

\*

# HET

\*

# BRONSGROEN

\*



- T J E

\*

= 2 =

\*

## PROLOOG

Het is uitermate positief te kunnen vaststellen, dat het eerste Bronsgroen Eikeltje met instemming is ontvangen.

Tevens kan worden geconstateerd, dat het in onze regio gonst van de activiteiten, zowel op het gebied van de hard-, als op het gebied van de software.

Regelmatig druppelen de listings, beschrijvingen e.d. binnen en kan er een keus worden gemaakt uit de cogst.

Als U dit leest, hebt U reeds de tweede aflevering van Het Bronsgroen Eikeltje in handen. Er is geprobeerd om het met de inhoud iedereen zo'n beetje naar de zin te maken. U kunt ervan verzekerd zijn, dat, ook in de toekomst, zoveel mogelijk geprobeerd zal worden om dit blad zo veelzijdig mogelijk te doen zijn.

Wat iedere keer weer opvalt op een vergadering of bijeenkomst is, dat alles zo vriendelijk en welwillend verloopt. Iedereen helpt iedereen, er wordt driftig gehobbied, EPROM's worden geprogrammeerd en de demonstraties stralen van alle kanten.

De resultaten kunnen dan ook niet uitblijven, zoals iedereen vast kan stellen.

Wilt U met Uw op- of aanmerkingen Uw redactie bijstaan, dan graag. Een periodiek wordt toch geboren uit samenwerking!

Naast de punten, die de aandacht hebben, en waarvan U een lijst vond in aflevering 1, wordt ook nog aandacht besteed aan het punt, dat computers kunnen storen in andere apparatuur. Hierop zal in de toekomst zeker worden teruggekomen.

Evenals de vorige keer hartelijk dank aan allen, die op enigerlei wijze meegewerkt hebben aan dit Bronsgroen Eikeltje, want zonder die medewerking was het onmogelijk.

Dat ons Bronsgroen Eikeltje moge groeien tot een Bronsgroene Eikel, is de wens van Uw redacteur.

De vorige regioavond was erg interessant. Een behoorlijk aantal computers (waaronder een BBC en een Acorn met BBC-kaart) was aanwezig. (Brengt U die van U ook eens mee. Steeds de zelfde mensen, is en blijft voor hen niet leuk. Misschien heeft U zelfs iets aardigs om te demonstreren.)

Een speciaal woord van dank aan de heren Marchal en Ronda voor hun werk, ROMARBOX, en aan Marchal nog eens apart voor zijn werk aan en demonstratie van de MDCK. (De software hiervoor is bij hem persoonlijk te verkrijgen. Wel even afspreken.)

Aanvulling ledenlijst augustus 1983.

INHOUD BRONSGROEN EIKELTJE nr. II

- 1 - Voorwoord van de Heer Campers.
- 2 - Aanvulling van de ledenlijst gepubliceerd in Het Bronsgroen Eikeltje van augustus.
  - Inhoud van dit nummer
  - Eikeltjes-markt
- 3 - De SOURCEMAKER, eerst keken we gewoon naar de geheugenplaats, toen kwam als hulp een disassembler en nu hebben we de Source-maker. Marchal en Ronda hebben weer hun best gedaan.
- 5 - Complex rekenen. Werkelijkheid en denkbeeldigheid bij het rekenen, onze Acorn zorgt voor de oplossing.
  - Tea Cups, een eenvoudig spelletje.
- 6 - Wereldtijd. Het tijdsverschil rekent de Acorn voor U uit.
- 7 - Advertenties in onze eigen Eikeltjes markt.
  - Slippertjes, deze komen helaas ook voor.
  - Brief van John van Schaijik met tips.
- 8 - Adressen van routines uit het operating system. Kent u deze al?
  - Tips van Wim Ernst.
- 9 - De 6847 en zijn (on)mogelijkheden. Een artikel uit Practical Electronics. Wie neemt deze uitdaging aan? + FASTER Atom.
- 12 - Twee routines die u op kunt nemen in een eigen 'toolkit' of in de 'schakelsoft', een voor PAUZE en een voor TABULATIE. En een routine om het scherm (gedeeltelijk) schoon te maken.
- 13 - De Battery Back-up van onze Schakelkaart, Leon Heesakkers.
- 15 - De SCHAKELKAART Wilt U ook eens het fijne hier van weten? Leon Heesakkers legt U het haar fijn uit. Compleet met het schema, en wijzigingen. Ons aller dank Leon.

EIKELTJES-MARKT

Wie wil mee werken aan een CAD-programma. Graphics op 'hoog' niveau, dus niet voor spelletjes.

GEVRAAGD!!

H.Heuts, Kleine Steeg 44, Sittard.  
tel: 04490-19969

Op de volgende regio-avond is er een demonstratie van het CALC-ROM.  
Wat kan je er nu eigenlijk mee doen!  
Deze demonstratie wordt verzorgd door:  
John van Schaijik uit Roermond.

AANGEBODEN!!

# \* \* SOURCEMAKER \* \*

## HET GEBRUIK VAN DE SOURCEMAKER

### INLEIDING

In het algemeen wordt een programma in machine-code via een hulpprogramma (in Basic) aangemaakt. Dit hulpprogramma wordt ook wel Source-programma genoemd. De assembler zorgt er dan voor, dat de mnemonics (zoals b.v. STA of LDA) in machine-code wordt omgezet. Deze code kan met behulp van een disassembler-programma op eenvoudige manier worden teruggelezen. Willen we de code veranderen, dan zullen de veranderingen via de Source moeten worden aangebracht. Wanneer we echter niet de beschikking hebben over het Source-programma, dan wordt het problematisch.

We kunnen dan het Source-programma maken aan de hand van een disassembler listing. Er is echter een veel eenvoudiger manier:

Het programma - - - SOURCEMAKER

### WERKING

Laden programma

Het programma dient te worden geladen via `LOAD"SOURCEMAKER"`

Het programma is zodanig ontworpen dat het zowel in het hoge geheugen-gebied als ook in het lage geheugen-gebied kan worden gerund. Het mag dus in b.v. `#8200`, maar ook in `#2900` worden geladen.

Starten programma

Nadat het programma is geladen, moet de textpointer worden verplaatst naar een waarde 2 hoger dan waar het programma werd ingeladen. Dus is het programma geladen in `#8200`, dan moet de textpointer naar `#84`.

Het programma werkt conversationeel. Dat wil zeggen dat aan de hand van een aantal vragen het programma bepaald welke uitvoering gewenst is. Na het RUN-commando is de eerste vraag die wordt gesteld:

STARTADRES ?

Hier dient het beginadres van de machine-code te worden ingevuld. Dit adres mag met of zonder `#`-teken worden ingevoerd. Bij ongeoorloofde karakters volgt de melding "INPUT FCUT". Het startadres kan dan opnieuw worden ingetikt. Bijvoorbeeld:

STARTADRES ? A002

Dan volgt de vraag:

WINDADRES ?

Het adres moet op dezelfde wijze worden ingevoerd als bij de eerste vraag, alleen mag hierbij direct een return worden ingetikt. Wordt op deze vraag een return gegeven, dan zal het programma verder alleen nog als disassembler functioneren.

Is het eindadres opgegeven dan is de volgende vraag:

ADRES VOOR OPSLAG SOURCE ?

Hier voeren we het adres in waar we de source willen hebben. Ook hier kunnen we volstaan met een return, waarna het programma alleen als disassembler werkt.

Opmerking: Wanneer het opgegeven adres binnen het programma valt, dan volgt de melding: `GEEN RUIJTE VOOR OPSLAG SOURCE !`

Het programma stopt hierna.

Hebben we bij bovenstaande vraag wel een adres opgegeven, dan volgt:

GEEF SOURCENAAM ?

Deze vraag spreekt voor zich.

Dan krijgen we nog 2 vragen, die ook worden gesteld wanneer we het programma alleen als disassembler gebruiken.

PAGE MODE J/N ?

PRINTER J/N ?

Ook deze vragen behoeven geen verdere toelichting.

Gebruiken we het programma als disassembler dan volgt de melding:

pass 1

Hierna volgt dan de gedisassembleerde code over het scherm. De opties (PAGE MODE en PRINTER) zijn nu actief.

Wanneer er geen eindadres was opgegeven, dan loopt het programma door tot ca. 31 februari. In het andere geval stopt het programma bij het bereiken van het eindadres.

Opmerking: Het disassembleren kan d.m.v. de shift-toets worden afgebroken. Het programma komt dan weer met de vraag "STARTADRES".

Zijn we daartegen bezig met de aanmaak van een source-programma dan vervolgt het programma na "PRINTER J/N" met:

EIGEN LABELS INVULLEN:

LABEL 1 ADRES ?

We kunnen nu adressen invullen, die later in de source van een label moeten worden voorzien. Deze loop kan worden gestopt d.m.v. een return. Hierna vervolgt het programma met "pass 1" en is de gedisassembleerde code op het scherm te zien.

Opmerking: Wanneer het aantal labels groter is dan 250, volgt de melding: TEVEEL LABELS !

Het programma stopt dan.

Bij het bereiken van het eindadres volgt de melding:

pass 2

WACHT....

Het programma is nu bezig met de aanmaak van de source. Dit kan, afhankelijk van de grootte van de code, enkele seconden tot tientallen minuten duren.

Opmerking: Tijdens de aanmaak van de source kan het programma worden afgebroken met de melding: GEEN RUIJITE VOOR OPSLAG SOURCE !

Het eventueel aangemaakte stuk programma is NIET te gebruiken.

Is het aanmaken van de source zonder problemen verlopen, dan stelt het programma tot slot de vraag:

TABEL MAKEN J/N ?

Indien we bij deze vraag een return of N intikken dan stopt de SOURCE-MAKER.

In het andere geval kunnen we nu in het Source-programma een tabel opnemen. Dit kan bijvoorbeeld Data zijn die voor of achter de machine-code ligt. Het begin- en eindadres van dit stuk Data geven we dan op bij de vragen "STARTADRES" en "EINDADRES". De Sourcemaker stopt dan met de melding:

klaar

We kunnen nu de tekstpointer verplaatsen naar het geheugengebied waar de Source is aangemaakt.

VEEL SUCCES . . . .

m.v.g.

A.Marchal

E.konda

INHOUD

Assembler	A	Meldingen: klaar	B
Data	B	Meldingen: pass 1	A,B
Disassembler	A	Meldingen: pass 2	B
Geheugengebied	A,B	Meldingen: TEVEEL LABELS	B
INLEIDING	A	Meldingen: WACHT	B
Laden programma	A	Shift-toets	B
Machine code	A,B	Source-programma	A
Meldingen: GEEN RUIJITE	A,B	STARTADRES	B
Meldingen: INPUT FOUT	A	Starten programma	A

vervolg INHOUD artikel SOURCEMAKER

Textpinter	A,B		
Vragen: EIGEN LABELS	B	Vragen: PAGE MODE	A
Vragen: EINDADRES	A,B	Vragen: PRINTER	A
Vragen: GEEF SOURCNAAM	A	Vragen: STAARTADRES	A,B
Vragen: OPSLAG SOURCE	A	Vragen: TABEL MAKEN	B
		WERKING	A

\* \* \*

C O M P L E X - R E K E N E N  
'n wiskunde hulpje

```

4 REM COMPLEX REKENEN
5 REM VOOR ACORN ATOM BEWERKT DOOR C EN W CAMPERS ROERMOND
6 REM DATUM 1-8-1982
10 P.$12"COMPLEX REKENEN"====="1'07
20 FIN."EERSTE COMPLEXE GETAL - REEEL GEDEELTE "%A;P.'07
25 FIN."IMAGINAIR GEDEELTE "%B;P.'07
30 FIN."TWEEDE COMPLEXE GETAL - REEEL GEDEELTE "%C;P.'07
40 FIN."IMAGINAIR GEDEELTE "%D
45 %Z=((%A*%C)-(%B*%D)); %Y=((%B*%C)+(A*%D))
50 %W=((%A*%C)+(B*%D))/((%C*%C)+(D*%D))
51 %V=((%B*%C)-(%A*%D))/((%C*%C)+(D*%D))
52 P.$12$7$7"UITKOMSTEN"====="1"Z1 ="
71 FP." "%A;PIF %B>=0 THEN P."+"
75 FP.%B"J""Z2 =" "%C;PIF %D>=0 THEN P."+"
85 FP.%D"J""Z1+Z2 =" "%A+%C;PIF %B+%D>=0 THEN P."+"
122 FP.%B+%D"J""Z1*Z2 =" "%Z;PIF %Y>=0 THEN P."+"
132 FP.%Y"J""Z1/Z2 =" "%W;PIF %V>=0 THEN P."+"
142 FP.%V"J""====="1
155 IN."NOG EEN BEREKENING J/N "N; IF K=N G.190
170 IF K=J P.$12;%A=0;%B=0;%C=0;%D=0;G.10
175 G.155
190 END
    
```

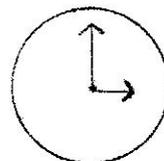
T. E A - C U P S  
'n spelletje

```

0P.$12"shift=LEFT rept=RIGHT"" "WAIT";F.N=OT0200;WAIT;N.
1CLEAR0;S=0;C=0;?#E1=0
2P.$30" CUPS SCORE"
4K=33260;?K=255
8F.C=OT050
10I=A.R.%32+32832
20P.T=OT013;L=T*32+P
21IF?#B002&64=0;K=K+1;?K=255;?(K-1)=192;IF K=33279;K=K-1
22IF?#B001&128=0;K=K-1;?K=255;K?1=192;IF K=33248;K=K+1
23?33248=192;?33279=192
25WAIT;WAIT;WAIT
30?L=123;?(L-32)=192
35IF L>33247;?L=192
40N.
45IF K=L;P.'07;S=S+1;?32790=S/10+48;?32791=S%10+48
46?32778=C/10+48;?32779=C%10+48
48N.;E.
    
```



## W E R E L D T I J D



```

10 REM WERELDTIJD
20 REM AUTEUR: CHRISTIAN BOHN
30 REM UIT CHIP JUNI 1981
40 REM BEWERKT VOOR ATOM DOEL W ERNST, ECHT
50 PRINT $12;Z=0;e=3;F=10000;PRINT:'''
60 PRINT"      +++++++"
70 PRINT"      +++ WERELDTIJD +++"
80 PRINT"      +++++++"
85 FOR N=1 TO 30;WAIT;NEXT N
90 IF Z=1 THEN GOTO 240
100 DIM P(20),Q(1);Z=1
110 INPUT'''PLAATS (IN NEDERLAND)'''$P
115 INPUT"ZOMER- OF WINTERTIJD Z/W"$$Q
120 PRINT"GEEF DE JUISTE TIJD AAN ";INPUT"H,M,S"
125 IF $Q="Z";T=T-10000
130 REM SAN FRANCISCO
140 A=T-90000;IF A<0 THEN A=A+240000
150 REM NEW YORK
160 B=T-60000;IF B<0 THEN B=B+240000
170 REM MOSKOU
180 C=T+20000;IF C>240000 THEN C=C-240000
190 REM MBAY
200 D=T+20000;IF D>240000 THEN D=D-240000
210 REM MELBOURNE
220 E=T+90000;IF E>240000 THEN E=E-240000
230 IF $Q="Z";T=T+10000
235 PRINT $12;GOTO 60
240 PRINT'''$P;GOSUB 370
250 PRINT T/F,(T%F)/100,T%100
260 PRINT"SAN FRANCISCO";GOSUB 370
270 PRINT A/F,(A%F)/100,A%100
280 PRINT"NEW YORK";GOSUB 370
290 PRINT B/F,(B%F)/100,B%100
300 PRINT"MOSKOU";GOSUB 370
310 PRINT C/F,(C%F)/100,C%100
320 PRINT"BOMBAY";GOSUB 370
330 PRINT D/F,(D%F)/100,D%100
340 PRINT"MELBOURNE";GOSUB 370
350 PRINT E/F,(E%F)/100,E%100
360 PRINT"DRUK TOETS";LINK#FPE3;RUN
370 DO PRINT" ";UNTIL COUNT=20;RETURN

```

## EIKELTJES-MARKT

## AANGEBODEN:

Acorn Atom met schakel- en geheugenkaart, diverse toolkits o.a.  
 CALC-rom, Josbox en KOMAR.  
 Inlichtingen: H.Hoenen, Floribertusstr.3, Brunssum, 045-253919.

## AANGEBODEN:

Wisser voor EPROM's, dus lamp met behuizing.  
 Inlichtingen: W.Ernst, Cypresstr. 94, Echt, 04754-5653

## TIPS van John van Schaijik.

----Op een van de Club-avonden hoorde ik dat er geen foutloos werkende afrondings-routine was; sommige getallen o.a. 2,345 zouden problemen geven. Hierbij een listing van een programma'tje waar van ik nog geen fout gevonden heb. Listing 1.

----Voor degene die met een Disk-drive werken is misschien het volgende van belang. Het commando ~~W~~VDU maakt het scherm schoon (misschien nog meer?).

Met het commando ~~W~~SPOOL is het mogelijk een FILE die ge-DELETED is weer terug te krijgen. U typt na abusievelijk wissen ~~W~~SPOOL"FILENAME" (quotes mogen weg) en hup een ramp is weer voorkomen.

Het aandruk-viltje in de Drive blijkt nog al eens problemen te geven (Dank aan J.Vogten voor de tip). Het laat gemakkelijk los. U kunt met een zaklamp van buitenaf zien of het nog aanwezig is.

Het programma ATOM-STORE geeft bij gebruik van een D-Drive problemen om de DATA weer terug van schijf te krijgen. Hierbij een laad-routine die alle problemen oplost. Listing 2.

Roermond 13.08.1983

Met vriendelijke groet,  
John van Schaijik.

## LISTING 1

```

1 REM SUBROUTINE "SCHRIJVEN
  ACHTER DE KOMMA"
2 FIN.%A;GOSUB a
20 END
10000a C=0
10005 A=0;B=0;C=0;D=0
10010 %B=%A*1000;A=%B
10020 B=A/1000
10030 C=1000*B
10040 D=A-C;B=D+5;D=D/10
10050 P.B,"",",D;R.
```

## LISTING 2

```

20 P.Q12
25 P."adressen-bestand""
30 P."INSERT FLOPPY NO.1"
32 P."ALS KLAAR DRUK OP SPATIE-
  TOETS"
35 LINK#EPEB
50 P.Q21
60[;JSR #B000;]
65 P.Q6
66 WLO.STDAT 2800
70 WLO."STORE" 8200
72 P."EVEN GEDULD"
75 P.Q=1TO100;WAIT;N.
80 ?18=#82
100 RUN
```

## SLIFFERTJES!!!!

Bronsgroen Eikeltje nr.1

Screencopy blz. 14

270 ;IF N=1;R=P

390;PF12LDA#93,X;JSRMP20;STY#93,X;DEX;BNRMP12;INC#90;LDA#90

Bij de veranderingen voor de MX-82 moet regel 150 zijn:

150 R!16=#1B4108;R.

In het volgende nummer van Het Bronsgroen Eikeltje zal nog een kortere routine voor de GP-250 opgenomen worden.

Het programma voor de GP-250 zou ook werken voor de STAR-printer(?).

## EIKELTJES-MARKT

## AANGEBODEN:

Heeft U een printer en bent U geïnteresseerd in recycling printer-papier en/of printer etiketten? Verkrijg-baar in wit en grijs, grijs voor ca. f 35,- per 2000 vel.

Inlichtingen: A. de Jong, St. Josephstr. 82, 6136 VW Sittard.

tel. 04490-16097.

\* TIP-tip \* TIP-tip \* TIP-tip \* TIP-tip \* TIP-tip \* TIP-tip \*

Twee tip's uit de praktijk. Ingezonden door Wim Ernst.

Mensen die in het bezit zijn van een MONITOR of een tot monitor omgebouwde T.V. en vinden dat de LICHT OMBRENGST onvoldoende is, kunnen het volgende doen.

Op de hoofdprint (computer) LINK5 doorkrassen. Pen 39 van de 6847 via een weerstand van  $\pm$  4700 Ohm met de + 5 Volt verbinden.

Degene die een printer hebben en deze LINK reeds doorgekrast, hebben pen 39 reeds aan massa gelegd. Door deze verbinding te vervangen door een weerstand van 4700 Ohm naar de + 5 Volt wordt de lichtopbrengst met ca. 50 % verhoogd.

Mensen die het gemis van een AUTO-REPEAT functie van de toetsen als een lacune ervaren, kunnen dit verhelpen door een draadbrug over de REFP-toets te solderen. Het is echter wel even wennen!!!

\* \* OPERATING SYSTEM ROUTINES \* \*

#C3C8 als #C3CB maar leest eerst argument in (#C8BC)  
 #C4E1 roept eerst #C78B aan en vervolgt dan met #C4E4  
 #C55B interpreteert eerst volgend statement  
 #C8E3 als #C8E1 alleen worden de variabelen van: #0321,Y naar #15,X - #0357,Y naar #33,X - #0330,Y naar #24,X - #0372,Y naar #42,X verplaatst  
 #C97C verplaatst de accu naar: #16,X #25,X #34,X #43,X en naar 0  
 #CDBC telt de Y-waarde bij de TOP pointer op  
 #F379 print een spatie  
 #F72C voert m.b.v. een twee bytes pointer op (#54),Y aan AND uit en saved deze in (#54),Y  
 #F731 voert m.b.v. een twee bytes pointer op (#54),Y een EOR uit en saved deze in (#54),Y  
 #F736 voert m.b.v. een twee bytes pointer op (#54),Y een ORA uit en saved deze in (#54),Y  
 #FA0B incrementeer #0,X en #1,X als zijnde een 2 byte pointer  
 #FB7D delay loop; wacht 2 seconden  
 #FB8A voer een delay van 0,1 sec. uit  
 #FBEE haalt byte van tape; bij exit byte in accu, X en Y onveranderd  
 #FD48 voer op de cursor een EOR uit met de accu waarde  
 #FE40 indien de accu kleiner is aan 5 wordt er "REWIND TAPE" geprint indien de accu groter is dan 5 wordt er "RECORD TAPE" geprint  
 #FE55 print ASCII karakter op scherm, control codes worden geserviced, alle registers blijven bewaard, de printer wordt niet aangesproken  
 #FE5F haalt accu en stack register van stack en plaatst #E4 en #E5 in resp. X en Y register  
 #FEB1 kijkt of ingetoetste toets een controle karakter is en voer dit zonodig uit  
 #FF10 stuurt de waarde van de accu buiten de computer om naar de printer, mits deze is geenabled (dus ook de waarde 2 en 3)  
 #FFE3 leest karakter en plaatst dit in de accu  
 #FFE6 leest karakter in de accu en stuurt daarna een copy naar het beeldscherm  
 #FFED return/line feed routine  
 #FFEF operating system write character routine  
 #FFF2 return routine  
 #FFF7 interpreteert een string van karakters op adres #100 en afgesloten met een carriage return als command; fouten worden gemeld met BREAK alle reg. worden gebruikt en de decimal-mode flag wordt op bin. gezet.



Two of the most popular microcomputers currently on sale, the Acorn Atom and the Dragon, have many similarities because they both use the same video display chip, the Motorola 6847. This versatile chip not only contains 64 programmed characters for a 32 x 16 text display, but will also give graphics displays up to a resolution of 256 x 192. However the 6847's internal character set does not include lower case, and so both the Atom and the Dragon display lower case as inverted upper-case characters.

NEW CHARACTERS FOR ATOM

The following circuit shows how to add an external character generator to the 6847 to provide 256 different characters in the alphanumeric display modes. The circuit can be added as a plug-in module, and no alterations are necessary to the Atom. With careful choice of the new character set lower-case can be added, while retaining compatibility with the Atom's existing VDU software. The circuit was designed by W. Chadwick of Welwyn, and the following description is based on his letter.

CIRCUIT DESCRIPTION

The new character generator is stored in a 4K x 8 EPROM. The top 8 address lines of this are connected to the 6847's Video Data Bus; see Fig.1. When the 6847 puts data onto these lines they will select one of 256 groups of 16 EPROM locations. Each displayed character is 8 dots wide and 12 scan lines deep, so that the first 12 EPROM locations of each group contain the precise bit patterns for each character.

The lowest 4 address lines of the EPROM are connected to a row counter (74LS161). This counter is clocked once per scan line by the HS signal from the 6847 and so will sequentially select character row information from the EPROM. The 6847 provides a clear control, RP, every 12 scan lines for the row counter. The beginning of each display frame contains a border of non-active display lines, but unfortunately the 6847 does not provide an RP pulse to clear the character counter for the first character row. Thus, to ensure that the top line of the display is displayed correctly the Frame Synch (FS) signal is used to preset the row counter to 9 at the beginning of each frame.

When using the graphics modes of the 6847 the video data must be passed on to the chip unchanged. The control line A/G is used to select between alphanumeric and graphics modes. Since only 12 out of every 16 ROM locations are used for each character displayed, the 4 unused locations in each case are programmed to contain logical mappings of the top eight address lines. The signal A/G is used to hold the row counter preset to row 12 or above when graphics are selected, thus causing the ROM to appear transparent. To illustrate how this works the coding of a single character is shown in Fig.2.

CHARACTER SET

To maintain compatibility with existing software, the following character set is recommended:

Character Code:	Description:
00 to 1F	@ to ← as standard Atom
20 to 3F	Space to ? as standard Atom
40 to 7F	White block characters as standard Atom
80 to 9F	Inverted @, a to z, inverted [A] ↑ ←
AO to BF	Inverted space to inverted ? as standard Atom
CO to FF	64 custom characters; 00 should be blank

Two examples of possible character sets following this design are given in Fig.3. The only limitation of this set is that if the cursor is placed

on an upper-case character it will appear as lower-case and vice-versa.  
**CIRCUIT CONSTRUCTION**

The circuit can be constructed on a piece of perforated board using a 40-pin header to fit in the socket of IC31. The board should easily fit inside the Atom's case, though be careful to avoid the case support pillar. The only precaution needed is to keep wires carrying video signals as short as possible.

**POSSIBLE EXTENSIONS**

In the Atom the 6847's Colour Select Signal CSS is used to select between two possible sets of four colours. However, on monochrome displays this signal has little use, and so it can alternatively be used to add extra facilities to the character generator board. At power-up the CSS signal is set to logic 0. By gating CSS with A/G it is possible to cause the counter to preset to either row 14 or row 15 depending on the state of CSS. If rows 13 and 15 are programmed with a true logic mapping, and rows 12 and 14 with an inverted mapping, as shown in Fig.2, then the CSS signal (which is controlled by bit 3 of the port at address B002) can be used to invert the entire graphics display. The circuit in Fig.1 shows the links made for monochrome use to provide this facility; for colour use, both links should be changed so that CSS operates in the usual way. Good animated effects can be achieved with the colour graphics modes by drawing the background colour, then changing CSS to make the drawing instantly visible. An alternative use for CSS would be to switch between one of two character generator EPROMS, one compatible with the existing Atom software, and the other giving 256 totally different display characters.

**FASTER ATOM**

Two other ways of improving the Atom's circuit have been submitted by R.Brain of Devon. First, the Atom can be persuaded to run at 2MHz rather than the usual 1MHz by fitting a switch as shown in Fig.4. Note that some of the other components, such as the memory or the 8255, may not be capable of the extra speed, and faster versions may have to be fitted. The second modification, shown in Fig.5, speeds up the rate at which characters are printed to the Atom's screen. Normally, character output is synchronised with the FS output of the 6847 to prevent screen noise. By feeding a high frequency to the 8255 this delay of 1/60th of a second is circumvented.

Fig. 2 ROM programming details for a sample character.

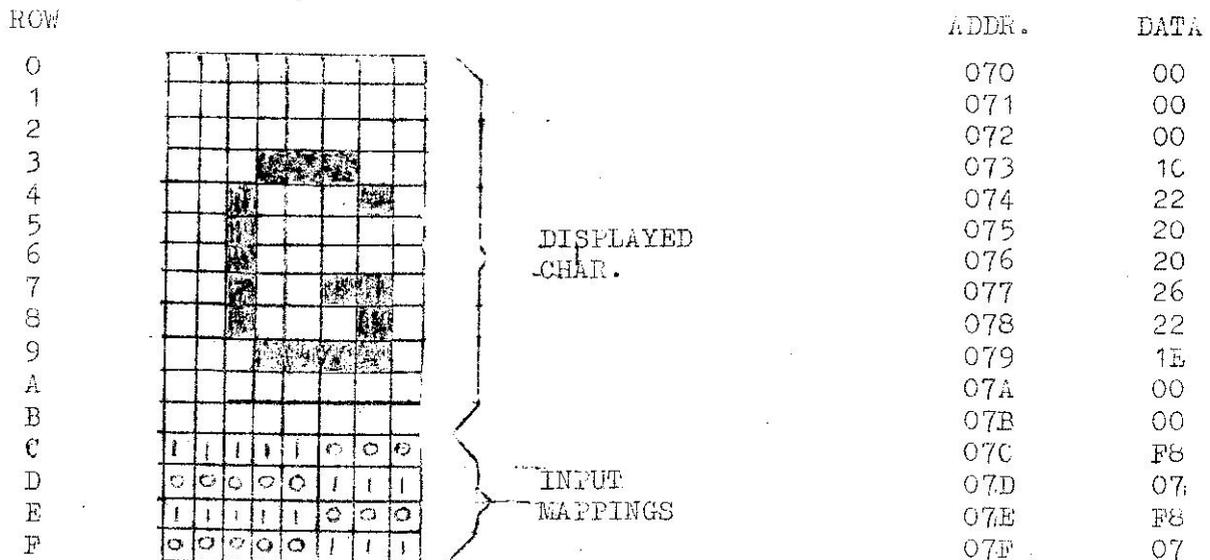
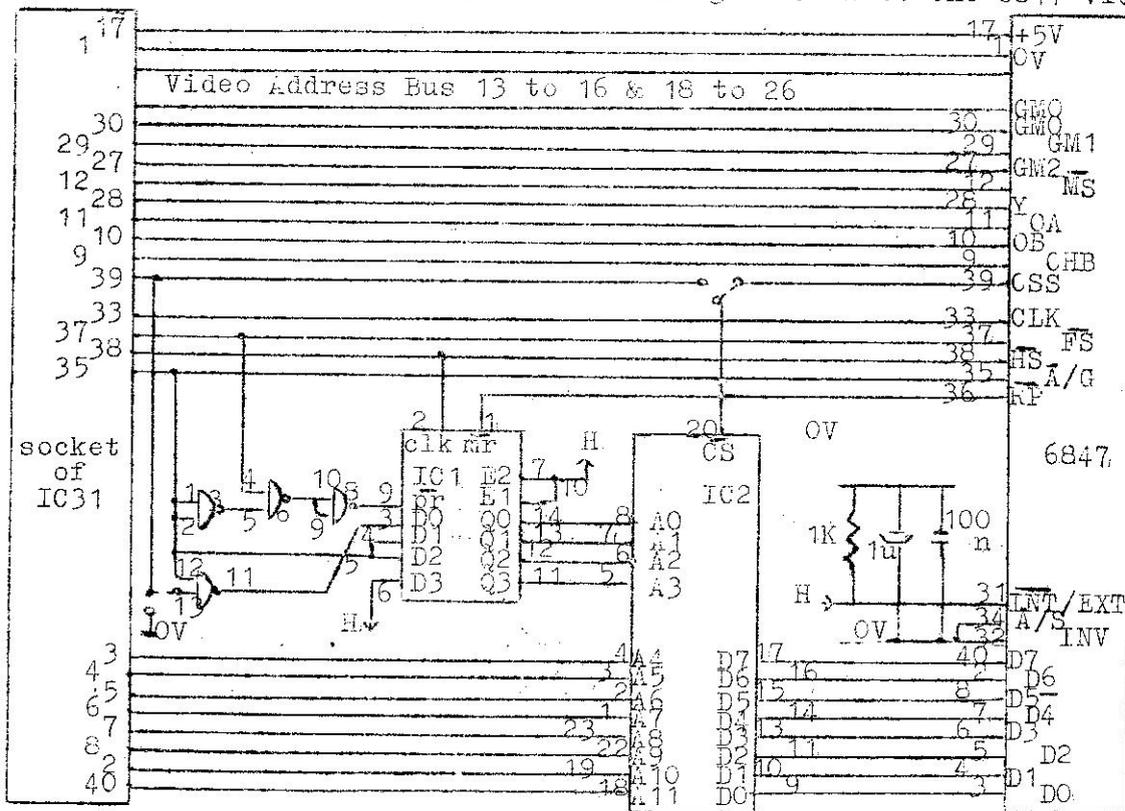


Fig.1 Circuit adds an external character generator to the 6847 Video chip



D = LS00                      IC1 = LS161                      IC2 = EPR0M char.gen. 2532

Fig.4 Circuit increases the speed of the Atom to 2 MHz.

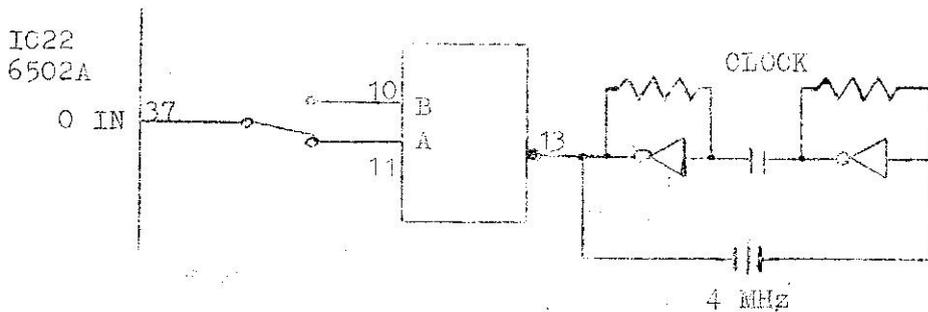
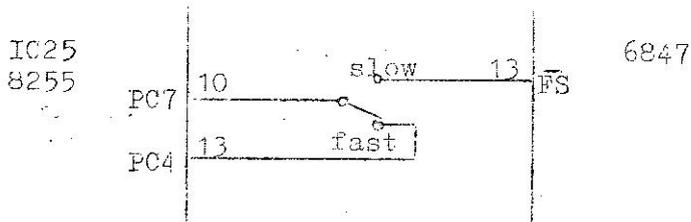


Fig.2 Examples of possible character sets. Helaas was de copie die wij van dit artikel hadden zo slecht dat wij dit voorbeeld niet over konden nemen. Zie hiervoor het drukwerk-archief.  
 Fig.5 Circuit increases the speed of displaying characters on the screen.



## TABULATOR

Deze routines verzorgen een horizontale en verticale tabulator. Zowel op het beeldscherm, als op de printer!!  
Ze MOETEN opgenomen worden in een 'toolkit' of 'schakelsoft'.  
Formaat: HTAB x of VTAB x, waarbij x een getal is, dat aangeeft waar het printen begint. Als argument (x) is ook zondermeer een variabele of zelfs een wiskundige uitdrukking mogelijk.

```

0 REM HTAB; VTAB
10 DIM LL(2); INPUT"ASSEMBLE TO"A
20 FOR N=0 TO 2; LL(N)=A; NEXT N
30 FOR N=1 TO 2; P=A;[
40 \ HTAB
50 JSR#C4E1;DEX;STX 4;LDY #16,X;BEQ LL1;DEY
60:LLO JSR#F379;DEY;BNE LLO
70:LL1 JMP#C55B
80 \ VTAB
90 JSR#C4E1;DEX;STX 4;LDY #16,X;BEQ LL1;DEY
100:LL2 JSR#FFED;DEY;BNE LL2;BEQ LL1;JMP#C55C;]
110 NEXT N; C=0;PRINT"CODE VAN #"&A" TOT #"&P-1"; C=8;END

```

## PAUZE

Deze routine maakt een wachttijd mogelijk van max. 4,25 sec.  
Formaat: PAUZE x; waarbij x niet groter mag zijn dan 255 en niet kleiner dan 0. Ze MOET ook weer gebruikt worden in een 'toolkit' of 'schakelsoft'.

```

0 REM PAUZE
10 DIM LL(2);INPUT"ASSEMBLE TO"A
20 FOR N=0 TO 2; LL(N)=A; NEXT N
30 FOR N=1 TO 2; P=A;[
40 JSR#C4E1; DEC4; LDX4; LDA#25,Y; TAY; INY; LDA#16,X; TAX; BEQLL1
50:LLO JSR#FB83
60:LL1 DECO; BEQLLO; JMP#C55B;]
70 NEXT N; C=0;PRINT"CODE VAN #"&A" TOT #"&P-1"; C=8;END

```

## ROUTINE 1

Deze routine maakt het scherm schoon vanaf de cursor. De regel waar de cursor staat wordt ook schoongemaakt.

```

10 DIM LL(2);F.N=0 TO 2;LL(N)=#FFFF;N.
20 IN."OBJ.-CODE"A;F.N=1TO2;P=A;[
30 LDA#11;JSR#FFF4;LDA#DE;STA#91;LDA#DF;STA#92;JMP LL1
40:LLO LDY#0;LDA #20;STA(#91),Y
50:LL1 CLC;LDA#91;ADC#1;STA#91;LDA#92;ADCC#0;STA#92;CMP #82
60 BEQ LL2;JMP LLO
70:LL2 RTS
80;]N.
90 REM DEMONSTRATIE
100 P.#30;F.N=1TO5;P.#10;N.;LINK A;END

```

## EIKELTJES-MARKT

## GEVRAAGD:

Duidelijke copie van het artikel uit Practical Electronics, zie blz. 9-11. Speciaal het voorbeeld van de nieuwe character set. Het drukwerk-archief; op de regioavond of opsturen aan de Heer Tossaint of de Heer Proeme.