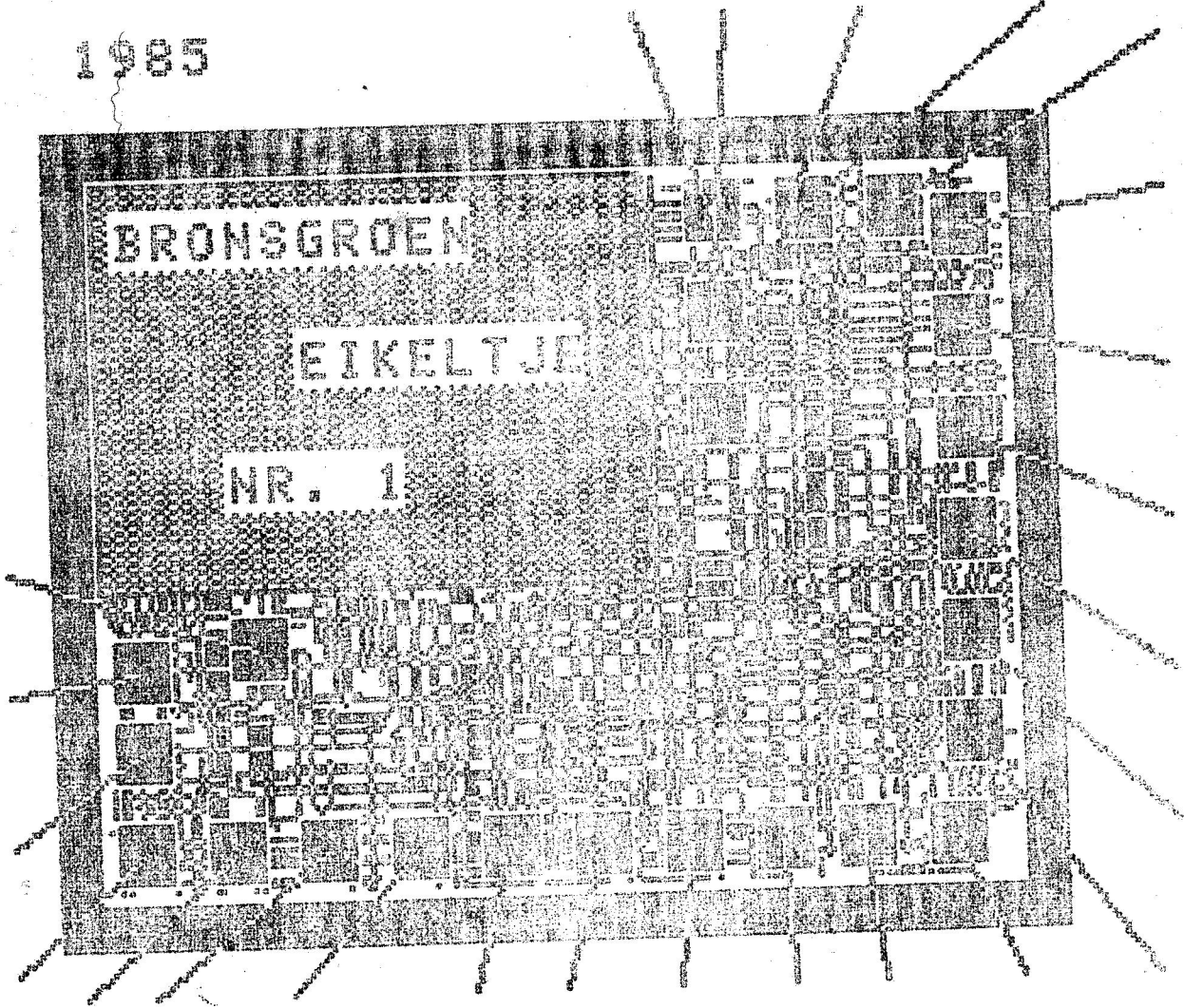


SJAAR VOCHTEN
19430

STAPPEN MOTOR
OLIVETTI DRIVE

1085



ACORN COMPUTERCLUB

LINGURD

M.D.C.R.

A.MARCHAL

Korte beschrijving nieuwe software voor in EPROM/RAM op A-gebied.

Deze software is upwards compatible met de vorige MDCR software. Dus oude file's kunnen weer teruggelezen worden etc.

In het hieronder staande overzicht is met een % aangegeven welke commando's niet gebruikt hoeven te worden als u volledig automatisch wilt werken, dus met een Header.

Alleen als u zonder Header wilt werken moet u gebruik blijven maken van sash en salo, in het andere geval is om te saven alleen SAM nodig.

In de hieronder getoonde voorbeelden kunt u zien dat de commando's in een basic programma opgenomen kunnen worden ook SAM commando's.

Voor de mensen die zelf basic programma's schrijven zijn hieronder een drietal mogelijkheden opgenomen.

1. Laden van een gewenste file kan ook als volgt:
%%140="FILENAAM";LI.#A6C9

2. Saven van een file met de naam "FILENAAM" kan ook als volgt:
%%140="FILENAAM"
!#70=Beginadres
!#72=Eindadres
!#74=Startadres (zie ook bij salo).
LI.#A6D7

3. Wilt u van een bepaalde filenaam het nummer weten waarmee het op de band staat dan als volgt:
%%140="FILENAAM";LINK#A545;P.??#17F

Voor de mensen die zelf eenvoudige interface-schakelingen kunnen bouwen is ook nog het schema opgenomen.

U heeft dus geen extra VIA nodig. Bij mij loopt dit nu bijna 2 Jaar met op de A-poort een printer aangesloten.

HELP

====

Geeft een overzicht van alle commando's.

- HELP ini
sash % DEL
salo % LOCK
VERM UNL
LAAD CAT
REW MHAND
wis MAUTO
LRUN UPD
SYN FWD
MCAT BVER
SAM BACKUP

Bij nogmaals op een toets drukken, worden een aantal voorbeelden getoond.

```

ini "BANDNAAM"
LAAD4; LAAD "FILES"
FOR X=2TO5; LOCK X; N.
UNL "FILE3"; UNL6      UNL=UNLOCK
DEL9; REV; WIS; FWD    DEL=DELETE
LRUN(S+Q)      LAAD EN RUN
SAM"FILE" 2900 3000 29; P."BASIC"
SAM"PAINTER" 2600 3000 2613
MHAND; F. X=3TO12; DELX; N.; NAUTO
UPD"FILE"; UPD4      =UPDATEN
BVER = BAND VERMISSELD
CAT
BACKUP = NAAR TAPE

```

AMMUS

```

sash
====

```

Save met na het programma 40 milliseconden wissen.

```

salo
====

```

Save met na het programma 1 seconde wissen.

Alle programma's of datafile's die U op band zet krijgen een nummer en komen met een tussenruimte, die gewist moet zijn, op de band.

Het eerste programma begint pas na 1 seconde (aanloopstuk).

Dit wordt automatisch verzorgd.

Doordat een programma altijd gevuld moet worden door een stuk van 0,45 seconde gewiste band wordt normaal gesaved met salo.

Hierop is natuurlijk een uitzondering nl. als u een bestaand programma of datafile wilt overschrijven, dan wordt sash gebruikt.

Buidelijk is wel dat er dan natuurlijk een file op band gezet moet worden die even groot is als de vorige file.

Een gesaved programma wordt automatisch gecontroleerd door het terug te lezen en de checksum te bepalen (verify).

```

salo "PROGNAAM" BBBB EEEE SSSS
                OF SS

```

BBBB = Beginadres

EEEE = Eindadres

SSSS = Assembler startadres

SS = Basic start adres met textpointer op SS

TOP en DIM wordt automatisch goed gezet.

```

VERM
====

```

Verify het programma met het aangegeven nummer X of "programmenaam".

Haal het programma van band maar zet het niet in het geheugen, bepaal wel de checksum en controleer die met de op de band staande checksum.

LAAD

====

Laad programmanummer X of "programmanaam" naar het geheugen zoals door u op band gezet.

REW

===

Rewind de tape, deze is dan tevens gesynchroniseerd.

WIS

===

Wis de tape vanaf deze positie tot aan het einde van de band.

LRUN

====

Laad programmanummer X of "prog.naam" en start het hierna op. Dit kan dus zowel in basic als in assembler zijn. Zie hiertoe bij salo en sash.

SYN

===

Synchroniseer het bandje.

Dat wil zeggen positioneer juist en vul het juiste programmanummer in.

Dit is het nummer waar de MDCR voor staat.

MCAT

====

Laat een catalogus zien van wat er op band staat.

De band moet natuurlijk gesynchroniseerd zijn.

Dit commando is te onderbreken met de CTRL-toets.

Wel even vasthouden. Het bandje blijft dan gesynchroniseerd achter.

SAM

===

SAVE AUTOMATISCH MDCR.

SAM"FILE" 2900 3000 29 (voor basic).

SAM"PAINTER" 2800 3000 2813 (voor assembler).

Het programma wordt gesaved en de Header wordt automatisch bijgewerkt.

U hoeft zich geen zorg te maken over kort of lang saven. Dit wordt automatisch verzorgd.

INI

===

INI"BAND 1A", hierna wordt gevraagd "Zeker weten j/n"

Bij "ja" wordt het bandje van voor tot achter gewist, en vooraan wordt hierna de Header aangemaakt van #2400 tot #2800

DEL

===

Delete X of "prog.naam".

In de Header verschijnt dan een ruit-tekentje tussen het programmanummer en de programmanaam. Let wel het programma staat nng steeds op band dit wordt pas overschreven bij een volgende save, die exact dezelfde programma lengte heeft.

LOCK

====

Lock X of "prog.naam"

Het programma wordt gelockt (beschermd).

Gelockte file kunnen niet geupdate of gedelete worden.

Tussen programmanummer en naam verschijnt het "*" teken.

UNL

===

UnlockX of "prog.naam". Het # teken verdwijnt en het programma kan worden geupdate of delete.

CAT

===

U krijgt een overzicht van de op de band staande programma's, tevens het totaal aantal gebruikte bytes, het aantal keren dat reeds gesaved is en het nummer waar de MDCR voor staat.

Mocht de Header in het geheugen "beschadigd" zijn dan wordt deze opnieuw van band geladen.

MHAND

=====

Met dit commando wordt het MDCR-systeem op handbediening gezet.

Na iedere transactie wordt de Header bijgewerkt, maar nog niet op de band geschreven. Makkelijk als meerdere programma's

gesaved of bijv. gedelete moeten worden, na de laatste transactie en voordat u de computer uitzet voert u in MAUTO en de Header wordt automatisch op de band bijgewerkt.

MAUTO

=====

Zet het MDCR-systeem op automatisch en saved de Header zoals die nu is.

UPD

===

Commando om een bestaand programma of file te updaten.

Let wel het programma of file kan niet groter zijn als het origineel programma.

Dus houdt er rekening mee als het een program is waaraan nog geknutseld moet worden, en u wilt het over de vorige versie heenleggen save dan met meer ruimte. Overschrijven van het volgende programma is niet mogelijk. (Is tegen beveiligd in de soft).

FWD

===

Laat de tape vooruit lopen. Is te stoppen met ESC-toets. De synchronisatie wordt automatisch verstoord.

BVER

=====

Bij iedere bandverwisseling te gebruiken, als u automatisch wilt werken.

De Header wordt geladen en er verschijnt een overzicht zoals genoemd onder CAT.

BACKUP

=====

Backup van MDCR naar normale cassette recorder.
Kan zowel met 300 als 1200 Baud. Bij het comando BACKUP wordt
het eerste op de band staande programma geladen en er verschijnt
RECORD TAPE

Zet de bandrecorder op opnemen en druk op een toets.
Het programma wordt nu op cassette gesaved.
Daarna laadt de MDCR het tweede en volgende programma's en deze
worden weer gesaved. (Volledig automatisch).

FOUTMELDINGEN

=====

- 101 Geen cassette aanwezig.
- 102 Begin of einde van de band
- 103 Niets op de band.
- 104 Preamble niet ontvangen.
- 105 Niet correct geladen
- 106 Schrijven niet toegestaan.
- 107 Bij controle van het programmanummer op de band met het
door de software aan de hand van de Header berekende
programmanummer is een verschil geconstateerd. Dit kan bijv.
optreden als u gaat saveen terwijl u een ander bandje in de MDCR
heeft gedaan zonder EVER te doen.

De werking van de MDCR is te onderbreken door de escape-toets
vast te houden, of door de break-toets in te drukken. In het
laatste geval wordt de zaak altijd direct afgebroken.
Voor de liefhebbers er is nog wat ruimte over, als u de soft in
EPROM zet.

De software komt in het bandjes archief.

VEEL SUCCES

RTTY-ONTVANGST

=====

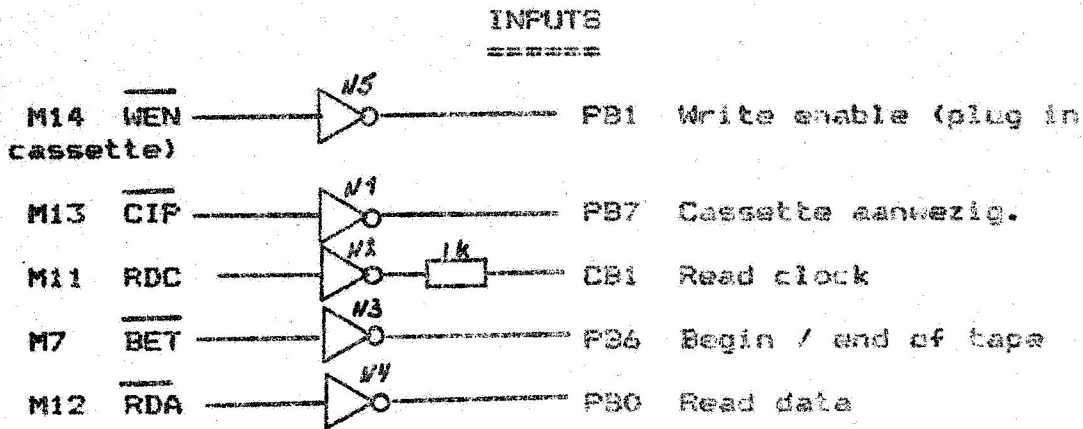
N. SCHREURS

=====

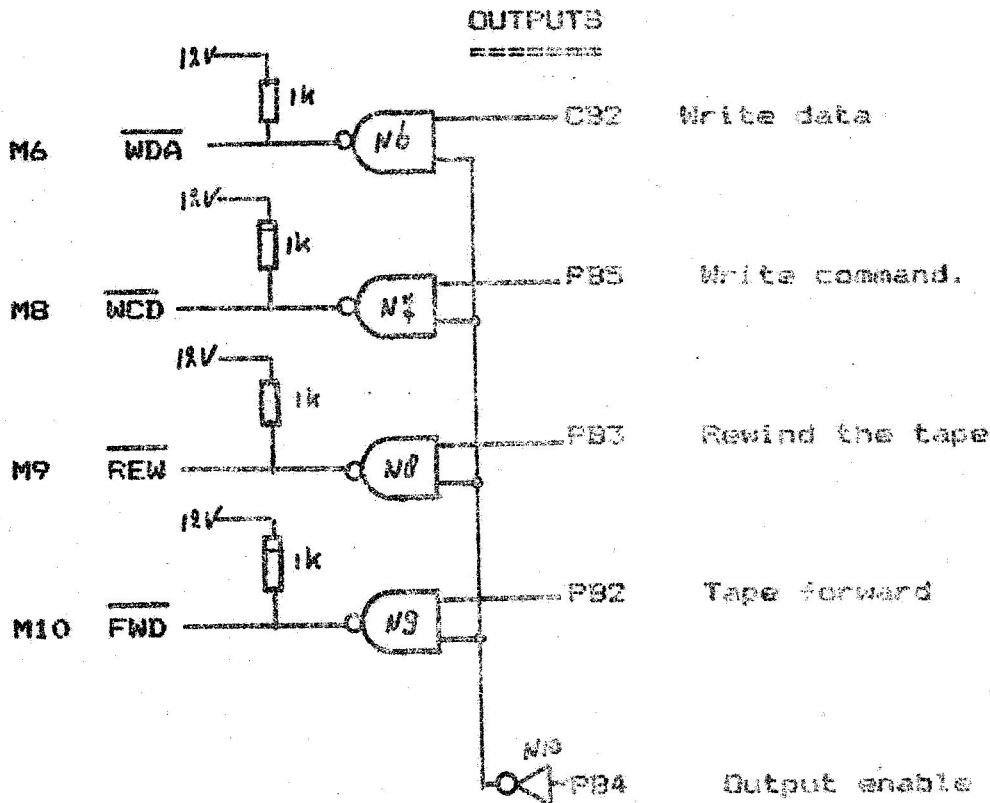
Voor de ontvangst van RTTY is een klein ontvangertje in de
handel, waarmee redelijke resultaten te behalen zijn.
(Compleet bouwpakket met alle onderdelen en bouwbeschrijving Fa.
Biermans, Oude Kerkstraat 7 te Berg en Terblijt kosten ongeveer
F58,-).
Als decoder is goed bruikbaar de Morse Interface (Acorn Nieuws
Jrg 2 nr. 1).
Software, in verschillende vormen circuleert in de club.
(komt t.z.t. in het bandjes archief).
Mocht u moeilijkheden ondervinden bij de bouw of met de
software, er zijn diverse mensen in de club die u verder kunnen
helpen.
(Even informeren).

MDCR INTERFACE

Interface schakeling die op de B-poort van de ACORN ATOM moet worden aangesloten.



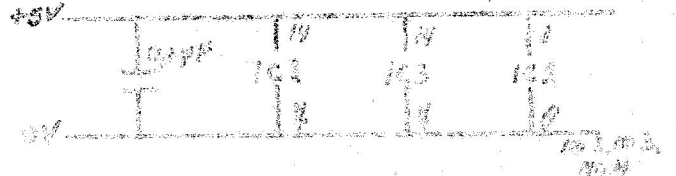
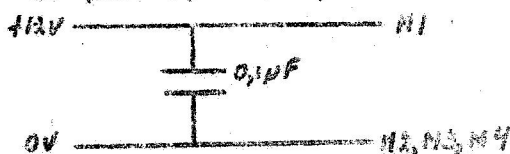
Al deze input signalen worden geïnverteerd en aangepast door IC1 (CD4049 N1 - N5).



IC1	N1-N5	CD4049
IC2	N6-N9	74LS26
IC3	N10	74LS04

M1 tot en met M14 zijn de MDCR interfacednummers

+5V pen 1; 0V pen 32 van de ACORNBUS.



=====
BEGINNERSHOEKJE E. SANDERS
=====

1.3.6. CONTROL - CTRL

Er zijn verschillende speciale functies beschikbaar via het toetsenbord welke worden verkregen door bepaalde toetsen samen met de "control" toets, gemerkt met "CTRL", in te drukken. Voorlopig worden hier slechts de volgende twee control functie's genoemd:

CTRL - G (dus de CTRL toets samen met de G toets indrukken) geeft een pieptoon in ATOM's luidspreker)

CTRL - L maakt het scherm schoon en zet de cursor links boven op het scherm. (precies wat PRINT \$12 ook doet)

LET OP: om na een bv. CTRL - L verder te kunnen gaan moet je eerst op ESC drukken zodat de prompt weer verschijnt.

Doe je dat niet dan krijg je, als je dan bv. LIST intypt, een ERROR 94, dus een foutmelding.

De "control" functie kunnen we ook in een programma verwerken. Hoe, dat zullen we later nog zien.

1.3.7. BREAK

De BREAK toets zal de computer resetten (als het ware op 0 stellen) en hem terugbrengen in de toestand zoals die was nadat hij net was aangezet. Het moet normaal niet nodig zijn om de BREAK toets te gebruiken, maar sommige programma's kunnen in een zgn "loop" geraken, die op geen enkele andere manier gestopt kan worden dan met de BREAK toets.

De inhoud van het geheugen wordt echter bewaard als de BREAK toets is ingedrukt geworden. Ieder opgeslagen programma kan weer worden teruggekregen.

Als je een BREAK hebt gedaan, typ dan OLD, geef een RETURN en typ dan LIST en zie, daar is je programma (hopelijk) weer. (LIST typen na OLD is echter niet noodzakelijk, je kunt ook typen bv. OLD, RETURN en dan RUN).

1.4. SCROLLING

Als de cursor de onderkant van het scherm bereikt heeft en je typt meer regels dan er op het scherm kunnen (er kunnen max. 16 regels op) dan gaat het scherm scrollen, dwz iedere regel schuift steeds een naar boven, zodat je altijd de laatste 16 regels ziet van wat je hebt ingetypt.

De bovenste regel van de tekst op het scherm zal dus steeds verdwijnen.

1.5. STORING TEKST (bewaren of opslaan van gegevens, programma's of tekst in het geheugen)

Iedere regel die achter de prompt ">" wordt ingetypt en begint met een regel- of lijnnummer wordt na een RETURN niet meteen uitgevoerd, maar opgeslagen als tekst in het geheugen. Alle soorten input, data enz kunnen op deze manier worden opgeslagen. Het zou bv de tekst van een boek, "n basic programma, "n

assembler programma of data (gegevens voor een programma) kunnen zijn.

Nu zullen we zien hoe je een stuk tekst of een programma moet intoetsen. Deze kan dan bv bewaard worden op een cassette, of naar een printer worden gestuurd. Ook kunnen we de ingebrachte data cq tekst "editten" dwz bewerken, verbeteren, toevoegen ed. Het navolgende is dan ook van toepassing als een nieuw programma moet worden ingetoetst.

TIP: Men je aan om, voordat je een nieuw programma intypt, altijd eerst NEW intypt, waarna een RETURN.

De regels van een programma moeten altijd beginnen met een regelnummer. Dit mag ieder getal zijn tussen 0 en 32767. Het hoogste regelnummer mag dan ook niet groter zijn dan 32767. Je hoeft de regelnummers ook niet achter elkaar te laten volgen, dus bv. 1,2,3,4.

Dit is zelfs af te raden, je kunt beter de regelnummers per bv 10 laten verspringen, dus bv. 10,20,30,40. Waarom.....?

Dat zul je nog gauw genoeg ontdekken.

Achter het regelnummer typ je dan je programmaregel of je tekst.

Bv. typ het volgende:

```
10 VEEL COMPUTERS HEBBEN EEN VIDEO-UITGANG
20 DEZE AANSLUITING GEEFT EEN VEEL SCHERPER BEELD
30 ALLE MONITORS HEBBEN EEN VIDEO- OF MONITOR AANSLUITING
40 DE VIDEOAANSL. VAN UW TV IS HIERVOOR NIET GESCHIKT
```

Vergeet niet achter iedere regel een RETURN te geven. Met iedere regel bedoel ik een programmaregel, dus geen RETURN geven als de regel op het scherm vol is maar pas als de programmaregel af is. Als de beeldschermregel vol is gewoon doortypen, het komt vanzelf op de volgende regel.

Iedere programmaregel kan 64 karakters (incl. spaties, regelnummers, leestekens ed.) bevatten. Als je probeert meer dan 64 karakters in te typen, dan weigert ATOM verder te gaan. Je kunt dan ook geen RETURN geven om naar de volgende regel te gaan.

Je zult nu eerst het laatste karakter met de DELETE toets moeten verwijderen, pas daarna kun je weer een RETURN geven en naar de volgende regel gaan. Je mag natuurlijk ook meerdere karakters DELETEN (verwijderen)

De reden waarom we de regelnummers beter met bv 10 kunnen laten verspringen is dat we nu op een simpele manier regels kunnen tussenvoegen. Bv. om tussen regel 30 en 40 (zie vorig programmaatje) een regel tussen te voegen typ je gewoon achter de laatste regel (regel 40 in ons geval):

```
36 OOK ONZE COMPUTER HEEFT EEN VIDEO-AANSLUITING
```

Typ nu LIST en zie, ATOM heeft regel 36 reeds ertussen gesorteerd en zelfs op de juiste plaats. Je zou een programma dus desnoods geheel door elkaar kunnen gooien (tenminste wat regelnummers betreft), ATOM zet de boel dan wel weer netjes op volgorde.

1.6. COMMANDS (opdrachten oftewel instructies)

Opdrachten die achter de prompt worden ingetypt zonder dat ze voorafgegaan worden door een regelnummer en gevolgt door een

RETURN worden onmiddellijk door de computer uitgevoerd zonder dat deze eerst in het geheugen wordt opgeslagen. Vandaar dat je deze dingen niet meer kunt opvragen zoals bv wel kan met een programma.

Wil je ze wel opslaan in het geheugen dan MOET je regelnummers gebruiken. Typ nu de opdracht (de 5 regels uit het vorige voorbeeld moeten nu wel in het geheugen zitten):

```
LIST (=uitlijsten of afdrukken)
```

Deze opdracht zal ervoor zorgen dat de opgeslagen tekst op het scherm geprint gaat worden.

Er zijn verschillende mogelijkheden met de LIST opdracht bv.:

- LIST het hele progr. wordt tot het eind gelist
- LIST 10 list alleen regel 10
- LIST 20,40 list de regels 20 tm 40
- LIST 20, list de regels vanaf regel 20 tot het einde
- LIST ,30 list de regels vanaf het begin t/m regel 30

Soms wil men de listing rustig bekijken omdat, er bv een fout ergens zit, maar als de listing bv 100 regels lang is, zou je steeds bv LIST 20,30 enz. moeten doen, om toch alles gedurende een langere tijd op het scherm te houden.

Nu, het kan gemakkelijker: typ: PRINT \$14, RETURN, en typ nu LIST. Er verschijnen nu maar 16 regels op het scherm (de eerste keer kunnen er meer verschijnen zodat je de eerste niet meer ziet).

Als er meer zouden verschijnen druk op ESC en typ weer LIST, nu gaat het wel goed.

Als nu de eerste 16 regels op het scherm staan en je wilt verder gaan hoef je alleen maar op de spacebar (spatiebalk) te drukken voor de volgende 16 regels.

En dat gaat zo door tot het einde.

Vergeet echter niet om als je klaar bent het volgende in te typen: PRINT \$15, anders blijft de computer steeds 16 regels schrijven en stopt daarna steeds.

Je kunt ipv PRINT \$15 ook de BREAK toets indrukken en dan OLD typen. (Dit lukt niet altijd)

Als je zou vergeten PRINT \$15 in te typen is dat meestal niet zo erg want de meeste programma's lopen toch wel gewoon door.

1.7. EDITING (verbeteren, tussenvoegen, weghalen enz.)

Een opvallend iets van de ATOM is dat opgeslagen tekst en/of programmaregels gemakkelijk kunnen worden gewijzigd door hetzelfde regelnummer (van de regel waar de fout inzit) te typen gevolgt door de nieuwe of verbeterde regel (met het zelfde regelnummer). Om in het voorgaande voorbeeld bv regel 20 te veranderen typ:

```
20 DEZE REGEL IS VERANDERD
```

List het programma is zie het effect.

In de volgende aflevering gaan we verder in op het feit dat ATOM ook karakters in een bestaande regel kan toevoegen of zelfs weghalen zonder dat er een spatie ontstaat.

=====

NIEUWE WCH-ROUTINE TON GERAETS

=====

```

10 REM NIEUWE WCHR
20 REM AUTEUR: A.G.GERAETS
30 REM      NEER
40 REM DATUM : 840411
50 REM PGM GEEFT FAST OUTPUT MET SINT MOGELIJKHEDEN
60 REM DE SINT-ROUTINE IS ONTLEEND AAN EEN GELIJKNAMIG
70 REM PGM UIT AN 1983
80 DIM SS27;F,I=0T027;SSI=#FFFF;N.
90 IN."WAAR MOET ROUTINE KOMEN TE STAAN"Z
100 F,I=1T02;P=Z;I
110:SS0;LDA@ SS1Z256;STAN#208;LDA@ SS1/256;STAN#209;JMP#C55B
120:SS1;PHA;STX#E4;STY#E3;CMP@#0A;BNE SS2;JSR SS14
130:SS2;PLA;JSR #FEFB;PHP;PHA;CLD;JSR SS3;JMP #FESF
140:SS3;CMP@#06;BEQ SS4;CMP@#15;BEQ SS5;LDY #E0;BMI SS6
150:SS4;CMP@#1B;BEQ SS4;CMP @#07;BEQ SS7;JSR SS13;LDX @#0A
160:SS5;JMP #FEC5;BNE SS10;JMP #FEB7
170:SS6;CLC;LDX @#00;STX #B000
180:SS7;LDX @#02;PHP;ASL #DE,X;PLP;ROR #DE,X
190:SS8;RTS
200:SS9;LDA @#05;TAY
210:SS9;STA #B003
220:SS9;DEX;BNE SS9;EOR @#01;INY;BPL SS8;RTS
230:SS10;CMP @#20;BCC SS13;ADC @#1F;BMI SS11;EOR @#60
240:SS11;STA(#DE),Y;INY;CPY @#20;BCC SS12;JSR SS19;LDY @#00
250:SS12;STY #E0
260:SS13;PHA;LDA (#DE),Y;EOR #E1;STA (#DE),Y;PLA;RTS
270:SS14;JSR #FE71;BCS SS18;CPY @#00;BEQ SS15
280:SS15;JMP SS18
290:SS15;LDX @#20;JSR #FB83
300:SS16;JSR #FE71;BCS SS16;CPY @#00;BEQ SS17
310:SS17;JMP SS18
320:SS17;LDX @#10;JSR #FB83
330:SS18;RTS
340:SS19;LDA #DE;LDY#DF;CPY @#01;BCC SS26;CMP @#E0;BCC SS26
350:SS20;LDY #E6;BMI SS22;DEY;BNE SS21
360:SS20;JSR #FE71;BCS SS20;LDY @#10
370:SS21;STY #E6
380:SS22;LDY @#20
390:SS23;LDA #B000,Y;STA #7FE0,Y;INY;BNE SS23
400:SS24;LDA #B100,Y;STA #B0E0,Y;INY;BNE SS24;LDY @#1F;LDA
@#20
410:SS25;STA (#DE),Y;DEY;BPL SS25;RTS
420:SS26;ADC @#20;STA #DE;BNE SS27;INC #DF
430:SS27;RTS
440J
450 P.#6;N.
460 @=0;P."ROUTINE AANROEPEN OP #"&SS0";@=8
470 END

```

=====
DOCUMENTATIE ATOM-MERGE H. LINNSEN
=====

1. Algemeen

- Met dit programma is het mogelijk om 2 programma's samen te voegen (mergen).
- De mogelijkheid bestaat ook voor 2 bestanden mits het regelnummer als recordkey gebruikt wordt.
- Als enigste toolbox wordt de 'P-CHARME' gebruikt.

2. Programmawerking

- Het programma vraagt om de adressen van de samen te voegen programma's of bestanden. Optioneel is het ingeven van een bepaalde serie regelnummers die met het tweede programma (of een serie regelnummers daarvan) samengevoegd dienen te worden. Men kan ook volstaan met het ingeven van 'CR'.
- Als in de samen te voegen programma's dezelfde regelnummers voorkomen wordt een error 400 gegenereerd en is het programma alleen nog met 'ESC' af te breken.
- Het statement 'END', al of niet met spaties en/of afgekort, moet zich als enigste statement op de laatste regel van het programma bevinden.
- Tevens dient men er zelf op te letten dat voor alle programma's (atom-merge, A en B) of programma en bestanden voldoende geheugenruimte beschikbaar is.

3. Programmavariabelen

- Voor 'programma A' : A = programma adres
B = adrespointer regelnummer
C = regelnummer
I = einde programma indicator
SB = statement
SF = regelnummer va
SG = regelnummer tot en met
- Voor 'programma B' : V = programma adres
W = adrespointer regelnummer
X = regelnummer
J = einde programma indicator
BY = statement
BS = regelnummer van
BT = regelnummer tot en met
- Voor 'nieuwe programma' : K = programma adres
L = adrespointer regelnummer
SN = statement
- Algemeen : SZ = inputvariabele

4. Listing

```
10 PROGRAM ATOM-MERGE
20 REM AUTEUR
30 REM H.LINNSEN
40 REM ROERMOND
50 REM 04750-27879
60 REM VERSIE 1.0 010484
70 REM TOOLBOX:P-CHARME
80
```

```

90 PROC FOOTNOTE
100 IF I<1 C=?B*256+B?1
110 IF J<1 X=?M*256+W?1
120 VTAB15
130 P.C" "X
140 PEND
150
160 PROC COPY-D
170 FOOTNOTE
180 D=B+2
190 IF C=VAL6 I=1
200 IF #D=" E." OR #D=" END" OR #D="E." OR #D="END"
I=1
210 IF I=1 IF J=0 GOTO 280
220 ?L=?B
230 L?1=B?1
240 N=L+2
250 #N=#D
260 L=L+LEN(N)+3
270 S=B+LEN(D)+3
280 PEND
290
300 PROC COPY-Y
310 FOOTNOTE
320 Y=W+2
330 IF Y=VALT J=1
340 IF #Y=" E." OR #Y=" END" OR #Y="E." OR #Y="END"
J=1
350 IF J=1 IF I=0 GOTO 420
360 ?L=?M
370 L?1=W?1
380 N=L+2
390 #N=#Y
400 L=L+LEN(N)+3
410 M=W+LEN(Y)+3
420 PEND
430
440 PROC ERROR
450 VTAB 14
460 P.#7
470 P."ERROR 400 "
480 P.C" "X" EQUAL LINE NUMBERS"
490 PEND
500
510 REM Init
520 DIM Z1,F4,S4,S4,T4
530 P.#12;P." ATOM-MERGE"
540 P."=====
550 I=0
560 B=0
570 J=0
580 IN."ADDRESS PROGRAM A: "A
590 IN."FROM LINE-NUMBER : "SF;IF SF="" SF="0"
600 IN." TO LINE-NUMBER : "SG;IF SG="" SG="99999"
610 IN."ADDRESS PROGRAM B: "V
620 IN."FROM LINE-NUMBER : "SS;IF SS="" SS="0"
630 IN." TO LINE-NUMBER : "ST;IF ST="" ST="99999"
640 IN."DESTINATION ADDRESS: "X

```

```

660 IN." "          ORAY 1/7/CR" #2
670 IF #Z="N" GOTO 630
680 P.#12;?#E1=0
690 P."          ATOM-MERGE"
700 P."=====
710 P." " "MERGE OF PROGRAM AT #"&A'
720 P."          AND PROGRAM AT #"&V'
730 P." TO NEW PROGRAM AT #"&K'
740
750 REM MERGE
760 B=A+1
770 W=V+1
780 ?K=#00
790 L=K+1
800 VTAB 14;F."BEGIN MERGE"
810aC=?B*256+B?1
820 IF C<VALF D=B+2;B=D+LEND+3;6.a
830 X=?W*256+W?1
840 IF X<VALS Y=W+2;W=W+LENY+3;6.830
850 IF C=X ERROR
860 IF C<X;COPY-D
870 IF X<C;COPY-Y
880 IF I=1;DO COPY-Y;UNTIL J=1;6.910
890 IF J=1;DO COPY-D;UNTIL I=1;6.910
900 GOTO a
910 ?L=#FF
920 ?#E1=128
930 P.#7"   READY"
999 END

```

-0-0-0-0-0-0-0-0-0-0-

=====
TOP- EN DIMPONITER AUTOMATISCH GOEDZETTEN YON GERAETS
=====

```

10 REM TOP TG /V1\
20 REM TOP EN DIM-POINTER GOEDZETTEN IN EEN PROGRAMMA
30 REM AUTEUR A.G.GERAETS
40 REM          NEER
50 REM TEL. (04759) 2012
60 REM DATUM : 830731
70 REM GEBRUIK:
80 REM - IN BEGIN PROGRAMMA LINKEN (LLO)
90 REM - ALS STATEMENT (TOP), DAN LAATSTE RTS
100 REM   VERANDEREN IN 'JMP#C55B'
110 REM HANDIG BIJ AUTOSTART PROGRAMMA'S
120 REM P (REGEL 150) VERANDEREN IN GEMENSTE ADRES
130 DIMLL5
140 F.X=1TO2;P.#21
150 P=#2800
160I:LLO LDA#12;STAKE;LDY#0;STY#0;LEV
170:LL1 INY;LDA(#D),Y;CMP#0;BNELL1
180 JSR#C5BC;LDA(#D),Y;BNILL2;INY;BNELL1
190:LL2 INY;JSR#C5BC;LDA#D;STAC23;LDASE;STAX24;RTS
200J
210 M.;F.#6
220 END

```

VERKLARENDE WOORDENLIJST MICROCOMPUTERS "E-F" N. SCHREURS

EBCDIC - Extended Binary Coded Decimal Interchange Code. 8-bit code die bijv. door IBM wordt gebruikt voor het coderen van alfanumerieke symbolen.

ECHO - Een karakter dat op het toetsenbord wordt ingetypt wordt nadat het door de computer is ontvangen, naar een printer en/of een beeldscherm gezonden.

ECL - Emitter Coupled Logic, emitter gekoppelde logica.

EDITOR - Programma dat de invoer van tekst in de computer bestuurt. Een editor biedt bovendien de mogelijkheid om een karakter of een regel te wissen of toe te voegen, een bepaalde tekst (string) op te zoeken, enz.

EMULATION - Het simuleren van de ene computer door de andere computer in "werkelijke tijd" (real time). Computer A voert een simulatieprogramma uit waardoor hij zich precies zo gedraagt als computer B, d.w.z. hij simuleert de instructieset.

EOC - End Of Character, controlecode die het einde van het transport van een karakter aangeeft.

EOF - End Of File, controlecode die het einde van een file (bestand) aangeeft.

EOB - Exclusive OR, uitsluitend OR. Logische schakeling waarvan de uitgang alleen dan "hoog" wordt, wanneer een van de ingangen "hoog" is. Ook wel aangeduid als EXOR.

EOT - End Of Transmission, controlecode die het einde van het data-transport aangeeft.

EPROM - Erasable Programmable Read Only Memory. Geheugen waarvan de inhoud kan worden gewist door de chip voor een bepaalde tijd bloot te stellen aan (hard) ultra-violet licht. Het geheugen kan daarna weer worden geprogrammeerd (van nieuwe informatie worden voorzien) m.b.v. een speciaal PROM-programmeerapparaat.

EVEN PARITY - Pariteitsbit die aan een binair woord wordt toegevoegd, zodanig dat het aantal eenen even wordt.

EXCESS-3 CODE - Variant op de BCD-code, waarbij de coden 3 t/m 12 worden gebruikt.

EXECUTION TIME - Uitvoeringstijd, de tijd die nodig is om een instructie uit het geheugen te halen, te decoderen en uit te voeren.

FAMOS - Floating gate Avalanche MOS, technologie die wordt gebruikt bij de fabricage van EPROM's.

FAN IN - Elektrische belasting die een uitgang vormt voor een ingang.

FAN OUT - Elektrische belasting die een uitgang maximaal aan kan. Wordt meestal uitgedrukt in het aantal ingangen dat op die uitgang kan worden aangesloten.

FDC - Floppy Disk Controller, besturingseenheid voor floppy disks.

FETCH - De eerste cyclus in de verwerking van een instructie, waarin de instructie uit het geheugen wordt gehaald en in het instructieregister geplaatst.

FIFO - First In, First Out. Buffergeheugen waarin de data aan de ene zijde wordt ingevoerd en er aan de andere zijde weer uit komt. De data die het eerst zijn ingevoerd verschijnen dus ook als eerste weer op de uitgang.

FILE - Bestand, bij elkaar behorende hoeveelheid informatie die door de gebruiker als een eenheid, en onder een bepaalde naam, kan worden aangesproken.

FIRMWARE - Een programma dat in ROM is opgeslagen, d.w.z. software in hardware.

FIXED-HEAD DISK - Schijfgeheugen waarbij boven elk spoor op de schijf een lees/schrijfkop is aangebracht. Omdat de positionering van de lees/schrijfkop niet meer nodig is, zijn deze geheugens zeer snel.

FIXED POINT - Getalweergave waarbij de decimale punt (komma) zich op een vaste plaats bevindt.

Bijv.:

23.456
123.097
3.67
enz.

FLAG - Informatiebit, meestal opgeslagen in een afzonderlijke flipflop, die een speciale conditie aangeeft, bijv. resultaat is nul, resultaat is negatief, overdracht, enz.

FLOATING POINT - Getalweergave waarbij de lengte van de getallen altijd dezelfde is. Een floating point getal bestaat uit een mantisse en een exponent. Het getal 23,5731 bijvoorbeeld, wordt weergegeven als .235731 E02 d.w.z., 0,235731 maal 10 tot de tweede macht. De mantisse is .235731; de exponent is E02 (E staat voor 10 tot de macht...).

FLOPPY DISK - Massageheugen-eenheid met als informatiedrager een flexibele (floppy) schijf, de diskette. De doorende van de schijf is 8 inch, 5,25 inch of 3,5 inch.

FLOW CHART - Stroomdiagram of stroomschema. Schematische voorstelling van de oplossingsmethode van een probleem. Een stroomdiagram verduidelijkt bijvoorbeeld het programma door het op te splitsen in een aantal logische blokken.

FORMATTER - Circuit of program, dat op een diskette of cassetteband het begin van een file, de tracks (sporen) de adressen en de controle-karakters aangeeft.

FORTRAN - Formula Translator; hogere programmeertaal, speciaal ontwikkeld voor het uitvoeren van rekenkundige bewerkingen door de computer.

FPLA - Field Programmable Logic Array, PLA die door de gebruiker kan worden geprogrammeerd. FPLA's worden veel gebruikt als besturingsorgaan van een bit-slice processor.

FSK - Frequency Shift Keying, methode van datatransport waarbij een logische 0 en 1 worden voorgesteld door signalen met verschillende frequenties.

EIKELTJES MARKT

TE KOOP:

16K CMOS geheugenkaart.
Inlichtingen bij:
W. SALDEN



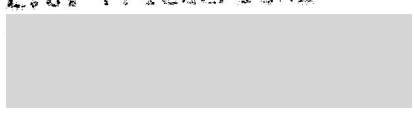
TE KOOP:

Losse foetsen voor de Accorn Atom
Inlichtingen bij:
Lex van Zantvoort



TE KOOP:

ACCORN ATOM 12K + 12K sets
voeding en BBC kaart voor F 500,-
L.J. Friedrichs



BOOTSTRAP ROUTINE

Door Charl de Moor

Deze bootstrap opstart routine is geschreven voor de 6502 bootstrap van het Bronsgroen Eikeltje Nr.7 of Acorn Nieuws 3.7. De routine kan los gebruikt worden maar kan ook een plaatsje vinden in de schakelsoft of in een Utilityrom. (voor mensen zonder geheugenkaart of schakelkaart). De atom zal bij gebruik van deze routine opstarten met de tekst "ACORN ATOM DOS BASIC" wanneer een disc-drive aanwezig is, de disk wordt uiteraard uiteraard automatisch aangezet. Het test-byte dat hiervoor gebruikt wordt is de data #00 op adres #EFAA, dit hoeft niet altijd overeen te komen omdat er meerdere versie's bestaan. De tekst "BOOTSTRAP" die hieronder volgt kan iedereen zelf veranderen en is hier alleen als voorbeeld gekozen. De bootstrap zal verder na een break niet meer terug springen naar de tekst-page #29 maar blijven staan waar hij stond. De oude break kan men weer verkrijgen door het samen intoetsen van shift en break.

Was er in de textpage voor de break nog een programma aanwezig dan zal er na de break informatie over de textpage en de top van het programma verschijnen. Men kan nu zonder old of end direkt weer verder werken aan datzelfde programma.

De plaats van de routine in het geheugen is niet belangrijk echter deze uitvoering is zodanig geschreven dat hij direkt werkt bij een beginadres van bijv. #1F00, #2F00, #3F00 enz. Wilt men de routine toch op een andere plaats zetten dan zullen de interrupt-vectoren niet direkt achter het programma gezet kunnen worden omdat deze niet de laatste 6 bytes van een 4Kbyte geheugen blok zijn. Men zal nu zelf de vectoren in het geheugen moeten zetten. Voor de instelling van de bootstrap wordt verwezen naar het artikel "bootstrap" in de Acorn Nieuws van december 1984.

Verder nog veel plezier met deze routines:

```
10REM BOOTSTRAP ROUTINE
20REM VERSIE 2.5
30REM CHARL DE MOOR
40REM DATE: 18.12.84
50 DIM L9, A8, B8; @=0; F.L=0T09; LLL=#FFF; N.
60 P. $6$12 "BOOTSTRAP ROUTINE" "VERSIE 2.5 CHARL DE MOOR"
70 P. " VOORBEELD STARTADRES: #.EED" " " " = PAGINA: 0-F"
80 P. " BIJ ANDER START ADRES INTERRUPT" " POINTERS ZELF"
90 P. "AANPASSEN" " ZIE A.N. DEC.1984 BLZ.68"
100 IN. "START ADRES : "Z; IF ?18=Z/255; G.100
110 IN. "PRINTER (J/N)" $A
120 IN. "LISTING (J/N)" $B
130 P. "PASS: 1" $21; F. I=1T02; P=Z; ?#E8=0; IF I=16.170
140 P. $6$BI " $21
150 IF ?A=CH "J" P. $2
160 IF ?B=CH "J" P. $6
170E
180: LLO; LDX #17; LDA #FE; CMP #0A; BEQ LL1
185 LDA #0A; STA #FE; LDA #29; STA #12
```

```

190:LL1;LDA#FF9A,X;STA#204,X;DEX;BPLLL1;TXS;TXA;INX
200 STX#EA;STX#E1;STX#E7;STX#CB;LDX##33
210:LL2;STA#2EB,X;DEX;BPLLL2;LDA##07;STA#B002;LDA##8A;STA#B003
220 JSR#F7D1;J; !P=#0F0C0A;P=P+3
230 $P=" ACORN ATOM ";P=P+LENP;I;LDA#EFAA;BNELL3
240 JSR#E000;JSR#F7D1;J;$P="DOS ";P=P+LENP;I;NOP
250:LL3;JSR#F7D1;J;$P="BASIC";P=P+LENP;!P=#0D0A;P=P+2
260 $P=" BOOTSTRAP";P=P+LENP;!P=#0D0A0A;P=P+3;I
270 LDA#B001;CMP##7F;BNELL4 SHIFT TOETS ?
280 JSRLL8;JMP#C2B2
290:LL4;LDA#12;STA#0E;LDA##01
300 LBRA;STA#0D;TAY;LDA(#0D),Y;CMP##0D
310 BNELL3;INY;LDA(#0D),Y;BPLLL5
320:LL5LDY##00;LDA##0D;STA(#0D),Y;INY;LDA##FF;STA(#0D),Y;INY
330 STY#0D;JSRLL8;JMP#C2B6
340:LL6;LDY##00
350:LL7;INY;BEQLL5;;LDA(#0D),Y;CMP##0D;BNELL7
360 JSR#CDBC;LDA(#0D),Y;BNELL9;INY;BNELL7
370:LL9;JSR#F7D1;J;!P=#0D0A;P=P+2;$P=" TEXT POINTER: #"
380P=P+LENP;I;NOP;LDA#12;JSR#F802;JSR#F7D1;J;!P=#0D0A;P=P+2
390 $P=" TOP OP      : #";P=P+LENP;I
400 LDA#0E;JSR#F802;LDA#0D;JSR#F802;JSR#FFED;JSRLL8;JMP#C2CA
410:LL8;JSR#F7D1;J;$P=" READY";P=P+LENP
420 !P=#0D0A0A;P=P+3;I;NOP;RTS
430I;N.;IF7#EB=#23;G.470
440 !P=#FFC70206;P=P+4;?P=ZZ256;P?1=Z/256;P=P+2;!P=#FFB2;P=P+2
450 IFP%1024<>OP.$A"INTERUPT POINTERS AANPASSEN !"
460 P.$6"READY""TOP OP: #"$P"LENGTE: #"$P-Z" BYTES"";END
470 P.$6$7$7$7"OUT OF RANGE DETECTED"";END

```

LET OP:

Het kan zijn dat de routine bij gebruik met sommige schakelsoften problemen oplevert.

Mochten er onverhoopt problemen zijn of heeft U nog suggesties, bel of schrijf dan even naar C. de Moor of E. Sanders.