

KIM GEBRUIKERS CLUB NEDERLAND

Zevende Jaargang Nr. 25

Februari 1983

De 6502 KENNER is een uitsave van de KIM Gebruikers Club Nederland.

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:

Willem L. van Pelt,
Jacob Jordaensstraat 15,
2923 CK KRIMPEN AAN DEN
IJSSSEL.

Redactie De 6502 KENNER:

Vaste medewerkers:

Willem van Pelt
Gerard van Roekel
Frans Smeehuijzen

Freelance medewerkers:

Adri Hankel
Renee de Hoop
Fridus Jonkman
Herman Kuyvenhoven
Anton Muller
Koen van Nieuwenhove
Jaap van Toledo

Gehele of gedeeltelijke overname van de inhoud van DE 6502 KENNER zonder toestemming van het bestuur is verboden. Toepassing van gepubliceerde programma's, hardware etc. is alleen toegestaan voor persoonlijk gebruik.

DE 6502 KENNER verschijnt 5 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1983 KIM Gebruikers Club Nederland.

De voorpagina is een aquarel van een KIM, geschilderd door:

Rinus Vleesch Dubois.

INHOUDSOPGAVE DE 6502 KENNER NR. 25 FEBRUARI 1983

1.	Van het bestuur ... door Rinus Vleesch Dubois	2.
2.	UITNODIGING Kim Club Bijeenkomst	3.
3.	Agenda	4.
4.	Vraag en Aanbod ... Koen van Nieuwenhove vraagt aandacht voor zijn floppy-systeem ... digitale cassette-recorder ... PC-100 met hard- en software	4.
5.	Van de redactie procedure opstarten Micro-Ade assembler ... door Willem van Pelt	5.
6.	Patches op KIM-1 BASIC t.b.v. JUNIOR (DEEL 4, slot) ... door Koen van Nieuwenhove + bugs in deel 1, 2 en 3	6.
7.	BASIC Program Compressor ... door Ruud Uphoff	10.
8.	Gebruik van de VIA 6522 ... door Willem Kuitens	22.
9.	JUNIOR en dubbeladressering VIA 6522 ... door Koen van Nieuwenhove	24.
10.	BUGS in Zenuwslag	26.
11.	BASIC Dataregels genereren ... door Maarten van Hintum	27.
12.	VIC modulebordje	29.
13.	KIM-1. Spiraal reactiespel voor hex-display ... door A.W. den Hartog	31.
14.	Elektuur Ram-Epromkaart 16K Eprom op een kaart ... door Hans Christen	34.
15.	AIM-65 BASIC voor de JUNIOR ... door E.J.M. Visschedijk	35.
16.	Service Rapporten Bestand praktische toepassing in BASIC (apart verkrijgbaar op cassette) ... door Maarten van Hintum	43.
17.	Marktinfo overzicht aanbiedingen PROTON speciaal voor clubleden	48.
18.	Ledenlijst	middenpagina.

Op 20 november 1982 ben ik officieel benoemd tot voorzitter van onze club. Ik ben direct geconfronteerd met enkele problemen die, naar ik aanneem, door Uw medewerking opgelost zullen worden. Bij deze doe ik een beroep op U om U kandidaat te stellen voor een functie in het bestuur.

We zijn vooral op zoek naar mensen met de benodigde 'kwaliteiten' op het gebied van soft- c.q. hardware. Bovendien stellen we het op prijs als men over organisatorische talenten beschikt.

Het afgelopen jaar is wederom een goed jaar voor de club geweest. Vooral de iustrumviering was een succes. De H.C.C.-dag verdient meer aandacht. We zullen ons dit jaar trachten beter voor te bereiden op deze gebeurtenis.

De redactie is ontzettend gelukkig met de vele copy die zij krijgt toegezonden. Bij deze wil ik al diegenen bedanken die voor de copy gezorgd hebben, en hopenlijk blijft U copy sturen. Doch hierbij zou ik wel een kritische noot willen plaatsen; namelijk dat 'veel' copy niet automatisch inhoudt dat het ook 'goed' is.

Vooraf 'commentaar' bij de programma's is een vereiste. Wanneer men publiceert, is het de bedoeling dat iedereen begrijpt wat U schrijft.

Het is ook nooit weg, als men d.m.v. een inleiding beschrijft wat het programma doet en waarvoor het bedoeld is.

Het jaar 1983 zal vooral gericht zijn op continuering en verbetering van de kwaliteit van DE 6502 KENNER. Met de huidige redactie onder de uitstekende leiding van Willem van Peit zal e.e.a. best lukken, indien U maar de hierboven bedoelde 'goede' copy blijft inzenden.

Tenslotte verzoek ik eenieder vooral naar de bijeenkomsten te komen. Het bestuur zal ook in het jaar 1983 haar uiterste best doen om die bijeenkomsten zo aantrekkelijk mogelijk te maken.

Ik wens U allen een goed 1983 toe.

R. Vleesch Dubois
voorzitter

MICRO - ADE ASSEMBLER, DIS- ASSEMBLER EN TEXT-EDITOR.

Sinds kort heeft onze club de copyrights verworven van de Micro-Ade Assembler, Disassembler en Texteditor van Peter Jennings. De club is het toegestaan, tot wederopzegging de MICRO-ADE aan haar leden ter beschikking te stellen. De MICRO-ADE kan worden besteld bij de redaktiesekretaris, onder opgave van wat u nodig hebt.

Cassette 4K KIM (hypertape)	12,50
Cassette 8K KIM (hypertape)	12,50
Cassette 8K JUNIOR (hypertape)	12,50
Manual	30,00
Source 4K KIM	30,00
Patches S.Woldringh (4K KIM wordt 8K KIM)	75,00
Patches A.Hanke	gratis
(alleen bij cassette voor JUNIOR + Manual)	

Aanschaf van source en patches is niet perse direct noodzakelijk; men kan volstaan met cassette + manual + patches S.Woldringh di.10 (= overzicht commando's zie DE 6502 KENNER nr. 17).

Patches di. 10 apart 5,00

Overschrijvings op postrekening 841433 t.n.v. W.L. van Peit te Krimpen a.d. IJssel, of d.m.v. blauwe girobetaalkaart. Geen andere wijze van bestellen mogelijk!

Uitnodiging KIM-Club Bijeenkomst.

=====
Datum : zaterdag 19 maart 1983
Lokatie : Gemeenschapshuis "De Zes Gehuchten",
Papenvoort 10 te Geldrop (bij Eindhoven)

Reisroute :

- per auto - vanaf snelweg E3 België - Venlo

Afslag Geldrop, onder E3 door bij Sheraton hotel. 1e stoplicht l.a., na 800 meter bij kruising l.a., die weg 1 KM volgen tot iets voorbij de kerk. Hiernaast is het.

- vanuit Eindhoven

Rondweg, afslag Geldrop bij DAF. Bij stoplicht voor spoorwagviadukt in Geldrop r.a. Weg 1 KM volgen tot iets voorbij de kerk. Hiernaast is het.

vanuit Nuenen - Helmond

Richting Eindhoven door Geldrop volgen. Bij 1e stoplicht na spoorwagviadukt l.a. Verder als bij bovenstaande route.

- per trein - Stoptrein Eindhoven - Weert (ieder half en heel uur vanuit Eindhoven). In Geldrop uitstappen. Voetsanserstunnel naar station door. Deze weg (Toernooiveld) oversteken en r.a. slaan. L.a. (Hertogenlaan). R.a. (Gildestraat). L.a. (Diepenvaart). Aan het einde bij de kruising is het (rechts).

Toegangspreis: incl. koffie, lunch en frisdrankje : Fl.25,-

Programma :
=====

9.30 Ontvangst met koffie
10.00 Openings.
10.15 Fred Harthoorn. Inleiding 6502 simulator in FORTH.
11.00 Koffiepauze
11.30 Fred Harthoorn. Demonstratie 6502 simulator.
12.00 Lunch.
13.00 Ruud Uphoff. Lezing over universele 6502 software.
13.30 Diskussie met Ruud over universele 6502 software.
14.15 Forum
14.45 Markt
15.00 Informeel gedeelte
17.00 Sluiting.

BRENG UW APPARATUUR MEE !!!

6502

A G E N D A

- DO 24-02-83 6500 - GEBRUIKERSGROEP E-CAFE DELFT
DI 15-03-83 HCC-DAGEN IN ANTWERPEN
WO 16-03-83
DO 17-03-83
ZA 19-03-83 KIM CLUB DAG
VOOR 6502 GEBRUIKERS EN ANDEREN
DO 31-03-83 6500 - GEBRUIKERSGROEP E-CAFE DELFT
ZA 09-04-83 REGIONALE KIM CLUB DAG
DO 28-04-83 6500 - GEBRUIKERSGROEP E-CAFE DELFT
DO 05-05-83 TECHNIEK IN VRIJE TIJD IN DE
ZO 08-05-83 JAARBEURS TE UTRECHT
ZA 21-05-83 KIM CLUB DAG
VOOR 6502 GEBRUIKERS EN ANDEREN
DO 02-06-83 6500 - GEBRUIKERSGROEP E-CAFE DELFT
ZA 11-06-83 REGIONALE KIM CLUB DAG
ZA 20-08-83 VERSCHIJNINGSDATUM DE 6502 KENNER
ZA 17-09-83 KIM CLUB DAG
VOOR 6502 GEBRUIKERS EN ANDEREN
ZA 15-10-83 HCC-DAGEN IN UTRECHT
ZO 16-10-83
DO 27-10-83 6500 - GEBRUIKERSGROEP E-CAFE DELFT
ZA 15-11-83 VERSCHIJNINGSDATUM DE 6502 KENNER
ZA 19-11-83 KIM CLUB DAG
VOOR 6502 GEBRUIKERS EN ANDEREN
DO 24-11-83 6500 - GEBRUIKERSGROEP E-CAFE DELFT
ZA 17-12-83 VERSCHIJNINGSDATUM DE 6502 KENNER

'S ZONDAGS VANAF 19.10 UUR VIA HILVERSUM 2
HOBBYSCHOOP, VAAK MET COMPUTERPROGRAMMA'S, DIE
M.B.V. EEN CASSETTERECORDER ZIJN OP TE NEEMEN.
DOOR HET GEBRUIK VAN 'BASICODE' IS ELK UITGE-
ZONDEN PROGRAMMA MET UW COMPUTER IN TE LEZEN.
DE 'BASICODE' VOOR DE OSI EN VOOR DE APPLE II
ZIJN GEPUBLICEERD IN DE HCC-NIEUWSBRIEF NR. 34
VAN NOV/DEC. 1981.
DE 'BASICODE' VOOR JUNIOR IS GEPUBLICEERD IN
EDITIE 23 VAN DE 6502 KENNER, OKTOBER 1982.

TE KOOP:

WEGENS AANSCHAF FLOPPY DRIVE:
DIGITALE CASSETTDRIVE (FOR-
MAAT VAN DE CASSETTES: AUDIO-
CASSETTES) +/- 6000 BAUD, IN-
CLUSIEF BESTURINGSSOFTWARE
VOOR JUNIOR IN EPROM OP DE
ADRESSEN \$E000 - \$E600. WORDT
BESTUURD DOOR DE VIA 6522.
TEGEN ELK AANNEEMLIJK BOD.
Koen van Nieuwenhove,
Consciensestraat 60/4,
3000 LEUVEN BELGIE.

VRAAG EN AANBOD

Ik maak vorderingen met een DOS voor mijn controller.
Ik ben bezig het volledige DOS uit "K2FDOS: A floppy
disk operating system for the 8080" om te zetten voor
de JUNIOR en mijn controller. Na wat ik van het OHIO-
DOS gezien heb in Elektoor meen ik dat het daarvoor
niet onderdoet, echter zonder de nadelen welke aan de
oplossing van Elektoor kleven: geen cassette, alle
oude software overboord, complex gebruik van het DOS.
De volledige referentie:

K2FDOS : A floppy disk operating system for the
8080
Kenneth B. Welles \$20.00
BYTE BOOKS
ISBN 0-87069206-8

Ik ben nu zover dat ik over een basis DOS beschik.
Het is ongeveer 4K groot en bevat de volgende bevelen:
SAVE XXXXX:Y XXXXXX = filename van 6 tekens
beginnend met een letter
Y = disknummer waarop de
file moet komen
LIMITS XXXX,YYYY = grenzen van de te save
file
QUIT = na laden van programma
van de schijf terug
naar DOS hoofdprogramma
GO XXXX = na laden van een program-
ma van de schijf direkt
runnen
RUN XXXXX:Y = filename + drivenummer
laadt en run een program-
ma van schijf (nummer
hanst af van QUIT en GO)
FORMAT = formatteren schijf
DELETE XXXXX:Y = uitwissen filename
RENAME (new filename):(old filename)

Bovensgenoemde werkt allemaal al. Wie geïnteresseerd is
in het samen verder uitbouwen van de DOS gelieve contact
op te nemen met:
Koen van Nieuwenhove, Consciensestraat 60-4,
B 3000 LEUVEN België.
Verder ben ik benieuwd of er mensen zijn die naar een al-
ternatief zoeken voor het OHIO-DOS. Laat het mij weten.
Of neem contact op met de redactiesekretaris.

0 LIST

SCR # 0

- 0 # ## ### #### ##### TE KOOP ##### ### ## #
1 - Een PC-100 (=AIM65) met 8K monitor + 4K ram + 4K assembler
2 8K BASIC (microsoft) + handboeken voor Fl. 975.=
3 - Een 32K ram AIM65 in mooie behuizing met voeding en direkt
4 aansluitbaar op PC-100, AIM65 of KIM via de
5 expansion connector + handboek voor Fl. 780.
6 - Een Rockwell microflex 65 adaptor/buffer module (eurokaart) +
7 asynchronous communications interface adaptor
8 (ASIA) module met user's manual voor ... Fl. 395.
9 - Een 2 x 4K rom FORTH compiler direkt in expansion sockets van
10 PC-100 of AIM65 toepasbaar, origineel van Rockwell
11 met uitgebreid FORTH user's manual voor .. Fl. 175
12 - Bij aankoop van de 4 items ineens zijn de kosten totaal Fl.1990
13 Bovenstaand bij M.A.M. van STEEN
14 VINGERHOED 11 (tel.: 08330 - 21004)
15 6953 BZ DIEREN.

OK

Sedert enige tijd heeft de KIM Gebruikers Club de copyrights verworven op de Micro-Ade van Peter Jennings. Tot wederopzeggings mogen wij de leden de Micro-Ade ter beschikking stellen. Dat doen we sindsdien middels de cassettebibliotheek. De aandachtige lezer zal dat ongetwijfeld niet ontsaan zijn. Inmiddels geniet de Micro-Ade een levendige belangstelling. Bezitters van een KIM of een JUNIOR kunnen beschikken over een voor hun systeem passend programma. De Micro-Ade is echter ontworpen voor gebruik met elke 6502 microcomputer en bestaat zowel uit drie hoofdprogramma's als uit een aantal utility programma's. De hoofdprogramma's zijn een assembler, een disassembler en een teksteditor. De minimum systeem configuratie bestaat uit een 6502 computer, 16K RAM, 1 of 2 cassetterecorders met of zonder start/stop control, en een ASCII input/output device.

De oorspronkelijke Micro-Ade van 4K voor de KIM werd door Sebo Woldringh uitgebreid tot een 8K KIM-versie met commando's als Append, Blockmove, Clear Buffer, Delete Lines, Display address/number last line, Fix (wijzig) Line, Get Source File(s), Append Source File(s), Insert Line(s), Set/Reset Form Feed Flag, Chose Line(s), List, Move Line(s), Number, Load ASCII-format Files, Set/Reset Page Mode, List Memory-bezetting, Duplicate File(s), Save Source File(s), Print Symbol/XREF Tables, Set/Reset Page-per-file/Eject Flag, Print/Change string, Search Line, Assemble/Execute, Set Lines/Screen and Lines/Page, Disassemble, List without Line Numbers, Save chosen lines, Save ASCII-format Files, etc.

Daarna werd door Adri Hankel de 8K KIM-versie weer aangepast voor JUNIOR. De bij de cassette verkrijgbare documentatie is, m.u.v. het manual van meer dan 50 pagina's, niet enkel veelomvattend, het is ook wat moeilijk het direct te overzien. Velen beginnen echter met cassette + manual, het- een heel goed mogelijk is. In dat geval komen af en toe toch wat telefoontjes met de redactie tot stand, omdat bij het opstarten van de assembler in geringe mate van het manual afwijkende handelingen nodig blijken. Hieronder wordt de procedure nog even weergegeven.

- na invoeren X-commando en CR verschijnt PASS 1, waarachter insetoetst moet worden: 00 en CR.
- de computer reageert met een niet zichtbare actie: de PASS 1, waarna weer CR insetoetst moet (soms duurt het even, afhankelijk van de lengte van het programma).
- dan volgt de vraag PRINT ? welke gevolgd moet met Y of N of niets en CR.
- dan XREF? te beantwoorden met Y of N of niets en CR.
- dan ID ? welke gevolgd kan worden door een ID-code of niets en CR.
- hierna staat de cursor op een lese regel; typ weer 00 en CR, en de PASS 2 start en list het programma op scherm uit. Werd geen listing gevraagd, dan verschijnt op de lese regel een punt, waarachter 00 en CR moet komen. Er volgt dan als reactie weer een punt, en met CR beëindigt men.

Het gebruik van Micro-Ade onder de leden vergemakkelijkt het ontwikkelen van programma's in machinetaal. Het aantal inzendingen voor publicatie neemt toe, op papier, op band, of beide. Een verheusende ontwikkeling. De kwaliteit van ons blad neemt toe. Maar goede programma's moeten wel van commentaar worden voorzien, besloot het bestuur, en daar stemt de redactie mee in, al kunnen zich zeer speciale situaties blijven voordoen. Mijn verzoek aan alle inzenders is dan ook: neem hier goede nota van. Met behulp van Micro-Ade is het al helemaal een peuleschil om er commentaar bij te schrijven.

W.L. van Pelt.

Basic Programma Service Rapporten (zie elders in deze editie) is apart op cassette verkrijgbaar door Fl. 12,50 over te schrijven op postrekening 841433 ten name van W.L. van Pelt te Krimpen aan den IJssel, onder vermelding van: cassette Service Rapporten. Andere wijzen van bestellen zijn niet mogelijk. Levering binnen enkele dagen na bijschrijving.

6502

PROGRAMMEERTALEN

```
0010 ;VAN NIEUWENHOVE KOEN.
0020 ;AANPASSINGEN AAN BASIC VN V2.1 VOOR JUNIOR
0030 ;FILE #04
0040 ;GEHEUGENBEREIK : $4774 - $4871
0050 ;ADRES VAN AUTO AANVULLEN OP $50E(H)-$51C(L)
0060 .BA $50E
050E- 06 0070 .BY $06
0080 .BA $51C
051C- D8 0090 .BY $D8
0100 .CE
0110 RGFL .DE $CC
0120 OUTPUT .DE $345
0130 BEGINP .DE $2427
0140 FFF .DE $448
0150 GETNMRS .DE $6AC
0160 .BA $A9
00A9- 0170 ACCU1 .DS 2
00AB- 0180 ACCU2 .DS 2
0190 ;
0200 .BA $43B
043B- D0 0210 .BY $D0
0220 ;
0230 .BA $DD
00DD- 0240 TEMPX .DS 1
00DE- 00 0250 .BY $00
00DF- 00 0260 .BY $00
00E0- 0270 TEMPY .DS 1
00E1- 0280 RGBEGL .DS 1
0290 .BA $EA
00EA- 0300 RGBEGH .DS 1
00EB- 0310 RGINCL .DS 1
00EC- 0320 RGINCH .DS 1
00ED- 0330 TLINEL .DS 1
00EE- 0340 TLINEH .DS 1
0350 BBB .DE $45A
0355 .BA $440
0440- 19 0356 .BY $19
0360 ;
0370 .BA $6D9
06D9- 20 01 07 0380 RG JSR GETNMR
06DC- 20 30 07 0390 JSR FSTRING
06DF- A9 01 0400 LDA #$01
06E1- 85 CC 0410 STA *RGFL
06E3- A9 00 0420 LDA #$00
06E5- 60 0430 RTS
0440 ;
0450 ;
06E6- A5 22 0460 RG1 LDA *$22
06E8- C9 30 0480 CMP #$30
06EA- D0 20 0490 BNE BLANCO
06EC- 4C B5 07 0500 JMP $7B5
```

KB-9

6502

PROGRAMMEERTALEN

06EF-	EA		0510	NOP
06F0-	EA		0511	NOP
06F1-	EA		0512	NOP
06F2-	EA		0513	NOP
06F3-	A5	C7	0530	RETURN2
06F5-	C9	1B	0540	LDA *\$C7
06F7-	F0	03	0550	CMP #\$1B
06F9-	4C	48 04	0560	BEQ BBBB
06FC-	68		0570	JMP FFF
06FD-	48		0580	PLA
06FE-	4C	5A 04	0590	PHA
0701-	20	8F 07	0600	JMP BBB
0704-	20	AC 06	0610	JSR RESREG
0707-	A9	20	0620	JSR GETNMRS
0709-	85	1B	0630	LDA #\$20
070B-	60		0640	STA *\$1B
			0650	RTS
			0660	;
070C-	AC	AC 07	0670	BLANCO
070F-	A2	07	0680	LDY TEMPN
0711-	B9	A3 07	0690	LDX #\$07
0714-	95	1A	0700	LDA STRING-1, Y
0716-	CA		0710	STA *\$1A, X
0717-	88		0720	DEX
0718-	D0	F7	0730	DEY
071A-	A9	20	0740	BNE NCHAR
071C-	95	1A	0750	LDA #\$20
071E-	CA		0760	STA *\$1A, X
071F-	10	FB	0770	DEX
0721-	85	1B	0780	BPL BCHAR
0723-	20	51 07	0790	STA *\$1B
0726-	90	03	0800	JSR ADD
0728-	4C	75 07	0810	BCC FSTRING1
072B-	20	30 07	0820	JMP ERRMESS
072E-	10	C3	0830	JSR FSTRING
			0840	BPL RETURN2
			0850	;
0730-	20	63 07	0860	FSTRING
0733-	A2	00	0870	JSR CSTRING
0735-	E8		0880	LDX #\$00
0736-	BD	FF 00	0890	INX
0739-	9D	A3 07	0900	LDA \$FF, X
073C-	48		0910	STA STRING-1, X
073D-	20	45 03	0920	PHA
0740-	68		0930	JSR OUTPUT
0741-	D0	F2	0940	PLA
0743-	A9	20	0950	BNE NEXTY
0745-	9D	A3 07	0960	LDA #\$20
0748-	8E	AC 07	0970	STA STRING-1, X
074B-	A2	07	0980	STX TEMPN
074D-	8E	27 24	0990	LDX #\$07
0750-	60		1000	STX BEGINP
				RTS

6502

PROGRAMMEERTALEN

```
1010 ;
1020 ;
0751- 18      1030 ADD          CLC
0752- A5 EB   1040          LDA *RGINCL
0754- 65 ED   1050          ADC *TLINEL
0756- 85 EB   1060          STA *RGINCL
0758- A5 EC   1070          LDA *RGINCH
075A- 65 EE   1080          ADC *TLINEH
075C- 85 EC   1090          STA *RGINCH
075E- B0 02   1100          BCS RETURN3
0760- C9 FA   1110          CMP ##FA
0762- 60      1120 RETURN3    RTS
1130 ;
1140 ;
0763- A5 EC   1150 CSTRING   LDA *RGINCH
0765- A4 EB   1160          LDY *RGINCL
0767- 85 AF   1170          STA ##AF
0769- 84 B0   1180          STY ##B0
076B- A2 90   1190          LDX ##90
076D- 38      1200          SEC
076E- 20 C1 3A 1210          JSR $3AC1
0771- 20 69 3C 1220          JSR $3C69
0774- 60      1230          RTS
1240 ;
1250 ;
0775- 20 8F 07 1260 ERRMESS  JSR RESREG
0778- A9 82   1270          LDA #L,ERRMESS1
077A- A0 07   1280          LDY #H,ERRMESS1
077C- 20 18 2A 1290          JSR $2A18
077F- 4C AD 07 1300          JMP $7AD
1310 ;
1320 ;
0782- 4F 55 54 1330 ERRMESS1 .BY 'OUT OF RANGE'
0785- 20 4F 46
0788- 20 52 41
078B- 4E 47 45
078E- 00      1340          .BY $00
1350 ;
1360 ;
078F- A9 0A   1370 RESREG   LDA #$0A
0791- 85 ED   1380          STA *TLINEL
0793- A9 00   1390          LDA #$00
0795- 85 EE   1400          STA *TLINEH
0797- 85 CC   1410          STA *RGFL
0799- 85 EC   1420          STA *RGINCH
079B- 85 EA   1430          STA *RGBEGH
079D- A9 64   1440          LDA ##64
079F- 85 EB   1450          STA *RGINCL
07A1- 85 E1   1460          STA *RGBEGL
07A3- 60      1470          RTS
1480 ;
1490 ;
07A4-          1500 STRING   .DS 8
```

```

07AC-          1510  TEMPN          .DS 1
07AD- A9 00          1520          LDA #$00
07AF- 8D 27 24      1530          STA BEGINP
07B2- 4C 48 23      1540          JMP $2348
07B5- AC AC 07      1550          LDY TEMPN
07B8- A2 07          1560          LDX #$07
07BA- B9 A3 07      1570  NYCHAR    LDA STRING-1,Y
07BD- 95 1A          1580          STA *$1A,X
07BF- CA            1590          DEX
07C0- 98            1600          DEY
07C1- D0 F7          1610          BNE NYCHAR
07C3- A9 20          1620          LDA #$20
07C5- 95 1A          1630  BYCHAR    STA *$1A,X
07C7- CA            1640          DEX
07C8- 10 FB          1650          BPL BYCHAR
07CA- 85 1B          1660          STA *$1B
07CC- A9 00          1670          LDA #$00
07CE- 8D 27 24      1680          STA BEGINP
07D1- 85 CC          1690          STA *RGFL
07D3- 4C F3 06      1700          JMP $EF3
07D6- EA            1710  ENDDAT    NOP
                                1720          .BA $4287
4287- A9 D6          1730  VERPL3    LDA #L,ENDDAT
                                1740          .BA $428D
428D- A9 07          1750          LDA #H,ENDDAT
                                1760          .EN

```

PATCHES KIM-1 BASIC DEEL 1
(zie DE 6502 KENNER nr. 22)

```

Regel          3600  was          21C0  4F D7          .BY #4F #D7
                wordt         21C0  4C 4F D7        .BY $4C $4F $D7
                3680  was          22E8  20 0E 03      JSR ERRSTA
                wordt         22E8  4C 0E 03      JMP ERRSTA

```

PATCHES KIM-1 BASIC DEEL 2
(zie DE 6502 KENNER nr. 23)

```

Regel          1660  vervalt
                1520  was          (niets ingevuld, red.)
                1520  wordt         042E  EA          ENDDAT NOP
                1700  was          4287  A9 2E          VERPL3
                1700  wordt         4287  A9 2E          VERPL3
                1730  was          428D  A9 04          LDA #$2E
                1730  wordt         428D  A9 04          LDA #L,ENDDAT
                                                LDA #$04
                                                LDA #H,ENDDAT

```

PATCHES KIM-1 BASIC DEEL 3
(zie DE 6502 KENNER nr. 24)

```

Regel          1970  was          0571  8D 2E 02      STA $22E
                wordt         0571  8D 2E 03      STA $32E

```

6502 SYSTEM SOFTWARE

```

0010      * * * * *
0020      *
0030      * BASIC PROGRAM COMPRESSOR *
0040      *
0050      * FOR "MICROSOFT" BASIC *
0060      *
0070      * BY: RUUD UPHOFF 1981 *
0080      *
0090      * --- UNIVERSAL VERSION --- *
0100      *
0110      * SET YOUR TYPE AT LINE 210 *
0120      *
0130      * * * * *
0140
0150 TYPE      .DE 0
0160 PET.2001  .DE 1
0170 CBM.3032  .DE 2
0180 APPLE.II  .DE 3
0190 SYM       .DE 4
0200
0210          SET TYPE=APPLE.II
0220
0230          .OC
0240
0250          =====
0260          MACHINE TYPE DEPENDENT VARIABLES
0270          =====
0280
0290          IFE TYPE-PET.2001      (OLD ROM'S)
0300
0310          PET BASIC STATEMENT TOKENS
0320
0330 DATA      .DE $83
0340 DIM         .DE $86
0350 LET        .DE $88
0360 GOTO       .DE $89
0370 GOSUB      .DE $8D
0380 RETURN     .DE $8E
0390 REM        .DE $8F
0400 THEN       .DE $A7
0410
0420          .BA $7000
0430
0440          BASIC POINTERS:
0450
0460 STRT       .DE $7A          PTR TO SCND BYTE OF PGM
0470 ENDP       .DE $7C          PTR JUST BEYOND PGM TEXT
0480 ENDP1      .DE $7C          COPY OF SAME
0490 BASIC      .DE $C38B       WARM RESTART OF BASIC
0500
0510          ***
0520
0530          IFE TYPE-CBM.3032      (NEW ROM'S)
0540
0550          CBM BASIC STATEMENT TOKENS
0560          CHANGE FOR OTHER MPU 6502 MACHINE

```

6502 SYSTEM SOFTWARE

```
0570
0580 DATA .DE #83
0590 DIM .DE #86
0600 LET .DE #88
0610 GOTO .DE #89
0620 GOSUB .DE #8D
0630 RETURN .DE #8E
0640 REM .DE #8F
0650 THEN .DE #A7
0660
0670 .BA $7000
0680
0690 BASIC POINTERS:
0700
0710 STRT .DE #28 PTR TO SCND BYTE OF PGM
0720 ENDP .DE #2A PTR JUST BEYOND PGM TEXT
0730 ENDP1 .DE #2A COPY OF SAME
0740 BASIC .DE #C389 WARM RESTART OF BASIC
0750
0760 ***
0770
0780 IFE TYPE-APPLE.II (APPLESOFT=MICROSOFT)
0790
0800 APPLESOFT STATEMENT TOKENS
0810 CHANGE FOR OTHER MPU 6502 MACHINE
0820
0830 DATA .DE 131
0840 DIM .DE 134
0850 LET .DE 170
0860 GOTO .DE 171
0870 GOSUB .DE 176
0880 RETURN .DE 177
0890 REM .DE 178
0900 THEN .DE 196
0910
0920 .BA $9000
0930
0940 BASIC POINTERS:
0950
0960 STRT .DE #67 PTR TO SCND BYTE OF PGM
0970 ENDP .DE #69 PTR JUST BEYOND PGM TEXT
0980 ENDP1 .DE #AF COPY OF SAME
0990 BASIC .DE #D43C WARM RESTART OF BASIC
1000
1010 ***
1020
1030 IFE TYPE-SYM ALSO FOR MOST KIM/JUNIOR VERSIONS (KB9)
1040
1050 SYM BASIC STATEMENT TOKENS
1060 CHANGE FOR OTHER MPU 6502 MACHINE
1070
1080 DATA .DE #83
1090 DIM .DE #85
1100 LET .DE #87
1110 GOTO .DE #88
1120 GOSUB .DE #8C
1130 RETURN .DE #8D
1140 REM .DE #8E
```

6502 SYSTEM SOFTWARE

```

1150 THEN .DE $A1
1160
1170 .BA $B000 CHANGE START ADDRESS FOR OTHER MACHINES
1180 .OS
1190
1200 BASIC POINTERS:
1210
1220 STRT .DE $7B PTR TO SCND BYTE OF PGM
1230 ENDP .DE $7D PTR JUST BEYOND PGM TEXT
1240 ENDP1 .DE $7D COPY OF SAME
1250 BASIC .DE $C27E WARM RESTART OF BASIC
1260
1270 ***
1280
1290
1300 =====
1310 PGM "COMPRESS" FOR ALL 6502 MPU
1320 VERSIONS OF "MICROSOFT" BASIC
1330 =====
1340
1350 PAGE ZERO: DON'T CHANGE !
1360 PTRR .DE 0
1370 TPTR .DE 2
1380 PTRS .DE 4
1390 NUMB .DE 6
1400 LEN .DE 8
1410 TMPD .DE 9
1420 SVKEY .DE 10
1430
1440 MAIN PGM
1450
9000- 78 1460 START SET NO ITP'S
9001- A2 00 1470 LDX #$00 SAVE PAGE ZERO AREA
9003- B5 00 1480 LDA *PTRR, X
9005- 48 1490 PHA
9006- E8 1500 INX
9007- E0 0B 1510 CPX #SVKEY+1
9009- D0 F8 1520 BNE SAVE
900B- 20 F0 90 1530 JSR INIT INIT POINTERS
900E- 20 C3 90 1540 REP JSR CPLINE COMPRESS CURRENT LINE
9011- 20 4F 90 1550 JSR NXTLN CHECK NEXT LINE
9014- 90 F8 1560 BCC REP IF NOT END OF PGM, REPEAT
9016- 20 6C 91 1570 JSR GETTAB COMPUTE REFERENCE TABLE
9019- 20 E0 91 1580 JSR APPEND APPEND LINES
901C- A2 0A 1590 LDX #SVKEY RESTORE PAGE ZERO AREA
901E- 68 1600 RE PLA
901F- 95 00 1610 STA *PTRR, X
9021- CA 1620 DEX
9022- 10 FA 1630 BPL RE
9024- 58 1640 CLI ENABLE ITP'S AGAIN
9025- A2 FA 1650 LDX #$FA RESET STACK
9027- 9A 1660 TXS
9028- 4C 3C D4 1670 JMP BASIC REENTER BASIC INTERPRETER
1680
1690 GET NEXT BYTE FROM OLD PGM
1700
902B- E6 00 1710 GET INC *PTRR
902D- D0 02 1720 BNE READ

```

6502 SYSTEM SOFTWARE

902F- E6 01	1730	INC *PTRR+1	
9031- A0 00	1740 READ	LDY #400	
9033- B1 00	1750	LDA (PTRR),Y	
9035- 60	1760	RTS	
	1770		
	1780		PUT BYTE AT NEXT ADDRESS INTO NEW PGM
	1790		
9036- E6 04	1800 PUT	INC *PTRS	
9038- D0 02	1810	BNE STORE	
903A- E6 05	1820	INC *PTRS+1	
	1830		
	1840		STORE BYTE AT CURRENT ADDRESS INTO NEW PGM
	1850		
903C- A0 00	1860 STORE	LDY #400	
903E- 91 04	1870	STA (PTRS),Y	
	1880		
	1890		TEST BYTE LAST COPIED FOR ALPHA
	1900		
9040- C9 41	1910 TEST	CMP #41	IF BELOW ALPHA
9042- 90 07	1920	BCC NOT	
9044- C9 5B	1930	CMP #5B	OR ABOVE ALPHA
9046- B0 03	1940	BCS NOT	THEN RETURN WITH C=0
9048- 38	1950	SEC	IF ALPHA THEN C=1
9049- B0 01	1960	BCS BACK1	
904B- 18	1970 NOT	CLC	
904C- 29 FF	1980 BACK1	AND #4FF	RESET Z AND N FROM A
904E- 60	1990	RTS	
	2000		
	2010		GOTO NEXT VALID LINE
	2020		
	2030		1. TEST FOR END OF PGM
	2040		
904F- A0 02	2050 NXTLN	LDY #402	
9051- B1 00	2060	LDA (PTRR),Y	
9053- D0 1A	2070	BNE SKIP	
	2080		
	2090		2. IF END OF PGM
	2100		
9055- 20 36 90	2110 END	JSR PUT	PUT 00 00 BEYOND PGM
9058- 20 36 90	2120	JSR PUT	
905B- 20 36 90	2130	JSR PUT	ADJUST PTR BEYOND PGM
905E- A5 04	2140	LDA *PTRS	COPY PTR AS NEW PTR
9060- 85 69	2150	STA *ENDP	END OF PGM
9062- 85 AF	2160	STA *ENDP1	ALSO INTO COPY OF SAME
9064- A5 05	2170	LDA *PTRS+1	
9066- 85 6A	2180	STA *ENDP+1	
9068- 85 B0	2190	STA *ENDP1+1	
906A- 20 B6 92	2200	JSR RECOMP	RECOMPUTE LINKS
906D- 38	2210	SEC	AND RETURN WITH C=1 MEANS "ENDOF PGM"
906E- 60	2220	RTS	
	2230		
	2240		3. IF NOT END OF PGM
	2250		THEN SKIP TO NEXT VALID BYTE
	2260		
906F- A0 04	2270 SKIP	LDY #404	INDEX OVER HEADER
9071- C8	2280 SCAN	INY	INDEX TO NEXT BYTE
9072- B1 00	2290	LDA (PTRR),Y	
9074- F0 0C	2300	BEQ DISMISS	IGNORE EMPTY LINE

6502 SYSTEM SOFTWARE

9076- C9 20	2310	CMP #420	SKIP BLANKS
9078- F0 F7	2320	BEQ SCAN	
907A- C9 3A	2330	CMP #43A	SKIP COLONS
907C- F0 F3	2340	BEQ SCAN	
907E- C9 B2	2350	CMP #REM	IF NOT "REM"
9080- D0 0A	2360	BNE ACCEPT	GO ACCEPT LINE
	2370		
	2380		
	2390		
9082- 20 10 91	2400 DISMISS	JSR DSMHDR	SKIP HEADER
9085- 20 2B 90	2410 DISS	JSR GET	SKIP BYTES
9088- D0 FB	2420	BNE DISS	UNTIL END OF LINE
908A- F0 C3	2430	BEQ NXTLN	GO TRY NEXT LINE
	2440		
	2450		
	2460		
908C- 98	2470 ACCEPT	TYA	SAVE INDEX
908D- 48	2480	PHA	
908E- 20 A0 90	2490	JSR CPHDR	COPY HEADER
9091- 68	2500	PLA	GET INDEX IN A
9092- 38	2510	SEC	
9093- E9 05	2520	SBC #405	MINUS HEADER AND 1 VALID BYTE
9095- 18	2530	CLC	
9096- 65 00	2540	ADC *PTRR	
9098- 85 00	2550	STA *PTRR	REST MUST BE GARBAGE
909A- 90 02	2560	BCC BACK4	
909C- EE 01	2570	INC *PTRR+1	
909E- 18	2580 BACK4	CLC	C=0 MEANS "NOT END OF PGM"
909F- 60	2590	RTS	
	2600		
	2610		
	2620		
90A0- A2 04	2630 CPHDR	LDX #404	COPY 4 BYTES
90A2- 20 2E 90	2640 CPH	JSR GET	
90A5- 20 36 90	2650	JSR PUT	
90A8- CA	2660	DEX	
90A9- D0 F7	2670	BNE CPH	
90AB- 60	2680	RTS	
	2690		
	2700		
	2710		
90AC- A9 22	2720 ST.QUOTE	LDA #' "	INSERT A QUOTE
90AE- 20 3C 90	2730	JSR STORE	
90B1- A9 00	2740	LDA #400	BEFORE END OF LINE
90B3- 20 36 90	2750	JSR PUT	
90B6- 60	2760	RTS	
	2770		
	2780		
	2790		
90B7- 20 2B 90	2800 STRING	JSR GET	COPY BYTES
90BA- 20 36 90	2810	JSR PUT	
90BD- F0 ED	2820	BEQ ST.QUOTE	UNTIL END OF LINE
90BF- C9 22	2830	CMP #422	
90C1- D0 F4	2840	BNE STRING	OR NEXT QUOTE
	2850		
	2860		
	2870		
90C3- 20 2B 90	2880 CPLINE	JSR GET	GET NEXT BYTE

6502 SYSTEM SOFTWARE

90C6- C9 20	2890	CMP #20	
90C8- F0 F9	2900	BEQ CPLINE	SKIP BLANKS
90CA- C9 AA	2910	CMP #LET	
90CC- F0 F5	2920	BEQ CPLINE	SKIP "LET"
90CE- C9 B2	2930	CMP #REM	IF "REM"
90D0- D0 0A	2940	BNE LEGAL	
90D2- 20 2B 90	2950 SKPREM	JSR GET	SKIP BYTES
90D5- D0 FB	2960	BNE SKPREM	UNTIL END OF LINE
90D7- 20 3C 90	2970	JSR STORE	STORE END OF LINE
90DA- F0 13	2980	BEQ ENDSUB	AND RETURN
90DC- 20 36 90	2990 LEGAL	JSR PUT	COPY ALL OTHER BYTES
90DF- F0 0E	3000	BEQ ENDSUB	GO RETURN IF END OF LINE
90E1- 90 06	3010	BCC NOVAR	IF BYTE WAS ALPHA CHAR.
90E3- 20 36 92	3020	JSR CPVAR	COPY VARIABLE
90E6- 4C C3 90	3030	JMP CPLINE	AND REPEAT
90E9- C9 22	3040 NOVAR	CMP #22	IF NO QUOTE
90EB- D0 D6	3050	BNE CPLINE	GO DO NEXT BYTE
90ED- F0 C8	3060	BEQ STRING	ELSE GO COPY STRING FIRST
90EF- 60	3070 ENDSUB	RTS	
	3080		
	3090	INIT PTRS/PTRP	
	3100		
90F0- A5 67	3110 INIT	LDA *STRT	SET BOTH POINTERS
90F2- 38	3120	SEC	
90F3- E9 02	3130	SBC #402	TWO BYTES BELOW START OF PGM
90F5- 85 04	3140	STA *PTRS	(JUST BEFORE DUMMY END OF LINE)
90F7- 85 00	3150	STA *PTRR	
90F9- A5 68	3160	LDA *STRT+1	
90FB- E9 00	3170	SBC #400	
90FD- 85 05	3180	STA *PTRS+1	
90FF- 85 01	3190	STA *PTRR+1	
9101- A9 00	3200	LDA #400	
9103- 85 09	3210	STA *TMPD	
9105- 85 0A	3220	STA *SVKEY	
	3230		
	3240	INIT TABLE POINTER	
	3250		
9107- A9 FE	3260 INTAB	LDA #L, TABLE-2	
9109- 85 02	3270	STA *TPTR	
910B- A9 92	3280	LDA #H, TABLE-2	
910D- 85 03	3290	STA *TPTR+1	
910F- 60	3300	RTS	
	3310		
	3320	DISMISS HEADER	
	3330		
9110- A9 04	3340 DSMHDR	LDA #404	ADD LENGHT OF HEADER
9112- 18	3350	CLC	
9113- 65 00	3360	ADC *PTRR	TO READ POINTER
9115- 85 00	3370	STA *PTRR	
9117- 90 02	3380	BCC BACK2	
9119- E6 01	3390	INC *PTRR+1	
911B- 60	3400 BACK2	RTS	
	3410		
	3420	PUT NUMBER FROM PTRS IN TABLE	
	3430		
911C- 20 36 91	3440 PUTNUM	JSR SRCHNUM	IF NOT ALREADY PRESENT
911F- B0 14	3450	BCS BACK5	
9121- 20 60 91	3460	JSR ADVANCE	ADVANCE PTR TO NEXT TWO BYTES

6502 SYSTEM SOFTWARE

9124- A0 00	3470	LDY #00	
9126- A5 06	3480	LDA *NUMB	PUT NEW NUMBER THERE
9128- 91 02	3490	STA (TPTR),Y	
912A- C8	3500	INY	
912B- A5 07	3510	LDA *NUMB+1	
912D- 91 02	3520	STA (TPTR),Y	
912F- C8	3530	INY	WRITE NEW "END OF TABLE"
9130- C8	3540	INY	
9131- A9 FF	3550	LDA #FF	
9133- 91 02	3560	STA (TPTR),Y	
9135- 60	3570 BACK5	RTS	
	3580		
	3590		SEARCH NUMBER IN TABLE
	3600		
9136- A5 02	3610 SACHNUM	LDA *TPTR	SAVE CURRENT POINTER
9138- 48	3620	PHA	
9139- A5 03	3630	LDA *TPTR+1	
913B- 48	3640	PHA	
913C- 20 07 91	3650	JSR INTAB	SET POINTER TO BEGIN
913F- 20 60 91	3660 NXTNUMB	JSR ADVANCE	PTR TO NEXT NUMBER
9142- A0 01	3670	LDY #01	
9144- B1 02	3680	LDA (TPTR),Y	READ NUMBER
9146- C9 FF	3690	CMP #FF	IF END OF TABLE
9148- F0 0E	3700	BEQ NTFND	THEN NUMBER NOT PRESENT
914A- C5 07	3710	CMP *NUMB+1	ELSE, COMPARE NUMBERS
914C- D0 F1	3720	BNE NXTNUMB	AND TRY NEXT IF NOT SAME
914E- 28	3730	DEY	
914F- B1 02	3740	LDA (TPTR),Y	
9151- C5 06	3750	CMP *NUMB	
9153- D0 EA	3760	BNE NXTNUMB	
9155- 38	3770	SEC	C=1 MEANS FOUND
9156- 00 01	3780	BCS BACK3	
9158- 18	3790 NTFND	CLC	C=0 MEANS NOT FOUND
9159- 68	3800 BACK3	PLA	
915A- 85 03	3810	STA *TPTR+1	RESTORE CURRENT PTR
915C- 68	3820	PLA	
915D- 85 02	3830	STA *TPTR	
915F- 60	3840	RTS	BEFORE RETURN
	3850		
	3860		ADVANCE TPTR
	3870		
9160- 18	3880 ADVANCE	CLC	ADD 2
9161- A5 02	3890	LDA *TPTR	TO TABLE POINTER
9163- 69 02	3900	ADC #02	
9165- 85 02	3910	STA *TPTR	
9167- 90 02	3920	BCC BACK6	
9169- E6 03	3930	INC *TPTR+1	
916B- 60	3940 BACK6	RTS	
	3950		
	3960		COMPUTE TABLE OF REF. LINES
	3970		
916C- 20 F0 90	3980 GETTAB	JSR INIT	INIT "TABLE EMPTY"
916F- A9 FF	3990	LDA #FF	SET "END OF TABLE"
9171- A0 03	4000	LDY #03	
9173- 91 02	4010	STA (TPTR),Y	
9175- A9 00	4020	LDA #00	SET "NO JUMP INSTRUCTION"
9177- 85 09	4030	STA *TMP0	
9179- A0 02	4040	LDY #02	

6502 SYSTEM SOFTWARE

917B- B1 57	4050	LDA (STRT),Y	PUT FIRST LINE FROM PGM
917D- 85 06	4060	STA *NUMB	
917F- C8	4070	INY	
9180- B1 57	4080	LDA (STRT),Y	
9182- 85 07	4090	STA *NUMB+1	
9184- 20 1C 91	4100	JSR PUTNUM	ALWAYS INTO TABLE
	4110		
	4120		
	4130	SCAN PGM FOR REFERENCED LINES	
9187- 20 2B 90	4140 RPT	JSR GET	CHECK NEXT BYTE
918A- D0 22	4150	BNE FURTHER	
	4160		
	4170	1. IF END OF LINE	
	4180		
918C- A0 02	4190	LDY #402	IF END OF PGM
918E- B1 00	4200	LDA (PTRR),Y	
9190- F0 4D	4210	BEQ ENDSB	GO RETURN
9192- A5 09	4220	LDA *TMPD	IF PREVIOUS LINE WAS NOT A JUMP
9194- F0 12	4230	BEQ HDR	GO SKIP HEADER AND REPEAT
9196- A0 03	4240	LDY #403	ELSE GET NUMBER FROM NEXT LINE
9198- B1 00	4250	LDA (PTRR),Y	
919A- 85 06	4260	STA *NUMB	INTO NUMBER BUFFER
919C- C8	4270	INY	
919D- B1 00	4280	LDA (PTRR),Y	
919F- 85 07	4290	STA *NUMB+1	
91A1- 20 1C 91	4300	JSR PUTNUM	AND PUT NUMBER INTO TABLE
91A4- A9 00	4310	LDA #400	SET "PREVIOUS LINE NO JUMP"
91A6- 85 09	4320	STA *TMPD	
91A8- 20 10 91	4330 HDR	JSR DSMHDR	SKIP HEADER
91AB- 4C 87 91	4340	JMP RPT	AND GO TEST NEXT LINE
	4350		
	4360	2. IF NOT END OF LINE	
	4370		
91AE- C9 AB	4380 FURTHER	CMP #GOTO	IF "GOTO"
91B0- F0 10	4390	BEQ JUMP	PUT LINE INTO TABLE
91B2- C9 B0	4400	CMP #GOSUB	IF "GOSUB"
91B4- F0 0E	4410	BEQ JMP	PUT INTO TABLE BUT DON'T SET FLAG
91B6- C9 C4	4420	CMP #THEN	IF "THEN"
91B8- F0 08	4430	BEQ JUMP	PUT INTO TABLE
91BA- C9 B1	4440	CMP #RETURN	IF NOT "RETURN"
91BC- D0 C9	4450	BNE RPT	GO CHECK NEXT BYTE
91BE- 85 09	4460	STA *TMPD	ELSE SET FLAG ONLY
91C0- F0 C5	4470	BEQ RPT	AND GO REPEAT
	4480		
	4490	FOUND A JUMP (GOTO OR THEN)	
	4500		
91C2- 85 09	4510 JUMP	STA *TMPD	SET FLAG
	4520		
	4530	FOUND A JUMP (GOSUB)	
	4540		
91C4- A0 01	4550 JMP	LDY #401	TEST NEXT BYTE
91C6- B1 00	4560	LDA (PTRR),Y	
91C8- C9 2C	4570	CMP #',	IF TERMINATOR (ON GOTO)
91CA- D0 05	4580	BNE CNM	
91CC- 20 2B 90	4590	JSR GET	SKIP IT,
91CF- D0 05	4600	BNE CLCT	AND COLLECT NEXT NUMBER
91D1- 20 5C 92	4610 CNM	JSR CHNUM	IF NOT NUMERIC
91D4- B0 B1	4620	BCS RPT	GO REPEAT

6502 SYSTEM SOFTWARE

91D6- 20 69 92	4630	CLCT	JSR COLLECT	ELSE, COLLECT NUMBER
91D9- 20 1C 91	4640		JSR PUTNUM	PUT INTO TABLE
91DC- 4C C4 91	4650		JMP JMP	TRY MORE NUMBERS
91DF- 60	4660	ENDSB	RTS	
	4670			
	4680		APPEND LINES	
	4690			
91E0- 20 F0 90	4700	APPEND	JSR INIT	INIT. POINTERS
91E3- A9 00	4710		LDA #00	CLEAR FLAG
91E5- 85 09	4720		STA *TMPD	
91E7- 85 08	4730		STA *LEN	CLEAR LENGTH COUNT
91E9- 20 2B 90	4740		JSR GET	COPY FIRST DUMMY END OF LINE
91EC- 20 36 90	4750		JSR PUT	
91EF- A0 02	4760	CPNXT	LDY #02	CHECK HIGH BYTE FROM LINK
91F1- B1 00	4770		LDA (PTRR),Y	
91F3- D0 03	4780		BNE NUMBER	IF ZERO,
91F5- 4C 55 90	4790		JMP END	END OF PROGRAM.
91F8- C8	4800	NUMBER	INY	PUT LINE NUMBER
91F9- B1 00	4810		LDA (PTRR),Y	
91FB- 85 06	4820		STA *NUMB	INTO SAVE AREA
91FD- C8	4830		INY	
91FE- B1 00	4840		LDA (PTRR),Y	
9200- 85 07	4850		STA *NUMB+1	
9202- 20 36 91	4860		JSR SRCHNUM	SEARCH NUMBER IN TABLE
9205- 90 18	4870		BCC SCANLN	
	4880			
	4890			
	4900			
	4910			
			1. FOUND NUMBER IN TABLE.	
			SO DON'T APPEND NEXT LINE	
9207- 20 A0 90	4920	EXACT	JSR CPHDR	COPY HEADER
920A- 86 08	4930		STX *LEN	SAVE NEW LENGHT
920C- A9 00	4940		LDA #00	SET PREVIOUS KEYWORD INVALID
920E- 85 0A	4950		STA *SVKEY	
9210- 20 2B 90	4960	EXT	JSR GET	READ NEXT BYTE
9213- 20 E3 92	4970		JSR SPEC	CHECK FOR DIM OR DATA
9215- E5 08	4980		INC *LEN	AND COUNT LENGHT
9218- 20 36 90	4990		JSR PUT	COPY BYTE
921B- D0 F3	5000		BNE EXT	NEXT BYTE, IF NOT END OF LINE
921D- F0 D0	5010		BEO CPNXT	ELSE, GO DO NEXT LINE
	5020			
	5030			
	5040			
	5050			
			2. NUMBER NOT FOUND IN TABLE	
			SO APPEND TO PREVIOUS LINE	
921F- A0 04	5060	SCANLN	LDY #04	INDEX BEYOND HEADER
9221- C8	5070	CNTLEN	INY	COUNT BYTES IN Y
9222- B1 00	5080		LDA (PTRR),Y	
9224- D0 FB	5090		BNE CNTLEN	UNTIL END OF LINE
9226- 98	5100		TYA	
9227- 18	5110		CLC	ADD RESULT TO COUNT
9228- 65 08	5120		ADC *LEN	IF RESULT >255
922A- B0 DB	5130		BCC EXACT	DON'T APPEND, BUT CREATE NEW LINE
922C- A9 3A	5140		LDA #3A	ELSE, INSERT COLON
922E- 20 3C 90	5150		JSR STORE	
9231- 20 10 91	5160		JSR DSMHDR	SKIP HEADER
9234- D0 DA	5170		BNE EXT	GO COPY LINE
	5180			
	5190			
	5200			
			COPY VARIABLE NAME	

6502 SYSTEM SOFTWARE

9236- C8	5210 CPVAR	INY	TEST NEXT BYTE
9237- B1 00	5220	LDA (PTRR),Y	
9239- 20 40 90	5230	JSR TEST	IF ALPHA
923C- B0 05	5240	BCS TAKE	
923E- 20 5C 92	5250	JSR CHNUM	OR NUMERIC
9241- B0 18	5260	BCS BACK8	
9243- 20 2B 90	5270 TAKE	JSR GET	THEN COPY 2ND CHAR FROM VAR.NAME
9246- 20 36 90	5280	JSR PUT	
9249- C8	5290 SKPNAME	INY	REST FROM NAME IS GARBAGE
924A- B1 00	5300	LDA (PTRR),Y	
924C- 20 40 90	5310	JSR TEST	THUS IF ALPHA
924F- B0 05	5320	BCS REMV	
9251- 20 5C 92	5330	JSR CHNUM	OR NUMERIC
9254- B0 05	5340	BCS BACK8	
9256- 20 2B 90	5350 REMV	JSR GET	SKIP CHARACTER
9259- D0 EE	5360	BNE SKPNAME	AND REPEAT
925B- 60	5370 BACK8	RTS	
	5380		
	5390		CHECK FOR NUMERIC CHARACTER
	5400		
925C- C9 3A	5410 CHNUM	CMP #3A	IF ABOVE,
925E- B0 07	5420	BCS NTNM	
9260- C9 3A	5430	CMP #30	OR BELOW
9262- 90 03	5440	BCC NTNM	THEN C=1 MEANS NOT NUMERIC
9264- 18	5450	CLC	ELSE C=0 MEANS NUMERIC
9265- 90 01	5460	BCC BACK9	
9267- 38	5470 NTNM	SEC	
9268- 60	5480 BACK9	RTS	
	5490		
	5500		COLLECT DECIMAL NUMBER
	5510		
9269- A9 00	5520 COLLECT	LDA #00	CLEAR ANSWER AREA
926B- 85 06	5530	STA *NUMB	
926D- 85 07	5540	STA *NUMB+1	
926F- 20 2B 90	5550 LOOP	JSR GET	GET NEXT CHARACTER
9272- 20 5C 92	5560	JSR CHNUM	IF NOT NUMERIC
9275- B0 12	5570	BCS BACKA	THEN READY
9277- 48	5580	PHA	SAVE DECIMAL CHAR.
9278- 20 32 92	5590	JSR MULTY	MULTIPLY RESULT BY 10
927B- 68	5600	PLA	RESTORE DIGIT
927C- 29 0F	5610	AND #0F	DROP ASCII BASE
927E- 18	5620	CLC	
927F- 65 06	5630	ADC *NUMB	ADD TO RESULT
9281- 85 06	5640	STA *NUMB	
9283- 90 EA	5650	BCC LOOP	
9285- E6 07	5660	INC *NUMB+1	
9287- B0 E6	5670	BCS LOOP	REPEAT FOR NEXT DIGIT
9289- A5 00	5680 BACKA	LDA *PTRR	IF READY, DECREASE POINTER
928B- D0 02	5690	BNE BRW	
928D- C6 01	5700	DEC *PTRR+1	
928F- C6 00	5710 BRW	DEC *PTRR	
9291- 60	5720	RTS	AND RETURN
	5730		
	5740		MULTIPLY NUMB BY 10
	5750		
9292- 20 B1 92	5760 MULTY	JSR TWO	10 = 2 * 5
9295- 20 99 92	5770	JSR FIVE	
9298- 60	5780	RTS	

6502 SYSTEM SOFTWARE

	5790			
	5800		MULTIPLY NUMB BY 5	
	5810			
9239-	A5 07	5820 FIVE	LDA *NUMB+1	SAVE NUMBER
923B-	48	5830	PHA	
923C-	A5 06	5840	LDA *NUMB	
923E-	48	5850	PHA	
923F-	20 B1 92	5860	JSR TWO	MULTIPLY BY TWO
92A2-	20 B1 92	5870	JSR TWO	TWO TIMES MAKES * 4
92A5-	68	5880	PLA	RESTORE NUMBER
92A6-	18	5890	CLC	
92A7-	65 06	5900	ADC *NUMB	ADD TO RESULT
92A9-	85 06	5910	STA *NUMB	
92AB-	68	5920	PLA	
92AC-	65 07	5930	ADC *NUMB+1	
92AE-	85 07	5940	STA *NUMB+1	
92B0-	60	5950	RTS	
	5960			
	5970		MULTIPLY NUMB BY 2	
	5980			
92B1-	06 06	5990 TWO	ASL *NUMB	LEFT SHIFT
92B3-	26 07	6000	ROL *NUMB+1	ON DOUBLE BYTE
92B5-	60	6010	RTS	
	6020			
	6030		RECOMPUTE LINKS	
	6040			
92B6-	A5 67	6050 RECOMP	LDA *STRT	COPY START POINTER
92B8-	85 06	6060	STA *NUMB	
92BA-	A5 68	6070	LDA *STRT+1	
92BC-	85 07	6080	STA *NUMB+1	
92BE-	18	6090	CLC	
92BF-	A0 01	6100 NXTLNK	LDY #01	GET HIGH BYTE FROM LINK
92C1-	B1 06	6110	LDA (NUMB),Y	IF ZERO,
92C3-	F0 1D	6120	BEQ BACK7	THEN END OF PGM
92C5-	A0 04	6130	LDY #04	
92C7-	C8	6140 COUNTER	INY	ELSE, COUNT OFFSET,
92C8-	B1 06	6150	LDA (NUMB),Y	
92CA-	D0 FB	6160	BNE COUNTER	UNTIL END OF LINE
92CC-	C8	6170	INY	
92CD-	98	6180	TYA	
92CE-	65 06	6190	ADC *NUMB	ADD TO PREVIOUS LINK
92D0-	AA	6200	TAX	
92D1-	A0 00	6210	LDY #00	
92D3-	91 06	6220	STA (NUMB),Y	
92D5-	A5 07	6230	LDA *NUMB+1	
92D7-	69 00	6240	ADC #00	
92D9-	C8	6250	INY	
92DA-	91 06	6260	STA (NUMB),Y	STORE LINK
92DC-	86 06	6270	STX *NUMB	
92DE-	85 07	6280	STA *NUMB+1	
92E0-	D0 DD	6290	BNE NXTLNK	AND REPEAT
92E2-	60	6300 BACK7	RTS	
	6310			
	6320		APPEND DIM AND DATA	
	6330			
92E3-	10 1A	6340 SPEC	BPL NOKEY	IF KEYWORD TOKEN
92E5-	C5 0A	6350	CMP *SVKEY	AND SAME AS PREVIOUS
92E7-	F0 04	6360	BEQ DTSMT	THEN GO CHECK DIM OR DATA

6502 SYSTEM SOFTWARE

92E9- 85 0A	6370	STA *SVKEY	ELSE, SAVE NEW KEYWORD
92EB- D0 12	6380	BNE NOKEY	AND GO RETURN
92ED- C9 83	6390 DTSMT	CMP #DATA	IF DATA
92EF- F0 04	6400	BEQ DCRS	
92F1- C9 86	6410	CMP #DIM	OR DIM
92F3- D0 0A	6420	BNE NOKEY	
92F5- A5 04	6430 DCRS	LDA *PTRS	DECREASE POINTER INTO NEW TEXT
92F7- D0 02	6440	BNE NOTZER	
92F9- C6 05	6450	DEC *PTRS+1	
92FB- C6 04	6460 NOTZER	DEC *PTRS	
92FD- A9 2C	6470	LDA #',	INSERT A ",,"
92FF- E0	6480 NOKEY	RTS	AND RETURN
9300-	6490 TABLE	.DS 512	
	6500	.EN	

DE BASIC-COMPRESSOR

- SPATIES WORDEN VERWYDERD
- LET WORDT VERWYDERD
- ZET ZOVEEL MOGELYK "DATA" IN EEN DATA STATEMENT
- ZET ZOVEEL MOGELYK "DIM'S" IN EEN STATEMENT
- VERWYDERD ALLE LEGE REGELS ZOALS

100 :::::

- GOOI ALLE "REM" STATEMENTS WEG
- ZET ZOVEEL MOGELIJK STATEMENTS ACHTER EEN REGELNUMMER
- KORT ALLE VARIABELE-NAMEN IN TOT MAXIMAAL 2 KARAKTERS

EEN GECOMPRIMEERD PROGRAMMA MAG U NIET PROBEREN TE VERANDEREN
BRENG DE WYZIGINGEN AAN IN HET ORGINEEL EN COMPRIMEER OPNIEUW !

VOORWAARDEN:

- WEES VOORZICHTIG MET HET WEGLATEN VAN HET LAATSTE AANHALINGS-
TEKEN ACHTER EEN STRING, VOORAL IN HET BEGIN VAN EEN PROGRAMMA

100 PRINT "STRING" (GOED)
110 PRINT "TEKST" (FOUT)

- GOTO'S OF GOSUB'S NAAR EEN REM-STATEMENT OF LEGE REGEL
ZYN VERBODEN. ALS EEN GECOMPRIMEERDE VERSIE NIET WERKT IS
KAN ALLEEN DAT DE OORZAAK ZYN !

BUG GEVONDEN ? GRAAG BERICHT !

OVERIGENS: PROBEER EENS MEER PROGRAMMA'S TE SCHRYVEN VOOR DE
.I.P.V. JUNIOR, PET, APPLE, KIM ETC. U HEEFT HIER EEN VOORBEELD
HOE DAT KAN, VOORAL ALS U DE ASSEMBLER "ASSM/TED" VAN C. MOSER
HEEFT GEKOCHT.

Gebruik van de VIA 6522,
door Willem Kuitems

De adressen van de VIA zijn te vinden in het stukje programma wat ik als bijlage heb ontwikkeld.

Ter verduidelijking; de VIA is op veel meer adressen te bereiken, evenals de PIA, maar ik vond het makkelijker om op 1800 te beginnen.

SYSTEEM RESET

Tijdens een reset, worden alle registers van de VIA op 00000000 gezet, alle I/O lijnen tot ingang verklaard en alle interrupt vlaggen gecleared.

DATA-TRANSPORT LIJNEN

Voor het gebruik van de datalijnen gelden de zelfde regels als voor de PIA:

- Een 0 in het DD-Register; bijbehorende lijn ingang.
- Een 1 in het DD-Register; bijbehorende lijn uitgang.

Een bijzonderheid: Poort B is gebufferd zodat de inhoud ervan correct gelezen kan worden, ook wanneer deze poort als uitgang wordt gebruikt.

Poort A is niet gebufferd zodat z'n inhoud alleen correct gelezen kan worden indien de belasting niet te groot is of wanneer de lijn tot ingang is verklaard.

HET AUXILIARY CONTROL REGISTER (figuur 2)

Indien datalijnen tot ingang zijn verklaard, bestaat de keuze van het 'latchen' van de poorten A of B, m.b.v. het ACR. Dit is handig indien men op het moment dat data aanwezig is niet direct wil of kan lezen, omdat de data aanwezig blijft. Dit wordt dus gedaan met de bittens b0 en b1 van het ACR.

HET PERIPHERAL CONTROL REGISTER (figuur 3)

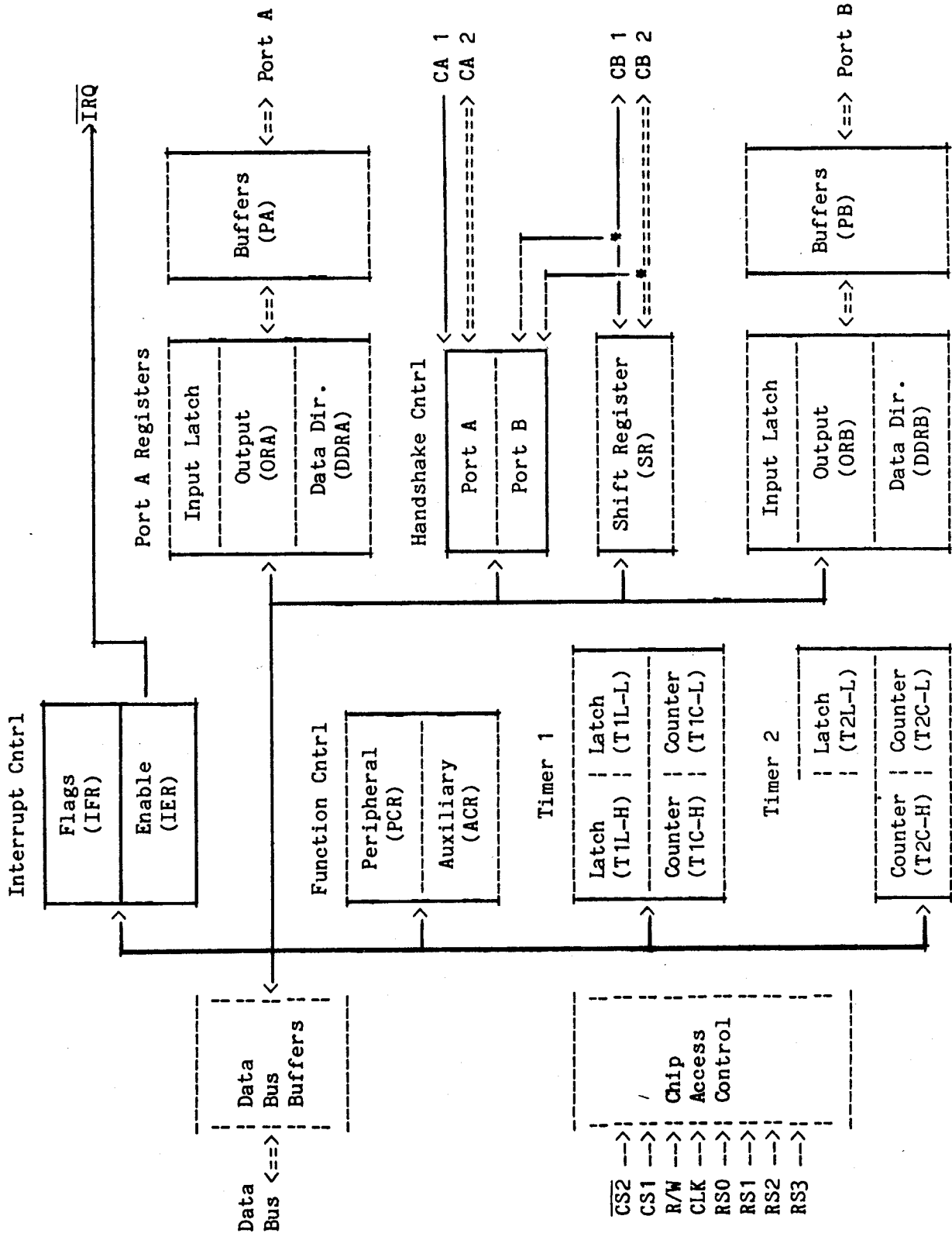
De meeste I/O poort functies worden bestuurd vanuit het PCR. Zo worden de controle lijnen CA1 en CA2 bestuurd met de bittens 0 t/m 3 van het PCR en de controle lijnen CB1 en CB2 met de bittens 4 t/m 7 van het PCR. De betekenis van de waarden van de bittens 0 t/m 3 is gelijk aan die van de bittens 4 t/m 7. LET OP: CA1 en CB1 zijn altijd ingang, terwijl CA2 en CB2 zowel ingang als uitgang kunnen zijn.

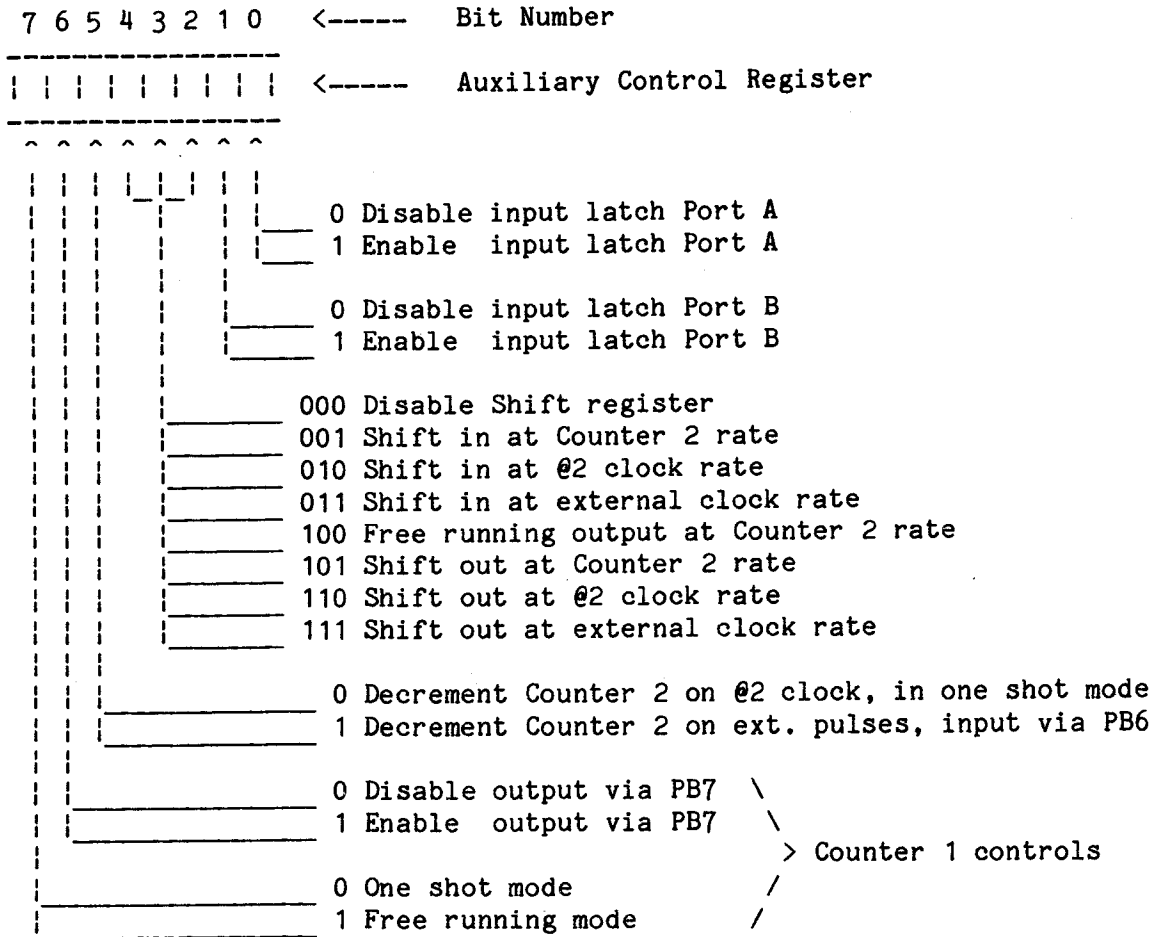
HET INTERRUPT FLAG REGISTER (figuur 4)

In dit register worden bittens geset of gecleared overeenkomstig de gebruikte onderdelen van de VIA.

Waarom ik in het kort maar een paar aspecten van de VIA heb genoemd, zal blijken uit het stukje programmatuur dat ik ervoor heb gemaakt.

Het gaat hierom: we willen een 8 bits parallel data-transport plegen, waarbij we tevens een strobe-puls aanbieden als "Data Available" signaal. Wanneer de data ingelezen is, laten we zien wat de hexa-decimale waarde is en daarnaast printen we de afzonderlijke bits van die waarde. Een aangepast programma kan bijvoorbeeld gebruikt worden om input te plegen vanaf een keyboard.



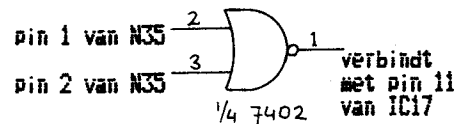


lees verder op volgende pagina !

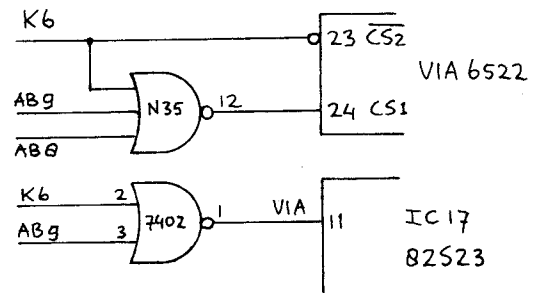
JUNIOR en de dubbeladressering VIA 6522

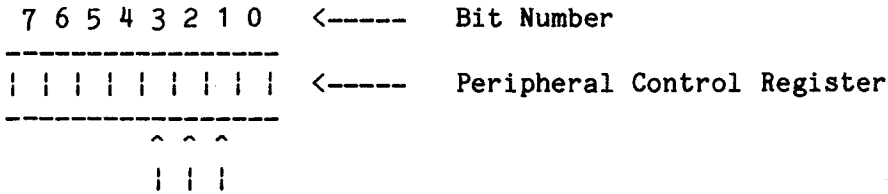
Koen van Nieuwenhove zond de redactie een oplossing voor dit probleem. Het verst een ingreep die er als volgt uitziet:

1. snijdt de baan door die pin 13 van N35 (op de interfacekaart) met GND verbindt;
2. verbindt pin 13 van N35 met pin 9 van IC9;
3. snijdt baan door die pin 12 van N35 verbindt met pin 11 van IC17;
4. snijdt baan door die pin 11 van IC17 verbindt met pin 24 van de VIA;
5. verbindt pin 12 van N35 met pin 24 van de VIA;
6. maak de volgende schakeling:



het wordt dus:





3

- 0 0 0 Input mode - Set CA2 Flag (IFR bit 0) on a high to low CA2 input signal. Clear IFR bit 0 on reading or writing the Port A Output register.

- 0 0 1 Independent Interrupt Input mode - Set IFR bit 0 on a high to low input. Reading or writing the Port A Output register will not clear IFR bit 0.

- 0 1 0 Input mode - Set IFR bit 0 on a low to high input. Clear IFR bit 0 on reading or writing the Port A Output register.

- 0 1 1 Independent Interrupt Input mode - Set IFR bit 0 on a low to high input. Reading or writing the Port A Output register will not clear IFR bit 0.

- 1 0 0 Handshake Output mode - Set CA2 output low on a read or write of the Port A Output register. Reset CA2 output high with an active pulse on CA1

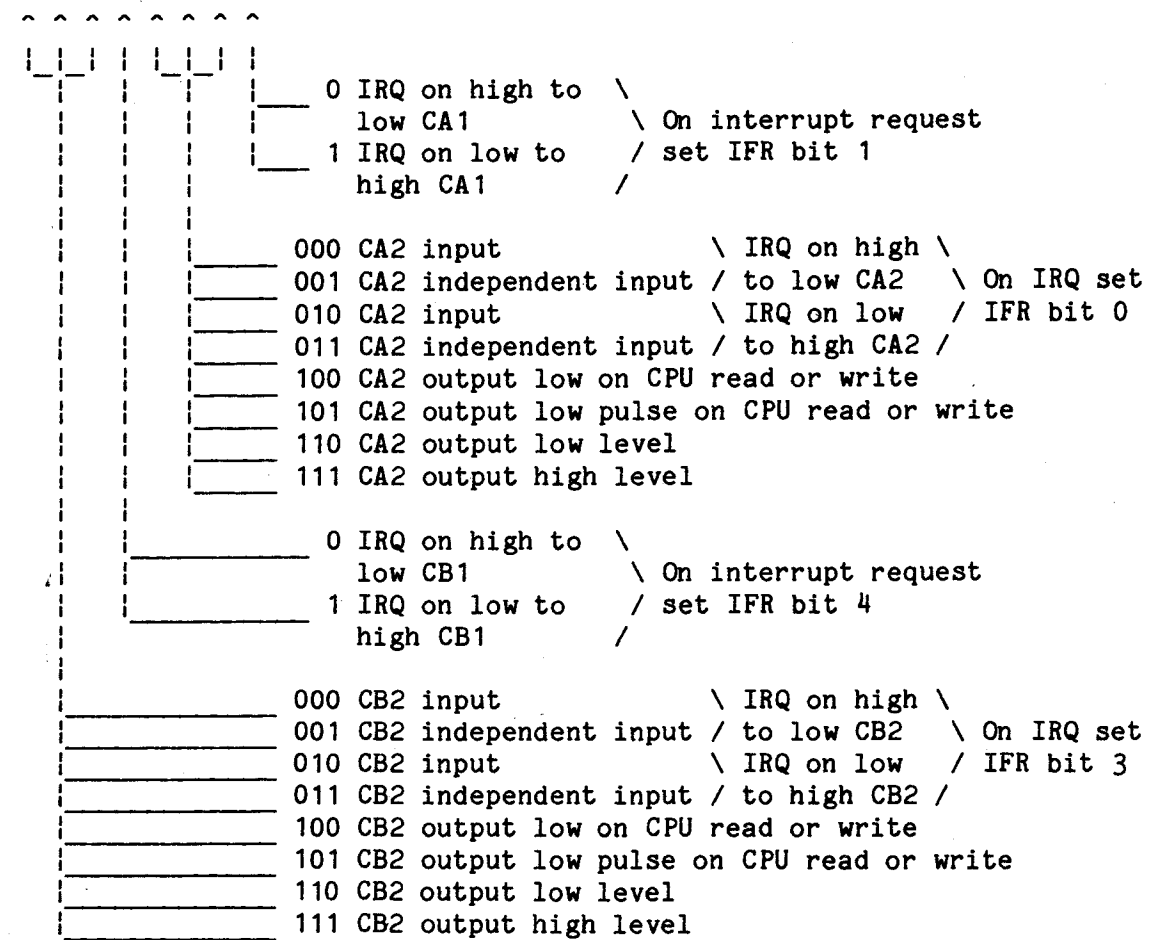
- 1 0 1 Pulse Output mode - CA2 goes low for one cycle after a read or write of the Port A Output register.

- 1 1 0 Manual Output mode - CA2 output as a low level.

- 1 1 1 Manual Output mode - CA2 output as a high level.

7 6 5 4 3 2 1 0 <----- Bit Number

| | | | | | | | <----- Peripheral Control Register

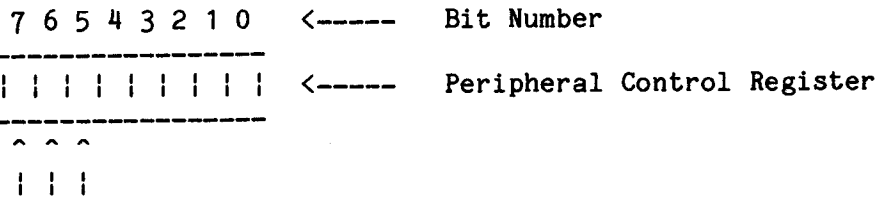


lees verder op volgende pagina !

BUGS in Zenuwslag

(zie DE 6502 KENNER 18, p. 32)

1600	029E	85 FB	moet zijn:	1600	029E	85 F9	STA INH
1710	02B4	84 D7	moet zijn:	1710	02B4	EA EA	2 x NOP
1730	02B9	C6 D7	moet zijn:	1730	02B9	C6 D9	DEC PLCOU



- 0 0 0 Interrupt Input mode - Set CB2 Flag (IFR bit 3) on a high to low CB2 input signal. Clear IFR bit 3 on reading or writing the Port B Output register.
- 0 0 1 Independent Interrupt Input mode - Set IFR bit 3 on a high to low input. Reading or writing the Port B Output register will not clear IFR bit 3.
- 0 1 0 Input mode - Set IFR bit 3 on a low to high input. Clear IFR bit 3 on reading or writing the Port B Output register.
- 0 1 1 Independent Input mode - Set IFR bit 3 on a low to high input. Reading or writing the Port B Output register will not clear IFR bit 3.
- 1 0 0 Handshake Output mode - Set CB2 output low on a write ORB. Reset CB2 output high with an active pulse on CB1 input signal.
- 1 0 1 Pulse Output mode - CB2 goes low for one cycle after a write ORB operation.
- 1 1 0 Manual Output mode - CB2 output as a low level.
- 1 1 1 Manual Output mode - CB2 output as a high level.

lees verder op volgende pagina !

```

=====
10 INPUTA$:A$=A$+" "
20 FORI=PEEK(123)+256*PEEK(124)TOPEEK(125)+256*PEEK(126)
30 IFPEEK(I)=46THENT=T+1:POKEI,ASC(MID$(A$,T,1))
40 IFT=LEN(A$)THEN100
50 NEXTI
60 DATA .....
70 DATA .....
80 DATA .....
90 DATA .....
100 T=0:GOTO10

```

OK

DEZE KLEINE ROUTINE IS BEDOELD OM IN BASIC DOOR HET PROGRAMMA ZELF DATA-REGELS TE LATEN MAKEN. WERKGEBIED VAN DE BASIC LOOPT VAN \$2000 TOT \$C000. AUTEUR: M.R. VAN HINTUM TE DOETINCHEM.

6502

INFORMATIE

7	6	5	4	3	2	1	0	<--- Bit Number
IRQ	T1	T2	CB1	CB2	SR	CA1	CA2	<--- Interrupt Flag Register 4

Bit	Set By	Cleared By
0	Active change of CA2	Read or write ORA
1	Active change of CA1	Read or write ORA
2	Completion 8 shifts	Read or write SR
3	Active change of CB2	Read or write ORB
4	Active change of CB1	Read or write ORB
5	Time-out of Timer 2	Read T2L or write T2H
6	Time-out of Timer 1	Read T1L or write T1LH
7	Active & enabled cond.	Active clear condition

Bits 0,1,3 & 4 are I/O handshake signals. Bit 7 (IRQ) is 1 if any of the interrupts is both active & enabled.

7	6	5	4	3	2	1	0	<--- Bit Number
SCC	T1	T2	CB1	CB2	SR	CA1	CA2	<--- Interrupt Enable Register

Designated interrupt enable
 1 - Interrupt enabled
 0 - Interrupt disabled

Set Clear Control
 Set or clear bit 0 - 6
 1 - Writing 1 sets bit to 1
 0 - Writing 1 sets bit to 0
 Writing 0 to any of bits 0 - 6 has no effect

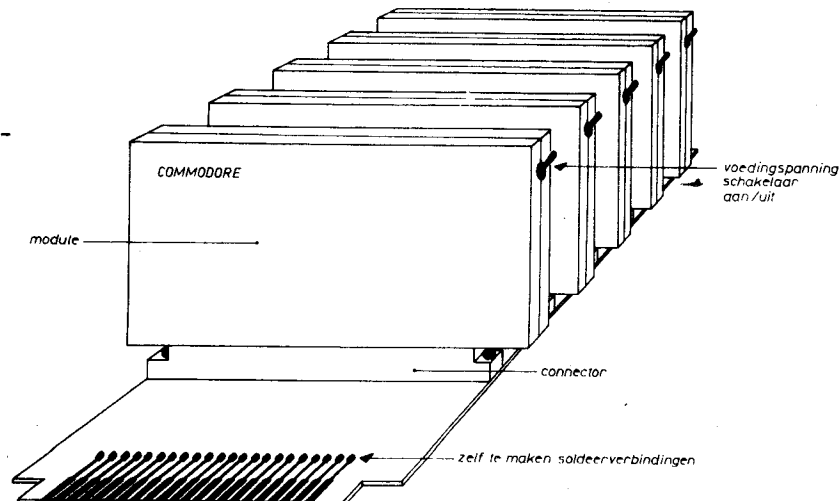
;ADRESSEN VAN DE 6522 PLUS HUN MNEMONICS

DRB	=	1800	;OUTPUT REGISTER FOR I/O PORT B
DRAH	=	1801	;OUTPUT REGISTER FOR I/O PORT A ; WITH HANDSHAKE
DDRB	=	1802	;I/O PORT B DATA DIRECTION REGISTER
DDRA	=	1803	;I/O PORT A DATA DIRECTION REGISTER
T1L	=	1804	;READ TIMER 1 COUNTER LOW-ORDER BYTE ; WRITE TO TIMER 1 LATCH LOW-ORDER BYTE
T1CH	=	1805	;READ TIMER 1 COUNTER HIGH-ORDER BYTE ; WRITE TO TIMER 1 LATCH HIGH-ORDER BYTE ; AND INITIATE COUNT
T1LL	=	1806	;ACCESS TIMER 1 LATCH LOW-ORDER BYTE
T1LH	=	1807	;ACCESS TIMER 1 LATCH HIGH-ORDER BYTE
T2L	=	1808	;READ LOW-ORDER BYTE OF TIMER 2 AND RESET ; COUNTER INTERRUPT ; WRITE TO LOW-ORDER BYTE OF TIMER 2 BUT ; DO NOT RESET INTERRUPT
T2H	=	1809	;ACCESS HIGH-ORDER BYTE OF TIMER 2; RESET ; COUNTER INTERRUPT ON WRITE
SR	=	180A	;SERIAL I/O SHIFT REGISTER
ACR	=	180B	;AUXILIARY CONTROL REGISTER
PCR	=	180C	;PERIPHERAL CONTROL REGISTER
IFR	=	180D	;INTERRUPT FLAG REGISTER
IER	=	180E	;INTERRUPT ENABLE REGISTER
DRA	=	180F	;OUTPUT REGISTER FOR I/O PORT A; ; NO HANDSHAKE

lees verder op volgende pagina !

VIC MODULEBORDJE

Voor het plaatsen van standaard ROM spelletjes, Superexpander, RAM uitbreidingen, enz. zijn allerlei dure uitbreidingen in de handel. De Pet Benelux Exchange biedt nu een uitbreidingsbordje met een vijftal connectoren, waarin de modules worden geplaatst. Omdat alle modules nauwelijks vermogen vragen, is GEEN extra voeding nodig! Verder wordt het bordje kant en klaar geleverd en kan dit zo achterin de VIC gestoken worden. Wanneer U beschikt over modules, die niet tegelijkertijd actief mogen zijn, dan is het vrij eenvoudig om zelf op de betreffende modules een voedingsspanningsschakelaar aan te brengen, door het 5V koperbaantje op de print te onderbreken (deze oplossing voldoet ook uitstekend in de dure Arfon uitbreiding!), zodat U ze nooit meer hoeft te verwijderen. Het PBE modulebordje kost Fl. 250,- (incl. BTW en verzendkosten) en is te bestellen via giro 4088944 t.n.v. Copytronics, Deventer.



```

1      ;GEBRUIKTE LIJNEN: PA0 T/M PA7   ALS DATA-INPUT
2      ;                                CA1   ALS STROBE-INPUT
3
4      PREG      =      OOF1      ;STATUS REGISTER
5      CRLF     =      11E8     ;PRINT CARRIAGE RETURN/ LINEFEED
6      LF       =      11ED     ;PRINT ENKELE LINEFEED
7      PRSP     =      11F3     ;PRINT EEN SPACE
8      SHOWPR  =      1228     ;PRINT PREG
9      PRBYT   =      128F     ;PRINT BYTE UIT ACCUMULATOR
10     RECCHA  =      12AE     ;WACHT VOOR ASCII INPUT
11     PRCHA   =      1334     ;PRINT ASCII KARAKTER UIT ACCU
12     ;VIA ADRESSEN
13     DRAH    =      1801     ;OUTPUT REGISTER VOOR I/O POORT A
14     ;      MET HANDSHAKE
15     DDRA    =      1803     ;I/O POORT A DATA DIRECTION REGISTER
16     ACR     =      180B     ;AUXILIARY CONTROL REGISTER
17     PCR     =      180C     ;PERIPHERAL CONTROL REGISTER
18     IFR     =      180D     ;INTERRUPT FLAG REGISTER
19
20     0200          .LOC      0200
21
22     0200  20  E8  11  KEYBIN: JSR      CRLF      ;RETURN/LINEFEED
23     0203  20  ED  11          JSR      LF        ;GENERATE LINEFEED
24     0206  A9  00          LDA      #0        ;CA1 INPUT "DATA AVAILABLE" PULS
25     0208  8D  0C  18          STA      PCR      ;MOET VAN HOOG NAAR LAAG GAAN
27     020B  8D  03  18          STA      DDRA     ;POORT A LIJNEN INPUT
28     020E  A9  01          LDA      #1
29     0210  8D  0B  18          STA      ACR      ;ENABLE LATCHING VAN POORT A
30     0213  A9  3F          INBYT: LDA      #3F     ;LAAT "? " ZIEN ALS
31     0215  20  34  13          JSR      PRCHA     ;PROGRAMMA KLAAR
32     0218  20  F3  11          JSR      PRSP     ;IS VOOR INPUT
33     021B  A9  02          LDA      #2        ;CHECK CA1 STATUS
34     021D  2C  0D  18  AVAIL: BIT      IFR      ;STROBE OP CA1?
35     0220  F0  FB          BEQ      AVAIL     ;ZO NIET, WACHT TOT ER EEN IS
36     0222  AD  01  18          LDA      DRAH     ;HAAL DATA-BYTE, CLEAR IFR-BIT 1
37     0225  49  FF          EOR      #OFF     ;KEER ALLE BITS OM (OPTIONAL)
38     0227  85  F1          STA      PREG     ;ACCU IN PREG
39     0229  20  8F  12          JSR      PRBYT    ;PRINT HEX-WAARDE
40     022C  20  F3  11          JSR      PRSP     ;DAN EEN SPATIE
41     022F  20  28  12          JSR      SHOWPR   ;DAN DE AFZONDERLIJKE BITS
42     0232  20  E8  11          JSR      CRLF     ;NIEUWE REGEL
43     0235  4C  13  02          JMP      INBYT    ;WACHT OP VOLGENDE STROBE
44
45     .END      KEYBIN

```

De overige figuren zijn ter informatie. In een volgende 6502-KENNER zal ik trachten iets over ander registers te vertellen en met voorbeelden te verduidelijken. Suggesties zijn natuurlijk van harte welkom bij:

W.J. KUITEMS
Dr. NEHER LAB PTT
LEIDSCHENDAM
070-755197

W.J. KUITEMS
ANTHEUNISSTR. 195
DEN HAAG
070-992661

SPIRAA

KIM-1

```

0010: *****
0020: *   PROGRAMMA SPIRAAL   *
0030: *   AUTEUR:  A.W. DEN HARTOG   *
0040: *                   TIENVOET 2   *
0050: *                   3261 TP OUD BEYERLAND   *
0060: *****
0070:
0080: PROGRAMMA BESCHRIJVING:
0090:
0100: Spiraal is een reactiespel voor de KIM.
0110: Het idee achter het spel is afkomstig van
0120: een KIM-achtige leercomputer van Hewlett
0130: Packard met de 8085.
0140: De twee spelers reageren met de 0, en de
0150: 3 toets van het KIM keyboard.
0160: Als het spel wordt gestart op $0200 wordt een
0170: spiraal opgebouwd op het KIM display
0180: van rechtsboven naar linksonder.
0190: Zodra de spiraal af is blijft hij enige
0200: tijd staan. In deze tijd moet de speler aan
0210: wiens kant de spiraal geëindigd is
0220: reageren (0=linker speler, 3=rechter speler).
0230: Het is de bedoeling dat zo laat mogelijk
0240: wordt gereageerd want dan zal de spiraal,
0250: die afwisselend van RB naar LD en van LD naar
0260: RB wordt getekend, hierna met grotere snelheid
0270: worden getekend als de tegenstander aan de
0280: beurt is. Die heeft het dan moeilijker omdat
0290: de tijd waarin gereageerd kan worden ook kleiner
0300: wordt. Als te laat wordt gereageerd omdat het
0310: te snel gaat of omdat iets te ver is gegaan
0320: in het pesten van de tegenstander krijgt de
0330: tegenstander er een punt bij. Als gereageerd
0340: wordt voor dat de spiraal af is krijgt de
0350: tegenstander er ook een punt bij.
0360: Na elke ronde, dus als iemand op het verkeerde
0370: moment heeft gereageerd, wordt een puntentelling
0380: zichtbaar waarin de letters "dd" als
0390: scheiding fungeren.
0400: De puntentelling is decimaal en kent geen
0410: maximum. Na 99 wordt verder gegaan met 00.
0420: Na elke ronde wordt de spiraal weer met de
0430: langzaamste snelheid geschreven.
0440:
0450: 0200 SPIRAA ORG   $0200
0460:
0470: DEFINITIES:
0480:
0490: DD 00 SCHEID *   $00DD
0500: F9 00 PNTR  *   $00F9
0510: FB 00 PNTL  *   PNTR   +02
0520: E4 00 SNELH *   $00E4
0530: E1 00 SPELER *  $00E1
0540: E0 00 TEMP  *   $00E0
0550: 40 1F ANYKEY *  $1F40
0560: 6A 1F GETKEY *  $1F6A
0570: E8 00 DISP  *   $00E8
0580: 1F 1F SCANDS *  $1F1F
0590: 07 17 TIMER *   $1707
0600: E7 00 COUNT *   $00E7
0610:
0620: HOOFDPROGRAMMA:
0630:
0640: 0200 A9 DD START LD A, M SCHEID INITIALISATIE

```

SPIRAA

0650:	0202	85	FA		STAZ	PNTR	+01	
0660:	0204	A9	FF		LDAIM	\$FF		
0670:	0206	85	E4		STAZ	SNELH	LANGZAAMSTE SNELHEID	
0680:	0208	85	E1		STAZ	SPELER	RECHTER SPELER EERST	
0690:	020A	A9	00		LDAIM	\$00	RESET PUNTEN TELLING	
0700:	020C	85	F9		STAZ	PNTR		
0710:	020E	85	FB		STAZ	PNTL		
0720:	0210	A2	05	NEXT	LDXIM	\$05	MAAK DISPLAY SCHOON	
0730:	0212	A9	00		LDAIM	\$00		
0740:	0214	95	E8	A	STAZX	DISP		
0750:	0216	CA			DEX			
0760:	0217	10	FB		BPL	A		
0770:	0219	A5	E1		LDAZ	SPELER	ANDERE SPELER AAN BEURT	
0780:	021B	49	FF		EDRIM	\$FF		
0790:	021D	85	E1		STAZ	SPELER		
0800:	021F	38			SEC		ZET COUNT OP BEGINWAARDE	
0810:	0220	D0	02		BNE	U	SPELER NOG IN ACCU	
0820:	0222	E9	15		SBCIM	\$15		
0830:	0224	E9	EB	U	SBCIM	\$EB		
0840:	0226	85	E7		STAZ	COUNT	COUNT=FF OF 14	
0850:	0228	A5	E4	STREEP	LDAZ	SNELH	INIT TIMER MET SNELH.	
0860:	022A	8D	07	17	STA	TIMER		
0870:	022D	20	A2	02	JSR	CARLO	WACHT EN SCHRIJF DISP	
0880:	0230	AD	07	17	LDA	TIMER		
0890:	0233	10	FB		BPL	B		
0900:	0235	A5	E1		LDAZ	SPELER	VOLGENDE SEGMENT	
0910:	0237	D0	04		BNE	C		
0920:	0239	E6	E7		INCZ	COUNT		
0930:	023B	90	02		BCC	D		
0940:	023D	C6	E7	C	DECZ	COUNT		
0950:	023F	A4	E7	D	LDYZ	COUNT	MAAK SEGMENTBIT HOOG	
0960:	0241	B5	CF	02	LDXAY	TBLDIS		
0970:	0244	B5	E8		LDAZX	DISP		
0980:	0246	19	E3	02	DRAAY	TBLMSK		
0990:	0249	95	E8		STAZX	DISP		
1000:	024B	A5	E7		LDAZ	COUNT	IS DE HELE SPIRAAL GESCHREVEN ?	
1010:	024D	18			CLC			
1020:	024E	65	E1		ADCZ	SPELER		
1030:	0250	C9	FF		CMPIM	\$FF		
1040:	0252	F0	06		BEQ	REAKTI		
1050:	0254	C9	13		CMPIM	\$13		
1060:	0256	F0	02		BEQ	REAKTI		
1070:	0258	D0	CE		BNE	STREEP		
1080:	025A	20	BD	02	JSR	REAKT	WORDT ER GEDRUKT ?	
1090:	025D	F0	16		BEQ	AC	NU AL ? DAN VALS GESPEELD	
1100:	025F	20	A2	02	JSR	CARLO	LEES SPIRAAL UIT	
1110:	0262	A9	FF		LDAIM	\$FF	GEEF EEN KLEINE DELAY	
1120:	0264	85	E0		STAZ	TEMP		
1130:	0266	C6	E0	AA	DECZ	TEMP		
1140:	0268	D0	FC		BNE	AA		
1150:	026A	C6	E4		DECZ	SNELH	DEZE REGEL IS DE GEIN VAN HET SPEL	
1160:	026C	F0	07		BEQ	AC		
1170:	026E	20	BD	02	JSR	REAKT		
1180:	0271	D0	EC		BNE	WACHT		
1190:	0273	F0	9B		BEQ	NEXT		
1200:	0275	A2	00	AC	LDXIM	\$00	VERHOOG SCORE VAN TEGENSTANDER	
1210:	0277	A5	E1		LDAZ	SPELER		
1220:	0279	F0	02		BEQ	R		
1230:	027B	E8			INX			
1240:	027C	E8			INX			
1250:	027D	B5	F9	R	LDAZX	PNTR		
1260:	027F	F8			SED			
1270:	0280	18			CLC			
1280:	0281	69	01		ADCIM	\$01		

```

1290: 0283 95 F9          STAZX PNTR
1300: 0285 D8          CLD
1310: 0286 A9 06          LDAIM $06      LEES PUNTENSTAND UIT
1320: 0288 85 E0          STAZ  TEMP
1330: 028A A9 FF          LDAIM $FF
1340: 028C 8D 07 17      T          STA  TIMER
1350: 028F 20 1F 1F      S          JSR  SCANDS
1360: 0292 AD 07 17          LDA  TIMER
1370: 0295 10 F8          BPL  S
1380: 0297 C6 E0          DECZ TEMP
1390: 0299 D0 EF          BNE  T
1400: 029B A9 FF          LDAIM $FF      NIEUWE RONDE BEGINT MET LANGZAAMSTE
1410: 029D 85 E4          STAZ SNELH     SNELHEID
1420: 029F 4C 10 02          JMP  NEXT
1430:
1440:
1450:
1460: 02A2 A0 06          CARLD LDYIM $06      ZET DISPLAYVELD BITLIJK
1470: 02A4 20 88 1E          JSR  $1E88     (VGL. LETTERLIJK) IN KIM DISPLAY
1480: 02A7 A9 7F          LDAIM $7F
1490: 02A9 8D 41 17          STA  $1741     DEZE ROUTINE IS HET WERK VAN
1500: 02AC A2 09          LDXIM $09      C.T.A. TOTTE UIT ZWIJNDRECHT
1510: 02AE 84 FC          LDEP STYZ  $FC
1520: 02B0 B9 E7 00          LDAAY $00E7
1530: 02B3 20 4E 1F          JSR  $1F4E
1540: 02B6 88          DEY
1550: 02B7 D0 F5          BNE  LDEP
1560: 02B9 8C 40 17          STY  $1740
1570: 02BC 60          RTS
1580:
1590: 02BD A5 E1          REAKT LDAZ  SPELER  HAAL AFH. VAN SPELER DE 0 OF 3 TOETS
1600: 02BF 18          CLC                                     OP EN SET DE ZEROFLAG INDIEN INGEDRUKT
1610: 02C0 F0 02          BEQ  L
1620: 02C2 69 04          ADCIM $04
1630: 02C4 85 E0          L          STAZ  TEMP
1640: 02C6 20 40 1F          JSR  ANYKEY
1650: 02C9 20 6A 1F          JSR  GETKEY
1660: 02CC C5 E0          CMPZ TEMP
1670: 02CE 60          RTS
1680:
1690:
1700:
1710: 02CF 00          TBLDIS = $00
1720: 02D0 01          = $01
1730: 02D1 02          = $02
1740: 02D2 03          = $03
1750: 02D3 04          = $04
1760: 02D4 05          = $05
1770: 02D5 05          = $05
1780: 02D6 05          = $05
1790: 02D7 04          = $04
1800: 02D8 03          = $03
1810: 02D9 02          = $02
1820: 02DA 01          = $01
1830: 02DB 00          = $00
1840: 02DC 00          = $00
1850: 02DD 00          = $00
1860: 02DE 01          = $01
1870: 02DF 02          = $02
1880: 02E0 03          = $03
1890: 02E1 04          = $04
1900: 02E2 05          = $05
1910:
1920: 02E3 01          TBLMSK = $01

```

SPIRAA

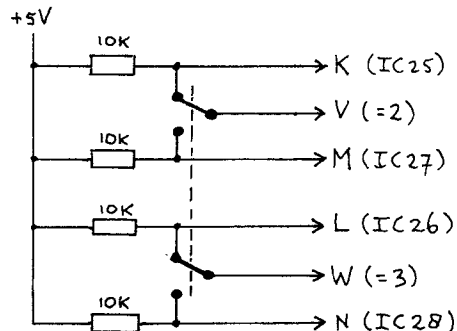
1930:	02E4	01	=	\$01
1940:	02E5	01	=	\$01
1950:	02E6	01	=	\$01
1960:	02E7	01	=	\$01
1970:	02E8	01	=	\$01
1980:	02E9	20	=	\$20
1990:	02EA	40	=	\$40
2000:	02EB	40	=	\$40
2010:	02EC	40	=	\$40
2020:	02ED	40	=	\$40
2030:	02EE	40	=	\$40
2040:	02EF	40	=	\$40
2050:	02F0	04	=	\$04
2060:	02F1	08	=	\$08
2070:	02F2	08	=	\$08
2080:	02F3	08	=	\$08
2090:	02F4	08	=	\$08
2100:	02F5	08	=	\$08
2110:	02F6	08	=	\$08
2120:			=	

HARDWARE

ELEKTUUR RAM-EPROM KAART.

HANS CHRISTEN, POSTBUS 50662, 3007 JD ROTTERDAM 010-851481

- (1) STEL DE KAART IN OP HET GEBRUIK VAN DE 2732.
- (2) SELEKTEER 8K \$2000 - \$4000
IC 5 PIN 2 => V
IC 5 PIN 3 => W
- (3) VERWIJDER IC 7.
- (4) LEG DE PUNTEN K, L, M EN N ELK MET EEN WEERSTAND VAN 10K AAN + 5V.
IC 25 T/M 28 ZIJN NU DESELECTED.
- (5) NEEM EEN DUBBELPOLIGE WISSELSCHAKELAAR EN MAAK DE VOLGENDE AANSLUITINGEN:



- (6) PLAATS NU IN BIJV. 25 EN 26 BASIC OF IETS ANDERS. EN BIJV. 27 EN 28 MICRO-ADE D. I. D.
- (7) MEN HEEFT NU OP EEN ADRESGEBIED VAN 8K TWEE PROGRAMMA'S VAN 8K DIREKT VORHANDEN, SWITCH-SELECTABLE (64K IS TENSLOTTE NIET ONEINDIG!).

AIM 65-BASIC VOOR DE JUNIOR

=====

DOOR : E. J. M. VISSCHEDIJK, DRAKENSTEYN 299,
7608 TR ALMELO. Tel.: 05490 - 71416.

DAT DE INTERESSE VAN DE JUNIOR-BEZITTERS NIET ENKEL OP MACHINETAAL GERICHT IS, IS GEEN NIEUWTJE. HET KLINKT ECHTER WEL NIEUW TE HOREN DAT ER VOOR DE JUNIOR NU AL DRIE BASIC'S BESTAAN, BUITEN TINY BASIC. IN DE 6502 KENNER WERDEN DE AANPASSINGEN VERMELD VAN SYM-BASIC VOOR JUNIOR. MOMENTEEL LOPEN ER AL PUBLIKATIES OM DE KIM-BASIC GESCHIKT TE MAKEN VOOR DE JUNIOR, EN TEVENS UIT TE BREIDEN MET ALLERLEI HANDIGE ROUTINES. THANS IS OOK DE AIM 65-BASIC VOOR DE JUNIOR Aangepast.

DEZE BASIC STAAT HOOG IN HET GEHEUGEN, EN WEL VANAF \$B000 - \$D000. HAAR WERKRUIJTE BEGINT OP ADRES \$2000; BASIC GAAT ERVAN UIT DAT DAAR DE VRIJE RAM BEGINT. DE VERANDERINGEN WELKE AAN DEZE BASIC ZIJN AANGEBRACHT, ZIJN NIET HELEMAAL LOS TE DENKEN VAN DE HIERNA TEVENS VOLGENDE ROUTINES M.B.T. TAPE MANAGEMENT, ELEKTERMINAL PAGE CLEAR EN RESET PM START (de laatste staat wel helemaal los van Basic).

TAPE MANAGEMENT BIJ DEZE BASIC WERKT ALS VOLGT:

- PROGRAMMA'S OP TAPE ZETTEN.
TYPE IN: SAVE 'RETURN'. DAN VOLGT DE VRAAG FILENAME: ER KAN NU EEN NAAM VAN MAXIMAAL 126 KARAKTERS WORDEN MEEGEGEVEN. NA DE NAAM VOLGT 'RETURN'. DE CASSETTE-RECORDER MOET DAN KLAAR STAAN, EN BEGINT TE LOPEN. ALS HET PROGRAMMA OP DE BAND STAAT VOLGT DE MELDING READY EN EEN PIEPJE (mits dit piepje hardwarematig aanwezig is met Control G).
 - PROGRAMMA'S LEZEN.
TYPE IN: LOAD 'RETURN'. DE VRAAG FILENAME: KOMT NU WEER EN ER WORDT EEN NAAM OPGEGEVEN VAN EEN PROGRAMMA DAT EERDER OP TAPE WERD GEZET. BIJ IEDER GEVONDEN PROGRAMMA VOLGT: FOUND:
ALS HET GOEDE PROGRAMMA GEVONDEN IS VOLGT LOADING:
ALS HET GEHELE PROGRAMMA IS INGELEZEN VOLGT: READY EN WEDEROM EEN PIEPJE.
- BOVENSTAANDE PROGRAMMATUUR MOET IN TM STAAN !

IN DEZE JUNIOR BASIC ZIJN TWEE SLEUTELWOORDEN VERVALLEN, T.W. NULL EN GET. DAT NULL IS VERVALLEN, DAAR ZAL NIEMAND OM HUILEN. HIERVOOR IN DE PLAATS IS GEKOMEN HET SLEUTELWOORD HOME. DIT HEEFT TOT GEVOLG (ZOWEL IN DIREKTE ALS IN INDIREKTE MODE) DAT HET GEHELE BEELDSCHERM SCHOONGEMAAKT WORDT. DIT ALLEEN, ALS HET MACHINETAALPROGRAMMA 'ELEKTERMINAL PAGE CLEAR' IN DE TM-EPROM IS OPGENOMEN. OP WELKE ADRESSEN DIT STAAT EN HOE DIT PROGRAMMA'TJE ER UITZIET, LATER.

HET SLEUTELWOORD GET IS OOK VERVALLEN, HELAAS. GET SCANT NAMELIJK HET TOETSENBOORD ALS ER EEN BASICPROGRAMMA GEROND WORDT. ECHTER, DE JUNIOR ZIET EEN INGEDRUKTE TOETS IN DE RUN-MODE ALS EEN BREAK (ALTHANS, TIJDENS HET PRINTEN). DEZE BASIC ZIET IN DE RUN-MODE EN NIET TIJDENS HET PRINTEN EEN INGEDRUKTE TOETS OOK ALS EEN BREAK. HET IS DUS DUIDELIJK DAT GET NIET TE GEBRUIKEN IS. HIERVOOR IS MON IN DE PLAATS GEKOMEN. ALS DIT WORDT GEKOZEN, WORDT TERUGGESPRONGEN NAAR JUNIOR PM.

ADRES:	WORDT:	REDE VAN VERANDERING:
B000	4C 50 CF	NA KOUDE START: SCHOON BEELDSCHERM, INPUT & OUTPUT DEVICE WORDEN IN STAND TERMINAL GEZET.
CF50	A9 00	NB: IN DE TM-EPROM MOET HET BEELDSCHERM SCHOONMAAKPROGRAMMA AANWEZIG ZIJN. ANDERS MOET CF5B T/M CF5D DOOR IETS ANDERS VERVANGEN WORDEN. JMP KOUDE START.
CF52	8D 21 02	
CF55	8D 22 02	
CF58	BD 24 02	
CF5B	20 92 0C	
CF5E	4C A3 CE	
CF61 T/M		
CF69		9 X NOP
CFAC	4C 43 CF	AAN HET EINDE VAN KOUDE START WORDT NAAR WARMER START GESPRONGEN. DIT IS EEN NIEUWE W. START.
B003	4C 43 CF	JMP NIEUWE WARMER START.
CF43	A9 58	BREAK ADRES WORDT MET B658 GELADEN.
CF45	8D 7C 1A	DAAR WORDT HEENGESPRONGEN NA BREAK.
CF48	A9 B6	OOK ADRESPLAATSEN B640, B646, B64B ZIJN NU VERANDERD.
CF4A	8D 7D 1A	
CF 4D	4C 7F B2	
B640	AD 80 1A	
B646	8D 80 1A	
B64B	AD 80 1A	
B6D2	20 E8 11	ZODAT ER EEN CRLF KOMT OOK IN TERMINAL MODE. OP 0222 STAAT HET OUTPUT-DEVICE. OP 0221 STAAT HET INPUT-DEVICE
CEFB T/M		
CF00		6 X NOP
CEED	4C 01 CF	OMZEILEN V.D. VRAAG "MEMORY SIZE" ER VOLGT NU ALTIJD EEN RAM-TEST NA EEN KOUDE START.
CF01	A9 06	LB EERSTE
CF03	A0 20	HB VRIJE
CF70	A2 06	LB RAM
CF72	A0 20	HB ADRES
CF40	4C 6A CF	OMZEILEN VAN DE VRAAG "WIDTH" BREEDE BEELDSCHERM VAST OP 63
CF6A	A9 3F	OM TE VOORKOMEN DAT EN BASIC EN DE ELEKTERMINAL BEIDE EEN LF GEVEN.
CF6C	85 12	
CF6E	85 13	
B359	20 F0 CE	MOET NAAR EEN RUB-OUT SUBR. WIJZEN.
CEFO	A9 08	DEZE SUBR. GEEFT EEN BACKSPACE OP DE TERMINAL.
CEF2	4C 34 13	
B9C3	20 F5 CE	WEES NAAR ONDERSTAANDE SUBR., DIE OORSPRONKELIJK ELDERS STOND.
CEFS	20 C5 00	
CEFS	4C 9D B8	
B6B6	4C AE 12	INPUT
B6C3	4C 34 13	OUTPUT
B6CF	4C 34 13	OUTPUT
B6BA	AD 22 02	OUTPUT DEVICE (DEFAULT = 00)
B6AB	AD 21 02	INPUT DEVICE (DEFAULT = 00)
B384	C9 7F	OM OOK DE KARAKTERS 7D, 7E DOOR TE LATEN

ADRES:	WORDT:	REDE VAN VERANDERING:
B9BE	AD 24 02	WAS BIJ AIM-65 ENABLE OF DISABLE PRINTER. 80 IS ON. 00 IS OFF. BIJ 'PRINT!' 'TEST' IS TIJDENS HET RINNEN \$0224 EVEN 80. DIT ALS GEVOLG VAN !
B9A9	AD 24 02	
B9A3	8D 24 02	
B9B3	8D 24 02	
B36E	EA EA EA	DIT WAREN ALLEMAAL ADRESSEN DIE NAAR AIM-65 ROUTINES WEZEN OF NAAR ANDERE TYPISCHE AIM-65 POINTERS O.I.D. DEZE WORDEN ALLE NIET OF MOGEN NIET GE- BRUIKT WORDEN, VANDAAR EA EN EEN KEER 60.
B536	EA EA EA	
B650	EA EA EA	
B655	20 AE 12	
B6A0	EA EA EA	
B6B3	EA EA EA	
B6E3	EA EA EA	
B6E9	60 EA EA	
B02C	91 0C	PAGE-CLEAR -1 SUB (HOME) TAPE MANAGEMENT LOAD -1 (LOAD) TAPE MANAGEMENT SAVE -1 (SAVE) JUNIOR PM -1 (MON) ALLES -1 VANWEGE RTS NA PHA, PHA
B030	FD 0C	
B032	A1 0C	
B040	5E 10	
B0D1	48 4F 4D C5	'HOME' MET MSB=1 VAN DE LAATSTE LETTER.
B0FA	4D 4F CE	'MON' MET MSB=1 VAN DE LAATSTE LETTER.

PAGECL

E. J. M. VISSCHEDIJK ALMELO

PAGE 01

```

0010: ELEKTERMINAL PAGE CLEAR
0020: =====
0030:
0040: E. J. M. VISSCHEDIJK, DRAKENSTEYN 299, 7608 TR ALMELO.
0050:
0060: ALS MEN DE ASCII-CODE $0C NAAR DE ELEKTERMINAL ZENDT,
0070: ZAL DEZE HET GEHELE BEELDSCHERM SCHOONMAKEN. DIT GAAT
0080: ECHTER ALLEEN OP ALS DE ELEKTERMINAL HIERVOOR 132 ms
0090: DE TIJD WORDT GEGUND. DE ONDERSTAANDE SUBROUTINE
0100: ZENDT $0C NAAR DE VIDEOKAART, EN NEEMT DAN 132 ms
0110: PAUZE IN ACHT, ZODAT EEN EVENTUEEL VOLGEND KARAKTER
0120: NOOIT DE PAGE-CLEAR TIJD ZAL ONDERBREKEN. OOK DIT
0130: PROGRAMMA WORDT IN HET VRIJE STUK IN 'TM' GEZET.
0140: ALS DE JUNIOR-BASIC (OORSPR. AIM65) GEBRUIKT WORDT,
0150: MOET HET ZELFS HIER STAAN, OMDAT DEZE BASIC HIER
0160: NAARTOE SPRINGT.
0170:
0180: 0C92 PAGECL ORG $0C92
0190:
0200: 34 13 PRCHA * $1334
0210:
0220: 0C92 A9 0C LDAIM $0C SEND $0C TO VIDEOCARD
0230: 0C94 20 34 13 JSR PRCHA
0240: 0C97 A2 80 LDXIM $80 DELAY
0250: 0C99 A0 FF LDYIM $FF 132
0260: 0C9B 88 DEY ms
0270: 0C9C D0 FD BNE $0C9B
0280: 0C9E CA DEX
0290: 0C9F D0 F8 BNE $0C99
0300: 0CA1 60 RTS RETURN
  
```

6502

PROGRAMMEERTALEN

TAPEUT

E. J. M. VISSCHEDIJK ALMELD

PAGE 01

0010:
0020:
0030:
0040:
0050:
0060:
0070:
0080:
0090:
0100:
0110:
0120:
0130:
0140:
0150:
0160:
0170:
0180:
0190:
0200:
0210:
0220:
0230:
0240:
0250:
0260:
0270:
0280:
0290:
0300:
0310:
0320:
0330:
0340:
0350:
0360:
0370:
0380:
0390:
0400:
0410:
0420:
0430:
0440:
0450:
0460:
0470:
0480:
0490:
0500:
0510:
0520:
0530:
0540:
0550:
0560:
0570:
0580:
0590:
0600:
0610:
0620:
0630:
0640:

0CA2

TAPE MANAGEMENT BIJ AIM65-BASIC VOOR JUNIOR.

=====

```

;TAPE MANAGEMENT BIJ DEZE BASIC WERKT ALS VOLGT:
; - PROGRAMMA'S OP TAPE ZETTEN.
;   TYPE IN 'SAVE' "RETURN". DAN VOLGT DE VRAAG
;   FILENAME:
;   ER KAN NU EEN NAAM VAN MAX. 126 KARAKTERS
;   WORDEN MEEGEGEVEN. NA DE NAAM VOLGT "RETURN".
;   DE CASSETTERECORDER MOET DAN KLAAR STAAN, EN
;   BEGINT DAN TE LOPEN. ALS HET PROGRAMMA OP DE
;   BAND STAAT, VOLGT DE MELDING 'READY' EN EEN
;   PIEPJE.
;
; - PROGRAMMA'S LEZEN.
;   TYPE IN: 'LOAD' "RETURN". DE VRAAG: FILENAME:
;   KOMT NU WEER EN ER WORDT EEN NAAM OPGEGEVEN
;   VAN EEN PROGRAMMA DAT EERDER OP DE BAND WERD
;   GEZET. BIJ IEDER GEVONDEN PROGRAMMA VOLGT:
;   FOUND: .....
;   ALS HET GOEDE PROGRAMMA GEVONDEN IS VOLGT:
;   LOADING: .....
;   ALS HET GEHELE PROGRAMMA IS INGELEZEN VOLGT:
;   'READY' EN WEDEROM EEN PIEPJE.
;
; N. B. DE PROGRAMMATUUR MOET IN 'TM' STAAN!
    
```

TAPEUT ORG \$0CA2

```

MEPNT * $0023
PNTRS * $0073
BUF * $0200
BUF2 * $0280
DUMP * $09DF
RDTAPE * $0E02
CRLF * $11E8
RECCHA * $12AE
PRCHA * $1334
DOTP * $1476
GB * $1496
RESTTY * $14BC
SAL * $1A70
SAH * $1A71
EAL * $1A72
EAH * $1A73
ID * $1A79
AREA * $2000
    
```

= DUMP + 'READY'
= RDTAPE + 'READY'

```

LDXIM $04
LDZAX PNTRS
STAAX AREA
SAVPTR SAVE BASIC POINTERS
DEX
BPL SAVPTR
PRFNM JSR FLNME PRINT : 'FILENAME:'
      LDXIM $00 GET FILENAME
      JSR FFNME
      BCS PRFNM TRY AGAIN AFTER WRONG FILENAME
      JSR CRLF
      INX X=X+1
      STX EAL X=ENDADDRESS+1 (LB)
      LDAIM $01 ID-CODE =01
      STA ID
    
```

6502 PROGRAMMEERTALEN

TAPEUT

E. J. M. VISSCHEDIJK ALMELO

PAGE 02

```

0650: 0CC2 A9 00          LDAIM $00      STARTADDRESS (LB) = 00
0660: 0CC4 8D 70 1A      STA SAL
0670: 0CC7 A9 02          LDAIM $02      STARTADDRESS (HB) = 02
0680: 0CC9 8D 71 1A      STA SAH
0690: 0CCC 8D 73 1A      STA EAH      ENDADDRESS (HB) = 02
0700: 0CCF 20 DF 09      JSR DUMP      PUT FILENAME ON TAPE
0710: 0CD2 20 BC 14      JSR RESTTY
0720: 0CD5 A9 02          LDAIM $02      ID-CODE = 02
0730: 0CD7 8D 79 1A      STA ID
0740: 0CDA A9 00          LDAIM $00      STARTADDRESS (LB) = 00
0750: 0CDC 8D 70 1A      STA SAL
0760: 0CDF A9 20          LDAIM $20      STARTADDRESS (HB) = 20
0770: 0CE1 8D 71 1A      STA SAH
0780: 0CE4 18          CLC
0790: 0CE5 AD 02 20      LDA AREA      +02 ENDA = BASIC ENDADDRESS +1
0800: 0CE8 69 01          ADCIM $01
0810: 0CEA 8D 72 1A      STA EAL      ENDADDRESS (LB)
0820: 0CED AD 03 20      LDA AREA      +03
0830: 0CF0 69 00          ADCIM $00
0840: 0CF2 8D 73 1A      STA EAH      ENDADDRESS (HB)
0850: 0CF5 20 76 14      JSR DOTP      PUT BASIC PROGRAM ON TAPE
0860: 0CF8 A9 07          LDAIM $07
0870: 0CFA 20 34 13      JSR PRCHA     RING THE BELL
0880: 0CFD 60          RTS           RETURN TO BASIC
0890:
0900: 0CFE 20 67 0D      LOAD JSR FLNME PRINT : FILENAME
0910: 0D01 A2 80          LDXIM $80     GET FILENAME
0920: 0D03 20 89 0D      JSR FFNME
0930: 0D06 B0 FE 0D      BCS LOAD     TRY AGAIN AFTER WRONG FILENAME
0940: 0D08 20 E8 11      JSR CRLF
0950: 0D0B A9 01          IDOTP LDAIM $01  ID-CODE = 01
0960: 0D0D 8D 79 1A      STA ID
0970: 0D10 20 02 0B      JSR RDTAPE   GET FILENAME FROM TAPE
0980: 0D13 20 BC 14      JSR RESTTY
0990: 0D16 A2 00          LDXIM $00
1000: 0D18 BD 00 02      COMP LDAAX BUF  COMPARE FILENAME FROM TAPE
1010: 0D1B DD 80 02      CMPAX BUF2   & WANTED FILENAME
1020: 0D1E D0 08          BNE PRFND
1030: 0D20 D9 0D          CMPIM $0D
1040: 0D22 F0 18          BEQ PRLOAD
1050: 0D24 E8          INX
1060: 0D25 4C 18 0D      JMP $0D18
1070: 0D28 20 CF 0D      PRFND JSR PRINT PRINT : FOUND
1080: 0D2B 20          =
1090: 0D2C 4E          = 'F
1100: 0D2D 4F          = 'D
1110: 0D2E 55          = 'U
1120: 0D2F 4E          = 'N
1130: 0D30 44          = 'D
1140: 0D31 3A          = ':'
1150: 0D32 03          = $03
1160: 0D33 20 76 0D      JSR PNAME   PRINT FILENAME FROM TAPE
1170: 0D36 20 E8 11      JSR CRLF
1180: 0D39 4C 0B 0D      JMP IDOTP   GET NEXT FILENAME FROM TAPE
1190: 0D3C 20 CF 0D      PRLOAD JSR PRINT PRINT : LOADING
1200: 0D3F 0A          = $0A      LINEFEED
1210: 0D40 4C          = 'L
1220: 0D41 4F          = 'D
1230: 0D42 41          = 'A
1240: 0D43 44          = 'D
1250: 0D44 49          = 'I
1260: 0D45 4E          = 'N
1270: 0D46 47          = 'G
1280: 0D47 3A          = ':'

```

6502

PROGRAMMEERTALEN

TAPEUT

E. J. M. VISSCHEDIJK ALMELO

PAGE 03

```

1290: 0D48 03          =          $03
1300: 0D49 20 76 0D   JSR        PNAME   PRINT FILENAME FROM TAPE
1310: 0D4C 20 E8 11   JSR        CRLF
1320: 0D4F A9 02      LDAIM     $02      ID-CODE = 02
1330: 0D51 8D 79 1A   STA      ID
1340: 0D54 20 96 14   GBAS     JSR        GB   GET BASIC PROGRAM
1350: 0D57 A2 04      LDXIM     $04
1360: 0D59 8D 00 20   STBPTR  LDAAX     AREA  RESTORE BASIC POINTERS
1370: 0D5C 95 73      STAAX     PNTRS
1380: 0D5E CA        DEX
1390: 0D5F 10 F8     BPL      STBPTR
1400: 0D61 A9 07      LDAIM     $07     RING THE BELL
1410: 0D63 20 34 13   JSR        PRCHA
1420: 0D66 60        RTS
1430: 0D67 20 CF 0D   FLNME   JSR        PRINT  PRINT : FILENAME
1440: 0D6A 20        =
1450: 0D6B 46        =          'F
1460: 0D6C 49        =          'I
1470: 0D6D 4C        =          'L
1480: 0D6E 45        =          'E
1490: 0D6F 4E        =          'N
1500: 0D70 41        =          'A
1510: 0D71 4D        =          'M
1520: 0D72 45        =          'E
1530: 0D73 3A        =          ':'
1540: 0D74 03        =          $03
1550: 0D75 60        RTS
1560: 0D76 A2 00 02   PNAME   LDXIM     $00
1570: 0D78 8D 00 02   PNAME   LDAAX     BUF   PRINT FILENAME GOT FROM TAPE
1580: 0D7B 48        PHA
1590: 0D7C 20 34 13   JSR        PRCHA
1600: 0D7F 68        PLA
1610: 0D80 C9 0D      CMPIM     $0D
1620: 0D82 F0 04      BEQ      RET
1630: 0D84 E8        INX
1640: 0D85 4C 78 0D   JMP      PNAME
1650: 0D88 60        RET
1660: 0D89 20 AE 12   FFNME  JSR        RECCHA  GET FILENAME FROM KEYBOARD
1670: 0D8C C9 7F      CMPIM     $7F
1680: 0D8E F0 1A      BEQ      STORE   STORE FILENAME
1690: 0D90 48        PHA
1700: 0D91 20 34 13   JSR        PRCHA
1710: 0D94 E8        PLA
1720: 0D95 9D 00 02   STAAX     BUF   0200 ..... 027F => IF X = 00 (SAVE)
1730: 0D98 C9 0D      CMPIM     $0D   0280 ..... 02FF => IF X = 80 (LOAD)
1740: 0D9A F0 0C      BEQ      EXIT
1750: 0D9C E8        INX
1760: 0D9D E0 7F      CPXIM     $7F   FILENAME TOO LARGE (IN CASE OF SAVE)
1770: 0D9F F0 12      BEQ      ERROR  ERROR EXIT
1780: 0DA1 E0 FF      CPXIM     $FF   FILENAME TOO LARGE (IN CASE OF LOAD)
1790: 0DA3 F0 0E      BEQ      ERROR  ERROR EXIT
1800: 0DA5 4C 89 0D   JMP      FFNME  GET NEXT CHARACTER
1810: 0DA8 18        CLC
1820: 0DA9 60        RTS
1830: 0DAA CA        STORE  DEX
1840: 0DAB A9 08      LDAIM     $08   PRINT : BACKSPACE
1850: 0DAD 20 34 13   JSR        PRCHA  IF DEL OR RUBOUT WAS DEPRESSED
1860: 0DB0 4C 89 0D   JMP      FFNME
1870: 0DB3 20 CF 0D   ERROR  JSR        PRINT  PRINT : FILENAME TOO LARGE
1880: 0DB6 0A        =          $0A
1890: 0DB7 46        =          'F
1900: 0DB8 49        =          'I
1910: 0DB9 4C        =          'L
1920: 0DBA 45        =          'E

```

6502

PROGRAMMEERTALEN

TAPEUT E. J. M. VISSCHEDIJK ALMELO

PAGE 04

```

1930: 0DBB 4E      =      'N
1940: 0DBC 41      =      'A
1950: 0DBD 4D      =      'M
1960: 0DBE 45      =      'E
1970: 0DBF 20      =      'T
1980: 0DC0 54      =      'O
1990: 0DC1 4F      =      'O
2000: 0DC2 4F      =      'O
2010: 0DC3 20      =      'L
2020: 0DC4 4C      =      'A
2030: 0DC5 41      =      'R
2040: 0DC6 52      =      'R
2050: 0DC7 47      =      'G
2060: 0DC8 45      =      'E
2070: 0DC9 03      =      $03
2080: 0DCA 20      JSR      CRLF
2090: 0DCD 38      SEC
2100: 0DCE 60      RTS
2110:
2120:
2130:
2140:
2150: 0DCF 68      PRINT  PLA      PULL RETURN ADDRESS FROM
2160: 0DD0 85 23      STAZ  MEPNT  STACK AND SAVE IT
2170: 0DD2 68      PLA
2180: 0DD3 85 24      STAZ  MEPNT  +01
2190: 0DD5 E6 23      PRTA  INCZ  MEPNT
2200: 0DD7 D0 02      BNE   PRTB
2210: 0DD9 E6 24      INCZ  MEPNT  +01
2220: 0ddb A0 00      PRTB  LDYIM $00  FETCH CHARACTER AND
2230: 0ddd B1 23      LDAIY MEPNT  PRINT IT
2240: 0ddf C9 03      CMPIM $03  EOT?
2250: 0DE1 F0 06      BEQ   PRTC
2260: 0DE3 20 34 13    JSR   PRCHA
2270: 0DE6 4C D5 0D    JMP   PRTA
2280: 0DE9 A5 24      PRTC  LDZ   MEPNT  +01 PUSH RETURN ADDRESS
2290: 0DEB 48      PHA   ON STACK AND
2300: 0DEC A5 23      LDZ   MEPNT  RETURN
2310: 0DEE 48
2320: 0DEF 60

```

```

;FIND FOLLOWING SUB IN JUNIOR PAPERWARE NR 1
;AN EDITION OF ELEKTUUR

```

```

2150: 0DCF 68      PRINT  PLA      PULL RETURN ADDRESS FROM
2160: 0DD0 85 23      STAZ  MEPNT  STACK AND SAVE IT
2170: 0DD2 68      PLA
2180: 0DD3 85 24      STAZ  MEPNT  +01
2190: 0DD5 E6 23      PRTA  INCZ  MEPNT
2200: 0DD7 D0 02      BNE   PRTB
2210: 0DD9 E6 24      INCZ  MEPNT  +01
2220: 0ddb A0 00      PRTB  LDYIM $00  FETCH CHARACTER AND
2230: 0ddd B1 23      LDAIY MEPNT  PRINT IT
2240: 0ddf C9 03      CMPIM $03  EOT?
2250: 0DE1 F0 06      BEQ   PRTC
2260: 0DE3 20 34 13    JSR   PRCHA
2270: 0DE6 4C D5 0D    JMP   PRTA
2280: 0DE9 A5 24      PRTC  LDZ   MEPNT  +01 PUSH RETURN ADDRESS
2290: 0DEB 48      PHA   ON STACK AND
2300: 0DEC A5 23      LDZ   MEPNT  RETURN
2310: 0DEE 48
2320: 0DEF 60

```

SYMBOL TABLE 4000 4140 111282 PAGE 01

AREA	2000	BUF	0200	BUFR	0280	COMP	0D18
CRLF	11E8	DOTP	1476	DUMP	09DF	EAH	1A73
EAL	1A72	ERROR	0DB3	EXIT	0DA8	FFNME	0D89
FLNME	0D67	FOTP	0CCF	GB	1496	GBAS	0D54
ID	1A79	IDOTP	0D0B	LOAD	0CFE	MEPNT	0023
PNAME	0D76	PNTRS	0073	PRCHA	1334	PRFND	0D28
PRFNM	0CAC	PRINT	0DCF	PRLOAD	0D3C	PRTA	0DD5
PRTB	0ddb	PRTC	0DE9	RDTAPE	0B02	RECCHA	12AE
RESTTY	14BC	RET	0D88	SAH	1A71	SAL	1A70
SAVPTR	0CA4	STBPTR	0D59	STORE	0DAA	TAPEUT	0CA2

RESET

PAGE 01

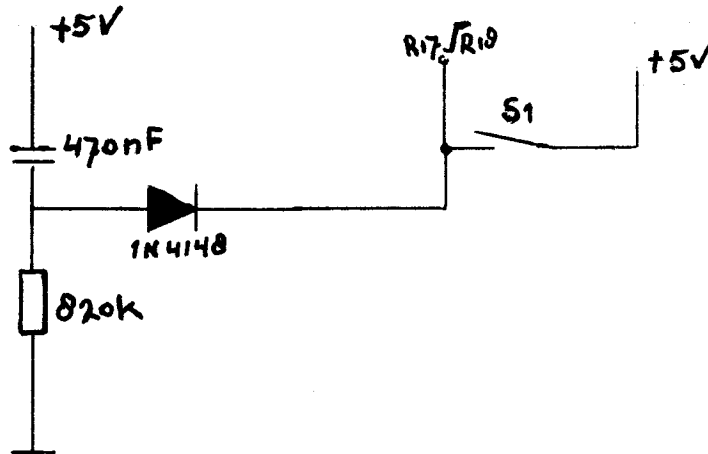
0010:
0020:
0030:
0040:
0050:
0060:
0070:
0080:
0090:
0100:
0110:
0120:
0130:
0140:
0150:
0160:
0170:
0180:
0190:
0200:
0210:
0220:
0230:
0240:
0250:
0260:
0270:
0280:
0290:
0300:
0310:
0320:
0330:
0340:
0350:
0360:
0370:
0380:
0390:
0400:
0410:
0420:
0430:
0440:
0450:
0460:
0470:
0480:
0490:
0500:
0510:
0520:
0530:
0540:
0550:
0560:
0570:
0580:
0590:
0600:

RESET PM START

=====

E. J. M. VISSCHEDIJK, DRAKENSTEYN 299, 7608 TR ALMELO.

MET BEHULP VAN EEN CONDENSATOR, WEERSTAND EN DIODE
WORDT ERVOR GEZORGD DAT BIJ POWER UP EEN RESET
WORDT GEGEVEN (zie Elektuur maart '81, p.55).



IN DE EPROM DIE DE SYSTEEMVECTOREN HERBERGT MOETEN
2 WIJZIGINGEN PLAATSVINDEN. DE RESETVECTOR MOET NA-
MELIJK ERGENS ANDERS HEENWIJZEN.

ADRES:	WAS:	WORDT:
FFFC	1D	80
FFFD	1C	0C

HET GEVOLG IS DAT BIJ EEN 'RESET' NAAR ADRES \$0C80
WORDT GESPRONGEN. EEN PLEK IN DE TM EPROM WAAR NOG
ZOVEEL RUIMTE IS. HIERIN MOET NU EEN OPSTARTPROGRAM-
MA'TJE ALS HIerna BESCHREVEN.
ALS HET SYSTEEM NU WORDT AANGEZET, DAN HOEFT ALLEEN
MAAR DE RUB-OUT TOETS TE WORDEN INGEDRUKT.

RESET	ORG	\$0C80
F1 00	PREG	* \$00F1
F2 00	SPUSER	* \$00F2
FF 00	MODE	* \$00FF

CLD	
SEI	DISABLE IRQ
LDXIM \$FF	ADJUST STACK POINTER
TXS	
STX	SPUSER
LDAIM \$04	RESET P-REG.
STA PREG	
LDAIM \$03	SET AD MODE
STA MODE	
JMP \$1000	JUMP TO INITPR. PM STARTUP

```

10 DEFFNZZ(C)=INT(C*100+.5)
20 DEFFNWW(C)=INT(C/60*100+.5)/100
30 DEFFNXX(C)=INT(C*100+.5)/100
40 GOSUB2600:Q2$="X":B=0
50 FORX=1TO10000:B=B+1
60 READW$:IFW$="←"THEN80
70 NEXT
80 B=B/10
90 PRINTB
100 RESTORE
110 Q9=0
120 F2=0:F3=0:F4=0:F1=F2+F3+F4:DIMA$(B,10):DIMX$(10)
130 FOR K=1 TO 10:READ X$(K):NEXT K
140 FORI=1TOB:FORJ=1TO10:READA$(I,J)
150 IFA$(I,9)="0"THEN170
160 NEXTJ:NEXTI
170 PRINT"##### SERVICE RAPPORTEN BESTAND M.R. VAN HINTUM #####"
180 PRINT:PRINT
190 PRINT"TYPE 0 VOOR OUTPUT VAN BEPAALDE DATUM"
200 PRINT"TYPE 1 VOOR OUTPUT VAN BEPAALD UI-NUMMER"
210 PRINT"TYPE 2 VOOR OUTPUT VAN BEPAALDE CLIENT"
220 PRINT"TYPE 3 VOOR OUTPUT VAN BEPAALDE WOONPLAATS"
230 PRINT"TYPE 4 VOOR OUTPUT VAN TOTAAL OVERZICHT"
240 PRINT"TYPE 5 VOOR OUTPUT VAN MACHINE+JOB-CODE+WERKTIJD+REISTIJD"
250 PRINT"TYPE 6 VOOR OUTPUT VAN NAAM CLIENT+WOOPLAATS+MACHINE+OPMERKING"
260 PRINT"TYPE 7 VOOR OUTPUT VAN OVERZICHT PER TYPE MACHINE
270 PRINT"TYPE 8 VOOR OUTPUT VAN DIRECTE UREN OP ALLE DATA'S
280 PRINT"TYPE 9 VOOR OUTPUT VAN GEHELE PROGRAMMA NAAR TAPE
290 PRINT
300 INPUT"UW KEUZE : ";L:PRINT
310 IFL=9THEN390
320 INPUT"WILT U EEN OUTPUT OP PAPIER ? (JA OF NEE) ";L9$
330 IFLEFT$(L9$,1)="J"THEN360:IFLEFT$(L9$,1)="N"THEN370:GOTO320
340 IFLEFT$(L9$,1)="N"THEN370
350 GOTO320
360 GOSUB2590:NULL3:GOSUB2600
370 GOSUB2600
380 PRINT
390 IFL=0THEN440
400 ONLGO410,470,500,530,1590,1860,2100,1980,1700
410 INPUT"WAT IS HET UI-NUMMER?":X1$
420 GOSUB2600:J=1:GOSUB1380
430 PRINT:GOTO190
440 INPUT"WAT IS DE DATUM ? (DD/MM/JJJJ) ":X1$
450 GOSUB2600:J=2:GOSUB1380
460 GOTO430
470 INPUT"WELKE CLIENT?":X1$
480 GOSUB2600:J=3:GOSUB1380
490 GOTO430
500 INPUT"WELKE WOONPLAATS?":X1$
510 GOSUB2600:J=4:GOSUB1380
520 GOTO430
530 T1=0:T2=0:T3=0:V1=0:V2=0:V3=0:V4=0:V5=0:V6=0:I1=0:I2=0
540 INPUT"WILT U EEN OVERZICHT VAN EEN BEPAALDE PERIODE?":X$
550 IFLEFT$(X$,1)="N"THEN600
560 GOSUB2600:INPUT"GEEF BEGINDATUM (DD/MM/JJJJ) : ";PA$
570 PRINT:INPUT"GEEF EIND-DATUM (DD/MM/JJJJ) : ";PC$
580 PB$=LEFT$(PA$,5):PD$=LEFT$(PC$,5)
590 GOSUB2600:Q9=0:GOTO610
600 GOSUB2600:Q9=1
610 J=2:F=0
620 FOR I=1 TO B
630 IF A$(I,J)="0" THEN 700
640 IFQ9=1THEN670
650 IFA$(I,J)<PB$THEN690
660 IFA$(I,J)>PD$THEN700
670 IF A$(I,J)=A$(I-1,J) THEN 690

```

```

680 F=F+1
690 NEXT I
700 FORI=1TOB
710 J=2
720 IFQ9=1THEN740
730 IFA$(I,J)=PC$THEN780
740 IF A$(I,J)="0" THEN 760
750 NEXT I
760 A8#=A$(I-1,2)
770 GOTO790
780 A8#=A$(I,J)
790 IFQ9=1THEN840
800 FORI=1TOB
810 J=2
820 IFA$(I,2)=PA$THEN850
830 NEXT
840 I=1
850 FORI=ITOB:J=6
860 IFQ9=1THEN880
870 IFA$(I,2)=PC$ANDA$(I+1,2)(<)PC$THEN1000
880 IFA$(I,J)="01"THEN970
890 IF A$(I,J)="02" THEN 980
900 IF A$(I,J)="03" THEN 990
910 IF A$(I,J)="0" THEN 1000
920 IFA$(I,J)="IND"THEN950
930 IFA$(I,J)="VAK"THEN960
940 NEXT I
950 I1=I1+VAL(A$(I,8)):GOTO940
960 I2=I2+VAL(A$(I,8)):GOTO940
970 T1=T1+1:V1=V1+VAL(A$(I,7)):V2=V2+VAL(A$(I,8)):GOTO 940
980 T2=T2+1:V3=V3+VAL(A$(I,7)):V4=V4+VAL(A$(I,8)):GOTO 940
990 T3=T3+1:V5=V5+VAL(A$(I,7)):V6=V6+VAL(A$(I,8)):GOTO 940
1000 V7=V1+V2:V8=V3+V4:V9=V5+V6:W9=V7+V8+V9
1010 IFV7=0THENV7=1:IFV8=0THENV8=1
1020 IFV9=0THENV9=1:IFW9=0THENW9=1
1030 W0=V1/V7:W1=V3/V8
1040 W2=V5/V9:W3=V2/V7:W4=V4/V8:W5=V6/V9:W6=V7/W9
1050 W7=V8/W9:W8=V9/W9
1060 V9=FNWW(V9):W9=FNWW(W9):W0=FNZZ(W0):W1=FNZZ(W1):W2=FNZZ(W2)
1070 W3=FNZZ(W3):W4=FNZZ(W4):W5=FNZZ(W5):W6=FNZZ(W6):W7=FNZZ(W7)
1080 W8=FNZZ(W8):V1=FNWW(V1):V2=FNWW(V2):V3=FNWW(V3):V4=FNWW(V4)
1090 V5=FNWW(V5):V6=FNWW(V6):V7=FNWW(V7):V8=FNWW(V8)
1100 PRINTTAB(22)"INSTALL.      ":"STORINGEN      ":"S-BEURTEN"
1110 IFQ9=1THEN1140
1120 A7#=PA$
1130 GOTO1150
1140 A7#=A$(1,2)
1150 PRINTA7#;"-":A8#;" ":"#####      ":"#####      ":"#####"
1160 PRINT"AANTAL";TAB(21)T1:TAB(33)T2:TAB(45)T3
1170 PRINT"REISTIJD IN UREN";TAB(21)V1:TAB(33)V3:TAB(45)V5
1180 PRINT"WERKT IJD IN UREN";TAB(21)V2:TAB(33)V4:TAB(45)V6
1190 IFT1=0THENV7=0:IFT2=0THENV8=0:IFT3=0THENV9=0
1200 PRINT"TOTAAAL-A";TAB(21)V7:TAB(33)V8:TAB(45)V9
1210 PRINT"REIST. IN % V. TOT-A";TAB(21)W0:Q2#:TAB(33)W1:Q2#:TAB(45)W2:Q2#
1220 PRINT"WERKT. IN % V. TOT-A";TAB(21)W3:Q2#:TAB(33)W4:Q2#:TAB(45)W5:Q2#
1230 IFT1=0THENW6=0:IFT2=0THENW7=0:IFT3=0THENW8=0
1240 PRINT"J-CODE IN % V. TOT-OV";TAB(21)W6:Q2#:TAB(33)W7:Q2#:TAB(45)W8:Q2#
1250 PRINT:T4=T3+T2+T1
1260 I1=INT(I1/60):I2=INT(I2/60)
1270 F1=I1+I2
1280 F2=(F*8)-F1:F3=INT(W9/F2*100+.5)
1290 PRINT"AANTAL TOTAAAL GEWERKTE UREN T/M ";A8#;" IS ";W9
1300 PRINT"UREN VAKANTIE IS ":";I2:"INDIRECTE UREN IS ":";I1
1310 PRINT"ZODAT MAXIMUM DIRECTE UREN IS ":";F2
1320 PRINT"TOTALE DIRECTE UREN VAN MAXIMUM IS ":";F3;" %"
1330 PRINT"AANTAL SERVICE RAPPORTEN T/M ";A8#;" IS ";T4

```

```

1340 GOSUB2580
1350 PRINT:INPUT"VOOR ANDERE OUTPUT TYPE '1' ";Q$
1360 GOSUB2600
1370 IF Q$="1" THEN 180
1380 FOR I=1TOB
1390 IF A$(I,J) = X1$ THEN 1430
1400 IF A$(I,J)="0" THEN 1550
1410 NEXT I
1420 GOTO 1380
1430 PRINT:K=1
1440 FOR M=1 TO 10
1450 PRINTX$(K);A$(I,M)
1460 LET K=K+1
1470 NEXT M
1480 PRINT
1490 PRINT"VOOR MEER GEGEVENS TYPE '1' IN "
1500 INPUT "VOOR ANDERE OUTPUT TYPE '2' IN ";T
1510 GOSUB2610
1520 IF T=1 THEN 1410
1530 GOSUB2580
1540 IF T=2 THEN 430
1550 GOSUB2600:PRINT:PRINT"MEER DATA IS ER NIET"
1560 INPUT"VOOR ANDERE DATA TYPE '1' IN. ";X2$
1570 GOSUB2600:GOSUB2580
1580 RETURN
1590 G1=1
1600 PRINT"TYPE MACHINE", "JOB-CODE", "REISTIJD", "WERKTIJD", "WOONPLAATS"
1610 PRINT:PRINT
1620 FORI=1TOB
1630 FORJ=5TO8:PRINTA$(I,J),
1640 IFA$(I,1)="0"THEN1830
1650 NEXT:PRINTA$(I,4)
1660 G1=G1+1:IFG1=60THEN1680
1670 NEXT
1680 PRINT:PRINT"TYPE MACHINE", "JOB-CODE", "REISTIJD", "WERKTIJD", "WOONPLAATS"
1690 PRINT:PRINT:NEXT
1700 GOSUB2600
1710 PRINT:PRINT
1720 FORY6=20TO0STEP-1
1730 FORY7=1TO2
1740 PRINT"OVER";Y6;
1750 PRINTTAB(7)"SECONDEN WORDT HET PROGRAMMA OP TAPE GEZET.":" ";
1760 PRINTCHR$(11)
1770 NEXT:NEXT
1780 PRINT
1790 PRINT
1800 PRINT"PROGRAMMA WORDT NU WEGGESCHREVEN.":SAVEA:PRINT
1810 PRINT"GEHELE PROGRAMMA STAAT NU OP TAPE ONDER ID-NUMMER # A #.
1820 PRINT:PRINT"ER ZIJN NU NOG";FRE(0);"PLAATSEN IN HET GEHEUGEN VRIJ."
1830 PRINT:INPUT"VOOR ANDERE OUTPUT TYPE '1' IN: ";X2$
1840 GOSUB2600:GOSUB2580:IFX2$="1"THEN190
1850 END
1860 PRINT"OVERZICHT VANAF ";A$(1,2);" VAN : "
1870 PRINT"NAAM CLIENT";TAB(30)"WOONPLAATS";TAB(50)"DATUM"
1880 PRINT"TYPE MACHINE";TAB(30)"OPMERKINGEN";TAB(50)"JOB-CODE"
1890 PRINT
1900 FORI=1TOB
1910 IFA$(I,10)="GEEN"THEN1970
1920 IFA$(I,2)="0"THEN1830
1930 IFA$(I,10)="X"THEN1970
1940 PRINT"--"
1950 PRINTA$(I,3);TAB(30)A$(I,4);TAB(50)A$(I,2)
1960 PRINTA$(I,5);TAB(30)A$(I,10);TAB(50)A$(I,6)
1970 NEXT
1980 PRINT"DATUM", "REIS UREN", "WERK UREN", "% PER DAG", "AANTAL CALL'S"
1990 PRINT:PRINT
2000 FORI=1TOB
2010 IFA$(I,J)="0"THEN1830
2020 AB=AB+VAL(A$(I,7)):AC=AC+VAL(A$(I,8)):AD=AD+1

```

```

2030 IFA$(I, 2) (>) A$(I+1, 2) THEN 2050
2040 NEXT
2050 AB=FNWW(AB):AC=FNWW(AC)
2060 AE=AB+AC:AF=AE/.08:AF=FNXX(AF)
2070 PRINTA$(I, 2), AB, AC, AF: TAB(49) Q2$, AD
2080 AB=0:AC=0:AE=0:AD=0:AF=0
2090 NEXT
2100 PRINT"OVERZICHT PER MACHINE VANAF ";A$(1, 2)
2110 PRINT:PRINT
2120 BA$="REL. 450":GOSUB2330
2130 BA$="REL. 550":GOSUB2330
2140 BA$="REL. 700":GOSUB2330
2150 BA$="REL. 750":GOSUB2330
2160 BA$="250":GOSUB2330
2170 BA$="RP-1":GOSUB2330
2180 BA$="RV-1":GOSUB2330
2190 BA$="RV-2":GOSUB2330
2200 BA$="ROTOLINE":GOSUB2330
2210 BA$="STARVUE":GOSUB2330
2220 BA$="MOTORMATIC":GOSUB2330
2230 BA$="MIRACODE II":GOSUB2330
2240 BA$="321":GOSUB2330
2250 BA$="322":GOSUB2330
2260 BA$="321-323":GOSUB2330
2270 BA$="322-323":GOSUB2330
2280 BA$="PROSTAR":GOSUB2330
2290 BA$="KOD. 324":GOSUB2330
2300 BA$="VER. II":GOSUB2330
2310 BA$="R. 420A":GOSUB2330
2320 GOTO1830
2330 Y1=0:Y2=0:Y3=0:Y4=0:Y5=0:Y6=0
2340 IFBA$="322" THENY3=30
2350 FORI=1TO6
2360 IFA$(I, 1)="0" THEN2440
2370 IFA$(I, 5)=BA$ THEN2390
2380 NEXT
2390 IFA$(I, 6)="02" THEN2420
2400 Y1=Y1+VAL(A$(I, 7)):Y2=Y2+VAL(A$(I, 8)):Y3=Y3+1
2410 NEXT
2420 Y4=Y4+VAL(A$(I, 7)):Y5=Y5+VAL(A$(I, 8)):Y6=Y6+1
2430 NEXT
2440 IFY3=0 THENY3=1
2450 IFY6=0 THENY6=1
2460 Z1=Y1/Y3:Z1=FNWW(Z1):Z2=Y2/Y3:Z2=FNWW(Z2)
2470 Z3=Y4/Y6:Z3=FNWW(Z3):Z4=Y5/Y6:Z4=FNWW(Z4)
2480 PRINT:PRINT"##### ";BA$;" #####"
2490 IFZ1+Z2=0 THENY3=0
2500 IFZ3+Z4=0 THENY6=0
2510 PRINT"AANTAL SERVICE BEURTEN : ";Y3;
2520 PRINTTAB(32)"AANTAL STORINGEN : ";Y6
2530 PRINT"GEMIDDELDE REISTIJD : ";Z1;
2540 PRINTTAB(32)"GEMIDDELDE REISTIJD : ";Z3
2550 PRINT"GEMIDDELDE WERKTIJD : ";Z2;
2560 PRINTTAB(32)"GEMIDDELDE WERKTIJD : ";Z4
2570 RETURN
2580 POKE&"1A05", 00: RETURN
2590 POKE&"1A05", 01: RETURN
2600 P8=USR(&"FF12", 0): RETURN
2610 P9=USR(&"FF0D", 0): RETURN
2620 DATA"UI-NUMMER.....":", "DATUM.....":
2630 DATA"NAAM CLIENT.....":", "WOONPLAATS.....":
2640 DATA"TYPE MACHINE.....":", "JOB-CODE.....":
2650 DATA"REISTIJD IN MIN. ....":", "WERKTIJD IN MIN. ....":
2660 DATA"ONDERDELEN.....":", "OPMERKINGEN.....":
2670 DATA01, 07/07/1982, JANSEN, DIDAM, REL. 450
2680 DATA01, 20, 30, X, X
2690 DATA02, 08/07/1982, LINSEN, ALSMEER, REL. 550
2700 DATA02, 40, 50, X, X
2710 DATA03, 09/07/1982, KASEREN, OS, MIRACODE II

```

```
2720 DATA03, 90, 70, X, X
2730 DATAX, 09/07/1982, X, X, X, IND, X, 300, X, X
2740 DATA03, 10/07/1982, HINSEN, BASR, STARVUE
2750 DATA03, 40, 70, X, X
2760 DATAX, 10/07/1982, X, X, X, VAK, X, 200, X, X
2770 DATA02, 11/07/1982, DENSNEN, ECH, REL. 450
2780 DATA02, 20, 80, X, X
2790 DATA0, 0, 0, 0, 0, 0, 0, 0, 0, +
```

OK

BESCHRIJVING SERVICE-RAPPORTEN PROGRAMMA.

AUTEUR: MAARTEN VAN HINTUM, Wolborgenmate 130, 7006 DL DOETINCHEM, Tel.08340-25320

Het programma is bedoeld om van service-rapporten een overzicht te geven omtrent informatie over bepaalde:

- identificatienummers (= uniek kenteken per machine)
- data
- namen van cliënten
- woonplaatsen van cliënten
- typen machines
- aard van het service-rapport (= bijv. storing; onderhoud)
- reistijden
- geplaatste onderdelen
- opmerkingen

De bedoeling is dat het programma steeds aangevuld wordt met nieuwe gegevens, door middel van nieuwe data-regels. De eerste vijf data-regels behoren vast bij het programma. De daarop volgende zijn regels met informatie. De laatste data-regel moet echter altijd als volgt zijn: DATA0,0,0,0,0,0,0,0,0, ←

De informatie moet in dezelfde volgorde geschreven worden als in de eerste vijf data-regels. Tijden moeten in minuten opgegeven worden. Als andere typen machines gebruikt worden, dienen de regels aan het eind van het hoofdprogramma waarin BA\$ benoemd wordt natuurlijk veranderd te worden. Er kunnen zoveel typen opgegeven worden als nodig is. Als aard van het service-rapport heb ik gekozen voor:

```
01 = INSTALLATIE
02 = STORING
03 = ONDERHOUDSBEURT (= SERVICEBEURT)
IND = VERANTWOORDE INDIRECTE MINUTEN (= bijv. VERGADERINGEN)
VAK = VAKANTIE MINUTEN
```

Een voorbeeld: DATA X,10/10/1982,X,X,X,VAK,0,480,X,X

Het werken met minuten lijkt erg lastig, maar door dit vaker te gebruiken valt het erg mee. Het is noodzakelijk voor de nauwkeurigheid.

Als er voor opmerkingen als data 'X' of 'GEEN' opgegeven wordt, dan verschijnt dit service-rapport niet bij het uitprinten wanneer Keuze '6' is gedaan.

Als het programma gestart wordt, verschijnt er een getal dat aangeeft hoeveel ingevoerde service-rapporten aanwezig zijn. Ter controle moet dit altijd een geheel getal zijn. De verdere gebruiksaanwijzing volgt vanzelf uit het laten 'runnen' van het programma.

POKE&"1A05",00 en POKE&"1A05",01 zet mijn teletype printer van 110 BAUD resp. aan of uit. Telkens als de subroutine Print Character (= 20 34 13) voorkomt springt mijn Basic naar adres \$1500. In deze routine wordt voortdurend gekeken of plaats \$1A05 geladen is met 00 of 01. In het laatste geval gaat de Elekterminal over op 110 BAUD en verschijnt er data op mijn printerlijn. In alle andere gevallen werkt alles normaal.

P8=USR(&"FF12",0) roept de subroutine Clear Sreen aan.

P9=USR(&"FF0D",0) roept de subroutine Cursor Home aan.

Deze routines kan men natuurlijk ook achterwege laten.

Door kleine veranderingen in het programma aan te brengen is het mogelijk om voor iedereen met dit programma een toepassing te vinden.

Een advertentie in Elektuur waarin PROTON Electronics aandacht vraagt voor haar 65XX kaarten en floppy disk met DOS was voor de redactieleden van Pelt en Smeehuizen en ook voor ons lid Franssen aanleiding te kontakten met genoemd bedrijf. Uiteindelijk vloede daaruit een gesprek voort tussen het bedrijf en ons bestuur om erachter te komen wat PROTON en haar produkten voor de club en wat de club voor PROTON te betekenen kan hebben.

Wat PROTON voor de club kan betekenen :

=====

Levering van hard- en software aan onze clubleden, tegen een speciale prijs.

- Disk-drive 5-1/4" (single-density, single-sided)
- Floppydisk controllerkaart FDC-4 (IBM3740 compatible, geschikt voor aansluiten van max. 4 drives en een Centronics compatible printer en een serial printer).
- Disk Operating System/Monitor als software ondersteuning.
- Diskette met systeem-software:
 - * ASM65 2-pass symbolic assembler (MOS-TECHNOLOGY standaard)
 - * EDIT65 Regelgeoriënteerde editor
 - * BASIC Protons snelle BASIC interpreter/compiler (werkt met LABELS en regelnummers en kent namen met variabele lengte tot 16 karakters).
 - * FORTH65 Forth programmeertaal (zeer uitgebreide versie met de disk als virtueel geheugen)
 - * UTILITY Een aantal programma's zoals:
 - OLOAD Load with offset
 - MAP Print memory usage of program
 - COMP Compare file to memory (+offset)
 - MERGE Combine two files (text)
 - SPLIT Split a file in 2
 - COUNT Count the lines of a file
 - * FORM Een simpele format-lister voor het maken van documentaties

Al deze programma's werken met disk.

Bestelnaam: FD-J voor JUNIOR, DOS/MON in 2 x 2732
FD-K voor KIM-1, DOS/MON in 2 x 2732
FD-A voor AIM-65, DOS in 1 x 2732

De nu geldende aanbieding is fl. 1085,-- voor:

- 1 enkelzijdige 5-1/4" diskdrive (single density)
- 1 Floppydisk controller FDC-4
- 1 DOS/Monitor in Eprom
- 1 Systeem disk met alle bovengenoemde programma's.

Verder zijn de volgende kaarten beschikbaar:

- PROM-8 kaart (in combinatie met FD-x) fl. 125,-- *
8 sockets (24) voor 2708, 2716, 2732 en 2764 (hiervoor moeten wel 28-pens voetjes worden gemonteerd). Instelbaar over het gehele adresbereik.
- DR32 32 Kbyte dynamische RAM-kaart fl. 390,--
- Keyboard KB-1 fl. 185,--

* Deze prijs geldt UITSLUITEND in combinatie met floppydisk-pakket.