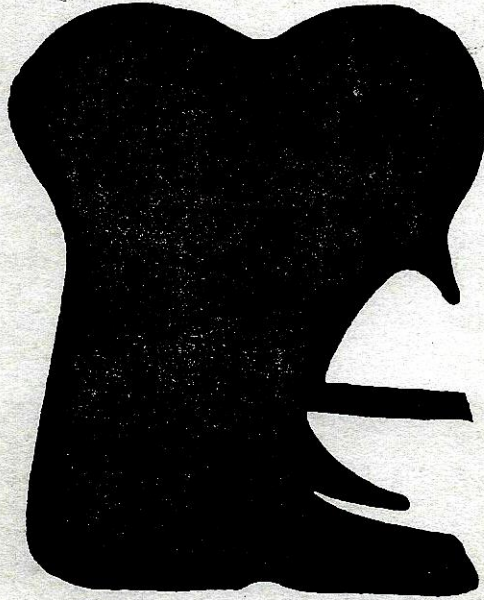


PRIJS F.2.50

Jaargang: **EE E**

Nummer: **EE E E**



Acorntjesbrood

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

ACORNTJESBROOD IS HET ORGAAN VAN DE
ATOM COMPUTER CLUB

TWENTJE

=====

NAMENS: De redactie

Gerrit Hillebrand

=====

REDAC (ORN) TIONEEL

Hallo Atomisten. Allen terug van vakantie? Weggeregend zoals ik? Hopelijk was het voor U allen een plezierige (en misschien zonnige) vakantie, een atomaire vakantie kan het niet geweest zijn gezien de copij wat binnen is gekomen. Niks, nop, zero, vin-de-blanc. Treurig. Dit blad is door ongeveer vier man volgeschreven.

Toch is het naar mijn mening wel weer het nivo wat we gewend zijn: nivo Twente. Een van de hoogtepunten is wel (voor mij althans) de openbaring van heuse macro's op de Atom, gedirigeerd door SALFAA 1.5. Dit nummer bevat ook de laatste voor aankondiging van GDOS (nu 1.3 en Double Density!). Afwachten wat het syndicaat er mee doet.

Hopelijk staat het volgend Broodje vol met Uw bijdrage, we zien het met grote belangstelling tegemoet.

Hebt U trouwens het artikel over de Federatie gelezen in het maadblad PCM (juli/augustus 1985)? Onder de titel "ACORN ATOM rol voor soldeerbout" kunnen we lezen dat de nieuwe voorzitter Nico Stad de club weer nieuw leven heeft ingeblazen. Prachtig gewoon, WE zijn met nogal wat dingen bezig volgens Nico. Als U na het lezen van dit artikel nog niet bent overgehaald om een ATOM aan te schaffen voor het luttel bedrag van 149 gulden, dan bent U, voor wat betreft het computerisme en in het bijzonder: het Atomisme, verloren. Dus wat dat betreft: U zit gebakken!

Ik zou het bijna vergeten, wat vindt U van ons nieuwe blad? Het traditionele Broodje op A4 is verdwenen en heeft plaats gemaakt voor dit nieuwe op A5 gedrukte exemplaar. Met de overgang van het formaat zijn we ook overgestapt wat betreft de drukker, daar we wegens bepaalde, voor mij nog steeds zeer onduidelijke, redenen bij de oude drukker moesten opstappen. Zoals U ziet, verliezen we daar niets aan.

Er is nog wat te verdienen aan de Atom. Althans, deze conclusie trek ik bij het zien van de nieuwe catalogus van de Engelse firma Bearsoft. Enige tijd geleden kreeg ik zo'n catalogus toegestuurd, waaruit blijkt dat deze firma oude software van A&F, Acornsoft en Program Power opnieuw in de handel brengt met daarnaast ook software uit de eigen keuken. Sinds kort zijn ook te koop: backplanes, 1770-disc controller, Atom emulator en verschillende nieuwe utilities, zoals databases, network-communication package en een nieuwe editor. Over het algemeen kan men zeggen dat deze firma zeker niet goedkoop is, b.v.: de disc-controller print (+onderdelen) plus de apart bij te bestellen DOS kost ongeveer 550 gulden!! (exclusief drive !!!) Wie meer informatie wil kan de catalogus bestellen bij Bearsoft (officieel heten ze BEAR HARDWARE ?!?), te vinden bij het volgende adres:

Brand 05490 67906

Ledenlijst Atom Club Twente: September 1985

Naam:	Straat:	Plaats:	Telefoon:
Adeema F.	Wierdensestraat 14	7443 AC Nijverdal	05486-16225
Alphen M.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	053-895011
Boon H.M.	Maldengenstraat 9	7641 DW Wierden	05496-5187
Borgerink J.A.L.	Geert Grootestraat 9	7665 AV Albergen	05494-1612
Brand J.B.M.	Valeriusstraat 177	7604 CM Almelo	05490-25030
Bruggen T.van	Spartastraat 67	7535 BL Enschede	053-316709
Busschers G.M.v.	Krabbenbosweg 45	7555 EC Hengelo	074-439048
Collet C.	Prins Bernhardlaan 8	7622 BG Borne	074-661989
Couprie J.P.	Braakwansdijk 52	7462 LW Rijssen	05480-19477
Dam H.ten	Josef Israelstr. 13	7606 HM Almelo	05490-24703
Dijkstra J.K.	Julianastraat 13	7586 AT Overdinkel	05423-83698
Droog E.	Hanenberglanden 111	7542 ET Enschede	053-769228
Eerden H.van	Hofkampstraat 42	7607 NJ Almelo	05490-24168
Egberink P.	Anjerstraat 52	7572 VR Oldenzaal	
Elhorst W.	Leerlooierstraat 9	7447 XZ Hellendoorn	05486-54916
Exter Th.den	Calslaan 52-13	7522 ZE Enschede	053-895055
Gosselink H.J.	Pathmosplein 12	7545 TT Enschede	
Guley F.	Boekeloseweg 134	7553 DR Hengelo	074-436452
Hillebrand G.	Mendelssohnstraat 30	7557 BJ Hengelo	074-912931
Hofstede C.	Grintenbosweg 51	7482 SL Haaksbergen	05427-13862
Horning R.J.	Varviksingel 198	7512 ES Enschede	053-317943
JansenKlopp P.	Drieturven 3	7552 KT Hengelo	074-427191
Janssen B.	Geulstraat 56	7523 TV Enschede	053-355046
Jelma B.	Biljoen 43	7608 MB Almelo	05490-67479
Jong B.de	Emastraat 268a	7513 BJ Enschede	053-300463
Kamphuis B.	Rebrandtlaan 274	7545 ZB Enschede	053-322906
Klijnstra P.	Calslaan 7-304	7522 MH Enschede	053-895012
Laarman J.	Potgieterstraat 13a	7442 XP Nijverdal	05486-17178
Leeuw H.de	Meleveldstraat 46	7557 JH Hengelo	074-421873
Moraal A.M.	Steenweg 52	7531 BW Enschede	053-336621
Muiser J.N.	Kotkampweg 188	7531 JK Enschede	053-354225
Mullink S.	Handellaan 20	7522 KM 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 MG Enschede	
Poot G.B.	IJsselstraat 1	7555 KT Hengelo	074-434779
Raven M.	Calslaan 3-202	7522 MM Enschede	053-895005
Scherphof A.T.	Noorderstraat 57	7607 VT Almelo	05490-25151
Scholten J.A.	Rijnstraat 30	7442 ER Nijverdal	05486-13129
Schuringa W.	Velveweg 109	7533 XG Enschede	053-300728
Sessink W.	Janesrossstraat 35	7534 ZZ Glanerbrug	053-612772
Silvis J.M.	Else Mausstraat 62	7558 RB Hengelo	074-775963
Sinke A.	Grotestraat 66	7443 BK Nijverdal	
Snijder P.	Bakkersveenweg 17	7676 AM Westerhaar	
Speel R.	Alexanderstr.54-2 Hg	7553 WZ Hengelo	
Spengelink F.	Veldsweg 98	7441 CN Nijverdal	05486-11403
Stekelenburg R.	Prins Hendrikstr.122	7571 BX Oldenzaal	
Stoop M.B.M.	De Grutto 27	7609 DC Almelo	05490-24178
Veen A.van de	Merelweg 26	7442 CB Nijverdal	05486-14234
Veenstra R.E.	Revisiaan 57	8024 CD Zwolle	038-532984
Velde K.van der	De Gast 27	9861 BM Grootegast	05946-12142
Veldhuis H.	Sperwerweg 29	7701 JV Dedeasvaart	05230-12605
Verhoeven W.L.A.	Witbreuksweg 377-403	7522 ZA Enschede	053-337026
Vink A.	Lankheethoek 13	7546 BP Enschede	053-775298
Visser V.	Witbreuksweg 395-308	7522 ZA Enschede	
Wamelink H.J.	Pashegge 62	7103 BJ Winterswijk	05430-19601
Werken W.J.A.	Moellenbergstraat 47	7582 ZX Losser	
Mezel R.van	Kroonprins 20	7481 CJ Haaksbergen	05427-12417
Wietmarschen H.L.M.van	Bachstraat 41	7651 MS Tubbergen	05493-2870
Wisse R.G.A.	--- Onbekend ---	--- Onbekend ---	
Zieleman G.	Lochterseweg 41	7442 BN Nijverdal	05486-16992

Het kan natuurlijk wel wat fraaier:

```

10 PROGRAM SPREEK!
20 PROC SPEAK(N),I
30   FOR I=1 TO N
40     A=FF(I) OR #80
50     LINK #FEFB
60   NEXT I
70 PEND
80 !#208=!#208+3
90 READ X
100 DIM FF(X)
110 FOR I=1 TO X
120   READ FF(I)
130 NEXT I
140 PRINT $2
150 SPEAK(X)
160 PRINT $3
170 !#208=!#208-3
180 DATA .. ; REM AANTAL FONEEMEN
190 DATA .....
200 DATA ..... etc.

```



Als u tijdens het uitspreken van een foneem de processor i.p.v. een eeuwigheid te laten wachten, nog iets nuttigs wilt laten doen, dient u de routine #FEFB te herschrijven tot iets soortgelijks, maar dan zonder BUSY-check. Als u de ACKNOWLEDGE en BUSY verknoopt, kunt u zelfs onder interrupt spreken. Ik vermoed echter dat het computerge-OH dermate snel gaat vervelen dat u hier geen behoefte aan zult hebben.

Het bovenstaande verhaal is geschreven nog voordat er ook maar 1 SP0256 daadwerkelijk aangesloten was. Toen dat eenmaal wel het geval was, is er een ware wildgroei ontstaan inzake de spraaksoftware. Een van de gemeenschappelijke onderdelen van de diverse benaderingen betrof het spreken onder interrupt, waarbij de klanken in een buffer worden bijgehouden. Het programmeren van deze zgn. queue is tamelijk ingewikkeld (kwalitatief niet, maar in concreto wel degelijk), zodat het voordeel van de eerder vermelde eenvoudige software inmiddels schijnbaar als een lachertje moet worden gezien.

Schijnbaar, omdat er zeer zeker toepassingen zijn waarbij het reeds aanwezig zijn van software bijna als een noodzaak moet worden beschouwd.

De voordelen van de queue moeten daarentegen ook weer niet onderschat worden. De huidige oplossing maakt het mogelijk om tijdens een programma (bijna) zonder tijdverlies hoorbare boodschappen aan de programmeur danwel spelletjesspeler te geven. Denk in dit verband bv. aan een "adventure" waarin de beschrijving van de ruimte hoorbaar wordt gemaakt, terwijl de computer intussen de nodige zaken instelt, controleert en/of aanpast.

Elders in dit blaadje staat een programma dat heel netjes zo'n queue aanmaakt en afhandelt. Het werkt zelfs samen met andere interrupts en ik moet zeggen dat het een waar genoegen is om de VIA in een dergelijke situatie zo druk bezig te zien.

KLOKJE is niet helemaal netjes geprogrammeerd omdat het de oude interruptvector niet opslaat, S&S daarentegen is wat dat betreft beter verzorgd. De volgorde is dus: KLOKJE runnen, SPEAK (initieren), SAY (kletsen EN tijd aflezen).

Aangezien tijdens het SAVEN en LOADen van files alle interrupts worden tegengehouden, loopt de klok niet door en wordt het spreken onderbroken. Als de COS-operatie afgerond is, gaat het babbelen door waar het gebleven was. Een DOS-operatie echter beïnvloedt geen van beiden merkbaar.

Met dit verhaal hebben we u voorlopig even genoeg verteld; het is nu de beurt aan de SPO256.

REAL TIME CLOCK

```

10 REM REAL TIME CLOCK
20 DIM KK(5)
30 INPUT "WAAR CODE "Q
40 T=50000;REM 20 MS
50 REM T PLUS OF MIN 100
60 REM VOOR FIJNREGELING
70 REM VIA ADRESSEN:
80 REM #BB08 TIMER 2 LAAG
90 REM #BB09 TIMER 2 HOOG
100 REM #BB08 AUXILIARY REG.
110 REM #BB08 INT. FLAG REG.
120 REM #BB0E INT. ENAB REG.
130 FOR I=0 TO 5;KKI=0;N.I
140 S=#235;REM 11 BYTES TIJDSTRING
150 P.#21;FOR A=0 TO 1;P=#
160
170:KK0 SEI
180 LDA@CH"0";LDX@9
190:KK1 STA S+1;X;DEX;BNEKK1
200 LDA@13;STA S+4
210 STA S+7;STA S+10
220\ ZET INTERRUPT VECTOR
230 LDA@KK2/256;STA#204
240 LDA@KK2/256;STA#205
250\ SET T2 TIMED INTERRUPT
260 LDA#BB08;AND@#DF;STA#BB0B
270\ ENABLE T2 INTERRUPTS
280 LDA@#A0;STA#BB0E
290\ START TIMER 2
300 LDA@T/256;STA#BB08
310 LDA@T/256;STA#BB09
320 CLI;RTS
330:KK2
340 LDA#BB08;AND@#20;BEQKK3+3
350 INC S+1;LDA S+1
360 CMP@1000000/T;BNEKK3
370 LDA@0;STA S+1
380 INC S+3;LDA S+3
390 CMP@CH":";BNEKK3
400 LDA@CH"0";STA S+3
410 INC S+2;LDA S+2
420 CMP@CH"6";BNEKK3
430 LDA@CH"0";STA S+2
440 INC S+6;LDA S+6
450 CMP@CH":";BNEKK3
460 LDA@CH"0";STA S+6
470 INC S+5;LDA S+5
480 CMP@CH"6";BNEKK3
490 LDA@CH"0";STA S+5
500 INC S+9;LDA S+9
510 CMP@CH"4";BNEKK4
520 LDA S+8;CMP@CH"2";BNEKK3
530 LDA@CH"0";STA S+9;STA S+8
540:KK4 LDA S+9
550 CMP@CH":";BNEKK3
560 LDA@CH"0";STA S+9;INC S+8
570:KK3 LDA S;BEQKK5
580 LDA@CH" ";STA#B016
590 LDA S+8;STA#B017
600 LDA S+9;STA#B018
610 LDA@CH":";STA#B019
620 LDA S+5;STA#B01A
630 LDA S+6;STA#B01B
640 LDA@CH":";STA#B01C
650 LDA S+2;STA#B01D
660 LDA S+3;STA#B01E
670:KK5
680 LDA@T/256;STA#BB08
690 LDA@T/256;STA#BB09
700 PLA;RTI;J;N.A;P.#6
710 @=0
720 P."TOTAAN:#"&P'
730 INPUT"OPSTARTEN "#P
740 IF ?P="CH"J";LINK KK0
750 END

```

TE*KOOP-AANGEBODEN: Draagbare Z/W televisie.

Deze TV heeft een beeld diagonaal van 31 cm, en is zeer geschikt als beeldscherm aan de Atom.
De TV is voorzien van een monitoringang waardoor 80 karakters per regel goed kunnen worden weergegeven m.b.v. een 80 kolommenkaart.
Ook geschikt voor ruisvrij plotten zoals Ed Droog heeft beschreven in Acorntjesbrood 1.4 pag.6.
Prijs fl.75.-- Te bevragen bij Ronald Boers.

```

550     STA B;STY B+1
580     LDA@#40
590     STA B+4;\ LOW VOICE
592     \LDA@#00
593\ NO MATTER WHAT VALUE,
594     STA B+2;STA B+3
595\ AS LONG AS THESE ARE EQUAL
596
597\ NEGATIVE TRANSITION CA1
598     LDA#B80C;AND@#FE
599     STA#B80C
600\ IRQ-VECTOR AT I10?
610     JSR SS9;BEQ SS5
620\ SAVE OLD IRQ-SERV.ROUTINE
630\ AND INSTALL NEW ONE.
640     LDA#204;STA V
650     LDA#205;STA V+1
660:SS4  STX#204
670     STY#205
680:SS5
740\ LET SP0256 BE QUIET
750\ BY SAYING HI OR LO PA1
760     LDY B+3;DEY
765     LDA(B),Y;AND@#40
770     JSR SS12
775     CLI;JMP#C55B
780:SS6  JSR#C4F6;\ Y=1
782\ CHECK IF IN DIRECT MODE
784     CPY #06;BEQ SS11+1
790\ NO ARGUMENT, SO PLACE
800\ QUEUE AT DIM-POINTER
810     LDY#24;INC#24
820\ X=2 (R.414) SO #21,X=#23
830     LDA(#21,X);EOR@#FF
840     STA(#21,X)
860     CMP(#21,X);BNE SS7
870     LDX@#0D;\ #16,X=#23
875     BNE SS2;\ ALWAYS
880:SS7
890\ NO DIMSPACE AVAILABLE:
900\ GENERATE ERROR 30
910     JMP#F11C
915
920:SS8
930\ IF ARGUMENT=0 THEN
940\ RESTORE IRQ VECTOR
960     LDX V
970     LDY V+1
990     JMP SS4
1000
1030:SS9
1040\ CHECK IRQ-SERV.ROUTINE
1050     LDX@I10%256
1060     LDY@I10/256
1070     CPX#204;BNE SS10
1080     CPY#205
1090:SS10 RTS
1095
1100:SS11
1110\ SAY HIGH PA1

```



Now all we have to do is miniaturize ourselves!

```

1710:LL6 JMP#C558
1720:LL7 STY#03
1730:LL8
1731\ IF SPO256 ISN'T SPEAKING
1732 LDA#02;BIT#BBOE
1733 BNE LL9
1734\ THEN FORCE IT TO DO SO
1735 \LDA#02;\ ALREADY SO
1736 JSR SS12;JSR SS11
1737\ ENABLE CA1 INTERRUPTS
1738 LDA#B2;STA#BBOE
1739:LL9
1740\ WAIT FOR SPACE IN QUEUE
1750 JSR#C504;\ ESCAPE
1760 LDY B+3;DEY
1770 CPY B+2;BEQ LL9
1780 LDA Z
1790 CMP#66;\ 'WAIT'?
1800 BNE LL11
1810:LL10
1820\ WAIT UNTIL QUEUE EMPTY
1830 JSR#C504;\ ESCAPE
1840 LDA B+2;CMP B+3
1850 BNE LL10;BEQ LL1
1860:LL11
1870 CMP#64;\ 'LO'?
1880 BNE LL12
1890 TAX;\ EQUALS #40
1900 LDA B+4
1905 STX B+4
1910 JMP LL14
1920:LL12 CMP#65;\ 'HI'?
1930 BNE LL13
1940 LDA B+4
1950 LDX#00;STX B+4
1970:LL13
1980\ MASK HIGH/LOW BYTE
1990 ORA B+4
2000:LL14
2010\ STORE ALLOPHONE IN QUEUE
2020 LDY B+2
2030 STA(B),Y
2040 INC B+2
2050 JMP LL1
2060:LL15 STY#03
2070 JSR#C78B
2080 DEC#04;LDX#04
2090 LDA#16,X;STA Z
2100 JMP LL7
2110:TTO;J;
2120 DATA"PA1","PA2","PA3","PA4","PA5","OY","AY","EH","KK3"
2130 DATA"PP","JH","NN1","IH","TT2","RR1","AX","MM","TT1"
2140 DATA"DH1","IY","EY","DD1","UW1","AD","AA","YY2","AE"
2150 DATA"HH1","BB1","TH","UH","UW2","AW","DD2","GG3","VV"
2160 DATA"GG1","SH","ZH","RR2","FF","KK2","KK1","ZZ","NG"
2170 DATA"LL","WW","XR","WH","YY1","CH","ER1","ER2","OW"
2180 DATA"DH2","SS","NN2","HH2","OR","AR","YR","GG2","EL","BB2"
2190 DATA"LO","HI","WAIT"
2200
2210 RESTORE

```



Don't let them see what you can do with that or it's 8 hours a day in the office! ...

Hoe was het ook al weer?

Herinner je je het verhaal over BPROM in Acorntjesbrood 3.3 nog? En was je teleurgesteld dat het programma nog niet op de regioband beschikbaar was gesteld? Tja, dat krijg je als de redactie een niet te stillen copy-honger aan de dag legt. Dan wil je de dingen wel eens te snel melden. Enfin, nu is dit verhaal in ieder geval wel verantwoord d.w.z. BPROM wordt nu wel verspreid. De beschrijving van vorige keer is nog steeds geldig op een paar dingen na:

1. het programma maakt alleen gebruik van SALFAA 1.5, dus geen gedoe met P-charge op #1XXX (of waar dan ook).
2. er zijn drie commando's bijgekomen:
 1. A, waarmee geschakeld kan worden op #BFFF, zodat je bv. snel 1 of 2 toolkits uit eprom kunt inladen.
 2. E, (exam) waarmee geheugeninhouden bekeken kunnen worden.
 3. CR, het indrukken van de returntoets maakt het scherm schoon
3. "FAST" is nu wel geïmplementeerd. Het gebruik van dit commando verlangt echter een hardwarewijziging op de programmerkaart, waarover later meer.
4. het zero page gebied is met 2 toegenomen: #80..#BF.
5. na de commando's E en I kun je, nadat je de eerste keer een adres hebt opgegeven, in plaats van het volgende antwoord + of - intikken in welk geval het volgende resp. het vorige adres wordt gespecificeerd.

Het FAST-algoritme.

Het algoritme dat in het F-commando wordt gevolgd, bestaat uit drie delen:

1. als het te programmeren byte reeds in de eprom staat, worden de punten 2 en 3 overgeslagen en is het volgende byte aan de beurt.
2. het byte wordt X keer gedurende 1 ms (milliseconde) geprogrammeerd net zolang tot de eprominhoud eraan gelijk is. Het maximaal aantal keren is 15 waarna het nog een keer op de normale manier wordt geprobeerd. Is het dan nog niet gelukt dan wordt het aldus ontstane probleem gemeld.
3. als het binnen 15 keer al gelukt was, wordt het byte gedurende 3*X ms nagebrand en het volgende byte is aan de beurt.

Als de eprom een beetje meewerkt, zijn slechts een aantal "slagen" voldoende om hem te programmeren. Stel dat dit aantal vijf bedraagt, dan wordt 4k in 40% van 3.5 minuut (minder dan 1.5 minuut) geprogrammeerd.

Wijziging van de programmerkaart.

De hardwarewijziging van de programmer is nodig omdat voor het commando "FAST" softwarematig geschakeld moet kunnen worden tussen de programmeerspanning van 21 c.q. 25 Volt en de normale spanning van de betreffende eprompin. En om het niet al te eenvoudig te maken, wordt de normale spanning van een 2732 gevormd door een laag niveau en die van een 2532 door een hoog niveau.

Achtergronden.

Hoewel in de sourcelisting van BPROM staat dat FAST via het INTEL-algoritme werkt, is dit niet helemaal waar. Ten eerste verlangt dit algoritme dat de voedingsspanning van de eprom tijdens het programmeren aan 6 Volt wordt gehangen en ten tweede wordt er gedurende 4X ms nagebrand i.p.v. 3X ms. Bovendien wordt het algoritme alleen gegarandeerd voor een beperkt aantal soorten "grotere" eproms (8k en groter). In ieder geval niet voor de 2732, laat staan de 2532. Maar als je net doet of je neus bloedt, valt er met BPROM ook voor deze kleinere types eprom wel tijdwinst te behalen.

Het schijnt overigens zo te zijn dat de 2732 van Texas Instruments niet geschikt is om een enkel byte in te programmeren (I-commando). Mede gezien mijn slechte ervaringen met de Texas 2532 moet gevreesd worden dat alle Texas-eproms interessante mogelijkheden zoals het I- en F-commando niet waard zijn.

Praktijkervaring.

Ik heb BPROM op een aantal soorten eproms losgelaten en heb er tot nu toe de volgende ervaringen mee:

- de MSL2732K van Mitsubishi reageert slecht op FAST
- de D2732D van NEC en de HN462532G van Hitachi zijn uitstekend met FAST te programmeren in ca. 45 seconden
- de 2732DC van AMD kan soms wel, vaak niet met FAST worden ge-programmeerd

Als je BPROM vertaalt zoals deze op het regio-bandje staat, wordt de voortgang van het programmeren van een eprom (P- en F-commando) bijgehouden met een teller in de linkerbovenhoek van het scherm. Per byte worden 6 karakters geprint waarbij een aantal malen op de frame sync puls wordt gewacht. Dit levert voor een 4k eprom een extra vertraging op van ongeveer anderhalve minuut.

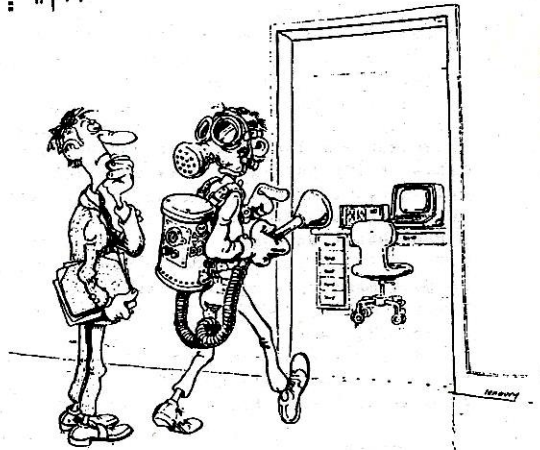
Als je dus geen hardwarewijzigingen hebt gebouwd in de vorm van een andere VDU of een turbolistschakeling, kun je het beste de boolean "hurry" (in het declaratieblok) gelijk aan 1 (TRUE) maken. De teller wordt dan niet meer afgedrukt, zodat je ook geen extra vertraging meer hebt.

Een tweede boolean is "softsw" waarmee je kunt aangeven of je je programmer wel of niet hebt omgebouwd. Als je de bovenstaande schakeling dus niet inbouwt, maak je softsw gelijk aan 0 (FALSE). In dat geval vraagt BPROM telkens als dat nodig is om de READ/PROGRAM-schakelaar in te stellen. Als je dat gedaan hebt, geef je het aan BPROM te kennen door een tik op de return-toets. Het moge duidelijk zijn dat het commando FAST dan niet meer naar behoren werkt. (N.B. BPROM is niet getest op deze manier, omdat het niet is geschreven voor een niet-omgebouwde programmer)

```

10 PROGRAM MASTERMIND
20
30 DIM DD13, BB9, CC9
40 ZERO
50 PRINT#12
60 RESTORE 570
70 FOR N=0 TO 13
80 READ DDN
90 NEXT N
100 FOR P=0 TO 13
110 U=DDP
120 FOR D=1 TO 5
130 FOR Q=0 TO 5
140 BBQ=0;CCQ=0
150 NEXT Q
160 PRINT"Gewenste uitkomst : 'U'"
170 READ A,B,C,D
180 @=2
190 PRINT"Zet           : 'A,B,C,D'"
200 @=@
210 PRINT"";GOSUBd;PRINT""
220 PRINT"Totaal       : 'T'"
230 T=@
240 NEXT Q
250 RESTORE 580
260 NEXT P
270 PRINT"READY"
280 END
290dFOR I=0 TO 5
300 FOR J=0 TO 5
310 FOR K=0 TO 5
320 FOR L=0 TO 5
330 GOSUBc
340 NEXT L
350 NEXT K
360 NEXT J
370 NEXT I
380 R.
390cBBI=BBI+1
400 BBJ=BBJ+1
410 BBK=BBK+1
420 BBL=BBL+1
430 CCA=CCA+1
440 CCB=CCB+1
450 CCC=CCC+1
460 CCD=CCD+1
470 W=0; Z=0
480 FOR M=0 TO 5
490a IF BBM=0 OR CCM=0;GOTO b
500 W=W+1;BBM=BBM-1;CCM=CCM-1;GOTO a
510b BBM=0;CCM=0
520 NEXT M
530 Z=Z+(A=I)+(B=J)+(C=K)+(D=L)
540 W=W-Z
550 IF 10*Z+W=U;T=T+1
560 R.
570 DATA 0,1,2,3,4,10,11,12,13,20,21,22,30,40
580 DATA 0,0,0,0,0,0,0,1,0,0,1,1,0,0,1,2,0,1,2,3

```



This time I'm debugging it for keeps!

De software (GDOS 1.3)

- softwarematige density selectie
- 46 files per kant
- filenamen met lengte van 15 karakters mogelijk
- alle 'oude' DOS commando's (excl. random files)
- vele nieuwe commando's
- geschreven in SALFAA 1.5
- codelengte ongeveer 3 3/4 K.
- echte foutmeldingen

Instructies GDOS 1.3

Stel, er komt snel een print en daarmee ook de software, dan wil je natuurlijk direct aan het werk. Daarom volgt hier een alfabetische opsomming van de commando's die GDOS 1.3 op dit moment biedt:

Legende: n = drive nummer (0,1,2 of 3)
<naam> = file naam, maximaal 15 karakters
t = track nummer (hex, zonder #)
s = sector nummer (hex, zonder #)
q = aantal sectoren (hex, zonder #)
l = load adres (hex, zonder #)
e = eind adres (hex, zonder #)
x = executie adres (hex, zonder #)
k = qualifier (ASCII karakter)

- *CAT n - displayed een catalogus op het scherm van de geselecteerde drive
- *COS - schakelt het cassette operating system weer in, zodat alle vectoren weer goed staan voor b.v. COS1 of FCOS
- *DIR n - haalt een directory van schijf n naar binnen op #2000-#2480
- *DD n - zet de GDOS in double-density mode en schakelt drive n voor.
- *DELETE <naam> - haalt de filenaam weg uit de directory. De DOS kent nu de file niet meer.
- *DOS - schakelt het GDOS 1.3 in, waarna alle commando's te gebruiken zijn.
- *DRIVE n - selecteerd de drive n
- *ENABLE - 'enabled' de gebruiker WIPE of FORMAT te gebruiken
- *FOLDOUT t - 'vouwt' een condensed opgeslagen track formaat van track t uit in het geheugen vanaf #2000
- *FORMAT n - formatteerd een schijf in de geschakelde density in drive nummer n
- *GO adress - te vergelijken met LINK
- *HELP - geeft op het scherm een overzicht van alle commando's inclusief hun startadres
- *INFALL n - geeft op het beeldscherm van alle files op de schijf

- *WIPE - dit commando zal, indien de faciliteit ge-'enabled' is de eerste directory wissen, d.w.z. bij een catalog opvraag zal de GDOS geen files meer vinden. Heb je er spijt van dan zal via RECOVER de tweede directory weer in de eerste geplaatst worden en men heeft weer de oude schijf (natuurlijk alleen indien deze ge-'keep'ed was).
- *WSECT t s q mem - schrijft q sectoren naar schijf vanaf het adres mem, op track t te beginnen bij sector s
- *WTRACK t - schrijft de track t, welke uitgevouwen is op adres #2000, naar schijf.
- *<naam> - autostart. File wordt binnen gehaald en gestart.

De foutmeldingen

-NAME OF FILE?

Indien de opgegeven file de 15 karakters overschrijdt of de file-naam is voorafgegaan door een dubbele quote maar hiermee niet afgesloten, dan zal deze foutmelding gegeven worden.

-SYNTAX?

Indien een commando niet beëindigd is met ';' of een 'return' zal deze melding gegeven worden.

-UNKNOWN DRIVE NUMBER

bij elk drivenummer ongelijk aan 0,1,2 of 3 zal deze melding kenbaar gemaakt worden.

-FILE PROTECTED

Een file welke 'gelocked' is, zal deze niet vervangen kunnen worden door een nieuwe met dezelfde naam of 'gedelete' kunnen worden. Bovenstaande foutmelding maakt dit duidelijk.

-FILE NOT FOUND

Een file opvragen welke niet op schijf staat of een qualifier heeft die niet overeenkomt met de huidige qualifier, wordt beloond met bovenstaande uitspraak.

-COMMAND NOT YET SUPPORTED

Het spreekt voor zich. Commando welke opgegeven is, bestaat (nog) niet.

-FATAL SEEK ERROR TRACK:t

De track t is niet gevonden. De oorzaak kan zijn: een ongeformatteerde schijf, een 'kapotte' schijf of de verkeerde density-selectie

-CRC

CRC fout. Oorzaak: kapotte sector, klok verschuivingen. Probeer de handeling nog een keer of probeer met RSECT te redden wat er te redden valt en formatteer de track opnieuw met FOLDOUT en WTRACK en save de sectors met WSECT.

ACORNTJESBRDD

```

10 REM ACORN DOS ==> GDOS 1.3
20 DIM SS(31),OO(31),A(15)
30 FOR I=0 TO 31;DIM B6;SSI=B;N.
40 FOR I=0 TO 31;DIM B15;OOI=B;N.
50 F=0;P.#12'' "ACORN DOS TO GDOS 1.3"''
60 IN."WHAT DRIVE CONTAINS"'' "THE OLD ATOM DISC"G
70 IN."WHAT DRIVE CONTAINS"'' "THE DD DISC"H
80 IF (G+H<2) OR (G+H=2 AND G=1) OR (G+H=4 AND G=2) OR (G+H>4);F=1
90 GOS.a
100 *RSECT 0 0 2 2400
110 COPY #2400,#2407,#2800;COPY #2500,#2504,#2808;?#280D=13
120 P.#12,"TITLE ACORNDISC: ""##2800"
130 Q=(?#2505)/8
140 P."NUMBER OF FILES:"Q''
150 P."FILENAME          TO BE SAVED"''
160 P."-----"''
170 FOR I=0 TO Q-1
180 COPY (#2408+(I*B)),(#240E+(I*B)),#2800
190 ?#2807=13;P.##2800,"          ";IN.#A
200 #SSI=##2800
210 #OOI=#A
220 NEXT I
230 P.#12;e=0
240 FOR I=0 TO Q-1
250 GOS.a
260 D=256*(? (#250E+(I*B))&15)+? (#250F+(I*B))
270 K=D/10;L=D%10;M=10-L
280 N=256*(? (#250D+(I*B)))+? (#250C+(I*B))
290 R=256*(? (#2509+(I*B)))+? (#2508+(I*B))
300 T=256*(? (#250B+(I*B)))+? (#250A+(I*B))
310 U=N/256+(N/256=0)+(N%256<>0)
320 DO
330 IF UKM;M=U
340 *OSCLI "RSECT",#32,&K,#32,&L,#32,&M,#32,&R,#13
350 U=U-M;K=K+1;L=0;M=10;R=R+M*256
360 UNTIL U=0
370 GOS.b
380 *OSCLI "SAVE",#34,#OOI,#34,&V,#32,&(V+N),#32,&T,#13
390 P.'#SSI," SAVED AS ",#OOI'
400 NEXT I
410 P.''' "DISC HAS BEEN COPIED"''
420 *LIB
430 END
440a*OSCLI"SD",G,#13
450 IFF;P."INSERT OLD DISC & PRESS KEY";INKEY C;P.'
460 R.
470b*OSCLI"DD",H,#13
480 IFF;P."INSERT NEW DISC & PRESS KEY";INKEY C;P.'
490 R.

```

ADVERTENTIE

~~~~~

## Te koop aangeboden:

Motorola 6845 VDU-controller  
bestemd voor bv. de Elektuur  
VDU-kaart. Prijs: ca.fl.30,=  
te bevragen bij G.B.Poot

## ADVERTENTIE:

Wegens Aquarius RAM packs  
te koop aangeboden:  
8 stuks Hitachi 6116  
fl. 10,= per stuk  
fl. 75,= in 1 koop  
te bevragen bij Bram Poot

2. De relatieve sprongen worden omgezet in absolute, zodat de offset groter kan worden dan een halve pagina. Waar nodig, dient dit verschijnsel expliciet bekeken en behandeld te worden.
3. Als ge- of misbruik wordt gemaakt van de hardwarefouten van de 6502 die in de 65C02 verbeterd zijn, geeft dat bij combinatie van beide implementaties problemen. Zoals bekend levert de instructie JMP (#XXXX) een verkeerd adres op op de 6502. Als je hiervan gebruik maakt, gaat het mis op de 65C02 die het juiste adres berekent.
4. Matthes heeft twee series macro's gebouwd. De eerste dient er, zoals gezegd, voor om de 65C02-instructies op een 6502 te simuleren; de tweede serie echter maakt een echte 65C02-code aan. Je kunt dus op je Atom een programma voor de 65C02 schrijven en testen m.b.v. de eerste serie macro's en als alles naar tevredenheid draait m.b.v. de tweede serie macro's de definitieve code genereren die dan, als het goed is (laat ik voorzichtig zijn), op je alternatieve systeem (bv. Elektaur CPU-kaart met R65C02) zonder meer geïmplementeerd kan worden.

De macro-implementatie van SALFAA maakt gebruik van het feit dat door het aanroepen van een P-charge FUNCTION de aldaar gedefinieerde instructies worden uitgevoerd. Aangezien deze instructies kunnen bestaan uit met basic gemengde machinetaalinstructies verkrijgen we het effect van de macro. Omdat SALFAA eveneens met basic mengbaar is en tevens het "statement" ASM-C (continue) bevat, kunnen we met minimale overhead macro's definiëren met alle FUNCTION-faciliteiten zoals daar zijn: locale parameters, recursie, nesting en, tot op zekere hoogte, tijdelijke of locale symbol tables. Daar SALFAA, waar mogelijk, compatible is gehouden met de Atom assembler, kun je in de macro's vaak i.p.v. de combinatie ASM-C en .END het hakenpaar (L J) gebruiken. Soms, maar lang niet altijd, is dat prettiger.

Als je een FUNCTION aanroept, wordt in principe een resultaat aan de functienaam gehecht. SALFAA maakt hier geen gebruik van, zodat de instructie "naam = expressie" in het functieblok ontbreekt. Intern wordt wel degelijk een waarde opgeleverd, maar die verdwijnt "ins Blaue hinein" en daar heb je dus niets meer mee te maken. Zou je echter wel een waarde aan het oproepende programma willen overdragen via de functienaam, dan kan dat door in het SALFAA-statusbyte bit2 hoog te zetten bv. met .OPTION :0100 0100. SALFAA zorgt er dan voor dat de waarde die je in de functie aan de functienaam hebt gegeven, wordt gekopieerd in de basic-variabele M (van Macro), zodat deze voor later gebruik is terug te vinden.

Voor een goed gebruik van de SALFAA-macro dien je te beseffen dat de assemblermode door de macro-aanroep wordt veranderd in basicmode. Tijdens de FUNCTION mag je dit naar believen veranderen, maar ten tijde van FEND moet de mode weer basic zijn.

Wat is nu de ZIN van deze macro's?

Wel, de zin van macro's in het algemeen is dat programma's (en daar gaan we weer) gestructureerder worden. Dus: beter leesbaar,

## WEDSTRIJD

De Redactie

Tot haar grote teleurstelling heeft de redactie slechts twee inzendingen (en wel een voor wedstrijd A en een voor wedstrijd B) ontvangen op de Reversi wedstrijd uit het vorige Broodje. Hiervan is er een (1), althans volgens de inzender ervan, ook nog buiten mededinging. Kortom: een minimale respons dus !! Het is ons niet bekend of dit te wijten is aan de vakantieperiode, of aan een andere oorzaak. Laten we daarom maar snel overgaan tot het bespreken van de twee oplossingen.

De oplossing voor wedstrijd A is afkomstig van Frits Spengelink en is inderdaad verrassend kort. In plaats van de computer te laten beginnen heeft Frits het heft in eigen hand genomen door zelf de eerste zet te doen. Hier volgt zijn oplossing:

volgorde: zet, Frits, Reversi.SEW

- 1 D3 - C3
- 2 B3 - E3
- 3 F3 - F4
- 4 F5 - F6
- 5 G7 - PAS
- 6 PAS

Het resultaat is dus 13 - 0. Hoe de stenen dan op het bord liggen illustreert onderstaand plaatje.

**SCORE 0-13 !**  
**U HEEFT GEWONNEN**

|   | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 |   |   |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |   |   |
| 3 |   | ○ | ○ | ○ | ○ | ○ |   |   |
| 4 |   |   |   | ○ | ○ | ○ |   |   |
| 5 |   |   |   | ○ | ○ | ○ |   |   |
| 6 |   |   |   |   |   | ○ |   |   |
| 7 |   |   |   |   |   |   | ○ |   |
| 8 |   |   |   |   |   |   |   |   |

**PIET HEIN**  
**NIVEAU: 1**

- |   |    |     |
|---|----|-----|
| 1 | D3 | C3  |
| 2 | B3 | E3  |
| 3 | F3 | F4  |
| 4 | F5 | F6  |
| 5 | G7 | PAS |

Daar dit de enige ontvangen oplossing is voor wedstrijd A krijgt Frits dit Broodje dus gratis.

Voor wedstrijd B ontvingen wij een oplossing van Bram Poot, naa zijn eigen zeggen buiten mededinging. Deze oplossing speelt zich af op niveau 6. Voor het spelverloop laten wij nu Bram zelf aan het woord.

- 13 H5.. het is me alleen de eerste keer overkomen dat hij hier 13 F7.. deed. Hij werkt niet met RND dus hoe dat kan is me een raadsel. 13 H5.. dus, ook goed. "-2" geeft aan dat hij vindt dat hij er niet goed voor staat
- 13 ..F7 gat opvullen
- 14 E2.. hij moet hier lang over nadenken en veel anders is er ook niet
- 14 ..H4 pak maar op 15 H3.. dan doe ik 15 ..G3
- 15 D8.. 15 ..E8 kan ik niet. Slim
- 15 ..H3 pak maar op 16 H2.. dan doe ik 16 ..G3
- 16 G3.. het veld direct naast een hoek (H2) is inderdaad niet erg gunstig
- 16 ..H7 toch pak ik hier zo'n veld om zijn aantal mogelijke zetten klein te houden

Links zit voor hem helemaal dicht. Houden zo.

- 17 E8.. wat moet hij anders
- 17 ..D1 aanval op de bovenrand
- 18 E1.. aanval op C1, maar....
- 18 ..G1! nu maak ik het hem moeilijk
- 19 G8.. "-13", het gaat goed
- 19 ..D2 hou hem kort
- 20 G7.. kat in 't bakkie: "-18"

Hoewel de diagonaal helemaal zwart is, kost het me geen moeite om ergens een steen wit te maken, zodat ik op H8 kan spelen. Na 20 ..B4 en 20 ..B5 kan hij de witte weer zwart maken, echter niet na 20 ..B6 dus...

20 ..B6

Dat ik het veld links opengooi, is nu niet meer van belang.

- 21 C1.. uitstel
- 21 ..B1 weinig keus
- 22 A5.. "-16"
- 22 ..H8 ik ga zijn mogelijkheden weer beperken
- 23 B4.. "-24", het gaat goed
- 23 ..B8 er rest hem 1 zet: B5
- 24 B5.. zie je?
- 24 ..H2 nu moet hij op B7 of G2 zetten
- 25 G2.. "-24"
- 25 ..A3 om F3 wit te maken
- 26 A4.. "-37"!
- 26 ..H1 de randen boven, rechts en onder kunnen niet meer van kleur veranderen. 20 punten al "-37", het maakt allemaal niet meer uit
- 27 C2.. minste risico
- 27 ..A6
- 28 A7.. tja
- 28 ..B2 niet te happig op A8
- 29 B7.. 29 A2.. misschien toch iets beter
- 29 ..A8 nu dan
- 30 A2.. wat is de waardering nu nog waard?
- 30 ..A1 zo doe je dat dus: 17-47

Met zeer hartelijke groeten,



Bram Poot



SOFTWARE: Direct mode statement voor P-charme Theo den Exter

P-CHARME gebruikers bezitten nu al een jaartje een commando waarmee P-CHARME functies in direct mode kunnen worden uitgevoerd, nadat een programma minstens 1 keer gerund heeft. Hierbij was het niet mogelijk om ook PROCEDURES vanuit direct mode aan te roepen. Er kleefde volgens Frans van Hoesel enige bezwaren aan het gebruik van dit statement. Zo mag een programma niet ondertussen worden ge-edit, omdat P-CHARME dan de functie niet meer kan vinden.

Wat schetste mijn verbazing toen ik bemerkte dat ook PROCEDUREES met dit commando vanuit direct mode geexecuteerd konden worden, en wel op de volgende vreemde manier:

-P. procedurenaam

Het "-" commando, welke normaal alleen gebruikt wordt om functies te activeren (-P. functienaam) werkte in combinatie met PRINT ook uitstekend als executeur van PROCEDURES. Het PRINT statement is hierbij volkomen dummy. Het activeert de procedure, maar drukt nooit (!!!!) wat af. Een niet zo fraai foutje eigenlijk, HOEWEL?

In het onderstaande programma is dit fraai in een nieuw direct mode commando omgezet, welke uitstekend procedures kan uitvoeren. Waarom het precies werkt? Ik weet het niet, maar handig is het in ieder geval wel. "-" heeft dus een nieuw broertje!

Een procedure kan aangeroepen worden middels ": procedurenaam". Aan het gebruik kleven waarschijnlijk dezelfde door van Hoesel bedoelde bezwaren, wees dus voorzichtig in gebruik.

```

10 PROGRAM DIR-PROC.STAT
20
30 REM *****
40 REM * STATEMENT FOR *
50 REM * DIRECT MODE: ':' *
60 REM *AUT.: TH. DEN EXTER*
70 REM * DATUM: 13-06-1985 *
80 REM *****
90
100 REM *****
110 T=#3000;A=#3100
120 !T=#80C6E3FF
130 T=T+3
140 REM *****
150
160 DIM LL1
170 FOR I=0 TO 1;LLI=#FFFF;N.
180 P.#21;GOS.a
190 P.#6;GOS.a
200
210 $T="";T=T+LEN T
220 ?T=LL0/256!#80
230 T?1=LL0%256
240 T?2=#80
250 T=T+2;A=P
260 ?#3FC=#30
270
280 PROC DEMO
290 P."DEMO RUNNING"
300 PEND
310
320 P."VOER IN DIRECT MODE IN:"
330 :DEMO
340 END
350
360a P=A
370I
380:LL0
390 \ DIR MODE ?
400 LDA 6
410 CMP @#01
420 \ NO DIR MODE !
430 BNE LL1
440 JSR #A42E\ P-CH BOX
450 JMP #C334\ PRINT STMNT
460:LL1
470 \ DIR MODE !
480 \ ERROR 94, PLEASE !
490 JMP #C558
500I
510 RETURN

```

```

260bFOR I=(W-W)TO(W-1)
270 P.#I?T
280 NEXT I
290 F=0
300cIF W>=60;GOTO d
310 GOTO e
320aP.#3#6#12'"DONE.'"';END
330DATA 60,1,12,26,9,12,3,8,24,17,8,4,6,23,21,6,4,6,22,12,5,6,5
340DATA 4,6,21,11,8,6,4,4,6,21,10,10,5,4,4,6,21,9,11,5,4
350DATA 4,6,21,8,11,6,4,4,6,21,7,11,7,4,4,6,21,6,11,8,4
360DATA 4,6,19,1,1,5,11,9,4,4,6,19,1,1,5,10,10,4,4,6,18
370DATA 2,1,6,8,11,4
380DATA 4,6,17,3,1,7,5,13,4,4,6,15,5,2,23,5,1,29,5,17,8
390DATA 1,29,9,9,12,1,13,5,40,1,1,13,5,40,1,4,6,13,3,10,6,12
400DATA 5,1
410DATA 5,6,11,3,11,6,14,3,1,5,6,11,3,11,6,15,2,1
420DATA 6,6,9,3,12,6,16,1,1,6,6,9,3,12,6,7,1,10
430DATA 7,6,7,3,13,6,6,2,10,7,6,7,3,13,14,10,8,6,5,3,14,6,6
440DATA 2,10
450DATA 8,6,5,3,14,6,7,1,10,9,6,3,3,15,6,16,1,1
460DATA 9,6,3,3,15,6,15,2,1,10,6,1,3,16,6,14,3,1,10,10,16,6
470DATA 12,5,1
480DATA 11,8,13,27,1,11,8,13,27,1,60

```

```

10 REM MERLIN'S MAZE
20 REM ORIG:APPLE II, S.A. BAKER
30 REM ADAPTED FOR ACORN ATOM
40 REM BY GERRIT HILLEBRAND
50
60 DIM K(2)
70 P.#12;VTAB 7;HTAB 8;P."merlins"#128"maze"
80 P."DO YOU WANT INSTRUCTIONS ";INKEY A
90 IF A=CH"Y";GOSUB t
100 S=123;L=1;@=3
110dREM:VUL SPEELVELD
120 H=#B200;FOR I=1TO100;I?H=0;NEXTI
130 FOR I=1TO25;?(H+1+A.R.%100)=1;NEXTI
140 FOR I=1TO5;?(H+1+A.R.%100)=2;NEXTI
150 X=1+A.R.%10;Y=1+A.R.%10
160 FOR I=1TO2;?(H+1+A.R.%100)=3;NEXTI
170a0=2+A.R.%8;P=2+A.R.%8;IF?((P-1)*10+0+H)<>0 GOTO a
180
190 CLEAR0;P.#30;?#E1=0
200 HTAB 3;FOR I=1TO21;P."=";NEXT;P.
210 FOR I=1TO10;HTAB 3;P."!"
220 HTAB 3;FOR I=1TO21;P."=";NEXTI
230 VTAB 13;P."LEVEL "L
240 T=2;V=2
250pVTAB Y;HTAB X*2+2;P."*"
260 I=1
270bG=A.(0-X)*A.(0-X)+A.(P-Y)*A.(P-Y);Z=I*I
280 IF Z>6 GOTO c
290 I=I+1;IF I<21 GOTO b
300cVTAB 14;P."DISTANCE TO EXIT "I-1" UNITS"
310 P."MOVES LEFT : "S
320 A=X;B=Y
330 IF X=0 AND Y=P GOTO e
340 INKEY K
350 VTAB 12;P."
360 VTAB Y;HTAB X*2+2;P." "
370 IF K=CH"U" Y=Y-1
380 IF K=CH"D" Y=Y+1
390 IF K=CH"R" X=X+1

```

SOFTWARE: Revised TOOLBUG

Bram Foot

Inleiding.

De Toolbug is een softwareuitbreiding voor de Atom geschreven door de engelse firma PSION. Deze biedt de gebruiker een aantal programmeerhulpmiddelen op twee verschillende en van elkaar losstaande terreinen. Ten eerste bevat de Toolbug een basic-editor (EDIT) en ten tweede een machinetaal-debugger (DEBUG). Een beschrijving annex handleiding van de Toolbug is verschenen in AcornNieuws 3.1 p.53 e.v., zodat ik hier niet hoeft uit te weiden over de aanwezige mogelijkheden.

Dat de Toolbug weinig bekendheid geniet, vindt z'n oorzaak in 2 redenen:

1. de utilities zijn bedoeld voor direct mode en worden dus niet in programma's gebruikt c.q. verspreid (over RENUM en FIND hoor je ook nooit iets).
2. er zitten enkele irritante foutjes in, waardoor sommigen hem al gauw terzijde hebben gelegd.

Ik bedacht me, dat als ik de fouten zou verbeteren, ik er een op sommige punten redelijk geavanceerd softwarepakket bij zou hebben. Ik heb dus de Toolbug en z'n "dislist" uit de kast gepakt en de eerste weer in de schakelkaart geprikt. De hernieuwde belangstelling heeft het volgende opgeleverd.

Ongedocumenteerde instructies.

Hiervan is er slechts 1 en heeft de vorm EDn, waarmee je n regels vanaf de "current line" kunt "extracten" en tegelijkertijd "deleten".

De commandostring L.200^E4^L.200^D4 kan dus verkort worden tot L.200^ED4.

Helaas bestaat de instructie EDM (nog?) niet.

Fouten en hun verbeteringen.

1. De Toolbug-interpreter herkende het commando EDITOR ook als EDIT. Hij testte niet op het einde van het commando. De commando's kunnen nu bovendien niet meer in een programma worden aangeroepen.
2. Het commando L.n waarbij n een niet-bestaand regelnummer is, listte een ogenschijnlijk willekeurig (groot) aantal regels van het programma. De nieuwe versie volgt de beschrijving in de handleiding.
3. Het commando DM wiste precies 1 karakter te weinig in het met M1 en M2 gedefinieerde gebied. Uit de dislist bleek overduidelijk dat er waar een gedachtenfout was gemaakt.
4. In de dump-routine van DEBUG werden de 7-bits ASCII-karakters groter dan #1F afgedrukt INCLUSIEF de DEL (#7F). Een bekende en veelgemaakte fout.

En verder is M nogal gewijzigd:

1. als de inhoud van een byte een afdrukbaar ASCII-karakter bevat, wordt dat ook als zodanig kenbaar gemaakt.
2. de na M te geven antwoorden spatie en return zijn vervangen door resp. return en Q (van quit). Dat werkt prettiger.
3. als extra antwoord na M is de ^ geïntroduceerd, waarmee je achterwaarts het geheugen kunt doorlopen.
4. als je de inhoud van een adres verandert, wordt ter verificatie dit adres met z'n huidige inhoud nogmaals afgedrukt.

Zero page gebruik.

De opmerking dat de Toolbug gebruik maakt van de zero page adressen #08 t.e.m. #0C en #70 t.e.m. #7F is helemaal juist (vreemd dat je daar tegenwoordig niet meer zonder meer vanuit gaat). De indeling staat er echter niet bij en zal ik dus hier geven.

#### EDX

```

#08      delimiter van strings in F, C en S
#09      "herhalingsbyte"
#0A,#0B  TDP van de buffer
#0C      high byte van startadres van de buffer
#72,#73  adres van "current line"
#7A,#7B  waarde van decimale parameter
#7C,#7D  adres van M1
#7E,#7F  adres van M2

```

#### DEBUG

```

#08      FORMAT-parameter
#0A,#0B  adres van break point
#0C      originele inhoud van break point
#72,#73  referentie adres (commando W)
#79      opslag van stack pointer
#7A,#7B  eindadres in dump en crack (nu: list)
#7C      A-image
#7D      X-image
#7E      Y-image
#7F      P-image

```

De niet vermelde adressen worden voor tijdelijke pointers en opslag gebruikt.

Tot slot.

Ik kan me goed voorstellen dat er mensen zijn die zeggen: "De Toolbug is wel leuk, maar....." en vult u de cliché's maar in. Bedenk dan wel dat echt interessante mogelijkheden, vooral als daar wat intelligentie bij komt kijken, niet in een 4k eprom passen. De Toolbug levert, mede gezien z'n lengte, heel aardige faciliteiten waarvan EXTRACT/INSERT de meest waardevolle is naast z'n niet-overschrijvende-AUTO en z'n niet-meer-dan-nodig-RENUMBER.

En met een beetje schakelsoft vallen de Toolbug en de P-charmeditor heel goed te combineren, zodat ze elkaar uitstekend kunnen aanvullen.

=====  
scientists manage to get degrees without ever learning FORTRAN!  
Are we destined to become an industry of Unix hackers and  
Pascal programmers?

From my experience, I think it's safe to report that the future is bright for Real Programmers. Neither OS/30 nor Fortran shows any sign of dying out, despite the efforts of Pascal programmers. Even more subtle tricks, like adding structured coding constructs a Fortran 77 compilers, but every one of them has a way of converting itself back into a Fortran 66 compiler at the drop of an option card-to compile DO loops as God intended.

Even Unix might not be as bad on Real Programmers as it once was. The latest release of Unix has the potential of an operating system worthy of any Real Programmer. It has two different and subtly incompatible user interfaces, an arcane and complicated teletype driver and virtual memory. If you ignore the fact that it's structured, even C programming can be appreciated by the Real Programmer. After all, there's no type checking, variable names are seven (ten? eight?) characters long, and the added bonus of the pointer data type is thrown in. That's like having the best parts of FORTRAN and assembly language in one place, not to mention some of the more creative uses for # define. No, the future isn't all that bad. Why, in the past few years, the popular press has even commented on the bright new crop of computer nerds and hackers leaving places like Stanford and MIT for the Real World. From all evidence, the spirit of Real Programming lives on in these young men and women. As long as there are ill-defined goals, bizarre bugs, and unrealistic schedules, there will be Real Programmers willing to jump in and solve the problem, saving the documentation for later. Long live Fortran!!!

Ed Post  
Wilsonville, Oregon.

Dit was dan helaas weer het laatste deel van Real Programmers Don't Use Pascal. Herhaalde malen ben ik zekere thema's uit Ed's betoog in Real Live tegen gekomen. Bovendien past het begrip The Real Programmer het beste bij de Atom-man. Iemand met zo'n computer moet wel een echte Real Programmer zijn. Ik denk daarom dat iedereen er wel iets van in zich herkend heeft. Ongetwijfeld zal Ed's betoog niet voor iedereen toegankelijk zijn geweest. Ten eerste is het geschreven in het Engels, maar daarnaast dien je ook nog eens over een behoorlijke dosis (ouderwetse) kennis te bezitten. De redactie hoopt dat nog meer leuke stukken boven water zullen komen, zodat dit 2 maandelijks terugkerende hoofdstuk nog zijn vervolg zal kunnen hebben.

Ik heb toen de zes controllijnen (Q2, R/NW, NBLKO, NRST, NRDS & NWDS) ook met een 74LS245 (fan out van 30, zo moet dat) gebufferd en vervolgens het systeem de vuurproef gegeven. Pattern test heeft gedurende tweeneenhalf etmaal bijna 4.5 Gbytes (om precies te zijn: 4718592000) behandeld (schrijven en lezen) zonder ook maar 1 fout te ontdekken. Het gebied dat hiervoor in aanmerking kwam, liep van #3FF totaan #A3FF (40k), zodat ook afwisselend binnen en buiten de bus werd beschouwd.

Mijn conclusies en ervaringen uit deze ellende-periode zijn de volgende:

- buffering op de juiste plaatsen is zeer belangrijk
- kabellengte is niet kritisch; zodra ik in de gelegenheid ben, ga ik 100 cm proberen (uiteraard onafgeschermd)
- ik heb geen scoop, dus ik kan het niet verifiëren, maar de signalen op PL6 zien er waarschijnlijk erg beroerd uit
- pattern test is bijzonder rigoreus; als dit programma geen fouten meldt, ben je er zeker van (voor zover dat ooit mogelijk is) dat het behandelde geheugen in orde is.

Doe uw voordeel met dit verhaal. De periode die ik nu achter de rug heb, wens ik geen enkele seriële Atom-gebruiker toe.

SOFTWARE: Foutje in GRMOD

Theo den Exter

Er zit helaas nog een (?) storend foutje in GRMOD. Als je PRINT \$0 probeert uit te voeren, dan zal de ATOM niet naar behoren functioneren. Wat er precies gebeurt is erg afhankelijk van de staat waarin het geheugen zich bevindt. Diegene die een 16K kaart vanaf #4000 bezitten, zullen merken dat de ATOM dan doorgaans "vast" loopt. Diegenen die daar geen geheugen hebben, zullen merken dat GRMOD gewoon zij weg vervolgt, nadat op ESC is gedrukt. Wat is er nu precies aan de hand?

De outchar routine van GRMOD, welke op #AD6E is gelocaliseerd zal onder andere eventuele controlcodes proberen te onderscheppen. Dat gebeurt in een routine vanaf #ACAB, welke hieronder is afgedrukt. Daarin wordt de subroutine op #FEC5 aangeroepen, die de controlcode probeert op te zoeken in een tabel die vanaf #FECB staat. Dat lukt, en deze routine levert als resultaat een index X=0 op, welke de onderhavige controlcode vanaf #FECB geïndiceerd aanwijst ( $(\#FECB + 0) = 0$ ). Daarna gaat de JOSBOX in de fout, want zij meent hier een apart startadres voor te hebben. In de tabel vanaf #AF02 wordt het startadres opgezocht van de uitvoer routine van controlcode 0. Omdat X=0 en deze tabel geïndiceerd wordt vanaf #AF01, zal op #AF01 naar het startadres worden gezocht. Dit ligt net buiten de tabel die pas geldig is vanaf #AF02, dus voor een X waarde groter dan 0. Hierdoor wordt #40 in de accu geladen (een RTS van een vlak ervoor staande routine) en op de stack gezet als high-order adres. Daarna wordt in de tabel naar het low-order adres gezocht en ook op de stack gezet (#AC). Middels een RTS wordt de routine nu aangeroepen. Er wordt op deze wijze eigenlijk gesprongen naar #40AD.

INTIKKEN EN RUNNEN MAAR

DOP.#C.#(C.&gt;31);U.C.=0

**hoezo, onleesbaar?**

```

10 REM hoezo, onleesbaar?
20 Q=#2800; P=Q; X=#80
30 ILDY@X; STY X+1
40 LDX@0; STX X
50 STA (X,X)
60 INC X \ niet INX!
70 BNE P-4
80 RTS
90 INPUT "een getal graag" A
100 LINK Q
110 END

```

```

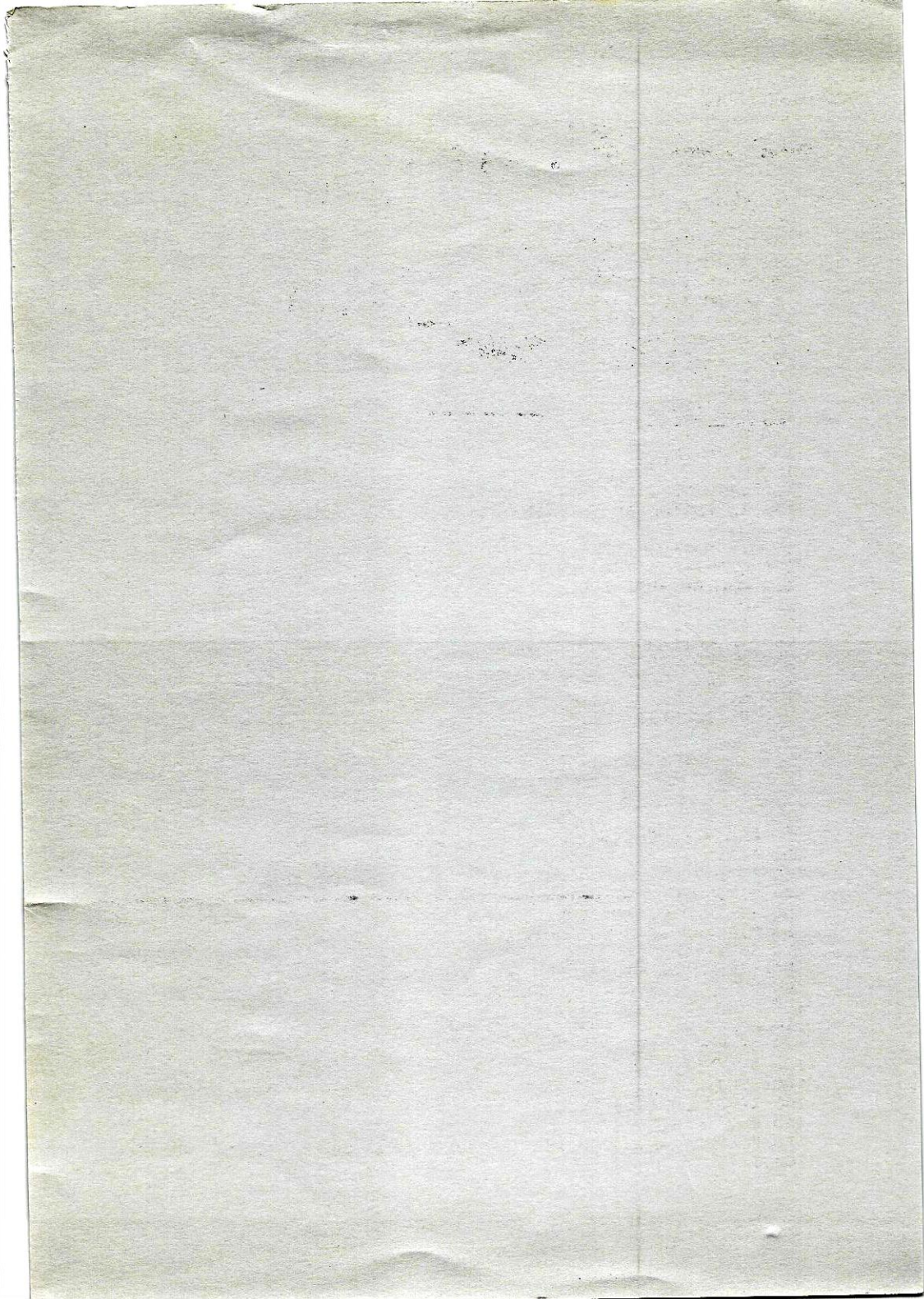
10 REM 3D GRAPHIC
20 ZZ=90; CLEAR4; F.N=0T0180
30 %S=2*RADN; %R=%Z*SIN(%S*2)
40 X=%(%R*SIN%S+128); Y=%(%R*COS%S+96)
50 IF N=0 THEN MOVE X,Y
60 DRAW X,Y; %R=%Z*COS(%S*2)
70 X=%(%R*COS%S+128); Y=%(%R*SIN%S+96)
80 MOVE X,Y; N.; F.N=0T0200; WAIT; N.; Z=50; CLEAR4; F.N=0T0180
100 %S=2*RADN; %R=%Z*SIN(%S*2)
110 X=%(%R*COS%S+100); Y=%(%R*SIN%S+140)
120 IF N=0 THEN MOVE X,Y
130 DRAW X,Y; %R=%Z*(SIN%S*2-COS%S*2)
140 X=%(%R*SIN%S+100); Y=%(%R*COS%S+140)
150 MOVE X,Y; N.; F.N=0T0200; WAIT; N.; E.

```

## INFOKAARTEN:

Bram Poot &amp; Theo den Exter

Ook deze maand zijn er weer infokaarten waarmee U uw voordeel kunt doen. Als eerste een kaart met de aansluitingen van de meest gebruikte IC's. Daarnaast is er nog een handige infokaart voor spraak-ic (SP0256) gebruikers. Als laatste is een handige originele ACORN ATOM library sheet bijgevoegd. Deze kunt U gebruiken om Uw programma tapes netjes op te administreren. We wensen U er veel plezier mee.





Acornntjesbrood

Acornntjesbrood

*Acornntjesbrood*

**Als u het een half jaar  
niet leest bent u  
al verouderd in uw vak.**

```

10 PROGRAM PI
20 F=1000; H=200; @=4; ?#E1=0
30 IN."HOEVEEL DECIMALEN (<1000) "M
40 M=(M+1)/3+1; DIM BBM
50 D=0; F.I=1TDM; BBI=1; N.
60 @=%(M*3*LOG10/LOG2)-1
70 F.N=D TO 1 STEP -1
80 BBI=BB1+H; P=2*N+1
90 F.I=1TDM
100 D=D+F+BB1*N
110 BBI=D/P
120 D=D-BBI*P
130 N.
140 D=0; P.N
150 N.
160 F.I=M TO 1 STEP -1
170 IF BBI>=F; BB(I-1)=BB(I-1)+1; BBI=BBI-F
180 N.
190 IF M>160 P."ZET DE PRINTER AAN" #2#15
200 P."Hier volgt het getal PI met "3*M-1" decimalen""
210 @=0; BB1=BB1-100
220 P."3."BB1
230 FOR I=2 TO M
240 IF BBI<100 P."0"
250 IF BBI<10 P."0"
260 P.BBI
270 N.
280 P."#3; ?#E1=12B; E.

```

```

10 REM faculteiten
20 P.#12"*****
30 P."DIT PROGRAMMA BEREKEND FACULTEITEN OOK > 33""
40 FIN."GEEF HET GETAL WAARVAN DE ! BE-PAALD MOET WORDEN" %X
50 FIF%X<0 P.#12#7"GETALLEN <0 ZIJN NIET TOEGESTAAN""; G.40
60 FIF%X=0P.#12"HET ANTWOORD IS 1""; G.40
70 X=%X; %I=%X-X
80 FIF%X>0P.#12"ALLEEN GEHELE GETALLEN !"" #7; G.40
90 P.""" ==== EVEN GEDULD A.U.B. ==== "#30
100 %Y=0
110 DD
120 %Y=LOG%X+%Y
130 %X=%X-1
140 FUNTIL %X=0
150 %Z=%Y/LOG10 ; F=%Z
160 @=0 ; P.#12#7
170 FP."HET ANTWOORD IS"" EXP(LOG10*(%Z-F))"E"; P.F""
180 G.40

```

```

ACAB: C9 07      CMP @#07
ACAD: F0 9E      BEQ #AC4D
ACAF: A2 09      LDX @#09
ACB1: 20 C5 FE   JSR #FEC5
ACB4: D0 31      BNE #ACE7
ACB6: BD 01 AF   LDA #AF01,X
ACB9: 4B         PHA
ACBA: BD 0A AF   LDA #AFOA,X
ACBD: 4B         PHA
ACBE: 60         RTS

```

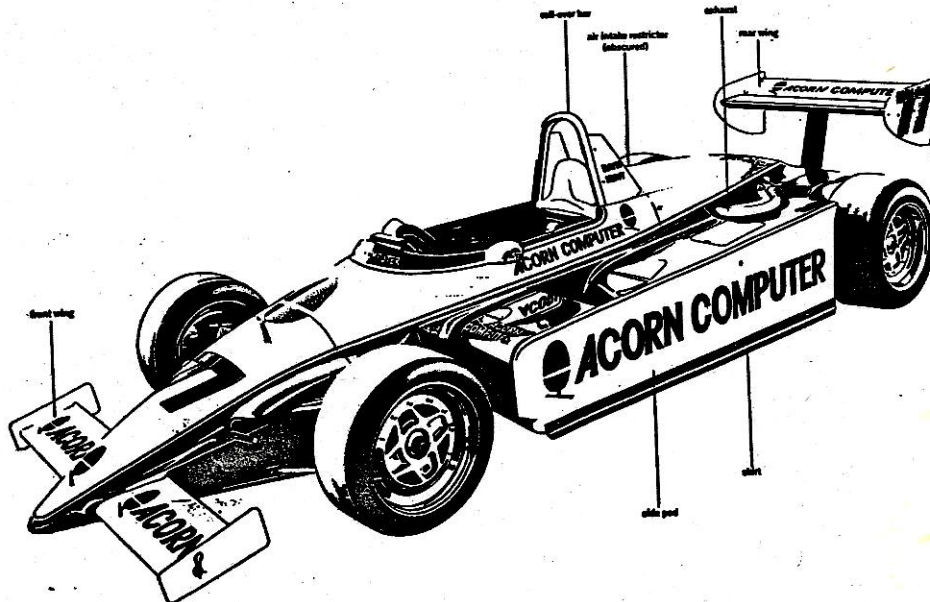
Dat kan als volgt aannemelijk gemaakt worden:

```

10 P=#40AD
20 JSR #FD1A
30 LDA @#33
40 JMP #FFF4
50 J
60 GRMOD
70 P.$0
80 END

```

Diegenen die geen geheugen vanaf #4000 hebben zullen merken dat GRMOD vastloopt totdat op ESC wordt gedrukt. De "routine" vanaf #40AD zal namelijk #40 = RTS opleveren, welke zo weer terugspringt naar GROMOD met bovenstaande als resultaat. Bezitters van een geheugenkaart vanaf #4000 kunnen een sprong verwachten naar #40AD met allerlei mogelijke resultaten tot gevolg. Hiermee is dit PRINT \$0 raadsel ook weer opgelost.



BUSWARE:Hoe hoort het nu eigenlijk WEL?

Bram Poot

Al zeer geruime tijd heb ik achter mijn Atom een rack aangesloten met daarin als belangrijkste kaarten de VDU- en geheugenkaart van Elektuur. Lange tijd verkeerde ik in de veronderstelling dat het geheel feilloos werkte totdat ik het lage tekstgeheugen uit de Atom verwijderde t.b.v. een zelf m.b.v. Aquarius RAM-packs gefabriceerde 32k kaart (#0000..#7FFF). Niet alleen bleek dat deze kaart niet goed functioneerde (getest met Pattern Test, zie AcornNieuws 3.1 p.28), maar ook de 2k VDU RAM (80-kolommenkaart) bleek niet in orde te zijn, hoewel ik daar bij het normale gebruik nooit wat van had gemerkt. Bovendien bleek de afscherming die ik rond de verbindingkabel had aangebracht zeer essentieel te zijn, hoewel niet voldoende. Het systeem was dus volkomen onbetrouwbaar. Dat de hardware in principe goed was, werd bewezen door het RAM-test programma dat in het handboek staat. Dit programma kon geen fouten ontdekken.

Dat de afscherming van de kabel voor minder fouten zorgde wees erop dat de signalen last hadden van een of ander stoorveld. De suggestie die op de federatieve vergadering begin '85 werd gedaan om de kabeladers om en om aan massa te leggen ten einde overspraak tussen de signalen te voorkomen, leek dan ook plausibel genoeg om dit eens te proberen. Helaas, geen of hooguit nauwelijks resultaat.

Op dezelfde vergadering werd een tweede oorzaak van storingen genoemd, zijnde reflecties tegen de "achterkant" van de bus. Om deze te voorkomen moet je aan het eind van de bus voor elk signaal een "opsloppschakelingetje" bouwen. Niemand schijnt je echter te kunnen vertellen hoe zoiets dan wel in zijn werk gaat. Bovendien hoor je dan van semi-professionals dat reflecties überhaupt flauwekul zijn bij de huidige Atom snelheden, maar dat storingen heel makkelijk worden veroorzaakt door aardlussen. Inderdaad had ik zo'n aardlus, maar na het doorbreken ervan bleef de puinhoop zoals hij was.

De inmiddels tot 20 cm ingekorte kabel wenste ik niet nog korter te maken, omdat ik dan net zo goed de Atom aan het rack kon vastplakken.

Begrijp me goed: ik bestrijd de gefundeerdheid van al deze suggesties op geen enkele wijze (ik ben maar een leek op dit gebied). Ik heb slechts de suggesties op mijn configuratie toegepast en helaas moeten constateren dat ze stuk voor stuk geen verbeteringen gaven.

Uiteindelijk ben ik dan maar na al dit geklungel mijn eigen eenzame weg gegaan en heb de databus direct achter de Atom, dus vóórdat deze de kabel ingaat, gebufferd. Bij de aankondiging van de eerste serie geheugenkaarten stond dat buffering erg belangrijk is, en ik weet nu wat die opmerking waard is, want ziet: pattern test gaf geen foutmeldingen meer!

Ik heb toen onmiddellijk de kabel verlengd tot 50 cm (nog steeds onafgeschermd) waarna de fouten weer optraden. Zonder direct in paniek te raken, bedacht ik me dat, meer nog dan de databus, de controlbus gebufferd moet worden. Het schrijpendste voorbeeld is wel de NWDS. Deze komt uit een 74LS139 (niet eens een driver, fan out van 1 (hooguit)) en wordt bij mij viervoudig, in sommige Atoms twintigvoudig belast voordat hij op PL6 beschikbaar is.

## TER LEARNING ENDE VERMAEK: REAL PROGRAMMERS DON'T USE PASCAL

REAL PROGRAMMERS DON'T USE PASCAL  
LAST EPISODE

What sort of environments does the Real Programmer function best in? This is an important question for the managers of Real Programmers. Considering the amount of money it costs to keep one on the staff, it's best to put him or her in optimal environment.

The typical Real Programmer lives in front of a computer terminal. Surrounding this terminal are the listings of every program he has ever worked on. These are piled in roughly chronological order on every flat surface in the office. You will also find some half dozen or so partly filled cups of cold coffee. In some cases, the cups will contain Orange Crush. And, unless he is very good, there will be copies of the OS JCL manual and the Principles of Operation open to some particularly interesting pages. Taped to the wall is a line-printer Snoopy calendar for the year 1985. Strwn about the floot there will be several wrappers for peanut butterfilled cheese bars (the type that are made stale at the bakery so they can't get any worse while waiting in the vending machine). Finally, in the top left-hand desk drawer, underneath the box of Oreos is a flowcharting template, left there by the previous occupant. Real Programmers write programs, not documentaion, which is left to the maintenance people.

The Real Programmer is capable of working 30, 40, even 50 hours at a stretch, under intense pressure. In fact, he prefers it that way. Bad response time doesn't bother the Real Programmer; it gives him a chance to catch a little sleep between compiles. If there is not enough schedule pressure on the Real Programmer, he tends to make things more challenging by working on some small but interesting part of the problem for the first nine weeks. Then he finishes the task in the last week, in two or three 50-hour marathons. This not only impresses his manager, but creates a convenient excuse for not doing the documentation. In general: no Real Programmer works 9 to 5, except those on the night shift.

Real Programmers don't wear neckties. Real Programmers don't wea high-heeled shoes. Real programmers arrive at work in time for lunch. A Real Programmer may or may not know his spouse's name. He does, however know the entire ASCII (or EBCDIC) code table. Real Programmers don't know how to cook. Crocery stores aren't often open at 3 a.m., so they must survive on Twinkies and coffee.

Looking to the future, some Real Programmers are concerned that the latest generation of programmers are not brought up with th ame outlook on life as their elders. Many of them have never seen a computer with a frontpanel. Hardly anyone graduating from school these days can do hex arithmetic without a calculator. Today's college graduates are soft-protected from the realities of programming by source level debuggers, text editors that count parentheses, and user-friently operating systems. Worst of all, some of these alleged computer

Uitbreidingen en veranderingen.

Een logisch gevolg van al het gezocht en gespit in de kit is het veranderen van dingen zodanig dat ze wat meer voor de hand liggen en/of makkelijker te onthouden zijn. Waarom bv. heet het find-commando H (van hunt) en niet gewoon F? De eerste verandering dus.

Om dezelfde reden heb ik de commando's B (bottom) en T (top) omgewisseld, dus T (top) en B (begin). Top breng ik in verband met het eind van het programma (P.&TOP) en niet met het begin. Een nadeel is nu dat B= veranderd is in T=. De bij de originele top behorende string **\*\*TOP\*\*** is veranderd in **\*BEGIN\***.

EDIT is vervangen door EDX t.g.v. EDIT van de word pack.

Het commando ? dat informatie over het programma en de buffer afdruckt en dat ook met OLD is te verkrijgen, heb ik veranderd in @. Deze jumpt naar #2800, zodat een daargeplaatste routine (eindigende met RTS) gebruikt kan worden voor extra's naar keuze.

Als je bv. zou willen weten hoe vaak een bepaalde string in een programma voorkomt, dan tik je eerst deze twee regels in:

```
P=#2800;[SEC;LDA@255;SBC9;STA#16;LDA@0;]
[STA#25;STA#34;STA#43;JSR#C589;JSR#FFED;RTS;]

```

waarna je in EDX de volgende commandostring intikt:

```
?B^F.string.255^@
```

en het antwoord verschijnt in beeld.

De listing na L. kun je onderbreken met ESCape.

Uit gesprekken met voormalige Toolbug-gebruikers bleek dat de melding TOO LONG, RETYPE verwarring stichtte. De nieuwe versie meldt in een dergelijk geval FULL, HIT CR en wacht dan net zolang tot de RETURN-toets wordt beroerd, waarna de regel zoals hij daar staat wordt geïnterpreteerd; geen misverstanden meer. De melding komt nu overigens pas na 128 i.p.v. 64 ingetikte karakters.

De regels van het basic-programma mogen tot 128 karakters lang zijn (was 64).

Voor de DOSsers (maar misschien ook wel voor de COSsers) een extraatje. Met Q<cr> verlaat je de Toolbug op de bekende wijze, maar als je na Q een naam opgeeft, wordt het programma eerst onder die naam geSAVED voordat je terugkeert naar direct mode. Bv. QSTITCH. Het executieadres is #C2B2; is eventueel te veranderen in #AFAF of om het even welk ander adres.

Tot zover de veranderingen in EDX. Ook in DEBUG was reden tot "ontevredenheid"

C (van crack (?)) is L geworden (van list) naar analogie van de monitor.RH

ACORNTJESBROOD

```

=====
400 IF K=CH"L" X=X-1
410 IF K=CH"S" GOTO f
420 IF X<1 OR X>10 OR Y<1 OR Y>10 GOTO g
430 W=?((Y-1)*10+X+H)
440 IF W=0 GOTO j
450 IF W=1 GOTO g
460 IF W=2 GOTO i
470 IF W=3 GOTO h
480eP."";L=L+1;IF L>10 GOTO m
490 VTAB 12;P."YOU FOUND THE PATH TO LEVEL "L
500 PAUSE 420;GOTO d
510gVTAB 12;P."SORRY, THAT'S A WALL"
520 IF X>0 AND X<11 AND Y>0 AND Y<11 GOTO n
530 GOTO o
540nVTAB Y;HTAB X*2+2;P."w"
550oX=A;Y=B
560 GOTO j
570fS=S-4
580iVTAB 12;P."YOU FOUND A SECRET PASSAGE"
590 VTAB Y;HTAB X*2+2;P."s"
600 IF X>0 X=X-1-A.R.%3
610 IF X<0 X=X+1+A.R.%3
620 IF Y>P Y=Y-1-A.R.%3
630 IF Y<P Y=Y+1+A.R.%3
640 IF X>10 X=10
650 IF X<1 X=1
660 IF Y<1 Y=1
670 IF Y>10 Y=10
680 IF ?((Y-1)*10+X+H)=0 GOTO l
690 X=A;Y=B
700lPAUSE 180
710 VTAB B;HTAB A*2+2;P." "
720jS=S-1;IF S<1 GOTO k
730 IF X=0 AND Y=P GOTO e
740 GOTO p
750hL=L-1;IFL<1 L=1
760 VTAB 12;P."THROUGH TRAPDOOR TO LEVEL "L
770 PAUSE 420;GOTO d
780mP.#12;VTAB 7;P."YOU HAVE ESCAPED FROM"
790 P."----MERLIN'S MAZE----";?#E1=#80
800 END
810kP.#12;VTAB 7;P."YOU ARE TRAPPED IN MERLIN'S"
820 P."MAZE AT LEVEL "L"
830 P."!!!!!! YOU LOSE";?#E1=#80;END
840tP.#12;HTAB 10;P."MERLIN'S MAZE"
850 P."WELCOME TO MERLIN'S MAZE." "THIS IS A 10 LEVEL MAZE IN"
860 P." WHICH HIDES WALLS, TRAPDOORS, SECRET PASSES, AND "
870 P."LAST BUT NOT LEAST ESCAPE PATHS!!!"
880 P."THE OBJECT IS TO FIND YOUR WAY THROUGH 10 LEVELS "
890 P."OF MERLIN'S MAZE!!!"
900 P."HIT KEY TO CONTINUE.";INKEY A
910 P.#12"TO MOVE:TYPE -U- FOR UP"
920 HTAB 13;P."-D- FOR DOWN"
930 HTAB 13;P."-L- FOR LEFT AND"
940 HTAB 13;P."-R- FOR RIGHT."
950 HTAB 13;P."-S- FOR SECRET PASS"
960 P."YOU ONLY HAVE 123 MOVES TO ESCAPE FROM"
970 P." MERLIN'S MAZE."
980 P."IF YOU DECIDE TO TRY A SECRET PASSAGE, YOU WILL "
990 P."FORFEIT" "5 MOVES."
1000 P."HIT KEY TO START....";INKEY A
1010 RETURN

```

Uit de oude doos.....

Hier volgen twee programma's welke ik in de oude doos gevonden heb. Het zijn programma's a la 'intikken en runnen maar' en kunnen toch veel plezier opleveren tijdens een vrij uurtje.

Het eerste programma 'LOVE' tekent het LOVE logo, een overblijfsel van de flower-power tijd, met behulp van de letters welk U aaneengesloten dient in te typen, op papier via een printer. U moet dus over een printer beschikken, want via het beeldscherm heeft het geen enkel effect.

Het tweede programma is een leuk, onderhoudend spelletje voor de minimale Atom (wel P-Charme). De spelregels zitten in het programma, dus hier verder geen uitleg.

Ook gevonden: TWENTS BASIC voor de beginner:

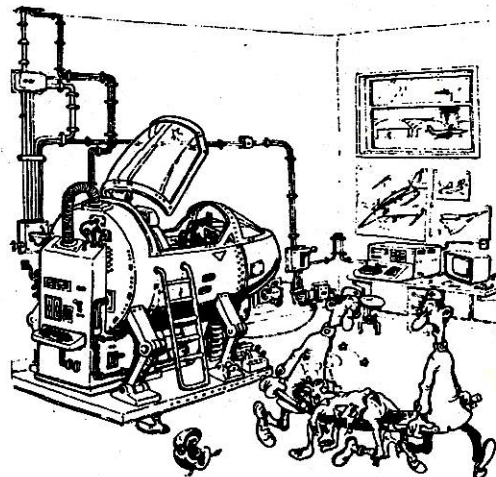
|              |                   |
|--------------|-------------------|
| PRINT        | SCHRIEF           |
| LET          | LOAT              |
| INPUT        | GEF OP            |
| GOTO         | GOA NOA           |
| RETURN       | GOA TRUG          |
| SAVE         | ONTHOAL           |
| READ         | LEES              |
| CLEAR        | MOAK SCHOONE      |
| CLOSE        | DOO TOO           |
| OPEN         | DOO LOS           |
| RUN          | LÖP VERDAN        |
| ERROR        | DAT DOOJ VERKEERD |
| GOSUB        | DOO TUSS'NSTUK    |
| IF/THEN/ELSE | AS/DEN/AANDERS    |

Gerrit Hillebrand

```

10 REM LOVE - LOGO BY DAVID AHL
20 REM CREATIVE COMPUTING
30 REM ADAPTED FOR ACORN ATOM
40 REM BY GERRIT HILLEBRAND
50
60 P.$12,"      LOVE"
70 RESTORE
80 DIMT(120),A(60)
90 IN."YOUR MESSAGE : "$A
100 L=LEN(A)
110 P.$2$21"
120 FOR J=0TO(60/L)
130 FOR I=0TO(L-1)
140 $T+(J*L+I)=$A+I
150 NEXT I
160 NEXT J
170 C=0
180 W=0;P=1;C=C+1;IFC=37;G.a
190 P.
200 READQ
210 W=W+Q;IFP;GOTO b
220 FOR I=1TOQ
230 P." "
240 NEXT I
250 P=1;GOTO c

```



*These computerized flight simulators are just a bit too realistic—especially for crash landings...*



Het resultaat van dit alles wordt door onderstaand plaatje nog eens geïllustreerd.

**SCORE 17-47 !  
U HEEFT GEWONNEN**

|   | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | ● | ● | ● | ● | ● | ● | ● | ● |
| 2 | ● | ● | ○ | ○ | ○ | ○ | ○ | ○ |
| 3 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 4 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 5 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 6 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 7 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 8 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

**PIET HEIN  
NIVEAU: 6**

|    |    |    |
|----|----|----|
| 18 | E1 | G1 |
| 19 | G8 | D2 |
| 20 | G7 | B6 |
| 21 | C1 | B1 |
| 22 | A5 | H8 |
| 23 | B4 | B8 |
| 24 | B5 | H2 |
| 25 | G2 | A3 |
| 26 | A4 | H1 |
| 27 | C2 | A6 |
| 28 | A7 | B2 |
| 29 | B7 | A8 |
| 30 | A2 | A1 |

Bij het naspelen van deze oplossing kwam ondergetekende tot de conclusie dat het 'eindspel' van Bram niet helemaal je dat was. Onderstaand plaatje geeft het eindresultaat met een ander eindspel (vanaf zet nr. 27).

**SCORE 9-55 !  
U HEEFT GEWONNEN**

|   | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | ● | ● | ● | ● | ● | ● | ● | ● |
| 2 | ● | ● | ● | ● | ● | ● | ● | ● |
| 3 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 4 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 5 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 6 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 7 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 8 | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

**PIET HEIN  
NIVEAU: 6**

|    |     |     |
|----|-----|-----|
| 19 | G8  | D2  |
| 20 | G7  | B6  |
| 21 | C1  | B1  |
| 22 | A5  | H8  |
| 23 | B4  | B8  |
| 24 | B5  | H2  |
| 25 | G2  | A3  |
| 26 | A4  | H1  |
| 27 | C2  | B2  |
| 28 | A2  | B7  |
| 29 | A6  | PAS |
| 30 | A7  | A8  |
| 31 | PAS | A1  |

Omdat er verder geen oplossingen voor wedstrijd B binnengekomen zijn, heeft de redactie besloten zich niet te storen aan de opmerking van Bram : 'buiten mededinging'. Hij krijgt dit Broodje dus gratis.

Namens de redactie  
Jan Biel

GELEZEN: Spreuk

Ronald Boers

Programmeren is moeilijker dan je denkt,  
Programmeren is makkelijker als je denkt.

Beste Redactie,

Ik vind uw Reversi-wedstrijd (met een "i", dus niet Reversie) een bijzonder aardig alternatief. Het is wel duidelijk dat u graag enkele interessante spelontwikkelingen wilt ontvangen. Ik wilde daarom buiten mededinging een spelletje met u spelen op niveau 6. Ik zal daarbij proberen om de diverse zetten wat nader toe te lichten teneinde de achtergronden daarvan wat duidelijker te maken. Let hierbij ook vooral op de puntenwaardering die telkens linksboven in beeld verschijnt. Hoe hoger, hoe beter. Laten we voor het gemak de combinatie Atom-Reversi, SEW "hij" en "hem" noemen.

Ok, daar gaat-ie dan.

RUN

niveau : 6

beginnen : Nee

- 1 D3.. alles is in principe identiek
- 1 ..C5 zomaar
- 2 C6.. hij wil graag de binnenhoeken hebben
- 2 ..C3 ik ook
- 3 C4.. hij vult een gat op; laat mij maar 1 keus
- 3 ..E3 deze dus. Het is te vroeg om buiten het binnenblok (4\*4) te treden
- 4 F6.. hij wil graag de binnenhoeken hebben
- 4 ..D6 rustig aan
- 5 E6.. hij pakt nu niet de hoek omdat ik dat voorlopig toch niet kan
- 5 ..C7 ik moet nu naar buiten
- 6 F3.. hij pakt nu toch de hoek bij gebrek aan een beter alternatief
- 6 ..G6 beter dan 6 ..F4 of 6 ..F5 omdat hij daarna 7 F5.. resp. 7 F4.. doet. Leidt nergens toe
- 7 B3.. voorbereiding op 8 C8..
- 7 ..F5 nu kan hij niet naar F4 en moet iets anders verzinnen
- 8 C8.. inderdaad, en sluit daarmee het linkergebied voor zichzelf af
- 8 ..F4 ik kan lekker wel op F4
- 9 H6.. en sluit daarmee het ondergebied voor zichzelf af

Het is wel duidelijk dat degene die gedwongen zetten moet doen sterk in het nadeel is. In de huidige situatie kan hij nog maar 1 van de 4 kanten op. Het is voor mij zaak om dit uit te buiten.

- 9 ..F2 als hij op 10 F1.. zet, sluit hij nog meer mogelijkheden af
- 10 G4.. hij beseft dat ook
- 10 ..G5 hou het aantal antwoorden klein
- 11 F1.. "5" is hier te hoog gewaardeerd
- 11 ..E7 een lokkertje op 12 E8.. gevolgd door 12 ..D8. Redelijk risicoloos
- 12 F8.. hij trapt er niet in
- 12 ..D7 13 D8.. kan niet wegens 13 (E8..) en vice versa

bedoeling van de programmeur wordt duidelijker etc.  
 De zin van DEZE macro's is natuurlijk dat als een of andere  
 ..... (vul maar in) een schit-te-rend programma heeft  
 geschreven in 65C02-taal, wij dat dankzij Matthes ook kunnen  
 draaien, weliswaar met verlengde code en vertraagde executie,  
 maar toch....

Ik beseft terdege dat dit hak-tak-verhaal nog vele aspecten  
 onbehandeld laat en dat het aantal toepassingen bijzonder groot  
 is. Als eerste kennismaking moet ik het hier echter bij laten.  
 Speel er een beetje mee, let goed op de vertaalde code (.LIST)  
 en laat eens wat van je .....

P.S. Ik heb een kopie van het originele artikel uit c't aan het  
 drukwerkarchief cadeau gedaan.

## PROGRAM MACRO DEMO

### INFORMATION:

START:#7800, END:#7B4F, LENGTH:#34F (=847)  
 #LINES: 33, #PAGES: 1, #LINES ON LAST PAGE: 43  
 LAST LINE: 330, LONGEST LINE: 120, (58 CHAR'S)

```

10 PROGRAM MACRO DEMO
20
30 FUNCTION RMB(B,Z)
40 (* zoals je uit de bijbehorende listings kunt afleiden,
50   mag de naam ook uit een bestaande instructie bestaan
60   bv. FUNCTION DEC *)
70   ASM-C
80   .LIST
90   (* zonder R.80 worden de macro instructies
100   niet afgedrukt; ook niet bij .LIST in
110   het hoofdprogramma *) PHP;PHA
120   :DUMMY=%(2^B+.5);LDA@DUMMY;EOR@#FF;AND Z;STA Z;PLA;PLP
130   (* uiteraard moet de macro niet eindigen met RTS *)
140   .END
150 (* aan de naam 'RMB' wordt geen waarde toegekend *)
160 FEND
170
180 PASS 0; GOSUB c
190 PASS 1; GOSUB c
200 LINK (DEMO)
210 END
220
230c ASM-BEGIN
240 .LIST .CODE #2800
250:DEMO
260 LDA@#7F; STA #80; JSR #F7FA
270 .MACRO RMB(O,#80)
280 .MACRO RMB(3,#80)
290 .MACRO RMB(5,#80)
300 (* #7F (:01111111) VERANDERT IN #56 (:01010110) *)
310 LDA #80; JSR #F7FA; RTS
320 .END
330 RETURN
  
```

SOFTWARE:65C02-instructies op de Atom

Bram Poot

Trouwe lezers en lezeressen van het Acorn-tjesbrood zullen zo langzamerhand wel gemerkt hebben dat ik nogal eens aan het Duitse computerblad c't refereer omdat daar vaak alleszins belangwekkende artikelen in worden afgedrukt. Zo ook weer in de onlangs verschenen uitgave nr.9 '85 (aug./sept.) waar een faciliteit wordt besproken die (natuurlijk) ook op de Atom gedraaid kan worden. Laat ik ter introductie de inleiding van het artikel even vertalen.

"Sinds het beschikbaar komen van de CMOS-6502 met z'n nogal uitgebreide instructieset heeft vermoedelijk menige 6502-fan overwogen hoe je dit nieuwe familielid kunt inzetten. Maar bij gebrek aan fatsoenlijke testmogelijkheden bestaat het eind van die overwegingen meestal uit het besef dat je zult terugvallen op de standaardversie en deze net zo goed kunt handhaven. Toch is er een alternatief, dat niet veel moeite kost: het eenvoudigweg simuleren van de nieuwe instructies op de standaard 6502."

Frank Matthes, de schrijver van het artikel, heeft deze simulatie verwezenlijkt door de nieuwe instructies (een bestaande instructie met nieuwe adresseringsmode wordt ook als nieuwe instructie opgevat) als macro te definiëren. Op de plek van zo'n nieuwe instructie wordt dan een routine ingelast die exact hetzelfde resultaat oplevert als die instructie op een 65C02 gedaan zou hebben. Dus niet alleen A, X en Y maar ook P en S zijn volgens de instructie veranderd (of gelijk gebleven). Voor het implementeren van een voor de 65C02 geschreven programma op een 6502 heb je dus een macro-assembler nodig, en het leuke is nu dat SALFAA 1.5 macro's kan simuleren. Dat je daar tot nu toe nog niets van wist, komt omdat ik het niet gedocumenteerd heb. En ik heb het niet gedocumenteerd omdat deze materie voor een simpele machine als de Atom veel te moeilijk is om het boem-bats zomaar ineens te introduceren. Bovendien had ik tot voor kort geen goed voorbeeld voor het nuttig gebruik van macro's (wat dus de beschrijving ervan niet veel duidelijker had kunnen maken). Dankzij Matthes heb ik dat nu wel.

Overigens wil ik op deze plek wel even waarschuwen dat je voor een goed begrip van het nu volgende vertrouwd moet zijn met het gebruik van SALFAA, aangezien het geen algemeen verhaal is, maar toegespitst op deze implementatie.

Voor de goede orde: het gebruik van SALFAA-"macro's" verlangt het volledig mengen van SALFAA en P-charme. Op dit moment is dat alleen getest met P-charme op #1XXX, en ik ben bang (nou ja...) dat dit überhaupt de enige mogelijkheid is.

Ten aanzien van de macro's van Matthes vallen vier opmerkingen te maken:

1. Sommige routines lijken nodeloos ingewikkeld. Dit komt echter vaak doordat die ene instructie tot in de puntjes gesimuleerd dient te worden. Het incrementeren van de accu kan bv. via ADC@1, maar deze instructie verandert de C- en V-vlaggen terwijl de INA-instructie (of INC A) dat niet doet. Het statusbyte moet dus via een omweg (INX) aangepast worden.

**-BAD DATA**

Klokverschuivingen in de hardware is hier meestal de oorzaak.  
Probeer het nog een keer of anders zie CRC.

**-FACILITY NOT ENABLED**

FORMAT en WIPE geprobeerd, zonder een ENABLE vooraf geeft deze foutmelding.

**-WRITE PROTECTED DISC**

Schrijfoperaties naar een beschermde schijf zullen beloond worden met deze spreuk.

**-WRITE FAULT TR:t SCT:s**

Schrijffout in sector s op track t. Oorzaak kan een kapotte sector zijn. Zie CRC.

**-RECORD NOT FOUND TR:t SCT:s**

Sector s niet gevonden in track t. Oorzaak: verkeerde density, of foutief sectornummer bij RSECT en/of WSECT of een kapotte sector/track of een ongeformatteerde schijf.

**-NO SECTORS?**

Voor het aantal sectoren in RSECT of WSECT is 0 opgegeven. Dit is natuurlijk tamelijk onzinnig.

**-MAXIMUM ACCOUNT ON DISC**

Het maximum van 46 files op een schijfkant is bij deze melding bereikt.

**-NO ROOM ON DISC**

De te saven file past niet meer op de geselecteerde schijfkant.

Tot zover de beschrijving van GDOS 1.3. Hopelijk is er in het volgend broodje meer te melden.

Gerrit Hillebrand  
Mendelssohnstraat 30  
7557 BJ Hengelo (0)

---

TE KOOP : ACDRN ATOM

IN DEZE ATOM IS INGEBOUWD :

- SCHAKELKAART
- GEHEUGENKAART
- SCREENNOISE ELIMINATOR (HARDWARE)
- SERIELE INTERFACE RS232 (+-12V)
- P-CHARME OP #1000
- VIA
- MONITORAANSLUITING
- RAM OP #800, #2000

TOTAAL: 12\*6116, 24\*2114, 6522, 27128, 3\*2732, 6850, P-CHARME, COMBOX,  
JOBBOX, SALFAA, GAGS, ED64, FLOATING POINT

VRAAGPRIJS (MET VOEDING) : f1 350,--

TE BEVRAGEN : HARM DE LEEUW  
WELEVELDSTRAAT 46  
7557 JH HENGEL0  
TEL 074-421873

- \*INFO <naam> - displayed informatie tabel omtrent de opgegeven file volgens:  
loadadr execadr endadr filelen startsector
- \*KEEP n - slaat de directory van drive n op in de tweede directory op dezelfde schijf t.b.v. directory-crashes
- \*LIB - zelfde als CAT, doch directory wordt gelezen vanuit het geheugen en niet vanaf schijf.
- \*LOAD <naam> {l} - laadt een file in het geheugen vanaf adres l. Indien l niet opgegeven, dan vanaf zijn loadadres zoals deze gesaved is op schijf.
- \*LOCK <naam> - 'locked' een file op schijf
- \*MON - displayed op het beeldscherm tijdens DOS handling
- \*NOMON - displayed NIET tijdens DOS handling
- \*OSCLI .....\$13 - waanzinnig mooi commando. De volledige beschrijving is te vinden in Acorn Nieuws Nr.4.4 geschreven door de programmeur van dit commando: Bram Poot.
- \*RECOVER - haalt tweede directory van schijf en zet het in het geheugen en saved het wederom in de eerste directory.
- \*RENAME <naam1> <naam2> - de naam van de file (naam1) wordt gewijzigd in de tweede naam (naam2).
- \*RSECT t s q mem - Haalt sector(s) vanaf schijf binnen, te weten : beginnende bij sector s op track t. In totaal haalt het q sectoren binnen en plaatst deze achter elkaar in het geheugen vanaf mem.
- \*RUN <naam> - file met de juiste naam wordt geladen en wordt direct gestart (springt naar executie adres).
- \*SAVE <naam> l e {x} - saved file op schijf onder de naam <naam> met loadadres l en eindadres e. Optioneel is het executieadres x.
- \*SD n - schakelt single-density mode en drive n in.
- \*SETk - zet de qualifier op karakter k
- \*TITLE <naam> - geeft de schijfkant een titel. (titel maximaal 14 karakters)
- \*UNLOCK <naam> - opgegeven file wordt 'geunlocked'.
- \*USEk - geeft tijdelijk de qualifier de karakter k
- \*VERIFY n - verifieerd de schijf in drive n. (indien onbetrouwbaar dan zal er een vraagteken achter het track nummer geplaatst worden)

**GDOS 1.3 : DOUBLE DENSITY**

(of: Can You Control Your Drives ?)

Zoals je in het vorige broodje hebt kunnen lezen is de nieuwe Atom disc-pack met GDOS 1.0 (gebaseerd op de goedkope, maar o zo krachtige WD1793) volledig operationeel. Doch een week na deze demonstratie was de hard- en de software al zodanig veranderd, dat er vanuit GDOS 1.0 de nieuwe GDOS 1.3 was ontstaan. Wat is er gebeurd? Direct na het verschijnen van het artikel begon men enkele zeer lastige vragen te stellen, zoals:

- 'kan ie ook double density?'

Neen, dat KON het niet i.v.m. het feit dat de Atom op 1Mhz loopt en daardoor de loops in het programma om data te lezen en te schrijven te langzaam voor de controller waren.

Oplossing: disc-controller communiceert nu met interrupts.

Daardoor kreeg ik meer speelruimte in de lees- en schrijf routines met als gevolg dat double density mogelijk was en daardoor nu de gebruiker de mogelijkheid geeft om 160K per kant op te slaan.

- 'kan ie ook oude Atom schijven lezen?'

Neen, dat KON het niet. Dit kwam doordat de 8271 controller uit de oude Atom-disc de IBM standaard volledig aan zijn laars lapt, want de standaard vermeldt dat de sectoren moeten beginnen met nummer 1 en de 8271 begint met nummer 0 !!

Daarbij komt nog dat de directory-structuur van GDOS volledig verschilt met die van de oude dos t.g.v. de mogelijkheid om filenamen een lengte van 15 karakters te geven en het maximaal aantal files per kant nu 46 bedraagt.

Oplossing: introduceer trucs (m.b.v. een programma b.v.) waarbij je van de oude Atom schijf sector voor sector of track voor track binnen haalt en wegschrijft volgens de nieuwe GDOS. Het programma welk zulks doet (??) vind je aan de staart van dit artikel.

Op deze manier is ook het gebrek van random-access files op te heffen: m.b.v. RSECT en WSECT zijn ook commando's als SPOOL etc. na te bootsen!!

In ieder geval is het nu mogelijk de veeel te dure ( en beperkte ) controller 8271 te laten varen en gebruik te gaan maken van de overweldigende kracht van de Western Digital 1793.

**De hardware**

- disc controller: WD1793 of WD1797
- 40 of 80 tracks
- Single- en Double density
- shugart compatible bus
- write precompensation
- eigen clock
- 4 55- of 20S drives aan te sluiten
- 15 IC's (totale kosten, excl. print: 100 gulden)
- gedecodeerd op #BFC0-#BFC7

BPROM wil nog wel eens wat veel boodschappen op het scherm doen verschijnen. Dit is een gevolg van het ontbreken van een commando of controlcode om 1 regel te wissen. P.#12 is in deze te rigoreus. Ook LINK#FE22 voldoet niet, omdat dit niet op andere VDU's dan de normale Atom VDU is te gebruiken. Als je echter de voortgang van de meldingen en vooral de cursorpositie in de gaten houdt (en regelmatig het scherm even schoon maakt), is er heel goed mee te werken.

#### Literatuur.

BPROM is een vrijwel direct gevolg van een artikel over eproms dat is verschenen in het duitse computerblad c't nr.2 jan/febr. 1985 p.88. Een kopie van dit artikel heb ik aan het drukwerkarchief cadeau gedaan.

Na keuring van dit artikel (copy cirkuleert bij ons meestal alvorens het geplaatst wordt), bleek dat de schema's van de programmer die in AcornNieuws zijn verschenen niet consistent zijn (verwarrend dus). Ter completering van dit verhaal dus de opmerking dat ik mijn programmer gebouwd heb volgens het allerorigineelste schema dat in Practical Computing d.d. juni '82 p.152 is verschenen. Dit schema staat ook in AcornNieuws nr.7 1982 p.51. Verder heb ik deze programmer geschikt gemaakt voor het programmeren van zowel de 2532- als de 2732-eprom via de methode die als "beste redactie"-item is beschreven in Acorn-tjesbrood 2.4/5 p.77.

SOFTWARE: Mastermind

Jan Biel

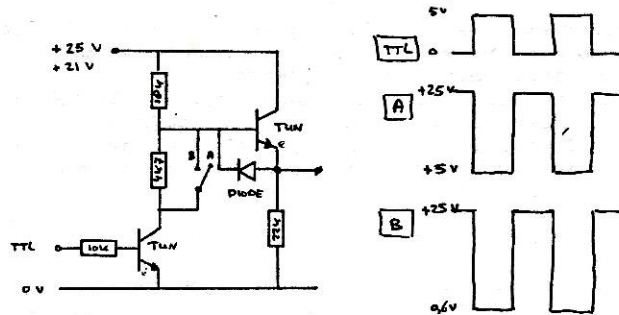
Hoewel het programma nog niet geschreven is wil ik toch alvast enige uitleg geven. Het gaat hier om het spelletje Mastermind, waarbij de computer uw code probeert te achterhalen. De code wordt niet met kleuren maar met cijfers aangegeven; voorlopig voor 6 cijfers die op 4 posities kunnen staan. Om te weten te komen hoeveel geheugenruimte hiervoor nodig is heb ik een programma geschreven wat uitrekent hoeveel mogelijke antwoorden overblijven bij een gewenst antwoord (U weet wel, de witte en zwarte paaltjes). Dit programma vind U aan het einde van dit stukje. De waardering voor de antwoorden is als volgt: de eenheden (0 t/m 4) stellen de witte paaltjes voor; de tientallen (1 t/m 4) de zwarte paaltjes. Daar het programma geheel in BASIC geschreven is, is het traag: de totale rekentijd bedraagt ongeveer 3 uur. Het test dan ook niet of een bepaald antwoord wel mogelijk is bij een bepaalde code. Maar daar ging het ook niet om.

Het is nu mijn bedoeling met behulp van de door het programma geleverde antwoorden een tabel samen te stellen waaruit de computer dan die codes haalt welke tot een optimale (dus zo min mogelijk zetten) oplossing moeten leiden. Uiteraard wordt het uiteindelijke programma dan ook geheel in machinetaal geschreven.



Op dit moment is het dus zaak om de kennis en ervaring van de vakman te raadplegen, weshalve ik onze voorzitter opbelde om te vragen of hij eens over dit probleem wilde nadenken. Via zijn grandiose 24-uurs service had ik echter de volgende avond reeds de beschikking over een schema. En je weet: Maarten, dan is 't goed!

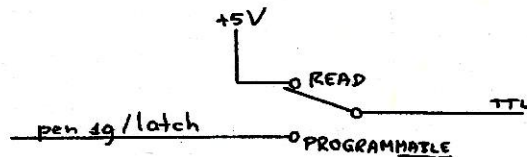
SCHAKELING VOOR INTELLIGENTE EPROM PROGRAMMER



DIODE: LIEFST SCHOTTKY TYPE (BAT85)

MAAR DUS GAAT OOK.

Een bijkomend voordeel van een dergelijke schakeling is dat BEPROM ook bij de andere commando's zelf kan bepalen wat de instelling moet zijn, hetgeen de kans op fouten verkleint. De TTL-ingang kun je verbinden met 1 van de 4 vrije latch-uitgangen bv. pen 19. Nog handiger is het om dit m.b.v. de vrijgekomen READ/PROGRAM-schakelaar te doen op de volgende manier:



Aangezien pen 19 bij de klassieke programmeerprogramma's continu laag is, behoudt de schakelaar zijn oorspronkelijke functie en kunnen deze programma's NOG STEEDS gebruikt worden (belangrijk!!).

Voor BEPROM staat de schakelaar in de stand 'programmable', waarna het programma voor de rest zorgt, ervan uitgaande dat de softswitch is geplaatst op de hoogste adreslijn (PB7, pen 19 van de latch).

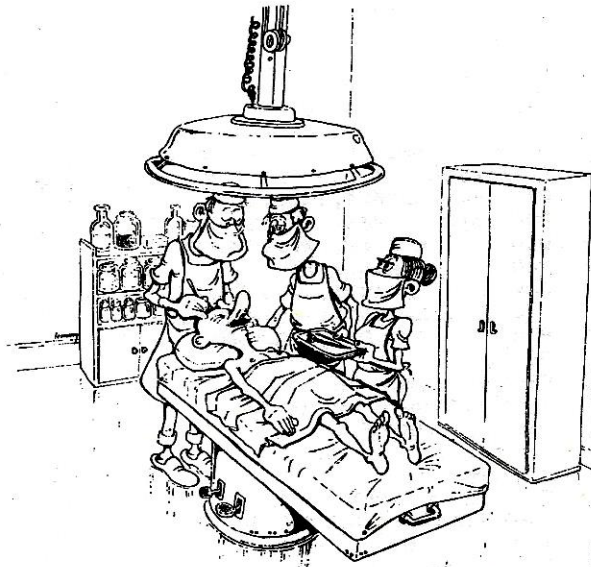
Meer dan deze informatie kan ik je helaas niet geven. Je zult de concrete realisatie dus zelf moeten verzorgen. Er zijn teveel verschillende configuraties van de programmer om hieraan te beginnen. Het is daarentegen niet echt moeilijk om e.e.a. in je eigen systeem in te bouwen en het loont, denk ik, zeker de moeite.



```

1120      LDA@#00
1130:SS12
1140\ SAY ALLOPHONE IN ACCU:
1150\ PRINTER ON
1160      PHA;LDA@#02;JSR#FEFB
1170\ MASK MSB
1180      PLA;ORA@#80;JSR#FEFB
1190\ PRINTER OFF
1200      LDA@#03;JMP#FEFB
1210
1220\=====
1230\ INTERRUPT SERVICE ROUTINE
1240:II0
1250\ INT. FROM CA1?
1260      LDA#BB0D
1270      AND@#02;BEQ II3;\NO
1280      TYA;PHA
1290\ RESET INT.
1300      \LDA#BB01 AUTOM. IN #FEFB
1310\ IF QUEUE EMPTY, SEND PA1
1320      LDA B+3;CMP B+2
1330      BNE II1
1342\ DISABLE CA1 INTERRUPT
1344      LDY@#02;STY#BB0E
1346\ CHECK HIGH OR LOW VOICE
1350      TAY;\=B+3;DEY
1360      LDA(B),Y
1370      AND@#40
1380      BPL II2;\ ALWAYS
1390:II1
1400\ QUEUE NOT EMPTY, SO GET
1410\ ALLOPHONE FROM QUEUE.
1420      TAY;\=B+3;LDA(B),Y
1430      INC B+3
1440:II2 JSR SS12;\ SAY IT
1450      PLA;TAY
1460      PLA;RTI
1470
1480:II3 JMP(V)
1490
1500\=====
1510\ SAY STATEMENT
1520\ READ ALLOPHONES AND
1530\ STORE IN SPEECH QUEUE
1540:LL0
1550\ CHECK IRQ-VECTOR
1560\ IF NOT AT II0, ERROR!
1570      JSR SS9
1580      BEQ LL1;BRK
1590:LL1 LDX@#FF;STX Z
1600:LL2 INC Z;LDY#03
1610:LL3 INX
1620      CPX@TT1-TT0;BCS LL6
1630:LL4 LDA(#05),Y;INY
1640      CMP@CH" ";BEQ LL4
1650      CMP@CH" ";BEQ LL15
1660      EOR TT0,X;BEQ LL3
1670      CMP@#80 ;BEQ LL7
1690:LL5 LDA TT0,X;BMI LL2
1700      INX;BNE LL5

```



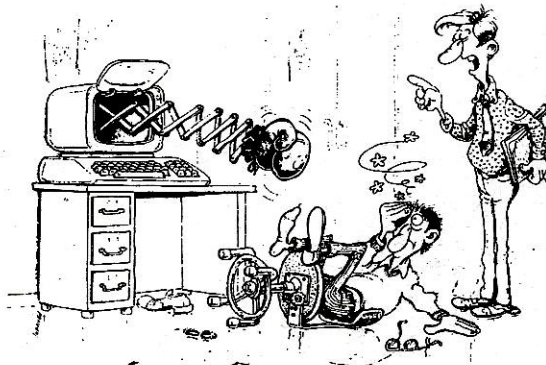
*This is it? No panel screws? No circuit boards? ...*

## PROGRAM QUEUED SPEECH

```

10 PROGRAM QUEUED SPEECH
40
50 REM VOOR SPO256-AL2 OP
60 REM CENTRONICS POORT.
65 REM TH. DEN EXTER,
70 REM R.BOERS & B.POOT
80
90 Z=#70;REM TEMP.ZEROPAGE IN SAY
100 B=#80
110 REM ZEROPAGE B,B+1,B+2,B+3,B+4 ARE USED
120 REM B,B+1:QUEUE-ADDRESS
130 REM B+2 :FIRST FREE BYTE IN SPEECH QUEUE
140 REM B+3 :FIRST ALLOPHONE TO BE SAID
150 REM B+4 :HIGH/LOW VOICE TESTBYTE
160 V=#21C
170 REM V,V+1:SAVE OF IRQ-VECTOR
180 DIM S6,SS12,II9,LL15,TT1
190 FOR I=0 TO 40;SSI=A;NEXT
200 P.$21;P=A;GOSUB a
210 P.$21;P=A;GOSUB a
220
230 $T="SPEAK";T=T+LEN(T)
240 T?0=SS0/256;#80
250 T?1=SS0%256
260 T=T+2
270 $T="SAY";T=T+LEN(T)
280 T?0=LL0/256;#80
290 T?1=LL0%256
300 T?2=#80;T=T+2
310 A=P;T!1=A
315 P.$6$7$7$7
320 P."TOT: #"&P'
330 ?#3FC=#25
340 END
350
360aI
370\=====
380\ SPEAK [QUEUE-ADDRESS]
390\ SPEAK : QUEUE AT DIM-POINTER
400\ SPEAK 0 : END OF QUEUED SPEECH
410:SS0
412\ DISABLE CA1 INTERRUPT
414 LDX#02;STX#BBOE
416\ AVDID ALL INTERUPTS
418 SEI
420\ READ OPTIONAL ARGUMENT
430 LDY#03;DEY
440:SS1 INY;LDA(#05),Y
450 CMP#CH" ";BEQ SS1
460 CMP#CH" ";BEQ SS6
470 CMP#0D ;BEQ SS6
480 STY#03
490 JSR#C4E1
500 DEC#04;LDX#04
510 LDY#25,X;TYA;ORA#16,X
520 BEQ SS8;\ ARGUMENT=0
530:SS2 LDA#16,X

```



*It tries that on everyone. Let it know who's boss!*

## SOFTWARE:Speech queue Ronald Boers, Theo den Exter &amp; Bram Foot

Met het programma Queued Speech beschik je over de mogelijkheid om op interruptbasis de SP0256 aan te sturen. Daartoe wordt een zgn. queue, een buffer, in het leven geroepen ter lengte van 256 bytes waarin de allophonen gebufferd worden alvorens ze naar de SP worden gestuurd. Terwijl deze dan aan het leuteren is, kun je verder gaan met het programma of wat je maar wilt.

Zoals je direct uit de listing kunt aflezen, worden de statements SPEAK en SAY op een gestandaardiseerde P-charme manier aangemaakt.

Het statement SPEAK wordt normalerwijze gevolgd door een adres dat het begin van de queue (256 bytes lang) vastlegt. Bij uitzondering mag in een programma (dus niet in direct mode) SPEAK zonder adres worden gebruikt in welk geval aan de DIM-pointer wordt gerefereerd. SPEAK alleen staat in feite gelijk aan DIM A(256); SPEAK A. En er mag daarna gerust verder geDIMd worden. Als er na het laatste woord niets meer gezegd hoeft te worden, kun je het queue-gebeuren (hoewel dit niet expliciet noodzakelijk is) afschakelen met SPEAK 0 (of na PROGRAM: SPEAK FALSE).

Nadat SPEAK de queue geactiveerd heeft, kan de SP0256 aan het babbelen gezet worden met het statement SAY. Achter SAY komt een willekeurig aantal argumenten die twee vormen kunnen hebben. De ene heeft de vorm van de allofoon-representatie zoals die in de bij het IC geleverde datasheet staat. Dus bv. SAY EY PA3 KK3 OR NN1 ("acorn") (zie ook Acorntjesbrood 3.1 p.36). De andere vorm bestaat uit een expressie voorafgegaan door een single quote ('). Dus bv. SAY '20 '2 '8 '58 '11. De beide vormen mogen ook gemengd worden en de spaties zijn optioneel zolang er geen misverstanden mogelijk zijn. Dus bv. SAYEY'2KK3ORNN1. In een constructie als SAY'#3 EY is de spatie nodig omdat E als hexadecimaal cijfer opgevat zou worden zonder spatie.

Als argument achter SAY kunnen als extraatje de pseudo-klanken HI, LO, NOW: en WAIT gebruikt worden. HI en LO schakelen het zevende bitje van de centronicspoort teneinde bij geschikte "spraakhardware" de intonatie van de stem te regelen. WAIT dient ervoor om in voorkomende gevallen het effect van de queue weer teniet te doen door te wachten tot de queue leeg is alvorens de interpretatie van de klanken voort te zetten. In SAY AX LL AR MM WAIT;P."!!!!!" worden de uitroeptekens pas afgedrukt als "alarm" volledig is uitgesproken.

NOW: maakt stante pede de buffer leeg, zodat hetgeen erop volgt onmiddellijk wordt uitgesproken. Bv. FOR I=0 TO 63;SAY 'I;NEXT I;SAY NOW: NN1 AW

Queued Speech is waar mogelijk behoorlijk (of liever: onbehoorlijk) gekrunchd hetgeen e.e.a. misschien ietwat onleesbaar maakt, maar waardoor het wel bijna binnen 2 geheugenpagina's past (inclusief allofoontabel).

Ten aanzien van de interrupts valt nog te vermelden dat alleen de A-poort met z'n controllijnen wordt gebruikt en wel zonder de overige instellingen van de VIA aan te tasten. B-poort, shiftregister en timers blijven dus onverkort toepasbaar. Ter demonstratie hiervan staat op het Acorntjesbroodbandje het programma KLOKJE, een real time klokje op interruptbasis (timer 2 van de VIA). Als u dit programma RUNt, verschijnt rechts boven in beeld een digitaal klokje, dat ook bij het gebruik van SPEAK en SAY doorloopt.

PROJEKT: SPO256-AL2 op Centronics poort

Bram Poot

Als u Acorntjesbrood en de andere regiobladen goed gelezen heeft, weet u dat er al een heleboel koppelingen aan de Atom van het spraak-IC SPO256-AL2 zijn beschreven.

Waarom dan nog een keer deze materie behandeld?

Omdat de hier te beschrijven methode geen (of hooguit nauwelijks) extra software nodig heeft. Er wordt namelijk gebruik gemaakt van de Atom printerpoort.

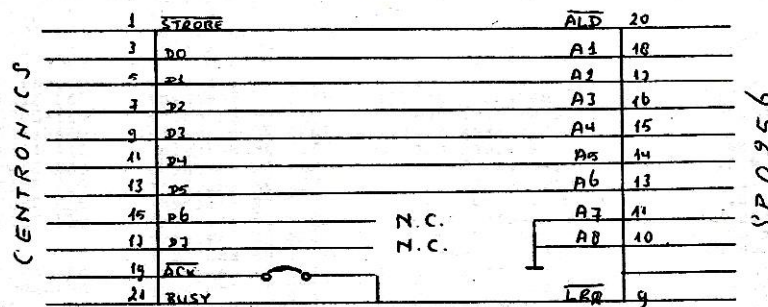
Voordelen:

1. software reeds aanwezig
2. gebufferde signalen (verbinding mag meters lang zijn)
3. universele koppeling (ook op andere computers te gebruiken)

Nadelen:

1. VIA (IC1), 74LS244 (IC50), printerconnector en printerkabel nodig
2. printer en spraak-IC niet tegelijkertijd te gebruiken
3. niet "rack-compatible" (wat dat ook moge betekenen)

Voor ons (een klein groepje TH-studenten) golden en gelden de nadelen niet, zodat de keuze van implementatie wel duidelijk is. Het principeschema ziet er aldus uit:



Hoe de rest van het spraak-IC moet of kan worden aangesloten, kunt u in de eerder vermelde bronnen lezen.

De software schittert door eenvoud:

```

10 REM SPREEK!
20 RESTORE
30 PRINT $2
40 DO READ X
50 PRINT $X:#80
60 UNTIL X=#80
70 PRINT $3
80 END
90 DATA 20,2,8,58,11,#80

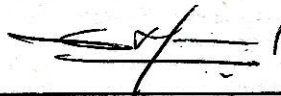
```

De printerdriver zorgt voor de handshake (die overigens de VIA enigszins tekort doet), zodat u zich geen zorgen hoeft te maken over de tijdsduur van de fonemen. Pas als het IC uitgekletst is, wordt de volgende klank verstuurd.

BEAR HARDWARE  
68 Harmondsworth Lane  
Harmondsworth  
Middlesex UB7 0AA

In dit vierde nummer is ook de uitslag van de reversi-wedstrijd te vinden. Gezien het aantal deelnemers kunnen niet veel mensen zenuwachtig zijn voor wat betreft de uitslag, maar de oplossingen die er binnen kwamen waren dan ook fanfabutastisch.

Laat dan nu de blad'ren ritselen en verdiep U in dit vierde nummer des 1985 van..... Acorntjesbrood !! Veel plezier!!!!!!!!!!!!!!




---

IN DIT NUMMER :

- Blz. Wat staat daar?
1. Redac(orn)tioneel, deze keer door Gerrit
  2. Inhoudsopgave
  3. Ledenlijst
  4. Het SPRAAK IC op de Centronics poort, door Bram
  6. SPEECH-QUEUE, door Bram, Theo en Ronald
  7. Een toepassing hiervoor: REAL TIME CLOCK
  8. Het SPEECH-QUEUE programma
  12. CITYBOMBER met spraak, door Ronald
  13. BPR0M : nu wel beschikbaar, door Bram
  17. TESTPROGRAMMA voor een nieuwe mastermind door Jan
  18. Comin' on strong: GDOS 1.3 DOUBLE DENSITY door Gerrit
  23. ACORN DOS schijven lezen met GDOS door Gerrit
  24. MACRO: 65C02 instructies op de Atom door Bram
  27. Uitslag van de WEDSTRIJD uit Broodje 3.3
  31. Direct mode statement voor P-CHARME door Theo
  32. Uit de oude doos: LOVE en MERLIN'S MAZE
  35. Revised TOOLBUG door Bram
  38. Het slot van REAL PROGRAMMERS DON'T USE PASCAL
  40. Het wel en wee van de EXPANSION BUS door Bram
  41. Theo vind een foutje in GRMDD
  43. Intikken en runnen maar: ALLERLEI

ALGEMENE INFORMATIE:

---

Bestuur:

Voorzitter: M.v.Alphen  
Secretaris: J.Laarman  
Penningmeester: W.Verhoeven  
B.Foot  
M.Stoop

Redactie:

J.Biel  
G.Hillebrand  
Th.den Exter

Programma-archief: M.Stoop      Printdienst: W.Verhoeven  
Drukwerkarchief: R.Boers      Ledenadministratie: W.Verhoeven  
Epromprogrammeerdienst: R.Boers

