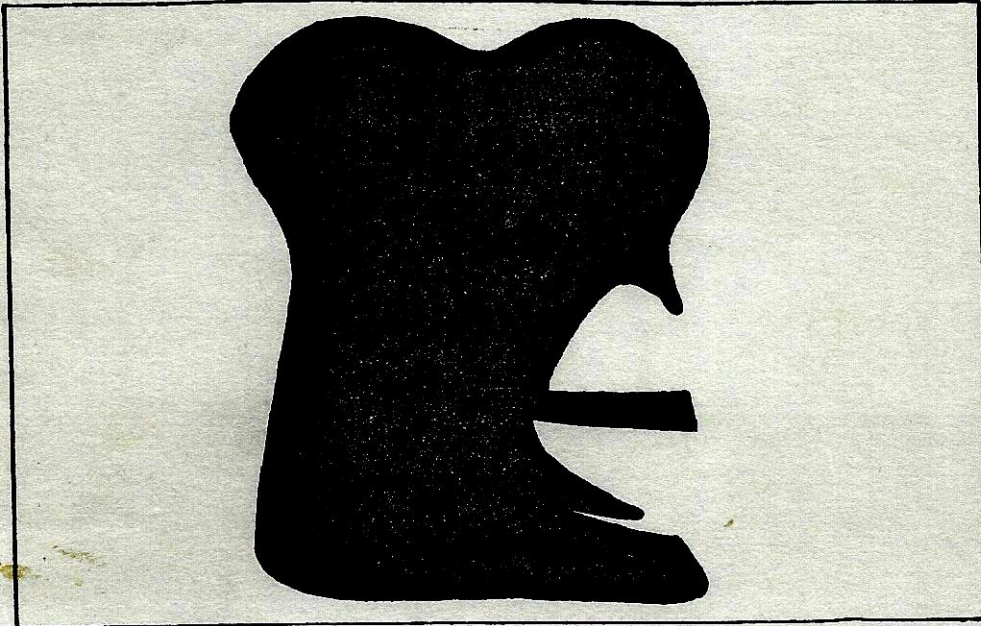


PRIJS F.2.50

Jaargang: **EEEEEE**

Nummer: **EE**



Acortjesbrood

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

ACORTJESBROOD IS HET ORGAAN VAN DE
ATOM COMPUTER CLUB

WENTE

REDAC(ORN)TIONEEL

Geheel volgens schema ligt nu het tweede nummer van Acorntjesbrood 1987 voor U. Als een van de eerste dingen die opvallen noem ik de aanzienlijke daling van het aantal clubleden in onze regi (ongeveer 50% ten opzichte van de laatst gepubliceerde leden lijst). Op de landelijke vergadering van 14 maart jl. bleek dat landelijk gezien het aantal geregistreerde leden ongeveer 550 bedraagt.

Het verslag van deze vergadering zal in het volgende Broodje gepubliceerd worden, mede doordat in dit nummer pas het verslag van de vorige ALV geplaatst is (hoe bedoelt U: 'trage' ambtenaren). Verder brengt Paul in dit nummer verslag uit van wat zich zoal het afgelopen jaar in onze regio afgespeeld heeft.

Het 'einde' van de officiële mededelingen in dit nummer wordt gevormd door de nieuwe spelregels van de eprom-programmeerdienst, die nu in handen is van 'burning' Rob van de Schepop.

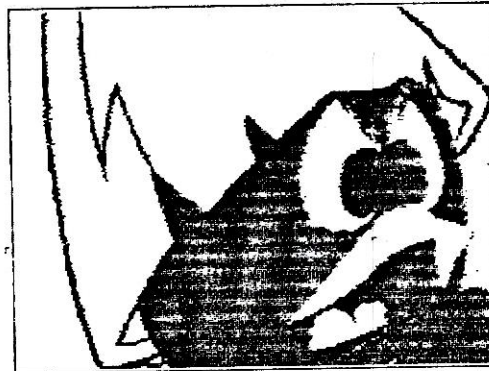
Tot onze grote vreugde heeft een voormalig lid van de regi Zwolle ijverig in zijn oude doos zitten spitten, met als resultaat een viertal bijzonder educatieve programmatjes: Simpel, maar daarom niet minder geslaagd.

Verder wordt in dit nummer een begin gemaakt met een 'thuis' cursus LISP, akomstig van Paul. Met uiteraard een simpel voorbeeld, wat, als we het bijbehorende plaatje moeten geloven, zelf het landelijke nieuws gehaald heeft.

Tot slot in dit nummer van uw lijfblad nog enkele 'harde' aangelegenheden, variërend van wat doe ik met de lengte van mijn flatcables tot een afgezwakte 80-kolommenkaart.

Kortom, weer een interessant en afwisselend nummer (onder het motto voor elk wat wils) volgens de redactie.

Jan



22-03-87

LEDENLIJST ATOM CLUB TWENTE

NAAM:	ADRES:	PC:	WOONPLAATS:	TELEFOON:
Alphen, M.P. van	Korianderhof 40	7641	XK Wierden	05496 - 2264
Berendsen, H.H.J.	Larenseweg 19	7475	PV Markelo	05476 - 1290
Biel, J.K.	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
Borgerink, J.A.L.	Geert Grootestraat 9	7665	AV Albergen	05494 - 1612
Breden, H.	Brinkgreverweg 23	7413	AA Deventer	05700 - 36182
Bruggen, T. van	Spartastraat 67	7535	BL Enschede	053 - 316709
Coupric, J.P.	Braakmansdijk 52	7462	LW Rijssen	05480 - 19477
Drijver, G. den	Kotkampweg 188	7531	JK Enschede	053 - 354225
Hillebrand, G.	Mendelssohnstraat 30	7557	BJ Hengelo	074 - 912931
Huisken, P.	Hogelandstraat 6	7573	CK Oldenzaal	05410 - 14713
Jansen Klomp, J.P.	Drieturven 3	7552	KT Hengelo	074 - 427191
Jelsma, S.	Lampertheimstraat 8	7641	DP Wierden	05490 - 67479
Krabbenbos, J.	Oude Almeloseweg 123	7622	CD Borne	074 - 662939
Laarman, J.	Smidsweg 92	7441	EP Nijverdal	05486 - 17178
Lankheet, J.	Veldmaterstraat 91	7481	AC Haaksbergen	05427 - 15220
Noorland, G.J.	Pr. Ireneweg 4	7433	DE Schalkhaar	05700 - 25294
Overeem, R.	Molenstraat 108	7514	DW Enschede	
Protzman, B.D.J.	De Aalscholver 29A	7671	LK Vriezenveen	05499 - 64173
Scherpop, R. van de	Noorderstraat 57	7607	VT Almelo	05490 - 25151
Scherphof, A.T.	Rijstraat 30	7442	ER Nijverdal	05486 - 13129
Scholten, J.A.	Velweg 109	7533	XG Enschede	053 - 300728
Schuringa, W.	Grotestraat 66	7443	BK Nijverdal	05486 - 12676
Sinke, A.				
Slendebroek, W.F.	Markestraat 4	9531	EL Borger	
Slomp, R.	Oldenzaalsestr 74-II	7551	AL Hengelo	074 - 437564
Spel, R.	Merelweg 26	7442	CB Nijverdal	05486 - 14234
Veen, A. van der	Reviuslaan 57	8024	CD Zwolle	038 - 532984
Veenstra, R.E.	Hengveldbrink 108	7541	AX Enschede	053 - 761294
Velde, K. van der	Lankheethoek 13	7546	BP Enschede	053 - 775298
Vink, A.	Moellenbergstraat 47	7582	ZX Losser	05423 - 82499
Werken, W.J.A.				



=====

Een van de opmerkelijkste demonstraties was wel de demonstratie van Gerrit die een videokamera aan zijn Atom had gekoppeld. Met behulp van een AD-kaart kan een beeld naar de Atom over gebracht worden, om vervolgens op een printer te worden afgedrukt. Vele "pasfoto's" zijn er op deze manier uit de printer gerold.

Rudi Slomp introduceerde het wiskunde programma MATH. Voor de rekenaars onder ons een haast onmisbaar hulpmiddel. De kracht van het programma zit 'em vooral in de universele opbouw van de datafiles en de procedure achtige structuur van het programma. Op deze manier kan iedereen er heel eenvoudig routines bij maken (als dat nodig mocht zijn).

Geheel volgens afspraak kondigden de bestuursleden in mei aan dat zij per 1 juli zouden opstappen omdat zij tegen die tijd allen voor korte of lange tijd de regio zouden verlaten (stage, etc.). Geheel boven verwachting melden er zich zeer snel drie nieuwe kandidaten aan welke zonder problemen door de rest van de leden akkoord werden bevonden. Deze hebben in de bijeenkomst van juni, de laatste voor de grote vakantie, het roer van het oude bestuur overgenomen.

De financiering van de Z80-kaart lijkt toch wat te riskant voor twee regio's en er wordt besloten te trachten de landelijke organisatie in te schakelen voor productie en verspreiding van de CP/M kaart. Lukt dit niet dan zullen we zelf e.e.a. ter hand nemen maar dan moet er wel van tevoren betaald worden.

Na de vakantie een nieuw onderkomen, met de MEPA verhuist ook de Atomclub naar het gebouw van de MTS aan de rand van Almelo. De eerste keer wordt er bij het oude gebouw gewacht zodat iedereen de weg kan vinden.

In de vakantie is er overleg geweest tussen vertegenwoordigers van het bestuur en P.Huisken waarbij bleek dat men zeer geïnteresseerd was in het landelijk uitbrengen van de CP/M kaart.

Peter zal schema's, printontwerp, software en beschrijving overdragen en ontvangt daarvoor een vergoeding.

Onder het motto "Het moet uit de lengte of uit de breedte komen" komt er begin september een brief van het landelijk bestuur waarin min of meer te lezen staat dat GDOS niet zal worden uitgebracht in verband met de op komst zijnde Z80-kaart. Een bedankje aan de maker van GDOS stond er niet bij en ook was deze niet op de hoogte gebracht van deze beslissing die naar later bleek al in augustus bleek te zijn genomen.

H. de Ruiter, legde e.e.a. uit over een financiële mislukking rond de club-80-kolommenkaart: er zijn door enkele bestuursleden 400 kaarten incl. connectoren, onderdelen etc. besteld en tot dan toe slechts 60 stuks verkocht. Een strop van duizenden guldens die het federatie-bestuur noopt tot grote voorzichtigheid bij het uitbrengen van nieuwe club-artikelen. Het is nog onzeker of de federatie zich onder de aangegane contracten zal kunnen uitdraaien.

Het officiële deel van de volgende bijeenkomst (oktober) handelt voornamelijk over zaken die in de landelijke vergadering ter sprake zijn gekomen. Zie hiervoor het verslag van deze vergadering.

Helaas zijn wij als vertegenwoordigers van regio Twenthe een beetje laat in Utrecht, zodat wanneer we de vergaderruimte binnen komen er al enkele agendapunten besproken zijn.

Onder behandeling is de gang van zaken rond de voorraden.

Wat betreft de 80-kolommen-kaarten: de zaak met ITT, waar componenten voor 400 kaarten besteld waren is nog niet afgesloten. Volgens N.Stad maakt de federatie een goede kans om onder het contract uit te komen omdat ITT onjuiste informatie verstrekt heeft over bepaalde componenten en zich bovendien niet aan afspraken rond levertijden heeft gehouden. Het gaat hier om een investering van ca. fl. 20.000,-.

De AB-connectoren die al besteld en geleverd waren (200 sets male/female) worden, over de regio's verdeeld voor fl. 8,50 per set, in eerste instantie zijn er 15 sets per regio beschikbaar, eventuele nalevering is via de landelijke penningmeester te regelen.

Er zijn nog ca. 60 16k-kaarten in voorraad, die zullen tegen kostprijs (fl. 25,-) aan de regio's geleverd worden.

Boekjes (Jaargangen, AtomWare) worden niet meer bijgedrukt, er liggen nog ongeveer 600 exemplaren AtomNieuws van de jaargangen '82-'86.

Vaststelling contributie: huidige contributie en verdeelsleutel zullen gehandhaafd worden.

Volgend agendapunt betreft het instellen van een "Softwarecommissie" en vergoedingen voor "Supersoftware" (bv. GDOS, P-CHARME e.d.). Ten aanzien van dit laatste blijken de zaken nogal moeilijk te liggen: waar ligt de grens voor een hobbyclub. Er wordt hierover geen beslissing genomen. De bedoeling van een softwarecommissie is dat deze systeemprogrammatuur zal uittesten op goede samenwerking met het operating-system en algemeen ge-bruikte toolkits. Zo blijken er onder de federatieve 80-kolommen-soft een aantal statements van P-CHARME niet goed te werken. Verder moet de commissie trachten een standaard te formuleren waaraan software voor de Atom moet voldoen, dingen als zeropage-gebruik, jump-tabellen, het gebruik van vectoren i.p.v. recht-streekse aanroepen van routines, etc.

Over AtomNieuws valt ook een en ander te zeggen: Eerstens is er kritiek op het ontbreken van een schema bij het artikel over de PH-80-kolommenkaart en de aanwezigheid van lege bladzijden in een aantal exemplaren van het laatste Nieuws.

Een mededeling van de redactie: Alle schijven met kopij moeten naar Joop Ballijns gestuurd worden.

Regio Gelderland stelt voor dat de redactie van AtomNieuws een kaartje stuurt naar schrijvers van interessante artikelen als de regio niet heeft gezorgd voor het doorgeven van de kopij, zodat uiteindelijk alles wat de moeite waard is ook landelijk gepubliceerd wordt.

Omdat ik de eeprom programmeerdienst op mij genomen heb, wil ik de mogelijkheden en regels even op een rijtje zetten wat het wissen en programmeren betreft.

Mijn eepromprogrammer is ontworpen voor 27256's, maar ook andere typen in de 27xx serie kunnen worden geprogrammeerd. De programmer heeft 21 volt en 12.5 volt ter beschikking, zodat officieel er geen 25 volts typen zouden kunnen worden geprogrammeerd, maar in de praktijk lukt dit toch wel. Eprom's van het type 2532 of 2532A kunnen alleen met een tussenvoet geprogrammeerd worden (is aanwezig).

mogelijk zijn zonder meer:

2732A (programmeerspanning Vpp=21 volt)

2764 (Vpp=21 volt)

2764A (Vpp=12.5 volt)

27C64 (Vpp=21 volt)

27C64 (Vpp=12.5 volt) Intel,Thompson

27128 (Vpp=21 volt)

27128A (Vpp=12.5 volt)

27256 (Vpp= 12.5 volt) AMD,intel

De volgende typen eeprom's worden niet gegarandeerd:
2532, 2732 en 2532A, alle typen eeprom's van het merk Texas Instruments.

Met de eeprom programmer zijn al eens IC's geprogrammeerd van de volgende merken: Intel (2732 2764 2764A 27256), Hitachi (2532 2732 2764 27128) AMD (2764 27256), Fujitsu (27C32A 27C64), NEC (27C64). Dus als u een eeprom heeft van een van deze typen, dan zal het programmeren geen problemen geven. Het programmeren (en wissen) van eeprom's geschied uiteraard OP EIGEN RISICO!!!

Wat software, dan wel firmware betreft: de floating point ROM (met driewegwissel), en de bekende toolkits (SALFAA c2.0, P-charme, joxbox, Gags, enz.) zijn aanwezig, maar meer exotische dingen zoals eigen boxen of basicprogramma's moeten aangeleverd worden liefst op band (300 baud) of eventueel op schijf (GDOS).

Te programmeren eeprom's (hoeven niet leeg te zijn) kunnen aan mij worden gegeven (eventueel samen met programma) op de bijeenkomsten in Almelo, of u kunt bij mij thuis in Deventer langskomen. De te programmeren eeprom is meestal na enige dagen gereed. Per post (terug-) sturen kan ook, maar dan moet wel voldoende verpakkingsmateriaal en postzegels bij worden geleverd. Zoniet, dan wordt de eeprom niet terug gestuurd en kan dan op de afdelingsbijeenkomst worden afgehaald.

Het is niet mogelijk om eeprom's te kopen via de programmeer dienst, de club kan er namelijk niet aan beginnen om in eeproms te gaan handelen.

Rob van de Schepop

SOFTWARE: LPR statement

Rob van de Schepop

```

1 REM STATEMENT LPR
2 REM DIT PROGRAMMA IS AFKOMSTIG UIT DE ROMMELDOOS, MAAR
3 REM WORDT NOG STEEDS GEBRUIKT (IN EEN EIGEN TOOLBOX).
4 REM HET STATEMENT (LPR = LINEPRINTER) STUURT EEN PROGRAMMA
5 REM NAAR DE PRINTER MET HET BEELDSCHERM UITGESCHAKELD.
6 REM VAN TE VOREN WORDT DE PAPIERLENGTE INGESTELD EN DE
7 REM PERFORATIESKIP. DE SYNTAX IS IDENTIEK AAN LIST, DUS
8 REM LPR 234,567
9 REM LPR ,789
10 REM IS ALLEMAAL TOEGESTAAN.
11 REM OMDAT HET BEELDSCHERM UIT STAAT, WORDT DE OVERDRACHT-
12 REM SNELHEID ALLEEN DOOR DE PRINTER BEPAALD. BIJ EEN PRIN-
13 REM TER MET EEN INTERNE BUFFER ZAL DE CURSOR BIJ KLEINERE
14 REM LISTINGS BIJNA METEEN TERUGKEREN.
15 REM
16 REM
17 REM *****
18 REM
19 REM
20 IN."WAAR CODE",A
21 DIM XX20;FOR I=0TO20;XX1=A;N.
22 P.$21;FOR I=1TO2;P=A;?#E8=0;[
23 50:XX0 BIT#B801
24 60 BPL XX1
25 70 JSR#F7D1
26 75;J;$P="PRINTER OFFLINE";P=P+LEN(P);[NOP
27 80 BRK
28 90:XX1 LDA#E0;ORA#80;STA#E0 SCHERM UIT
29 100 LDA#02;JSR#FEFB PRINTER AAN
30 120 LDA#1B;JSR#FEFB PAGINA LENGTE ELF INCH
31 130 LDA#43;JSR#FEFB
32 140 LDA#00;JSR#FEFB
33 150 LDA#0B;JSR#FEFB
34 160 LDA#1B;JSR#FEFB PERF SKIP VIJF REGELS
35 170 LDA#4E;JSR#FEFB
36 180 LDA#05;JSR#FEFB
37 190\START ORIGINELE LIST ROUTINE
38 210 LDA#00
39 220 JSR#C97C BASIC ACCU OP 0
40 230 LDA#FF
41 240 JSR#C97C
42 250 STA#04
43 260 LDY#7F
44 270 STY#26
45 280 JSR#C465
46 290 BCC XX7
47 300 JSR#C231
48 310 BCS XX8
49 320 JSR#C465
50 330:XX9 LDX#01
51 340 STX#04
52 380 JSR#C4E4
53 390 JSR#C62E
54 400 BCC XX2

```

OUDE DOOS: GaltonGert Jan Noorland

Een jaar of drie geleden was de GAGS-rom nog niet gemaakt en verscheen er een artikel onder de naam 'ATOM PLOTTING POWER' in het blad YOUR COMPUTER. Dit was naast de JOSBOX de eerste mogelijkheid om eenvoudig met bewegende voorwerpen te werken in een grafische mode. In ACORN NIEUWS jrg.3 nr.2 blz. 58 wordt het artikel beschreven. Ik was erg enthousiast, omdat de machinecode na het basicprogramma staat en je dus geen toolkit of schakelkaart nodig had. In dit programma gebruikt men de JOSBOX alleen om in CLEAR4 tekst te kunnen schrijven.

Voor liefhebbers is bij mij nog het ATOM PLOTTING POWER sourceprogramma en een demo te verkrijgen indien men dit wenst. Bij dit programma staat de machinecode van af adres #3900 tot een adres wat ik niet meer weet. Vandaar dat ik het gesaved heb tot #3C00.

Nu het programma zelf. In het onderwijs wordt om een statistische verdeling te demonstren gebruikt gemaakt van een spijkerbord waarover knikkers of kogeltjes naar beneden vallen via de spijkers. Bij elk spijkertje kan de knikker naar links of naar rechts. De verdeling die dan ontstaat noemt men een 'normaalverdeling'.

Voor het aantal knikkers heb ik tot maximum 800 genomen, omdat bij een groter aantal de nauwkerigheid niet veel groter wordt en het te lang gaat duren. Wel wordt bij meer knikkers het histogram iets breder, maar door schaling gaat dit effect weer verloren.

Nadat het gewenste aantal knikkers gevallen is krijgt men van elk bakje het aantal knikkers dat erin gevallen is. De leerling zou nu zelf het histogram kunnen maken. Vervolgens wordt over het histogram de berekende grafiek van de verdeling getrokken om te kijken of deze afwijkt t.o.v. het histogram. Bij weinig knikkers zal de afwijking het grootst zijn in een aantal gevallen.

Dit programma is geschreven toen de PCHARME nog niet aanwezig was, vandaar dat er nog weinig structuur in het programma zit. Misschien heeft iemand er iets aan.

Zoja, dan is de oude doos niet voor niets opengegaan. Voor leuke verbeteringen kun je altijd bij mij terecht.



Leer jezelf Lisp - Een samenvatting van 4 artikelen van de hand van Dick Pountain, verschenen in PCWorld augustus..december 1984.

Noot: er is niet naar gestreefd de artikelenserie geheel weer te geven, hier en daar zijn gedeelten toegespitst op het gebruik van Atom-Lisp. Bovendien ontbreken er 2 van de zes afleveringen in mijn collectie. Naar de daarin behandelde materie moest ik radenderwijs mijn eigen verhaaltje schrijven.

De meeste kunstmatige-intelligentie-talen werden ontwikkeld uit en dikwijls geschreven in een Lisp-dialect. In het eerste deel van de leer-jezelf-Lisp-serie gaat Dick Pountain in op de concepten van List-processing:

In de nu veertig-jarige geschiedenis van digitale computers zijn er allerlei prachtige ideeën ontstaan, maar het prachtigst van allemaal zou wel eens "List-processing" (of Lisp!) kunnen zijn.

In de een of andere vorm is List-processing de overheersende programmeer-techniek geworden in het onderzoek naar Kunstmatige Intelligentie (KI), maar het is totaal niet doorgedrongen in het gebied van zakelijke programmatuur en onbekend voor de meeste microcomputer-enthousiasten. Er is een enorme kloof ontstaan, bijna te breed voor wederzijds begrip, tussen KI-programmeurs die zich vastbeten in Lisp (of Prolog of POP-2) en anderen die begonnen te programmeren in Basic en later mogelijk in Assembler, Pascal, Forth of C.

In deze serie zullen we proberen de principes uit te leggen waarop Lisp, moeder en vader van list-processing talen (en volgens velen nog steeds de meest elegante hiervan) gebaseerd is.

Verrassend genoeg is Lisp een van de oudste computertalen, ontwikkeld tussen 1960 en 1965 op het Massachusetts Institute of Technology (MIT) door John McCarthy. Dat het een betrekkelijk inpopulaire taal is kon er wel aan liggen dat het niet erg geschikt is voor het routinematig behandelen van zakelijke gegevens omdat het inefficiënt met tijd en geheugenruimte omspringt. Alle modernere KI-talen zoals Prolog, Logo en POP-2 zijn sterk beïnvloed door Lisp (en werden vaak ook in Lisp geschreven).

Beschikbare boeken en handboeken over Lisp onderschatten vaak de mate van vreemdheid waarmee een Basic-programmeur tegen de gebruikte concepten aankijkt en zijn om die reden niet zo nuttig als zou kunnen. We hopen deze leemte op te vullen bij de zo belangrijke eerste kennismaking.

Concepten

Het is zeer verleidelijk om bij een inleiding in een computertaal maar gelijk in de voorbeelden te duiken, al was het maar een programma om 'Dag Meneer' op het scherm te laten

den opgeslagen en later met het statement VAL(..) worden omgewerkt.

Basic kent ook de READ ... DATA constructie die ons de mogelijkheid geeft om:

```
100 READ EGG$,EGG,MILK$,MILK,COFFEE$,COFFEE
```

```
200 DATA "Eggs",12,"Milk",4,"Coffee",1
```

te schrijven maar het wordt wel zeer lastig om nog iets aan de lijst te veranderen daar we onze toevlucht zouden moeten zoeken tot "zelf-modificerende" code met behulp van PEEK en POKE!

In ieder geval was dit nog maar het topje van de ijsberg: Wat te doen met een meer gestructureerde lijst zoals:

```
Maandag lunch (tonijn erwten truffels)
```

```
Maandag diner (oesters fazant spinazie)
```

```
Dinsdag lunch (rouwe-ham ham kronkel (bladerdeeg abricozen  
amandelen) roquefort)
```

In deze lijst zijn sommige van de gegevens zelf lijsten en een van die sublijsten (kronkel) heeft weer een sublijst.

Om hiermee in Basic om te gaan zouden we een drie-dimensionaal string-array moeten definiëren (met dimensies voor Dag, Menu, en Ingredienten) waarin de meeste gereserveerde strings leeg zouden zijn met als gevolg een flinke verspilling van geheugenruimte.

Als het enkel om menu's en boodschappenlijsten ging dan was er niets aan de hand. Maar het is niet onwaarschijnlijk dat de lezer al de analogie heeft gezien met Nederlandse zinnen (deze inclus) die voorgesteld kunnen worden als lijsten met precies deze soort structuur, waarmee de interesse van KI-programmeurs in list processing verklaard moge zijn. Als je er eenmaal mee begint om tekst op een beschaafde manier te manipuleren (bijvoorbeeld een zin verdelen in benoemde zinsdelen) dan wordt een taal die in staat is om direkt met lijsten om te gaan een voorwaarde; de hoeveelheid gehussel met strings en karakters die hiervoor in Pascal of Basic nodig is zijn zou iemand er van kunnen weerhouden ooit tot het wezen van een probleem door te dringen.

Het menu-voorbeeld van daarnet laat ook zien hoe lijsten gebruikt kunnen worden om structuur aan te brengen in concepten: een kronkel is op het ene niveau een gerecht, maar op een meer

gedetailleerd niveau een lijst van ingredienten. Op deze manier kunnen complexe boom-structuren van willekeurige orde keurig voorgesteld worden door een een-dimensionale rij van symbolen, hetgeen essentieel is voor invoer in de meeste computers.

Er is een notatie nodig waardoor het voor computers mogelijk wordt gemaakt om lijsten te herkennen als gegevens van een hoger niveau dan strings en getallen: strings en getallen (en lijsten!) worden de bouwstenen waaruit een lijst is samengesteld.

Bovendien moeten we een aantal bewerkingen definiëren die op lijsten kunnen worden uitgevoerd (net zoals MID\$, LEFT\$ en RIGHT\$ op strings kunnen worden losgelaten). Je zou een eindeloze waslijst van mogelijke bewerkingen kunnen opstellen (bv. vervang het vijfde gegeven in een lijst met het een-na-laatste gegeven van een andere lijst) maar in een computertaal gebiedt zuinigheid ons een minimaal aantal fundamentele bewerkingen te kiezen, plus een manier om deze aan elkaar te knopen om meer ingewikkelde taken te volbrengen.

'rest' zouden moeten herhalen (afhankelijk van de manier waarop we combinaties van de bewerkingen gedefinieerd hebben).
Bijvoorbeeld:

'eerste' -> 'rest' -> 'rest' -> (Eieren Melk Koffie Suiker)
zou precies goed uitkomen als de pijltjes betekenen: 'pas dit toe op alles dat rechts van me staat' en wanneer deze bewerkingen van rechts naar links zouden worden uitgevoerd.

Om een onderdeel van een lijst te verwijderen moet je eerst een nieuwe lijst 'construeren' waarin het te verwijderen element ontbreekt om daarna met 'eerste' en 'rest' de elementen die je wel wilt toe te voegen.

Op dit punt loopt de analogie met een concrete boodschappenlijst op de klippen, een mens zou immers het betreffende onderdeel gewoon doorstrepen. Zoals we echter zullen zien, doet LISP het op dezelfde manier. Als oefening zou je kunnen proberen een manier te formuleren om Melk uit (Eieren Boter Melk Koffie) te verwijderen, met gebruikmaking van bovengenoemde pijltjes-afspraken; neem hierbij aan dat 'construeren' -> a (b) oplevert: (a b).

De soort vergelijkende tests die we nodig hebben moeten ons vertellen of twee elementen gelijk zijn, of een lijst leeg is, en of een element van een lijst al of niet een lijst is.

Het zou erg handig zijn om nog een paar primitieve bewerkingen tot onze beschikking te hebben (bv. 'laatste' om het laatste element van een lijst te verkrijgen) en sommige list-processing talen beschikken hier ook over, maar het zijn luxe-toevoegingen in die zin dat ze kunnen worden samengesteld uit combinaties van de drie genoemde fundamentele bewerkingen.

Er is geen goede reden te bedenken waarom list-processing primitieven niet zouden kunnen worden toegevoegd aan conventionele talen zoals Basic (er zijn verscheidene Pascal-achtige talen die erover beschikken).

Lisp benadert deze zaken zeer radicaal: Notatie in lijsten wordt overal gebruikt; Lisp programma's zijn zelf lijsten en kunnen als data in andere Lisp programma's worden ingevoerd (nog een reden voor de aantrekkelijkheid van deze taal voor KI-onderzoekers die zelf-modificerende programmatuur nodig hebben).

Lisp programmeren bestaat voornamelijk uit het definiëren en evalueren van functies, op een manier die veel dichter bij wiskunde staat dan de manier waarop opdrachten in een Basic programma gecombineerd worden. De bekende controle-structuren als IF .. THEN, FOR .. NEXT en GOTO kennen in zuiver Lisp geen alternatieven, hierin wordt controle over het programma voornamelijk bereikt door de lijst-structuur zelf en door recursie: dus functies die zichzelf aanroepen.

Tussen twee haakjes, opdat de lezer niet denkt dat hij hiervan niets heeft kunnen opsteken: het eenvoudigste programma om 'Dag Meneer' te produceren is:

'(Dag Meneer)
als je niet om een paar haakjes geeft. Geef je daar wel om, denk dan nog maar eens drie keer na voordat je verder gaat met Lisp!

Dit programma is eigenlijk als aanloop bedoeld voor een groter programma waarin de eiwitsynthese in een ribosoom grafisch wordt weergegeven. De tijd heeft me tot nu toe ontbroken dit idee verder uit te werken, maar dat komt misschien nog een keer. Zoals je misschien al weet is een eiwit opgebouwd uit bouwsteentjes, die men aminozuren noemt. Er komen ongeveer 20 verschillende aminozuren voor in de eiwitten.

In het DNA molekuul dat zich in de genen op de chromosomen bevindt liggen de erfelijke eigenschappen van de mens en de opbouw van de eiwitten vast in een code. Deze code wordt afgelezen met behulp van het een boodschappermolekuul (=mRNA) en naar het ribosoom (= de eiwitfabriek) gebracht.

Dit programma geeft nu de mogelijkheid om na te gaan welk aminozuur bij een bepaalde code hoort. Een triplet oftewel een codon is combinatie van drie letters, dit zijn de beginletters van de basen, die afgelezen worden van het DNA. De basen worden in het programma met naam en letter genoemd.

Ook kan het programma uit een lange rij basen, het startcodon zoeken en vervolgens de aminozuurvolgorde bepalen tot het afsluitcodon gevonden wordt.

Dit programma is dus eigenlijk een decodeer programma en meer niet. Voor biochemie, biologie studenten en andere geïnteresseerden nuttig zijn i.p.v. elke keer een tabel te moeten raadplegen voor elke letter, alles automatisch te laten bepalen.

Ik ben me bewust dat niet iedereen zich hiermee bezig houdt, maar je weet maar nooit wie je er een plezier mee doet door het te publiceren en misschien bouwt er wel iemand op voort. In dat laatste geval zou ik er ook graag iets over horen.



HARDWARE: Het misterie van lange kabelsA. van der Veen

Aan de hand van het schema van Gerrit Hillebrand heb ik onlangs het GDOS floppy disk interface gebouwd. Zonder noemenswaardige problemen begonnen mijn (Philips) drive en de GDOS kaart hun taak uit te voeren. Maar al gauw zag ik foutjes optreden in programma's die ik van schijf geladen had. Door in een lus als maar eenzelfde picture vanaf schijf te laden werd al snel duidelijk dat er af en toe wat mis ging bij het lezen.

Na enig zoekwerk of dit door de drive of door de interface kaart werd veroorzaakt, bleek dat de lengte van de bandkabel tussen de PL6 connector en de interface print van invloed was op de storingen.

Om er achter te komen wat hiervan de reden was ben ik m.b.v. een oscilloscoop gaan zoeken. Wat ik toen allemaal zag deed me wel even schrikken. Wat was namelijk het geval?

Tussen de PL6 connector van de ATOM en de GDOS kaart zat een bandkabel van ca. 0,5 meter. Gemeten tussen de voedingsnul van de ATOM en de voedingsnul van de interfacekaart, zag ik een spanning met pieken van wel enkele volts top-top. Dat kon natuurlijk nooit goed gaan.

Ik kon zelfs vaststellen dat de adresdecoder op de GDOS-kaart tijdens het optreden van de pieken werd gede-adresseerd. Om er achter te komen wat de oorzaak van deze pieken was ben ik nog wat verder gaan zoeken.

Uit het schema van de ATOM blijkt dat er op PL6 slechts 2 connectorpunten voor de logische nul worden gebruikt (32 a/b). De gedachte dat het iets te maken kon hebben met overgangswaerstand of leidingswaerstand van de bandkabel bleek niet te kloppen, gezien de relatief kleine stroom die er in de nulleiding loopt. Maar wat dan wel?

Ter verduidelijking heb ik een deel van de verbindingen tussen de ATOM en de GDOS-kaart in bijgaand schema aangegeven en tevens de ingangen en uitgangen van de buffers vereenvoudigd erbij getekend.

Naast ohmse weerstand heeft een geleider ook zelfinductie. Door een stroomverandering ontstaat er een spanning volgens de formule $U = -L * di/dt$. L is de zelfinductie en di/dt is de stroomverandering per tijdseenheid. De hoogste pieken kon ik meten op het moment dat de GDOS kaart werd geadresseerd en de databusbuffer werd geopend. Er gebeurt dan het volgende: op het moment dat de databusbuffer geactiveerd wordt, zal deze, afhankelijk van het datapatroon, de databuslijnen in de bandkabel hoog of laag trekken. Dat betekent dat op dat moment de kabelcapaciteiten worden geladen of ontladen via de uitgangen van de databusbuffer. De retourweg voor deze stroom loopt via de logische nul verbinding tussen de GDOS kaart en de ATOM.

Deze stroom geeft over de leidingweerstand wel enige spanningsval, maar dat leidt niet tot problemen. De hoge piekspanning ontstaat doordat deze stroom in zeer korte tijd van nul tot een bepaalde waarde verandert. Deze verandering geeft samen met de zelfinductie een prachtige spanningspiek. T.g.v. de aanwezige capaciteiten, zelfinductie en leidingweerstand zal deze piek vervolgens gedempt gaan uitslingeren.

HARDWARE: Elektuur 80-kolommen kaart
met High- en Low Video

Jan Biel

In Acorntjesbrood 4.3 bladzijde 6 e.v. heeft u een artikel kunnen lezen over de 80-kolommen kaart van Peter Huisken. Deze kaart biedt de mogelijkheid tot het weergeven van tekst in normaal- en 'gedimd' video (ook wel high & low video genoemd). De binnen onze club ook veel toegepaste 80-kolommen kaart van Elektuur biedt deze mogelijkheid echter niet. Aan de hand van het schema van Peter hebben Ronald Boers en ondergetekende de Elektuur kaart nog eens nader bekeken en zijn tot de conclusie gekomen dat ook deze kaart relatief eenvoudig omgebouwd kan worden naar high & low video. Eenvoudig, daar slechts twee extra IC'S nodig zijn, die op bestaande IC's gestapeld worden. Aan de hand van het aan het einde van dit artikel geplaatste schema zal ik nu de ombouw toelichten. Een opmerking vooraf: de EPROM met de karakterset (IC19 op de Elektuur kaart) moet opnieuw geprogrammeerd worden.

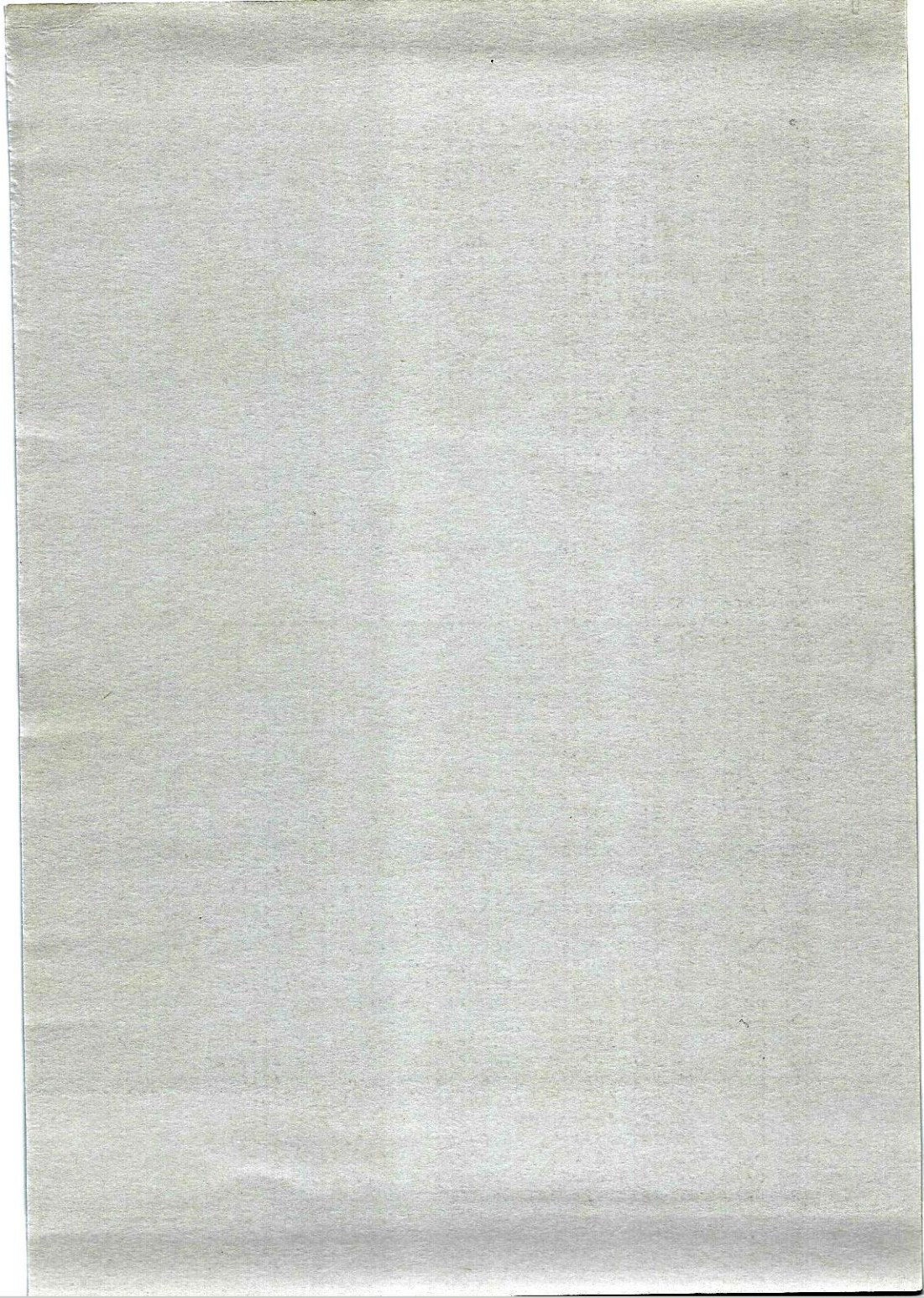
- Op de kaart moeten twee printsporen doorgekraast worden:
 1. Het spoor tussen IC18 pen 16 en IC19 pen 21
 2. Het spoor tussen N22 pen 6 en weerstand R4
- Pen 21 van IC19 wordt aan massa gelegd. (B.v. pen 20 van IC19)
- Boven op IC17 (LS175) wordt een extra LS175 geplaatst. Hiervan worden de pennen 8 en 16 aan de overeenkomstige pennen van IC17 gesoldeerd. Hetzelfde geldt voor de pennen 1 en 9 (nRESET en CLOCK).
- Boven op IC3 (LS04) wordt een IC van het type LS06 gesoldeerd (door pen 7 en 14 door te verbinden).
- Van de extra LS175 wordt nu pen 2 (= Q-uitgang) verbonden met pen 3 van de LS06.
- Pen 12 van N31 wordt verbonden met pen 1 van de LS06.
- Pen 16 van IC18 wordt verbonden met pen 4 (= D-ingang) van de extra LS175.
- De pennen 2 en 4 van de LS06 worden via een weerstand van 470 ohm met elkaar verbonden.
- Dit gemeenschappelijke punt wordt verbonden met R4 (aan de kant van het doorgekraste spoor).

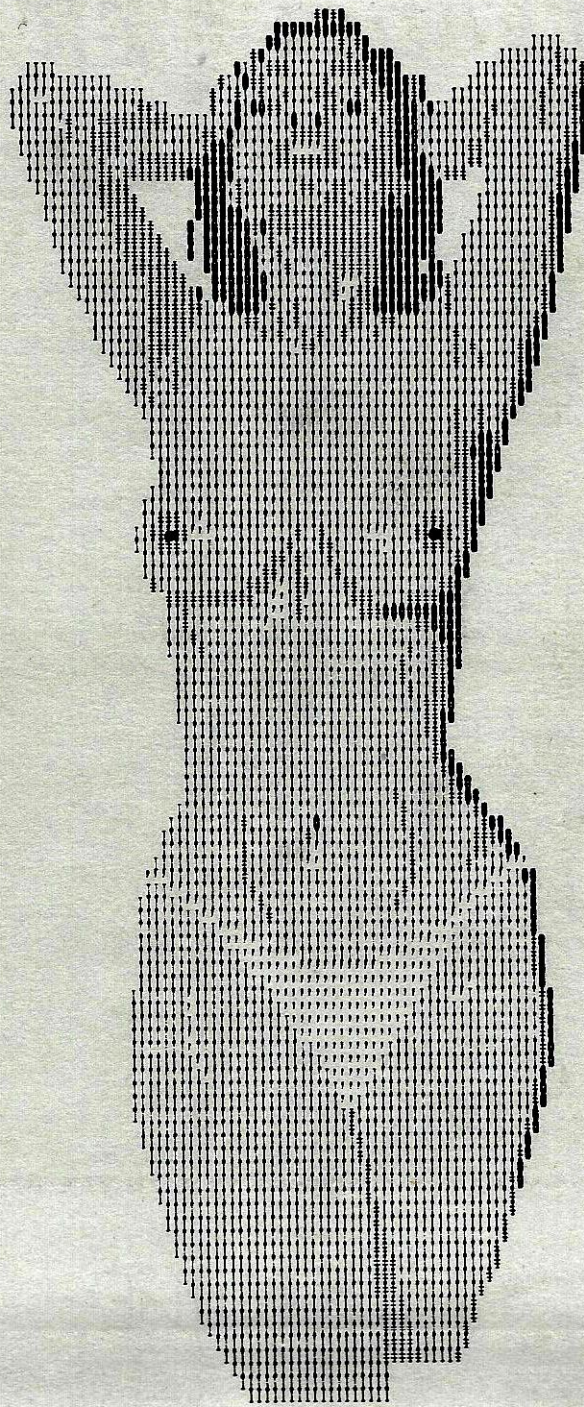
Hiermee is de ombouw van de Elektuur 80-kolommen kaart voltooid.

Om in het gewijzigde schema duidelijk aan te geven wat de extra componenten zijn heb ik deze als volgt genummerd:

- de extra LS175 is IC30 (de hiervan gebruikte flipflop FF30)
- de LS06 is IC31 (met als gebruikte poorten N40 en N41)
- de weerstand van 470 ohm R30

Met de extra D-flipflop wordt het video-signaal 1 karakter-tijd vertraagd, waardoor een gedimd video-signaal ontstaat. Als alles goed gaat zal de inhoud van de gewijzigde karakter-EPROM op de regio avond van april via de balk verspeid





530 DATA "1 914=101=3051==4'==413=3061==101=00"
 540 DATA "2 91=418=40518=314=4051==91==0"
 550 DATA "2 91==18=601714=4013=2 91=00"
 560 DATA "3 814=8 401=71==11==11=14=405 91=00"
 570 DATA "4 715=7 401=1013=14=0=504 91=00"
 580 DATA "4 815=6 603114=804 91=00"
 590 DATA "5 815=4 01 60=014=803 101=00"
 600 DATA "5 816=3 01 6011=1111=703 111=0"
 610 DATA "6 816=2 01 7011=517=703 101==0"
 620 DATA "6 916=3 7031=516=501 001 111==0"
 630 DATA "7 1015=3 7081=' '1==80=1210"
 640 DATA "8 916=01 7081=' '1==801=111=0"
 650 DATA "9 817=31401=11=613=6031=111=0"
 660 DATA "10 716=61==11=1=61==11=171=0"
 670 DATA "10 715=91=1=11=1=11=31=91=81=00"
 680 DATA "11 614=141'==171=1=51=00"
 690 DATA "12 51==1=341=11=41=00"
 700 DATA "13 41=11=421=0"
 710 DATA "14 11=31=411=00"
 720 DATA "15 11=441=0"
 730 DATA "16 391=11=11=0" 770 DATA "18 3713=30"
 740 DATA "17 381=11=110" 780 DATA "18 381==00"
 750 DATA "17 3813=00=0" 790 DATA "17 391==00"
 760 DATA "17 381==40" 800 DATA "16 401=00"

810 DATA "16 113=11' 2713=11=00"
 820 DATA "15 11==3' 101=11=513' 31== '==1=0"
 830 DATA "15 313=131=31==61' 413=11=0"
 840 DATA "16 161==1=31=151=0"
 850 DATA "16 151==11=41=14100"
 860 DATA "17 =1113=1' 1==41==101==0"
 870 DATA "19 11=11' '1=513=414=30"
 880 DATA "19 11=101' '1=713=100"
 890 DATA "19 11=91' 181==40"
 900 DATA "19 31=30130", "19 41=251=11==00", "20 281=31==00"
 910 DATA "20 281=31==0", "20 291=11==0", "20 3113=0"
 920 DATA "20 311==00", "21 301==0", "21 311=0"
 930 DATA "21 311=00", "21 321=00", "21 331=00"
 940 DATA "21 261=61==00", "20 281=81=0", "19 91=810101=91=00"
 950 DATA "18 181=121=713=00", "18 101=81' 111=71==1100"
 960 DATA "17 111=71' '111=61=413'", "16 3' 91=201=91' '1100"
 970 DATA "16 31' '81=181=81' '510", "15 61' '71=161=71' '710"
 980 DATA "15 81' '61=221' '810", "15 1013' 2413' 910"
 990 DATA "15 1114' 1915' 101=0", "15 1317' 1018' 111=0"
 1000 DATA "15 14123' 121=0", "15 16120' 51' 810"
 1010 DATA "14 71' 10118' 41' '91=0", "14 71' 11116' 161=0"
 1020 DATA "14 20114' 171=0", "14 71' 13112' 181=0"
 1030 DATA "14 71' 1519' 191=0"
 1040 DATA "14 81' 1517' 201=0", "14 91' 1515' 2110", "14 2613' 211=0"
 1050 DATA "14 261' '221=0", "14 271=2210", "15 261=211=0"
 1060 DATA "15 271=2010", "16 261=191=0", "16 271=1810"
 1070 DATA "17 261=171=0", "17 261=171=", "18 261=161=" "
 1080 DATA "18 261=151=", "19 251==141=", "19 251==131=" "
 1090 DATA "20 2413=121=", "20 2413=111=", "21 2313=101=" "
 1100 DATA "22 2313=81=", "22 2313=81=", "23 2214=61=" "
 1110 DATA "24 2114=61=", "25 2015=41=", "26 201" "
 1120 DATA "27 191", "28 181", "STOP"

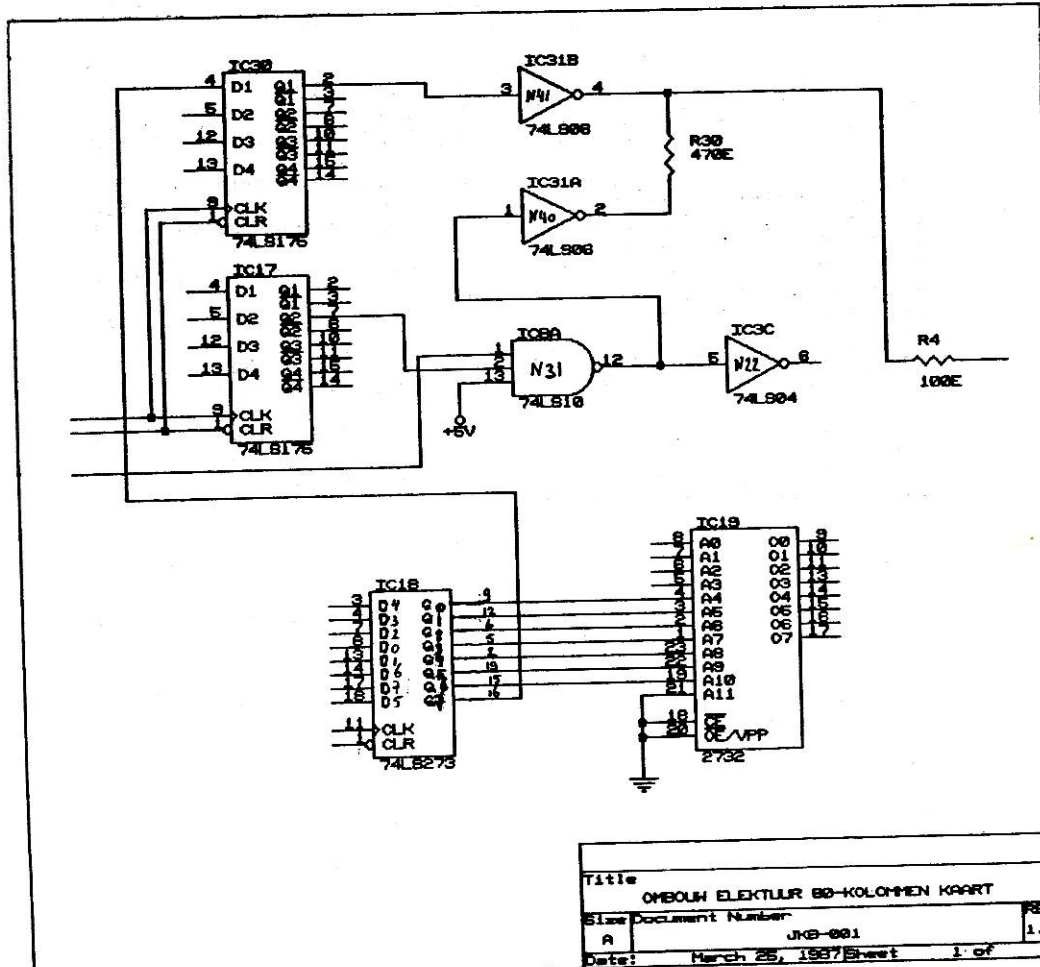
worden. Hierin zijn de oorspronkelijke 'block-graphics' vervangen door low-video karakters. Overigens, de nieuwe karakterset komt overeen met de karakterset van de Electron (kwa karakter opbouw).

Over de balk zal tevens een nieuwe versie van de 80-kolommen soft van Bram Poot verspreid worden (versie 2.11), geschreven in SALFAA C2.0. Hiermee kan men omschakelen tussen high en low video met behulp van CTRL-a (low video) en CTRL-z (high video).

LET OP: het 'printen' van de low video karakters wordt bereikt door in write-character routine #80 bij de ASCII-waarde van het te 'printen' karakter op te tellen. Bij gebruik van de COPY-toets in combinatie met low video karakters op het scherm zal na afloop in het geheugen de ASCII-waarde vermeerderd met #80 staan (de read-character routine is namelijk niet aangepast !!).

Veel plezier met de ombouw van de Elektuur 80-kolommen kaart.

Jan Biel



Ter illustratie enkele getallen:

Voor een ader in de bandkabel kan volgens een vuistregel worden gerekend met een zelfinductie van 10 nH/cm.

Laten we voor de stijgtijd van een bufferuitgang eens 10 ns rekenen en verder aannemen dat elke databuslijn 2 mA gaat voeren, zodat er 16 mA in de logische nul gaat lopen.

De hierboven vermelde formule levert dan op:

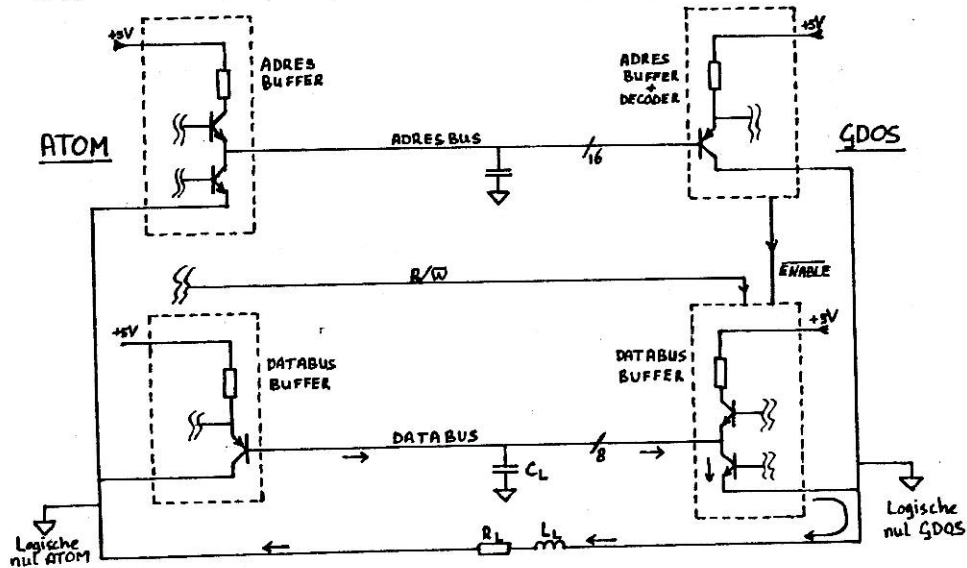
$$U = -500 \text{ nH} \times 16 \text{ mA} / 10 \text{ ns} = -800 \text{ mV.}$$

Bij het uitslingeren draait de polariteit om, zodat een top-top waarde van meer dan 1,5 volt ontstaat. Dit komt aardig overeen met wat ik heb gemeten.

Wat valt hier nu tegen te doen?

- Kort de kabel in tot de kortste nog praktische lengte. Hierdoor dalen zowel de zelfinductie als de parasitaire capaciteiten.
- Kies busbuffers met langere stijgtijd. Door de 74LS245 te vervangen door een 74HCT245 kon ik een duidelijke vermindering van de stoorpiek constateren. Dit is toe te schrijven aan de hogere uitgangsweerstand van de CMOS uitgang, wat leidt tot minder steile flanken.
- Gebruik de nog vrije verbindingen op PL6 als logischenul leidingen. Leg ook in de ATOM en op de GDOS-kaart deze punten aan de nul. Door de parallelschakeling van meerdere aders zal de zelfinductie evenredig dalen. (In feite had ACORN voor elke signaalader ook een retourader moeten nemen, zoals b.v. op de printeraanpluiging wie het geval is.)
- Neem evt. een kleine serieweerstand op in de databuslijnen (enkele tientallen ohms). Ook dit geeft minder steile flanken en dempt de uitslingering van de LC-kring.

Met bovenstaande maatregelen heb ik de leesproblemen volledig kunnen verhelpen. Trouwens, is het nu duidelijk waarom aan de originele ATOM-drive zo'n akelig lastig kort bandkabeltje zit?



SOFTWARE: Dragcar design

Gert Jan Noorland

Voor enthousiaste ontwerpers is er nu de kans om zelf een dragcar te ontwerpen. Het aardige is echter dat de computer zelf een redelijk sterk wagentje ontwerpt en deze gaat aan het einde van het programma een wedstrijdje met het door jou ontworpen wagentje aan. Ook corrigeert de computer je met aardige boodschappen, als er onmogelijke grootheden ingevoerd worden. Het programma is van oorsprong van een canadees, vandaar dat in het programma bij de in en uitvoer omrekeningen zijn gemaakt naar SI eenheden. Zelf heb ik het omgezet voor de ATOM en de layout verfraait in vergelijking met het originele programma.

1 foot = 30,48 cm en 1 mile = 1,6093 km.

Het programma vraagt de volgende eigenschappen van je dragcar :

1. Het motorvermogen in PK.

2. De achterasoverbrenging, X:1

Dit is het aantal omwentelingen van de achteras maakt, als de motor 1 omwenteling maakt.

3. De bandbreedte van de achterband in cm.

4. De diameter van het achterwiel in cm.

Met behulp van formules, die in de praktijk bepaald zijn, worden deze basisgegevens van de dragcar omgerekend naar:

a. De massa.

b. De windweerstand.

c. De maximale versnelling.

d. De factor : toerental/vermogen.

Deze nieuw berekende gegevens bepalen onder andere de snelheid, die gehaald wordt. Neemt men bijvoorbeeld een te groot vermogen, dan treedt een zgn. 'wheelspin' op oftewel je slipt en dan kom je niet vooruit!

De wedstrijd bepaalt de kwaliteit van de door jouw ontworpen auto. Verlies je, dan kunt misschien door de aanschaf van een eigen CAD/CAM systeem te overwegen, de zaken gaan stroomlijnen. Goedkoper is het gewoon opnieuw te proberen met je 'oude' vertrouwde atompje.

Als je wint dan heb je het idee een goede DESIGNER te zijn. Dit programma komt ook weer uit de oude doos en werkt dus op een standaard ATOM zonder toolkits. Met behulp van de PCHARME zou het geheel erg goed te structureren zijn, maar daar heb ik geen tijd voor over.

Veel succes met je eigen ontwerp.



LISPWARE: Haiku : het toeval der schoonheid.

P. Klomp

Nu ja, wat heet schoonheid.. Ik ken mensen die als ze de resultaten van dit programma zien hard lachen, drie gedichtjes bekijken en zeggen dat er alleen maar onzin uitkomt. Maar de aanhouder wint.

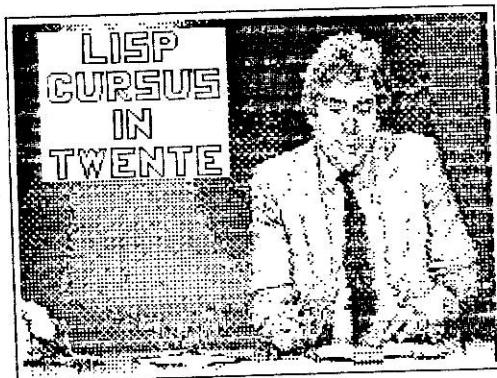
Een grote korrel zout lijkt wel op z'n plaats. Oorspronkelijk was het programma in Microsoft-Basic geschreven, aangepast voor BasiCode. Het omzetten van de vele string-manipulaties verliep moeizaam en nog steeds gaat er nu en dan iets fout in mijn Atom-Basic versie. Vandaar dat ik de Lisp uit de kast gehaald heb om het daarmee te proberen. En met betrekkelijk weinig inspanning (met name: zonder rekenwerk aan allerlei string-pointers) is dit programma ontstaan.

Hoe draait men het? Als volgt:

laad eerst "LISP1" op #8200, dan *RUN "HAIKU.lsp".

De Lisp-interpretor vraagt nu: "~EVALUATE:" waarop het antwoord (MAIN) dient te luiden. De rest wijst zichzelf, verwacht niet serieus een echt poem!

Door de volgordes in PATTERN te veranderen kan men experimenteren met de gevolgen van bepaalde woord-volgordes, het getal waarmee elke lijst begint staat voor het teken waarmee de zin als hij klaar is op het scherm afgesloten wordt; als er niets achter de zin moet staan vul dan 32 in. De data-lijsten NOUN, ART, VERB, ADJ en PREP kunnen met wat creativiteit omgeschreven worden zo dat het programma i.p.v. mooie, vieze gedichtjes produceert. Hiertoe moeten alle mooie woorden door vieze woorden worden vervangen. Hou het echter wel in het engels want om met nederlandse lidwoorden om te gaan is informatie nodig omtrent het geslacht van de zelfstandige naamwoorden.



Wat betreft de notatie, die hebben we grotendeels al klaar. In het menu-voorbeeld heb ik haakjes '(' en ')'' gebruikt om sublijsten in een lijst te laten zien en spaties om de gegevens in een lijst van elkaar te scheiden. Als we stellen dat alles tussen haakjes een lijst voorstelt en dat de onderdelen door spaties worden gescheiden dan zouden we de voorbeelden als volgt herschrijven:

(Eieren Melk Koffie Cornflakes) en:

(Maandag lunch (tonijn erwten truffels) Maandag diner (oesters fazant spinazie) Dinsdag lunch (rouwe-ham ham kronkel (bladerdeeg abricozen amandelen) roquefort)).

De onderdelen kunnen getallen, woorden of sublijsten zijn, of een combinatie. Vierkante haken of accolades hadden ook gekozen kunnen worden en ook een ander scheidingsteken in plaats van <spatie>, bijvoorbeeld een komma zou voldoen. Het voordeel van <spatie> is echter dat dit teken in geschreven menselijke taal gebruikt wordt.

Het menu-voorbeeld zit niet precies zo in elkaar als we zouden willen (bv. Maandag en lunch zijn gescheiden onderdelen) maar daar maken we voorlopig geen punt van. Als we het eenvoudiger boodschappen-voorbeeld nemen dan kunnen de haakjes zien als het papier waarop de lijst geschreven is. Het voordeel hiervan is dat we '()' kunnen beschouwen als een leeg stuk papier, wat later nuttig zal blijken. Deze analogie gaat niet helemaal op voor de 'geneste' lijsten in het tweede voorbeeld tenzij we ons deze voorstellen als kladjes die met punaises op elkaar geprikt zijn.

Welke bewerkingen hebben we nu nodig?

We zouden graag iets aan de lijst willen toevoegen, delen van een lijst verwijderen, een lijst doorzoeken naar een bepaald gegeven, lijsten aan elkaar bevestigen, vergelijken en verplaatsen. Met in ons achterhoofd de behoefte aan zuinigheid blijkt dat dit alles mogelijk wordt door het combineren van een aantal logische vergelijkingen en slechts drie fundamentele bewerkingen: 'construeer een lijst', 'geef het eerste gegeven van een lijst' en 'geef alles behalve het eerste gegeven van de lijst'.

Deze drie primitieve bewerkingen kunnen als volgt worden voorgesteld:

```

'construeren':
Boter (Eieren Melk Koffie) -----> (Boter Eieren Melk Koffie)
'eerste':
(Eieren Melk Koffie) -----> Eieren
'rest':
(Eieren Melk Koffie) -----> (Melk Koffie)

```

Merk op dat 'eerste' een element oplevert terwijl 'construeren' en 'rest' een lijst, als resultaat hebben. In het geval van de boodschappenlijst stelt de lijst een stukje papier voor met daarop potloodtekens, terwijl de elementen zich 'in het hoofd' bevinden, erop wachtend om geschreven te worden, dan wel net gelezen. Maar neem dit voorbeeld niet te letterlijk want in het menu-voorbeeld kon een element zelf ook weer een lijst zijn, bijvoorbeeld de ingrediënten van een kronkel..

Het moge duidelijk zijn dat we, om het derde element van een lijst te verkrijgen, een bepaalde combinatie van 'eerste' en

=====
 verschijnen. Maar laten we in het eerste hoofdstuk kijken naar het concept van List-processing op zich, want zolang men dat niet begrijpt blijft programmeren in Lisp zeer myterieus.

In het dagelijks leven zijn lijsten een heel gewoon middel om een aantal dingen of activiteiten bij te houden; iedereen maakt weleens een lijstje met boodschappen of dingen die hij/zij nog moet doen. De volgende lijst:

Eieren
 Melk
 Koffie
 Cornflakes

heeft weinig uitleg nodig: het is een lijst van voedingsmiddelen die in een winkel gehaald moeten worden (en in dit geval is de volgorde onbelangrijk). Zo'n lijst als deze is een natuurlijke "data-structuur" die objecten representeert die in werkelijkheid bestaan. Het soort handelingen dat we met een dergelijke lijst kunnen doen is: het toevoegen van een iets nieuws (bv. de boter is op..), het wegstrepen van iets om te laten zien dat we het gekocht hebben en het doorzoeken van 'de lijst om er achter te komen of iets er op staat.

We verwachten eigenlijk geen moeilijkheden bij het verrichten van deze handelingen (of het papier moest te kort blijken). Spijtig genoeg kennen computers, en de meeste conventionele computertalen, geen fraaie manier om zulke lijsten te behandelen.

De enige gegevens die een computer begrijpt zijn binaire getallen. Hogere talen als Basic en Pascal geven ons de mogelijkheid om te doen alsof de computer ook andere dingen begrijpt. Door gebruik te maken van de ASCII-codes kunnen we hem de letters van het alfabet "leren": als alles goed is stelt het getal 65 bv. de letter "A" voor. Deze letters kunnen worden aaneengeregen in strings zoals "CORNFLAKE". Door een array-data-structuur te gebruiken kunnen strings of getallen tot grotere eenheden worden opgebouwd, bv.:

```
SHOPLIST$(4)="Cornflakes".
```

Zijn dit nu toereikende gereedschappen voor het omgaan met lijsten? Niet echt. Als we de boodschappenlijst als een string voorstellen:

```
"Eieren Melk Koffie Cornflakes"
```

dan beschouwt de computer dit als een ongestructureerde opeenvolging van letters, omdat de spatie ook een letter is (ASCII code 32). Wanneer we in zo'n string naar een bepaald woord willen zoeken moeten we een stukje programma schrijven dat alleen om de verschillende woorden te scheiden de spaties opzoekt. De functies voor string-handling zoals die in Basic aanwezig zijn zijn weliswaar krachtig maar produceren ook moeilijk leesbare tekst, bv:

```
IF MID$(SHOPLIST$,SPACES(COUNT)+1,SPACES(COUNT+1))=ITEM$  
of in Atom-Basic:
```

```
$A=$(S+SS(C));A?(SS(C+1)-SS(C)-1)=#0D;IF $I=$A THEN ..  
om een woord te isoleren.
```

Een array is iets handiger omdat de gegevens dan tenminste als aparte eenheden worden opgeslagen, maar wordt omslachtig als je onbeperkt willekeurige gegevens wilt verwijderen of toevoegen met behoud van een bepaalde ordening. Beide methoden zijn onbevredigend als we tekst- en numerieke gegevens door elkaar willen gebruiken: de nummers moeten dan als strings wor-

Simulatie programma's zijn werkelijk leuk om mee te spelen en het is natuurlijk wel de bedoeling dat er meer inzicht in een bepaald systeem ontstaat. Vier jaar geleden, toen ik mijn ATOM al een jaar had, waren simulaties voor mij een uitdaging. Het probleem was alleen mijn geringe programmeer ervaring met BASIC. In je enthousiasme ben je dan niet te stuiten en begin je echt 'vies' te programmeren. De GOTO's zijn niet van de lucht. De JOSBOX was nog niet op de markt vandaar dat er in dit programma met CLEARO gewerkt moest worden. Het leuke zou natuurlijk een echt simulatieprogramma zoals TUTSIM zijn voor de ATOM. Ik weet nu in 1987 al twee jaar dat er ook in een gestripte versie voor de ATOM is, maar ik heb het nog steeds niet. Een gebruiksvriendelijke versie ervan is bijna klaar, maar deze ligt al twee jaar in de kast bij iemand. Zelf kan ik weinig in machinetaal en dan maar hopen dat deze versie nog een keer afgemaakt wordt. Misschien is dit dan wel een nieuwe stimulans om het uit de kast te halen en ook het laatste stukje af te maken. Ik zou er heel erg blij mee zijn.

Mijn simulatie is maar heel simpel en is bedoeld voor leerlingen biologie op MAVO niveau.

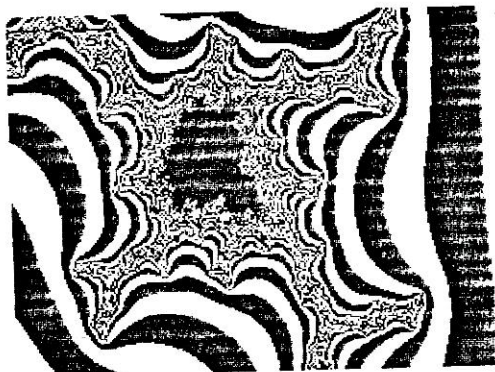
Er zijn twee aparte programma's:

1. spider&fly : vliegen , spinnen en hun onderlinge relatie
2. woodland : zieke, gezonde en immune bomen in een bos.

In beide gevallen wordt na de initiatie eerst het aantal gewenste cycli in beeld gebracht. Daar dit vrij traag werkt kan men door de <SHIFT> toets ingedrukt te houden dit onderdeel stoppen en komt de commandokeuze regel onderin het beeld met de volgende mogelijkheden :

- n - opnieuw
- h - histogram maken
- t - tabel op het scherm
- s - stoppen

De tabel is ingebouwd, omdat een leerling dan zelf het histogram zou kunnen maken en de conclusies eruit trekken. Bij de spinnen en de vliegen kan men ook nog de voortplantings variabelen veranderen. Veel plezier ermee.



```
=====
410      DEY
420      BCS XX3
430:XX6  LDA@#05
440      STA#0321
450      JSR#C589
460      LDA@#08
470      STA#0321
480      LDY#03
490:XX5  LDA(#58),Y
500      CMP@#0D
510      BEQ XX4
520      JSR#CA4C
530      INY
540      BNE XX5
550:XX4  JSR#CD54
560      JSR#CEA1
570:XX3  LDA(#58),Y
580      STA#25
590      INY
600      LDA(#58),Y
610      STA#16
620      INY
630      STY#03
640:XX2  LDA#16
650      CLC
660      SBC#17
670      LDA#25
680      SBC#26
690      BCC XX6
700      JMP XX10
710:XX7  JSR#C231
720      INC#04
730      JSR#C465
740      JMP XX9
750:XX8  LDA#16
760      LDY#25
770      STA#17
780      STY#26
790      BCS XX9
810
820:XX10 LDA@#0C;JSR#FEFB FORM FEED
830      LDA@#03;JSR#FEFB PRINTER UIT
840      JSR#FDOB          SCHEM AAN
850      JMP#C55B
860];N.;P.$6
870IF ?#E8=#23 P."OVERFLOW""
880P."EINDE STATEMENT:","&P'
900E.
```



MACORN VOOR GDOS.
=====

In ATOM NIEUWS 5-3 stond een beschrijving van MACORN, een leuk tekenprogramma, zonder schakelsoft en met de mogelijkheid om tekst in te voeren. Helaas traden er bij gebruik van GDOS problemen op, omdat enkele hulpprogramma's in de disk-catalog ruimte werden opgeslagen. Bovendien werden de hulpprogramma's tijdens de aanroep van schijf geladen, wat de snelheid niet bevorderde.

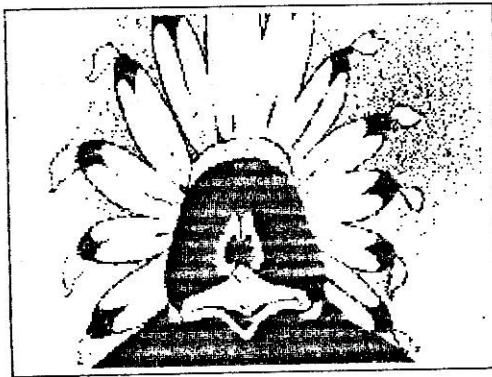
Ik heb het oorspronkelijke programma zodanig aangepast, dat er geen conflicten meer ontstaan door dubbel geheugengebruik. De belangrijkste hulpprogramma's behoeven nu bovendien slechts eenmalig te worden geladen.

Verder heb ik enkele onvolkomenheden opgelost en in het hoofdprogramma wat commentaar bijgevoegd om het de knutselaars wat gemakkelijker te maken de weg te vinden in de spaghetti. De save- en loadroutines, filenaamlengthe, e.d. zijn gericht op gebruik van GDOS, voor de dumproutine wordt m.b.v. een trucje de RX-box gebruikt voor een General Electric printer. Dit is gemakkelijk aan eigen wensen aan te passen.

De oorspronkelijke source programma's heb ik niet opgenomen, deze zijn via de balk al eens verspreid. De aangepaste versie gaat als 2.30 door het leven, op de gok dat het nummer nog uniek is.

Voor de gebruiksaanwijzing en verdere details verwijs ik naar AN 5-3, blz. 62/66.
Veel tekenplezier toegewenst.

Bert van der Veen



Regio Limburg stelt een andere gang van zaken voor rond de verspreiding van hardware-ontwerpen: Publiceer alle ontwerpen in AtomNieuws en peil door middel van inschrijving de interesse voor een bepaald ontwerp. Is er voldoende animo dan kan er een print in productie genomen worden. Eventuele bij regionale ontwerpen behorende print-tekeningen zouden op film over de regio's verspreid kunnen worden zonder uitgebreide verantwoordelijkheden van de hardwarecommissie voor de correctheid van ontwerp of print; op die manier kan men per regio ook tot productie op kleine schaal overgaan en zijn ontwerpen snel beschikbaar.

Regio Den Haag stelt voor om als federatie in een landelijke krant te adverteren om op die manier nieuwe leden te werven. Uit dezelfde regio een voorstel om zich te buigen over de mogelijkheid tot het aansluiten van een compactdisk-speler op de Atom.

Regio Twente uit haar kritiek op het idee van het bestuur om GDOS niet verder te ontwikkelen, ten slotte is alles vrijwel klaar en zitten er mensen op te wachten. De print zal verder afgemaakt worden en de interesse van de leden voor GDOS zal uit een inschrijving met vooruitbetaling moeten blijken.

Er zal een artikel in AtomNieuws verschijnen waarin kosten en mogelijkheden van GDOS en de Z80-kaart tegen elkaar afgewogen worden zodat leden tot een weloverwogen keuze kunnen komen.

De Z-80-kaart: Peter Huisken vertelt hoe het ontwerp in grote lijnen tot stand is gekomen en wat de mogelijkheden met deze kaart zijn. Dan demonstreert hij de werking van de kaart en bijbehorende programmatuur, waaronder het simuleren van GDOS en AtomDos, de mogelijkheid tot het instellen van vele CP/M-formaten en wat er alles nog zo bij hoort. Men is onverdeeld enthousiast.

De penningmeester rekent voor dat aan het eind van dit jaar de kas nog niet helemaal leeg is, ook wanneer ITT haar geld eist, en dat Peter een vergoeding kan ontvangen voor het overdragen van zijn ontwerp + software aan de federatie. Niemand heeft er iets op tegen dat de kaart door de federatie wordt aangekocht. Het is de bedoeling zo snel mogelijk tot levering over te gaan, ook hier (met in het achterhoofd de gang van zaken rond de 80-kolommen-kaart) de verplichting tot inschrijving door vooruitbetaling.

Paul

Ook in de laatste twee vergaderingen van 1986 speelt het beleid van de federatie een grote rol: Zo wordt er in AtomNieuws aangekondigd dat men ter verkrijging van de GDOS-print fl 80,- vooruit zal moeten betalen.. Verder klachten over de kwaliteit van AtomNieuws: bladzijden ontbreken of zijn dubbel aanwezig, tekeningen waarnaar in artikelen verwezen wordt blijken nergens te vinden etc.

Het blijft voorlopig nog geheel onduidelijk wat nu de leverdatum wordt van de printen voor CP/M en GDOS, hoewel de overdracht van bv. het CP/M-project al enige maanden geleden gebeurd is (om over GDOS maar te zwijgen). Rene belooft dat hij het bestuur zal vragen hoe de stand van zaken is.

Gelukkig gebeurt er ook in de laatste maanden van het jaar nog allerlei positiefs:

Gerrit demonstreert zijn nieuwe graphics-toolkit "Provence Deco".

Daarom zijn de aanwezigen druk met het kammen der haren, rechte trekken van stroddassen en het vertonen van vriendelijke dan wel interessante gezichten voor Gerrit's Digitizer die de aanwezigen met bijna fotografische kwaliteit even later uit één printer laat rollen.

Peter Huisken ontwerpt nog meer:

- Een binnenwerk dat op 1 gaatjesprint 80-kolommen, schakelkaart (8x#A000), boot-electronica, 64k Ram, extra VIA, klokomschakeling en een optie op 640x400 graphics combineert. De schema's ervoor zijn intussen in het broodje gepubliceerd.

- Een Ram-disk die gemakkelijk kan worden aangesloten op de Z80-kaart compleeteert deze super-Atom, ook hiervoor heeft Peter een ontwerp dat echter nog gebouwd en getest moet worden.

Ondanks het feit dat de Atom onderhand tot de bejaarde computers gerekend mag worden blijven de regionale gebruikers enthousiast werken aan hard- en software uitbreidingen. Ook het ledental lijkt regionaal gesproken redelijk constant te blijven, er vertrekken mensen maar er komen er ook bij. Landelijk ligt dit anders, daar is duidelijk een afnemende tendens te zien in het aantal leden.

Of dit in de toekomst gevolgen zal hebben moeten we afwachten maar het zou geen verbazing wekken als het aantal donateurs in verhouding toeneemt.

Paul

Het lijkt erop dat vele leden hebben gedacht "een goed begin is het halve werk", want meteen al bij de eerste bijeenkomst van dit jaar dienen zich twee nieuwe hardware projecten zich aan. Bram poot is bezig met het ontwikkelen van een universele epromprogrammer. Het wordt waarschijnlijk een programmer waarmee alle 27XX typen geprogrammeerd kunnen worden. Van verschillende kanten werd hierop gereageerd met tips en ideeën. Het andere project is een universele tweede processor kaart. Hieraan wordt gewerkt door Fernando en Gerrit. Het idee is een kaart waarin in principe het operatingsysteem van elke computer kan draaien, b.v. Apple, BBC, IBM, etc. Over de haalbaarheid hiervan lopen de meningen sterk uiteen. Een maand later werd medegedeeld dat het project gestaakt was omdat het zoals verwacht te grote hardware problemen opleverde.

In een hele lange discussie bleek dat er onder sommige leden ontevredenheid bestaat omtrent het huidige landelijke bestuur en de mentaliteit in de regio's. Als bezwaar werd naar voren gebracht dat men niet genoeg meldt waar men mee bezig is. Met blijkt steeds maar weer dat er, zelfs binnen de regio, langs elkaar heen gewerkt wordt, met als gevolg dat vele leden het zelfde wiel uitvinden. Dat dit probleem er inderdaad ligt is wel duidelijk maar een eenvoudige oplossing is hier dan ook niet voor te vinden. Een 'komputeraar' (M/V) blijkt in de praktijk vaak zeer individualistisch te zijn, en zeer moeilijk over te halen tot bekendmaking van zijn/haar projecten. Zodra je de mensen maar kan overtuigen van het belang van samenwerking of idee-uitwisseling dan is het probleem de wereld uit, maar ja

De onvrede over het landelijke bestuur was voornamelijk dat alles altijd zo onnodig lang duurt. Uit de discussie kwam naar voren dat het inderdaad allemaal wel wat te lang duurt, maar dat beslissingen in het algemeen ook wel wat moeilijker liggen (landelijke structuur).

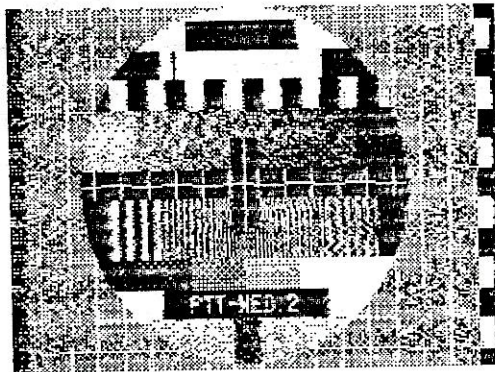
Van J. Jager, de beheerder van het MEPA gebouw is het verzoek gekomen voor een programma voor de aansturing van een XYZ-tafel met boor- en freeskop. Er is al een programma in PASCAL voor (op een apple) maar ze willen laten zien dat zelfs zo'n eenvoudige(?) computer als een Atom een XYZ-tafel kan besturen. Er is rondvraag gedaan of hier interesse voor is. Na herhaald verzoek zijn hier een paar leden voor gevonden.

Peter Huisken heeft zich helemaal gestort op CP/M. Met behulp van een Z80 kaartje blijkt de Atom om te bouwen te zijn tot een CP/M machine. Hij is bereid gevonden tot het schrijven van documentatie over de bouw en het gebruik. Tevens zal hij zorgen voor een basispakket software (monitor, dos, printerroutines). Rien tekent het schema grondig uit en Fernando zal zorgen dat er een print voor het geheel ontworpen wordt. Het is de bedoeling dat het straks in de verkoop gaat voor ca. F 100,- waarvoor men dan de print, een eprom, twee floppies met software en de documentatie (ca. 70 blz) krijgt. De financiering van het project zal waarschijnlijk worden gedeeld met regio zwolle waar dan ook de helft van de oplage heen gaat.

INHOUDSOPGAVE

Biz. Wat staat daar ?!

1. Zoals gebruikelijk, deze keer door Jan
2. DAT leest U nu !!
3. Terug van weggeweest, alleen nu ietsje korter
4. Is onze regio samen te vatten in drie bladzijden ??
7. Jaarverslag, verliesrekening niet gepubliceerd !!
9. Burning rubber blubber Rob
10. Mac Power, nu ook voor GDOS, van Bert
11. LPR: de zoveelst met een rommeldoos thuis
13. No Structured Programming Detected: Program OK !!!!
14. Atom, it's a way of living, ook van Gert Jan
15. Mens durf te LISPen, door Paul
20. Haiku: maak je eigen Avondspits pop-limerick
21. Ontdek de voedzame molecuul structuur in uw ATOM
22. Gert Jan heeft zijn maximale overbrenging gevonden
voor copy: ongeveer 4:1
23. Platgeslagen, maar toch kort s.v.p.
25. Jan houdt bij hoog en laag vol dat het kan !!
27. Stuur je eigen tanga-loze versie in ?!?!



ALGEMENE INFORMATIE:

Bestuur:
 Voozitter : R.Spel
 Secretaris : P.JansenKlomp
 Penningmeester: G.J.Noorland
 W.Buning

Redactie:
 J.Biel
 G.Hillebrand

Epromprogr.dienst: R.van de Schepop
 Programma-archief: P.Huisken Printdienst : G.J.Noorland
 Drukwerkarchief : R.Spel Ledenadministratie: G.J.Noorland

