

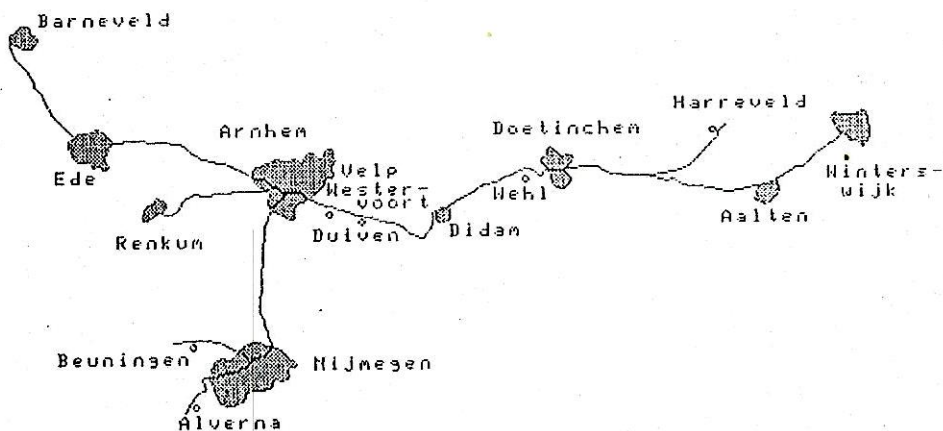
Tempus Fugit

knipselkrant van de Acorn Computerclub

Regio Arnhem



EDAKTIE : G. Bouman



AUGUSTUS

1985

Nieuwe Word Pack Rom

I Inleiding

Bijgaand source-programma (versie 5) vormt een uitbreiding van de bestaande WPR; het kan zowel gekoppeld worden aan de oorspronkelijke versie (op #A000), mits deze in RAM staat en dus gewijzigd kan worden, als aan een gereloceerde versie (bv. naar #7000: zie A.N. 4.3, pp. 56-57).

De uitbreiding neemt, inclusief de papierspaarder (de VDU 6424 routine van Jos Korsmeier), 2K bytes in beslag, waardoor de Nieuwe WPR geheel onder 'write protect' op de 16K RAM kaart geplaatst kan worden in het gebied #6800-#8000 (zie de integrale versie). Maar zoals hierboven reeds aangeduid kan hij bv. ook op #9800-#B000 worden neergezet; in dat geval moet wel de source worden aangepast.

De VDU 6424 routine tenslotte kan ook *onafhankelijk* van de WPR worden aangeroepen, nl. met: 'LINK (startadres + #4D8)'

N.B. Indien de Nieuwe WPR via een P-Charme tabel wordt aangeroepen (zie bv. de integrale versie), kunnen er soms problemen ontstaan: de textfile, die begint op #2900, kan namelijk, wanneer tijdelijk uit de WPR wordt gesprongen en een P-Charme programma elders in het geheugen wordt geRUNd, door P-Charme beschadigd worden, met name het gebied #29BD-#29FF: de adressen van procedures en functies! Het beginadres van de textfile is echter met opzet niet verplaatst: ten eerste om zonder meer textfiles vervaardigd met de oude WPR te kunnen bewerken, en ten tweede om de ruimte voor de textfile zo groot mogelijk te houden: meestal van #2900 tot #6800, d.w.z. 16K bytes.

II Kenmerken

De nieuwe WPR werkt met 2 karaktersets: de normale 3220 VDU in input zowel als in command mode (voor de leesbaarheid), en de 6424 VDU voor output van de tekst naar het scherm (zie het ED64 programma van Ronald Boers in A.N. 4.2, pp. 34-35): de "papierspaarder".

Verder is het mogelijk om, m.b.v. geïnverteerde karakters, speciale printercommando's in de tekst (dus ook binnen een regel!) te integreren, waarbij de tekst, zij het met enige beperkingen, meestal toch correct wordt uitgevuld.

Tenslotte ondervangt de Nieuwe WPR het probleem dat bij sommige printers in de onderstreep mode ook de spaties die de linker marge vormen, worden onderstreept!

III Printercommando's

De printercommando's staan opgeslagen in een tabel, zulks om aanpassing van de codes aan de in gebruik zijnde printer vrij gemakkelijk te kunnen verrichten (zie hiervoor het overzicht van de printer codes door Johan Siegers in A.N. 4.2, pp. 84-89). In deze tabel zijn de codes opgeslagen voor de STAR 56-10 printer, werkend in IBM mode.

De tabel bevat op dit moment 12 printercommando's:

'R' : reset printer
'Q' : Near Letter Quality
'U' : underline
'I' : *italic*
'D' : double-strike
'M' : emphasized
'X' : **enlarged**
'P' : proportional
'E' : elite
'C' : condensed
'^' : ^{superscript}
'S' : _{subscript}

Deze tabel kan zonder meer worden ingekort, uitgebreid of anderszins gewijzigd (zie verder onder *Opbouw Commando-tabel*).

Het inbrengen van de printercommando's (in de EDIT mode) geschiedt op de volgende wijze:

- 1) Druk tegelijk op CTRL en de letter 'a', en laat los.
- 2) Druk vervolgens op de betreffende toets, bv. de U voor underline: gewoon 'u' geeft hierbij de kleine letter, SHIFT 'u' de hoofdletter; beiden worden geïnverteerd op het scherm weergegeven.

Hierbij gelden de volgende conventies:

- 1) Hoofdletter - zet printercommando aan
- 2) Kleine letter - zet printercommando uit
- 3) Er kunnen geen parameters aan het commando worden meegegeven.
- 4) Alle printercommando's moeten, na een <CR>, achter de processorcommando's (bv '.1') worden gezet!
- 5) Bij 'R' (initialisatie van de printer) werkt alleen de hoofdletter: de kleine letter heeft geen uitwerking.
- 6) Zowel Elite als Condensed worden uitgeschakeld naar Pica.
- 7) Superscript en Subscript kunnen zowel door SHIFT '^' als door 's' worden uitgeschakeld.
- 8) Er wordt een vaste enlarged mode gebruikt, die expliciet uitgeschakeld moet worden: dit gebeurt dus niet automatisch aan het einde van een regel.

De printercommando's worden in de textfile opgeslagen met bit 7 'geset' (=1).

In de output mode, naar scherm zowel als naar printer, worden deze printercommando's niet op het scherm weergegeven. Zulks om een correct beeld te krijgen van de vorm waarin de tekst op papier verschijnt. Om dezelfde reden worden voortaan ook de '.o' commando's uitsluitend naar de printer gestuurd.

IV Combinaties

De meeste printercommando's kunnen vrij met elkaar gecombineerd worden. Er zijn echter enige beperkingen, die deels voortkomen uit de specifieke eigenschappen van de printer, in dit geval de STAR SG-10, deels uit het programma zelf:

- 1) Het commando 'Q' (Near Letter Quality) kan met niets anders gecombineerd worden, uitgezonderd onderstrepen.
- 2) Het is het beste om de commando's die de 'pitch' instellen ('Q', 'P', 'E' en 'C') eerst uit te schakelen voordat een ander commando uit deze set wordt geactiveerd. Deze 4 commando's zijn dus niet met elkaar te combineren. Bovendien zijn 'P', 'E' en 'C' niet te combineren met 'M' (emphasized), echter wel met 'D' (double-strike), hetgeen gelukkig vrijwel hetzelfde resultaat oplevert als de emphasized mode.
- 3) Normaliter zal in de pitches Proportional, Elite en Condensed de letterbreedte niet veranderen wanneer 'U', 'D' of 'I' wordt geactiveerd. In deze gevallen zal dus de tekst correct worden uitgevuld, zolang tenminste niet meerdere pitches door elkaar worden gebruikt. Wanneer echter de 'enlarged' mode wordt ingeschakeld (commando 'X'), wordt elk karakter in dubbele breedte geprint, waardoor het uitvullen de soep in zal draaien!

Vandaar de volgende aanbevelingen:

- A) Gebruik binnen een tekst slechts 1 pitch-instelling (bv. Pica, Elite of Condensed).
- B) Vermijd de 'enlarged' mode, of schakel anders het uitvullen uit (met '.n').

U Opbouw Commando-tabel

De tabel met printercommando's is als volgt opgebouwd (zie de DATA-regels 3570-3680):

'L', A1, A2, A3, B1, B2, B3

Hierbij staat 'L' voor de gebruikte letter, A1 t/m A3 geven de (decimale) controle-codes voor het inschakelen van het commando, en B1 t/m B3 de codes voor het uitschakelen.

De voor de printercommando's gekozen letters zijn vrij willekeurig, en kunnen dan ook zonder meer veranderd worden, echter met 1 uitzondering, nl. het onderstreep-commando 'U': dit wordt expliciet in de source gebruikt.

Wanneer voor een commando minder dan 3 controle-codes nodig zijn, worden ze tot 3 aangevuld met vooraafgaande nullen. Bv. het reset-commando 'R' bestaat uit slechts 2 codes: 'ESC' en '@' (decimaal 27 en 64), zodat de code-string moest worden aangevuld tot: "0,27,64". Mocht Uw printer voor een bepaald commando minder dan 3 controle-codes benodigen, volg dan bij het wijzigen van de tabel dezelfde procedure.

Wanneer Uw printer een bepaald printercommando helemaal niet kent (bv. 'Q', de Near Letter Quality mode), kunt U twee dingen doen: ofwel U verwijderd de betreffende DATA-regel uit het programma (pas dan ook de variabele "code'aantal" in regel 100 aan). Ofwel U zet allemaal nullen op de plaats van de betreffende controle-codes: zie bv. B1 t/m B3 in regel 3570.

```

10 PROGRAM WPR.6E-6F.V5
20
30 PRINT #21;FOR A=1 TO 2
40
500
60 .BA #6E00
70 .TA #9800
80
90:CODE'NAAM =TOP
100:CODE'AANTAL =12
110:NAAM'TABEL ==#6F80
120:CODE'TABEL ==#6FA0
130
140:EOT ==#04
150:CR ==#0D
160:SPATIE ==#20
170
180:PL ==#52
190:TEL'KAR ==#54
200:KAPAF'POS ==#55
210:JUSTIFY ==#56
220:CENTER'CB ==#57
230:TEMP'INDENT ==#5A
240:WORDCOUNT ==#5C
250:KARNUM'ACT ==#8B
260:INDENT ==#8D
270:KARNUM'COR ==#9B
280:KARNUM'TOT ==#9C
290:UNDERLINE'CB=#9D
300:TEMP1 ==#9E
310:TEMP2 ==#9F
320:XSAV ==#E4
330:YSAV ==#E5
340
350:OUTPUT'CONT ==#A8B5
360:MAAK'NUL ==#A96F
370:STOP'PRINTER=#AA0A
380:SFACEPRINT ==#AAD
390:KAR'ADRES ==#ABA1
400:NAAR'SCHERM ==#ABAE
410:SCHEM ==#ABD9
420:RDCHVEC'OLD ==#AC7F
430:OSRDCH ==#FE94
440:PR'DIR ==#FF08
450:OUT ==#FFF4
460
470 \READ-CHARACTER-ROUTINE
480
490:RDCHVEC'NEW
500 JSR RDCHVEC'OLD
510 PHP
520 CMP @#01
530 BNE NO'PR'COM
540 JSR RDCHVEC'OLD
550 CMP @#40
560 BCC NO'PR'COM
570 CMP @#7F
580 BCS NO'PR'COM
590 ORA @#80

```

```

600:NO'PR'COM
610 FLP
620 RTS
630
640:PRINT'OP'SCHERM
650 PHA
660 JSR KAR'ADRES
670 LDA @#00
680 STA TEMP2
690 PLA
700 BFL NORMAAL
710 CMP @#FF
720 BEQ NORMAAL
730 \INVERTEER KARAKTER
740 DEC TEMP2
750:NORMAAL
760 TYA
770 PHA
780 LDY @#00
790 JMP NAAR'SCHERM
800
810 \STUUR NAAR PRINTER
820:OUTPUT
830 JSR PROCODES
840 BFL OUTPUT'QUIT
850 \AFHANDELING UNDERLINE
860 CMP @#D5
870 BEQ SET'UNDERLINE
880 CMP @#F5
890 BNE OUTPUT'QUIT
900 CLC
910:SET'UNDERLINE
920 LDA @#00
930 ROR A
940:STORE
950 STA UNDERLINE'CB
960:OUTPUT'QUIT
970 RTS
980
990 \AFHANDELING
1000 \PRINTERCOMMANDO'S
1010
1020:DRUKAF
1030 JMP OUT
1040
1050:PROCODES
1060 ASL A
1070 STA TEMP1
1080 ROR A
1090 BFL DRUKAF
1100 PHA
1110 STX XSAV
1120 STY YSAV
1130 AND @#5F
1140 LDY @CODE'AANTAL-1
1150:ZOEK
1160 CMP NAAM'TABEL,Y
1170 BNE VERDER
1180 JSR STUUR
1190:VERDER

```

1200 DEY
1210 BPL ZOEK
1220 LDY YSAV
1230 LDX XSAV
1240 PLA
1250 RTS
1260
1270:STUUR
1280 TYA
1290 ASL A
1300 STA TEMP2
1310 ASL A
1320 ADC TEMP2
1330 STA TEMP2
1340 LDA @#03
1350 TAY
1360 BIT TEMP1
1370 BVS PAK
1380 ASL A
1390:PAK
1400 ADC TEMP2
1410 TAX
1420:LUS1
1430 LDA CODE'TABEL-1,X
1440 BNE OK
1450 DEX
1460 DEY
1470 BNE LUS1
1480 BEQ TERUG
1490:LUS2
1500 LDA CODE'TABEL-1,X
1510:OK
1520 JSR STUUR'CODE
1530 DEX
1540 DEY
1550 BNE LUS2
1560:TERUG
1570 RTS
1580
1590:STUUR'CODE
1600 PHA
1610 JMP PR'DIR
1620
1630 \NIEUWE ROUTINE
1640 \OUTPUT TO PRINTER
1650
1660:KLAAR
1670 JMP STOP'PRINTER
1680
1690:SCAN'TEXT
1700 STY TEMP1
1710 STY TEMP2
1720 \SET DEFAULT AANTAL
1730 \KARAKTERS OP DEZE REGEL
1740 LDA KARNUM'ACT
1750 STA KARNUM'COR
1760
1770 \LEES 1 PRINT-REGEL
1780 \TEL SPATIES
1790 \TEL PRINTERCOMMANDO'S

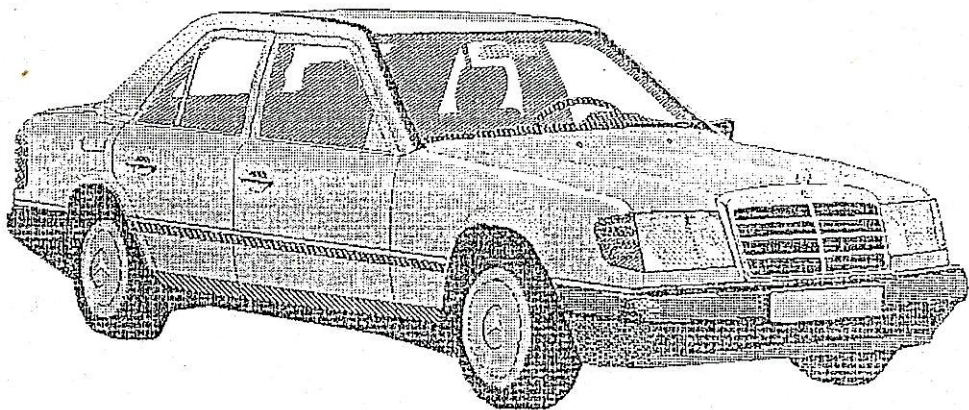
1800:LEES
1810 LDA (PL),Y
1820 BPL GEEN'PR'COM
1830 INC KARNUM'COR
1840 INC TEMP1
1850 BNE VOLGENDE
1860:GEEN'PR'COM
1870 CMP @CR
1880 BEQ CR'QUIT
1890 CMP @EOT
1900 BEQ KLAAR
1910 CMP @SPATIE
1920 BNE VOLGENDE
1930 LDA TEMP1
1940 STA TEMP2
1950 STY KAPAF'POS
1960 INC WORDCOUNT
1970:VOLGENDE
1980 INY
1990 CPY KARNUM'COR
2000 BNE LEES
2010
2020 \REGEL TE LANG
2030 \ZOEK PRINTERCOMMANDO'S
2040:PR'COM
2050 LDA (PL),Y
2060 BPL TE'LANG
2070 INY
2080 INC TEMP1
2090 BNE PR'COM
2100
2110 \ZOEK SPATIE OF <CR>
2120:TE'LANG
2130 CMP @SPATIE
2140 BEQ KAP'NIET'AF
2150 CMP @CR
2160 BEQ KAP'NIET'AF
2170
2180 \KAP AF ACHTER LAATSTE
2190 \SPATIE OF REGEL
2200 LDY KAPAF'POS
2210 DEC WORDCOUNT
2220 CLC
2230
2240:KAP'NIET'AF
2250 DEY
2260 \GEEF AAN DAT REGEL
2270 \TE LANG IS
2280 LDX @#00
2290 BCC KARNUM'DEF
2300
2310:CR'QUIT
2320 LDA TEMP1
2330 STA TEMP2
2340
2350:KARNUM'DEF
2360 INY
2370 \AANTAL KARAKTERS PLUS
2380 \PRINTERCOMMANDO'S
2390 STY KARNUM'TOT

```

2400 TYA
2410 SEC
2420 SBC TEMP2
2430 \AANTAL TE PRINTEN KAR'S
2440 STA TEL'KAR
2450
2460 \BEREKEN UITVUL-SPATIES
2470 TXA
2480 BNE UNDERLINE'1
2490 SEC
2500 LDA KARNUM'ACT
2510 SBC TEL'KAR
2520 STA JUSTIFY
2530
2540: UNDERLINE'1
2550 BIT UNDERLINE'CB
2560 BPL SPRING'IN
2570 \UNDERLINE OFF
2580 LDA @#F5
2590 JSR FRCODES
2600
2610: SPRING'IN
2620 LDA TEMP'INDENT
2630 BEG CENTER
2640 CMP @#FF
2650 BNE SPACES
2660 LDA INDENT
2670: SPACES
2680 JSR SPACEPRINT
2690
2700: CENTER
2710 LDA CENTER'CB
2720 BPL UNDERLINE'2
2730 SEC
2740 LDY KARNUM'ACT
2750 INY
2760 TYA
2770 SEC TEL'KAR
2780 JSR SPACEPRINT+1
2790
2800: UNDERLINE'2
2810 BIT UNDERLINE'CB
2820 BPL CONTINUE
2830 \UNDERLINE ON
2840 LDA @#D5
2850 JSR FRCODES
2860
2870: CONTINUE
2880 JMP OUTPUT'CONT
2890
2900 \UITBREIDING .1 COMMANDO
2910: STORE'ZERO
2920 JSR MAAK'NUL
2930 \INIT UNDERLINE'CB
2940 JMP STORE
2950
2960J
2970
2980 C=OUTPUT
2990 D=STUUR'CODE
3000 I=TEMP
3010 K=KARNUM'TOT
3020 N=NAAR'SCHERM
3030 O=OSRDCH
3040 R=RDCHVEC'NEW
3050 S=PRINT'OP'SCHERM
3060 T=SCAN'TEXT
3070 Z=STORE'ZERO
3080
3090
3100 REM VERANDER WPR IN RAM
3110 ?#BFFF=0
3120
3130 REM AANPASSING ROUTINE
3140 REM PRINT'OP'SCHERM
3150
3160 P=N-1
3170
3180I
3190 RTS
3200 CLC
3210 ADC #94
3220 STA #96
3230 TYA
3240 ADC #95
3250 STA #97
3260 LDA (#98),Y
3270 EOR I
3280J
3290
3300 P=#A831;C;JMP T;J
3310 P=#A8CB;C;JSR C;INY;CPY K;J
3320 P=#A930;C;JMP D;J
3330 P=#A99C;C;JSR Z;J
3340 P=#ABF2;C;JSR S;J
3350 P=#AC11;C;JSR S;J
3360
3370 ?#AAD9=#07;?#ABCD=#EA
3380 ?#AC8B=#AE;?#AC96=#AE;?#ACA4=#AE
3390 ?#ACED=R;?#ACEE=R/256
3400 ?#ACF1=0;?#ACF2=0/256
3410
3420 REM TERUG NAAR P-CHARME
3430 ?#BFFF=5
3440
3450 PRINT #6;NEXT A
3460
3470 RESTORE
3480 FOR I=0 TO CODE'AANTAL-1
3490 READ #CODE'NAAM
3500 ?(NAAM'TABEL+I)=?CODE'NAAM
3510 FOR J=0 TO 5
3520 READ K
3530 ?(CODE'TABEL+I*6+5-J)=K
3540 NEXT J
3550 NEXT I
3560
3570 DATA "R",0,27,CH"@",0,0,0
3580 DATA "Q",0,27,CH"4",0,27,CH"5"
3590 DATA "I",27,CH"I",1,27,CH"I",0

```

3600 DATA "U", 27, CH"-", 1, 27, CH"-", 0
3610 DATA "^", 27, CH"S", 0, 0, 27, CH"T"
3620 DATA "S", 27, CH"S", 1, 0, 27, CH"T"
3630 DATA "E", 0, 27, CH"M", 0, 0, 18
3640 DATA "C", 0, 0, 15, 0, 0, 18
3650 DATA "F", 27, CH"p", 1, 27, CH"p", 0
3660 DATA "X", 27, CH"W", 1, 27, CH"W", 0
3670 DATA "D", 0, 27, CH"G", 0, 27, CH"H"
3680 DATA "M", 0, 27, CH"E", 0, 27, CH"F"
3690
3700 END



Bootstrap

I Inleiding

Het programma "Bootstrap.V3" werd ontwikkeld voor de bootstrap-print van Charl de Moor (zie A.N. 3.7). Het is met name bedoeld voor mensen die een 16K kaart, de VIA 6522 en (eventueel) een schakelkaart hebben. Er zitten geen disk-drive voorzieningen in. Deze zijn echter vrij gemakkelijk in te bouwen, bv. de 'ready-test' van Frans van Hoesel (zie A.N. 4.4, p.32).

De bootstrap-print van Charl de Moor kan de laatste 8 bytes van het F-blok (#FFF8-#FFFF) verplaatsen naar een ander 4K-blok, waarna het sprong-adres van de reset-vector veranderd kan worden t.b.v. een bootstrap-routine. Bij mijn computer staan deze 8 bytes momenteel op het E-blok: zie de regels 3550 en 3560.

II Inhoud

In het programma zit een password-routine, een RAM-test, een ROM-test en een BREAK-routine ingebouwd. Deze 4 onderdelen zullen nu achtereenvolgens worden besproken.

A) Password-routine

Bij het inschakelen van de computer wordt normaal een reset-puls opgewekt, waardoor de processor springt naar de RESET-vector (#FFFC en #FFFD) die wijst naar adres #FF3F: "ACORN ATOM" verschijnt dan op het scherm. Met de bootstrap ingebouwd wijst de RESET-vector echter naar het begin van de bootstrap-routine (onder battery-backup in RAM): bij mijn computer staat deze vanaf #7B00 in het geheugen (zie regel 30). Kies eventueel een ander geheugeengebied. De machinecode beslaat ongeveer 3 pagina's.

Eerst wordt getest of het shift-register van de VIA de waarde 0 bevat: dit zal namelijk normaal alleen het geval zijn wanneer de computer wordt ingeschakeld! Van dit gegeven nu maakt de bootstrap-routine gebruik: na het doorlopen van de password-routine zal de waarde van dit register veranderd worden, zodat de password-routine (en de bijbehorende RAM- en ROM-tests) slechts 1 keer zal worden doorlopen, namelijk onmiddellijk na het aanzetten van de computer. Als conditie geldt echter dat dit register niet voor andere doeleinden mag worden gebruikt, bv. als schaduwbyte in een SOS. Mocht dit wel het geval zijn, kies dan een ander controle-byte, bv. #FE met als normale inhoud #0A.

Bevat het shift-register niet de waarde 0, dan wordt doorgesprongen naar de BREAK-routine (zie aldaar). Anders begint nu de eigenlijke password-routine. Op het scherm verschijnt de volgende tekst:

```
ACORN SYSTEM
```

```
ENTER PASSWORD
```

Het password (wachtwoord) dient hierna te worden ingetoetst, gevolgd door <CR>. Op het scherm verschijnt echter niets! Pas als het juiste wachtwoord is gegeven zal de routine verdergaan... Dit biedt een zeer goede beveiliging van het systeem, want hier is maar op 2 manieren doorheen te komen: ofwel je kent het wachtwoord, ofwel je maakt de computer open en haalt de bootstrap-print eruit! Het wachtwoord (zie regel 3540) mag max. uit 255 karakters bestaan, waarbij onder 'karakter' wordt verstaan: elke vanaf het toetsenbord bereikbare ASCII-waarde! In het geheugen is voor de opslag van het password een gebied van 64 bytes gereserveerd.

Wanneer het password wordt geaccepteerd, zal de melding "Welcome to the machine!" op het scherm verschijnen en start de "system check", bestaande uit achtereenvolgens een RAM- en een ROM-test.

B) RAM-test

Deze test maakt gebruik van de random-generator (zie ATP, p.92) en test die geheugengebieden die niet onder battery-backup staan: bij mijn computer zijn dit de gebieden #400-#4000 en #8200-#A000. Pas de betreffende waarden eventueel aan in de regels 1150 t/m 1190. Omdat de test vrij veel tijd kost, wordt hij maar 1 keer uitgevoerd. Wanneer ergens een fout optreedt, wordt het betreffende adres op het scherm afgedrukt. De test wordt echter niet afgebroken!

C) ROM-test

Dit is de inmiddels overbekende Cyclic Redundancy Check (zie bv. ATP, pp.92-93). Hierbij wordt telkens van een 4K-blok de signatuur berekend en vergeleken met de in een tabel opgeslagen waarde. Wanneer beiden overeenkomen verschijnt "OK" op het scherm, anders klinkt er een piepje en wordt de melding "CRC FAULT!" geprint. Ook hier zal de test niet worden afgebroken.

De tabel (zie de regels 3450 t/m 3520) bevat voor elk 4K-blok achtereenvolgens het 'high byte' van het startadres (bv. #C0), het boxnummer (dit is #FF indien het geen box op de schakelkaart betreft) en de signatuur.

Enige kanttekeningen hierbij:

- De signatuur voor de Gagsrom (regel 3490) betreft een gewijzigde versie: ten eerste is het 'PAINI'-statement veranderd (zie 'De Stacker' van februari 1985, p.6), en ten tweede wordt de commando-tabel niet meer beëindigd met #C558 maar met #C278 (adressen #AF01 en #AF02), zulks om stack-fouten te voorkomen.
De oorspronkelijke signatuur is: #4198.
- De signatuur van het D-blok (regel 3510) heeft betrekking op een FP-ROM met 3-weg-schakelaar die naar #E000 wijst.
De oorspronkelijke signatuur is: #AAA1.
- Verder klopt de signatuur van het F-blok (regel 3520) alleen bij de originele versie, met als RESET-vector #FF3F.
Bereken zelf de nieuwe signatuur, die zal afhangen van het gekozen startadres voor de bootstrap-routine.
Bij het hier gekozen startadres (#7B00) wordt deze: #89C4.

Tenslotte: het einde van de tabel wordt aangegeven door een byte met waarde 0.

Na de ROM-test zal de melding "system ready" verschijnen, en wordt gesprongen naar een afsluitende routine 'init' (adres staat op regel 370) die eenieder, afhankelijk van systeem en wensen, zelf kan invullen: bij mij bijvoorbeeld wordt dan P-Charme ingeschakeld, diverse pointers geïnitieerd, het COS op 1200 Baud gezet en het voorheen geteste RAM-geheugen schoongeveegd, waarna de routine eindigt met: 'JMP #C2B2'.

D) BREAK-routine

Deze is grotendeels overgenomen van Gerrit Bronsveld uit de Regio Noord: met dank.

Het werkt als volgt: wanneer na een 'reset' (de BREAK-toets) het shift-register van de VIA niet de waarde 0 blijkt te bevatten, wordt er geen koude maar een warme start gegeven, en het programma springt naar deze routine.

Er zijn dan 2 mogelijkheden:

- a) Tegelijk met de BREAK-toets was ook de CTRL-toets ingedrukt. In dat geval wordt een normale 'reset' gegenereerd: "ACORN ATOM".
- b) De CTRL-toets was niet ingedrukt: het indrukken van de BREAK-toets veroorzaakt dan alleen een onderbreking van het op dat moment draaiende programma, waarna de melding "BREAK AT LINE ..." verschijnt. Was er op dat moment geen programma actief, dan wordt alleen "BREAK" geprint. In beide gevallen worden er echter geen vectoren, arrays of labels 'gereset'!

Met andere woorden: voortaan zal het (per ongeluk) indrukken van de BREAK-toets geen fatale gevolgen meer hebben!

III Tot slot

Aan dit programma kan natuurlijk nog veel veranderd, verbeterd of toegevoegd worden. Om maar wat te noemen: codering van het wachtwoord, automatische opstart van de disk-drive, inschakeling van screenrom of 2440-VDU, enz. enz. Het geeft mijns inziens echter reeds een goede indruk van wat je zoal met een bootstrap-printje kunt doen: beschouw het maar als een eerste aanzet... tot volledige automatisering!

```

10 PROGRAM BOOTSTRAP.V3
20
30 S=#7B00;REM STARTADRES
40
50 P.#21;FOR A=1 TO 2
60
70[
80 .BA S
90 .TA #9B00
100
110:CR                =#0D
120:HASH              =#23
130
140:BASE              =#70
150:END               =#72
160:TELLER           =#73
170:CRC1              =#74
180:CRC2              =#75
190:XSAV              =#76
200:YSAV              =#77
210:CURSOR'MODE     =#E1
220
230:PASSWORD         =S+#280
240:TABEL             =S+#2C0
250
260:POORT'A          =#B000
270:POORT'B          =#B001
280:POORT'C          =#B002
290:POORT'CONTROL   =#B003
300:SHIFT'REG       =#B80A
310:SWITCH           =#BFFF
320
330:ESCAPE           =#C2CF
340:PR'DEC           =#C589
350:CR'LF            =#CD54
360
370:INIT             =#E6D0
380
390:PR'TEXT          =#F7D1
400:PR'HEX'SP        =#F7FA
410:PR'HEX           =#F802
420:WACHT            =#FB83
430:RESET            =#FF3F
440:OSRDCH           =#FFE3
450:OSWRCH           =#FFF4
460
470 \*****
480
490:BEGIN
500 LDA SHIFT'REG
510 BEQ STARTUP
520 JMP NO'STARTUP
530
540:STARTUP
550[
560 COPY #FF3F,#FF74,P
570 P=P+(#FF74-#FF3F)+1
580[
590 .AS "SYSTEM"

```

```

600 .DB #0A0A
610 .BY CR
620 LDA @#00
630 STA CURSOR'MODE
640 JSR PR'TEXT
650 .AS "ENTER PASSWORD"
660
670:ENTER
680 LDY @#FF
690:PASS
700 JSR OSRDCH
710 INY
720 CMP PASSWORD,Y
730 BNE ENTER
740 CMP @CR
750 BNE PASS
760
770 STA SHIFT'REG
780
790 JSR PR'TEXT
800 .BY #0C
810 .DB #0707
820 .AS "WELCOME TO THE MACHINE!"
830 .DB #0A0A
840 .BY CR
850 .AS "SYSTEM CHECK"
860 .DB #0A0A
870 .BY CR
880 .AS "START RAMTEST"
890 .DB #0A0D
900 NOP
910 JSR RAMTEST
920 JSR PR'TEXT
930 .AS "RAMTEST COMPLETED"
940 .DB #0A0A
950 .BY CR
960 .AS "START ROMTEST"
970 .DB #0A0D
980 NOP
990 JSR ROMS
1000 JSR PR'TEXT
1010 .BY CR
1020 .AS "ROMTEST COMPLETED"
1030 .DB #0A0A
1040 .BY CR
1050 .DB #0707
1060 .AS "SYSTEM READY"
1070 .DB #0A0A
1080 .BY #0D
1090 NOP
1100 JMP INIT
1110
1120 \*****
1130
1140:RAMTEST
1150 LDA @#04
1160 LDX @#40
1170 JSR TEST
1180 LDA @#82
1190 LDX @#A0

```

1200
1210:TEST
1220 STA BASE+1
1230 STX END
1240 LDY @#00
1250 STY BASE
1260 STY #0C
1270
1280:LOOP
1290 JSR RANDOM
1300 JSR MEMORY
1310 JSR UPDATE
1320 BCC LOOP
1330 RTS
1340
1350\
1360
1370:RANDOM
1380 LDY @#20
1390
1400:ROTATE
1410 LDA #0A
1420 LSR A
1430 LSR A
1440 LSR A
1450 EOR #0C
1460 ROR A
1470 ROL #0B
1480 ROL #09
1490 ROL #0A
1500 ROL #0B
1510 ROL #0C
1520 DEY
1530 BNE ROTATE
1540 RTS
1550
1560\
1570
1580:MEMORY
1590 LDY @#00
1600 LDX @#FC
1610
1620:MEM
1630 LDA #0C,X
1640 STA (BASE),Y
1650 CMP (BASE),Y
1660 BEQ NEXT
1670 JSR FAULT
1680:NEXT
1690 INY
1700 INX
1710 BNE MEM
1720 RTS
1730
1740\
1750
1760:FAULT
1770 STY YSAV
1780 JSR PR'TEXT
1790 .AS "FAULT AT #"

1800 LDY YSAV
1810 LDA BASE+1
1820 JSR PR'HEX
1830 TYA
1840 CLC
1850 ADC BASE
1860 JSR PR'HEX
1870 JMP CR'LF
1880
1890\
1900
1910:UPDATE
1920 CLC
1930 LDA BASE
1940 ADC @#04
1950 STA BASE
1960 LDA BASE+1
1970 ADC @#00
1980 STA BASE+1
1990 CMP END
2000 RTS
2010
2020 *****
2030
2040:ROMTEST
2050 LDA @#10
2060 STA TELLER
2070 LDY @#00
2080 STY BASE
2090 STY CRC1
2100 STY CRC2
2110
2120:BYTE
2130 LDA (BASE),Y
2140 LDX @#0B
2150
2160:BIT
2170 LSR A
2180 ROL CRC1
2190 ROL CRC2
2200 BCC GO'ON
2210
2220 PHA
2230 LDA CRC1
2240 EOR @#2D
2250 STA CRC1
2260 PLA
2270
2280:GO'ON
2290 DEX
2300 BNE BIT
2310 INY
2320 BNE BYTE
2330 INC BASE+1
2340 DEC TELLER
2350 BNE BYTE
2360 RTS
2370
2380 *****
2390

2400:ROMS	3000 LDX @#02
2410 LDX @#00	3010 JSR WACHT
2420	3020 LDX @#8A
2430:ROM	3030 STX FOORT' CONTROL
2440 LDA TABEL, X	3040 LDX @#7
2450 BEQ KLAAR	3050 STX FOORT' C
2460 PHA	3060 AND @#F0
2470 LDA @HASH	3070 STA FOORT'A
2480 JSR DSWRCH	3080 JSR PR' TEXT
2490 PLA	3090 .DB #060F
2500 STA BASE+1	3100 LDA @#80
2510 JSR PR'HEX' SP	3110 STA CURSOR' MODE
2520 INX	3120 JSR PR' TEXT
2530 LDA TABEL, X	3130 .DB #0A0A
2540 BMI NO' SWITCH	3140 .BY CR
2550 STA SWITCH	3150 .AS "BREAK"
2560 PHA	3160 LDA #01
2570 JSR PR' TEXT	3170 ORA #02
2580 .AS "BOX "	3180 BEQ NO' RUN
2590 NOP	3190 JSR PR' TEXT
2600 PLA	3200 .AS " AT LINE "
2610 JSR PR'HEX' SP	3210 LDA #01
2620:NO' SWITCH	3220 STA #16
2630 STX XSAV	3230 LDA #02
2640 JSR FOMTEST	3240 STA #25
2650 LDX XSAV	3250 LDA @#00
2660 INX	3260 STA #34
2670 LDA TABEL, X	3270 STA #43.
2680 INX	3280 JSR PR' DEC
2690 CMP CRC1	3290:NO' RUN
2700 BNE WRONG	3300 JSR CR' LF
2710 LDA TABEL, X	3310 JSR CR' LF
2720 CMP CRC2	3320 JMP ESCAPE
2730 BNE WRONG	3330
2740 JSR PR' TEXT	3340]
2750 .AS "OK"	3350
2760 NOP	3360 P.#6;NEXT A
2770:VOLGENDE	3370
2780 JSR CR' LF	3380 RESTORE
2790 INX	3390 FOR I=0 TO 7#4 STEP 4
2800 BNE ROM	3400 READ A,B,C
2810:KLAAR	3410 !(TABEL+I)=A+B#256
2820 RTS	3420 !(TABEL+I+2)=C
2830	3430 NEXT I
2840:WRONG	3440
2850 JSR PR' TEXT	3450 DATA #A0,#02,#660E;REM PP TOOLBOX
2860 .BY #07	3460 DATA #A0,#03,#B326;REM WORD PACK ROM
2870 .AS "CRC FAULT!"	3470 DATA #A0,#04,#CC6B;REM JOSBOX
2880 NOP	3480 DATA #A0,#05,#C16B;REM P-CHARME V.173
2890 BMI VOLGENDE	3490 DATA #A0,#06,#DFE1;REM GAGS V2.0E
2900	3500 DATA #C0,#FF,#D67D
2910 *****	3510 DATA #D0,#FF,#F261
2920	3520 DATA #F0,#FF,#E3B6
2930:NO' STARTUP	3530
2940 BIT FOORT' B	3540 \$PASSWORD="TEST"
2950 BVS BREAK	3550 ?#EFFC=BEGIN%256
2960 JMP RESET	3560 ?#EFFD=BEGIN/256
2970	3570
2980:BREAK	3580 END
2990 LDA FOORT'A	

I Inleiding

Het programma "Schakel.02" vormt een aanvulling op de Monitor van Roel Keuvel (zie A.N. 1.7, pp.38-40 en A.N. 1.8, pp.58-59), een 1K programma gebaseerd op de 2K Apple monitor. De uitbreiding beslaat 88 bytes machinecode.

Velen zullen zich waarschijnlijk afvragen: waarom deze uitbreiding? Waarom überhaupt nog aandacht voor deze, ietwat verouderde, monitor annex disassembler/tracer, nu we over zoveel 'betere' debuggings-utilities beschikken, zoals de Josbox met zijn statements KDUMP, DISAS en STEP, de 6502 tracer van Bas Kasteel (zie A.N. 3.6, pp.54-55) en niet te vergeten de Program Flow Controller van Theo den Exter (zie A.N. 4.1, pp.65-75)?

Welnu, al deze utilities hebben hun specifieke voor- en nadelen: de STEP-routine uit de Josbox bijvoorbeeld, die op interruptbasis werkt, heeft als voordelen de zeer compacte weergave op het scherm en de mogelijkheid om langdradige edoch noodzakelijke initialisatie-routines snel te doorlopen. Nadelen zijn echter: het feit dat de routine op het A-blok staat, waardoor geen andere boxen geTRACEd kunnen worden, en de onmogelijkheid om er tussentijds even uit te stappen om bv. de inhoud van bepaalde geheugenlocaties te lezen: zolang de stack-pointer op #FF staat is er niets aan de hand, maar owee als je eruitstapt tijdens uitvoering van een subroutine!

De 6502 tracer van Bas Kasteel heeft eveneens zijn beperkingen: ten eerste vindt er tijdens het traceren geen disassembly plaats, waardoor een en ander zeer duister wordt, en ten tweede kunnen de registers niet tussentijds veranderd worden.

De Program Flow Controller (PFC) van Theo den Exter is ongetwijfeld de meest veelzijdige en ook de machtigste van dit drietal, zo machtig zelfs dat Theo ons afraadt om zijn debugger als monitor te 'misbruiken'... Goed, in elk geval kleeft aan zijn programma een nadeel dat ook de andere utilities kenmerkt: het gebruikt geheugen, en met name 2P-adressen... die natuurlijk net ook weer gebruikt worden door de machinetaal-routine die je wilde debuggen! Gevolg van dit "verboden access op het werkgeheugen van de debugger" (A.N. 4.1, p.73): de PFC laat het afweten.

Ook de Monitor van Roel Heuvel laat het op dit punt afweten. Het programma heeft echter een aantal duidelijke voordelen:

- zijn compactheid: slechts 1K groot!
- zijn makkelijke hanteerbaarheid en grote flexibiliteit door de relatief eenvoudige opzet: werkt niet op interruptbasis en kan op elk moment worden onderbroken om bv. registers bij te werken of de executie elders voort te zetten.
- zijn toch vrij grote veelzijdigheid: het kan een hexdump geven, disassembleren, STEPpen, TRACEn, gegevens verplaatsen en verifiëren, machinetaal-routines uitvoeren, hexadecimaal optellen en aftrekken, en dat alles met zeer eenvoudige commando's!

Kortom, een goed stuk gereedschap, ware het niet dat... Juist: "verboden access", in dit geval alleen in Zero-Page (bij de PFC bv. ook op pagina #28, hetgeen in bepaalde gevallen zeer hinderlijk is).

Om nu dit laatste bezwaar tegen de Monitor weg te nemen, werd de aanvulling "Schakel.U2" geschreven.

II Herking

Wat doet deze aanvulling?

- Zodra er een commando STEP ('S') of TRACE ('T') wordt gegeven, wordt, zowel voor als na het uitvoeren van een instructie, de *hele Zero-Page verwisseld* met de inhoud van een buffer-geheugen ter grootte van 256 bytes! Hierdoor wordt het gevaar van "shared facilities", zoals Theo den Exter ze noemt, grotendeels vermeden.
- Bovendien kan nu voor het buffer-geheugen een veel gunstiger gebied worden uitgezocht dan pagina #28. In "Schakel.U2" bv. is gekozen voor pagina #67 op de 16K kaart (zie de variabele "ZPSAV" in regel 190), maar dit mag natuurlijk ook een andere pagina zijn, bv. #3F of #9F.
- Verder zijn in het TRACE commando enige kleine veranderingen aangebracht: ten eerste worden de opeenvolgende instructies nu door een lege regel van elkaar gescheiden, wat de leesbaarheid ten goede komt, en ten tweede wordt de TRACE-mode voortaan niet meer gestopt door de ESC-toets (wat soms vervelende gevolgen had), maar door de CTRL-toets.

Ondanks de hierboven geschetste aanvulling is de Monitor echter niet volmaakt: zo heeft het programma nog steeds een stukje geheugen nodig voor de eigenlijke uitvoering van de instructie. Hier is gekozen voor het gebied #222-#22A (zie de variabele "STAP2" in regel 170), wat echter bv. bij het traceren van een programma als CK.SOS weer problemen geeft. Een beter alternatief is wellicht een stukje van de input-buffer, bv. #130-#138.

III Het programma

- * Eerst enige opmerkingen met betrekking tot de adres-variabelen in de regels 50 t/m 220:
 - r. 50 : hier staat het startadres voor de aanvullende machinecode. Als U de Monitor vast in het geheugen hebt staan dient U hiervoor een vrij stuk geheugen (88 bytes) onder battery-backup te kiezen.
 - r. 80 : dit is het masker van poort B van de 8255 (#B001), dat bepaalt door welke toets de TRACE-mode wordt gestopt: voor de ESC-toets is dit #20 en voor de SHIFT-toetsen #80.
 - r.100-160: de hier genoemde adressen zijn niet variabel, en mogen dus niet veranderd worden!
 - r.210 : de variabele "BEGIN" bevat het beginadres van de Monitor. Deze variabele dient U aan te passen aan de plaats waar de Monitor in het geheugen van uw computer staat geassembleerd, bv. vanaf #7C00.

- * In de regels 620 t/m 670 wordt de waarde van de stack-pointer gecorrigeerd, zodat bij aanroep van de Monitor een 'lege' stack-pointer wordt gesimuleerd. De stack-pointer kan overigens niet veranderd worden! Bovendien zal (in STEP- of TRACE-mode) bij de uitvoering van een subroutine de stack-pointer vertraagd worden bijgewerkt, hetgeen een enigszins vertekend beeld geeft. Hierover hoeft U zich echter niet ongerust te maken.

- * Een laatste opmerking betreft de regels 1220 t/m 1270: deze machinecode, die zorg draagt voor correcte uitvoering van een indirecte sprong via een ZP-pointer, overschrijft de oude Monitor-routine voor het opslaan van de registers, die immers is vervangen door een nieuwe routine: regels 550 t/m 810.

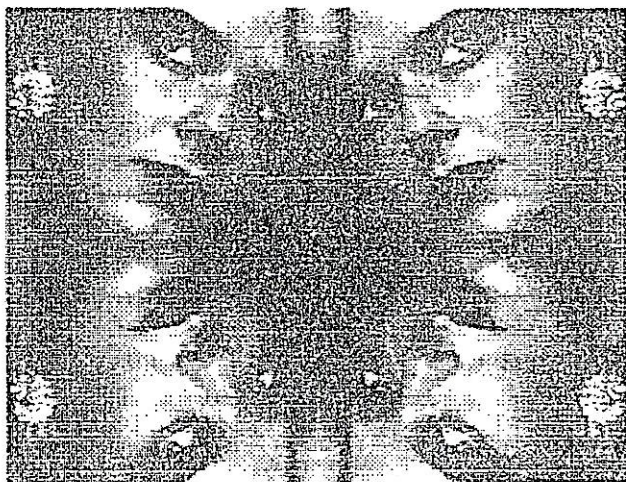
IV Tot slot

Zoals reeds gezegd, ook met deze aanvulling is de Monitor van Roel Heuvel niet volmaakt. In bepaalde debuggings-operaties zullen dan ook andere utilities, met name de PFC, de voorkeur verdienen. Niettemin heeft de Monitor zijn aantrekkelijke kanten, in het bijzonder zijn compactheid en zijn relatieve eenvoud. De uiteindelijke keus is, zoals altijd, aan U!

10 PROGRAM SCHAKEL.V2
20
30 P.#21;FOR A=1 TO 2
40
50 .BA #EE00
60 .TA #8200
70
80:CTRL=#40
90
100:UCCA=#9C
110:XREG=#9D
120:YREG=#9E
130:STEK=#9F
140:VLAG=#A0
150
160:STAP1 =#0092
170:STAP2 =#0222
180:ZP =#0000
190:ZPSAV =#6700
200:FOORT'B=#B001
210:BEGIN =#E800
220:CR'LF =#FFED
230
240\
250
260:HAAL
270 LDX @B
280:VERFL
290 LDA STAP1,X
300 STA STAP2,X
310 DEX
320 BFL VERFL
330
340 LDA VLAG
350 PHA
360 LDA UCCA
370 PHA
380 LDA XREG
390 PHA
400 LDA YREG
410 PHA
420
430 JSR SWITCH
440
450 FLA
460 TAY
470 PLA
480 TAX
490 PLA
500 PLS
510 RTS
520
530\
540
550:SLA'OP
560 PHP
570 PHA
580 TXA
590 PHA

600 TYA
610 PHA
620 TSX
630 TXA
640 CLD
650 CLC
660 ADC @B
670 PHA
680
690 JSR SWITCH
700
710 FLA
720 STA STEK
730 PLA
740 STA YREG
750 FLA
760 STA XREG
770 PLA
780 STA UCCA
790 PLA
800 STA VLAG
810 RTS
820
830\
840
850:SWITCH
860 LDX @B
870:SWAP
880 LDY ZP,X
890 LDA ZPSAV,X
900 STA ZP,X
910 TYA
920 STA ZPSAV,X
930 INX
940 BNE SWAP
950 RTS
960
970\
980
990:TRACE'CR
1000 JSR CR'LF
1010 LDA FOORT'B
1020 RTS
1030
1040\VERANDER
1050
1060 .BA BEGIN+#177
1070 JSR HAAL
1080 JMP STAP2
1090
1100 .BA BEGIN+#1AD
1110 BCS INDIRECT
1120
1130 .BA BEGIN+#201
1140 JSR SLA'OP
1150
1160 .BA BEGIN+#20C
1170 JMP WISSEL
1180:WISSEL
1190 JSR SWITCH

1200 JMP BEGIN+##1F3
1210
1220: INDIRECT
1230 TXA
1240 BNE BEGIN+##1A2
1250 LDA @ZPSAV/256
1260 STA #91
1270 BNE BEGIN+##1A2
1280
1290 .BA BEGIN+##232
1300 JSR TRACE'CR
1310 AND @CTRL
1320
1330: P.#6; NEXT A
1340
1350 END



```

10 REM RELOC.NORM
20 REM AANROEF: RELOC "VAN", "TOT", "NAAR"
30 DIM LL21
40 F.I=0T021;LLI=#FFF;N.
50 P.#21;P=A;GOSUB a
60 P.#06;P=A;GOSUB a
70 #T="RELOC";T=T+LENT
80 ?T=LL7/256;#80;T?1=LL7%256;T?2=#80;T=T+2
90 A=P;T!1=A
100 END
110aLL16=#C231;LL17=#C3CB
120LL18=#C4E4;LL19=#C55B;LL20=#C78B;LL21=#CDF6
130C
140:LL7LDX@#00;STX#04;JSRLL20
150LDA#34;ORA#43;BNELL7+1
160LDA#16;STA#72;LDA#25;STA#73;JSRLL16;JSRLL20
170LDA#35;ORA#44;BNELL7+1;JSRLL2;BCCLL7+1;JSRLL16;JSRLL20
180LDA#36;ORA#45;BNELL7+1
190JSRLL17;JSRLL18;DEY
200:LL2LDA(#72),Y;JSRLL3;STA#71;TAY;CMP#02;BEQLL14
210:LL9LDA(#72),Y;STA(#52),Y;CMP(#52),Y;BEQLL11
220:LL10JMPFL121
230:LL11DEY;BFLLL9;JSRLL0;BCCLL13
240LDA#52;ADC#71;STA#52;BCCLL12;INC#53
250:LL12INY;BEQLL8
260:LL13LDX@#00;STX#04;JMPFL19
270\
280:LL14LDA#17;DEY;SBC(#72),Y;LDA#26;INY
290SBC(#72),Y;BCCLL9;DEY;LDA(#72),Y;SBC#16
300STA#54;INY;LDA(#72),Y;SBC#25;BCCLL9
310STA#55;CLC;LDA#18;ADC#54;STA#56
320LDA#27;ADC#55
330:LL15STA(#52),Y;CMP(#52),Y;BNELL10;DEY;BEQLL9
340LDA#56;BCSLL15
350\
360:LL0LDA#72;ADC#71;STA#72;BCCLL1;INC#73
370:LL1SEC
380:LL2LDA#17;SBC#72;LDA#26;SBC#73;RTS
390\
400:LL3TAY;AND#03;TAX;TYA;LSRA;LSRA;TAY;LDALL6,Y
410:LL4DEX;BMILL5;LSRA;LSRA;BFLLL4
420:LL5AND#03;RTS
430:LL6;J
440P!00=#28041404;P!04=#28081405
450P!08=#2A041506;P!12=#28081405
460P!16=#2A041404;P!20=#28081405
470P!24=#2A041404;P!28=#28081405
480P!32=#2A001504;P!36=#08081505
490P!40=#2A041515;P!44=#2A081505
500P!48=#2A041505;P!52=#28081405
510P!56=#2A041505;P!60=#28081405
520P=P+64;RETURN
530
540 Deze versie is met name bedoeld voor hen die
550 in het bezit zijn van een schakelkaart (#AXXX)
560 maar niet over de Josbox beschikken.
570 Indien men ook andere boxen dan P-Charme vanaf #AXXX
580 wil kunnen reloceren, dan dient deze routine in het SOS
590 (bv. op #E000 of op #1000) te worden opgenomen.
600 Alle invoer-gegevens worden gecontroleerd.
610 De totale lengte bedraagt: 265 bytes (incl. tabel).
620
630 Joop Engels, Nijmegen.
640 Versie 1.0, 9-2-1985.

```

```

10 REM RELOC.P-CH
20 REM AANROEP: RELOC "VAN", "TOT", "NAAR"
30 DIM KK13
40 F.I=0T013;KKI=#FFF;N.
50 P.#21;P=A;GOSUB a
60 P.#06;P=A;GOSUB a
70 $T="RELOC";T=T+LENT
80 ?T=KK0/256;#80;T?1=KK0%256;T?2=#80;T=T+2
90 A=P;T:1=A
100 END
110a;[
120:KK0JSR#A4C9;LDY#00
130:KK1LDA(#56),Y;JSRKK10;STA#73;TAY;CMP#02;BEQKK8
140:KK2LDA(#56),Y;STA(#52),Y;CMP(#52),Y;BEQKK4
150:KK3JMP#CDF6
160:KK4DEY;BFLKK2;LDA#56;ADC#73;STA#56;BCCKK5;INC#57
170:KK5SEC;LDA#17;SBC#56;LDA#26;SBC#57;BCCKK7
180LDA#52;ADC#73;STA#52;BCCKK6;INC#53
190:KK6INY;BEQKK1
200:KK7JMP#C55B
210\
220:KK8LDA#17;DEY;SEC(#56),Y;LDA#26;INY;SBC(#56),Y;BCCKK2
230DEY;LDA(#56),Y;SBC#16;STA#70;INY;LDA(#56),Y;SBC#25;BCCKK2
240STA#71;CLC;LDA#18;ADC#70;STA#72;LDA#27;ADC#71
250:KK9STA(#52),Y;CMP(#52),Y;BNEKK3;DEY;BEQKK2
260LDA#72;BCSKK9
270\
280:KK10TAY;AND#03;TAX;TYA;LSRA;LSRA;TAY;LDAKK13,Y
290:KK11DEX;BMIKK12;LSRA;LSRA;BPLKK11
300:KK12AND#03;RTS
310:KK13;]
320P!00=#29041404;P!04=#28081405
330P!08=#2A041504;P!12=#28081405
340P!16=#2A041404;P!20=#28081405
350P!24=#2A041404;P!28=#28081405
360P!32=#2A001504;P!36=#08081505
370P!40=#2A041515;P!44=#2A081505
380P!48=#2A041505;P!52=#28081405
390P!56=#2A041505;P!60=#28081405
400P=P+64;RETURN
410
420 Deze versie is bedoeld voor diegenen die alleen
430 in het bezit zijn van P-Charme (versie 173).
440 De invoer-gegevens worden niet gecontroleerd.
450 Mede daardoor is deze versie iets sneller
460 (en 60 bytes korter!) dan de algemene versie.
470 De totale lengte bedraagt: 205 bytes (incl. tabel).
480
490 Joop Engels, Nijmegen.
500 Versie 1.1, 9-2-1985.

```

10REM FCOS
20J=50;DIMLLJ;F.I=0TOJ;LLI=#FFF;N.;@=0
30P.#12" FCOS "
40IN."*"GEEF STARTADRES BV #3C00 "Z;IF718=Z/256;G.40
50LL1=#CA;LL2=#209;LL3=#20E;LL35=#8002;LL36=#C384
60LL37=#C55B;LL38=#F84F;LL39=#FAB6;LL40=#FB83;LL41=#FCBF
70LL42=#FC23;LL43=#FC40;LL45=#FCB0;LL46=#FCCD
80LL47=#FCD8;LL48=#FCDA;LL49=#FD48;LL50=#FFD1
90P.#21;F.I=1TO2;P=Z;IFI=2;F.#6
100C
110:LL0LDX@#05;LDY@#09
120:LL6LDALL44,X;STALL3,Y;DEX;DEY;BMILL7
130CPY@#05;BNELL6;LDY@#01;BNELL6
140:LL7JMPLL37
150:LL8STX#EC;STY#C3;PHF;SEI
160JSRLL16
170:LL10LDA@#7E;STA#CO
180:LL11JSRLL45;BCLL10;INC#CO;BFLLL11
190:LL12CLC
200:LL13PHF;JSRLL45;JSRLL41;CPX@#12;BCSLL14
210PLP;BCSLL15;SEC;BCSLL13
220:LL14;PLP
230:LL15ROR#CO;BCLL12
240LDA#CO;PLP;LDY#C3;LDX#EC;PHA
250CLC;ADC#DC;STA#DC;JSRLL16;PLA
260RTS
270:LL16PHF;PHA;TYA;PHA;LDALL2
280CMP@#FE;BNELL17;LDA#EA;BNELL17;LDY#EO
290BMILL17;LDA(#DE),Y;EOR@#EF;STA(#DE),Y
300:LL17PLA;TAY;PLA;PLP;RTS
310:LL18STX#EC;STY#C3;PHF;SEI;PHA
320JSRLL16;JSRLL42;STA#CO;JSRLL47;LDA@#0A
330STA#C1;CLC
340:LL19BCLL20;LDX@#07;STXLL35;LDX@#01;JSRLL48
350BMILL21
360:LL20LDA@#04;STALL35;JSRLL47;INCLL35;JSRLL47
370:LL21SEC;ROR#CO;DEC#C1;BNELL19;JSRLL16
380LDY#C3;LDX#EC;PLA;PLP;RTS
390:LL22STXLL35;LDX@#3C;JMPLL40
400:LL23PHF;SEI;JSRLL38;PHF;LDA@#06
410JSRLL43;LDX@#07;JSRLL22;PLP;BNELL24
420JMPLL39
430:LL24LDX@#04
440:LL25LDA#CE,X;STA#D2,X;DEX;BNELL25;STX#D0
450STX#D1;LDA#D5;BNELL26;DEC#D6
460:LL26DEC#D5;CLC
470:LL27ROR#D2;SEC;LDX@#FF;LDA#D5;SEC#D3
480STA#CF;LDA#D6;SEC#D4;PHF;ROR#D2
490PLP;BCLL28;CLC;BEQLL28;STX#CF
500SEC
510:LL28ROR#D2;INX;JSRLL29;INC#D0;INC#D4
520INC#C0;ROL#D2;BCSLL27;PLP;RTS
530:LL29LDX@#07;JSRLL22;STX#DC;LDY@#04
540:LL30LDA@#2A;JSRLL50;DEY;BNELL30
550:LL31LDA(#C9),Y;JSRLL50;INX;CMP@#0D;BNELL31
560LDY@#08
570:LL32LDALL1,Y;JSRLL50;DEY;BNELL32;LDX@#10
580JSRLL40;BIT#D2;BVCLL34;DEY
590:LL33INX;LDA(#D3),Y;JSRLL50;CPY#CF;BNELL33
600:LL34LDA#DC;JSRLL50;LDX@#04;STXLL35;RTS;LL44;J
610P!0=LL23;P!2=LL8;P!4=LL18
620N.;F."CODE VAN "&2" TOT "%F+6"#3
630END