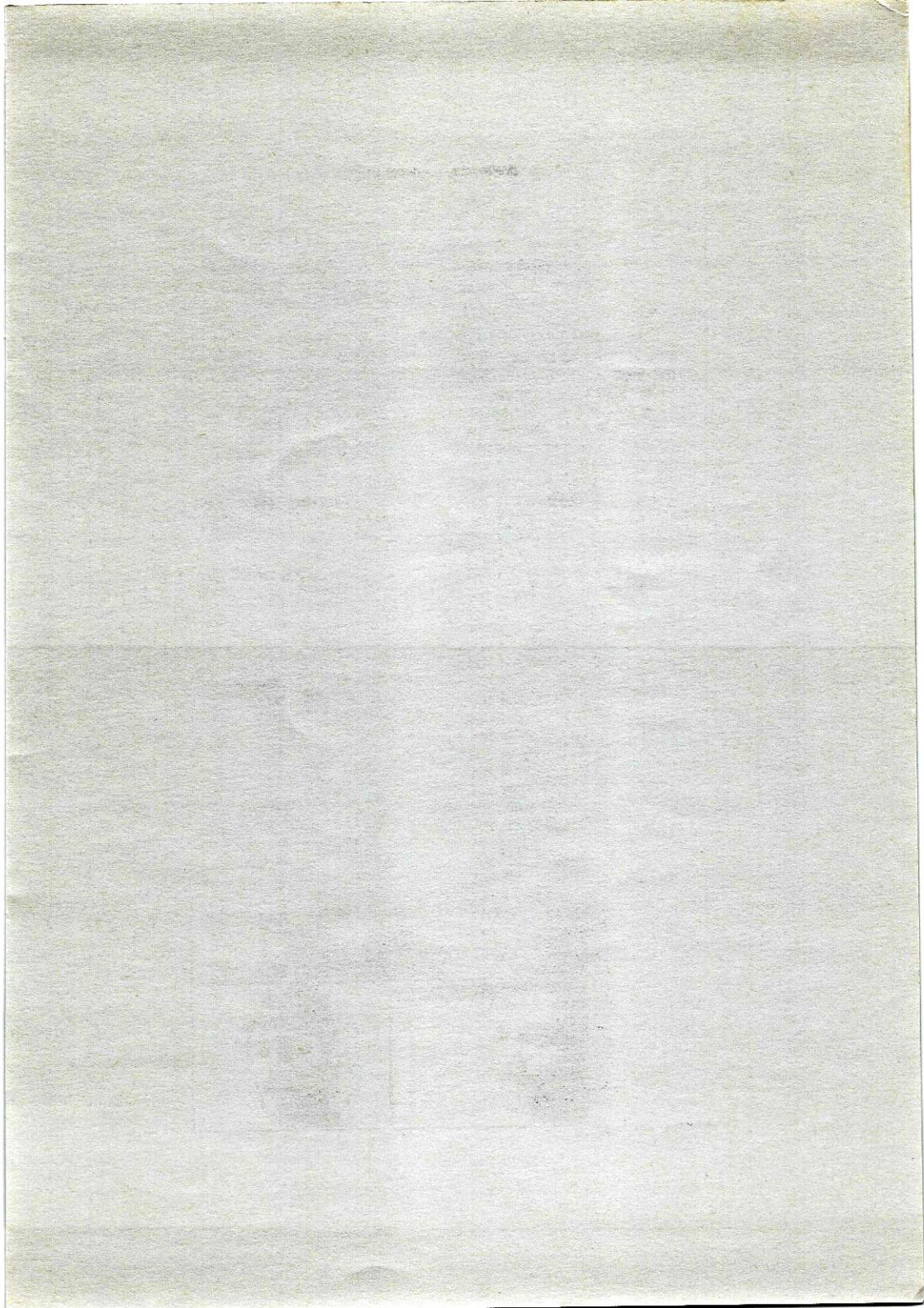
ACIA, volgens het volgende patroon:

B0	B1	B2	B3	B4	B5	ontvangen	zenden
1	0	0	x	x	x	300 Bd	
0	1	0	x	x	x	1200 Bd	
0	0	1	x	x	x	9600 Bd	
x	x	x	1	0	0		300 Bd
x	x	x	0	1	0		1200 Bd
x	x	x	0	0	1		9600 Bd

In mijn versie van het I/O-kaartje worden de overige poortlijnen + handshakelijnen van de VIA naar een 25-polige D-connector geleid. De aansluitingen zijn zo gekozen dat een bandkabelverbinding met de CP/M-kaart zonder gehutsel is te realiseren. Voor het handschudden met deze kaart zijn PB6\CA1 en PB7\CB1 doorverbonden.

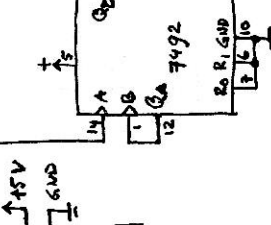
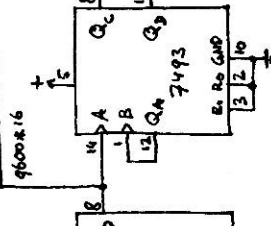
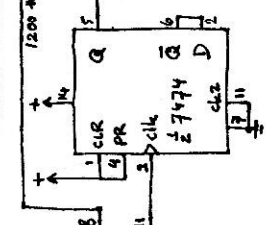
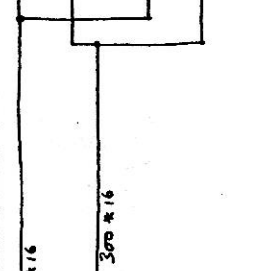
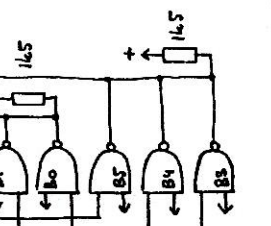
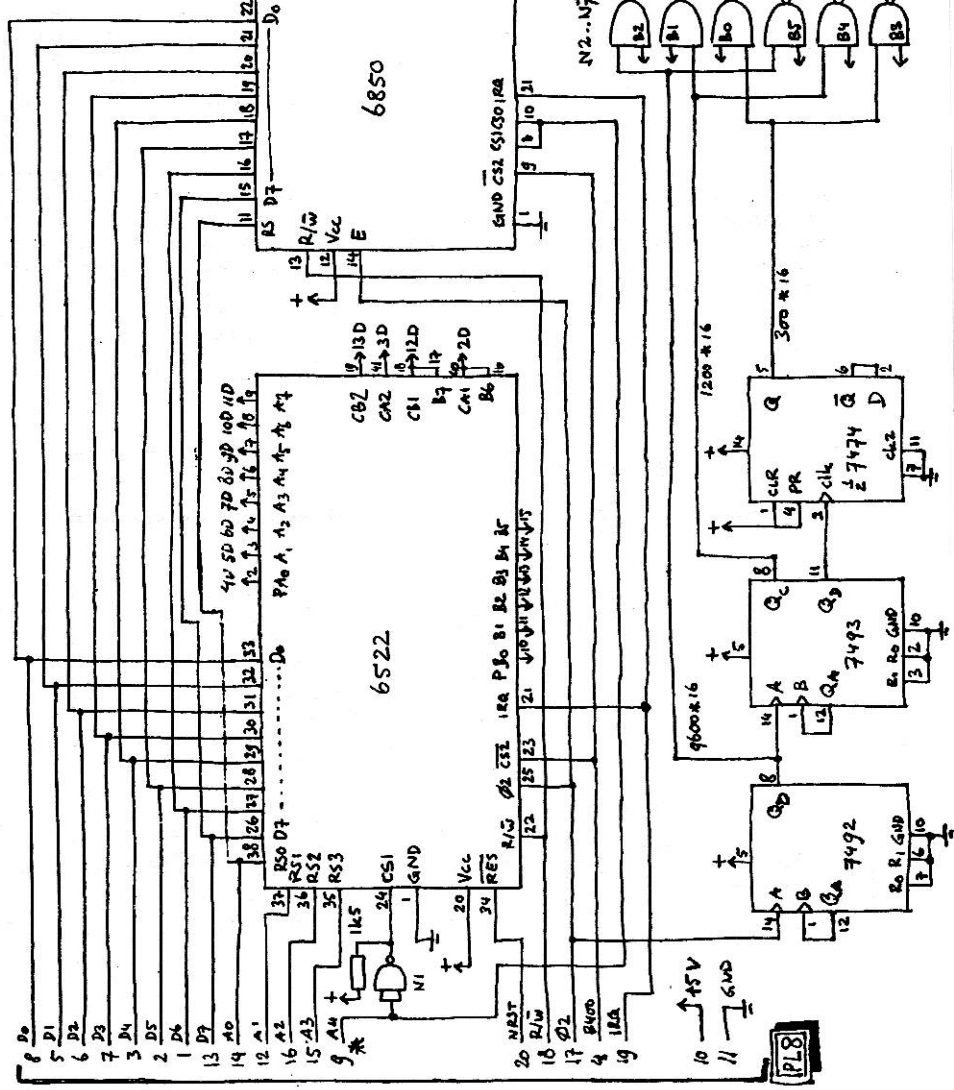
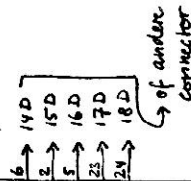
Voor voorlopig heb ik de zend- en ontvangstlijnen van de 6850 over dezelfde connector gevoerd; voor gelijktijdig gebruik van CP/M en een modem moet hiervoor een extra connector komen.







VIA en ACIA op BYXX  
 2D...25D  
 pen nummers D-connector  
 f<sub>2</sub> = 1.844 MHz  
 voor juiste bandbreedtes  
 \*PL8 peng = A4 i.p.v. NH1  
 NI...N7 = 7403



=====

HARDWARE: Extra VIA en ACIA op PL8

Paul Klomp

=====

Bij gebruik van de in de Atom ingebouwde VIA voor communicatie-doeleinden moet wanneer de van de handshakelijnen gebruik gemaakt wordt de printeraansluiting het ontgelden. Als we gebruik willen blijven maken van de printerpoort moet er een tweede VIA gemonteerd worden.

Voor mensen met een rack is dit geen probleem, ze solderen bijvoorbeeld de dubbele VIA-kaart in elkaar.

In het geval van mijn Atom ligt dat iets moeilijker: de interne busconnector is bezet door een RAM/80-kolomskaart, de connector achterop wil ik liever vrij houden voor b.v. een eprom-programmer of ander tijdelijk gebruik. De extra VIA moet er dus binnenin.

Nu is de Atom voorzien van een uitbreidingsconnector t.b.v. extra communicatie-hardware: PL8. Op deze connector zijn alle signalen voorhanden die nodig zijn voor het aansluiten van een tweede VIA!

Deze 20-pens aansluiting zit aan de soldeerzijde van de print en is alleen te bereiken als je de print geheel losschroeft.

Het is echter niet erg moeilijk om de oude connector er uit te solderen om er daarna aan de onderdelenkant van de print een haakse uitvoering niet in de plaats te stoppen. Dat uitsolderen is eenvoudig: haal de print helemaal uit de kast zodat hij aan beide zijden bereikbaar is en leg hem met de onderdelenzijde naar boven. Dan met een soldeerbout een voor een de pennen van de oude connector flink warm stoken en naar beneden drukken. Even een duwtje na met bijvoorbeeld de punt van een schaar en het pennetje zit helemaal los.

Nu toch de zaak open ligt kan er ook nog een kleine verandering aangebracht worden. Op PL8 is namelijk de NMI-lijn voorhanden die we normaliter niet zullen gebruiken. Door het printsnoertje naar pen 9 van PL8 door te snijden en in plaats van NMI adreslijn A4 door te voeren (bv. vanaf een van de nabijgelegen 2114-voeten) kan er op PL8 m.b.v. zeer eenvoudige decoding behalve de VIA ook nog een ACIA (6850) worden aangesloten, zodat communicatie niet alleen parallel maar ook serieel mogelijk is, bijvoorbeeld met een modem.

Verder moet LINK3 worden aangebracht zodat de processor-IRQ-lijn doorverbonden wordt met PL8.

Voor het berekenen van de baudrate-klok ben ik uitgegaan van een processorklok van 1.844 MHz. Deze klokfrequentie wordt aangemaakt op de 80-kolomskaart als je daarop een kristal van 14.7456 MHz gebruikt.

Als PHI-2 met de normale 1MHz bibbert is de delerschakeling zoals die in het bijgaande schema staat onbruikbaar. De beste benadering van baudratefrequenties wordt dan bereikt door het signaal op uitgang Qa van IC44-B door 13 te delen; hiervoor is al eerder een schakeling gepubliceerd in Acorn Nieuws 4-7. De mogelijkheid het gehele I/O kaartje met een 20-polige bandkabel op PL8 aan te sluiten vervalt dan omdat er een extra draadje naar de klokdeler moet komen.

De adresdecoding van de extra VIA loopt van #B400 tot #B40F, de ACIA wordt aangesproken op #B410 en #B411. Vanwege de onvolledige decoding manifesteren VIA en ACIA zich tussen #B400 en #B7FF, waartegen geen bezwaar is wanneer er in dit geheugengebied geen andere chips aangesloten zijn.

De poortuitgangen PB0..PB5 van de VIA worden gebruikt voor het instellen van de baudrates voor ontvangen en zenden van de

990 PLAY2,4,6,L;PLAY3,2,9,L;GOS.160;PLAY3,2,10,L;GOS.160;PLAY3,2,6,L  
 1000 GOS.160;PLAY3,3,9,L;GOS.150;PLAY3,3,9,0;GOS.150;PLAY3,3,9,L  
 1010 GOS.320;PLAY2,4,7,L;GOS.130;GOS.530;GOS.920;GOS.320  
 1020 PLAY2,4,7,L;GOS.130;GOS.530;GOS.300;PLAY2,4,6,L;PLAY3,2,6,L;GOS.140;PLAY  
 4,6,0;PLAY3,2,6,0;N.;R.  
 1030 GOS.340;GOS.120;GOS.350;GOS.120;GOS.340;GOS.120;GOS.320  
 1040 GOS.320;GOS.120;GOS.300;PLAY2,3,10,L;GOS.120;GOS.320  
 1050 GOS.120;GOS.300;PLAY2,4,8,L;PLAY3,3,10,L;GOS.120;GOS.290  
 1060 GOS.120;GOS.270;GOS.120;GOS.300;GOS.120;GOS.290;GOS.120  
 1070 GOS.270;GOS.120;PLAY2,3,11,L;PLAY3,3,10,0;R.  
 1080 GOS.1030;GOS.260;GOS.120;GOS.270;GOS.120;GOS.290;PLAY2,4,8,L  
 1090;PLAY3,3,7,L;GOS.120;GOS.260;GOS.120;GOS.220;GOS.120;GOS.240  
 1100 GOS.120;GOS.260;GOS.120;GOS.220;GOS.120;GOS.270;PLAY3,3,7,0  
 1110 PLAY2,3,10,L;GOS.120;GOS.260;GOS.120;GOS.270;PLAY2,4,8,L  
 1120 PLAY3,3,6,L;GOS.120;GOS.290;GOS.120;GOS.300;GOS.120;GOS.290  
 1130 GOS.120;GOS.300;GOS.120;GOS.320;GOS.120;GOS.340;PLAY3,3,6,0  
 1140 PLAY2,3,4,L;GOS.120;GOS.330;GOS.120;GOS.340;PLAY2,4,8,L  
 1150 PLAY3,3,11,L;GOS.120;GOS.330;GOS.120;R.  
 1160 PLAY3,2,6,0;PLAY2,4,6,0;F.E=1T02;L=8;IFE=2;L=5  
 1170 GOS.1080;GOS.1030;GOS.250;GOS.120;GOS.270;GOS.120;GOS.290  
 1180 PLAY2,4,3,L;PLAY3,4,8,L;GOS.120;GOS.250;GOS.120;GOS.220  
 1190 GOS.120;GOS.240;GOS.120;GOS.250;GOS.120;GOS.220;GOS.120  
 1200 PLAY2,3,11,L;PLAY3,4,8,0;GOS.240;GOS.120;GOS.250;GOS.120  
 1210 PLAY2,4,10,L;PLAY3,4,9,L;GOS.270;GOS.120;GOS.240;GOS.120  
 1220 GOS.210;GOS.120;GOS.220;GOS.120;GOS.240;GOS.120;GOS.210  
 1230 GOS.120;GOS.220;PLAY2,4,3,L;PLAY3,4,8,L;GOS.140  
 1240 PLAY2,4,3,0;PLAY3,4,8,0;N.;R.  
 1250 PLAY2,3,6,L;GOS.190;GOS.120;GOS.200;GOS.120;GOS.220  
 1260 PLAY2,4,3,L;PLAY3,4,8,L;GOS.120;GOS.230;GOS.120;GOS.250  
 1270 GOS.120;GOS.270;GOS.120;GOS.290;GOS.120;GOS.300;GOS.120  
 1280 PLAY3,4,8,0;R.  
 1290 GOS.250;GOS.120;GOS.230;GOS.120;GOS.220;GOS.120;GOS.200  
 1300 GOS.120;GOS.1250;PLAY2,3,7,L;GOS.300;GOS.120;GOS.290  
 1310 GOS.120;PLAY2,4,3,L;PLAY3,4,2,L;GOS.270;GOS.120;GOS.250  
 1320 GOS.120;PLAY3,4,2,0;PLAY2,3,11,L;GOS.250;GOS.120;GOS.230  
 1330 GOS.120;PLAY2,4,3,L;PLAY3,4,2,L;GOS.220;GOS.120;GOS.200  
 1340 GOS.120;R.  
 1350L=8;F.E=1T02;GOS.1290;GOS.1250;GOS.310;PLAY2,3,3,L  
 1360 GOS.130;GOS.320;PLAY2,4,2,L;PLAY3,3,11,L;GOS.130;GOS.1290;PLAY3,4,2,0  
 1370 PLAY2,3,6,L;GOS.220;GOS.120;GOS.250;GOS.120;PLAY2,3,10,L  
 1380 GOS.190;GOS.120;GOS.220;GOS.120;PLAY2,3,2,L;GOS.200  
 1390 GOS.120;GOS.230;GOS.120;PLAY2,3,3,L;GOS.180;GOS.120  
 1400 GOS.200;GOS.120;GOS.190;PLAY2,2,6,L;GOS.130;PLAY2,3,6,L  
 1410 GOS.130;PLAY2,3,6,0;GOS.1080;L=10;GOS.340;PLAY3,3,5,L  
 1420 GOS.120;GOS.330;GOS.120;GOS.340;PLAY3,3,10,L;GOS.120  
 1430 GOS.310;GOS.120;PLAY3,3,10,0;PLAY2,2,7,L;GOS.350;GOS.120  
 1440 GOS.340;GOS.120;PLAY3,3,10,L;F.6=1T03;GOS.350;GOS.120;GOS.340  
 1450 GOS.120;N.;L=8;PLAY3,3,10,0;GOS.350;GOS.120;GOS.340  
 1460 GOS.120;PLAY3,3,11,L;GOS.320;GOS.120;GOS.300;GOS.120  
 1470 GOS.290;GOS.120;GOS.300;GOS.120;GOS.320;GOS.120;GOS.290  
 1480 GOS.120;PLAY3,3,11,0;PLAY2,3,8,L;GOS.300;GOS.120;GOS.320  
 1490 GOS.120;PLAY2,3,6,L;PLAY3,3,10,L;GOS.340;GOS.120;GOS.300  
 1500 GOS.120;PLAY3,3,10,0;PLAY2,3,8,L;GOS.260;GOS.120;GOS.270  
 1510 GOS.120;PLAY2,3,7,L;PLAY3,3,11,L;GOS.290;GOS.120;GOS.260  
 1520 GOS.120;GOS.270;PLAY2,3,6,L;PLAY3,3,10,L;GOS.140;PLAY2,3,6,0  
 1530 PLAY3,3,10,0;N.;R.  
 1540F.E=1T02;GOS.340;PLAY2,5,3,L;GOS.470;GOS.120;PLAY3,3,6,0;GOS.120  
 1550 PLAY2,5,3,0;F.F=1T02;PLAY3,3,6,L;GOS.120;PLAY3,3,6,0;GOS.120;N.;N.  
 1560F.E=1T02;PLAY3,3,6,L;GOS.350;GOS.120;PLAY3,3,6,0;GOS.340;GOS.120;PLAY3,3,6  
 L;GOS.320;GOS.120;PLAY3,3,6,0;GOS.340;GOS.120;N.



SOFTWARE: Mozart op de Atom

Gerrit Hillebrand

Zoals in het vorige broodje beloofd, wederom een muziekstuk voor de Atom, gespeeld met behulp van de Genie-compatibele geluidsstaements: PLAY en GENIE, voor de AY-3-8910 geluidsgenerator.

Dit keer is het de beurt aan de 'Tuerkische Marsch', oftewel 'A La Turka' van de songwriter Wolfgang Amadeus Mozart. Ja, het is weer een klassiek nummer, maar misschien brengt U daar de volgende keer verandering in.

## GENIE-TURKA

## INFORMATION:

START: #2900, END: #4E76, LENGTH: #2576 (= 9590)  
 #LINES: 181, LAST LINE: 1810, #PAGES: 4  
 LONGEST LINE: 1630 (=137 CHAR'S)

```

10 PROGRAM GENIE-TURKA
20
30 REM DIT PROGRAMMA MAAKT GEBRUIK VAN P-CHARME
40 REM EN DE STATEMENTS PLAY EN GENIE T.B.V.
50 REM HET SIMULEREN VAN DE GELUIDSTATEMENTS VAN DE
60 REM COLOUR GENIE COMPUTER
70
80 GENIE
90 F.*12;?#E1=0;VTAB6;HTAB10;P."A LA TURKA"
100 VTAB7;HTAB5;P."WOLFGANG AMADEUS MOZART"
110 L=8;G.550
120 F.A=1T0250;N.;R.
130 F.A=1T0500;N.;R.
140 F.A=1T0600;N.;R.
150 F.A=1T0125;N.;R.
160 F.A=1T085;N.;R.
170 PLAY1,4,3,L;R.
180 PLAY1,4,11,L;R.
190 PLAY1,4,6,L;R.
200 PLAY1,4,7,L;R.
210 PLAY1,5,1,L;R.
220 PLAY1,5,8,L;R.
230 PLAY1,5,2,L;R.
240 PLAY1,5,9,L;R.
250 PLAY1,5,3,L;R.
260 PLAY1,5,4,L;R.
270 PLAY1,5,10,L;R.
280 PLAY1,5,5,L;R.
290 PLAY1,5,11,L;R.
300 PLAY1,5,6,L;R.
310 PLAY1,5,12,L;R.
320 PLAY1,5,7,L;R.
330 PLAY1,6,1,L;R.
340 PLAY1,6,8,L;R.
350 PLAY1,6,2,L;R.
360 PLAY1,6,3,L;R.
370 PLAY2,3,3,L;GOS.120;PLAY2,4,3,L;GOS.120;PLAY2,3,7,L;R.
380 PLAY2,5,8,L;GOS.300;GOS.470;GOS.120;PLAY3,3,6,0;GOS.120
390 PLAY3,3,6,L;GOS.120;PLAY3,3,6,0;GOS.120;PLAY3,3,6,L;R.
400 PLAY2,3,6,L;GOS.130;PLAY2,4,1,L;PLAY3,4,3,L;GOS.130;R.

```





Een leuke toepassing van gecompliceerde wiskunde, biedt het volgende programma. Ik zal niet ingaan op de theoretische achtergrond en volsta met te zeggen dat de werking onder andere gebaseerd is op het principe van iteratie.

Hoe werkt het en wat kunnen we ermee? Als het programma gerund wordt, ontstaat het zogenaamde "APPELMANNETJE", zie figuur 1. Het bevindt zich op een X-Y vlak. De zwarte en witte gedeeltes geven als het ware de hoogteverschillen in het plaatje aan.

Het aardige is nu, dat we in dit 'berg'landschap op ontdekkingsreis kunnen gaan! We kunnen elk deel van het plaatje tot het 1000-voudige of meer opblazen en in detail bekijken. Daarbij komen de meest bizarre en interessante vormen te voorschijn! Een uitvergroot deel is weergegeven in figuur 2. Om dit te verkrijgen moeten we in het programma de x en y grenzen aanpassen. Dit gebeurt in de regels 100 en 110, waarbij %X en %U respectievelijk de onder- en bovengrenzen in de x-richting voorstellen en %Y en %V idem voor de y-richting. In dit geval waren de waarden:

%X=+0.13 ; %U= +0.20

%Y= -1.06 ; %V= -1.015

Een belangrijk punt is het volgende: Hoe groter het detail des te meer rekenstappen er nodig zijn om een fatsoenlijk plaatje van het uitgekozen gebied te produceren. Het maximale aantal stappen wordt bepaald door de variable E in regel 30. Voor figuur 1 is een waarde van 50 al voldoende, voor figuur 2 is E=100 een betere keus. Echter, bij een groter aantal rekenstappen word ook de rekestijd langer en dus de tijd die nodig is om het plaatje op te bouwen. Voor figuur 1 is de ATOM al enkele uren (!!) bezig. Dat wordt in de meeste gevallen dus nachtwerk (voor de computer wel te verstaan). Diegenen die bijvoorbeeld PASCAL kunnen draaien, kunnen deze tijd aanzienlijk verkorten.

Voor gebruik met de kleurenkaart moeten de volgende veranderingen worden aangebracht.

Regel 20 wordt:

20X=127

Regel 270 wordt:

270COLOUR(K&3);PLOT13,N,M

SOFTWARE: Disc-disc copy - een GDOS 1.5 utility

Jan Biel

De hierna beschreven utility is bedoeld om complete schijven op GDOS 1.5 formaat te copieren. Een opmerking vooraf:

!! Double Density Only !!

Het programma gebruikt het gebied #2800 - #8000 als buffer; het programma moet dus op #8200 of zo staan.  
Eerst moeten we invoeren in welke drive(s) zich de source- en target disc bevinden; en hoeveel tracks deze schijven zijn.  
Hierbij geldt de restrictie dat we alleen een 80-tracks schijf kunnen copieren naar een 40-tracks schijf wanneer de 80 tracks schijf voor minder dan de helft beschreven is met programma's. Bovendien hebben we hier twee drives voor nodig.  
Vervolgens worden beide drives op track zero gezet m.b.v. \*DIR. Het programma rekent dan uit in hoeveel slagen de schijf gecopieerd kan worden. Het aantal te copieren sectoren wordt op het scherm geprint; bij gebruik van 1 drive tevens het aantal copieerslagen.  
Bij gebruik van twee drives gaat de rest vanzelf. Hebben we slechts 1 drive ter beschikking dan wordt steeds de mededeling

INSERT SOURCE DISC & PRESS KEY

dan wel

INSERT TARGET DISC & PRESS KEY

op het scherm getoverd. Duidelijk nietwaar?!  
Na het copieren wordt de directory getoond van de target disc. Let op: bij het copieren van 80- naar 40-tracks klopt het aantal vrije sectoren niet; Dit komt omdat de schijf 1 op 1 gecopieerd wordt: sector voor sector dus.



anders alleen maar uitgevoerd wordt en niet wordt teruggegeven aan de aanroepende routine. Dit vond ik uiterst "inconvenient", zodat ik besloten heb om t.b.v. de copy-toets behalve de read-character routine ook het originele Atom-toetsenbord een keer te scannen en alleen als de copy-toets wordt ingedrukt deze te veranderen in ascii-code 4. Als je nu een ascii-toetsenbord aan je Atom hebt aangesloten, hoef je er alleen maar voor te zorgen dat je op een of andere door jou gekozen manier een CTRL-D kunt overseinen.

Doordat de oorspronkelijke vdu-routines, die een onderdeel van de word pack vormden nu niet meer gebruikt worden, is er een zee van vrije ruimte ontstaan. Ik heb niet precies uitgezocht welk gedeelte dat is, maar alles bij elkaar moet het zo ongeveer een kleine anderhalf kbyte bedragen.

FOOTWARE:Nieuwe listing van SQUEEZE.BAS

Bram Poot

In de vorige aflevering van het Acorntjesbrood is de listing afgedrukt van de utility SQUEEZE. Helaas is bij de uitdraai van die listing overduidelijk het e.e.a. misgegaan. De regels 230, 240 en 390 zullen zeer zeker foutmeldingen veroorzaken en wie weet zijn er meer gedeeltes verminkt, die niet zo eenvoudig zijn terug te vinden. Vandaar dat nogmaals een (nu hopelijk correcte) listing wordt geplaatst.

Verder is het zeker niet nodig dat het programma op #B200 wordt geplaatst, zoals in de "information" wordt gesuggereerd. Sterker nog, het beginadres van de geheugenbuffer (zie regel 50) gaat er zelfs van uit dat het programma op #2900 ligt. U kunt natuurlijk (zoals al in de beschrijving was verteld) het programma vanuit de graphics draaien teneinde een grote buffer te kunnen definiëren. Een opmerking die hierop aansluit en eigenlijk de vorige keer al vermeld had mogen worden is dat als een file groter is dan de geheugenbuffer, deze niet wordt geSQUEEZEd, maar onveranderd op dezelfde plek op schijf blijft staan.

```

XXX   XXX  XX  X X  X  XXX  XX  X XXXX  X  XXX  X XX  X  XXX
X  X  X  X  X X X X X  X  X  X X X  X X X  X X X X X X X  X
X  X  X  X  X XX X X  X  X X X  XX X  X X X  X X  XX X  X
XXXXX XXXXX X  X XX  X  X X  X X  X X X XXX X  X  X XXX
X  X  X  X  X X X X  X  X X  X X  X X X  X X  X X  X  X
X  X  X  X  X  X X X  X  X X  X X X  X X X  X X  X  X  X
X  X  X  X  X  X  X XXX  X  X XXXX  X  XXX  X  X  X  XXX

```

In de volgende aflevering van het ACORNTJESBROOD zal bij leven en welzijn en goede zin een epromprogrammer gepresenteerd worden waarmee het mogelijk is om 4, 8 of 16k eeproms te programmeren.

## RESET-MODIFIER

## INFORMATION:

START: #2900, END: #2DE7, LENGTH: #04E7 (= 1255)

#LINES: 94, LAST LINE: 940, #PAGES: 1

LONGEST LINE: 80 (= 27 CHAR'S)

```

10 PROGRAM RESET-MODIFIER      250:new      = #C2B2          490 jsr vdu80_ini          730 stx #FE
20                               260                               500:no_vdu80              740 ldx #B35
30 COPY#C000#CFFF#6000        270 .code base          510 jsr message            750:more_2
40 COPY#F000#FFFF#7000        280 .ram (base-#B000)    520                               760 sta #02EB,x
50                               290                               530 .byte 6,12              770 dex
60 ASN-V                       300 ldx #B00              540 .byte "ACORN ATON"      780 bpl more_2
70 PASS2;P."PASS1";GOSUBa     310:more''              550 .byte 10,13             790 bai vdu80
80 PASS1;P."PASS2";GOSUBa     320 ldy #B80              560                               800
90 END                          330:more'                570 nop                     810 .code #C550
100                               340 dey                  580 jsr gdos                 820 .ram #6550
110aASM-B                      350 bne more'            590 jmp new                  830
120                               360 dex                  600                               840 nop;nop;nop
130 .opt:0100 0000             370 bai more''           610:reset                   850 nop;nop
140 .list                       380                               620 ldx #B17                860
150                               390 lda #B8A             630:more                    870 .code #CA24
160:base      = #FF3F           400 sta port_c            640 lda #FF9A,x             880 .ram #6A24
170:port_a    = #B001           410 lda #B07              650 sta #0204,x             890
180:port_b    = port_a+1        420 sta port_b            660 dex                      900 nop;nop;nop
190:port_c    = port_b+i        430 bit port_a            670 bpl more                 910 nop;nop
200                               440 bvc reset             680 txa                      920 .end
210:vdu80_ini = #0400           450:vdu80                690 inx                      930
220:vdu80_base = vdu80_ini      460 lda vdu80_base        700 stx #EA                  940 RETURN
230:message   = #F7D1           470 cmp #94C              710 stx #E1
240:gdos      = #E000           480 bne no_vdu80         720 stx #E7

```



```

1130 STY MEMH
1140 STY MENL
1150:ROMPLOT
1160 STY MEMH
1170 STY MENL
1180 LDA #17 \ SMALL/LARGE PICTURE INDICATION
1190 BNE SHALL

2310 UNTIL C=CH*C*
2320 ELSE IF C(=CH*7* AND C)=CH*0*;A=C
2330 MEND
2340
2350 GERRIT HILLEBRAND
2360 HENGELD
2370 CONFUSION 86

```

SYSTEMWARE: Gemodificeerd C- en F-blok

Jan Biel



De meesten van U zullen waarschijnlijk net als ik in de loop der jaren hun Atom uitgebreid hebben met een aantal zaken, zoals daar zijn Floating Point Rom, Ram van #0000 tot #A000, disk interface, 80-kolommen kaart e.d. Een nadere bestudering van het C- en F-blok (b.v. middels Atom-ware I en III) leert ons dan dat in deze beide blokken een aantal zaken overbodig c.q. lastig zijn. Om er twee te noemen: de test op de aanwezigheid van de FP-rom en de Reset-routine.

Om met deze laatste te beginnen: Wanneer we in het bezit zijn van Ram vanaf #2900 en verder is de in het F-blok aanwezige test hierop overbodig geworden (#FF7C - #FF9A). Met andere woorden: 27 bytes voor andere zaken beschikbaar. Verder is het in een aantal gevallen lastig dat na het indrukken van de Break-toets elke keer alle Array-adressen (#2EB, #320) gereset worden op #FF en dat tevens alle vectoren (#200 - #21C) gereset worden op hun default waarden. Dit laatste betekent dat telkens na Break een eventueel aanwezige DOS, VDUB0 routines etc. weer opnieuw aangelinkt moeten worden. Aangezien mijn systeem met al deze zaken uitgerust is en ik het zolangzamerhand heb ben steeds na Break alles weer goed in te moeten stellen middels een INI commando, dat een onderdeel van branquart is, heb ik de Reset-routine maar eens flink onder handen genomen.

Ten opzichte van de originele Reset-routine zijn de volgende zaken gewijzigd:

- na het loslaten van de Break-toets wordt gekeken of ook de CTRL-toets ingedrukt is.
- Drukken we alleen de Break-toets in dan wordt de 8255 opnieuw geïnitialiseerd, er wordt gekeken of er op #400 en verder VDUB0 routines staan (versie met jumptabel aan het begin --> zie Acortjesbrood 4.2.26 e.v.), op het huidige scherm wordt

## VIDEO-DIGITIZER

## INFORMATION:

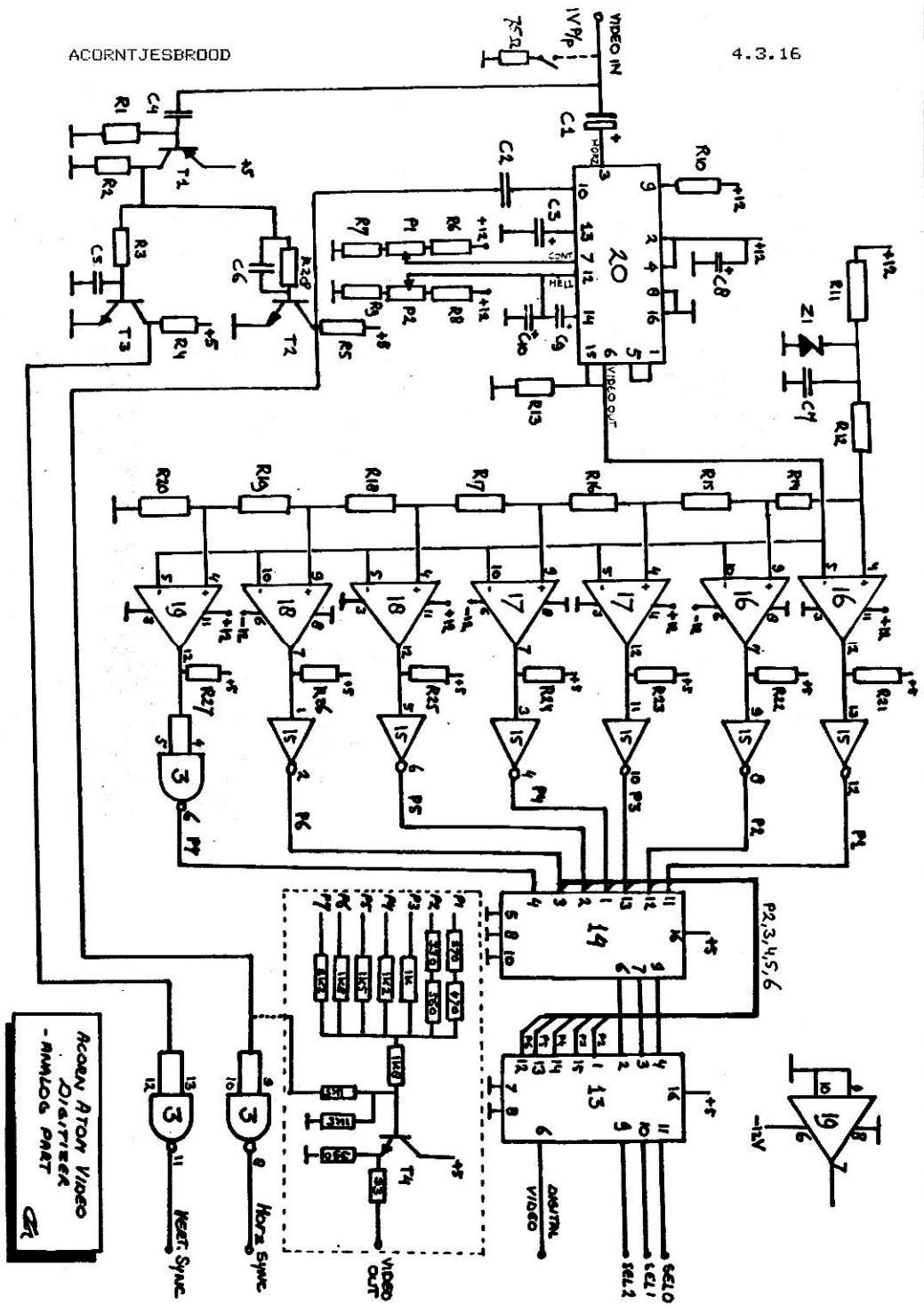
START: #2900, END: #3000, LENGTH: #1300 (= 506B)  
 #LINES: 237, LAST LINE: 2370, #PAGES: 3  
 LONGEST LINE: 2280 (= 68 CHAR'S)

```

10 PROGRAM VIDEO-DIGITIZER
20
30 REM VIDEO-DIGITIZER, ACORNTJESBROOD 4.3
40 REM GEBRUIKT SALFAA 1.5
50 REM VOOR DEELDEBEMERKINGEN:
60 REM GAGS V2.3
70 REM RXBOX
80 REM GDBS 1.5
90 REM DE CODE GEHEKRT MET **** KURKEN
100 REM AANGEPAST WORDEN VOOR UM EIGEN WENSEN
110
120 PRBC ASSEMBLE
130 ASH=0
140 .OPTIEN :0101 0000
150 .CODE 04000 \ ****
160 .RAM 04000 \ ****
170 .TABLE 04200,04300 \ ****
180 :PORT = 00FB4 \ ****
190 :PORTA = 00FB4 \ ****
200 :PORTB = 00FB5 \ ****
210 :PORTC = 00FB6 \ ****
220 :CNTRL = 00FB7 \ ****
230
240 :BUFFERH= 060
250 :BUFFERL= 070
260 :MEMH = 080
270 :MEML = 082
280 :LINE = 084
290 :ROW = 085
300 :GRAPH = 086
310 :SKIPVAL= 088
320 .BYTE 0FF,0E3,0C5 \ ****
330 .BYTE "INIT"
340 .BYTE 080+(INIT)/256
350 .BYTE (INIT)X256
360 .BYTE "SCAN"
370 .BYTE 080+(SCAN)/256
380 .BYTE (SCAN)X256
390 .BYTE 080
400
410 :INIT
420 JSR 0C4E4
430 NOP
440 LDA 0:1001 0001 \ INITIALIZE 0255
450 STA CNTRL
460 LDA 0430
470 STA PORTC
480 JNP 0C55B
490
500 :SCAN
510 JSR 0C8BC \ READ GREY VALUE
520 JSR 0C231
1200 LDA 00 \ LARGE PICTURE
1210 STA LINE \ NO LINES TO BE SKIPPED
1220 BEQ CONT
1230:SMALL
1240 LDA 0(256-192) \ SMALL PICTURE
1250 STA LINE \ 192 LINES,...
1260 LDA 04 \ 3 OF 4 LINES TO BE SKIPPED
1270:CONT
1280 STA SKIPVAL
1290 LDA 09B0 \ H.O ADDRESS SCREEN
1300 STA GRAPH+1
1310 LDY SKIPVAL \ X CONTAINS LINES TO DISPLAYED
1320:LINEPLOT
1330 DEX \ SKIP LINE ?
1340 BNE NOSKIP \ NO, CONTINUE
1350 LDY SKIPVAL
1360 BNE SKIPLINE \ YES, SKIP A LINE
1370:NOSKIP
1380 LDA (MEMH),Y
1390 EOR 00FF
1400 STA (GRAPH),Y
1410 LDA (MEML),Y
1420 INY
1430 EOR 00FF
1440 STA (GRAPH),Y
1450 DEY
1460 INC LINE
1470 LDA GRAPH
1480 CLC
1490 ADC 0920 \ ADAPT GRAPHIC POINTER
1500 STA GRAPH
1510 BCC SKIPLINE
1520 INC GRAPH+1
1530:SKIPLINE
1540 INC MEMH
1550 INC MEML
1560 LDA LINE
1570 BNE LINEPLOT
1580 INC MEMH+1
1590 INC MEML+1
1600 INC ROW
1610 INC ROW
1620 LDA ROW
1630 STA GRAPH
1640 CMP 032
1650 BNE ROWPLOT
1660 JNP 0C55B
1670
1680 .END OF ASS
1690
1700 PEND
1710

```







## De software

Het programma dat op een van de volgende bladzijden is afgedrukt, dient men te zien als een basispakket. Men kan 1-bits plaatjes (geen grijsnivo handling) uit 3-bits binnen halen met een keuze uit twee formaten:

- 'klein' plaatje 192\*256 geheugen #8000-#9800
- 'groot' plaatje 256\*256 geheugen #8000-#A000

Alle niet gedecodeerde beeldinformatie wordt eerst opgeslagen in #6000-#8000. Na het binnenhalen volgt het converteren naar het beeldscherm. Indien men de mogelijkheid heeft om het hele plaatje in het geheugengebied #8000-#A000 graphisch uit te printen, dan kan men het formaat 256\*256 kiezen.

In het programma zijn de graphische dump routines uit de RXBOX gebruikt en dus #8000-#9800 wordt uitgelezen.

Met enkele faciliteiten uit GAGS is het mogelijk om kadertjes om een plaatje te maken en het beeldscherm enigszins verschuiven. Het grijsnivo kan direct tijdens het scannen verandert worden door de toetsen 0 tot en met 7 als grijsnivowaarden in te drukken. Het scannen stopt bij het indrukken van H (hold). Nu kan men kiezen uit de volgende bewerkingen:

- |                      |                                 |
|----------------------|---------------------------------|
| -C- continue         | doorgaan met scannen            |
| -B- border           | maakt een klein lijstje (4 dik) |
| -E- extended border  | lijstje van 8 pixels dik        |
| -L- left scroll      | plaatje scrollt naar links      |
| -R- right scroll     | plaatje scrollt naar rechts     |
| -U- up scroll        | plaatje scrollt naar boven      |
| -D- down scroll      | plaatje scrollt naar beneden    |
| -P- printer dump     | plaatje wordt klein geprint     |
| -B- big printer dump | plaatje wordt groot geprint     |
| -S- save             | plaatje wordt op schijf gezet   |
| -Q- quit             | programma wordt beëindigd       |

De video-scan routine werkt als volgt:

- initieer grijswaarde
- wacht op verticale sync
- skip 40 lijnen (i.v.m. zwarte horizontale balken)
- lees 256 lijnen van 1 kolom en sla dit sequentieel op
- volgende kolom

De conversie routine is gebaseerd op:

- indien klein plaatje, om de 3 lijnen 1 lijn overslaan (want  $3/4 * 256 = 192$ )
- indien groot plaatje, alle lijnen converteren
- zet sequentiele data om in kolommen

Het programma maakt P-Charme uitbreidingen aan en wel de volgende statements:

- INIT - initialiseer de 8255
- SCAN g,p - Scan en display plaatje met
  - g= grijsnivo (0-7)
  - p= plaatgrootte (0=groot 256\*256, 1=klein 256\*192)

Deze statements zijn dan ook buiten dit programma te gebruiken om bijvoorbeeld later de 3bits informatie van een beeld op te vragen en te verwerken om op de printer in posterformaat met prachtige grijs tinten (van 0 to 7 = 8 grijsnivo's !!) het plaatje uit te printen. Een algoritme voor dit soort uitprinten naar aanleiding van beschikbare grijsnivo's heb ik nu ten behoeve van een andere digitizer voor een FX80 of MX80

helderheidsnivo's gesplitst werd, volgt nu in het digitale gedeelte het aftasten van dit signaal langs de tijds plaats. Hier volgt dan ook direct een hardwareaanpassing die ik ingebouwd heb. Na elke horizontale puls duurt het enige tijd (ongeveer 9 microseconden) voordat de eerste lijn informatie aangeboden wordt. Doch direct na de horizontale sync-puls zal de digitizer al beginnen te scannen en hierdoor alleen maar zwarte balken ontvangen. Ik heb de horizontale puls als ingangssignaal gebruikt voor een monostabiele (IC 22), die de puls ongeveer 9 microseconde vertraagd (in te stellen met P4) en daarna een kunstmatige syncpuls genereerd van ongeveer 4 microseconde. Hierdoor zullen de (voor mij althans) storende zwarte balken aan de linkerkant van het plaatje verdwenen zijn. Bij een videobeeld wordt de beeldinformatie niet punts-gewijs verstuurd, doch de beeldpunten vloeien als het ware in elkaar over. Omdat een computer alleen maar een 0 of een 1 begrijpt, moet de beeldinformatie met een bepaalde aftastfrequentie gesampled worden.

Het oplossend vermogen van de digitizer bedraagt 256 punten. Een lijn duurt 64 microseconden. Na het aftrekken van de tijd die de horizontale sync voor zijn rekening neemt blijft er ongeveer 51 microseconden over voor de beeldinformatie. Daaruit volgt de aftastfrequentie van  $51/256$  microsec = 0.199 microseconden per punt (dit komt overeen met een frequentie van 5.02 Mhz). Deze frequentie wordt opgewekt door de oscillator, gebouwd rondom de 74LS629 (IC2). Bij elke neergaande flank van de aftastfrequentie moet een beeldpunt in de computer gelezen worden. (elke 199 ns.) Omdat ons Atompje niet zo snel is, worden er steeds blokken van 16 punten per lijn ingelezen. Hiervoor heeft de Atom dan 64 microseconden. Daarna worden er 16 punten (van dezelfde kolom!) binnengehaald van de volgende lijn, etc tot en met lijn 256. Hierna wordt de volgende kolom binnengehaald totdat alle 16 kolommen (256/16) ingelezen zijn. Dit komt overeen met het lezen van de punten 1 tot 16, daarna 17 tot 31, etc.

Het beeld wordt dus in 16 kolommen binnen gelezen. Voor elk woord van 16 punten heeft de computer 64 microseconden, wat goed te doen is. De leestijd voor een compleet beeld bedraagt dus:  $64 \text{ micros} * 256 \text{ lijnen} * 16 \text{ kolommen} = 0.262$  seconde. Deze manier van beeld inlezen is wat deze digitizer zo goedkoop maakt. De Atom programmeert de digitizer voor het lezen van een bepaalde kolom, door de kolom waarde aan IC11 aan te bieden. Een 16-bits teller (IC 9 en 10) telt de beeldpunten en een comperator geeft een seintje indien de actuele kolom gelijk is aan de verlangde kolom. Dit signaal activeert op zijn beurt een schuifregister voor het lezen van de 16 punten (IC 4,5,6,7). Hierdoor is het ook mogelijk om delen van een video beeld zeer gedetailleerd te vertonen.

Het vertikale synchronisatiesignaal wordt softwarematig opgevraagd. Nadat deze actief is, begint het lezen van de lijnen. Het horizontale syncsignaal activeert de oscillator en reset de puntenteller. De aftastfrequentie dient na de horizontale sync puls altijd met de gelijke fase beginnen, anders kunnen de punten in een kolom ten opzichte van elkaar gaan verschuiven. Een eis is dan ook: zeer goede flanken van de horizontale sync en een goede oscillator.

Het storende effect van de zwarte balken aan de linkerkant van het beeld (na een horizontale sync even geen video) is hardwarematig opgelost. Hetzelfde zwarte-balken effect krijgen we ook aan de bovenkant van het beeld na de vertikale sync-puls. Dit effect is opgelost door de software.



Degenen die het vorige broodje hebben bekeken en op de regioavond van April aanwezig waren, hebben kunnen zien dat er veel nieuwe plaatjes voor de Atom in omloop zijn en nog steeds verschijnen. Dit alles is mogelijk geworden door een video-digitizer welke ik gebouwd heb naar een schema uit het Oktober 1985 nummer van het Duitse computertijdschrift MC. Hierin staat een complete video-digitizer voor een 68000-systeem met aanpassingen voor een Apple II beschreven.

Met behulp van de digitizer kun je nu beelden van camera's, video-recorders en TV's met video uitgang scannen en op de Atom in de hoogste graphic mode laten zien (en verwerken natuurlijk). Het nu volgende verhaal is een compilatie van de originele tekst uit MC, geschreven door Kurt Moraw en aangevuld met eigen tekst t.b.v. door mij aangebrachte veranderingen en software.

Wat is nu precies de werking van zo'n digitizer? Het is een schakeling welke een videosignaal van 1Vpp nodig heeft. Dit videosignaal wordt omgezet in digitale informatie (7 bits) en doorgegeven aan de computer (1 of 3 bit(s)). Displayed men deze informatie nu in de volgorde zoals het ook gesampled is, dan krijgt men een digitale voorstelling op de Atom van het oorspronkelijke videobeeld.

Enige theoretische zaken:

Het videobeeld wordt lijn voor lijn na elkaar verstuurd. Het opbouwen van een beeld duurt in de regel 20 msec. (50 Hz). Het totale beeld heeft 625 lijnen en elke lijn duurt 64 microsec (15625 Hz lijnfrequentie). In deze 64 microsec bevindt zich de totale video informatie van 1 lijn. Elke 20msec wordt een halfbeeld verstuurd (interlaced video). De resolutie van deze digitizer beslaat 256\*256 punten. Dit valt nog ruimschoots binnen de 312.5 lijnen van een halfbeeld. Deze digitizer gebruikt dus van deze 312.5 lijnen maar 256 lijnen. Bij elke frame-sync puls zal de digitizer opgestart worden voor het scannen van een compleet beeld. Tussen deze verticale-synchronisatie pulsen zullen er 312 lijnen ontvangen worden. Om de monitor, en dus ook de computer duidelijk te maken wanneer een lijn begint, wordt er aan het begin van elke lijn een zgn. horizontale syncpuls gegeven. Tussen twee van dit soort HS pulsen wordt de informatie van een lijn overgedragen. De frequentie van de informatie in een beeldsignaal voor een lijn bedraagt ongeveer 4 MHz en het ingangsgedeelte van de digitizer is hier volledig op afgestemd. De helderheids- en contrast nivo's (black-level op 0V en white-level op 0.5V) kunnen in de digitizer nog zelf ingesteld worden. De extra

## V4: Het klok-circuit.

De oscillator (74LS04) wekt een frequentie op van 14..16 MHz. Deze frequentie wordt toegevoerd aan het schuifregister (74LS166), dat met deze frequentie bits uitschuift naar de videocombiner.

Verder wordt dit kloksignaal gebruikt om een teller te sturen (74LS161) die dit signaal door 8 deelt. Als de teller stand 15 bereikt gaat de TC-uitgang hoog. Bij de volgende klokpuls worden teller en schuifregister parallel geladen, de teller met een preset waarde van 8 (om voor het 8-delen te zorgen), het schuifregister met data uit de karakter-eprom.

Het gedeelde kloksignaal wordt gebruikt als karakter-klok voor de videocontroller en om de data uit het videogeheugen vast te houden door middel van de beide 74LS174's.

## 5: De videocombiner

Op de uitgang van het schuifregister (74LS166) verschijnen met de regelmaat van de klok enen en nullen, bepaald door de inhoud van de laatst uitgelezen geheugenplaats van de karakter-eprom. Met drie NOR-poorten (open collector) wordt het uiteindelijke videosignaal gemaakt en opgeteld bij de synchronisatie-signalen die de controller op gezette tijden veroorzaakt. Als het hoogste bit van de op dat moment gelezen byte uit het videogeheugen hoog is, wordt het niveau van het videosignaal bijna gehalveerd, wat resulteert in een gefimde weergave van het betreffende karakter op het scherm. Wordt HS of VS hoog, dan wordt het videosignaal helemaal naar nul getrokken.

Ter voorkoming van storing in het videosignaal t.g.v. vrieze voeding verdient het de voorkeur de 74LS33 en de spanningsdeler via een filttertje te voeden. Zo'n filter kan bestaan uit een klein spoeltje (ferriet met ca. 40 windingen dun schellakdraad) in de voedingslijn en een condensator van 47nF naar de aarde. Vervolgens verkrijgt men een vlekkeloos videosignaal.

## 6. Lijst van ic's:

	6845	videocontroller
	6116	2k*8 statisch RAM
	2764	karakter-eprom
3*	74LS157	multiplexer
2*	74LS174	6-bits data flipflops
	74LS245	databus-buffer
	74LS04	6* inverter
	74LS161	(of 163) presettable teller
	74LS166	synchroon schuifregister
	74LS33	4* open collector NOR

Paul Klomp  
 Drieturven 3  
 7552KT Hengelo  
 074-427191

HARDWARE: de 80-KOLOMMENKAART van Peter Huisken P. Jansenkomp

Deze beschrijving behandelt het schema met de 2764 eeprom, die op nogal wat punten verschilt van de eerdere versies. Om iets van deze tekst te begrijpen lijkt het me noodzakelijk het schema ernaast te hebben liggen. Voor het tot in details nagaan van de timing zijn de datasheets van de 74LS161/163, de 74LS166 en de 6845 onmisbaar.

#### 1: Het videogeheugen

Een 6116 2K\*8 statische RAM-chip fungeert als videogeheugen. Als er een karakter in dit geheugen geschreven of gelezen moet worden, moet de processor toegang hebben tot niet-write en de data- en adresbus van het videogeheugen. Hiertoe wordt de niet-enable lijn laag gemaakt, een decodeerschakeling voor het betreffende geheugengebied moet dus nog aan de schakeling worden toegevoegd).

Wanneer de microprocessor het videogeheugen aanspreekt worden door de multiplexer (3\* 74LS157) de adreslijnen van de processor doorgeschakeld naar de adresingangen van de geheugenchip en ook wordt de buffer (74LS245) geactiveerd. De richting waarin de buffer doorlaat is afhankelijk van de read/write lijn en datatransport verloopt zo overeenkomstig de wensen van de 6502.

(In mijn proefversie van de 80-kolommenkaart voldeed een Hitachi 150nS (HM6116LP-3) chip niet helemaal: bij teruglezen uit het videogeheugen kwamen er regelmatig "foute bytes" voor; met een andere 150 nS-chip werkte alles goed... heel geheimzinnig.)

Steeds als de processor geen toegang eist (dus wanneer niet-enable hoog is) laat de multiplexer de adreslijnen (MA0 .. MA10) van de videocontroller door. Deze adressen leveren ASCII-waardes uit het videogeheugen die via de latch (2\*74LS174) een deel van de adressering van de karakter-eeprom bepalen.

De lees/schrijflijn wordt met de adreslijnen meegeschakeld, zodat alleen de microprocessor in het videogeheugen kan schrijven. Zijn de adreslijnen van de videocontroller geselecteerd dan is de lees/schrijflijn van de 6116 automatisch hoog.

#### 2: De videocontroller

De 6845 zorgt voor het opwekken van de uitleesadressen voor het videogeheugen, de rij-adressen voor de eeprom en de horizontale en verticale synchronisatiesignalen.

De interfacing met de 6502 gaat als volgt:

De 6845 gedraagt zich als 2 adressen in de geheugenkaart van de ATOM. Door eerst in het eerste adres een waarde te zetten (A0 = laag) wordt 1 van de achttien registers geselecteerd. Daarna wordt bij een schrijfactie de data in het hoge register geschreven; bij een leesactie wordt het hoge register gelezen. Het lezen en schrijven in de registers wordt gesynchroniseerd door PHI-2 (aan de E-ingang).

## HARDWARE: Programmeerbaar programmeren

Robert Bernink

De hier beschreven PSU regelaar kan men programmeren om een bepaalde uitgangsspanning te geven. De toepassingen zijn legio, zoals een electromotortje etc. Maar de allereerste gedachte gaat toch uit naar een eprom-programmer.

De trend tegenwoordig is, om de EPROM, van welk type dan ook, met behulp van een PIA of latches te programmeren. Deze nemen dan de functies van adres- en datalijnen van de EPROM over.

De programmeerspanningen van de diverse typen en fabrikanten lopen uiteen en zijn vaak 12.5, 21 en 25 Volt. Indien we toch een PIA of latch gebruiken, dan zijn er misschien ook nog wel 3 lijntjes over om deze programmeerspanning vol automatisch voor de gekozen EPROM in te stellen.  
Voordeel: geen schakelaars meer (en dus misschien foutief instellen); alles software gestuurd.

Zie schema.

Stel :  $R_s = P_1 + P_2 + P_3$   
met  $PA_1 = PA_2 = PA_3 = '1'$

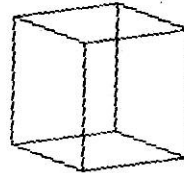
dan geldt:  $V_{out} = (5 * (R_s + 4K7) / 4K7)$

Na de bouw, de afregeling:  
Hierbij moet  $U_3 < U_2 < U_1$

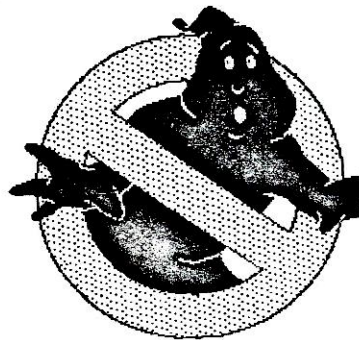
$PA_0 = PA_1 = '0'$   $PA_2 = '1'$   
 $P_3$  afregelen op gewenste spanning  $U_3$

$PA_0 = '0'$   $PA_1 = PA_2 = '1'$   
 $P_2$  afregelen op gewenste spanning  $U_2$

$PA_1 = PA_2 = PA_3 = '1'$   
 $P_1$  afregelen op gewenste spanning  $U_1$



Robert Bernink  
Oldenzaal

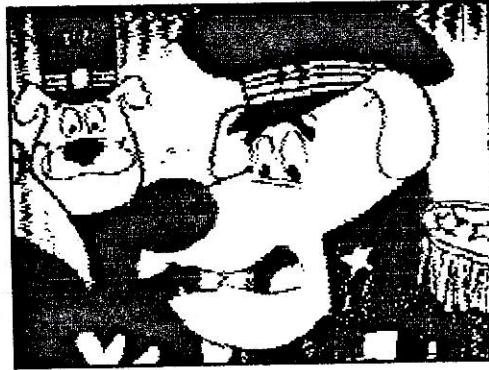


## INHOUDSOPGAVE:

Blz. Wat staat daar ?!

1. Redactioneel, zie foto
2. .... zoals altijd
3. Ledenlijst
4. Programmeerbaar programmeren door R. Bernink
6. 80 breed maar dan simpel, Peter en Paul
10. Voor het resultaat: zie broodje; Gerrit Digitaal
20. Reset a la Acorn : Jan
23. Lay-outen met tachtig tegelijk van Bram
24. Vorige maand iets te veel gsqzd; Bram reprise
26. Jan 80 doet het met twee tegelijk
28. Onduidelijk iets van Rob
30. Niet van Falco/Bolland/Bolland maar gewoon Gerrit
31. Wat U doen moet staat erbij vindt Rob
34. Retry op PL8 and Artist Impression: Paul

DEADLINE BROODJE 4.4: 21-5-'86



We houden u op de hoogte

Spreuk: IK ATOMIZE, DUS IK BESTA

