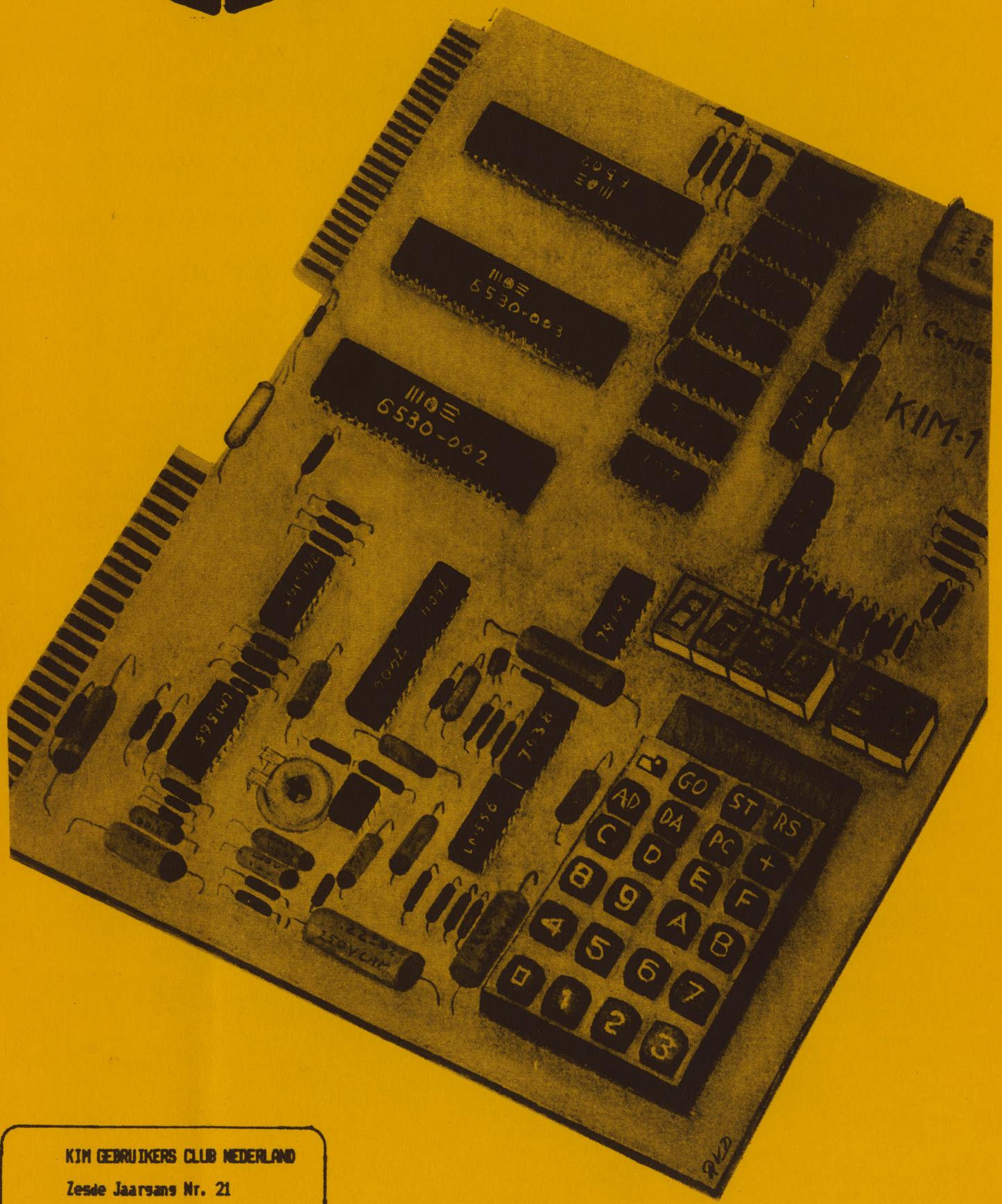


DE 6502 KENNER

NR. 21



KIM GEBRUIKERS CLUB NEDERLAND
Zesde Jaarsang Nr. 21
Mei 1982

www.dollflowers.com

KIM GEBRUIKERS CLUB

een club van 6502 gebruikers

NEDERLAND

Het doel van de vereniging is: het bevorderen van de kennisuitwisseling tussen de gebruikers van 6502-computers, zoals KIM, SYM, JUNIOR, AIM 65, System-65, ACORN, PET en CBM, APPLE, ATARI, ITT 2020, PC 100, OHIO Scientific Challengers, etc. etc.

Dit doel wordt o.a. gerealiseerd door vijf maal per jaar het huisorgaan "DE 6502 KENNER" te publiceren en vijf maal per jaar een clubbijeenkomst te houden, en wel als volgt:

Verschijningsdata
DE 6502 KENNER

derde zaterdag
van de maanden:
februari,
mei,
augustus,
oktober,
december.

Bijeenkomsten
van de club

derde zaterdag
van de maanden:
januari,
maart,
mei,
september,
november.

Naast deze activiteiten kunnen de leden gebruik maken van clubfaciliteiten, zoals de KIM-Club-KIM, de KIM-Club-JUNIOR, en de cassettebibliotheek.

De KIM GEBRUIKERS CLUB NEDERLAND is een volledig onafhankelijke vereniging met statuten en een bestuur. De club is ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

De samenstelling van het bestuur is thans als volgt:

Voorzitter:

Sekretaris:

Penningmeester:

Organisator:

Redactie:

Accomodatie:

Adres van de vereniging:

Lidmaatschappen:

Dagelijks bestuur:

Anton Müller
Sinj. Semeynstr. 78 - 1
1061 GM Amsterdam (020 - 860245)
Ruud H. Uphoff
Voorburgpad 10 (085 - 816935)
6843 EM Arnhem (19.00 - 20.00)
Herman C. Burgers
Manus Peetstraat 57
1183 LH Amstelveen (020 - 419958)
Leden:
Rinus Vleesch-Dubois
F. Nightingalestr. 212
2037 NG Haarlem (023 - 330993)
Willem L. van Pelt
Jacob Jordaensstr. 15
2923 CK Krimpen a.d. IJssel
(01807 - 19881)
Riche A. van Steen
Vingerhoed 11
6953 BZ Dieren (08330 - 21004)

zie adres v.d. sekretaris

fl. 40,= per kalenderjaar
gironummer 3757649 t.n.v.
H.C. Burgers te Amstelveen.

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

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

Willem van Pelt
Jacob Jordaensstr. 15
2923 CK Krimpen
aan de IJssel.

Redactie 6502 KENNER:

Vaste medewerkers:

- Anton Müller
- Willem van Pelt
- Frans Smeehuijzen

Freelance medewerkers:

- Frans Engelsman
- Renee de Hoop
- Fridus Jonkman

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

De 6502 KENNER verschijnt 5 x per jaar.

Copyright (C) 1982 KIM Gebruikers Club Nederland.

De voorpagina is een aquarel van een KIM, geschilderd door: Rinus Vlesch Dubois.

INHOUDSOPGAVE DE 6502 KENNER NR. 21 MEI 1982

1.	Inhoudsopgave	1.
2.	Van het bestuur ... door Anton Muller	2.
3.	Oorkonde, uitgereikt aan ... Hanny de Vries - v.d. Winden	4.
4.	Oorkonde, uitgereikt aan ... Siep de Vries	5.
5.	Brief van Ir. R. Vonk	6.
6.	Van de redactie ... door Willem van Pelt	7.
7.	Cassettebibliotheek ... door Hans Otten	8.
8.	Leesbaar programmeren in BASIC ... door Ruud Uphoff	10.
9.	Pocket Calculator ... door Siep de Vries	16.
10.	Patches op MICRO-ADE (vervolg dl.11) ... door Adri Hankel	27.
11.	JUNIOR's aan de PET ... door Ruud Uphoff	36.
12.	Marktinfo	26.
13.	Buss	35.
14.	Buss	48.
15.	Vraas en Aanbod	48.
16.	Asenda	
17.	Mededeling	

6502

VAN HET BESTUUR

Beste Clubsnoten,

Deze keer wil ik de pagina eens wijden aan de verkrijgbaarheid van bepaalde software voor de KIM, die wellicht van belang is voor onze JUNIOR-bezitters. Een aantal software producten, die bijna iedereen wel in bezit wil hebben, zijn een assembler/teksteditor en een of andere versie van BASIC. Wat betreft de assembler/teksteditor hebben we die van Carl W. Moser (BK + edit bufferruimte), verkrijgbaar bij Ingenieursbureau Schroder, Echternachlaan 161 te Eindhoven ... en Micro Ade van Peter Jennings, waarvan de verkrijgbaarheid door mij nog steeds in onderzoek is. Wat betreft de BASIC's hebben we TINY BASIC, te verkrijgen bij ITTY BITTY COMPUTERS, P.O. Box 23189 te San Jose, CA 95153 (U.S.A.), eveneens te verkrijgen bij The 6502 Program Exchange, 2920 Moana, Reno, NV 89509, U.S.A. Vraag tevens hun software brochure aan (1 dollar cash insluiten). Zij verkopen o.a. XPLO, FOCAL, FCL65, Text editor, assembler (ASM65), MICROCHESS, HUEY, WUMPUS. MICROSOFT BASIC is verkrijgbaar bij MSB-Verlag, att. R. Nedela, Mansdorfstrasse 10, D 7778 Markdorf (West Duitsland), Tel.: 07544 - 3058, voor zowel de KIM als de JUNIOR. Prijs: DM 288 incl. 13% Mehrwertsteuer. De patches om de KIM-versie aan te passen voor de JUNIOR staan gepubliceerd in Elektuur nr. 221 van maart 1982. MICROSOFT BASIC is ook verkrijgbaar bij JOHN-SON COMPUTER, P.O. Box 523, Medina, Ohio 44256 U.S.A. Bij MICROSOFT BASIC heb je ook het boek uit de Schaum's outline series, "Theorie and problems of Programming with BASIC" van Byron S. Gottfried nodis. Uitgever: McGraw-Hill Book Cy., ISBN-nr.: 0-07-023842-1. Nu we het toch over boeken hebben: een goedkope manier om aan een assembler te komen (om mee te experimenteren), is de aanschaf van het boek "Praktical Micro-computer Programming - The 6502" van Walter J. Weller, een, m.i. erg goed boek van 459 pagina's, waarin alles wat je over de 6502 wilt weten wordt behandeld, en waarbij je dan op de koop toe nog gratis (na het tekenen van een verklaring voor copyrights) een hex-dump van een assembler krijgt, waarvan de source-listing achterin het boek is afgedrukt. De kosten van het boek bedragen 32.95 US-dollar, excl. verzendkosten. Het boek is verkrijgbaar bij uitgeverij Northern Technology Books, Box 62, Evanston, IL 60204 U.S.A.), ISBN-nr.: 0-930594-98-8. Voor software-mensen, die ook wel eens met hardware willen stoeien, kan ik het boek "Programming & Interfacing the 6502; with Experiments" van Marvin L. de Jong aanbevelen, uitgever Howard W. Sams & Co., Inc., 4300 West 62nd Street, Indianapolis 46268, U.S.A.

Anton Muller,
voorzitter.

De eerste lustrum bijeenkomst van de KIM club op 20 maart j.l. is een enorm geslaagde dag geweest. Er waren 65 leden, waarvan er 5 hun vrouw hadden meegenomen. Ruim 20 leden hadden hun systeem bij zich, dus ook aan die oproep is ruimschoots voldaan. Even een korte samenvatting van wat er allemaal te doen was. In de ochtend de opstelling van de apparatuur en demonstraties. Er was o.a. een orgel van Bas van Bemmel, met drie klavieren, dat werkte op een SYM, met 8 toongeneratoren (AY3-8910) van General Instruments, waar een uitstekende kwaliteit geluid uit kwam. Als je niet beter wist zou je denken dat het een "normaal" orgel was. Dan was er nog een orgel, met 1 klavier en 1 AY3-8912. Verder een KIM (van mij) die een gesproken hex dump uitvoerde, met gebruik making van de DT1050 Digitaler van National Semiconductor. Een PC100 (= AIM65) van Riche van Steen, met o.a. demonstratie van FORTH en het schaakprogramma van Theo Kortekaas dat tegen zichzelf stond te schaken. Een Junior van De Haan, die een aantal 220 Volts lampen liet aan en uitschakelen. Een KIM van Dral, die de stoplichten op een verkeerskruispunt regelt. Een 6502 systeem van Bob van de Oudeweetering (helemaal zelf ontwikkeld), met daaraan gekoppeld een Tektronics grafisch beeldscherm en een A4-plotter, om tekeningen te plotten, waarbij gebruik gemaakt kan worden van een grote hoeveelheid standaard symbolen, die met een druk op de knop tevoorschijn kunnen worden gehaald op elke willekeurige plek op de tekening. Een Junior met een compleet administratie systeem, geschreven in machinetaal. De CBM van Herman Burgers, met daarop de ledenadministratie van de KIM club, geschreven door Ruud Uphoff in assembleer taal. De KIM van Gerrit de Vries (broer van Siep), met daaraan gekoppeld een hijskraan gebouwd van een Fischer Technik bouw pakket, waarbij hij een programma heeft geschreven in machine taal, dat een "leer"-mode en een "uitvoerings"-mode heeft, dat het geleerde tot het oneindige herhaalt. Een KIM van Theo Kortekaas, met daaraan gekoppeld twee diaprojectoren, die door de KIM worden bestuurd, alsmede een bandrecorder met muziek voor begeleiding van de diashow. Getoond werd de produktie "RITS". De Apple Van Uwe Schroeder, alsmede een videorecorder met kleuren TV, waarop hij de dansvoorstelling "Verte" liet zien, waarbij voor ons computeraars het meest interessante de aansturing van 4896 fietslampjes was, die de begeleidende lichtbeelden verzorgden. En last but not least Jos Koopmans, van het gelijknamige ingenieursbureau, met het Onyx C8002 systeem. De C8002 is een krachtige 16 bitter van Onyx, standaard uitgerust met 512 Kb RAM, uitbreidbaar tot 1 Mb en hard disk van 18 Mb uitbreidbaar tot 160 Mb online, cartridge tape backup met cyclic redundancy check. Kosten van de minimum configuratie met

standaard 8 terminal aansluit mogelijkheden (RS232) ongeveer f. 60.000,- Het geheel werkt onder de besturing van UNIX met 8 gebruikers gelijktijdig aangesloten, compilers o.a. COBOL, FORTRAN, BASIC, PASCAL etc. Kosten van UNIX ongeveer f. 7.500,-, kosten van de compilers tussen de f. 1.000,- en f. 2.000,- per stuk. Erg fraai. Jos Koopmans is een aantal jaren geleden begonnen met de verkoop van Superboards en andere OSI systemen en doet nu in systemen zoals ik zojuist heb omschreven, e.e.a. in een samenwerkingsverband/paraplu-organisatie onder de naam TRICOMP, samen met Ingenieurs bureau Schroeder en CAB Holland. Alle andere aanwezigen met hun systemen moeten het mij niet kwalijk nemen, dat ik hen niet heb opgesomd, kortweg omdat het mij aan tijd ontbrak om alles te bekijken.

Na de lunch volgde het officiële gedeelte van deze dag, de benoeming van Siep de Vries tot Ere-Voorzitter en Hanny de Vries - van der Winden tot Ere-lid. Siep heeft zich uitzonderlijk verdienstelijk gemaakt voor de vereniging door o.a. de oprichting van de club, het in grote lijnen bepalen van het karakter van de club, het houden van een aanzienlijke hoeveelheid hoogwaardige lezingen de microcomputer betreffende, het enorme geduld waarmee hij keer op keer, soms tot diep in de nacht, clubleden, die met technische problemen zaten uit de brand wist te helpen. Nogmaals van harte gefeliciteerd met je benoeming, Siep. In dezelfde periode dat Siep voorzitter was, heeft Hanny het sekretariaat gerund. Met de benoeming van Hanny de Vries tot Ere-lid, is zij ons eerste en enige vrouwelijke clublid, iets om bijzonder trots op te zijn. Wist zij er in het begin bijzonder weinig van, thans zeggen de woorden als KIM, RAM, ROM etc. haar boekdelen. Haar bijzondere inzet, door zich geheel belangeloos drie jaren in te zetten voor de opzet en de administratie en registratie van de club leden en alles wat daar om heen hangt, heeft ons doen besluiten haar voor te dragen voor deze ere titel. Nogmaals vanaf deze plaats van harte proficiat met je benoeming, Hanny.



Anton Muller,
Voorzitter

OORKONDA

Heden, de twintigste maart in het jaar negentienhonderd
tweeëntachtig is te Amersfoort

Johanna Alida

Wilhelmina van der Winden

geboren te Beverwijk op 2-1-1940 -
wonende te Haren (n-h), benoemd tot

ERE-LID

van de kin gebruikers club nederland.

de voorzitter

de secretaris

de penningmeester

A. Müller

R.H. Wiphoff

H.C. Burgers

OORFONDE

Meden, de
tweëtieste maart in het jaar
negentienhonderd tweeëntachtig is

Sybrén de Vries

geboren te Leentwarden, op 18-12-1941,
technisch directeur te Lincen (sch),
in Amersfoort benoemd tot

ERE-VOORZIJTER

van de kind gebruikers club Nederland.

de voorzitter

A. Müller

de secretaris

R. H. Uphoff

de penningmeester

H. C. Burgers

Ir. R. Vonk
St. Willibrorduslaan 2
5591 BG HEEZE.

Heeze, 20-2-82

Geachte Redactie,

Bij deze twee tips aangaande de Uitgebreide JUNIOR-computer

1. Bij lage Baudsnelheden (110 Baud bv.) blijken er moeilijkheden te ontstaan als bv. 'M' ingetoetst wordt, waarop de computer met 'HEXDUMP:' moet antwoorden. Alleen komt er geen 'HEXDUMP:', maar apekool. Reden: de subroutine RECCHA wacht na het label RECC een bittijd T; het tweede stopbit wordt niet (waarom?) afgewacht. In bovengenoemd voorbeeld stuurt de computer al een ASCII-teken terug tijdens de tweede stopbit; de UART raakt dan de kluts kwijt ... In mijn (eigen geschreven) Teletype Outputroutine heb ik v o o r d a t aan de uitvoer begonnen wordt met het startbit toegevoegd: JSR DELBIT (20 12 13); anders moet de Eprom veranderd worden door bv. een JMP te zetten.

Bijv.:

12 D4	4C 80 0C
0C 80	20 12 13
0C 83	20 12 13
0C 86	4C D7 12

2. Als in TM of PM de routine DUMP/DUMPT gebruikt wordt om een stuk geheugen op de band te zetten waarin pagina 01 voorkomt (stack !), kunnen er vreemde dingen gebeuren bij het inlezen; het terugkeeradres (in de stack) kan dan verminkt/overschreven zijn. In de praktijk geen probleem als:

inlezen en uitlezen (save en get) via TM
of idem via PM

wel problemen als:

inlezen via TM
uitlezen via PM (of omgekeerd)

In het geval het fout gaat, staat het programma wel in het geheugen, alleen moet via Reset de zaak tot de orde geroepen worden. Het zojuist (met problemen) ingelezen programma is dan wel bruikbaar.

3. No 3. Nog een vreemde zaak in PM-Editor is de warme CEND-start (17C5). Opgeven van BEGAD en ENDAD moet wel (subr. INPAR, 17CD), maar heeft geen effect; de waarden worden niet oversgenomen in 00E2/E5.

Hopenlijk is dit een bijdrage in geval anderen tegen soortgelijke problemen aanlopen.

R. V.

In de eerste plaats een enkele opmerking over de thans aangeboden programma's, waarvan m.i. het meest opvalt hoezeer de belangstelling is gericht op het feit dat er hogere programmeertalen, zoals BASIC, en tussen dat en de machinetaal, zoiets als MICRO-ADE en de assembler van Moser, bestaan. Hoewel ik enerzijds konstateer dat onze clubleden voortdurend stoeien met machinetaal, aan de andere kant neem ik waar dat er ook behoefte bestaat aan 'programmeer-gemak', als ik het zo even mag zeggen. Vanuit de idee dat onze edities zoveel mogelijk moeten aansluiten bij de behoeften van de leden, wordt getracht daaraan te voldoen. Ruud Uphoff zowel als Adri Hankel, maar ook onze Belgische vriend Van Nieuwenhoven Koen, van wie U binnenkort een serie aanpassingen op de KIM-1 BASIC kunt verwachten, hebben hun aandacht erop gericht. Een tijd geleden demonstreerde Ruud Uphoff, samen met Rinus Vleesch Dubois, hoe men met behulp van een PET vele JUNIOR's kon besturen. Om dat mogelijk te maken werden een aantal demonstratie-programma's gemaakt. Er is toen beloofd e.e.a. te publiceren. Aan die belofte komen we nu tesemoet. Wat de aangeboden herplaatsing van de Pocket Calculator van Siep de Vries betreft, het is niet alleen een mooi programma maar ook nog even een terugblik op Amersfoort, waar Siep de hulde kreeg die ook hem toekwam.

Op 28 februari en, indien volgens plan verlopen, op 10 april j.l. werd in Klubhuis 'De Boskant' te Den Haag een regionale bijeenkomst georganiseerd. Tom den Duijf vervolgde het initiatief van John van Sprang. Op eerstgenoemde bijeenkomst was ook Ruud Uphoff aanwezig, die de ca. dertig aanwezigen informatie verstrekte over de VIA 6522. Een schitterende (zon)dag! Het enthousiasme van de leden laat zich niet door een, voor bijeenkomsten zo riskante dag, van de wijs brengen.

Nog een opmerking over in te zenden materiaal. Indien U hardware of software hebt, schroomt U niet het in te zenden. Wij zullen proberen vast te stellen of plaatsing mogelijk is. Probeer in elk geval wel aan een aantal dingen te denken. Bijvoorbeeld de gewenste vorm, in welke het liefst een beschrijving van het programma, labelnamen en vooral ook kommentaar. Als het kan uitgetypt. Verder staan we U zoveel mogelijk bij. Wij schromen het contact met U niet. Gezamenlijk valt er heel wat te doen. Maar denk vooral niet: ach, dat is niet goed genoeg om in te sturen. Dan dreist het gevaar dat er te weinig kopy komt. Laten we niet vergeten dat het clubblad er niet alleen voor U is, maar vooral ook door U.

W.L. van Pelt.

CASSETTEBIBLIOTHEEK.

Zoals in DE 6502 KENNER al is aangekondigd, gaat de redactie de service uitbreiden met een cassette-service. Er zijn op het moment twee cassettes leverbaar, opgenomen in KIM of JUNIOR hypertape formaat. Als de source-code (alleen Micro Ade nog) aanwezig is, dan wordt dat op de andere zijde van de cassette ook opgenomen. In ieder geval is de object (tweemaal) aanwezig. Wat er op de cassette staat, en hoe, wordt met een begeleidend schrijven duidelijk gemaakt. Gebruik en beschrijving van de programma's is in DE 6502 KENNER of KIM KENNER te vinden. 'S' achter de programmaam betekent: ook in Micro Ade source aanwezig.

INHOUD KIM CASSETTE NR. 1 KANT 1 CODE

ID	NAAM	NR. KIM/6502	SA	EA
01	MASTERMIND	4	0000	0400
02	MASTERMIND	4	0000	0400
03	TALENSTUDIE HULP	13	0100	0400
04	TALENSTUDIE HULP	13	0100	0400
05	SCHAAKPROGRAMMA DL 1	11	0000	0400
06	SCHAAKPROGRAMMA DL 1	11	0000	0400
07	SCHAAKPROGRAMMA DL 2	11	1780	17E2
08	SCHAAKPROGRAMMA DL 2	11	1780	17E2
09	TARGET 1 KOLOM	8	0200	0364
0A	TARGET 1 KOLOM	8	0200	0364
0B	TARGET 6 KOLOM	8	0200	0400
0C	TARGET 6 KOLOM	8	0200	0400
0D	MOVIT + SCHAAKOPENINGEN MICROCHESS	8	06F2	0BBA
0E	MOVIT + SCHAAKOPENINGEN MICROCHESS	8	06F2	0BBA
0F	VERSNELLEN MICROCHESS	11	2000	2090
10	VERSNELLEN MICROCHESS	11	2000	2090
11	SUPERTAPE	13	3E00	3F4D
12	SUPERTAPE	13	3E00	3F4D
13	REACTIESNELHEIDSMETER	13	0200	0384
14	REACTIESNELHEIDSMETER	13	0200	0384
15	VERKEERSLICHTEN	16	0200	02D6
16	VERKEERSLICHTEN	16	0200	02D6
17	LOCATE EN REPLACE	18	1780	17E5
18	LOCATE EN REPLACE	18	1780	17E5
19	EPROM PROGRAMMER	5	0200	035C
1A	EPROM PROGRAMMER	5	0200	035C
1B	TAPE COPY	18	0200	0400
1C	TAPE COPY	18	0200	0400

INHOUD KIM CASSETTE NR. 1 KANT 2 MICRO ADE SOURCE

ID	NAAM	NR. KIM/6502	SA	EA
01	VERSNELLEN MICROCHESS	11	3600	3F36
02	VERSNELLEN MICROCHESS	11	3600	3F36
03	SUPERTAPE	13	3600	418C
04	SUPERTAPE	13	3600	418C
05	REACTIESNELHEIDSMETER	13	3600	48A4
06	REACTIESNELHEIDSMETER	13	3600	48A4
07	VERKEERSLICHTEN	16	3600	3F8C
08	VERKEERSLICHTEN	16	3600	3F8C
09	LOCATE EN REPLACE	18	3600	4052
0A	LOCATE EN REPLACE	18	3600	4052
0B	EPROM PROGRAMMER	7	3600	45C8
0C	EPROM PROGRAMMER	7	3600	45C8
0D	TAPE COPY	18	3600	5B50
0E	TAPE COPY	18	3600	5B50

INHOUD JUNIOR CASSETTE NR. 1 KANT 1 CODE

ID	NAAM	NR. KIM/6502	SA	EA
01	CONVERSIE DAG NAAR WEEK	18	0200	032F
02	CONVERSIE DAG NAAR WEEK	18	0200	032F
03	ZENUWSLAG	18	0200	028D
04	ZENUWSLAG	18	0200	028D
05	MUSIC BOX	17	0200	028B
06	MUSIC BOX	17	0200	028B
07	DOKATIMER	17	0200	026B
08	DOKATIMER	17	0200	026B
09	AUT. REG. UITLEZING	16	1A00	1A77
0A	AUT. REG. UITLEZING	16	1A00	1A77
0B	LOTTO	15	0000	0082
0C	LOTTO	15	0000	0082
0D	MOONLANDER	15	0000	00C4
0E	MOONLANDER	15	0000	00C4
0F	SC/MP INTERFACE AFREGELLEN	18	0200	023B
10	SC/MP INTERFACE AFREGELLEN	18	0200	023B
11	BROWSE	16	0100	018A
12	BROWSE	16	0100	018A
13	STOPWATCH	15	0000	004E
14	STOPWATCH	15	0000	004E
15	BELGISCHE LOTTO	18	0200	04D0
16	BELGISCHE LOTTO	18	0200	04D0

INHOUD JUNIOR CASSETTE NR. 1 KANT 1 SOURCE

ID	NAAM	NR. KIM/6502	SA	EA
01	CONVERSIE DAG NAAR WEEK	18		
02	CONVERSIE DAG NAAR WEEK	18		
03	ZENUWSLAG	18		
04	ZENUWSLAG	18		
05	MUSIC BOX	17		
06	MUSIC BOX	17		
07	DOKATIMER	17		
08	DOKATIMER	17		
09	AUT. REG. UITLEZING	16		
0A	AUT. REG. UITLEZING	16		
0B	LOTTO	15		
0C	LOTTO	15		
0D	MOONLANDER	15		
0E	MOONLANDER	15		
0F	SC/MP INTERFACE AFREGELLEN	18		
10	SC/MP INTERFACE AFREGELLEN	18		
11	BROWSE	16		
12	BROWSE	16		
13	STOPWATCH	15		
14	STOPWATCH	15		
15	BELGISCHE LOTTO	18		
16	BELGISCHE LOTTO	18		

Programma's in de cassettebibliotheek zijn of origineel van onze clubleden of bewerkingen van programma's uit het First Book of Kim. Alle rechten van de programma's blijven aan de auteurs voorbehouden. De kosten van de cassette zijn alleen maar kostendekkend en niet bedoeld om de programma's te kopen. **DE CLUB IS EN WORDT GEEN HANDEL!**
Kosten per cassette: fl. 12,50.

Bestellen van de cassettes onder opgave van naam, adres en nummer van de cassette (KIM = 1, JUNIOR = 2), vergezeld van een girobetaalkaart of bankcheque (groene of EURO) bij: Hans Otten, Ottoborrengoed 33, 3871 MJ Hoewelaken. Levering kan enige weken duren. Andere wijzen van bestellen bestaan niet.

LEESBAAR PROGRAMMEREN IN BASIC.

Op het oogenblik is de vraag naar "MICROSOFT" basic weer groot, dat komt vooral omdat veel JUNIOR bezitters hun systeem nu in een dusdanige staat hebben gekregen dat het echt de naam van "Personal Computer" mag gaan voeren: Meer geheugen, een ASCII keyboard en een video display zijn hiervoor de voorwaarden. Eigenlijk gebeurt hier iets in de verkeerde volgorde: De trotse bezitters van een dergelijk systeem hebben veelal leren programmeren in assembler en gaan nu overstappen op een hogere programmeertaal, die bij uitstek geschikt is om te leren denken in de simpele vorm die voor een computer nodig is. Elk detail moet de domme machine precies in een programma worden opgedragen. Basic laat een voortdurende interactie tussen mens en machine toe, die echter ook een aantal ernstige nadelen heeft:

De vrijheid die basic de programmeur toestaat is een van de oorzaken van het ontstaan van de enorme onleesbare ruinhooft die dan een "programma" wordt genoemd. Men hoeft de diverse tijdschriften die regelmatig iets in basic publiceren maar eens door te bladeren om diverse voorbeelden te kunnen noemen. Nu is het ontstaan van dergelijke rommel niet zo verwonderlijk als we ook nog bedenken dat basic wordt gebruikt op kleine systemen, die niet zijn uitgerust met een overcapaciteit aan geheugen. Bovendien haalt de snelheid van de meeste micro-processors op een stukken na de snelheid van de professionele grote computers. Laten we eens gaan zien welke suggesties het handboek van b.v. de PET de gebruiker doet om te komen tot een zo hoog mogelijke executiesnelheid en minimaal geheugen gebruik:

Verwijder alle onnodige spaties.
 Verwijder alle REM statements.
 Zet zoveel mogelijk instructies op een regel.
 Gebruik de "nul" elementen van een matrix.

Deze adviezen zijn zonder meer juist, als we alleen moeten letten op snelheid en geheugengebruik. Ze bereiken echter ook nog iets anders: We krijgen een volstrekt onleesbaar programma. Als er na enige tijd iets aan moet worden veranderd gaat dat heel moeizaam: Vaak besrijft de auteur zelfs zijn eigen programma al niet meer! We praten dan nog niet over het aanpassen van zo'n programma voor een andere machine. Het kan echter ook heel anders:

Met opzet heb ik hierboven de belangrijkste adviezen voor een optimale executiesnelheid achterwege gelaten. Ze zijn namelijk NIET in strijd met de eisen voor leesbaarheid van een programma. Integendeel:

GEBRUIK GEEN CONSTANTEN MIDDEN IN EEN PROGRAMMA
 DEFINIEER ALLE GEWONE VARIABELEN VOOR DE EERSTE MATRIX
 KIES DE VOLGORDE VAN DE DEFINITIES ZORGVULDIG
 DEFINIEER GEEN VARIABELEN MIDDEN IN EEN PROGRAMMA

Stel dat we de getallen van 1 tot 100 in een matrix willen zetten. Voor de hand liggend is:

```
100 DIMA(99):FORA=0TO99:A(A)=A+1:NEXT
```

Dit is de "vieze" methode. Er is gespaard op geheugenruimte en tijd door consequent alle mogelijke spaties weg te laten. Ook is het nul element van de matrix hier ten onrechte gebruikt, waardoor de onlogische aftrekking A-1 moet plaats vinden. Erser is dat er een enorme tijd wordt verspild met het 100 maal opnieuw inlezen van de constante "1". Ook moet de gehele matrix bij het definiëren van 7 plaats worden opgeschoven om plaats voor deze variabele te maken. Daarom wil ik voorstellen:

```
100 A=0:B=1:C=100
110 DIM A(100)
120 FOR A=B TO C:
130 : A(A)=A
140 NEXT
150 END
```

Op de 5 verspilde bytes voor A(0) kijken we niet. Het is veel logischer de 73e waarde van A te vinden in A(73) i.p.v. A(72). Laten we nu een klein praktisch probleem nemen. Het komt regelmatig voor dat we een machinetaalroutine in een basic programma willen gebruiken. Een handige methode is het schrijven van deze routine met een assembler, om hem vervolgens om te zetten in DATA statements. Daarbij moet elk byte naar decimaal worden omserekend en het is dus prettig dat dit klusje door een stukje software "gereedschap" in basic kan worden opgeknapt. De eerste listing laat een dergelijke routine op de min of meer gebruikelijke methode zien. Bezitters van een PET of CBM zullen ons wel kans zien deze routine te ontcijferen, of ons erger: Ze zullen daar misschien helemaal de moeite niet voor nemen, maar het geheel gewoon intypen en op cassette zetten. En dan werkt het. Dat wel. Jammer voor bezitters van een APPLE of een KIM want eer die er ooit achter komen wat die grafische tekens en diverse decimale adressen te betekenen hebben.... Het kan ook anders! De volgende lijst laat precies hetzelfde programma zien, maar op een manier die schijnbaar tegen alle regels indruist. Het klinkt gek, maar dit programma heeft niet ten doel, te doen waar het voor geschreven is, maar het staat hier op maar liefst 1,5 bladzijde "leesbaar" te zijn. Natuurlijk functioneert het wel. Alleen neemt het de gigantische geheugenruimte van meer dan 3K in beslag omdat we elke regel voor tijd en ruimtebesparing aan onze laars hebben gelapt, als deze de leesbaarheid ook maar enigszins dreigde aan te tasten. Er is

gebruik maakt van subroutines, niet omdat een programmadeel vaker nodig was, maar omdat deze een programma logisch onder verdelen. Het hoofdprogramma bestaat dan ook slechts uit de regels 100-170. Daarna volgen de diverse "block's". De eerste subroutine waarmee elk programma dient te beginnen is het definiëren van de variabelen. Ook constanten worden als variabele uitgevoerd. Dit levert een keurige lijst op van variabelen en hun betekenis. Een kleine toelichting bij de commentaren: Ik heb het REM statement in mijn interpreter door het teken "[" vervangen, omdat het woord REM de leesbaarheid van het commentaar afbreuk doet. Een kwestie van smaak, die de compatibiliteit niet aantast want als een andere CBM mijn tapes

leest verschijnt er gewoon REM in een listing en omgekeerd. Merk op dat de variabelen zoveel mogelijk zijn voorzien van logische namen, die al iets over hun gebruik vertellen. Jammer dat Microsoft hier een hinderlijke beperking heeft: In de naam van een variabele mag geen basic-sleutelwoord voorkomen. Als men dus de constante 1 wil benoemen is "ONE" niet toegestaan omdat de basicinterpreter dan het sleutelwoord "ON" zal herkennen. Het doel van de variabelen lijst is niet alleen maar de overzichtelijkheid van het programma, maar vooral de executie snelheid. Het vinden van de waarde van een variabele gaat namelijk een factor tien sneller dan het inlezen van een constante, dus is de winst binnen een FOR-NEXT loop enorm! De tijd, nodig voor het inlezen van een constante, wordt nu gebruikt op een moment dat we altijd tijd genoeg hebben: Tijdens het interactief gedeelte van het programma, waarin dit toch om invoer van de gebruiker aan het vragen is. Voor strings is de tijdwinst overigens nihil. Toch staan ze om een andere reden in de lijst: Natuurlijk vanwege het overzicht, maar bovendien is de volgorde in de lijst niet willekeurig: De variabelen staan in de volgorde waarin ze het meest worden gebruikt, want de eerste gedefinieerde variabele wordt ook het eerst weer teruggevonden. Meer tijdwinst dus. Behalve in deze subroutine staan we basic niet toe om nog een andere variabele of in het programma te creëren en het is eigenlijk een gebrek van basic dat men tijdens een programma executie zomaar variabelen mag definiëren.

Standaard basic mist structuren zoals IF-THEN-ELSE en bereikt het niet verder dan IF-THEN. In zo'n geval kan men toch weer kiezen tussen nette en vieze methoden. Principieel blijven: Na THEN behoort nooit een ander statement te staan dan een sprong. Constructie van b.v een splitsing waarbij routine "TRUE" of "FALSE" moet worden doorlopen:

```
100 IF A=ZERO THEN 120
110 : GOSUB 500:GOTO130
120 : GOSUB 600
130 END
```

6502

PROGRAMMEERTALEN

Niet proberen zoveel mogelijk achter THEN te frommelen. Microsoft gooit voorafgaande spaties in een regel weg! Bah! Gelukkig is er een mogelijkheid om onze eigenwijsheid bot te te vieren door een dubbele punt aan het begin van de regel te zetten. Zo worden twee of meer opeenvolgende FOR-NEXT loops leesbaar evenals de regels die tot het ELSE-gedeelte van een IF-THEN behoren.

In deze staat is het programma voor iedereen te bekijken, en wat veel belangrijker is: Er is een goed stuk documentatie en we kunnen later het programma eenvoudig wijzigen. Het aldus geschreven programma borgen we op in een achtergrond-geheugen en het gaat in ons archief. En nu de andere kant van de zaak:

We moeten nu een versie hebben die we slechts voor een doel op band of disk zetten: LADEN IN HET GEHEUGEN en RUNNEN. Hoe het programma is geschreven interesseert ons geen zier. Als het maar werkt. En snel. En zo klein mogelijk is! Daaraan voldoet het programma wat we zojuist hebben bekeken

beslist niet. Alle spaties en REM statements gaan we dus wegschrijven en we zetten zoveel mogelijk statements op een regel. Omdat dit met de hand een ondoenlijke zaak is, zal een mooi gestructureerd programma in basic wel altijd een onmogelijke wens blijven? Dus toch maar weer ruinhooft schrijven? Ja hier moeten we natuurlijk weer iets hebben om het vuile werk te doen. En dat is een programma in assembler van ca. 3/4 K dat ik de volgende keer zal publiceren. Nu zien we het resultaat van het gebruik in de derde lijst. Daar staat exact hetzelfde programma nog eens, maar nu zijn zoveel mogelijk instructies op een regel gekerf. Wat de basic compressor heeft gedaan is zelfs met de hand onmogelijk op de meeste systemen, want het aantal bytes per regel is nu maar liefst maximaal 240 en PRINT-FOR-DATA-LET etc. zijn dan slechts 1 byte! Deze afdruk van het resultaat is iets wat ik eigenlijk niet had mogen doen. Hij dient slechts om U een indruk van het eindresultaat te geven. Meer niet. Het is dus nooit de bedoeling een aldus gecomprimeerd programma met LIST tevoorschijn te roepen, laat staan te wijzigen! Daarvoor dient het origineel, waarvan we na wijziging in een seconde een gebruiksklare nieuwe versie maken met de BASIC-COMPRESSOR.

Ruud Uehoff.

* GEWIJZIGD REDAKTIE-ADRES !!! *

UNSTRUCTURED FORM

```

100 PRINT "J"; INPUT "BEGIN,END"; X,Y: IF Y-X > 203 THEN PRINT "CAN'T FIX ON SCREEN": END
110 FOR A=623 TO 635: POKE A, 13: NEXT: POKE 158, 12: N=1000: PRINT "J": FOR A=X TO Y STEP 17
120 PRINT N "DATA "; FOR B=A TO A+16: IF B > Y THEN 140
130 PRINT MID$(STR$(PEEK(B)), 2), ",";
140 NEXT: PRINT "|| "; N=N+10: NEXT: PRINT "8"; : SYS 50066

```

READY.

STRUCTURED FORM

100 GOSUB 450:	[INITIALIZE VARIABLES
110 PRINT CLEAR\$:	[CLEAR SCREEN
120 GOSUB 700:	[GET USER INPUT
130 GOSUB 380:	[PREFIX KEYBOARD RECEIVE BUFFER
140 GOSUB 220:	[WRITE DATA STATEMENTS ON SCREEN
150 PRINT HOME\$:	[SET CURSOR ON FIRST DATA STATEMENT
160 SYS WARM:	[RETURN TO DIRECT MODE WITHOUT "READY"
170 END:	["RETURN'S" FROM KEYBOARD ARE DUMPED,
180 :	[AND WILL APPEND DATA STATEMENTS TO
190 :	[THIS PROGRAMM.
200 :	
210 :	[WRITE DATA STATEMENTS ON SCREEN
220 PRINT CLEAR\$:	[CLEAR TV SCREEN, CURSOR "HOME"
230 FOR LOOP=BEGIN TO LAST STEP LINE:	[FOR 17 BYTES, EACH STATEMENT
240 : PRINT NUMBER "DATA ":	[PRINT LINE NUMBER AND KEY WORD.
250 : FOR ADDRESS=LOOP TO LOOP+COUNT:	[FOR EACH BYTE IN THIS LINE
260 : IF ADDRESS > LAST GOTO 310:	[SKIP BYTES BEYOND LAST ADDRESS
270 : BYTE=PEEK(ADDRESS):	[GET BYTE,
280 : BYTE\$=STR\$(BYTE):	[PUT IN STRING,
290 : BYTE\$=MID\$(BYTE\$, TWO):	[AND REMOVE LEADING BLANK.
300 : PRINT BYTE\$+TERM\$:	[PRINT BYTE AND TERMINATOR (,)
310 : NEXT:	[GO DO NEXT BYTE FOR THIS LINE.
320 : PRINT BACK\$+BLANK\$:	[REMOVE LAST TERMINATOR
330 : NUMBER=NUMBER+OFFSET:	[SET NEXT LINE NUMBER,
340 NEXT:	[AND GO DO NEXT LINE
350 RETURN	
360 :	
370 :	[PREFIX KEYBOARD RECEIVE BUFFER
380 FOR BUFFER=BUFFER TO BUFFER+FILL:	[FOR 15 LOCATIONS FROM BUFFER,
390 : POKE BUFFER, CARIRET:	[PUT "CRLF" IN BUFFER
400 NEXT:	
410 POKE INDEX, FILL:	[SET "15 BYTES IN BUFFER"
420 RETURN	
430 :	
440 :	[INITIALIZE VARIABLES
450 ADDRESS = 0:	[CURRENT ADDRESS
460 BYTE = 0:	[CONTENTS OF CURRENT ADDRESS
470 BYTE\$ = "":	[STRING REPRESENTATION OF SAME
480 TWO = 2:	[CONSTANT FOR BYTE EDITING
490 TERM\$ = ", ":	[TERMINATOR BETWEEN BASIC "DATA"
500 LOOP = 0:	[GENERAL INDEX VARIABLE (FOR-NEXT)

6502

PROGRAMMEERTALEN

```
510 LINE = 17: [ NUMBER OF DATA TO FIT ON A LINE
520 COUNT = 16: [ SAME NUMBER, COUNTED FROM ZERO
530 BACK$ = CHR$(157): [ BACK SPACE ( ASCII / CBM )
540 BLANK$ = CHR$(32): [ BLANK ( ASCII )
550 NUMBER = 1000: [ CURRENT LINE NUMBER
560 OFFSET = 10: [ OFFSET TO NEXT LINE NUMBER

570 CLEAR$ = CHR$(147): [ CLEAR SCREEN, CURSOR HOME ( ASCII )
580 BUFFER = 623: [ KEYBOARD RECEIVE BUFFER (ADDRESS)
590 INDEX = 158: [ INDEX INTO KEYBOARD BUFFER
600 CARIRET = 13: [ "CRLF" IN ASCII
610 FILL = 12: [ NUMBER OF CRLF'S TO PUT IN BUFFER
620 MAXIMUM = 204: [ MAXIMUM NUMBER OF BYTES IT CAN DO
630 HOME$ = CHR$(19): [ CURSOR HOME
640 WARM = 50066: [ ADDRESS FOR "END" WITHOUT "READY."
650 BEGIN = 0: [ BEGIN ADDRESS
660 LAST = 0: [ LAST ADDRESS
665 SIZE = 0: [ SIZE OF MACHINE LANGUAGE PROGRAM
670 RETURN
680 :
690 : [ USER INPUT ROUTINE
700 INPUT "BEGIN ADDRESS "; BEGIN: [ GET BEGIN ADDRESS
710 INPUT "LAST ADDRESS "; LAST: [ GET END ADDRESS
720 SIZE = LAST-BEGIN: [ COMPUTE SIZE
730 IF SIZE < MAXIMUM GOTO 770: [ IF NOT TO LARGE, GO RETURN
740 : PRINT: [ ELSE, PRINT "CRLF"
750 : PRINT "CAN'T FIX ON SCREEN": [ PRINT ERROR MESSAGE,
760 : GOTO 700: [ REPEAT UNTIL PROPER INPUT
770 RETURN
READY.
```

COMPRESSED FORM

```
100 GOSUB450:PRINTCL$:GOSUB700:GOSUB380:GOSUB220:PRINTHO$):SYSWA:END
220 PRINTCL$):FORLO=BETOLASTEPLI:PRINTNU"DATA "):FORAD=LOTOLO+CO:IFAD>LAGOTO310
270 BY=PEEK(AD):BY$=STR$(BY):BY$=MID$(BY$,TW):PRINTBY$+TE$:
310 NEXT:PRINTBA$+BL$:NU=NU+OF:NEXT:RETURN
380 FORBU=BUTOBU+FI:POKEBU,CA:NEXT:POKEIN,FI:RETURN
450 AD=0:BY=0:BY$="":TW=2:TE$="," :LO=0:LI=17:CO=16:BA$=CHR$(157):BL$=CHR$(32):N
U=1000:OF=10:CL$=CHR$(147):BU=623:IN=158:CA=13:FI=12:MA=204:HO$=CHR$(19):WA=5006
6:BE=0:LA=0:SI=0:RETURN
700 INPUT"BEGIN ADDRESS ";BE:INPUT"LAST ADDRESS ";LA:SI=LA-BE:IFSI<MAGOTO770
740 PRINT:PRINT"CAN'T FIX ON SCREEN":GOTO700
770 RETURN
READY.
```

```
*****
LET OP= NIEUW REDAKTIE-ADRES ! !
      zie de colofon
*****
```

6502 SYSTEM SOFTWARE

CALCUL

KIM SOFTWARE LIBRARY

PAGE 01

```
0010: 0000 CALCUL ORG $0000 POCKET CALCULATOR PROGRAM
0020:
0030: * * * THE FOLLOWING PROGRAM IS
0040: * A POCKET CALCULATOR
0050: *
0060: * INPUT/OUTPUT IS EITHER TELETYPE
0070: * OR KIM KEYBOARD AND DISPLAY
0080: *
0090: * INPUT IS GIVEN BY PRESSING THE
0100: * KEYS FOR A DECIMAL NUMBER
0110: * FOLLOWED BY A FUNCTIONKEY
0120: *
0130: * * * FUNCTIONS:
0140: *
0150: * A = + = ADD NUMBER TO RESULT
0160: * B = - = SUBTRACT NUMBER FROM RESULT
0170: * C = * = MULTIPLY RESULT BY NUMBER
0180: * D = / = DIVIDE RESULT BY NUMBER
0190: * E = C = CLEAR INPUT NUMBER
0200: * F = A = CLEAR RESULT
0210: * AD = R = REMAINDER OF LAST DIVISION
0220: * DA = I = NUMBER STORED IN MEMORY
0230: * + = C = NUMBER FROM MEMORY
0240: * PC = % = CALCULATE PERCENTAGE
0250: * GD = C = CLEAR RESULT
0260: *
0270: * THE PROGRAM MAKES USE OF THE
0280: * INTEGER CALCULATION PACKAGE
0290: * 'INTCAL' PARTNO 770110.
0300: * A 3-BYTE VERSION IS USED.
0310: *
0320: * WHENEVER A DIFFERENT
0330: * NUMBER OF BYTES PER
0340: * NUMBER IS REQUIRED, THE
0350: * DEFINITIONS CONTAINING
0360: * SIZE, SIZEA, SIZEB, SIZEC,
0370: * ACCU, ACCMSB, ACCLSB,
0380: * DACCU, DACLSB AND LOADAD
0390: * HAVE TO BE CHANGED TO THE
0400: * PROPER VALUE IN RELATION TO
0410: * SIZE AND THE PROGRAM MUST BE
0420: * REASSEMBLED.
0430: *
0440: * AUTHOR: SIEP DE VRIES
0450: * ----- BRUGSTRAAT 32
0460: * LIMMEN (NH)
0470: * THE NETHERLANDS
0480: *
0490: *
0500: *
```

6502 SYSTEM SOFTWARE

CALCUL KIM SOFTWARE LIBRARY PAGE 02

```

0510:
0520:
0530:
0540: 0000
0550: 0000
0560: 0000
0570: 0000
0580: 0000
0590: 0000
0600: 0000
0610:
0620:
0630:
0640: 0000
0650: 0000
0660: 0000
0670: 0000
0680: 0000
0690: 0000
0700: 0000
0710:
0720:
0730:
0740: 0000
0750: 0000
0760: 0000
0770: 0000
0780: 0000
0790: 0000
0800: 0000
0810: 0000
0820: 0000
0830: 0000
0840: 0000
0850:
0860:
0870:
0880: 0000 D8
0890: 0001 78
0900: 0002 A9 00
0910: 0004 8D FE 17
0920: 0007 8D FA 17
0930: 000A A9 1C
0940: 000C 8D FF 17
0950: 000F 8D FB 17
0960: 0012 A0 00
0970: 0014 A2 74
0980: 0016 20 13 03
0990: 0019 A2 7A
1000: 001B 20 27 03

```

```

*
* * * DEFINITIONS:
*
SIZE * $0003
SIZEA * SIZE -01 ; SIZE - 1
SIZEB * SIZE +03 ; SIZE * 2
SIZEC * SIZE +02 ; SIZE * 2 - 1
PIADAT * $1740 ; PIA TO TEST KIM/TTY
INITRP * $17FE ; INTERRUPT BREAK TRAP
NMITRP * $17FA ; NON-MASKABLE INTERRUPT TRAP
*
* * * GENERAL SUBROUTINES:
*
SCANDS * $1F1F ; DISPLAY DATA ON LED DISPLAYS
GETKEY * $1F6A ; READ DATA FROM HEX KEYPAD
OUTCH * $1EA0 ; PRINT CHARACTER ON TTY
GETCH * $1E5A ; READ CHARACTER FROM TTY
CRLF * $1E2F ; PRINT CARRIAGE RETURN/LINE FEED
PRIBYT * $1E3B ; PRINT BYTE AS 2 DIGITS
MONITR * $1C00
*
* * * CHARACTER SET
*
PLUS * $002B
MIN * $002D
MAAL * $002A
DEEL * $002F
CLRNUM * $0043
CLRALL * $0041
REP * $0052
MEMIN * $0049
MEMOUT * $004F
PRCNT * $0025
VRAAG * $003F
*
* START OF PROGRAM
*
START CLD ; INITIALIZE
SEI
LDAIM MONITR
STA INITRP
STA NMITRP
LDAIM MONITR /
STA INITRP +01
STA NMITRP +01
LDYIM $00 ; RESULT AND
LDXIM NULL ; MEMORY ARE BOTH
JSR LOAD ; ZERO
LDXIM MEMRY
JSR STORE

```

6502 SYSTEM SOFTWARE

CALCUL KIM SOFTWARE LIBRARY PAGE 03 KIM SOFTWARE LIBRARY PAGE 03 CALCUL

1010:	001E	20	00	02	GETINP	JSR	INPUT	; READ NUMBER + FUNCTION
1020:	0021	A2	F9		WEDER	LDXIM	DATA	; ADDRESS OF
1030:	0023	A0	00			LDYIM	#00	; OPERAND
1040:	0025	C9	2B			CMPIM	PLUS	
1050:	0027	D0	05			BNE	NOADD	
1060:					*			
1070:	0029	20	35	03		JSR	ADD	; ADD OPERAND
1080:	002C	B0	40			BCS	PROBLM	; CARRY SET IN ERROR
1090:	002E	C9	2D		NOADD	CMPIM	MIN	
1100:	0030	D0	05			BNE	NOMIN	
1110:					*			
1120:	0032	20	47	03		JSR	SUB	; SUBTRACT OPERAND
1130:	0035	B0	37			BCS	PROBLM	; CARRY CLEAR IS ERROR
1140:	0037	C9	2A		NOMIN	CMPIM	MAAL	
1150:	0039	D0	05			BNE	NOMAAL	
1160:					*			
1170:	003B	20	5F	03		JSR	MPY	; MULTIPLY OPERAND
1180:	003E	B0	2E			BCS	PROBLM	; CARRY SET IS ERROR
1190:	0040	C9	2F		NOMAAL	CMPIM	DEEL	
1200:	0042	D0	05			BNE	NODEEL	
1210:					*			
1220:	0044	20	AC	03		JSR	DVI	; DIVIDE BY OPERAND
1230:	0047	B0	25			BCS	PROBLM	; CARRY CLEAR IS ERROR
1240:	0049	C9	41		NODEEL	CMPIM	CLRALL	
1250:	004B	D0	05			BNE	NOCLER	
1260:	004D	A2	74		CLEAR	LDXIM	NULL	
1270:	004F	20	13	03		JSR	LOAD	; JUST LOAD ZERO
1280:	0052	C9	25		NOCLER	CMPIM	PRCNT	
1290:	0054	D0	0C			BNE	NOPER	
1300:					*			
1310:	0056	20	5F	03		JSR	MPY	; MULTIPLY BY
1320:	0059	B0	13			BCS	PROBLM	; PERCENTAGE
1330:	005B	A2	77			LDXIM	EENHON	
1340:					*			
1350:	005D	20	AC	03		JSR	DVI	; DIVIDE BY
1360:	0060	B0	0C			BCS	PROBLM	; ONE HUNDRED
1370:					*			
1380:	0062	C9	52		NOPER	CMPIM	REP	
1390:	0064	D0	05			BNE	NOREP	
1400:	0066	A2	85			LDXIM	DACCU	
1410:	0068	20	13	03		JSR	LOAD	
1420:	006B	4C	1E	00	NOREP	JMP	GETINP	
1430:	006E	20	A8	02	PROBLM	JSR	ERROR	
1440:	0071	4C	1E	00		JMP	GETINP	
1450:								
1460:								
1470:								
1480:								
1490:								
1500:								

6502 SYSTEM SOFTWARE

CALCUL KIM SOFTWARE LIBRARY

PAGE 04

```

1510:
1520:
1530:      * * * PAGE ZERO DEFINITIONS:
1540:      *
1550: 0074 00      NULL      =      $00
1560: 0075 00      =      $00
1570: 0076 00      =      $00
1580: 0077 00      EENHON   =      $00
1590: 0078 01      =      $01
1600: 0079 00      =      $00
1610: 007A 00      MEMRY    =      $00
1620: 007B 00      =      $00
1630: 007C 00      =      $00
1640: 007D 00      NUMCAR   =      $00
1650: 007E 00      PREV     =      $00
1660: 007F 00      ACCU     =      $00
1670: 0080      ACCMSB *      ACCU      -03 ; ACCU - SIZE
1680: 0080      ACCLSB *      ACCU      +03 ; ACCU + SIZE
1690: 0080      DACCU  *      ACCU      +06 ; ACCU + 2 * SIZE
1700: 0080      DACLSB *      DACCU     +02 ; DACCU + SIZE - 1
1710: 0080      LOADAD *      DACCU     +06 ; DACCU + 2 * SIZE
1720: 0080      ACCSAV *      LOADAD   +02
1730: 0080      MULIND *      ACCSAV   +01
1740: 0080      MULCNT *      MULIND   +01
1750: 0080      COUNT  *      MULCNT   +01
1760: 0080      DATA  *      $00F9
1770:
1780: 0200      ORG      $0200
1790:
1800:      * * * SUBROUTINE TO INPUT
1810:      *      DATA + FUNCTION
1820:      *
1830:      *      THE RESULT IS SHOWN,
1840:      *      A NUMBER CAN BE KEYED IN
1850:      *      FOLLOWED BY A NON-NUMERICAL
1860:      *      KEY, WHICH CAUSES A RETURN
1870:      *
1880:      * * * CLEAR INPUT, STORE IN MEMORY
1890:      *      AND GET FROM MEMORY ARE
1900:      *      PROCESSED IN THIS SUBROUTINE
1910:      *
1920: 0200 A2 F9      INPUT    LDXIM DATA      ; PERFORM OUTPUT
1930: 0202 A0 00      LDYIM $00      ; FIRST
1940: 0204 20 27 03      JSR      STORE
1950: 0207 20 92 02      WERONP JSR      OUTPUT
1960: 020A A9 00      LDAIM $00      ; CLEAR DIGIT-
1970: 020C 85 7D      STA      NUMCAR     ; COUNTER
1980: 020E 20 71 02      WERINP JSR      INCHAR   ; READ NEXT CHARACTER
1990: 0211 C9 43      CMPIM CLRNUM    ; CLEAR NUMBER
2000: 0213 F0 EB      BEQ      INPUT      ; RE-DISPLAY RESULT

```

6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 05

```

2010: 0215 C9 49          CMPIM MEMIN
2020: 0217 D0 0F          BNE  NOMEMI
2030: 0219 A5 F9          LDA  DATA          ; MEMORY IN
2040: 021B 85 7A          STA  MEMRY          ; DISPLAY NUMBER
2050: 021D A5 FA          LDA  DATA  +01
2060: 021F 85 7B          STA  MEMRY  +01
2070: 0221 A5 FB          LDA  DATA  +02
2080: 0223 85 7C          STA  MEMRY  +02
2090: 0225 4C 07 02      JMP  WERONP
2100: 0228 C9 4F          NOMEMI CMPIM MEMOUT ; NUMBER COMES
2110: 022A D0 0F          BNE  TESDEC        ; FROM MEMORY
2120: 022C A5 7A          LDA  MEMRY          ; DISPLAY IT
2130: 022E 85 F9          STA  DATA
2140: 0230 A5 7B          LDA  MEMRY  +01
2150: 0232 85 FA          STA  DATA  +01
2160: 0234 A5 7C          LDA  MEMRY  +02
2170: 0236 85 FB          STA  DATA  +02
2180: 0238 4C 07 02      JMP  WERONP
2190: 023B C9 30          TESDEC CMPIM '0
2200: 023D 10 01          BPL  GODEC
2210: 023F 60          NODEC  RTS          ; LEAVE CHARACTER UNKNOWN
2220: 0240 C9 3A          GODEC  CMPIM $3A
2230: 0242 10 FB          BPL  NODEC
2240: 0244 A6 7D          LDX  NUMCAR        ; IT IS DECIMAL
2250: 0246 D0 08          BNE  NOTFRS
2260: 0248 A0 00          LDYIM $00          ; FIRST DIGIT CLEARS
2270: 024A 84 F9          STY  DATA          ; NUMBER
2280: 024C 84 FA          STY  DATA  +01
2290: 024E 84 FB          STY  DATA  +02
2300: 0250 38          NOTFRS SEC
2310: 0251 E9 30          SBCIM '0          ; OBTAIN VALUE
2320: 0253 2A          ROLA          ; DIGIT LEFT POSITIONED
2330: 0254 2A          ROLA          ; IN ACCUMULATOR
2340: 0255 2A          ROLA
2350: 0256 2A          ROLA
2360: 0257 A2 04          LDXIM $04
2370: 0259 2A          ROLWER ROLA        ; SHIFT DIGIT IN
2380: 025A 26 F9          ROL  DATA
2390: 025C 26 FA          ROL  DATA  +01
2400: 025E 26 FB          ROL  DATA  +02
2410: 0260 CA          DEX
2420: 0261 D0 F6          BNE  ROLWER
2430: 0263 E6 7D          INC  NUMCAR
2440: 0265 A5 7D          LDA  NUMCAR
2450: 0267 C9 07          CMPIM $07          ; TEST IF TOO MANY DIGITS
2460: 0269 30 A3          BMI  WERINP
2470: 026B 20 A8 02      JSR  ERROR
2480: 026E 4C 00 02      JMP  INPUT
2490:
2500:

```

6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 06

```

2510:      *
2520:      * * * SUBROUTINE TO READ A
2530:      *      CHARACTER FROM KIM OR TTY
2540:      *      OUTPUT IS IN ASCII
2550:      *
2560: 0271 A9 01      INCHAR LDAIM $01
2570: 0273 2C 40 17      BIT      PIADAT      ; TEST IF KIM
2580: 0276 F0 16      BEQ      TTYIN
2590: 0278 20 1F 1F      PUNTA   JSR      SCANDS      ; WAIT FOR NO KEY
2600: 027B D0 FB      BNE      PUNTA
2610: 027D 20 1F 1F      PUNTB   JSR      SCANDS      ; WAIT FOR A KEY
2620: 0280 F0 FB      BEQ      PUNTB
2630: 0282 20 6A 1F      JSR      GETKEY
2640: 0285 C9 15      CMPIM  $15      ; IGNORE IT IF
2650: 0287 10 EF      BPL      PUNTA      ; BOUNCE
2660: 0289 AA      TAX
2670: 028A BD FD 03      LDAAX  ASCTAB      ; CONVERT TO ASCII
2680: 028D 60      RTS
2690: 028E 20 5A 1E      TTYIN   JSR      GETCH      ; TTY IS EASIER
2700: 0291 60      RTS
2710:      *
2720:      * * * OUTPUT NUMBER FROM
2730:      *      DATA AREA TO TTY IF CONNECTED
2740:      *
2750: 0292 A9 01      OUTPUT LDAIM $01
2760: 0294 2C 40 17      BIT      PIADAT
2770: 0297 F0 01      BEQ      ITSTTY
2780: 0299 60      RTS
2790: 029A A2 02      ITSTTY LDXIM $02
2800: 029C B5 F9      WTYP   LDAAX  DATA
2810: 029E 20 3B 1E      JSR      PRIBYT
2820: 02A1 CA      DEX
2830: 02A2 10 FB      BPL      WTYP
2840: 02A4 20 2F 1E      OUTYP  JSR      CRLF
2850: 02A7 60      RTS
2860:      *
2870:      * * * ERROR SUBROUTINE
2880:      *
2890: 02A8 A9 01      ERROR  LDAIM $01
2900: 02AA 2C 40 17      BIT      PIADAT
2910: 02AD F0 0A      BEQ      TTYERR
2920: 02AF A2 02      LDXIM  SIZEA
2930: 02B1 A9 FF      LDAIM  $FF
2940: 02B3 95 7F      SETFUL STAAX  ACCU
2950: 02B5 CA      DEX
2960: 02B6 10 FB      BPL      SETFUL
2970: 02B8 60      RTS
2980: 02B9 A9 3F      TTYERR LDAIM  VRAAG
2990: 02BB 20 A0 1E      JSR      OUTCH
3000: 02BE 4C A4 02      JMP      OUTYP

```

6502

SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 07

```
3010: 0300          ORG    #0300
3020:          * * * INTEGER PACKAGE 'INTCAL' *
3030:          *          #770110 *
3040:          *
3050:          *          COPYRIGHT (C) 1977, 1982 *
3060:          *          WESTVRIES COMPUTER CONSULTING B.V. *
3070:          *          P.O. BOX 20 *
3080:          *          OOSTZAAN *
3090:          *          THE NETHERLANDS *
3100:          *
3110:          * * * FUNCTIONS PROVIDED: *
3120:          *          LOAD, STORE, ADD, SUBTRACT, *
3130:          *          MULTIPLY, DIVIDE *
3140:          *
3150: 0300 86 8B  BEGALG STX  LOADAD ; GENERAL SETUP AND
3160: 0302 84 8C          STY  LOADAD +01 ; INITIALIZE
3170: 0304 85 8D          STA  ACCSAV
3180: 0306 A2 02          LDXIM SIZEA
3190: 0308 A0 00          LDYIM $00
3200: 030A 94 82  CLRWER STYZX ACCLSB
3210: 030C CA          DEX
3220: 030D 10 FB          BPL  CLRWER
3230: 030F A2 02          LDXIM SIZEA
3240: 0311 F8          SED
3250: 0312 60          RTS
3260: 0313 20 00 03  LOAD  JSR  BEGALG ; LOAD INTO
3270: 0316 B1 8B  NLOAD LDAYI LOADAD ; ACCU
3280: 0318 99 7F 00          STAYI ACCU
3290: 031B C8          INY
3300: 031C CA          DEX
3310: 031D 10 F7          BPL  NLOAD
3320: 031F D8  ALGOUT CLD ; GENERAL EXIT
3330: 0320 A6 8B          LDX  LOADAD ; RESTORE REGISTERS
3340: 0322 A4 8C          LDY  LOADAD +01
3350: 0324 A5 8D          LDA  ACCSAV
3360: 0326 60          RTS
3370: 0327 20 00 03  STORE JSR  BEGALG ; STORE FROM
3380: 032A B9 7F 00  NSTOR LDAYI ACCU ; ACCU
3390: 032D 91 8B          STAYI LOADAD
3400: 032F C8          INY
3410: 0330 CA          DEX
3420: 0331 10 F7          BPL  NSTOR
3430: 0333 30 EA          BMI  ALGOUT
3440: 0335 20 00 03  ADD   JSR  BEGALG ; ADD INTO
3450: 0338 18          CLC ; ACCU
3460: 0339 B9 7F 00  NADD  LDAYI ACCU
3470: 033C 71 8B          ADCIY LOADAD
3480: 033E 99 7F 00          STAYI ACCU
3490: 0341 C8          INY
3500: 0342 CA          DEX
```

6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY PAGE 08 KIM SOFTWARE LIBRARY PAGE 08

3510:	0343	10	F4		BPL	NADD		
3520:	0345	30	D8		BMI	ALGOUT		
3530:	0347	20	00	03	SUB	JSR	BEGALG	; SUBTRACT FROM
3540:	034A	38			SEC			; ACCU
3550:	034B	B9	7F	00	NSUB	LDAAY	ACCU	
3560:	034E	F1	8B		SBCIY	LOADAD		
3570:	0350	99	7F	00		STAYY	ACCU	
3580:	0353	C8			INY			
3590:	0354	CA			DEX			
3600:	0355	10	F4		BPL	NSUB		
3610:	0357	B0	03		BCS	CLICAR		
3620:	0359	38			SETCAR	SEC		
3630:	035A	B0	C3		BCS	ALGOUT		
3640:	035C	18			CLICAR	CLC		
3650:	035D	90	C0		BCC	ALGOUT		
3660:	035F	8E	8B		MPY	STX	LOADAD	
3670:	0361	84	8C			STY	LOADAD	+01
3680:	0363	85	8D			STA	ACCSAV	
3690:	0365	A2	06			LDXIM	SIZEB	
3700:	0367	B5	7E		WIMPY	LDAAX	ACCU	-01
3710:	0369	95	84			STAAX	DACCU	-01
3720:	036B	A9	00			LDAIM	*00	
3730:	036D	95	7E			STAAX	ACCU	-01
3740:	036F	CA				DEX		
3750:	0370	D0	F5			BNE	WIMPY	
3760:	0372	A8				TAY		
3770:	0373	A9	03			LDAIM	SIZE	
3780:	0375	85	8F			STA	MULCNT	
3790:	0377	F8				SED		
3800:	0378	B1	8B		OUTLUP	LDAIY	LOADAD	
3810:	037A	85	8E			STA	MULIND	
3820:	037C	F0	1C			BEQ	NOAD	
3830:	037E	18			PEUTER	CLC		
3840:	037F	A9	06			LDAIM	SIZEB	
3850:	0381	85	90			STA	COUNT	
3860:	0383	A2	00			LDXIM	*00	
3870:	0385	B5	7F		WERADD	LDAAX	ACCU	
3880:	0387	75	85			ADCAX	DACCU	
3890:	0389	95	7F			STAAX	ACCU	
3900:	038B	E8				INX		
3910:	038C	C6	90			DEC	COUNT	
3920:	038E	D0	F5			BNE	WERADD	
3930:	0390	B0	8D		ULGO	BCS	ALGOUT	
3940:	0392	A5	8E			LDA	MULIND	
3950:	0394	E9	00			SBCIM	*00	
3960:	0396	85	8E			STA	MULIND	
3970:	0398	D0	E4			BNE	PEUTER	
3980:	039A	A2	05		NOAD	LDXIM	SIZEC	
3990:	039C	B5	84		WUSTUR	LDAAX	DACCU	-01
4000:	039E	95	85			STAAX	DACCU	

6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY PAGE 09

4010:	03A0	CA		DEX	
4020:	03A1	D0	F9	BNE	WUSTUR
4030:	03A3	86	85	STX	DACCU
4040:	03A5	C8		INY	
4050:	03A6	C6	8F	DEC	MULCNT
4060:	03A8	D0	CE	BNE	OUTLUP
4070:	03AA	F0	B0	PULGO	BEG
4080:	03AC	86	8B	DVI	STX
4090:	03AE	84	8C		STY
4100:	03B0	85	8D		STA
4110:	03B2	A2	03		LDXIM
4120:	03B4	8E	8E		STX
4130:	03B6	A2	05		LDXIM
4140:	03B8	B5	7F	MOVDIV	LDAAX
4150:	03BA	95	85		STAAX
4160:	03BC	A9	00		LDAIM
4170:	03BE	95	7F		STAAX
4180:	03C0	CA		DEX	
4190:	03C1	10	F5		BPL
4200:	03C3	A0	02	DIVLUP	LDYIM
4210:	03C5	A6	8E		LDX
4220:	03C7	B5	87	TESWER	LDAAX
4230:	03C9	D1	8B		CMPIY
4240:	03CB	90	29		BCC
4250:	03CD	B5	87		LDAAX
4260:	03CF	F0	21		BEG
4270:	03D1	A6	8E		LDX
4280:	03D3	A0	00		LDYIM
4290:	03D5	A9	03		LDAIM
4300:	03D7	85	8F		STA
4310:	03D9	F8			SED
4320:	03DA	B5	85	DIVRE	LDAAX
4330:	03DC	F1	8B		SBCIY
4340:	03DE	95	85		STAAX
4350:	03E0	E8			INX
4360:	03E1	C8			INY
4370:	03E2	C6	8F		DEC
4380:	03E4	D0	F4		BNE
4390:	03E6	B5	7C		LDAAX
4400:	03E8	C9	99		CMPIY
4410:	03EA	B0	A4		BCS
4420:	03EC	69	01		ADCIM
4430:	03EE	95	7C		STAAX
4440:	03F0	D0	D1		BNE
4450:	03F2	CA		NOCOMP	DEX
4460:	03F3	88			DEY
4470:	03F4	10	D1		BPL
4480:	03F6	C6	8E	FINDIV	DEC
4490:	03F8	10	C9		BPL
4500:	03FA	4C	5C	03	JMP

6502 SYSTEM SOFTWARE

KIM SOFTWARE LIBRARY

PAGE 10

4510:	03FD	30	ASCTAB =	'0	:	0
4520:	03FE	31	=	'1	:	1
4530:	03FF	32	=	'2	:	2
4540:	0400	33	=	'3	:	3
4550:	0401	34	=	'4	:	4
4560:	0402	35	=	'5	:	5
4570:	0403	36	=	'6	:	6
4580:	0404	37	=	'7	:	7
4590:	0405	38	=	'8	:	8
4600:	0406	39	=	'9	:	9
4610:	0407	2B	=	PLUS	:	A
4620:	0408	2D	=	MIN	:	B
4630:	0409	2A	=	MAAL	:	C
4640:	040A	2F	=	DEEL	:	D
4650:	040B	43	=	CLRNUM	:	E
4660:	040C	41	=	CLRALL	:	F
4670:	040D	52	=	REP	:	AD
4680:	040E	49	=	MEMIN	:	DA
4690:	040F	4F	=	MEMOUT	:	PC
4700:	0410	25	=	PRCNT	:	+
4710:	0411	41	=	CLRALL	:	GO

SYMBOL TABLE D000 D270

ACCLSB 0082	ACCMSB 007C	ACCSAV 008D	ACCU 007F
ADD 0335	ALGOUT 031F	ASCTAB 03FD	BEGALG 0300
CALCUL 0000	CLEAR 004D	CLICAR 035C	CLRALL 0041
CLRNUM 0043	CLRWER 030A	COUNT 0090	CRLF 1E2F
DACCU 0085	DACLSB 0087	DATA 00F9	DEEL 002F
DIVLUP 03C3	DIVRE 03DA	DVI 03AC	EENHON 0077
ERROR 02A8	FINDIV 03F6	GETCH 1E5A	GETINP 001E
GETKEY 1F6A	GODEC 0240	INCHAR 0271	INITRP 17FE
INPUT 0200	ITSTTY 029A	LOAD 0313	LOADAD 008B
MAAL 002A	MEMIN 0049	MEMOUT 004F	MEMRY 007A
MIN 002D	MONITR 1C00	MOVDIV 03B8	MPY 035F
MULCNT 008F	MULIND 008E	NADD 0339	NLOAD 0316
NMITRP 17FA	NOAD 039A	NOADD 002E	NOCLER 0052
NOCOMP 03F2	NODEC 023F	NODEEL 0049	NOMAAL 0040
NOMEMI 0228	NOMIN 0037	NOPER 0062	NOREP 006B
NOTFRS 0250	NSTOR 032A	NSUB 034B	NULL 0074
NUMCAR 007D	OUTCH 1EA0	OUTLUP 0378	OUTPUT 0292
OUTYP 02A4	PEUTER 037E	PIADAT 1740	PLUS 002B
PRCNT 0025	PREV 007E	PRIBYT 1E3B	PROBLM 006E
PULGO 03AA	PUNTA 0278	PUNTB 027D	REP 0052
ROLWER 0259	SCANDS 1F1F	SETCAR 0359	SETFUL 02B3
SIZE 0003	SIZEA 0002	SIBZ 0006	SIZED 0005
START 0000	STORE 0327	SUB 0347	TESDEC 023B
TESWER 03C7	TTYERR 02B9	TTYIN 028E	ULGO 0390
VRAAG 003F	WEDER 0021	WERADD 0385	WERINP 020E
WERONP 0207	WIMPY 0367	WTYP 029C	WUSTUR 039C

6502 SYSTEM SOFTWARE

SYMBOL TABLE D000 D270

CALCUL 0000	START 0000	SIZEA 0002	SIZE 0003
SIZEC 0005	SIZED 0006	GETINP 001E	WEDER 0021
PRCNT 0025	MAAL 002A	PLUS 002B	MIN 002D
NOADD 002E	DEEL 002F	NOMIN 0037	VRAAG 003F
NOMAAL 0040	CLRALL 0041	CLRNUM 0043	MEMIN 0049
NODEEL 0049	CLEAR 004D	MEMOUT 004F	NOCLER 0052
REP 0052	NOPER 0062	NOREP 006B	PROBLM 006E
NULL 0074	EENHON 0077	MEMRY 007A	ACCMSB 007C
NUMCAR 007D	PREV 007E	ACCU 007F	ACCLSB 0082
DACCU 0085	DACLSB 0087	LOADAD 008B	ACCSAV 008D
MULIND 008E	MULCNT 008F	COUNT 0090	DATA 00F9
INPUT 0200	WERONP 0207	WERINP 020E	NOMEMI 0228
TESDEC 023B	NODEC 023F	GODEC 0240	NOTFRS 0250
ROLWER 0259	INCHAR 0271	PUNTA 0278	PUNTB 027D
TTYIN 028E	OUTPUT 0292	ITSTTY 029A	WTYP 029C
OUTYP 02A4	ERROR 02A8	SETFUL 02B3	TTYERR 02B9
BEGALG 0300	CLRWER 030A	LOAD 0313	NLOAD 0316
ALGOUT 031F	STORE 0327	NSTOR 032A	ADD 0335
NADD 0339	SUB 0347	NSUB 034B	SETCAR 0359
CLICAR 035C	MPY 035F	WIMPY 0367	OUTLUP 0378
PEUTER 037E	WERADD 0385	ULGO 0390	NOAD 039A
WUSTUR 039C	PULGO 03AA	DVI 03AC	MOVDIV 03B8
DIVLUP 03C3	TESWER 03C7	DIVRE 03DA	NOCOMP 03F2
FINDIV 03FE	ASCTAB 03FD	PIADAT 1740	NMITRP 17FA
INITRP 17FE	MONITR 1C00	CRLF 1E2F	PRIBYT 1E3B
GETCH 1ESA	OUTCH 1EA0	SCANDS 1F1F	GETKEY 1F6A

MARKTINFORM

VOOR DE BASIS VERSIE VAN DE JUNIOR IS ER EEN BUFFER-KAART TE KOOP. DEZE KAART GEEFT EEN VRIJ TE GEBRUIKEN GEHEUGENRUIMTE VAN \$2000 T/M \$F7FF. HET GEDEELTE VAN \$1800 T/M \$1FFF KOMT OOK VOOR OP \$F800 T/M \$FFFF. DIT, IN VERBAND MET DE RESET EN INTERRUPT-ADRESSEN. DE KAART KOST GEBOUWD EN GETEST FL. 112,==. INLICHTINGEN BIJ KOOS DE HAAN (015-568871).

.....
 LET OP: NIEUW REDAKTIE-ADRES ! !
 zie de colofon

6502 SYSTEEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240182 PAGE 01

```

0010: ***** PATCHES OP MICRO-ADE *****:
0020: ** VERVOLG DEEL 11 ** :
0030: :
0040: Door: A.S. Hankel :
0050: W. Kloosstraat 32 :
0060: 7606 BB Almelo :
0070: Tel: 05490-51151 :
0080: :
0090: Helaas, eerst een bus. In de Patches op Micro-Ade :
0100: deel 11 komt een patch voor welke onjuist is. De :
0110: aanpassing om een backspace-karakter te genereren :
0120: staat op een verkeerd adres. Het moet als volgt: :
0130: :
0140: 2DE3 ORG $2DE3 :
0150: :
0160: 2DE3 A9 08 LDAIM $08 ASCII-BACKSPACE :
0170: :
0180: ---- FILE 0A ---- :
0190: :
0200: DOEL: AANPASSING SAL/SAH, EAL/EAH I.V.M. :
0210: READ/WRITE-PROCEDURES. :
0220: :
0230: 70 1A SAL * $1A70 :
0240: 71 1A SAH * $1A71 :
0250: 72 1A EAL * $1A72 :
0260: 73 1A EAH * $1A73 :
0270: :
0280: SAL MOET WORDEN INGEVULD OP DE VOLDENDE :
0290: ADRESSEN: :
0300: $2106, $21EE, $268D, $382B, $389F. :
0310: :
0320: SAH MOET WORDEN INGEVULD OP DE VOLGENDE :
0330: ADRESSEN: :
0340: $21D0, $21F3, $26C3, $3830, $38A4. :
0350: :
0360: EAL MOET WORDEN INGEVULD OP DE VOLGENDE :
0370: ADRESSEN: :
0380: $21CB, $26CA, $2F75, $3848, $38AC, $38D5, $38EF. :
0390: :
0400: EAH MOET WORDEN INGEVULD OP DE VOLGENDE :
0410: ADRESSEN: :
0420: $21D5, $26D1, $2F7B, $384F, $38B3, $38DC, $38FE. :
0430: :
0440: ---- FILE 0B ---- :
0450: :
0460: DOEL: BIJ DE KIM WORDEN PB0 EN PB1 GEBRUIKT OM DE :
0470: RECORDERS TE SCHAKELEN. BIJ DE JUNIOR ZIJN :
0480: DIT PB5 EN PB6. :
0490: :
0500: 83 1A PBDD * $1A83 :

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 230182 PAGE 02

```
0510:      :
0520:      : PATCH OM PB5 EN PB6 ALS UITGANG TE DEFINIEREN.
0530:      :
0540: 3040      ORG    $3040
0550:      :
0560: 3040 A9 60  TAPSET LDAIM $60    PB5 + PB6 UITGANG
0570: 3042 0D 83 1A  ORA    PBDD
0580: 3045 8D 83 1A  STA    PBDD
0590: 3048 EA      NOP          DE REST NOPJES
0600: 3049 EA      NOP
0610: 304A EA      NOP
0620: 304B EA      NOP
0630: 304C EA      NOP
0640: 304D EA      NOP
0650: 304E EA      NOP
0660: 304F EA      NOP
0670:      :
0680:      :
0690:      : AND EN ORA WAARDEN T.B.V. PB5 EN PB6
0700:      :
0710: 3053      ORG    $3053
0720: 3053 A9 DF      LDAIM $DF
0730:      :
0740: 3054      ORG    $3054
0750: 3054 A9 20      LDAIM $20
0760:      :
0770: 305F      ORG    $305F
0780: 305F A9 BF      LDAIM $BF
0790:      :
0800: 307A      ORG    $307A
0810: 307A A9 40      LDAIM $40
0820:      :
0830: 38FD      ORG    $38FD
0840: 38FD A9 DF      LDAIM $DF
0850:      :
0860: 38FC      ORG    $38FC
0870: 38FC A9 BF      LDAIM $BF
0880:      :
0890:      : ---- FILE 00 ----
0900:      :
0910:      : DOEL: AANPASSING READ/WRITE PROCEDURES.
0920:      :
0930:      : DE R/W-ROUTINES BINNEN M.A. ROEPEN SUBROUTINES
0940:      : AAN UIT DE KIM-MONITOR-R/W-PROCEDURES.
0950:      : ER IS ECHTER EEN WEZENLIJK VERSCHIL TUSSEN DE R/W
0960:      : ROUTINES VAN DE KIM EN DE JUNIOR. BIJ DE JUNIOR
0970:      : WORDT BIJ HET INLEZEN EN WEGSCHRIJVEN EEN INDEX-Y
0980:      : POINTER GEBRUIKT. BIJ DE KIM VINDT DIT PLAATS VIA
0990:      : DE ROUTINE VEB. EEN SUBR. DIE EEN LDA OF STA INSTRUK-
1000:      : TIE EN EEN ADRESPOINTER BEVAT.
```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 230182 PAGE 03

```

1010:
1020:      00 1A VEB *      $1A00
1030:      04 1A XSAVE *    $1A04  TEMPORARY FOR X-REG.
1040:      6E 1A CHKL *    $1A6E
1050:      6F 1A CHKH *    $1A6F
1060:      70 1A SAL *      $1A70  STARTADDRESS LOW
1070:      71 1A SAH *      $1A71  STARTADDRESS HIGH
1080:      72 1A EAL *      $1A72
1090:      73 1A EAH *      $1A73
1100:      83 1A PBDD *     $1A83
1110:      38 3F PACKT *    $3F38
1120:
1130:      : READCH LEEST 1 CHAR. VAN TAPE
1140:
1150:      36 0C READCH *    $0C36
1160:
1170:      : INTVEB VERPLAATST SAL/SAH NAAR VEB +1,2
1180:
1190: 3F58          ORG      $3F58
1200: 3F58 AD 70 1A INTVEB LDA SAL
1210: 3F5B 8D 01 1A      STA VEB +01
1220: 3F5E AD 71 1A      LDA SAH
1230: 3F61 8D 02 1A      STA VEB +02
1240: 3F64 A9 60          LDAIM $60  SET RTS-INSTRUKTIE
1250: 3F66 8D 03 1A      STA VEB +03
1260: 3F69 A9 00          LDAIM $00  CLEAR CHECKSUM
1270: 3F6B 8D 6E 1A      STA CHKL
1280: 3F6E 8D 6F 1A      STA CHKH
1290: 3F71 60          RTS
1300:
1310:      : INCVEB VERHOOGT VEB +01,02
1320:
1330: 3F72 EE 01 1A INCVEB INC VEB +01
1340: 3F75 D0 03          BNE INCVEI
1350: 3F77 EE 02 1A      INC VEB +02
1360: 3F7A 60          INCVEI RTS
1370:
1380:      : COMPUTE CHECKSUM VERHOOGT CHKL/CHKH
1390:
1400: 3F7B A8          CHKT TAY          SAVE ACCU
1410: 3F7C 18          CLC
1420: 3F7D 6D 6E 1A      ADC CHKL  TEL CHKL ER BIJ OP
1430: 3F80 8D 6E 1A      STA CHKL  EN STORE
1440: 3F83 AD 6F 1A      LDA CHKH  HAAL CHKH
1450: 3F86 69 00          ADCIM $00  TEL CARRY-FLAG ER BIJ OP
1460: 3F88 8D 6F 1A      STA CHKH  EN STORE
1470: 3F8B 98          TYA
1480: 3F8C 60          RTS
1490:
1500:

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240192 PAGE 04

```

1510:          : DE JUNIOR KENT EVENALS DE KIM EEN ROUTINE RDCHT.
1520:          : BIJ DE JUNIOR GAAT HET X-REG. VERLOREN. DE ROUTINE
1530:          : WORDT DUS:
1540:          :
1550: 3F8D 8E 04 1A  RDCHT  STX   XSAVE  BEWAAR X-WAARDE
1560: 3F90 20 3E 0C          JSR   READCH
1570: 3F93 AE 04 1A          LDX   XSAVE  HERSTEL X-WAARDE
1580: 3F9E 60          RTS
1590:          :
1600:          : READBYTE LEEST 1 BYTE VAN TAPE
1610:          :
1620: 3F97 20 8D 3F  RDBYT  JSR   RDCHT
1630: 3F9A 20 38 3F          JSR   PACKT
1640: 3F9D 20 8D 3F          JSR   RDCHT
1650: 3FA0 20 38 3F          JSR   PACKT
1660: 3FA3 60          RTS
1670:          :
1680:          : GEPATCHTE WRITE-PROCEDURE:
1690:          :
1700: 2F35          ORG   $2F35
1710:          :
1720: 2F35 4C 53 30  CWRITE JMP   $3053  INITIEER PBS + DELAY
1730: 2F38 AD 83 1A          LDA   PBDD  PB7 UITGANG
1740: 2F3B 09 80          DRAIN $80
1750: 2F3D 8D 83 1A          STA   PBDD
1760: 2F40 EA          NOP
1770: 2F41 EA          NOP
1780: 2F42 A9 AD          LDAIM $AD   INITIEER VEB
1790: 2F44 8D 00 1A          STA   VEB   VOOR SCHRIJVEN
1800: 2F47 20 58 3F          JSR   INTVEB
1810: 2F4A A9 47          LDAIM $47   INITIEER FLAG
1820: 2F4C 85 F0          STAZ  $00F0 VOOR PBDD
1830: 2F4E A2 F0          LDXIM $F0   240 SYNCHR. KAR.
1840: 2F50 A9 1E          LDAIM $1E   SYNCHRONISATIEKARAKTER
1850: 2F52 20 A3 2F          JSR   HIC   SCHRIJF ZE WEG
1860: 2F55 A9 2A          LDAIM $2A
1870: 2F57 20 C6 2F          JSR   OUTCHT SCHRIJF DATASTART-KAR.
1880: 2F5A A5 62          LDAZ  $0062 HAAL FILE-ID
1890: 2F5C 20 B2 2F          JSR   OUTBT  SCHRIJF WEG
1900: 2F5F A5 60          LDAZ  $0060 EXECUTIE-ADRES
1910: 2F61 20 AF 2F          JSR   OUTBTC SCHRIJF WEG
1920: 2F64 A5 61          LDAZ  $0061 IDEM HOGE GEDEELTE
1930: 2F66 20 AF 2F          JSR   OUTBTC
1940: 2F69 20 00 1A  DUMPTA JSR   VEB   HAAL BYTE UIT GEHEUGEN
1950: 2F6C 20 AF 2F          JSR   OUTBTC EN SCHRIJF WEG
1960: 2F6F 20 72 3F          JSR   INCVEB VERHOOG VEB
1970: 2F72 AD 01 1A          LDA   VEB   +01 ALLES GEHAD?
1980: 2F75 CD 72 1A          CMP   EAL
1990: 2F78 AD 02 1A          LDA   VEB   +02
2000: 2F7B ED 73 1A          SBC   EAH

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240182 PAGE 05

```

2010: 2F7E 90 E9          BCC  DUMPTA:ZONEE, DOORGAAN
2020: 2F80 A9 2F          LDAIM $2F EINDDATA-KARAKTER
2030: 2F82 20 C6 2F      JSR  OUTCHT SCHRIJF WEG
2040: 2F85 AD 6E 1A      LDA  CHKL 3 SCHRIJF CHECKSUM
2050: 2F88 20 B2 2F      JSR  OUTBT
2060: 2F8B AD 6F 1A      LDA  CHKH30* OUTCHT LDYIM
2070: 2F8E 20 B2 2F      JSR  OUTBT
2080: 2F91 A2 02          LDXIM $02 2 KEER
2090: 2F93 A9 04          LDAIM $04 EDT-KARAKTER
2100: 2F95 20 A3 2F      JSR  HIC 3 SCHRIJF WEG
2110: 2F98 4C 61 30      JMP  $3061 EN SCHAKEL PB5
2120: 2F9B EA            NOP
2130: 2F9C EA            NOP
2140: 2F9D EA            NOP
2150: 2F9E EA            NOP
2160: 2F9F EA            NOP
2170: 2FA0 EA            NOP
2180: 2FA1 EA            NOP
2190: 2FA2 60          RTS  EINDE SCHRIJF-PROCEDURE
2200:
2210:
2220:
2230: 2FA3 86 F1          HIC  STXZ $00F1 BEWAAR X
2240: 2FA5 48            HICA  PHA  EN HET KARAKTER
2250: 2FA6 20 C6 2F      JSR  OUTCHT SCHRIJF WEG
2260: 2FA9 68            PLA  EN HAAL TERUG
2270: 2FAA C6 F1          DECZ $00F1 INDEX OMLAAG
2280: 2FAC D0 F7          BNE  HICA  TOT AAN 0
2290: 2FAE 60          RTS
2300:
2310:
2320:
2330: 2FAF 20 7B 3F      OUTBTC JSR  CHKT  KORRIGEER CHECKSUM
2340:
2350:
2360:
2370: 2FB2 48            OUTBT PHA  BEWAAR KARAKTER
2380: 2FB3 4A            LSRA  HOOGSTE NIBBLE
2390: 2FB4 4A            LSRA
2400: 2FB5 4A            LSRA
2410: 2FB6 4A            LSRA
2420: 2FB7 20 BB 2F      JSR  HEXT  EN SCHRIJF WEG
2430: 2FBA 68            PLA  HAAL BYTE TERUG
2440:
2450:
2460:
2470: 2FBB 29 0F          HEXT 8 ANDIM $0F HOGE NIBBLE ERAF
2480: 2FBD C9 0A          CMPIM $0A CONVERTEER NAAR ASCII
2490: 2FBF 18            CLC
2500: 2FC0 30 02          BMI  HEXAT

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240182 PAGE 06

```

2510: 2FC2 69 07      ADCIM $07
2520: 2FC4 69 30      HEXAT ADCIM $30
2530:                :
2540:                : SUBR. OM BYTE TE SCHRIJVEN
2550:                :
2560: 2FC6 A0 08      OUTCHT LDYIM $08      8 BITS
2570: 2FC8 84 F2      STYZ  $00F2      BEWAAR Y
2580: 2FCA A0 02      TRY   LDYIM $02      BEGIN
2590: 2FCC 84 FE      STYZ  $00FE      3600 HZ.
2600: 2FCE BE FC 2F    ZON   LDXAY NPUL    AANTAL 'HALF' PER.
2610: 2FD1 48          PHA   BEWAAR KARAKTER
2620: 2FD2 2C F7 1A    ZONA  BIT   $1AF7    WACHT TOT EINDE
2630: 2FD5 10 FB          BPL   ZONA
2640: 2FD7 B9 FD 2F    LDAAY TIMG    INITIEER TIMER
2650: 2FDA 8D F4 1A    STA   $1AF4
2660: 2FDD A5 F0      LDAZ  $00F0      VERANDER
2670: 2FDF 49 80      EDRIM $80      OUTPUT
2680: 2FE1 8D 82 1A    STA   $1A82      POORT
2690: 2FE4 85 F0      STAZ  $00F0      EN BEWAAR STATUS
2700: 2FE6 CA          DEX   ALLES GEHAD
2710: 2FE7 D0 E9      BNE   ZONA      ZONEE, TERUG
2720: 2FE9 68          PLA   HAAL BYTE TERUG
2730: 2FEA C6 FE      DECZ  $00FE      NA 3 KEER
2740: 2FEC F0 05      BEQ   SETZ      OMSCHAKELEN
2750: 2FEE 30 07      BMI   ROUT
2760: 2FF0 4A          LSRA  VOLGENDE BIT
2770: 2FF1 90 DB      BCC   ZON
2780: 2FF3 A0 00      SETZ  LDYIM $00
2790: 2FF5 F0 D7      BEQ   ZON
2800: 2FF7 C6 F2      ROUT  DECZ  $00F2  ALLE BITS GEHAD?
2810: 2FF9 D0 CF      BNE   TRY      ZONEE, DOORGAAN
2820: 2FFB 60          RTS
2830:                :
2840:                : TIMING TABEL
2850:                :
2860: 2FFC 02          NPUL  =   $02
2870: 2FFD C3          TIMG  =   $C3
2880: 2FFE 03          =   $03
2890: 2FFF 7E          =   $7E
2900:                :
2910:                : ---- FILE 0D ----
2920:                :
2930:                : LISTING M.A. READ-ROUTINE
2940:                :
2950:                80 1A PAD * $1A80
2960:                82 1A PBD * $1A82
2970:                C2 0B RDBIT * $0BC2 LEES 1 BIT VAN TAPE
2980:                :
2990: 2EAF          ORG  $2EAF
3000:                :

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240182 PAGE 07

```

3010: 2EAF 4C 6F 30  CREAD  JMP  $306F  INITIEER PB6
3020: 2EB2 20 13 3D  JSR  $3D13  BEWAAR POINTERS
3030: 2EB5 EA      NOP
3040: 2EB6 EA      NOP
3050: 2EB7 EA      NOP
3060: 2EB8 A9 7F  LDAlM $7F  INITIEER POORT B
3070: 2EBA 8D 83 1A  STA  PBDD
3080: 2EBD D8      SYNC  CLD
3090: 2EBE 20 48 3D  JSR  $3D48  VOOR ALLE ZEKERHEID
3100: 2EC1 EA      NOP
3110: 2EC2 20 6A 3F  JSR  INTVEB +12 GED. INITIERING
3120: 2EC5 A9 33  LDAlM $33  INITIEER PBD
3130: 2EC7 8D 82 1A  STA  PBD
3140: 2ECA 20 C2 0B  INBIT JSR  RDBIT  LEES EEN BIT
3150: 2ECD 66 F3  RORZ $00F3  SCHUIF IN TEMPORARY
3160: 2ECF EA      NOP
3170: 2ED0 EA      NOP
3180: 2ED1 A5 F3  LDAlZ $00F3  ZET OP DISPLAY
3190: 2ED3 8D 80 1A  STA  PAD
3200: 2ED6 C9 16  TST  CMPIM $16  SYNC. KAR?
3210: 2ED8 D0 F0  BNE  INBIT  ZONEE, BLIJF ZOEKEN
3220: 2EDA 20 8D 3F  JSR  RDCHT  LEES KARAKTER
3230: 2EDD 8D 80 1A  STA  PAD  EN DISPLAY
3240: 2EE0 C9 2A  CMPIM $2A  START VAN DATA?
3250: 2EE2 D0 F2  BNE  TST  ZONEE, BLIJF ZOEKEN
3260: 2EE4 20 F0 32  JSR  $32F0  CONTROLEER ID
3270: 2EE7 EA      NOP
3280: 2EE8 EA      NOP
3290: 2EE9 D0 C1  BNE  $2EAC  PRINT FOUTIEVE ID
3300: 2EEB 20 97 3F  JSR  RDBYT  LEES STARTADRES LOW
3310: 2EEE 20 7B 3F  JSR  CHKT  BIJ CHECKSUM
3320: 2EF1 EA      NOP
3330: 2EF2 EA      NOP
3340: 2EF3 EA      NOP
3350: 2EF4 20 97 3F  JSR  RDBYT  STARTADRES HIGH
3360: 2EF7 20 7B 3F  JSR  CHKT  BIJ CHECKSUM
3370: 2EFA EA      NOP
3380: 2EFB EA      NOP
3390: 2EFC EA      NOP
3400: 2EFD A2 02  LOADIT LDXIM $02  LEES DATA
3410: 2EFF 20 8D 3F  READIT JSR  RDCHT  LEES EEN KARAKTER
3420: 2F02 C9 2F  CMPIM $2F  END-OF-DATA KARAKTER?
3430: 2F04 F0 14  BEQ  ENDRD  ZOJA, STOP LEZEN
3440: 2F06 20 94 2E  JSR  $2E94  PACKT VIA OMWEG
3450: 2F09 D0 B2  BNE  SYNC  LEESFOUT
3460: 2F0B CA      DEX
3470: 2F0C D0 F1  BNE  READIT  LEES 2E HELFT
3480: 2F0E 20 7B 3F  JSR  CHKT  BIJ CHECKSUM
3490: 2F11 20 00 1A  JSR  VEB  NAAR GEHEUGEN VIA VEB
3500: 2F14 20 72 3F  JSR  INCVEB

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240182 PAGE 08

```

3510: 2F17 4C 51 3E      JMP      $3E51  DISPLAY FILE ID
3520: 2F1A 20 97 3F  ENDRD JSR      RDBYT  LEES CHECKSUM
3530: 2F1D CD 6E 1A      CMP      CHKL   VERG. MET BER. WAARDE
3540: 2F20 D0 9B      BNE     SYNC   TERUG INGEVAL FOUT
3550: 2F22 20 97 3F      JSR     RDBYT  IDEM VOOR CHECKSUM HIGH
3560: 2F25 CD 6F 1A      CMP     CHKH
3570: 2F28 D0 93      BNE     SYNC
3580: 2F2A 4C 7A 30      JMP     $307A  SCHAKEL PB6
3590: 2F2D 20 A1 30      JSR     $30A1  RENUMBER SOURCE
3600: 2F30 EA      NOP
3610: 2F31 EA      NOP
3620: 2F32 4C A4 3F      JMP     INIT   HERSTEL POORT
3630:      :
3640: 3FA4      ORG     $3FA4
3650:      :
3660: 3FA4 A9 FF      INIT   LDAIM $FF
3670: 3FA6 8D 80 1A     STA   PAD
3680: 3FA9 60      RTS
3690:      :
3700:      : ---- FILE 0E ----
3710:      :
3720:      : DOEL: VOORKOMEN DAT BIJ INLEZEN PB5 0 WORDT, EN DUS
3730:      : DE OPNAME RECORDER GAAT DRAAIEN.
3740:      :
3750:      :
3760:      :
3770:      :
3780: 3FAA 20 9E 3E     JSR   STAB
3790:      :
3800: 3E9E      ORG   $3E9E
3810:      :
3820: 3E9E 48     STAB  PHA      BEWAAR ACCU
3830: 3E9F 8A     TXA
3840: 3EA0 09 20     DRAIM $20     HAAL KARAKTER
3850: 3EA2 8D 82 1A  STA  PBD     ZORG DAT PB5 = 1
3860: 3EA5 68     PLA      EN STUUR DISPLAY
3870: 3EA6 60     RTS     HAAL KARAKTER TERUG
3880:      :
3890:      : ---- FILE 0F ----
3900:      :
3910:      : DOEL: INVULLING NIEUWE ADRESSEN BINNEN M. A.
3920:      :
3930:      : VEB - $1A00 OP:
3940:      : $2F11, $2F3F, $2F69, $390A, $3C0C, $3D54.
3950:      :
3960:      : VEB +01 - $1A01 OP:
3970:      : $2F72, $30C1, $30D6, $30DB, $32DF, $3C19, $3D13,
3980:      : $3D4A.
3990:      :
4000:      : VEB +02 - $1A02 OP:

```

6502 SYSTEM SOFTWARE

THE JUNIOR PROMOTING CY (W&J) 240182 PAGE 09

```
4010:      : $2F78, $30C7, $30E3, $30DE, $32E4, $3C1E,
4020:      : $3D18, $3D4F.
4030:      :
4040:      : INTVEB - $3F58 OP:
4050:      : $2F42, $390D, $3C0F.
4060:      :
4070:      : INTVEB +12 - $3F64 OP:
4080:      : $2EC2
4090:      :
4100:      : INCVEB - $3F72 OP:
4110:      : $2F14, $2F6F.
4120:      :
4130:      : C. CHECKSUM - $3F7B OP:
4140:      : $2EEE, $2EF7, $2F0E, $2FAF.
4150:      :
4160:      : READCH - $3F8D OP:
4170:      : $2EDA, $2EFF, $3C31.
4180:      :
4190:      : READBYTE - $3F97 OP:
4200:      : $2EEB, $2EF4, $2F1A, $2F22, $32F0.
4210:      :
4220:      : ONEBYT - $0BC2 OP:
4230:      : $2ECA, $3C21.
```

BUGS

IN DE 6502 KENNER NR. 19 IS EEN PROGRAMMA GEPUBLICEERD VAN DE AUTEUR A. S. HANKEL, T.W. EEN 4 KOLOMMEN PRINTER. IN DIT PROGRAMMA, ZO MELDT DE AUTEUR ONS, IS EEN FOUT GESLOPEN. DEZE ZIET ER ALS VOLGT UIT:

```
0222 : 85 03 STAZ HTEL
0222 : 85 02 STAZ VTEL
```

.....
LET OP: NIEM REDAKTIE-ADRES !!
zie de colofon
.....

6502 SYSTEEM SOFTWARE

JUNIORS AAN DE PET

Op verzoek van velen volgen hier dan de listings van de software die op de bijeenkomst van 19 september j.l. werd gebruikt. Veel commentaar wil ik hier niet geven: Dat staat wel voldoende in de listings. Een opmerking vooraf is hier echter wel op zijn plaats en wel een waarschuwing: Veel JUNIOR en PET gebruikers zijn nog onervaren in het programmeren in assembler. De software die hier voor U list is niet alleen systeem gebonden (Junior of PET) maar bovendien geschreven voor een speciale gelegenheid en voor bijzondere omstandigheden want mijn 6522 VIA was toen defect. Wat U hier ziet is dus "wespooi software" die na 19 sept. eigenlijk geen waarde meer heeft. Toch wel gewoon leuk om te publiceren en misschien eisen ideetjes uit te halen, overigens: We zullen binnenkort nog aandacht gaan besteden aan het "echt" programmeren van FIA en VIA (6532 en 6522)

Ruud Uehoff.

```
0001      .LS
0002 ;*****
0003 ;* PET NAAR JUNIOR *
0004 ;*****
0005 ;
0006 ;PET I/O ( USER PORT )
0007 ;
0008 DDRVIA      .DE $E843      ;DATA RICHTINGS REG.
0009 DATVIA      .DE $E84F      ;DATA REGISTER.
0010 ;
0011 MONIT       .DE $FD56      ;PET MONITOR.
0012 ;
0013 ;
0014 ;PET OPERATING SYSTEM.
0015 ;
0016 ; 1. ZEROPAGE LOKATIES.
0017 ;
0018 ADRLUR90 AN .DE $FB        ;ADRESTELLER.LAAG BYTE.
0019 ADRHMMMOJO .DE $FC        ;IDEM.HOOG BYTE.
0020 ENDL RUSTU .DE $FD        ;EINDADRES.LAAG BYTE.
0021 ENDH TITU  .DE $FE        ;IDEM.HOOG BYTE.
0022 ;
0023 ; 2. SYSTEEMROUTINES.
0024 ;
0025 SKIP        .DE $E7EB      ;LEES KARAKTER.(OCR)
0026 PRNTXA     .DE $E784      ;PRINT KARAKTERS IN X EN A.
0027 INADR      .DE $E7A7      ;INPUT,ADRES IN ADR.
0028 CHANGE    .DE $E797      ;VERWISSEL ADR EN END.
0029 ;
```

6502 SYSTEM SOFTWARE

```

0030 ;***** PROGRAMMA *****
0031 ;
0032 .BA #0400 ;STARTADRES.
0033 .OS
0034 ;
0400- A9 FF 0035 START LDA #FF ;RICHTINGSREG.OUTPUT
0402- 8D 43 E8 0036 STA DDRVIA
0405- 8D 4F E8 0037 STA DATVIA ;LIJN 'HOOG' IN RUST.
0408- A2 93 0038 LDX #93 ;SCHERM SCHOONMAKEN.
040A- A9 2D 0039 LDA #'- ;PROMPTING IN A.
040C- 20 84 E7 0040 JSR PRNTXA ;PRINT BEIDE KARAKTERS.
040F- 20 A7 E7 0041 JSR INADR ;INPUT STARTADRES JUNIOR.
0412- 20 97 E7 0042 JSR CHANGE ;OVERWISSEL ADR MET END
0415- 20 EB E7 0043 JSR SKIP ;SLA SPATIE OVER.
0418- 20 A7 E7 0044 JSR INADR ;LEES EINDADRES.
041B- 20 97 E7 0045 JSR CHANGE ;OVERWISSEL ADR MET END.
0046 ;
041E- 78 0047 SEI
041F- A0 00 0048 BYTE LDY #00 ;ZET INDEX OP NUL,
0421- B1 FB 0049 LDA (ADRL),Y ;OM VOLGEND BYTE TE LEZEN.
0423- 48 0050 PHA ;BYTE OP DE STACK.
0424- A2 08 0051 LDX #08 ;STELLER VOOR 8 BITS PER BYTE.
0426- 68 0052 BIT PLA ;HAAL BYTE VAN STACK.
0427- 4A 0053 LSR A ;SCHUIF VOLGEND BIT IN C.
0428- 48 0054 PHA ;RESTANT WEER OP DE STACK.
0429- 90 03 0055 BCC NULL ;INDIEN C=1,
042B- A9 FF 0056 LDA #FF ;CODE VOOR '1 IN A,
042D- 2C 0057 .BY #2C ;EN EEN INSTRUCTIE OVERSLAAN.
042E- A9 00 0058 NULL LDA #00 ;ANDERS CODE VOOR '0 IN A.
0430- 48 0059 PHA ;BITCODE OP DE STACK
0431- A9 00 0060 LDA #00 ;ZET CODE VOOR LAAG,
0433- 8D 4F E8 0061 STA DATVIA ;OP DE UITGANG.(STARTBIT)
0436- A0 10 0062 LDY #10 ;GEEF DE JUNIOR DE TYD,
0438- 88 0063 DELA DEY ;DIT BIT TE DETECTEREN.
0439- D0 FD 0064 BNE DELA
043B- 68 0065 PLA ;HAAL BITCODE VAN DE STACK,
043C- 8D 4F E8 0066 STA DATVIA ;EN ZET OP DE UITGANG.
043F- A0 40 0067 LDY #40 ;GEEF DE JUNIOR DE TYD,
0441- 88 0068 DELB DEY ;DIT BIT TE LEZEN.
0442- D0 FD 0069 BNE DELB
0444- A9 FF 0070 LDA #FF ;CODE VOOR 'HOOG'
0446- 8D 4F E8 0071 STA DATVIA ;OP DE UITGANG.(STOPBIT)
0449- A0 00 0072 LDY #00 ;GEEF DE JUNIOR DE TYD,
044B- 88 0073 DELC DEY ;DIT ALLES TE VERWERKEN.
044C- D0 FD 0074 BNE DELC
044E- CA 0075 DEX ;VERLAAG DE BITTELLER.
044F- D0 D5 0076 BNE BIT ;NIET KLAAR: VOLGEND BIT.
0451- 68 0077 PLA ;ANDERS BYT OP STACK WAARDELOOS.
0452- E6 FB 0078 INC *ADRL ;VERHOOG ADRES,LAAG.
0454- D0 02 0079 BNE SKP ;INDIEN LAAG BYTE NUL,
0456- E6 FC 0080 INC *ADRH ;OOK HOOG ADRESBYTE.
0458- 38 0081 SKP SEC ;AFTREKKEN:
0459- A5 FD 0082 LDA *ENDL ;EINDADRES,LAAG,
045B- E5 FB 0083 SBC *ADRL ;MINUS ADRES,LAAG.
045D- A5 FE 0084 LDA *ENDH ;EN EINDADRES,HOOG,

```

6502 SYSTEM SOFTWARE

```

045F- E5 FC      0085      SBC *ADRH      ;MINUS ADRES,HOOG.
0461- B0 BC      0086      BCS BYTE      ;INDIEN C=1,VOLGEND BYTE.
0463- 58          0087      CLI           ;
0464- 4C 56 FD    0088      JMP MONIT     ;ANDERS KLAAR.

```

0000,0467,0467

```

0091      .LS
0092      ;*****
0093      ;* ROUTINE JUNIOR VAN PET *
0094      ;*****
0095      ;
0096      ;JUNIOR ZEROPAGE
0097      ;
0098      ADRSL      .DE $DE
0099      ADRSH      .DE $DF
0100      ;
0101      ;JUNIOR I/O
0102      ;
0103      DATA      .DE $1A80      ;PIA DATAREGISTER.
0104      ;
0105      .BA $0180      ;STARTADRES.
0106      .MC $0500
0107      .OS
0180- A2 08      0108      BYTES      LDX #$08      ;TELLER VOOR 8 BITS.
0182- AD 80 1A   0109      BITS      LDA DATA     ;LEES DATAREGISTER.
0185- C9 FF      0110      CMP #$FF     ;WACHT OP STARTBIT.
0187- F0 F9      0111      BEQ BITS
0189- A0 20      0112      LDY #$20     ;WACHT EVEN,TOT DE PET,
018B- 88          0113      WAIT      DEY          ;DE GELEGENHEID HAD,
018C- D0 FD      0114      BNE WAIT    ;EEN BIT UIT TE ZENDEN.
018E- AD 80 1A   0115      LDA DATA     ;LEES DAN DE BITCODE.
0191- C9 FF      0116      CMP #$FF     ;INDIEN 'LAAG',
0193- F0 02      0117      BEQ HIGH    ;
0195- 18          0118      CLC          ;ZET C=0
0196- 24          0119      .BY $24     ;EN SLA EEN INSTRUCTIE OVER.
0197- 38          0120      HIGH      SEC          ;ZET ANDERS C=1.
0198- B1 DE      0121      LDA (ADRSL),Y ;LEES BYTE.
019A- 6A          0122      ROR A        ;SCHUIF HET BIT ER IN,
019B- 91 DE      0123      STA (ADRSL),Y ;EN ZET HET BYTE TERUG.
019D- AD 80 1A   0124      WAITSTOP  LDA DATA     ;WACHT NU TOT DE
01A0- C9 FF      0125      CMP #$FF     ;LIJN WEER HOOG IS.
01A2- D0 F9      0126      BNE WAITSTOP ;(STOPBIT)
01A4- CA          0127      DEX         ;VERLAAG DE BITTELLER.
01A5- D0 DB      0128      BNE BITS     ;NIET KLAAR: VOLGEND BIT.
01A7- E6 DE      0129      INC *ADRSL   ;VERHOOG ADRES,LAAG.
01A9- D0 02      0130      BNE SKPP     ;INDIEN NU NUL,
01AB- E6 DF      0131      INC *ADRSH   ;VERHOOG OOK HOOG BYTE.
01AD- D0 D1      0132      SKPP      BNE BYTES     ;GA VOLGEND BYTE DOEN.

```

0000,01AF,052F

6502 SYSTEM SOFTWARE

```

0135          .LS
0136 ;*****
0137 ;*** JUNIOR TERMINAL ONTVANG ROUTINE ***
0138 ;*****
0139 ;
0140          .BR $1A00          ;STARTADRES IN JUNIOR
0141          .MC $0600          ;OPSLAG-STARTADRES IN PET
0142          .OS                ;SLA OP IN PET
0143 ;
0144 DATA    .DE $1A80          ;PIA DATA REGISTER.
0145 ADRL     .DE $FA           ;LAAG ADRESBYTE.
0146 ADRH     .DE $FB           ;HOOG ADRESBYTE.
0147 CONT     .DE $F9           ;INHOUD VAN ADR.
0148 VECT     .DE $01           ;EXECUTIE VECTOR
0149 FLAG     .DE $00           ;EXECUTIE VLAG
0150 SCAN     .DE $1D88          ;DISPLAY ROUTINE
0151 ;
0152 ;HOOFDRoutine
0153 ;
1A00- A9 00 0154 START    LDA #$00          ;ZET "GEEN EXECUTIE"
1A02- 85 00 0155          STA *FLAG
1A04- 20 88 1D 0156 REPEAT JSR SCAN          ;TOON ADRES OP DISPLAY.
1A07- AD 80 1A 0157          LDA DATA          ;LEES INPUT.
1A0A- C9 FF 0158          CMP #$FF          ;INDIEN LIJN HOOG,
1A0C- F0 F6 0159          BEQ REPEAT          ;WACHT OP LAAG.
1A0E- AD 80 1A 0160 WAIT   LDA DATA          ;ANDERS WACHTEN,
1A11- C9 FF 0161          CMP #$FF          ;TOT DE LIJN,
1A13- D0 F9 0162          BNE WAIT          ;WEER HOOG WORDT
1A15- 20 2D 1A 0163          JSR BYTE          ;LEES EEN BYTE UIT DE PET.
1A18- 85 FA 0164          STA *ADRL          ;DAT IS LAAG ADRES.
1A1A- 20 2D 1A 0165          JSR BYTE          ;LEES EEN BYTE UIT DE PET
1A1D- 85 FB 0166          STA *ADRH          ;DAT IS HOOG ADRES.
1A1F- 20 2D 1A 0167          JSR BYTE          ;LEES EEN BYTE UIT DE PET.
1A22- A0 00 0168          LDY #$00          ;INDEX NUL.
1A24- 91 FA 0169          STA (ADRL),Y      ;ZET INHOUD OP ADRES.
1A26- A5 00 0170          LDA *FLAG          ;INDIEN EXECUTIE GEWENST
1A28- F0 DA 0171          BEQ REPEAT
1A2A- 6C 01 00 0172 EXEC   JMP (VECT)          ;VOER PROGRAMMA UIT
0173 ;
0174 ;LEES EEN BYTE UIT DE PET
0175 ;
1A2D- A2 08 0176 BYTE    LDX #$08          ;TELLER VOOR 8 BITS
1A2F- AD 80 1A 0177 BIT     LDA DATA          ;WACHT OP STARTBIT.
1A32- C9 FF 0178          CMP #$FF
1A34- F0 F9 0179          BEQ BIT
1A36- A0 20 0180          LDY #$20          ;WACHT OP GELDIG BIT.
1A38- 88 0181 DEL     DEY
1A39- D0 FD 0182          BNE DEL
1A3B- AD 80 1A 0183          LDA DATA          ;LEES DE BITCODE.
1A3E- C9 FF 0184          CMP #$FF          ;INDIEN "NUL",
1A40- F0 02 0185          BEQ HIGH
1A42- 18 0186          CLC                ;ZET "C=0".
1A43- 24 0187          .BY $24          ;EN SLA "C=1" OVER.
1A44- 38 0188 HIGH     SEC                ;ANDERS "C=1".
1A45- 66 F9 0189          ROR *CONT          ;SCHUIF BIT IN BYTE
1A47- AD 80 1A 0190 STOP   LDA DATA          ;WACHT OP "STOP"

```

6502 SYSTEM SOFTWARE

```

1A4A- C9 FF      0191      CMP #FF
1A4C- D0 F9      0192      BNE STOP
1A4E- CA         0193      DEX ;VERLAAG BIT TELLER.
1A4F- D0 DE      0194      BNE BIT ;NIET KLAAR: VOLGEND BIT.
1A51- A5 F9      0195      LDA *CONT ;ANDERS RESULTAAT IN A
1A53- 60         0196      RTS ; EN "EINDE SUBROUTINE"

```

0000, 1A54, 0654

```

0199      .LS
0200 ;*****
0201 ;** PET TERMINAL ZEND ROUTINE **
0202 ;*****
0203 ;
0204      .BA $0700
0205      .OS
0206 ;
0207 ;PET OPERATING SYSTEM
0208 ;
0209 ADRL      .DE $FB ;ADRESPOINTER, LAAG
0210 ADRH      .DE $FC ;ADRESPOINTER, HOOG
0211 ;
0212 DDRVIA    .DE $E843 ;RICHTINGSREGISTER.
0213 DATVIA    .DE $E84F ;DATAREGISTER.
0214 ;
0215 MONIT     .DE $FD56 ;RESTART MONITOR.
0216 INCH     .DE $FFCF ;LEES EEN KARAKTER.
0217 INBYT    .DE $E7B6 ;LEES EEN BYTE.
0218 INADR     .DE $E7A7 ;LEES EEN ADRES.
0219 PRAD      .DE $E76A ;PRINT HET ADRES.
0220 CRLF      .DE $FDD0 ;PRINT CRLF.
0221 PRINT     .DE $FFD2 ;PRINT EEN KARAKTER.
0222 ;
0700- A9 FF      0223 START    LDA #FF ;VIA VOOR OUTPUT.
0702- 8D 43 E8   0224      STA DDRVIA
0705- 8D 4F E8   0225      STA DATVIA ;LIJN IS "HOOG" IN RUST.
0708- 20 D0 FD   0226 NEW      JSR CRLF ;BEGIN OP NIEUWE REGEL.
070B- A9 2D     0227      LDA #' ;PRINT PROMPTING.
070D- 20 D2 FF   0228      JSR PRINT
0710- 20 A7 E7   0229      JSR INADR ;INPUT, ADRES.
0713- 20 D0 FD   0230 INSTR    JSR CRLF ;NIEUWE REGEL.
0716- 20 6A E7   0231      JSR PRAD ;PRINT ADRES.
0719- A9 00     0232 DATA    LDA #00 ;PRINT "NULL"
071B- 20 D2 FF   0233      JSR PRINT ;OM OP RETURN TE WACHTEN.
071E- 20 CF FF   0234      JSR INCH ;INPUT, EEN KARAKTER.
0721- C9 0D     0235      CMP #0D ;INDIEN "RETURN"
0723- F0 EE     0236      BEQ INSTR ;VOLGENDE INSTRUCTIE.
0725- C9 2A     0237      CMP #'* ;INDIEN STER,
0727- F0 DF     0238      BEQ NEW ;BEGIN OPNIEUW
0729- C9 20     0239      CMP #20 ;INDIEN GEEN SPATIE,
072B- F0 0D     0240      BEQ SKIP
072D- A9 FF     0241      LDA #FF ;MAAK LIJN HOOG.
072F- 8D 4F E8   0242      STA DATVIA
0732- A9 00     0243      LDA #00 ;ZET VIA VOOR INPUT.

```

6502

SYSTEM SOFTWARE

0734-	8D	43	E8	0244	STA DDRVIA	
0737-	4C	56	FD	0245	JMP MONIT	;EEN GA NAAR DE MONITOR.
073A-	20	B6	E7	0246	JSR INBYT	;ANDERS,LEES DATA.
073D-	20	49	07	0247	JSR OUTPUT	;STUUR NAAR DE PET.
0740-	E6	FB		0248	INC *ADRL	;HOOG ADRES OP.
0742-	D0	02		0249	BNE SKP	
0744-	E6	FC		0250	INC *ADRH	
0746-	4C	19	07	0251	JMP DATA	;VOLGEND BYTE.
				0252	;	
				0253	;	
				0254	;	
				0255	OUTPUT	
0749-	48			0255	PHA	;SAVE DATABYTE
074A-	78			0256	SEI	;BLOkkeER INTERRUPTS.
074B-	A9	00		0257	LDA #00	;ZENDT "ATTENTIE"
074D-	8D	4F	E8	0258	STA DATVIA	
0750-	A2	10		0259	LDX #10	;WACHT TOT DE JUNIOR,
0752-	A0	00		0260	LDY #00	;GEREED IS VOOR ONTVANGST.
0754-	88			0261	WAIT1	
0755-	D0	FD		0262	BNE WAIT1	
0757-	CA			0263	DEX	
0758-	D0	FA		0264	BNE WAIT1	
075A-	A9	FF		0265	LDA #FF	;MAAK DAN DE LIJN WEER HOOG.
075C-	8D	4F	E8	0266	STA DATVIA	
075F-	A0	00		0267	LDY #00	;PAUZE TUSSEN TWEE BYTES.
0761-	88			0268	HOLD	
0762-	D0	FD		0269	BNE HOLD	
0764-	A5	FB		0270	LDA *ADRL	;LEES LAAG ADRESBYTE.
0766-	20	74	07	0271	JSR OUTBYT	;VERZENDT HET.
0769-	A5	FC		0272	LDA *ADRH	;LEES HOOG ADRESBYTE.
076B-	20	74	07	0273	JSR OUTBYT	;VERZENDT HET.
076E-	68			0274	PLA	;RESTORE DATABYTE.
076F-	20	74	07	0275	JSR OUTBYT	;VERZENDT HET.
0772-	58			0276	CLI	;GEEF INTERRUPTS VRIJ
0773-	60			0277	RTS	;
				0278	;	
				0279	;	
				0280	;	
				0281	OUTBYT	
0774-	48			0281	PHA	;SAVE BYTE
0775-	A2	08		0282	LDX #08	;TELLER VOOR 8 BITS.
0777-	68			0283	BIT	
0778-	4A			0284	PLA	;RESTORE BYTE OF RESTANT
0779-	48			0285	LSR A	;SCHUIF EEN BIT UIT.
077A-	90	03		0286	PHA	;SAVE RESTANT VAN BYTE.
077C-	A9	FF		0287	BCC NULL	;INDIEN BIT=1
077E-	2C			0288	LDA #FF	;CODE VOOR "EEN" IN A.
077F-	A9	00		0289	.BY #2C	;SLA EEN INSTRUCTIE OVER.
0781-	48			0290	LDA #00	;ANDER CODE VOOR 0.
0782-	A9	00		0291	PHA	;SAVE BITCODE.
0784-	8D	4F	E8	0292	LDA #00	;ZENDT STARTBIT.
0787-	A0	10		0293	STA DATVIA	
0789-	88			0294	LDY #10	;GEEF DE JUNIOR DE
078A-	D0	FD		0295	DEY	;GELEGENHEID.
078C-	68			0296	BNE WAIT2	;DIT BIT TE DETECTEREN.
078D-	8D	4F	E8	0297	PLA	;RESTORE BITCODE
0790-	A0	40		0298	STA DATVIA	;ZENDT BITCODE.
0792-	88			0299	LDY #40	;GEEF DE JUNIOR DE
0793-	D0	FD		0300	DEY	;GELEGENHEID
					BNE WAIT3	;DIT BIT TE LEZEN.

6502 SYSTEM SOFTWARE

```

0795- A9 FF 0301 LDA #FF ;ZENDT "STOP".
0797- 8D 4F E8 0302 STA DATVIA
079A- A0 00 0303 LDY #00 ;GEEF NU DE JUNIOR DE TYD,
079C- 88 0304 WAIT4 DEY ;HET BIT TE VERWERKEN.
079D- D0 FD 0305 BNE WAIT4
079F- CA 0306 DEX ;VERLAAG BITTELLER
07A0- D0 D5 0307 BNE BIT ;NIET KLAAR: VOLGEND BIT.
07A2- 68 0308 PLA ;WUILNIS WEG VAN DE STAPEL.
07A3- 60 0309 RTS

```

0000, 07A4, 07A4

```

0312 .LS
0313 ;*****
0314 ;*** JUNIOR DISPLAY ONTVANGER ***
0315 ;*****
0316 ;
0317 .BA $0300 ;STARTADRES
0318 .MC $0800 ;OPSLAG ADRES IN DE PET
0319 .OS ;SLA OBJECT CODE OP.
0320 ;
0321 ;JUNIOR I/O LOCATIES.
0322 ;
0323 DDRA .DE $1A81 ;RICHTINGSREGISTERS.
0324 DDRB .DE $1A83
0325 DATA .DE $1A80 ;DATAREGISTERS
0326 DATE .DE $1A82
0327 ;
0328 ;DISPLAY BUFFER (6 KARAKTERS)
0329 ;
0330 BUFFR .DE $00
0331 ;
0332 ;STARTVECTOR.
0333 ;
0300- 4C 67 03 0334 START JMP BEGIN
0335 ;
0336 ;TOON INHOUD BUFFER OP HET DISPLAY
0337 ;
0303- A9 FF 0338 SHOW1 LDA #FF ;BEIDE POORTEN OUTPUT.
0305- 8D 81 1A 0339 STA DDRA
0308- 8D 83 1A 0340 STA DDRB
030B- A9 08 0341 SHOW2 LDA #08 ;BEGIN OP LINKER DISPLAY.
030D- 8D 82 1A 0342 STA DATE
0310- A2 00 0343 LDX #00 ;INDEX IN BUFFER OP NUL.
0312- B5 00 0344 NEXT LDA *BUFFR,X ;LEES KARAKTER)
0314- 8D 80 1A 0345 STA DATA ;EN ZET IN DISPLAY
0317- A0 00 0346 LDY #00 ;EVEN VASTHOUDEN.
0319- 88 0347 HOLD DEY
031A- D0 FD 0348 BNE HOLD
031C- A9 FF 0349 LDA #FF ;MAAK DISPLAY DONKER
031E- 8D 80 1A 0350 STA DATA
0321- EE 82 1A 0351 INC DATE ;VOLGEND DISPLAY
0324- EE 82 1A 0352 INC DATE
0327- E8 0353 INX ;VOLGEND KARAKTER

```

6502 SYSTEM SOFTWARE

```

0328- E0 06      0354      CPX #06
032A- D0 E6      0355      BNE NEXT
032C- A9 00      0356      LDA #00      ;PIAA WEER INPUT
032E- 8D 81 1A   0357      STA DDRA
0331- 60         0358      RTS
                0359 ;
                0360 ;NIEUW KARAKTER IN BUFFER
                0361 ;
0332- A2 01      0362 NEW      LDX #01      ;INDEX NAAR KARAKTER.
0334- A0 00      0363      LDY #00      ;INDEX NAAR VORIG KARAKTER.
0336- B5 00      0364 MOVE      LDA *BUFFER,X ;ZET KARAKTER.
0338- 99 00 00   0365      STA BUFFER,Y ;EEN PLAATS NAAR LINKS.
033B- E8         0366      INX          ;X EN Y EEN PLAATS VERDER.
033C- C8         0367      INY
033D- C0 06      0368      CPY #06      ;VERSCHUIF ALLE KARAKTERS.
033F- D0 F5      0369      BNE MOVE
0341- 60         0370      RTS
                0371 ;
                0372 ;LEES EEN BYTE UIT DE PET
                0373 ;
0342- A2 08      0374 BYTE      LDX #08      ;TELLER VOOR 8 BITS
0344- AD 80 1A   0375 BIT      LDA DATA    ;WACHT OP START
0347- C9 FF      0376      CMP #FF
0349- F0 F9      0377      BEQ BIT
034B- A0 20      0378      LDY #20      ;WACHT OP GELDIG BIT
034D- 88         0379 DEL      DEY
034E- D0 FD      0380      BNE DEL
0350- AD 80 1A   0381      LDA DATA    ;LEES DE BITCODE
0353- C9 FF      0382      CMP #FF      ;INDIEN "NUL",
0355- F0 02      0383      BEQ HIGH
0357- 18         0384      CLC          ;ZET C=0
0358- 24         0385      .BY #24     ;EN SLA "C=1" OVER
0359- 38         0386 HIGH     SEC          ;ANDERS ZET C=1
035A- 66 06      0387      ROR *BUFFER+6 ;SCHUIF IN BYTE
035C- AD 80 1A   0388 STOP      LDA DATA    ;WACHT OP STOP
035F- C9 FF      0389      CMP #FF
0361- D0 F9      0390      BNE STOP
0363- CA         0391      DEX          ;VERLAAG BITTELLER
0364- D0 DE      0392      BNE BIT      ;NIET KLAAR: VOLGEND BIT
0366- 60         0393      RTS
                0394 ;
                0395 ;HOOFDPROGRAMMA
                0396 ;
0367- A2 00      0397 BEGIN      LDX #00      ;VUL BUFFER MET SPATIES.
0369- A9 FF      0398      LDA #FF      ;CODE VOOR SPATIE
036B- 95 00      0399 FILL      STA *BUFFER,X
036D- E8         0400      INX
036E- E0 06      0401      CPX #06      ;
0370- D0 F9      0402      BNE FILL
0372- 20 03 03   0403 LOOP      JSR SHOW1     ;TOON DISPLAY
0375- AD 80 1A   0404      LDA DATA    ;LEES INPUT
0378- C9 FF      0405      CMP #FF      ;INDIEN HOOG,
037A- F0 F6      0406      BEQ LOOP     ;HERHAAL.
037C- AD 80 1A   0407 ATN      LDA DATA    ;WACHT TOT EINDE ATN
037F- C9 FF      0408      CMP #FF
0381- D0 F9      0409      BNE ATN

```

6502

SYSTEM SOFTWARE

```
0383- 20 42 03 0410 JSR BYTE ;ONTVANG DAN BYTE VAN PET.
0386- 20 32 03 0411 JSR NEW ;PAS DISPLAY AAN
0389- 4C 72 03 0412 JMP LOOP ;EN WACHT OP VOLGEND BYTE.
```

0000, 038C, 088C

```
0415 .LS
0416 ;*****
0417 ;*** PET DISPLAY ZENDER ***
0418 ;*****
0419 ;
0420 .BA $0900 ;STARTADRES
0421 .OS
0422 ;
0423 ;PET OPERATING SYSTEM
0424 ;
0425 BLINK .DE $A7 ;CURSOR KNIPPER (0=AAN)
0426 DDRVIA .DE $E843 ;RICHTINGSREGISTER.
0427 DATVIA .DE $E84F ;DATAREGISTER.
0428 GETCH .DE $FFE4 ;LEES TOETSENBORD.
0429 PRINT .DE $FFD2 ;PRINT KARAKTER.
0430 MONIT .DE $FD56 ;PET MONITOR
0431 ;
0432 ;HOOFDPROGRAMMA
0433 ;
0900- A9 FF 0434 START LDA #$FF ;VIA VOOR OUTPUT
0902- 8D 43 E8 0435 STA DDRVIA
0905- 8D 4F E8 0436 STA DATVIA ;LIJN IS HOOG IN RUST
0908- A9 0D 0437 LDA #$0D ;GA NAAR NIEUWE REGEL
090A- 20 D2 FF 0438 JSR PRINT
090D- A9 00 0439 NEXTCH LDA #$00 ;CURSOR KNIPPERT
090F- 8D A7 00 0440 STA BLINK
0912- 20 E4 FF 0441 GET JSR GETCH ;WACHT OP TOETSDRUK.
0915- F0 FB 0442 BEQ GET
0917- 20 D2 FF 0443 JSR PRINT ;PRINT KARAKTER
091A- C9 8D 0444 CMP #$8D ;INDIEN SHIFT-RETURN
091C- D0 03 0445 BNE SKIP
091E- 4C 56 FD 0446 JMP MONIT ;TERUG NAAR DE MONITOR
0921- C9 0D 0447 SKIP CMP #$0D ;INDIEN RETURN,
0923- F0 06 0448 BEQ LEGAL ;ACCEPTEEER.
0925- 29 7F 0449 AND #$7F ;ACCEPTEEER GEEN HOOFDLETTERS.
0927- C9 20 0450 CMP #$20 ;INDIEN CONTROLE KARAKTER,
0929- 90 E2 0451 BCC NEXTCH ;WIJS AF.
092B- 20 31 09 0452 LEGAL JSR TRANS ;ANDERS NAAR JUNIOR,
092E- 4C 0D 09 0453 JMP NEXTCH ;EN WACHT OP DE VOLGENDE.
0454 ;
0455 ;STUUR KARAKTER NAAR JUNIOR
0456 ;
0931- 20 3B 09 0457 TRANS JSR SEARCH ;ZOEK IN TABEL1
0934- B0 02 0458 BCS ACCEPT ;INDIEN GEVONDEN,
0936- A9 FF 0459 LDA #$FF ;ILLEGAAL=SPATIE
0938- 4C 4E 09 0460 ACCEPT JMP PUTJUN ;STUUR NAAR JUNIOR
0461 ;
```

6502 SYSTEM SOFTWARE

```

0462 ;ZOEK IN DE TABEL
0463 ;
393B- A2 00 0464 SEARCH LDX ##00 ;INDEX IN TABEL
393D- DD 99 09 0465 TRYNXT CMP TABLE1,X ;VERGELIJK KARAKTER
3940- F0 07 0466 BEQ FOUND ;INDIEN NIET GEVONDEN,
3942- E8 0467 INX ;PROBEER VOLGENDE
3943- E0 26 0468 CPX #38 ;37 MOGELIJKHEIDEN
3945- D0 F6 0469 BNE TRYNXT
3947- 18 0470 CLC ;ANDERS TERUG MET C=0
3948- 60 0471 RTS
3949- BD BF 09 0472 FOUND LDA TABLE2,X ;LEES CODE
394C- 38 0473 SEC ;EN KEER TERUG MET C=1
394D- 60 0474 RTS
0475 ;
0476 ;CODE NAAR DE JUNIOR
0477 ;
394E- 48 0478 PUTJUN PHA ;SAVE CODE
394F- 78 0479 SEI ;BLOKKEER INTERRUPTS
3950- A9 00 0480 LDA ##00 ;ZENDT "ATTENTIE"
3952- 8D 4F E8 0481 STA DATVIA
3955- A2 10 0482 LDX ##10 ;TYDSDUUR "ATTENTIE"
3957- A0 00 0483 LDY ##00
3959- 88 0484 WAIT1 DEY
395A- D0 FD 0485 BNE WAIT1
395C- CA 0486 DEX
395D- D0 FA 0487 BNE WAIT1
395F- A9 FF 0488 LDA #$FF ;EINDE "ATTENTIE"
3961- 8D 4F E8 0489 STA DATVIA
3964- A0 00 0490 LDY ##00 ;PAUZE TUSSEN TWEE BYTES.
3966- 88 0491 HLD DEY
3967- D0 FD 0492 BNE HLD
3969- A2 08 0493 LDX ##08 ;TELLER VOOR 8 BITS
396B- 68 0494 BIT PLA ;RESTORE CODE
396C- 4A 0495 LSR A ;SCHUIF EEN BIT UIT.
396D- 48 0496 PHA ;RESTANT TERUG OP STACK
396E- 90 03 0497 BCC NULL ;INDIEN BIT=1
3970- A9 FF 0498 LDA #$FF ;LAADT CODE VOOR "1"
3972- 2C 0499 .BY $2C ;EN SLA NUL OVER.
3973- A9 00 0500 NULL LDA ##00 ;ANDERS,LAADT CODE VOOR "0"
3975- 48 0501 PHA ;SAVE BITCODE
3976- A9 00 0502 LDA ##00 ;ZENDT STARTBIT
3978- 8D 4F E8 0503 STA DATVIA
397B- A0 10 0504 LDY ##10 ;BITTYD-LOOP
397D- 88 0505 WAIT2 DEY
397E- D0 FD 0506 BNE WAIT2
0980- 68 0507 PLA ;BITCODE VAN DE STACK
0981- 8D 4F E8 0508 STA DATVIA ;UITZENDEN,
0984- A0 40 0509 LDY ##40 ;EN EVEN LATEN STAAN.
0986- 88 0510 WAIT3 DEY
0987- D0 FD 0511 BNE WAIT3
0989- A9 FF 0512 LDA #$FF ;NU STOPBIT ZENDEN,
098B- 8D 4F E8 0513 STA DATVIA
098E- A0 00 0514 LDY ##00 ;EN EVEN LATEN STAAN
0990- 88 0515 WAIT4 DEY
0991- D0 FD 0516 BNE WAIT4
0993- CA 0517 DEX ;VERLAAG BIT TELLER.
0994- D0 D5 0518 BNE BIT ;NIET KLAAR: VOLGEND BIT

```

6502 SYSTEM SOFTWARE

```

0996- 68      0519      PLA      ;ANDERS VUIL VAN DE STACK
0997- 58      0520      CLI      ;STA WEER INTERRUPTS TOE
0998- 60      0521      RTS      ;EN KLAAR.
          0522 ;
          0523 ;KARAKTER TABEL
          0524 ;
0999- 41 42 43 0525 TABLE1 .BY 'ABC'
099C- 44 45 46 0526      .BY 'DEF'
099F- 47 48 49 0527      .BY 'GHI'
09A2- 4A 4B 4C 0528      .BY 'JKL'
09A5- 4D 4E 4F 0529      .BY 'MNO'
09A8- 50 51 52 0530      .BY 'PQR'
09AB- 53 54 55 0531      .BY 'STU'
09AE- 56 57 58 0532      .BY 'VWX'
09B1- 59 5A 20 0533      .BY 'YZ'
09B4- 0D 30 31 0534      .BY '$0D $01'
09B7- 32 33 34 0535      .BY '$234'
09BA- 35 36 37 0536      .BY '$567'
09BD- 38 39      0537      .BY '$89'
          0538 ;
          0539 ;CODE TABEL
          0540 ;
09BF- 08 03 27 0541 TABLE2 .BY $08 $03 $27
09C2- 21 06 0E 0542      .BY $21 $06 $0E
09C5- 42 0B 6F 0543      .BY $42 $0B $6F
09C8- 61 0F 47 0544      .BY $61 $0F $47
09CB- 48 2B 23 0545      .BY $48 $2B $23
09CE- 0C 18 2F 0546      .BY $0C $18 $2F
09D1- 12 07 63 0547      .BY $12 $07 $63
09D4- 43 41 09 0548      .BY $43 $41 $09
09D7- 11 24 FF 0549      .BY $11 $24 $FF
09DA- FF 40 79 0550      .BY $FF $40 $79
09DD- 24 30 19 0551      .BY $24 $30 $19
09E0- 12 02 78 0552      .BY $12 $02 $78
09E3- 00 10      0553      .BY $00 $10
0000,09E5,09E5
          0556 .LS
          0557 ;*****
          0558 ;* MINI LICHTKRANT *
          0559 ;*****
          0560 ;
          0561 .BA $0200
          0562 .MC $0A00
          0563 .OS
          0564 ;
          0565 DDRA .DE $1A81 ;RICHTINGSREGISTERS
          0566 DDRB .DE $1A83
          0567 DATA .DE $1A80 ;DATAREGISTERS
          0568 DATB .DE $1A82
          0569 ;
0200- A9 FF 0570 START LDA #$FF ;BEIDE POORTEN UITGANG
0202- 8D 81 1A 0571 STA DDRA
0205- 8D 83 1A 0572 STA DDRB

```

6502 SYSTEM SOFTWARE

```

208- A2 00      0573 RESTART      LDX #00          ;X EN Y OP START
20A- A0 00      0574          LDY #00
20C- BD 49 02   0575 SHOW      LDA TAB0,X      ;ZET VOLGENDE DATA,
20F- 8D 80 1A   0576          STA DATA
212- B9 56 02   0577          LDA DISP,Y      ;OP HET VOLGENDE DISPLAY
215- 8D 82 1A   0578          STA DATB
218- A9 00      0579          LDA #00        ;LAAT EVEN STAAN
21A- 69 01      0580 DEL       ADC #01
21C- D0 FC      0581          BNE DEL
21E- E8         0582          INX
21F- C8         0583          INY            ;VERHOOG BEIDE TELLERS
220- C0 06      0584          CPY #06        ;ALLE DISPLAYS GEHAD ?
222- D0 E8      0585          BNE SHOW      ;ZONIET,VOLGENDE
224- EE 5C 02   0586          INC CNT        ;LAAT EEN TYDJE ZIEN
227- D0 DF      0587          BNE RESTART
229- A9 C0      0588          LDA #C0        ;DAN TYDTELLER RESETTEN
22B- 8D 5C 02   0589          STA CNT
22E- AD 49 02   0590 SHIFT     LDA TAB0        ;SAVE EERSTE BYTE
231- 48         0591          PHA
232- A0 01      0592          LDY #01        ;INDEX NAAR ORGINEEL
234- A2 00      0593          LDX #00        ;INDEX NAAR NIEUWE PLAATS
236- B9 49 02   0594 MOVE      LDA TAB0,Y      ;ZET ELK BYTE,
239- 9D 49 02   0595          STA TAB0,X     ;EEN PLAATS TERUG
23C- C8         0596          INY
23D- E8         0597          INX            ;VERSCHUIF ZO DE HELE TABEL
23E- E0 0C      0598          CPX #0C
240- D0 F4      0599          BNE MOVE
242- 68         0600          PLA            ;ZET OPGESLAGEN BYTE
243- 8D 55 02   0601          STA TABC      ;ACHTERAAN
246- 4C 08 02   0602          JMP RESTART
                0603 ;
                0604 ;DATA TABEL
                0605 ;
249- E1 E3 AB   0606 TAB0      .BY $E1 $E3 $AB ;JUN
24C- FB A3 AF   0607          .BY $FB $A3 $AF ;IOR
24F- FF FF      0608          .BY $FF $FF
251- 8C 86 87   0609          .BY $8C $86 $87 ;PET
254- FF         0610          .BY $FF
255- FF         0611 TABC      .BY $FF
                0612 ;
                0613 ;DISPLAY TABEL
                0614 ;
256- 08 0A 0C   0615 DISP      .BY $08 $0A $0C
259- 0E 10 12   0616          .BY $0E $10 $12
                0617 ;
                0618 ;TYDTELLER
                0619 ;
25C- C0         0620 CNT        .BY $C0
000,025D,0A5D

```

6502 SYSTEM SOFTWARE

```

0003- A9 FF      0629 START      LDA #FF
0005- 8D 81 1A   0630          STA $1A81
0008- 8D 83 1A   0631          STA $1A83
000B- A9 08     0632 LB1        LDA #08
000D- 8D 82 1A   0633 LB2        STA $1A82
0010- A9 FE     0634          LDA #FE
0012- 8D 80 1A   0635          STA $1A80
0015- A0 00     0636 LB3        LDY #00
0017- A2 00     0637          LDX #00
0019- CA        0638 LB4        DEX
001A- D0 FD     0639          BNE LB4
001C- 88        0640          DEY
001D- D0 FA     0641          BNE LB4
001F- 38        0642          SEC
0020- 2E 80 1A   0643          ROL $1A80
0023- AD 80 1A   0644          LDA $1A80
0026- C9 7F     0645          CMP #7F
0028- D0 EB     0646          BNE LB3
002A- 18        0647          CLC
002B- AD 82 1A   0648          LDA $1A82
002E- 69 02     0649          ADC #02
0030- C9 14     0650          CMP #14
0032- D0 D9     0651          BNE LB2
0034- 4C 03 00   0652          JMP START

```

0000,0037,0037

DE HEER J. A. DEKKER UIT ALKMAAR MELDDE ONS TELEFONISCH DAT ER Z.I. EEN VIERTAL BUGS IN HET IN EDITIE 20 GEPUBLICEERDE SCHAAKPROGRAMMA MOETEN ZITTEN. SAMEN MET THEO KORTEKAAS ZIJN ER DRIE OPGESPOORD, EN WEL:

REGEL 1300 m. z. 0E
 REGEL 3300 m. z. C0 03
 REGEL 6470 m. z. D0 DF

DE VORIGE KEER DEED IK EEN OPRROEP MIJ TE HELPEN AAN EEN MONITOR-LISTING VAN DE AIM 65. DIE HEB IK ONTVANGEN. HET HELPT ONS BIJ HET UITZOEKEN VAN DIVERSE SOFTWARE. WIE HELPT MIJ DEZE KEER AAN EEN MONITORLISTING VAN DE OSI?

Willem van Pelt
 Redaktiesekretaris.

- DO 24-06-82 6500 - GEBRUIKERSGROEP DELFT E-CAFE
- ZA 21-08-82 VERSCHIJNINGSDATUM 6502-KENNER
- ZA 11-09-82 APPLE GEBRUIKERS GROEP DAG
- ZA 18-09-82 KIM CLUB DAG
OSI GEBRUIKERSGROEP DAG
- DO 30-09-82 6500 - GEBRUIKERSGROEP DELFT E-CAFE
- ZA 16-10-82 VERSCHIJNINGSDATUM 6502-KENNER
- DO 28-10-82 6500 - GEBRUIKERSGROEP DELFT E-CAFE
- ZA 20-11-82 KIM CLUB DAG
OSI GEBRUIKERSGROEP DAG
- DO 25-11-82 6500 - GEBRUIKERSGROEP DELFT E-CAFE
- ZA 18-12-82 VERSCHIJNINGSDATUM 6502-KENNER

'S ZONDAGS VAN 18.30 TOT 19.00 UUR VIA HILV. I
HOBBYSCHOOP, VAAK MET COMPUTERPROGRAMMA'S, DIE
M.B.V. EEN CASSETTERECORDER ZIJN OP TE NEMEN.
DOOR HET GEBRUIK VAN DE 'BASICODE' IS ELK
UITGEZONDEN 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.

HET BESTUUR VAN DE KIM-GEbruikers CLUB NEDERLAND HEEFT
BESLOTEN VOOR DE TOEKOMST VASTE LOCATIES TE BEPALEN VOOR
HET HOUDEN VAN CLUB-BIJEENKOMSTEN. TENZIJ ZICH ONVOOR-
ZIENE OMSTANDIGHEDEN VOORDOEN, ZIET HET ER THANS ALS
VOLGT UIT:

JANUARI	KROMMENIE
MAART	AMERSFOORT
MEI	GELDROP
SEPTEMBER	HEEMSTEDE
NOVEMBER	AMERSFOORT

HET BESTUUR IS OVERIGENS WEL GEINTERESSEERD IN MOGELIJK-
HEDEN VOOR HET HOUDEN VAN CLUB-BIJEENKOMSTEN ELDERS
IN HET LAND. WIE OP DE HOOGTE IS VAN GOEDE EN GOED-
KOPE ACCOMODATIES WORDT VERZOCHT DE VOORZITTER DAAR-
VAN IN KENNIS TE STELLEN.

LET OP: NIEUW REDAKTIE-ADRES !!
zie de colofon

DE **KIM** GEBRUIKERS CLUB
NEDERLAND

EEN CLUB VAN

6502 KENNERS