

Achtste Jaargang, nr. 5  
Oktober 1984



**KIM GEBRUIKERS CLUB  
NEDERLAND**

een club van 6502  
gebruikers

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, CHE-1, ATARI, de Proton-computers, ITT 2020, PEARCOM, PC-100, OSI, BBC, VIC-20 en COMMODORE 64, BASIS 108, PALLAS, SBC 65/68, FORMOSA, ORIC-1, ACE 1000, KEMPAC System 4, NCS 6502, MINTA, enz. enz.

Deze kennisuitwisseling wordt o.a. gerealiseerd door zesmaal per jaar DE 6502 KENNER te publiceren, vijf maal per jaar clubbijeenkomsten te houden, een cassettebibliotheek in stand te houden en paperware-service te bieden. Daarnaast organiseren de leden zelf regionale bijeenkomsten.

**Verschijningsdata  
DE 6502 KENNER**

---

derde zaterdag  
van de maanden:  
februari  
april, juni  
augustus  
oktober  
december

Regio Den Haag/R' dam e.o.  
=====

Inl. Gerard van Roekel  
v.d. Palmstraat 11-C  
3135 LK Vlaardingen  
Tel.: 010 - 351101

**Bijeenkomsten  
van de club**

---

derde zaterdag  
van de maanden:  
januari  
maart  
mei  
september  
november

Regio Arnhem e.o.

---

1e en 3e Ma-avond v.d. maand  
in HET DORP  
(zaal wordt aangegeven in de  
hal van de personeelsflat)  
Inl.: Henk Deutekom  
Tel.: 08303 - 8987

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.

Samenstelling van het bestuur is thans als volgt:

Voorzitter:

Dagelijks bestuur:  
Rinus Vleesch-Dubois  
Fl. Nightingalestraat 212  
2037 NG Haarlem (023 - 330993)

Erevoorzitter:  
Sekretaris:

Sieb de Vries  
Ruud Uhoff  
Voorburgpad 10  
6843 EM Arnhem (085 - 816935)  
(19.00 - 20.00 )

Penningmeester:

Riet Uhoff - Burghoorn  
voor adres: zie bij sekretaris

Ledenadministratie:  
Redactie DE 6502 KENNER

Leden:

Ruud Uhoff (zie bij sekr.)  
Willem L. van Pelt  
Jacob Jordaanstraat 15

2923 CK Krimozen a.d. IJssel

(01807 - 19881 )

Hard-/Software:

Adri Hankel (05490 - 51151)

John van Sorang (01807 - 20589)

Erwin Visschedijk (05490 - 71416)

Jan Tompot (070 - 931417)

Mw. H. de Vries - Van der Winden

Anton Mueller

zie adres sekretaris

Fl.45. = per kalenderjaar, postreke-

nino 3757649 t.n.v. KIM Gebruikers

Club te Arnhem

Oo aanvraag bij de redactie

Promotie:

Ereleden:

Adres van de vereniging:

Lidmaatschappen:

Advertisertarieven:

# DE 6502 KENNER

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

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

Willem L. van Pelt.  
Jacob Jordensstraat 15.  
2923 CK KRIMPEN AAN DEN IJssel.  
Tel.: 01807 - 19881

Redactie DE 6502 KENNER:  
Vaste medewerkers:

Willem van Pelt  
Gerard van Roekel  
Frans Smeehuijzen  
Jaap van Toledo

Freelance medewerkers:

Frans Bakx  
Adri Hankel  
Fridus Jonkman  
Bert Klein  
Herman Kuyvenhoven  
Roger Langeveld  
Anton Mueller  
Koen van Nieuwenhove

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

DE 6502 KENNER verschijnt 6 x per jaar en heeft een oplage van 500 exemplaren.

Copyright (C) 1984 KIM Gebruikers Club Nederland.

De voorpagina is een aqua-reel van een KIM, geschilderd door:  
Rinus Vleesch Dubois

## INHOUDSOPGAVE DE 6502 KENNER NR. 34 OKTOBER 1984

1. Van de redactie	2.
2. UITNODIGING Ledenvergadering/Landelijke Bijeenkomst	22.
3. MOSER Assembler Wordprocessing with Carl Moser's Text Editor ... Hans Buurman	3.
4. C-64 Cassette-interface: simpel ... Fer Weber C.B.M. 64 Kernel Test Routine ... Gerard van Roekel	32.
5. BASIC Tokenized Microsoft Basic Keywords and Addresses SYM-1 ... Willem L. van Pelt Save en Load Routines Proton DOS Voortzetting serie uitbreidingen SYM-Basic op JUNIOR ... Frans Smeehuijzen Maanlander. Basicprogramma. ... Maarten van Hintum	36.
6. FORTH Kleinste FORTH. Reakties. Hex/Ascii-dump ... Gert Klein Korrektie ... Fridus Jonkman	42.
7. Micro-ADE Patch op Micro-ADE ... Alfons v.d. Meutter. België	46.
8. COMAL.*KGN Programma Staafdiagram Staafdiagram afdrukken met Microline 80 printer ... Frans Bakx	48.
9. SENIOR-Monitor Patches on Senior Monitor V4.6 Voortzetting van een serie aanpassingen ... Rob Banen Versneld laden Proton Basic programma's ... Gert van Orobok	53.
10. OS65D Tios & Tricks ... met dank aan Wout van Dinther Wallbreaker for JUNIOR with OHIO-DOS Modifikatie van eerder gepubliceerd voor. Willem Kuitens ... Henk Mocking BUG in DOS-uitbreiding Elektuur. sept. 1984	58.
11. HARDWARE JUNIOR 8K of JUNIOR 64K ... Hans Mooij	62.
12. DIVERSEN Checktafel ... Phons Bloemen	64.
13. MARKTINFO 65816 microprocessor Elektuurs SAMSON-65 DOS Computer met OS65D FATE 65	68.
14. VRAGEN EN AANBOD	72.
	21.24.

# DE 6502 KENNER

Een van de heerlijke dingen die het geweten van een redakteur bij voortdurende beroeren is de vraag of hij het zijn lezers/leden van de club wel steeds naar de zin maakt. De realiteit waarschuwt hem daarbij keer op keer voor al te groot optimisme. Immers, zoveel hoofden zoveel zinnen. Of, in het eigen jargon: er is altijd wel een lezer die zich tekort gedaan voelt. Als dat een redacteur te ernstig aan het geweten knaagt, dan schiet hij mijns inziens te kort.

Het redactiewerk kan niet goed funktioneren als we alsovaar bezig moeten zijn het iedereen naar de zin te maken, het funktioneert pas optimaal als we proberen het zoveel mogelijke lezers naar de zin te maken.

Desalniettemin is het goed dat lezers bij tijd en wijle de

redacteur aanspreken op hun geweten, bijvoorbeeld door te

vragen "waarom komt in zo'n oer-nederlandse blad de Engelse taal voor?". Het is goed om dan eens op een rij te zetten hoe dat in elkaar steekt. Waarom gebruiken we eigenlijk

niet uitsluitend het Nederlandse?

Nu heb ik nogal de neiging te abstractheren alvorens de kern van de zaak te preciseren. Soms is dat grappig, soms stomvervelend. Beide risico's neem ik als ik me afvraag wat er nu feitelijk gevraagd wordt. Wat bedoelt men met de Nederlandse taal? Gaat het om de taal waarin geen uit andere talen overgenomen ver-Nederlandse woorden voorkomen? Hoe ziet eigenlijk het Nederlandse van vandaag eruit? Lijst het Nederlandse van morgen, door oponne van allerlei woorden uit andere talen, niet veel meer voor de hand?

Nog afgezien van het feit dat goede beantwoording van de vraag bemoeilijkt kan worden door onduidelijk gedefinieerde probleemstellingen, het kan ook betekenen dat zo'n eenvoudige vraag voortvloeit uit het gemak dat de mens dient.

Er is immers minder inspanning nodig als je uitsluitend de Nederlandse taal bezigt.

De redacteur heeft te maken met een aantal harde gegevens, welke mijns inziens onomkeerbaar zijn. Ik zal ze hieronder opsommen, maar vraag begrip voor het feit dat slechts de belangrijkste worden genoemd. Deze harde gegevens liggen ingebed in de strijd om de toekomst van blad en club te vergemakkelijken.

- a. De communicatie in de computerwereld wordt gedominated door het Engels. Niet enkel vanwege de oorsprong van de activiteiten, maar vooral omdat het Engels door vrijwel iedereen wordt begrepen en geaccepteerd.
- b. In den beginne bestond de club uit oer-Nederlandse leden. Het lidmaatschap staat echter open voor iedereen, dus ook van buiten Nederland. We herbergen nu ook leden uit Duitsland, België, Frankrijk, Spanje, Mexico, Portugal, en in India bestaat zelfs belangstelling voor de club. In sommige gevallen speelde Elektuur hier een rol.
- c. Kennisuitwisseling - doel van onze club - is een mondiale aangelegenheid. Wat onze lezers kunnen leren van de anderen wordt niet bepaald door landsgrenzen.
- d. Ons blad werd vroeger voornamelijk gevuld door bestuurs leden en redactiemedewerkers. Een enkel lid waaide het copy in te sturen. Dat is niet meer. De leden/lezers hebben begrepen dat zij het blad moeten vullen en dat de redactie sturen en stimulerend begeleiding geeft. Onder deze inzenders bevinden er zich die geen woord Nederlandse kennen, of die geheel of gedeeltelijk in het Engels oopenen.

Aangezien nu de inzenders zelf de inhoud van het blad bepalen ligt hier de beantwoording van de vraag vanwaar het Engels in ons blad. Der-Engels of steenkolen-Engels.

W.L. van Pelt

Een tijdje geleden is bij Adri Hankel en mij een wild idee geboren van een soort extra lange clubbijeenkomst. Deze zou dan een heel week-end kunnen duren. Overdag de computers aan en 's avonds gezellig bij elkaar zitten rond de barbecue. In het vorige nummer is hierover ook al iets vermeld. Bij zendamateurs blijkt zoets levensvatbaar. Zodra Adri en ik, of de redactie, reacties ontvangen gaan we bezien of welke wijze we tot realisering kunnen komen.

Er wordt door mij tussendoor ook nog gewerkt aan een monitor voor de nieuwe CPU-kaart uit Elektuur. Dit stuk bulksoftware begint al aardige vormen aan te nemen. Als het aan mij ligt kan de monitor algemeen clubdoed worden (red: al eerder publiceerde J.J. Janssen uit Nijmegen zijn software voor de VDU-kaart: de redactie soult naar nog meer leuke dingen voor beide kaarten).

Nog even een waarschuwing. In de ELEKTUUR staat een wijziging op de VDU-kaart voor 65C02 bezitters. Met deze wijziging wordt voorkomen dat er streepjes op het beeldscherm komen als de CPU in het beeldschermraam schrijft. Dit wordt gedaan met behulp van het lijntje RDY op de processor. Dit lijntje wordt met een stukje logica omlaag getrokken als de processor in het VDU-ram wil schrijven terwijl isolav enable op CRTC hoog is. Deze truc werkt aardig als je een eeprom programmer van Elektuur gebruikt. De eeprom programmer maakt ook gebruik van de RDY-lijn. En tot mijn grote verbazing zag ik dat zowel de lijn op de VDU-kaart als de lijn op de eeprom programmer gestuurd worden door TTL poorten met een totaalmoog uitoang. Dit is vreselijk om mogelijkheden. De oplossing ligt voor de hand. Zorg ervoor dat de TTL-poorten die de RDY-lijn moeten sturen een open collector uitoang hebben. Er is op beide kaarten een open collector-poort meer vrij, dus moet er oecens een 74LS05 of iets dergelijks bij oemonterd worden. Veel succes ermee.

Erwin Visschedijk.

12 K FATE 65 FOR JUNIOR.

=====

F ormat l ister      Formating Text files  
A ssembler      Assembling source code  
T ape-utilities      Utilities for tape and disk  
E ditor      Text editor

FATE 65 has the editor as a central program. With the editor one is able to develop text-files and source-files, to write on tape and to read from tape. With the format lister text-files are made on a printer. The lister recognises several commands, under which control-commands for the OKI Microline 80 printer. With the assembler source-files you can convert to object-code. From cassette to cassette. From memory to memory. From cassette to memory. From memory to cassette. Merging is possible. Splitting is possible. You need an extended JUNIOR-computer, two cassette recorders with motor-control, and at least 16 K RAM, better 32 K (2000-9000). Ask Mr. W.L. van Pelt, c/Jacob Jordaanstraat 15 Krimmen a.d. IJssel, 2923 CK, The Netherlands, for a list of prices.

# DE6502 KENNER

## WORD PROCESSING WITH CARL

### MOSER'S TEXT EDITOR

The 6502 macro assembler has a text editor that is well suited for its purpose, setting up and maintaining its sourcefile. However, to edit other sorts of text a different program is needed. The program described here enables the user to insert commands in the text which allow him to have the text outputted in a desired form, both on screen as on a printer. Here, the oki microline 80 is used, but the program can easily be modified to fit to other printers.

#### 1.General description of the program.

First, some of the assemblers vectors must be initialised by using RUN adress, where the value of adress depends on the programs assembly. During use, the text can be outputted with the USER command. If the break key is used, outputting can be resumed with CTRL Y. If this key is pressed at any other moment, the program will resume where it stopped last time, or after initialisation at the start of the textfile. However in the last case no output will be seen because both output devices have been switched off. After an error has been detected, CTRL Y will cause a restart at the start of the line in which the error occurred.

Commands must be enclosed between square brackets, two symbols with ascii values of hex 5B and 5D. These characters are printed by the microline as ↑ and ↓. After a ↑ is found, the next character is supposed to be the command. Commands must be followed by a colon or closing bracket, the former meaning another command follows, the latter ending the commands. The twenty-five recognised commands are explained below. It must be stressed that some of the commands are only effective at the end of a line. If commands set or reset a printing mode, the last time this command was used, holds. E.g. in a line the last H command encountered will determine the printer mode.

#### 2.Printing procedure.

At the end of a line, or on moving to another line, the current line is printed. This may take three passes:

- 1-On the first pass, the line is printed as usual.
- 2-On the second pass, if the 2 or B command has been used, desired parts of the line are printed again in the desired form. This second pass allows words to be printed extra dark or in a special way ( e.g. the dutch word Reel ).
- 3-On the third pass, if a U command has been used, an underline will be added. On default this will use the next line completely as is shown in this text's headings, but other settings can be obtained by using the C command.

The last two phases will only be used when needed, but they will all be used when a graphic code (127) has been used. This is because the microline has an advised duty cycle of 40% when printing graphics.

#### 3.Spaces.

During output spaces are subject to the following rules:

- 1-Between two groups of non-space characters only one space is allowed. More spaces can however be obtained with several commands, such as S, T and A.
- 2-At the end of each line in the original text, a space is inserted. However, rule 1 must be obeyed. Paragraphs are available with the P command. The first line of a paragraph is tabulated. The number of spaces here can be changed with the J command. It must be remarked that all above commands are useless when the lines are filled as in the next paragraph, after the Q command is given.

#### 4.Line sizes.

The length of a line is determined by the current printer mode and the positions of the borders. The printer mode is selected with de H and V commands for width and letter size, and letter height respectively. Columns are numbered from left to right with the leftmost being nr. 0, and the L and R commands set the borders to their values. The borders are reset after a H command. The T and . commands tabulate to a certain column. A terminal width can be given, if the width of a line is equal to this, the linefeed on screen will be suppressed to improve layout.

#### 5.Pages

After an F command, printing is done in pages. A page number is printed at the top of each page. The # command sets the page number to a value, which of course is a decimal. The I command allows the user to determine the size of the pages himself. It is assumed that V0 gives 6 lines/inch and V1 gives 8 lines/inch, so that the space between pages can be given in either 1/6 or 1/8 inch, the latter by simply adding 128. A value of 255 gives only a halt at the end of each page. The size of the page must be given in inches as an integer number. No assumptions made here. Only if an F command is given very close to the end of a page slight errors of a fraction of an inch will be made.

### 6. Miscellaneous.

The O command allows output to be sent to printer, screen neither or both, and the K command just waits for a key to be pressed.

Of course, sometimes an illegal input will be found. Most possibilities (as far as the author could see!) are quite harmless and will be ignored. If possible, unnecessary bits are thrown away and tabulations to non-existing columns aren't executed, but some things just cannot be allowed. In that case, an error message will be issued by means of the standard routine, showing !xx AT LINE yyyy/zz. In extreme cases yyyy may point to the line after the one containing the error. The zz field shows the value that caused the error. All error codes are shown in a table.

#### Detailed description of all commands.

=====

The following notations will be used:

n or m.....Either a decimal integer or the symbol ' followed by any character (this gives the ascii-value).

string.....A sequence of characters not containing : or , ended by : or .

1-Commands without operands.

D Move to the next line.

P n Or P , Start a new paragraph; move to next line and tabulate.

F If pages weren't used: start using pages, begin a new page (turned off by I).

I If pages were used: start a new page.

K Wait for any key depressed.

2 Start printing everything twice (extra dark).

1 Print everything once.

Defaults are: no pages, print once.

2-Commands with one operand.

Ln Set left hand border to n.

Rn Set right hand border to n.

Hn Set horizontal mode. Modes 0-2 give compressed, normal and double width on a long line, modes 3-5 the same on a short line.

Vn Set vertical mode. Mode 0 gives 6 lpi, mode 1 gives 8 lpi. Dat for the H and V commands are contained in a table at the end of the program.

Wn Set terminal width.

Tn Tabulate to column n. If the cursor is past that column, the command is ignored.

.n Same as the T command, only .'s are printed.

On Output control, n=0 means no output, n=1 uses the terminal, n=2 uses the printer and n=3 uses both.

Jn Set the paragraph tabulation to n.

Sn Print n spaces.

En Error control, n=0 means continue after error, n>0 causes a halt. See the assemblers .CE pseudo op.

In Set the page length to n inch. There is also a two operand version of this command, I0 or I turns the pages off.

+n Add n to all following ascii codes. This allows use of graphic codes

#n Set the page number to n.

An Send ascii code n.

Un Underline with symbol n. Useful are =, -, and -. Switched off with U.

Rn Fill out all lines (see fourth and fifth paragraph) unless the line has less than n words.

Bstring Print string, starting at current position, on the second pass.

Defaults are: H0;V1;I11,255;Q:#0;E1;O1;J5;+0;U.

3-Commands with two operands:

In,m Set page length to n inch, independent of printer mode. Set space between pages to m times

1/6 inch if m<128.

Set space between pages to (m-128) times 1/8 inch, if 127<m<255. If m=255, just stop

printing after each page.

Cn,m If m()>0, send a carriage return after printing pass n, if m=0 don't.

Error codes. Set to 00000000. Set the first terminal, then the last to 00000000. The first byte is the terminal number, the last byte is the printer number. The other bytes are reserved for future use.

=====

Following error codes are used:

BC Bad Command. Shown is the ascii value in hex.

30 Border setting out of range. Shown is the value of the L or R command involved.

31 Paragraph tabulation out of range. May occur after a L, R, or J command. Shown is the value of the last J command.

32 Tabulation out of range. A T or . command has been given to a position on the right of the right hand border. Shown is the operand.

33 Syntax error in a C command, or the first operand is zero. Shown is 00 in the last case, or an ascii value in the first.

All values are shown in hex.

**DE 6502 KENNER**

```

> PA

0001 :PRINTER FACILITIES PROGRAM
0002 .BA $9000
0003 BUflen .DE 132 :LARGEST NUMBER OF CHARACTERS/LINE POSSIBLE
0004 DUTCH .DE $C80 :CHARACTER TO SCREEN WITHOUT BREAKTEST
0005 RECHA .DE $12AE :READ CHARACTER FROM KEYBOARD
0006 PRINTER .DE $400C :CHARACTER TO PRINTER ONLY
0007 BREAK .DE $1A80 :BREAKTEST ADDRESS: BIT 7 IS 0 FOR BREAK
0008 WARMST .DE $2003 :ASSEMBLER WARM START
0009 ERRORM .DE $244B :ISSUE ERROR MESSAGE (ASSEMBLER)
0010 :ERROR FORM IS !XX AT LINE YYYY/ZZ WHERE XX IS X-REG.
0011 LINENR .DE $108 :VARIABLE FOR YYYY
0012 FILENR .DE $128 :SAME FOR ZZ
0013 CONTFL .DE $112 :FLAG () 0 AFTER .CE
0014 END .DE $D3 :END OF TEXTFILE ADDRESS
0015 BEGIN .DE $100 :BEGIN OF TEXTFILE
0016 PNT .DE $20 :POINTER TO TEXTFILE
0017 ;INIT - CHANGE ZERO PAGE VECTORS
0018 :CTRL Y GIVES JUMP TO $0
0019 :USER GIVES JUMP TO $3
0020 INIT LDX #5
0021 INITO LDA TABEL1.X
0022 STA *0,X
0023 DEX
0024 BPL INITO
0025 :SWITCH OFF OUTPUTS
0026 LDA #0
0027 STA SCREEN
0028 STA PRINT
0029 :INITIALISE VARIABLES
0030 JSR INIT2
0031 JMP WARMST
0032 :TABEL1 - ZERO PAGE DATA
0033 TABEL1 .BY $4C
0034 .SI CTRLY
0035 .BY $4C
0036 .SI USER
0037 :CTRLY - SEND CRLF AND CONTINUE
0038 CTRLY LDA ##$D
0039 JSR OUT
0040 LDA ##$A
0041 JSR OUT
0042 JMP LOOP
0043 :USER - ENABLE SCREEN, DISABLE PRINTER AND START
0044 USER LDA #0
0045 STA PRINT
0046 EOR #$1
0047 STA SCREEN
0048 JMP MAIN
0049 :OUTO - SEND CHARACTER TO SCREEN WITHOUT LF-SUPPRESSION
0050 OUTO TAX
0051 JMP OUT1
0052 :OUT - SEND CHARACTER TO SCREEN
0053 OUT TAX
0054 BPL SUPTEST
0055 :ASCII ) 127 : GRAPHIC CODE. REDUCE DUTY CYCLE
0056 STA USED2
0057 STA UNDERLD
0058 SUPTEST CMP #$A
0059 BNE OUT1
0060 :LINEFEED : IF WIDTH=TERMINAL WIDTH, DON'T SEND TO TERMINAL
0061 LDA HSIZE
0062 CMP TWIDTH
0063 BEQ OUT2
0064 :SEND TO TERMINAL ?
0065 OUT1 LDA SCREEN
0066 BEQ OUT2
0067 TXA
0068 JSR DUTCH
0069 :SEND TO PRINTER ?
0070 OUT2 LDA PRINT
0071 BEQ ENDOUT
0072 TXA
0073 JSR PRINTER
0074 ENDOUT CPX #$A

```

```

9065- D0 03      0075      BNE ENDOUT2
9067- 20 71 95    0076      :LINEFEED : CHECK IF END OF PAGE
906A- 60          0077      JSR NEWLINE
906A- 60          0078      RTS
906A- 60          0079      :NEXT - GET NEXT CHARACTER FROM 'MOSER' FILE
906B- A5 21      0080      :CHECK IF END OF TEXT
906D- C5 D4      0081      NEXT     LDA *PNT+1
906F- D0 04      0082      CMP *END+1
9071- A5 20      0083      BNE NEXT99
9071- A5 20      0084      LDA *PNT
9073- C5 D3      0085      CMP *END
9073- C5 D3      0086      :END OF TEXT : CARRY SET AND EXIT
9075- B0 2B      0087      NEXT99  BCS NEXTEND
9075- B0 2B      0088      :GET NEXT CHARACTER. SAVE COPY IN CCHAR
9077- A0 00      0089      LDY #0
9079- A2 01      0090      LDX #1
907B- B1 20      0091      LDA (PNT), Y
907D- 8D 2C 9A    0092      STA CCHAR
9080- 10 0E      0093      BPL NEXT1
9080- 10 0E      0094      :BIT 7 SET MEANS END OF LINE
9080- 10 0E      0095      :GET LINE NUMBER
9082- C8         0096      INY
9083- B1 20      0097      LDA (PNT), Y
9085- BD 08 01    0098      STA LINENR
9088- C8         0099      INY
9089- B1 20      0100      LDA (PNT), Y
908B- BD 09 01    0101      STA LINENR+1
908E- A2 03      0102      LDX #3
908E- A2 03      0103      :ADVANCE PNT TO NEXT CHARACTER IN TEXT
9090- 18         0104      NEXT1   CLC
9091- 8A         0105      TXA
9092- 65 20      0106      ADC *PNT
9094- 85 20      0107      STA *PNT
9096- A5 21      0108      LDA *PNT+1
9098- 69 00      0109      ADC #0
909A- 85 21      0110      STA *PNT+1
909A- 85 21      0111      :FETCH COPY AND MASK BIT 7
909C- AD 2C 9A    0112      LDA CCHAR
909F- 29 7F      0113      AND #%01111111
90A1- 60         0114      RTS
90A1- 60         0115      :END OF TEXT : RETURN TO 'MOSER'
90A2- 20 85 91    0116      NEXTEND JSR SENDBUFS
90A5- AD 00 01    0117      :INIT2 - INITIALISE ALL VARIABLES AND PRINTER
90A8- AC 01 01    0118      INIT2   LDA BEGIN
90A8- AC 01 01    0119      LDY BEGIN+1
90AB- 85 20      0120      :PNT TO START OF TEXT
90AD- 84 21      0121      STA *PNT
90AD- 84 21      0122      STY *PNT+1
90AF- A0 00      0123      :AND LINENR TO FIRST LINE
90B1- B1 20      0124      LDY #0
90B3- BD 08 01    0125      LDA (PNT), Y
90B6- 20 8F 92    0126      STA LINENR
90B9- B1 20      0127      JSR INC PNT
90BB- BD 09 01    0128      LDA (PNT), Y
90BE- 20 8F 92    0129      STA LINENR+1
90BE- 20 8F 92    0130      :FIRST CHARACTER IS AFTER LINE NUMBER
90C1- A9 05      0131      JSR INC PNT
90C3- BD 21 9A    0132      :PARAGRAPH TABULATION = 5
90C3- BD 21 9A    0133      LDA #5
90C3- BD 21 9A    0134      STA PTAB
90C6- A9 40      0135      :TERMINAL WIDTH = 64
90C8- BD 22 9A    0136      LDA #64
90C8- BD 22 9A    0137      STA TWIDTH
90CB- A9 00      0138      :DON'T USE SECOND BUFFER. UNDERLINE. PAGES
90CD- BD 10 9A    0139      LDA #0
90D0- BD 12 9A    0140      STA USE2
90D0- BD 12 9A    0141      STA UNDERL
90D3- BD 1B 9A    0142      STA PAGED
90D6- BD 2A 9A    0143      :DON'T FILL OUT. START AT TOP OF PAGE 0
90D9- BD 19 9A    0144      STA FILL
90DC- BD 1A 9A    0145      STA VCNT
90DF- BD 1C 9A    0146      STA VCNT+1
90E8- 20 11 91    0148      STA PAGE
90E8- 20 11 91    0149      :NORMAL CHARACTERS. NO LINEFEED AFTER PASS 1. COMPRESSED MODE
90E5- BD 27 9A    0150      STA OFFSET
90E8- 20 11 91    0150      STA CTRL
90E8- 20 11 91    0151      JSR HMODE

```

# DE6502 KENNER

```

0152      :DON'T CONTINUE AFTER ERRORS. 8 LINES/INCH
90EB- A9 01 0153      LDA #1
90ED- BD 12 01 0154      STA CONTFL
90F0- 20 3B 91 0155      JSR VMODE
90F3- A9 0A 0156      :LINEFEED AFTER PASS 2 AND 3
90F5- BD 28 9A 0157      LDA #$A
90F8- BD 29 9A 0158      STA CTRL+1
90FB- BD 29 9A 0159      STA CTRL+2
90FB- A9 02 0160      :PAGES OF 11 INCH
90FD- BD 25 9A 0161      LDA #$2
9100- A9 10 0162      STA PSIZE+1
9102- BD 24 9A 0163      LDA #$10
9105- A9 FF 0164      STA PSIZE
9107- BD 26 9A 0165      :STOP AFTER EACH PAGE
910A- 20 5D 91 0166      LDA #255
910A- 20 5D 91 0167      STA BSIZE
910D- 20 85 91 0168      :CLEAR BUFFERS
9110- 60 0169      JSR CLEARBUFS
9110- 60 0170      :SEND CR AND LF
9110- 60 0171      JSR SENDBUFS
9110- 60 0172      RTS
9111- 8D 2E 9A 0173      :HMODE - SET CHARACTER AND LINE WIDTH
9114- 8D 16 9A 0174      HMODE      STA TEMP
9114- 8D 16 9A 0175      STA HM
9117- 0A 0176      :MULTIPLY ACCU BY 5
9118- 0A 0177      ASL A
9119- 18 0178      ASL A
911A- 6D 2E 9A 0179      CLC
911D- AA 0180      ADC TEMP
911E- A0 04 0181      :SEND FOUR CODES FROM TABLE TO PRINTER
9120- BD 33 9A 0182      TAX
9120- BD 33 9A 0183      LDY #4
9120- BD 33 9A 0184      HM1      LDA HOR. X
9123- F0 03 0185      :DON'T SEND ZEROES
9125- 20 0C 40 0186      BEQ HM2
9128- E8 0187      JSR PRINTER
9129- 88 0188      HM2      INX
912A- D0 F4 0189      DEY
912A- D0 F4 0190      BNE HM1
912C- BC 33 9A 0191      :5TH IN TABLE IS LINE WITDH
912F- BC 14 9A 0192      LDY HOR. X
912F- BC 14 9A 0193      STY HSIZ
9132- 8C 20 9A 0194      :CLEAR BORDERS
9132- 8C 20 9A 0195      STY RBORD
9135- A9 00 0196      LDA #0
9137- BD 1F 9A 0197      STA LBORD
913A- 60 0198      RTS
913A- 60 0199      :VMODE - SET LINES/INCH
913B- 8D 2E 9A 0200      :THE FORM FEED ROUTINE EXPECTS VMODE 1 TO BE 8 LPI
913B- 8D 2E 9A 0201      :AND VMODE 0 NOT
913E- BD 17 9A 0202      VMODE      STA TEMP
913E- BD 17 9A 0203      STA VM
9141- 0A 0204      :MULTIPLY ACCU BY 5
9142- 0A 0205      ASL A
9143- 18 0206      ASL A
9144- 6D 2E 9A 0207      CLC
9144- 6D 2E 9A 0208      ADC TEMP
9147- AA 0209      :SEND FOUR CODES (BUT NO ZEROES) TO PRINTER
9148- A0 04 0210      TAX
9148- A0 04 0211      LDY #4
914A- BD 5B 9A 0212      VM1      LDA VER. X
914D- F0 03 0213      BEQ VM2
914F- 20 0C 40 0214      JSR PRINTER
9152- E8 0215      VM2      INX
9153- 88 0216      DEY
9154- D0 F4 0217      BNE VM1
9156- BD 5B 9A 0218      :5TH IN TABLE IS LINE HEIGHT (TIMES 1/48 INCH)
9159- BD 15 9A 0219      LDA VER. X
9159- BD 15 9A 0220      STA VSIZ
915C- 60 0221      RTS
915D- A2 00 0222      :CLEARBUFS - CLEAR ALL BUFFERS AND 'USED' FLAGS
915F- A9 20 0223      CLEARBUFS      LDX #0
9161- BD 2B 9A 0224      LDA '#'
9164- 9D 84 98 0225      STA LCHAR
9164- 9D 84 98 0226      CLE1      STA BUF1. X
9167- 9D 08 99 0227      STA BUF2. X
916A- 9D 8C 99 0228      STA BUF3. X

```

**DE 6502 KENNER**

916D- E8	0229	INX
916E- EO 84	0230	CPX #BUFLEN
9170- DO F2	0231	BNE CLE1
	0232	:CLEAR THE FLAGS
9172- AE 10 9A	0233	LDX USE2
9175- BE 11 9A	0234	STX USED2
9178- AE 12 9A	0235	LDX UNDERL
917B- BE 13 9A	0236	STX UNDERLD
	0237	:START AT LEFT BORDER
917E- AE 1F 9A	0238	LDX LBORD
9181- BE 18 9A	0239	STX HCNT
9184- EO 60	0240	RTS
	0241 :SENDBUFS	-SEND THE BUFFERS TO SCREEN AND/OR PRINTER
9185- AD 2A 9A	0242 SENDBUFS	LDA FILL
9188- FO 06	0243	BEQ SEO
	0244	:FILL<>0 MEANS FILL OUT LINES
	0245	:FIRST ALL TO THE LEFT
918A- 20 26 98	0246	JSR SWEEP
	0247	:THEN FILL OUT
918D- 20 4D 97	0248	JSR FILLOUT
	0249	:ANYTHING IN BUF1 ?
	0250	:IF NOT, DON'T DO PASS 1
9190- AE 18 9A	0251 SEO	LDX HCNT
9193- AO 00	0252	LDY #\$0
9195- EC 1F 9A	0253	CPX LBORD
9198- FO OC	0254	BEQ SE10
	0255	:PASS 1 : PRINT BUF1
919A- B9 84 98	0256 SE1	LDA BUF1.Y
919D- 20 3C 90	0257	JSR OUT
91A0- C8	0258	INY
91A1- CC 14 9A	0259	CPY HSIZE
91A4- DO F4	0260	BNE SE1
	0261	:SEND CR AND IF WANTED LINE FEED
91A6- A9 0D	0262 SE10	LDA #\$D
91AB- 20 3C 90	0263	JSR OUT
91AB- AD 27 9A	0264	LDA CTRL
91AE- FO 03	0265	BEQ SE11
91B0- 20 3C 90	0266	JSR OUT
	0267	:SECOND BUFFER USED ?
	0268	:IF NOT DON'T DO PASS 2
91B3- AD 11 9A	0269 SE11	LDA USED2
91B6- FO 0E	0270	BEQ SE3
91B8- AO 00	0271	LDY #0
	0272	:PASS 2 : EXTRA DARK AND SPECIAL USE (e" ETC.)
91BA- B9 08 99	0273 SE2	LDA BUF2.Y
91BD- 20 3C 90	0274	JSR OUT
91C0- C8	0275	INY
91C1- CC 14 9A	0276	CPY HSIZE
91C4- DO F4	0277	BNE SE2
	0278	:SEND CR AND IF WANTED. LF
91C6- A9 0D	0279 SE3	LDA #\$D
91C8- 20 3C 90	0280	JSR OUT
91CB- AD 28 9A	0281	LDA CTRL+1
91CE- FO 03	0282	BEQ SE30
91D0- 20 3C 90	0283	JSR OUT
	0284	:BUF3 USED ? IF NOT DON'T DO PASS 3
91D3- AD 13 9A	0285 SE30	LDA UNDERLD
91D6- FO 1B	0286	BEQ SE5
91D8- AO 00	0287	LDY #0
	0288	:PASS 3 : UNDERLINE
91DA- B9 8C 99	0289 SE4	LDA BUF3.Y
91DD- 20 3C 90	0290	JSR OUT
91E0- C8	0291	INY
91E1- CC 14 9A	0292	CPY HSIZE
91E4- DO F4	0293	BNE SE4
	0294	:SEND CR AND IF WANTED. LF
91E6- A9 0D	0295	LDA #\$D
91E8- 20 3C 90	0296	JSR OUT
91EB- AD 29 9A	0297	LDA CTRL+2
91EE- FO 03	0298	BEQ SE5
91F0- 20 3C 90	0299	JSR OUT
	0300	:BREAKTEST AFTER A LINE HAS BEEN PRINTED
91F3- 2C 80 1A	0301 SE5	BIT BREAK
91F6- 30 06	0302	BMI GOCL
	0303	:BREAK : CLEAR BUFFERS AND GOTO 'MOSER'
91F8- 20 5D 91	0304	JSR CLEARBUFS
91FB- 4C 03 20	0305	JMP WARMST

# DE6502 KENNER

```

91FE- 4C 5D 91 0306 :NO BREAK : CLEAR BUFFERS
91FE- 4C 5D 91 0307 GOCL JMP CLEARBUFS
91FE- 4C 5D 91 0308 ;MAIN - MAIN PROGRAM
9201- 20 A5 90 0309 MAIN JSR INIT2
9201- 20 A5 90 0310 :LOOP - MAIN LOOP : GET CHARACTER, SEND CHARACTER TO BUFFER
9204- AD 2B 9A 0311 LOOP LDA LCHAR
9204- AD 2B 9A 0312 :END OF 'MOSER' LINE ? THEN EXTRA SPACE !
9207- 10 07 0313 BPL NOXTRASP
9209- A9 20 0314 LDA #
920B- 8D 2C 9A 0315 STA CCHAR
920E- D0 03 0316 BNE XTRASP
9210- 20 6B 90 0317 :ELSE JUST GET NEXT CHARACTER
9210- 20 6B 90 0318 NOXTRASP JSR NEXT
9213- C9 5B 0319 XTRASP CMP #' '
9215- D0 03 0320 :DOES A COMMAND BEGIN HERE ?
9217- 4C 17 93 0321 BNE NOCOM
9217- 4C 17 93 0322 JMP COMM
921A- 18 0323 :NO COMMAND, SO JUST PUT TO BUFFER
921A- 18 0324 NOCOM CLC
921B- 6D 23 9A 0325 :OFFSET IS ADDED FOR GRAPHICS
921B- 6D 23 9A 0326 ADC OFFSET
921E- AA 0327 TAX
922B- 8A 0328 :WAS LAST CHARACTER A SPACE ?
921F- AD 2B 9A 0329 LDA LCHAR
9222- 29 7F 0330 AND #%01111111
9224- C9 20 0331 CMP '#'
9226- 8D 2B 9A 0332 STA LCHAR
9229- D0 05 0333 BNE NOSP
922B- 8A 0334 TXA
922C- C9 20 0335 :AND IS THIS CHARACTER A SPACE ?
922C- C9 20 0336 CMP '#'
922E- F0 D4 0337 :IF BOTH WERE SPACES, FORGET THIS SPACE
9230- 8A 0338 BEQ LOOP
9231- 20 37 92 0340 TXA
9234- 4C 04 92 0341 JSR PUT
9234- 4C 04 92 0342 :PUT - PUT A CHARACTER TO BUFFER
9237- AE 18 9A 0343 PUT LDY HCNT
923A- 9D 84 98 0344 STA BUF1.X
923A- 9D 84 98 0345 :IF WANTED, TO SECOND BUFFER
923D- AC 10 9A 0346 LDY USE2
9240- F0 03 0347 BEQ NOT2
9242- 9D 08 99 0348 STA BUF2.X
9245- 48 0349 NOT2 PHA
9246- AD 2C 9A 0350 :REMEMBER THIS CHARACTER
9249- 8D 2B 9A 0351 LDA CCHAR
924C- AD 12 9A 0352 STA LCHAR
924F- F0 03 0353 LDA UNDERL
924F- F0 03 0354 BEQ NOUNDE
9251- 9D 8C 99 0355 :IF WANTED, UNDERLINE
9254- 68 0356 STA BUF3.X
9254- 68 0357 NOUNDE PLA
9255- EE 18 9A 0358 INC HCNT
9258- AC 18 9A 0359 LDY HCNT
925B- CC 20 9A 0360 CPY RBORD
925B- CC 20 9A 0361 :RIGHT HAND BORDER REACHED ?
925E- D0 06 0362 BNE ENDPUT
925E- D0 06 0363 :IF SO, PRINT LINE
9260- 20 70 92 0364 :DON'T CUT WORDS IN TWO
9260- 20 70 92 0365 JSR CUTOFF
9263- 20 85 91 0366 :PRINT BUFFERS
9266- 60 0367 JSR SENDBUFS
9266- 60 0368 ENDPUT RTS
9267- A5 20 0369 :DEC PNT - PNT == PNT-1 (TWO BYTE DECREMENT)
9269- D0 02 0370 DEC PNT LDA *PNT
9269- D0 02 0371 BNE DEC1
926B- C6 21 0372 DEC *PNT+1
926D- C6 20 0373 DEC1 DEC *PNT
926F- 60 0374 RTS
9270- A0 00 0375 :CUTOFF - MAKE SURE WORDS AREN'T SPLIT
9270- A0 00 0376 CUTOFF LDY #0
9272- B1 20 0377 LDA (PNT), Y
9274- 29 7F 0378 AND #$7F
9276- C9 20 0379 CMP '#'
9276- C9 20 0380 :IF NEXT CHARACTER IS SPACE : NO PROBLEM

```

```

9278- D0 30    0381      :ELSE GOTO NOSPACE
927A- 60        0382      BNE NOSPACE
927A- 60        0383      RTS
927B- 20 67 92  0384      :LAST - OPPOSITE OF NEXT : PNT MOVES BACK TO LAST CHARACTER
927E- 20 67 92  0385      LAST JSR DECPNT
9281- 20 67 92  0386      JSR DECPNT
9284- A0 00    0387      JSR DECPNT
9286- B1 20    0388      LDY #0
9288- C9 9F    0389      LDA (PNT),Y
9288- C9 9F    0390      CMP #$9F
928A- B0 0E    0391      :BYTES >$9F ARE MOSERS END OF LINE BYTES
928A- B0 0E    0392      BCS LA2
928C- 20 BF 92  0393      :NO END OF LINE, MOVE ONLY ONE BYTE BACK
928C- 20 BF 92  0394      RTS
928F- E6 20    0395      JSR INCPNT
9291- D0 02    0396      INC *PNT
9291- D0 02    0397      BNE LA1
9293- E6 21    0398      INC *PNT+1
9295- B1 20    0399      LA1 LDA (PNT),Y
9295- B1 20    0400      .LI WORDPR.M02

F02 1C88 4292-5F1A

9297- C9 9F    0400      :ANOTHER END OF LINE CHECK
9299- 60        0401      CMP #$9F
9299- 60        0402      RTS
929A- C8        0403      :END OF LINE FOUND: SAVE LINE NUMBER
929A- C8        0404      LA2 INY
929B- 48        0405      PHA
929C- B1 20    0406      LDA (PNT),Y
929E- BD 08 01  0407      STA LINENR
92A1- C8        0408      INY
92A2- B1 20    0409      LDA (PNT),Y
92A4- BD 09 01  0410      STA LINENR+1
92A7- 68        0411      PLA
92AB- 38        0412      SEC
92A9- 60        0413      RTS
92AA- 20 7B 92  0414      :NO SPACE WAS FOUND: THIS WORD GOES TO THE NEXT LINE
92AA- 20 7B 92  0415      NOSPACE JSR LAST
92AD- B0 06    0416      :MOVE BACK UNTIL A SPACE OR END OF LINE IS FOUND
92AF- 29 7F    0417      BCS NO1
92B1- C9 20    0418      AND #$7F
92B1- C9 20    0419      CMP #
92B3- D0 F5    0420      BNE NOSPACE
92B5- 20 6B 90  0421      NO1 JSR NEXT
92B8- AE 20 9A  0422      :DO THE SAME IN BUF1
92BB- CA        0423      LDX RBORD
92BC- BD 84 98  0424      DEX
92BC- BD 84 98  0425      NO2 LDA BUF1,X
92BF- C9 20    0426      CMP '#'
92C1- F0 03    0427      BEQ NO3
92C3- CA        0428      DEX
92C4- D0 F6    0429      BNE NO2
92C6- A9 20    0430      :MAKE THE END OF THE LINE ALL SPACES
92C8- 9D 84 98  0431      NO3 LDA '#'
92C8- 9D 84 98  0432      STA BUF1,X
92C8- 9D 08 99  0433      STA BUF2,X
92CE- 9D 8C 99  0434      STA BUF3,X
92D1- E8        0435      INX
92D2- EC 20 9A  0436      CPX RBORD
92D5- D0 EF    0437      BNE NO3
92D7- 60        0438      RTS
92D8- A9 00    0439      :DECIM - READ IN A DECIMAL NUMBER OR ASCII VALUE
92DA- BD 2D 9A  0440      DECIM LDA #0
92DD- 20 6B 90  0441      STA NUMB
92DD- 20 6B 90  0442      DECIM1 JSR NEXT
92E0- C9 27    0443      :' FOUND MEANS ASCII CODE FOLLOWS
92E2- F0 0C    0444      CMP '#'
92E2- F0 0C    0445      BEQ ASCII
92E4- 20 B8 96  0446      :TEST WHETHER IT'S A NUMBER
92E7- 10 0E    0447      JSR NUMTEST
92E7- 10 0E    0448      BPL DECIM3
92E9- 20 7B 92  0449      :NOT A NUMBER, MOVE BACK AND EXIT
92EC- AD 2D 9A  0450      JSR LAST
92EF- 60        0451      LDA NUMB
92EF- 60        0452      RTS
92EF- 60        0453      :ASCII CODE, READ IT AND EXIT

```

# DE 6502 KENNER

```

92F0- 20 6B 90 0454 ASCII   JSR NEXT      1220 08 30 0A -3821
92F3- 8D 2D 9A 0455 STA NUMB     0000 00 00 00 -3822
92F6- 60          0456 RTS        0000 00 00 00 -3823
92F7- 38          0457          :IT'S A NUMBER: NUMB=NUMB*10+(THIS NUMBER)
92F8- E9 30 0458 DECIM3   SEC        0000 00 00 00 -3824
92FA- 48          0459 SBC #$30    0000 00 00 00 -3825
92FB- AD 2D 9A 0460 PHA        0000 00 00 00 -3826
92FB- AD 2D 9A 0461          :NUMB=10*NUMB MEANS NUMB=2*(NUMB+4*NUMB)
92FB- AD 2D 9A 0462 LDA NUMB    0000 00 00 00 -3827
92FB- AD 2D 9A 0463          :MULTIPLY BY FOUR
92FE- 0E 2D 9A 0464 ASL NUMB    0000 00 00 00 -3828
9301- 0E 2D 9A 0465 ASL NUMB    0000 00 00 00 -3829
9304- 18          0466 CLC        0000 00 00 00 -382A
9305- 6D 2D 9A 0467          :ADD ONE TIME
9305- 6D 2D 9A 0468 ADC NUMB    0000 00 00 00 -382B
9305- 6D 2D 9A 0469          :MULTIPLY BY TWO
9308- 0A          0470 ASL A      0000 00 00 00 -382C
9309- 8D 2D 9A 0471 STA NUMB    0000 00 00 00 -382D
930C- 68          0472 PLA        0000 00 00 00 -382E
930D- 18          0473          :ADD (THIS NUMBER)
930D- 18          0474 CLC        0000 00 00 00 -382F
930E- 6D 2D 9A 0475 ADC NUMB    0000 00 00 00 -3830
9311- 8D 2D 9A 0476 STA NUMB    0000 00 00 00 -3831
9314- 4C DD 92 0477 JMP DECIM1
9317- 20 6B 90 0481 COMM      JSR NEXT      0000 00 00 00 -3832
931A- C9 4C 0482          :WHAT FOLLOWS IS A SKIP LIST OF COMMANDS
931C- D0 03 0483          :IT'S RATHER LONG AND SIMPLE- PLEASE DON'T GET BORED
931E- 4C E6 93 0484          =====
9321- C9 52 0485 NOL       BNE NOL      = Bi: het ter perse gaan van deze editie bleek dat in =
9323- D0 03 0486          CMP #'L      = het Printer Facilities Program van het artikel Word =
9325- 4C 0A 94 0487          BNE NOR      = Processing with Carl Moser's Text Editor van auteur =
9328- C9 44 0488 NOR       CMP #'R      = Hans Buurman toch nog een kleine verbetering nodia =
932A- D0 03 0489          BNE NOD      = te zijn.
932C- 4C 27 94 0490          JMP RCOM    = Regel 116 had moeten worden gevuld door 4C 03 20 =
932F- C9 48 0491 NOD       JMP DCOM    = JMP WARMST.
9331- D0 03 0492          CMP #'H      = Deze kleinheid kan echter snel verholpen worden =
9333- 4C 2D 94 0493          BNE NOH      = door de volgende wijziging:
9336- C9 56 0494 NOH       JMP HCOM    = 9042 4C 83 9A      JMP LABEL =
9338- D0 03 0495          CMP #'V      = 9A83 20 85 91 LABEL JSR SENDBUFS =
933A- 4C 38 94 0496          BNE NOV      = 9A86 4C 03 20      JMP WARMST =
933D- C9 55 0497 NOV       CMP #'U      =====
933F- D0 03 0498          BNE NOU      = WIJ STAAN OP DE H.C.C.-COMPUTERDAGEN 16 EN 17 NOV 1984 =
9341- 4C 43 94 0499          JMP UCOM    = = JAARBEURS UTRECHT =
9344- C9 3A 0500 NOU       CMP #' :      = = DNTMOET DAAR ONS 6502-TEAM =
9346- F0 CF 0501          BEQ COMM    =====
9348- C9 32 0502          CMP #'2      =====
934A- D0 03 0503          BNE NOT2C   =====
934C- 4C 68 94 0504          JMP C2COM   =====
934F- C9 31 0505 NOT2C   CMP #'1      =====
9351- D0 03 0506          BNE NOT1      =====
9353- 4C 71 94 0507          JMP C1COM   =====
9356- C9 57 0508 NOT1    CMP #'W      =====
9358- D0 03 0509          BNE NOW      =====
935A- 4C 79 94 0510          JMP WCOM    =====
935D- C9 4A 0511 NOW     CMP #'J      =====
935F- D0 03 0512          BNE NOJ      =====
9361- 4C 82 94 0513          JMP JC0M    =====
9364- C9 45 0514 NOJ     CMP #'E      =====
9366- D0 03 0515          BNE NOE      =====
9368- 4C A7 94 0516          JMP ECOM    =====
936B- C9 54 0517 NOE     CMP #'T      =====
936D- D0 03 0518          BNE NOT      =====
936F- 4C B0 94 0519          JMP TCOM    =====
9372- C9 2E 0520 NOT     CMP #'.
9374- D0 03 0521          BNE NOPER   =====
9376- 4C B4 94 0522          JMP PERCOM =====
9379- C9 4F 0523 NOPER   CMP #'0      =====
937B- D0 03 0524          BNE NOO      =====
937D- 4C E6 94 0525          JMP OCOM    =====
9380- C9 53 0526 NOO     CMP #'S      =====
9382- D0 03 0527          BNE NOS      =====
9384- 4C F8 94 0528          JMP SCOM    =====
9387- C9 50 0529 NOS     CMP #'P      =====
9389- D0 03 0530          BNE NOP      =====

```



# DE6502 KENNER

```

938B- 4C 0F 95 0531 JMP PCOM
938E- C9 2B 0532 NOP CMP #''
9390- D0 03 0533 BNE NOPLUS = ASM 65C 65CXXX-CROSS-ASSEMBLER V1.0 =
9392- 4C 24 95 0534 JMP PLUSCOM = FOR SENIOR MONITOR =
9395- C9 42 0535 NOPLUS CMP #'B = (= JUNIOR-como. with PROTON D.O.S. ) =
9397- D0 03 0536 BNE NOB = The ASM 65C 65CXXX-CROSS-ASSEMBLER handles all new =
9399- 4C 2D 95 0537 JMP BCOM = instructions of 65C02-processors and is fully comba-
939C- C9 41 0538 NOB CMP #'A = table with 6502. Author: R. Banen. Schiedam. The =
939E- D0 03 0539 BNE NOA = Netherlands. Memory space needed: $3000 - $386A.
93A0- 4C 63 95 0540 JMP ACOM = Complete source listing available. English version:
93A3- C9 49 0541 NOA CMP #'I = Fl. 52.50 incl. postage, packing, admin. etc.
93A5- D0 03 0542 BNE NOI = Foreign countries (CEPT countries excl.) Fl. 6.75
93A7- 4C CC 96 0543 JMP ICOM = extra transfers.
93AA- C9 23 0544 NOI CMP #'# = To be paid to Mr. W.L. van Peit. postal account nr.
93AC- D0 03 0545 BNE NORC = 841433 at Krimoen a.d. IJssel, or to bank-account nr.
93AE- 4C 8E 96 0546 JMP RCCOM = 44.11.06.471 of Mr. W.L. van Peit AMRO-Bank at
93B1- C9 46 0547 NORC CMP #'F = Krimoen a.d. IJssel. Eurocheque allowed.
93B3- D0 03 0548 BNE NOF = =====
93B5- 4C 42 96 0549 JMP FCOM = =====
93B8- C9 4B 0550 NOF CMP #'K = =====
93BA- D0 03 0551 BNE NOK = LAAT UW ZELF ONTWIKKELDE PROGRAMMA'S VOOR UW C-64.
93BC- 4C C6 96 0552 JMP KCOM = APPLES, BBC, OSI, PET, ETC. ETC. NIET LIGGEN, MAAR =
93BF- C9 43 0553 NOK CMP #'C = STUUR HET OP NAAR DE REDAKTIE, U HELPT ER NIET AL-
93C1- D0 03 0554 BNE NOC = LEEN ANDEREN MEE. U HELPT HEN ER OOK MEE IDEEEN =
93C3- 4C 08 97 0555 JMP CCOM = TE ONTIKKELLEN DIE WEER GEOPUBLICEERD KUNNEN WOR-
93C6- C9 51 0556 NOC CMP #'Q = DEN. EN DAT IS UW BELANG !
93C8- D0 03 0557 BNE NOQ = =====
93CA- 4C 1D 98 0558 JMP QCOM = =====
93CD- 8D 28 01 0559 NOQ :UNKNOWN COMMAND: BC ERROR !
93D0- A2 BC 0560 STA FILENR
93D0- A2 BC 0561 LDX #$BC
93D2- 4C 51 94 0562 JMP ERROR
93D5- 20 6B 90 0563 :COMCOM COMMON RETURN ADRESS FOR ALL COMMANDS
93D5- 20 6B 90 0564 COMCOM JSR NEXT
93D5- 20 6B 90 0565 :: MEANS OTHER COMMAND FOLLOWS
93D8- C9 3A 0566 CMP #'':
93DA- F0 07 0567 BEQ TOCOM
93DC- C9 5D 0568 :: MEANS END OF COMMANDS
93DE- D0 F5 0569 CMP #'J
93E0- 4C 04 92 0570 BNE COMCOM
93E3- 4C 17 93 0571 JMP LOOP
93E6- 20 D8 92 0572 TOCOM JMP COMM
93E6- 20 D8 92 0573 ;LCOM - LEFT BORDER COMMAND
93E6- 20 D8 92 0574 LCOM JSR DECIM
93E9- CD 14 9A 0575 :: MAY NOT BE LARGER THAN LINE LENGTH
93EC- B0 14 0576 CMP HSIZ
93EE- 8D 1F 9A 0577 BCS ERR30
93F1- CD 18 9A 0578 STA LBORD
93F1- CD 18 9A 0579 :: IF TO THE RIGHT OF PRESENT POSITION, CHANGE THE LATTER
93F4- 90 03 0580 CMP HCNT
93F4- 90 03 0581 BCC LCOM1
93F6- 8D 18 9A 0582 STA HCNT
93F6- 8D 18 9A 0583 :: THERE MUST STILL BE ROOM FOR PARAGRAPH TABS
93F9- AD 21 9A 0584 LCOM1 LDA PTAB
93FC- 8D 2D 9A 0585 STA NUMB
93FF- 4C 85 94 0586 JMP JC0M1
9402- A2 30 0587 :ERROR 30 MESSAGE
9404- 8D 28 01 0588 ERR30 LDX #$30
9407- 4C 51 94 0589 STA FILENR
9407- 4C 51 94 0590 JMP ERROR
940A- 20 D8 92 0591 :RCOM - LEFT BORDER COMMAND
940A- 20 D8 92 0592 RCOM JSR DECIM
940D- F0 F3 0593 :: MUST BE INSIDE THE LINE
940F- CD 14 9A 0594 BEQ ERR30
9412- B0 EE 0595 CMP HSIZ
9412- B0 EE 0596 BCS ERR30
9414- CD 18 9A 0597 :: IF TO THE LEFT OF CURRENT POSITION, START NEW LINE
9414- CD 18 9A 0598 CMP HCNT
9417- F0 05 0599 BEQ RCOM1
9419- B0 03 0600 BCS RCOM1
941B- 20 85 91 0601 JSR SENDBUFS
941B- 20 85 91 0602 :: THERE MUST BE ROOM FOR PARAGRAPH TABS
941E- AD 2D 9A 0603 RCOM1 LDA NUMB
9421- 8D 20 9A 0604 STA RBORD
9424- 4C F9 93 0605 JMP LCOM1
9424- 4C F9 93 0606 :DCOM - START NEW LINE
9427- 20 85 91 0607 DCOM JSR SENDBUFS

```

# DE 6502 KENNER

942A- 4C D5 93	0608	JMP COMCOM	0600	R0 05 03 -0040
	0609	:HCOM - CHANGE HORIZONTAL MODE	0600	R1 09 -0002
942D- 20 D8 92	0610	HCOM JSR DECIM	0600	7630
9430- 29 07	0611	AND #X111	0600	0230 8030 R0 81 00 -1040
9432- 20 11 91	0612	JSR HMODE	0600	0530 R0 00 03 -0040
9435- 4C D5 93	0613	JMP COMCOM	0600	0630 R0 00 03 -0040
	0614	:VCOM - CHANGE VERTICAL MODE	0600	0630 R0 00 04 -0040
9438- 20 D8 92	0615	VCOM JSR DECIM	0600	0630 R0 00 04 -0040
943B- 29 07	0616	AND #X111	0600	0630 R0 00 04 -0040
943D- 20 3B 91	0617	JSR VMODE	0600	0630 R0 00 04 -0040
9440- 4C D5 93	0618	JMP COMCOM	0600	0630 R0 00 04 -0040
	0619	:UCOM - UNDERLINE	0600	0630 R0 00 04 -0040
9443- 20 D8 92	0620	UCOM JSR DECIM	0600	0630 R0 00 04 -0040
9446- 8D 12 9A	0621	STA UNDERL	0600	0630 R0 00 04 -0040
9449- F0 03	0622	BEQ COMCOM2	0600	0630 R0 00 04 -0040
944B- 8D 13 9A	0623	STA UNDERLD	0600	0630 R0 00 04 -0040
944E- 4C D5 93	0624	JMP COMCOM	0600	0630 R0 00 04 -0040
	0625	:ERROR - ISSUE ERROR MESSAGE	0600	0630 R0 00 04 -0040
9451- A5 20	0626	ERROR LDA *PNT	0600	0630 R0 00 04 -0040
9453- 48	0627	PHA	0600	0630 R0 00 04 -0040
9454- A5 21	0628	LDA *PNT+1	0600	0630 R0 00 04 -0040
9456- 48	0629	PHA	0600	0630 R0 00 04 -0040
	0630	:MOVE BACK TO START OF LINE FOR LINE NUMBER	0600	0630 R0 00 04 -0040
9457- 20 7B 92	0631	ERROR2 JSR LAST	0600	0630 R0 00 04 -0040
945A- 90 FB	0632	BCC ERROR2	0600	0630 R0 00 04 -0040
945C- 20 4B 24	0633	JSR ERRORM	0600	0630 R0 00 04 -0040
	0634	:CONTINUE	0600	0630 R0 00 04 -0040
945F- 68	0635	PLA	0600	0630 R0 00 04 -0040
9460- 85 21	0636	STA *PNT+1	0600	0630 R0 00 04 -0040
9462- 68	0637	PLA	0600	0630 R0 00 04 -0040
9463- 85 20	0638	STA *PNT	0600	0630 R0 00 04 -0040
9465- 4C D5 93	0639	JMP COMCOM	0600	0630 R0 00 04 -0040
	0640	:C2COM - PRINT EXTRA DARK	0600	0630 R0 00 04 -0040
9468- 8D 10 9A	0641	C2COM STA USE2	0600	0630 R0 00 04 -0040
946B- 8D 11 9A	0642	STA USED2	0600	0630 R0 00 04 -0040
946E- 4C D5 93	0643	JMP COMCOM	0600	0630 R0 00 04 -0040
	0644	:C1COM - PRINT NORMAL	0600	0630 R0 00 04 -0040
9471- A9 00	0645	C1COM LDA #0	0600	0630 R0 00 04 -0040
9473- 8D 10 9A	0646	STA USE2	0600	0630 R0 00 04 -0040
9476- 4C D5 93	0647	JMP COMCOM	0600	0630 R0 00 04 -0040
	0648	:WCOM - TERMINAL WITDH	0600	0630 R0 00 04 -0040
9479- 20 D8 92	0649	WCOM JSR DECIM	0600	0630 R0 00 04 -0040
947C- 8D 22 9A	0650	STA TWIDTH	0600	0630 R0 00 04 -0040
947F- 4C D5 93	0651	JMP COMCOM	0600	0630 R0 00 04 -0040
	0652	:JCOM - CHANGE PARAGRAPH TABULATION	0600	0630 R0 00 04 -0040
9482- 20 D8 92	0653	JCOM JSR DECIM	0600	0630 R0 00 04 -0040
	0654	:JCOM1 - CHECK TO SEE IF, WITH PARAGRAPH TAB. THE RESULTING	0600	0630 R0 00 04 -0040
	0655	:LINE IS STILL 16 CHARACTERS WIDE	0600	0630 R0 00 04 -0040
	0656	:LBORD+PTAB-RBORD<=-16	0600	0630 R0 00 04 -0040
9485- 18	0657	JCOM1 CLC	0600	0630 R0 00 04 -0040
9486- 6D 1F 9A	0658	ADC LBORD	0600	0630 R0 00 04 -0040
9489- 38	0659	SEC	0600	0630 R0 00 04 -0040
948A- ED 20 9A	0660	SBC RBORD	0600	0630 R0 00 04 -0040
948D- B0 04	0661	BCS ERR31	0600	0630 R0 00 04 -0040
948F- E9 FO	0662	SBC #\$FO	0600	0630 R0 00 04 -0040
9491- 90 0B	0663	BCC JACC	0600	0630 R0 00 04 -0040
	0664	:LINE TOO SMALL, ERROR 31	0600	0630 R0 00 04 -0040
9493- AD 2D 9A	0665	ERR31 LDA NUMB	0600	0630 R0 00 04 -0040
9496- 8D 28 01	0666	STA FILENR	0600	0630 R0 00 04 -0040
9499- A2 31	0667	LDX #\$31	0600	0630 R0 00 04 -0040
949B- 4C 51 94	0668	JMP ERROR	0600	0630 R0 00 04 -0040
	0669	:PARAGRAPH TAB ACCEPTED	0600	0630 R0 00 04 -0040
949E- AD 2D 9A	0670	JACC LDA NUMB	0600	0630 R0 00 04 -0040
94A1- 8D 21 9A	0671	STA PTAB	0600	0630 R0 00 04 -0040
94A4- 4C D5 93	0672	JMP COMCOM	0600	0630 R0 00 04 -0040
	0673	:ECOM - ERROR CONTROL COMMAND	0600	0630 R0 00 04 -0040
94A7- 20 D8 92	0674	ECOM JSR DECIM	0600	0630 R0 00 04 -0040
94AA- 8D 12 01	0675	STA CONFL	0600	0630 R0 00 04 -0040
94AD- 4C D5 93	0676	JMP COMCOM	0600	0630 R0 00 04 -0040
	0677	:TCOM - TABULATION COMMAND	0600	0630 R0 00 04 -0040
94B0- A9 20	0678	TCOM LDA #'	0600	0630 R0 00 04 -0040
94B2- D0 02	0679	BNE PER1	0600	0630 R0 00 04 -0040
	0680	:PERCOM - TABULATE WITH PERIODS (.....)	0600	0630 R0 00 04 -0040
94B4- A9 2E	0681	PERCOM LDA #'	0600	0630 R0 00 04 -0040
94B6- 8D 2E 9A	0682	PER1 STA TEMP	0600	0630 R0 00 04 -0040
94B9- 20 D8 92	0683	JSR DECIM	0600	0630 R0 00 04 -0040
	0684	:TABULATE OUT OF THE LINE TO GET A 32 ERROR	0600	0630 R0 00 04 -0040

# DE 6502 KENNER

```

94BC- CD 20 9A 0685      CMP RBORD
94BF- BO 1A 0686      BCS ERR32
94C1- AD 18 9A 0688 PER2      :SEND THE SYMBOL IN TEMP UNTIL IN THE PROPER COLUMN
94C4- CD 2D 9A 0689      LDA HCNT
94C7- BO 09 0690      CMP NUMB
94C9- AD 2E 9A 0691      BCS PER3
94CC- 20 37 92 0692      LDA TEMP
94CF- 4C C1 94 0693      JSR PUT
94D2- AD 2E 9A 0694      JMP PER2
94D5- 8D 2B 9A 0695 PER3      :SUPPRESS SPACES AFTER A TAB
94D8- 4C D5 93 0696      LDA TEMP
94DB- AD 2D 9A 0697      STA LCHAR
94E0- 20 D8 92 0698      JMP COMCOM
94E3- 4C 51 94 0699 ERR32      :SEND A 32 ERROR MESSAGE
94E6- 20 D8 92 0700      LDA NUMB
94E9- 48 28 01 0701      STA FILENR
94EA- 29 01 0702      LDX #$32
94EC- BD 1E 9A 0703 :OCOM - OUTPUT CONTROL      JMP ERROR
94EF- 68 0704 OCOM      JSR DECIM
94F0- 29 02 0705      PHA
94F2- 8D 1D 9A 0706      AND #X1
94F5- 20 D5 93 0707      :BIT1 SWITCHES TERMINAL OUTPUT
94FB- 20 D8 92 0708      STA SCREEN
94FB- FO 0F 0709      PLA
94FD- A9 20 0710      AND #X10
94FF- 20 37 92 0711      :BIT1 SWITCHES THE PRINTER
9502- CE 2D 9A 0712      STA PRINT
9505- DO F6 0713      JSR COMCOM
9507- A9 20 0714 :SCOM - SEND A NUMBER OF SPACES
9509- 8D 2B 9A 0715 SCOM      JSR DECIM
950C- 4C D5 93 0716      BEQ SCOM2
950F- 20 85 91 0717      :SEND NUMB SPACES
9512- AD 21 9A 0718 SCOM1      LDA #
9515- 18 0719      JSR PUT
9516- 6D 1F 9A 0720      DEC NUMB
9519- 8D 18 9A 0721      BNE SCOM1
9524- 20 D8 92 0722      :SUPPRESS FOLLOWING SPACES
9527- 8D 23 9A 0723      LDA #
952A- 4C D5 93 0724      STA LCHAR
952C- A9 20 0725 SCOM2      JMP COMCOM
952F- 20 85 91 0726 :PCOM - START A NEW PARAGRAPH
9532- AE 18 9A 0727 PCOM      JSR SENDBUFS
9535- 8E 2E 9A 0728      :TABULATE
9538- 20 6B 90 0729      LDA PTAB
9541- 8E 11 9A 0730      CLC
9544- 4C D5 93 0731      ADC LBORD
9547- AE 2E 9A 0732      STA HCNT
9550- 8D 2B 9A 0733      :SUPPRESS SPACES
9553- A2 FF 0734      LDA #
9556- 8E 11 9A 0735      STA LCHAR
9559- 4C D5 93 0736      JMP COMCOM
9562- 20 D8 92 0737 :PLUSCOM - SET AN OFFSET
9565- 8D 23 9A 0738 PLUSCOM      JSR DECIM
9568- 4C D5 93 0739      STA OFFSET
9571- AE 18 9A 0740      JMP COMCOM
9574- 20 6B 90 0741 :BCOM - SEND TEXT TO SECOND BUFFER ONLY
9577- AE 18 9A 0742      :USE POSITION IN TEMP, SECOND BUFFER
9580- 8E 2E 9A 0743 BCOM      LDX HCNT
9583- 8E 11 9A 0744      STX TEMP
9586- 4C D5 93 0745      LDX #$FF
9589- 8E 11 9A 0746      STX USED2
9592- AE 18 9A 0747      :GET CHARACTER
9595- 20 6B 90 0748 BCOM1      JSR NEXT
9598- AE 18 9A 0749      :IS IT A TERMINATOR ?
959B- 8E 2E 9A 0750      CMP #'J
959E- 8E 11 9A 0751      BEQ BCOMEND
959F- 4C D5 93 0752      CMP #':
9602- AE 18 9A 0753      BEQ BCOMEND
9605- 20 6B 90 0754      :SEND TO SECOND BUFFER
9608- AE 18 9A 0755      LDX TEMP
9611- 8E 2E 9A 0756      STA BUF2.X
9614- 8E 11 9A 0757      INX
9617- 4C D5 93 0758      CPX RBORD
9620- AE 18 9A 0759      STX TEMP
9623- 8E 2E 9A 0760      :STOP IF END OF LINE REACHED
9626- DO E6 0761      BNE BCOM1

```

```

9552- 20 6B 90 0762 : THEN SKIP THE REST
9555- C9 5D 0763 BCOM2 JSR NEXT
9557- F0 04 0764 CMP #'J
9559- C9 3A 0765 BEQ BCOMEND
955B- D0 F5 0766 CMP #':
955B- D0 F5 0767 BNE BCOM2
955D- 20 7B 92 0768 : TERMINATOR FOUND, MOVE BACK AND EXIT
9560- 4C D5 93 0769 BCOMEND JSR LAST
9560- 4C D5 93 0770 JMP COMCOM
9563- 20 D8 92 0771 :ACOM - SEND AN ASCII-CODE
9566- 20 37 92 0772 ACOM JSR DECIM
9566- 20 37 92 0773 JSR PUT
9569- A9 01 0774 :SPACES MAY FOLLOW
956B- 8D 2B 9A 0775 LDA #$1
956E- 4C D5 93 0776 STA LCHAR
956E- 4C D5 93 0777 JMP COMCOM
9571- AD 15 9A 0778 ;NEWLINE - AFTER A LINEFEED, CHECK FOR END OF PAGE
9571- AD 15 9A 0779 :ADJUST POSITION COUNTER
9574- 18 0780 NEWLINE LDA VSIZE
9575- 6D 19 9A 0781 CLC
9578- 8D 19 9A 0782 ADC VCNT
957B- AD 1A 9A 0783 STA VCNT
957B- AD 1A 9A 0784 LDA VCNT+1
957E- 69 00 0785 ADC #0
9580- 8D 1A 9A 0786 STA VCNT+1
9583- 20 A7 95 0787 :CHECK FOR END OF PAGE
9586- 90 1E 0788 JSR PCHECK
9586- 90 1E 0789 BCC ENDNL
9588- A9 00 0790 ;FFCOM - START A NEW PAGE
958A- 8D 19 9A 0791 :RESET COUNTER
958A- 8D 19 9A 0792 FFCOM LDA #0
958B- 8D 1A 9A 0793 STA VCNT
958D- 8D 1A 9A 0794 STA VCNT+1
9590- AD 1B 9A 0795 LDA PAGED
9593- F0 11 0796 BEQ ENDNL
9595- AD 1C 9A 0797 :PAGES USED : ADJUST PAGE NR
9598- 18 0798 LDA PAGE
9599- F8 0799 CLC
9599- F8 0800 SED
959A- 69 01 0801 ADC #$1
959C- D8 0802 CLD
959D- 8D 1C 9A 0803 STA PAGE
95A0- 20 B6 95 0804 :INSERT OPEN LINES AND HEADING
95A3- 20 FC 95 0805 JSR BETP
95A3- 20 FC 95 0806 JSR PAGEHEAD
95A6- 60 0807 ENDNL RTS
95A7- AD 1A 9A 0808 :PCHECK - COMPARE VCNT AND PSIZE
95AA- CD 25 9A 0809 PCHECK LDA VCNT+1
95AD- DO 06 0810 CMP PSIZE+1
95AD- DO 06 0811 BNE PCH1
95AF- AD 19 9A 0812 LDA VCNT
95B2- CD 24 9A 0813 CMP PSIZE
95B5- 60 0814 PCH1 RTS
95B6- AD 26 9A 0815 :BETP - SPACE BETWEEN PAGES
95B9- C9 FF 0816 BETP LDA BSIZE
95BB- DO 04 0817 CMP #255
95BB- DO 04 0818 BNE BETP1
95BD- 20 AE 12 0819 :BSIZE=255 : JUST WAIT FOR A KEYSTROKE
95C0- 60 0820 JSR RECHA
95C1- AD 17 9A 0821 RTS
95C4- 48 0822 :LINES NEEDED: SAVE PRINTER MODE
95C1- AD 17 9A 0823 BETP1 LDA VM
95C4- 48 0824 PHA
95C5- AD 26 9A 0825 :BSIZE>127 : MODE 1 ELSE MODE 0
95C8- 30 04 0826 LDA BSIZE
95CA- A9 00 0827 BMI MO1
95CC- F0 02 0828 LDA #0
95CE- A9 01 0829 BEQ BETP2
95CE- A9 01 0830 MO1
95D0- 20 3B 91 0831 BETP2 LDA #1
95D3- AD 26 9A 0832 JSR VMODE
95D6- 29 7F 0833 :SEND DESIRED NUMBER OF LINEFEEDS. TEMP+1 IS COUNTER
95D6- 29 7F 0834 LDA BSIZE
95D8- 8D 2F 9A 0835 AND #$7F
95DB- F0 1A 0836 BETP3 STA TEMP+1
95DB- F0 1A 0837 BEQ BETP4
95DD- AD 1E 9A 0838 :SEND TO SELECTED OUTPUTS, BUT DON'T CHECK FOR PAGES
95DD- AD 1E 9A 0839 LDA SCREEN

```

```

95E0- F0 05    0839      BEQ BETP3B
95E2- A9 0A    0840      LDA #$A
95E4- 20 80 0C  0841      JSR OUTCH
95E7- AD 1D 9A  0842      BETP3B
95EA- F0 05    0843      LDA PRINT
95EC- A9 0A    0844      BEQ BETP3A
95EE- 20 0C 40  0845      LDA #$A
95F1- CE 2F 9A  0846      JSR PRINTER
95F4- 4C DB 95  0847      DEC TEMP+1
                           JMP BETP3
                           : RESTORE PRINTER MODE, EXIT
95F7- 68       0848      PLA
                           .LI WORDPR.M03
                           0850

```

F03 1CE6 4292-5F78

```

95F8- 20 3B 91  0850      JSR VMODE
                           RTS
95FB- 60       0851
                           0852 :PAGEHEAD - PRINT PAGE HEADING (WITH LINE NR)
                           :FIRST: MOVE TO HSIZE/2-1 (THE MIDDLE)
                           0853 LDA HSIZE
95FC- AD 14 9A  0854 PAGEHEAD
                           LSR A
                           0855 STA TEMP+1
9600- BD 2F 9A  0856 DEC TEMP+1
9603- CE 2F 9A  0857 JMP BETP3
                           :SEND ENOUGH SPACES
                           0858 LDA '#'
9606- A9 20    0859 PAG1   JSR OUT
                           DEC TEMP+1
9608- 20 3C 90  0860 BNE PAG1
                           :GET PAGE NR HIGH NIBBLE
960B- CE 2F 9A  0861 LDA PAGE
                           LSR A
                           0862 STA TEMP+1
960E- D0 F6    0863 DEC TEMP+1
                           LSR A
                           :IF ZERO, PRINT A SPACE
                           0864 LDA '#'
9613- 4A       0865 JSR OUT
                           BNE PAG2A
                           LDA '#'
9614- 4A       0866 BNE PAG2B
                           CLC
9615- 4A       0867 ADC #$30
                           JSR OUT
                           :PRINT CR AND THREE LINEFEEDS
9616- 4A       0868 LDA PAGE
                           LSR A
                           0869 STA TEMP+1
9617- D0 04    0870 BNE PAG2A
                           LDA '#'
9619- A9 20    0871 BNE PAG2B
                           CLC
961B- D0 03    0872 ADC #$30
                           JSR OUT
                           :NOTE: TEMP+1 IS 00 AFTER A FORM FEED
961D- 18       0873 PAG2A
                           LDA '#'
961E- 69 30    0874 JSR OUT
                           :PRINT CR AND THREE LINEFEEDS
9620- 20 3C 90  0875 PAG2B
                           LDA PAGE
                           AND #$F
                           ORA #%110000
                           JSR OUT
                           LDA #$D
                           JSR OUT
                           LDA '#3
                           STA TEMP+1
                           LDA #$A
                           JSR OUT
                           DEC TEMP+1
                           BNE PAG3
                           :NOTE: TEMP+1 IS 00 AFTER A FORM FEED
9623- AD 1C 9A  0877 LDA '#'
                           JSR OUT
                           LDA PAGE
                           IF NOT USED BEFORE, JUST START NEW PAGE
                           0891 :FCOM - EJECT TO TOP OF NEXT PAGE, SEE .EJ PSEUDO OP
                           0892 :
                           0893 FCOM
                           LDA PAGED
                           BEQ PAGON
                           :TEMP+1 > 0, WILL BE 0 AFTER FORM FEED
                           STA TEMP+1
                           JSR SENDBUFS
                           :SAVE PRINTER MODE
                           LDA VM
                           PHA
                           :MODE 1 UNTIL A MULTIPLE OF 8/48 INCH REACHED
                           LDA #1
                           JSR VMODE
                           :IS IT REACHED ?
                           LDA VCNT
                           AND #%111
                           BEQ FCOM2
                           :TEST IF FF ALREADY GIVEN
                           LDA TEMP+1
                           BEQ ENDFCOM
                           :NO, SEND LF AND CONTINUE
                           0901
                           0902
                           0903
                           0904
                           0905 FCOM1
                           0906
                           0907
                           0908
                           0909
                           0910
                           0911

```

**DE6502 KENNER**

```

9662- A9 0A 0912 LDA #$A
9664- 20 38 90 0913 JSR OUTO
9667- 4C 56 96 0914 JMP FCOM1
9615 :NOW TRY MODE 0
966A- A9 00 0916 FCOM2 LDA #0
966C- 20 3B 91 0917 JSR VMODE
9618 :TEST IF FF ALREADY GIVEN
966F- AD 2F 9A 0919 FCOM2B LDA TEMP+1
9672- F0 08 0920 BEQ ENDFCOM
9621 :NO, SEND LINEFEEDS
9674- A9 0A 0922 LDA #$A
9676- 20 38 90 0923 JSR OUTO
9679- 4C 6F 96 0924 JMP FCOM2B
9625 :FF GIVEN, RESTORE PRINTER MODE AND EXIT
967C- 68 0926 ENDFCOM PLA
967D- 20 3B 91 0927 JSR VMODE
9680- 4C D5 93 0928 JMP COMCOM
9629 :PAGON - BEGIN PAGED MODE, START NEW PAGE
9683- A9 FF 0930 PAGON LDA #$FF
9685- 8D 1B 9A 0931 STA PAGED
9688- 20 88 95 0932 JSR FFCOM
968B- 4C D5 93 0933 JMP COMCOM
9634 :RCCOM - READ A DECIMAL PAGE NUMBER
968E- A9 00 0935 RCCOM LDA #$0
9690- 8D 1C 9A 0936 STA PAGE
9693- 20 6B 90 0937 RC1 JSR NEXT
9638 :TEST FOR NUMBER
9696- 20 B8 96 0939 JSR NUMTEST
9699- 30 17 0940 BMI RC2
9641 :IT IS, SHIFT IT IN
969B- 0E 1C 9A 0942 ASL PAGE
969E- 0E 1C 9A 0943 ASL PAGE
96A1- 0E 1C 9A 0944 ASL PAGE
96A4- 0E 1C 9A 0945 ASL PAGE
96A7- 29 0F 0946 AND #$F
96A9- 0D 1C 9A 0947 DRA PAGE
96AC- 8D 1C 9A 0948 STA PAGE
96AF- 4C 93 96 0949 JMP RC1
9650 :IT ISN'T, EXIT
96B2- 20 7B 92 0951 RC2 JSR LAST
96B5- 4C D5 93 0952 JMP COMCOM
9653 :NUMTEST - SET N=1 IF NO NUMBER, ELSE N=0
96B8- C9 30 0954 NUMTEST CMP #$30
96BA- 30 07 0955 BMI NUMT1
96BC- C9 3A 0956 CMP #$3A
96BE- 10 03 0957 BPL NUMT1
96C0- A0 00 0958 LDY #0
96C2- 60 0959 RTS
96C3- A0 FF 0960 NUMT1 LDY #$FF
96C5- 60 0961 RTS
9662 :KCOM - WAIT FOR ANY KEY
96C6- 20 AE 12 0963 KCOM JSR RECHA
96C9- 4C D5 93 0964 JMP COMCOM
9665 :ICOM - SET PAGE SIZE
96CC- 20 D8 92 0966 ICOM JSR DECIM
96CF- D0 06 0967 BNE ICOM2
9668 :FIRST OPERAND 0. SWITCH PAGED OFF AND EXIT
96D1- 8D 1B 9A 0969 STA PAGED
96D4- 4C D5 93 0970 JMP COMCOM
96D7- A2 00 0971 ICOM2 :FIRST OPERAND IS SIZE IN INCHES, MULTIPLY BY 48
96D9- 8E 25 9A 0972 LDX #0
9673 STX PSIZE+1
9674 :FIRST BY 3
96DC- 8D 24 9A 0975 STA PSIZE
96DF- 0A 0976 ASL A
96E0- 18 0977 CLC
96E1- 6D 24 9A 0978 ADC PSIZE
96E4- 8D 24 9A 0979 STA PSIZE
9680 :THEN 4 TIMES BY 2 (RESULT: BY 16)
96E7- A2 04 0981 LDX #$4
96E9- 0E 24 9A 0982 ICOM3 ASL PSIZE
96EC- 2E 25 9A 0983 ROL PSIZE+1
96EF- CA 0984 DEX
96F0- D0 F7 0985 BNE ICOM3
9686 :TOTAL RESULT: BY 48
9687 :NOW CHECK IF SECOND OPERAND FOLLOWS
96F2- 20 6B 90 0988 JSR NEXT

```

```

96F5- C9 2C 0989 CMP #'.
96F7- D0 09 0990 BNE ICOM4
96F9- 20 D8 92 0991 :IT DOES. JUST READ IT IN
96FC- 8D 26 9A 0992 JSR DECIM
96FF- 4C D5 93 0993 STA BSIZE
9702- 20 7B 92 0994 JMP COMCOM
9705- 4C D5 93 0995 :EXIT FROM ICOM
9708- 20 D8 92 0996 ICOM4 JSR LAST
9708- 29 03 0997 JMP COMCOM
970D- BD 2E 9A 0998 :CCOM - CHANGE LINEFEEDS AFTER EACH PASS
9710- D0 08 1000 JSR DECIM
9712- 8D 28 01 1001 AND #X11
9715- A2 33 1002 STA TEMP
9717- 4C 51 94 1003 BNE CCOM1
971A- 20 6B 90 1004 :FIRST OPERAND O. ERROR 33
971D- C9 2C 1005 STA FILENR
971F- D0 F1 1006 LDX #$33
9721- 20 D8 92 1007 JMP ERROR
9724- 29 01 1008 JSR NEXT
9726- F0 02 1009 :IF NO SECOND OPERAND. ERROR 33
9728- A9 0A 1010 CMP #'.
972A- AE 2E 9A 1011 BNE ERR33
972D- 9D 26 9A 1012 JSR DECIM
9730- 4C D5 93 1013 :BIT 0 DETERMINES LINEFEED OR NOT
9733- A9 00 1014 AND #X1
9735- BD 32 9A 1015 BEQ CCOM2
9738- AD 31 9A 1016 LDA #$A
973B- CD 30 9A 1017 LDX TEMP
973E- 90 0C 1018 STA CTRL-1.X
9740- ED 30 9A 1019 JMP COMCOM
9743- BD 31 9A 1020 :DIVIDE - QUOT:=INT(SPACES/WORDS)
9746- EE 32 9A 1021 : QUOT:=0:
9749- 4C 38 97 1022 : WHILE SPACES >= WORDS DO
974C- 60 1023 : QUOT:=QUOT+1: SPACES:=SPACES-WORDS:
974D- A2 00 1024 DIVIDE LDA #$0
974F- 8E 31 9A 1025 DIV1 STA QUOT
9752- CA 1026 LDA SPACES
9753- 8E 30 9A 1027 CMP WORDS
9756- AD 18 9A 1028 BCC ENDDIV
9759- 48 1029 SBC WORDS
975A- AD 1F 9A 1030 STA SPACES
975D- BD 18 9A 1031 INC QUOT
9760- AC 20 9A 1032 JMP DIV1
9763- AE 18 9A 1033 ENDDIV RTS
9766- BD B4 98 1034 ;FILLOUT - SPREAD WORDS OVER THE LINE
9769- C9 20 1035 FILLOUT :FIRST COUNT SPACES AND WORDS-1
976B- D0 0D 1036 LDX #$0
976D- EE 18 9A 1037 STX SPACES
9770- EE 31 9A 1038 DEX
9773- CC 18 9A 1039 STX WORDS
9776- D0 EB 1040 :SAVE HCNT. USE HCNT AS COUNTER
9778- F0 2C 1041 LDA HCNT
977A- EE 30 9A 1042 PHA
977D- EE 18 9A 1043 LDA LBORD
9780- CC 18 9A 1044 STA HCNT
9783- F0 21 1045 LDY RBORD
9785- AE 18 9A 1046 :FIND FIRST WORD. COUNT SPACES
9788- BD B4 98 1047 LDX HCNT
978B- CC 18 9A 1048 LDA BUF1,X
978D- D0 EE 1049 CMP '#'
978F- EE 31 9A 1050 BNE FILL2
9770- EE 18 9A 1051 INC HCNT
9773- CC 18 9A 1052 INC SPACES
9776- D0 EB 1053 CPY HCNT
9778- F0 2C 1054 BNE FILL1
977A- EE 30 9A 1055 BEQ ENDP1
977D- EE 18 9A 1056 :WORD FOUND. COUNT AND SKIP IT
9780- CC 18 9A 1057 FILL2 INC WORDS
9783- F0 21 1058 INC HCNT
9785- AE 18 9A 1059 CPY HCNT
9788- BD B4 98 1060 BEQ ENDP1
978B- CC 18 9A 1061 LDX HCNT
978D- D0 EE 1062 LDA BUF1,X
978F- EE 31 9A 1063 CMP '#'
9770- EE 18 9A 1064 BNE FILL3
9773- CC 18 9A 1065 :SPACES FOUND. COUNT AND SKIP THEM
9776- D0 EB 1066 INC SPACES

```

# DE6502 KENNER

9792- EE 18 9A	1066	INC HCNT	RCW TSWTR QWR:	2811	00	00	-0007
9795- CC 18 9A	1067	CPY HCNT	IN A01	0012	A0	00	-0008
9798- FO 0C	1068	BEQ ENDP1	T000 RT0	0013	00	00	-0009
979A- AE 18 9A	1069	LDX HCNT	X, 0000 001	1000	00	00	-0009
979D- BD 84 98	1070	LDA BUF1,X	0000 0000	0014	00	00	-0009
97A0- C9 20	1071	CMP #'	0000 0000	0015	00	00	-0009
97A2- FO EB	1072	BEQ FILL4	0000 0000	0016	00	00	-0009
97A4- DO D4	1073	BNE FILL2	0000 0000	0017	00	00	-0009
	1074	:END OF LINE REACHED, COUNTING DONE	0000 0000	0018	00	00	-0009
97A6- AD 30 9A	1075	ENDP1	LDA WORDS	0019	00	00	-0009
97A9- 30 6D	1076	BMI ENDP2	0000 0000	0020	00	00	-0009
	1077	:ENOUGH WORDS TO FILL OUT ?	0000 0000	0021	00	00	-0009
97AB- CD 2A 9A	1078	CMP FILL	0000 0000	0022	00	00	-0009
97AE- 30 68	1079	BMI ENDP2	0000 0000	0023	00	00	-0009
97B0- 20 33 97	1080	JSR DIVIDE	0000 0000	0024	01	00	-0009
	1081	:YES, DIVIDE AND START FILLING OUT	0000 0000	0025	01	00	-0009
97B3- AD 20 9A	1082	LDA RBORD	0000 0000	0026	00	00	-0009
97B6- 8D 18 9A	1083	STA HCNT	0000 0000	0027	00	00	-0009
97B9- A8	1084	TAY	0000 0000	0028	00	00	-0009
	1085	:MOVE FROM BUF,X TO BUF,Y (X FROM HCNT)	0000 0000	0029	01	00	-0009
	1086	:FIND LAST WORD	0000 0000	0030	01	00	-0009
97BA- AE 18 9A	1087	FILL5	LDX HCNT	0031	02	00	-0009
97BD- EC 1F 9A	1088	CPX LBORD	0000 0000	0032	00	00	-0009
97C0- FO 56	1089	BEQ ENDP2	0000 0000	0033	00	00	-0009
97C2- CE 18 9A	1090	DEC HCNT	0000 0000	0034	00	00	-0009
97C5- CA	1091	DEX	0000 0000	0035	00	00	-0009
97C6- BD 84 98	1092	LDA BUF1,X	0000 0000	0036	00	00	-0009
97C9- C9 20	1093	CMP #'	0000 0000	0037	00	00	-0009
97CB- FO ED	1094	BEQ FILL5	0000 0000	0038	00	00	-0009
97CD- 88	1095	DEY	0000 0000	0039	00	00	-0009
	1096	:WORD FOUND, MOVE IT	0000 0000	0040	00	00	-0009
97CE- BD 84 98	1097	FILL6	LDA BUF1,X	0041	00	00	-0009
97D1- C9 20	1098	CMP #'	0000 0000	0042	00	00	-0009
97D3- FO 19	1099	BEQ FILL7	0000 0000	0043	00	00	-0009
97D5- 99 84 98	1100	STA BUF1,Y	0000 0000	0044	00	00	-0009
97D8- BD 08 99	1101	LDA BUF2,X	0000 0000	0045	00	00	-0009
97DB- 99 08 99	1102	STA BUF2,Y	0000 0000	0046	00	00	-0009
97DE- BD 8C 99	1103	LDA BUF3,X	0000 0000	0047	00	00	-0009
97E1- 99 BC 99	1104	STA BUF3,Y	0000 0000	0048	00	00	-0009
97E4- EC 1F 9A	1105	CPX LBORD	0000 0000	0049	00	00	-0009
97E7- FO 2F	1106	BEQ ENDP2	0000 0000	0050	00	00	-0009
97E9- 88	1107	DEY	0000 0000	0051	00	00	-0009
97EA- CA	1108	DEX	0000 0000	0052	00	00	-0009
97EB- 4C CE 97	1109	JMP FILL6	0000 0000	0053	00	00	-0009
	1110	:SPACES FOUND, INSERT RIGHT NUMBER AND SKIP THE REST	0000 0000	0054	00	00	-0009
97EE- 8E 18 9A	1111	FILL7	STX HCNT	0055	00	00	-0009
	1112	:COMPUTE RIGHT NUMBER	0000 0000	0056	00	00	-0100
97F1- AD 31 9A	1113	LDA SPACES	0000 0000	0057	00	00	-0100
97F4- FO 05	1114	BEQ FILL8	0000 0000	0058	00	00	-0100
97F6- A9 01	1115	LDA #1	0000 0000	0059	00	00	-0100
97F8- CE 31 9A	1116	DEC SPACES	0000 0000	0060	00	00	-0100
97FB- 18	1117	FILL8	CLC	0061	00	00	-0100
97FC- 6D 32 9A	1118	ADC QUOT	0000 0000	0062	00	00	-0100
97FF- AA	1119	TAX	0000 0000	0063	00	00	-0100
	1120	:INSERT SPACES	0000 0000	0064	00	00	-0100
9800- A9 20	1121	LDA #'	0000 0000	0065	00	00	-0100
9802- 99 84 98	1122	FILL9	STA BUF1,Y	0066	00	00	-0100
9805- 99 08 99	1123	STA BUF2,Y	0000 0000	0067	00	00	-0100
9808- 99 BC 99	1124	STA BUF3,Y	0000 0000	0068	00	00	-0100
980B- 88	1125	DEY	0000 0000	0069	00	00	-0100
980C- CA	1126	DEX	0000 0000	0070	00	00	-0100
980D- DO F3	1127	BNE FILL9	0000 0000	0071	00	00	-0100
980F- CE 18 9A	1128	DEC HCNT	0000 0000	0072	00	00	-0100
9812- AE 18 9A	1129	LDX HCNT	0000 0000	0073	00	00	-0100
9815- 4C CE 97	1130	JMP FILL6	0000 0000	0074	00	00	-0100
	1131	:END OF FILLING OUT, RETSTORE HCNT	0000 0000	0075	00	00	-0100
9818- 68	1132	ENDP2	PLA	0076	00	00	-0100
9819- 8D 18 9A	1133	STA HCNT	0000 0000	0077	00	00	-0100
981C- 60	1134	RTS	0000 0000	0078	00	00	-0100
	1135	:QCOM - SET MINIMAL NUMBER OF WORDS TO FILL OUT	0000 0000	0079	00	00	-0100
981D- 20 D8 92	1136	QCOM	JSR DECIM	0080	00	00	-0100
9820- 8D 2A 9A	1137	STA FILL	0000 0000	0081	00	00	-0100
9823- 4C D5 93	1138	JMP COMCOM	0000 0000	0082	00	00	-0100
	1139	:SWEEP - REMOVE SUPERFLUOUS SPACES	0000 0000	0083	00	00	-0100
	1140	:QUOT IS USED TO HOLD PREVIOUS CHARACTER	0000 0000	0084	00	00	-0100
9826- AE 1F 9A	1141	SWEEP	LDX LBORD	0085	00	00	-0100
9829- AC 1F 9A	1142		LDY LBORD	0086	00	00	-0100

# DE 6502 KENNER

```

1143 :FIND FIRST WORD
982C- A9 20 1144 LDA #'          8801 AE 81 BB -001E
982E- BD 32 9A 1145 STA QUOT      1861 AP 61 CC -007E
9831- BD 34 98 1146 SW1   LDA BUF1,X 8802 00 00 -007E
9834- C9 20 1147 CMP #'          8802 8E 81 BB -007E
9836- FO 20 1148 BEQ SW2       8802 8E 80 BB -007E
1149 :WORD: MOVE IT
9838- 99 84 98 1150 SW12      8802 8E 81 BB -007E
983B- BD 32 9A 1151 STA QUOT      8802 00 00 -007E
983E- BD 08 99 1152 LDA BUF2,X  8802 8E 81 BB -007E
9841- 99 08 99 1153 STA BUF2,Y  8802 8E 80 BB -007E
9844- BD 8C 99 1154 LDA BUF3,X  8802 8E 81 BB -007E
9847- 99 8C 99 1155 STA BUF3,Y  8802 8E 80 BB -007E
984A- C8 1156 INY             8802 8E 81 BB -007E
984B- CC 20 9A 1157 CPY RBORD     8802 8E 81 BB -007E
984E- FO 1D 1158 BEQ ENDSW     8802 8E 81 BB -007E
9850- E8 1159 INX             8802 8E 81 BB -007E
9851- EC 20 9A 1160 CPX RBORD     8802 8E 81 BB -007E
9854- D0 DB 1161 BNE SW1       8802 8E 81 BB -007E
9856- FO 15 1162 BEQ ENDSW     8802 8E 81 BB -007E
1163 :SPACE: MOVE IF PREVIOUS WASN'T. ELSE REMOVE
9858- AD 32 9A 1164 SW2       8802 8E 81 BB -007E
985B- C9 20 1165 LDA QUOT      8802 8E 81 BB -007E
985D- D0 08 1166 CMP #'          8802 8E 81 BB -007E
985F- E8 1167 BNE SW1       8802 8E 81 BB -007E
9860- EC 20 9A 1168 INX             8802 8E 81 BB -007E
9863- D0 CC 1169 CPX RBORD     8802 8E 81 BB -007E
9865- FO 06 1170 BNE ENDSW     8802 8E 81 BB -007E
1171 :MOVE SPACE
9867- BD 84 98 1172 SW11      8802 8E 81 BB -007E
986A- 4C 38 98 1173 LDA BUF1,X  8802 8E 81 BB -007E
1174 JMP SW12       8802 8E 81 BB -007E
986D- A9 20 1175 ENDSW      8802 8E 81 BB -007E
986F- CC 20 9A 1176 LDA #'          8802 8E 81 BB -007E
9872- FO OF 1177 CPY RBORD     8802 8E 81 BB -007E
9874- 99 84 98 1178 BEQ ENDSW2    8802 8E 81 BB -007E
9877- 99 08 99 1179 STA BUF1,Y  8802 8E 80 BB -007E
987A- 99 8C 99 1180 STA BUF2,Y  8802 8E 80 BB -007E
987D- C8 1181 INY             8802 8E 80 BB -007E
987E- CC 20 9A 1182 CPY RBORD     8802 8E 80 BB -007E
9881- D0 EA 1183 BNE ENDSW     8802 8E 80 BB -007E
9883- 60 1184 ENDSW2      8802 8E 80 BB -007E
1185 :VARIABLES:
9884- 1186 BUF1   . DS BUFLen  :FIRST BUFFER
9908- 1187 BUF2   . DS BUFLen  :SECOND BUFFER
998C- 1188 BUF3   . DS BUFLen  :UNDERLINE BUFFER
9A10- 1189 USE2   . DS 1        :FLAG : COPY TO BUF2
9A11- 1190 USED2  . DS 1        :FLAG : BUF2 USED
9A12- 1191 UNDERL . DS 1        :FLAG : UNDERLINE IN BUF3
9A13- 1192 UNDERLD . DS 1        :FLAG : BUF3 USED
9A14- 1193 HSIZE   . DS 1        :HORIZONTAL SIZE
9A15- 1194 VSIZE   . DS 1        :VERTICAL SIZE
9A16- 1195 HM      . DS 1        :HORIZONTAL MODE
9A17- 1196 VM      . DS 1        :VERTICAL MODE
9A18- 1197 HCNT   . DS 1        :HORIZONTAL POSITION
9A19- 1198 VCNT   . DS 2        :SAME, BUT VERTICAL
9A1B- 1199 PAGED   . DS 1        :FLAG : PAGED MODE
9A1C- 1200 PAGE   . DS 1        :PAGE COUNTER
9A1D- 1201 PRINT   . DS 1        :FLAG : OUTPUT TO PRINTER
9A1E- 1202 SCREEN  . DS 1        :FLAG : OUTPUT TO SCREEN
9A1F- 1203 LBORD   . DS 1        :LEFT BORDER
9A20- 1204 RBORD   . DS 1        :RIGHT BORDER
9A21- 1205 PTAB    . DS 1        :PARAGRAPH TABULATION
9A22- 1206 TWIDTH  . DS 1        :TERMINAL WIDTH
9A23- 1207 OFFSET  . DS 1        :NUMBER ADDED TO DATA
9A24- 1208 PSIZE   . DS 2        :SIZE OF PAGE IN 1/48 INCH
9A26- 1209 BSIZE   . DS 1        :SIZE OF SPACE BETWEEN PAGES
9A27- 1210 CTRL    . DS 3        :PRINTER CONTROL CODES
9A2A- 1211 FILL    . DS 1        :FLAG : FILLING OUT LINES
9A2B- 1212 LCHAR   . DS 1        :LAST CHAR SENT TO BUF1
9A2C- 1213 CCHAR   . DS 1        :CHAR FETCHED BY NEXT OR END OF LINE
9A2D- 1214 NUMB   . DS 1        :BUFFER FOR DECIMAL INPUT
9A2E- 1215 TEMP    . DS 2        :TEMPORARY STORAGE
9A30- 1216 WORDS   . DS 1        :NUMBER OF WORDS-1 IN BUF1
9A31- 1217 SPACES  . DS 1        :NUMBER OF SPACES IN BUF1
9A32- 1218 QUOT    . DS 1        :SPACES/WORDS
1219 :PRINTER DEPENDENT PART

```

# DE 6502 KENNER

1220 :OKI MICROLINE 80 FORMATS  
1221 :1ST : HORIZONTAL SIZES  
1222 :FORMAT : .BY X X X X Y , WHERE XS ARE SENT TO THE PRINTER  
1223 :AND Y IS NUMBER OF CHARACTERS/LINE  
1224 :EIGHT MODES MAY BE USED (0...7)  
9A33- 1B 41 1D 1225 HOR .BY \$1B 'A' \$1D 0 132 :COMPRESSED LONG LINE  
9A36- 00 84 .BY \$1B 'A' \$1E 0 80 :NORMAL  
9A3B- 00 50 .BY \$1B 'A' \$1F 0 40 :DOUBLE WIDTH  
9A40- 00 28 .BY \$1B 'B' \$1D 0 105 :SAME BUT SHORT LINE  
9A42- 1B 42 1D 1228 .BY \$1B 'B' \$1E 0 64  
9A45- 00 69 .BY \$1B 'B' \$1F 0 32  
9A47- 1B 42 1E 1229 .BY \$1B 'B' \$1E 0 64  
9A4A- 00 40 .BY \$1B 'B' \$1F 0 32  
9A4C- 1B 42 1F 1230 .BY \$1B 'B' \$1F 0 32  
9A4F- 00 20 .BY \$1B 'B' \$1F 0 32  
9A51- 1B 42 1F 1231 .BY \$1B 'B' \$1F 0 32  
9A54- 00 20 .BY \$1B 'B' \$1F 0 32  
9A56- 1B 42 1F 1232 .BY \$1B 'B' \$1F 0 32  
9A59- 00 20 .BY \$1B 'B' \$1F 0 32  
1233 :2ND : VERTICAL  
1234 :SAME FORMAT, BUT Y IS HEIGHT OF LINE TIMES 1/48 INCH  
9A5B- 1B 36 00 1235 VER .BY \$1B '6' 0 0 8 ;6 LINES/INCH  
9A5E- 00 08 .BY \$1B '8' 0 0 6 ;8 LINES/INCH  
9A60- 1B 38 00 1236 .BY \$1B '8' 0 0 6  
9A63- 00 06 .BY \$1B '8' 0 0 6  
9A65- 1B 38 00 1237 .BY \$1B '8' 0 0 6  
9A68- 00 06 .BY \$1B '8' 0 0 6  
9A6A- 1B 38 00 1238 .BY \$1B '8' 0 0 6  
9A6D- 00 06 .BY \$1B '8' 0 0 6  
9A6F- 1B 38 00 1239 .BY \$1B '8' 0 0 6  
9A72- 00 06 .BY \$1B '8' 0 0 6  
9A74- 1B 38 00 1240 .BY \$1B '8' 0 0 6  
9A77- 00 06 .BY \$1B '8' 0 0 6  
9A79- 1B 38 00 1241 .BY \$1B '8' 0 0 6  
9A7C- 00 06 .BY \$1B '8' 0 0 6  
9A7E- 1B 38 00 1242 .BY \$1B '8' 0 0 6  
9A81- 00 06 .BY \$1B '8' 0 0 6  
1243 :WHEN CHANGING TO ANOTHER PRINTER, DON'T FORGET TO ADJUST BUflen !!!!  
1244 :WRITTEN BY J. BUURMAN  
1245 : v.d.FUYCKSTRAAT 136  
1246 : 3232 AS BRIELLE (01810-5153)  
1247 :LAST MODIFICATION 19-8-1984  
1248 .EN

## AANGEBODEN:

KIM-1 met 64K Ram (Elektuurkaart). Hazeltine 1400 Video-terminal. Eventueel voeding voor KIM-1 (5V. + en - 12V + en - , 25V). ASCII keyboard in behuizing. KIM-1 en terminal in een kooi, met beveken. Robert Korvind, W. Pijperstraat 6. 2551 Den Haag. Tel.: 070 - 680146 (tussen 19.00 en 21.00 uur).

## KLEINTJE FORTH. REACTIES.

Ronald Vinks oplossing voor het listen van screens 0 t/m 15 in editie 33 van DE 6502 KENNER p. 47 vroeg om een oplossing voor het variabel listen van screens. oo te nemen in het vocabulary. Wout van Dinther, Molenrijnselaan 27, 5262 TN Vught, heeft aan de ooroeo gevolg gegeven. Hier is zijn oplossing.

: QLIST 1+ SWAP DO KEY I LIST LOOP :

Met als definitie omschrijving ( n1 n2 -- ), waarbij n1 het eerste screen en n2 het laatste moet zijn. De KEY erin zorgt ervoor dat hij op ingave van het toetsenbord staat te wachten (b.v.: soatje), waarna het volgende screen oelist wordt. Wout denkt inmiddels na over een nog mooiere oplossing. De redactie houdt zich aanbevolen voor meer FORTH-screens.

Fridus Jonkman oaf in Kleintje Fortn voor het Recursie-probleem. Er zijn meer oplossingen denkbaar. Een ervan ontvinde we van Gert van Orobrouk. Hoofdlanden 20. 9801 LB Zuidhorn (Gr). Hi schreef ons het voldende:  
Recursie kan op zeer eenvoudige wijze gedaan worden d.m.v.  
de definitie:  
: RECUSIE ." 6502 " R) DROP { zorg ervoor dat de return- )  
stack niet volloopt }

## MYSELF :

Een ander voorbeeld:  
: RECFAC { bereken n! recursief )  
DUP 1 ) IF DUP 1- MYSELF \* ENDIF :

Aanroep : n RECFAC .

## 65816 MICROPROCESSOR

Wie mocht denken dat de 65XX-familie het moeilijk heeft in de concurrentiestrijd, komt van een koude kermis terug. De 65XX-familie baart nog steeds nieuwe chips. De 65816 microprocessor is zo'n nieuwe borelino, intern uitgerust met een 16 bits accumulator, 16 bits indexregister, een 24 bits (!) programcounter en een 16 bits stackpointer. Naar analogie van de 6809 een 'direct data register', waardoor zero page in elk deel v/h geheugen te plaatsen. Met behoud van compatibiliteit met de 6502 als 65C02 te gebruiken, en de mogelijkheid om 16 mega(!)byte te adresseren. En nog heel veel meer van dat leuks. We hopen hier binnenkort nog uitgebreider over te kunnen berichten. Bedankt inzenders!

# DE 6502 KENNER

## UITNODIGING BIJEEENKOMST

Datum : zaterdag 24 november 1984  
Lokatie : R.K. HTS "Rijswijk"  
Lange Kleiweg 4 te Rijswijk. Tel.: 070 - 907839

### Reisroute :

- per auto - komende uit de richting Utrecht  
Volg autoweg E8 Utrecht-Den Haag. Knooppunt Leidschendam via de hoge rijbaan linksaf richting Delft/Rotterdam. Aan het eind daarvan rechtsaf richting Den Haag. Rechts aanhouden. Onder verkeersbord Yoenburg door en rechtdoor met trambaan mee over Hoornbrug. Dan scherpe draai naar rechts en onder brug door. Delft/Rotterdam en Wateringen aanhouden. Bij stoellichten linksaf en Sir Winston Churchilllaan uitrijden tot laatste stoellichten voor spoorovergang. Links van U gebouw GAK en ABN bank. hier linksaf Huis te Landelaan op. uitrijden tot eind en rechtsaf. Uitrijden tot schoolplein.
  - komende uit de richting Amsterdam  
Volg A4 A'dam-Rotterdam. Knooppunt Leidschendam passeren. E8/E10 aanhouden tot eind. rechtsaf richting Den Haag. Rechts aanhouden. Onder verkeersbord Yoenburg door met trambaan mee over Hoornbrug. Verder als boven.
  - komende uit de richting Rotterdam  
Volg E10 R'dam-Den Haag tot vlak voor Hoornbrug. Onder verkeersbord Yoenburg door en rechtdoor met trambaan mee over Hoornbrug. Verder als boven.
- per trein - Station Rijswijk uitdando richting winkelcentrum De Boodaard. Bij uitgang linksaf. Spoorovergang Winston Churchilllaan over tot stoelicht. rechtsaf Huis te Landelaan. Verder als boven.

Lunchpakket zelf meenemen.  
Consumpties tegen betaling.

### Programma:

- 10.00 Opening Ledenvergadering 1984.
- 10.15 - Notulen jaarvergadering 1983  
Financieel Jaarverslag 1983  
Begroting 1984  
Verkiezing kascontrolecommissie  
Verkiezing bestuursleden:
  - aftredend en niet herkiesbaar: Ruud Uhoff  
voordracht van het bestuur : Gert Klein
  - aftredend en niet herkiesbaar: Mw. M.F. Uhoff-Burghoorn
  - verkiezing 1 bestuurslid (kandidaten kunnen schriftelijk worden aangemeld bij de sekretaris of mondeling voor de aanvang van de vergadering).
- Mededeling over c.o. uitreikings publikatie-aanmoedigingsremie(s)  
- Rondvraag en sluiting.
- 12.00 Lunch
- 13.00 Lezing Nico de Vries over universeel programmeren. 65C02. enz.  
14.00 INFORMATIE GEDEELTE  
BRENG OOK UW SYSTEEM MEE ALS DAT ENIGSZINS LUKT !!  
**MARKT.** Op eigen tafel(s) te regelen.
- 17.00 Sluiting.

# DE 6502 KENNER

```

100 // ****
110 // *
120 // * PROGRAMMA STAAFDIAGRAM *
130 // * voor microline 80 printer *
140 // *
150 // ****

Staafdiaogram afdrukken met een microline 80 printer

Dit programma is geschreven om de resultaten van proefwerken van
verschillende klassen met elkaar te vergelijken. Maar er zijn
meer mogelijkheden te bedenken.
Het programma is een voorbeeld om het gebruik van globale en lo-
cale variabelen te demonstreren. (PROC "voer het aantal in").
Van de andere kant is het in COMAL*KGN niet modelijk om karakters
met een nummer groter dan 127 in een print statement te gebruiken.
Helaas vormen de karakters boven de 127 juist de grafische set van
de microline 80 printer. De oplossing is een in COMAL beschreven
print-routine die het gewenste karakter direct in de printerlaat
"doorkt". Zie de procedure "print" onderaan. Om het programma aan-
pasbaar te maken heb ik de soorten bovenaan in het programma ge-
declareerd. Het zijn de variabelen:
STATUS      de inoutpoort van de PIA waarin de printer status
staat (P7=online/offline : P6=paper 1=tot of daer :
PS=BUSY : P4=not fault : P0-P3 =1 )
PIAPB       de outputpoort naar de printer (parallel)
ONLINE      de waarde die de statuspoort heeft als de printer
"busy" is.

Er kan in het programma gekozen worden : de resultaten afdrukken
zoals ze zijn, of de resultaten in procenten afdrukken.
In het eerste geval worden er details voor de regels afdrukt,
in het tweede geval geven streeptjes een verdeling aan.
Een voorbeeld is bijgevoegd. De getallen onderaan geven het num-
mer van de vraag.

100 // ****
101 // *
102 // * PROGRAMMA STAAFDIAGRAM *
103 // * voor microline 80 printer *
104 // * frans bakx *
105 // * huissteden 1112 *
106 // * 6605 hd wijchen *
107 // *
108 // * COMAL*KGN *
109 // ****

110 CLEAR
111 STATUS:=6176
112 PIAPB:=6178
113 ONLINE:=95
114 BUSY:=127
115 VRAGEN$:=" vraag "
116 LEERLINGEN$:=" leerlingen "
117 FOUTEN$:= " fouten "
118 PRINT CHR$(10)
119 PRINT "FOUTEN STAAFDIAGRAM"
120 PRINT
121 //
122 EXEC: "voer het aantal in".VRAGEN$.NVRAGEN%
123 DIM FOUTEN%(1..NVRAGEN%)
124 DIM KARAKTER%(NVRAGEN%)
125 EXEC: "voer het aantal in".LEERLINGEN$.NLEERLING%
126 FOR VRAAG:=1 TO NVRAGEN%
127 PRINT "Voor vraag "VRAAG" ":
128 REPEAT
129 EXEC: "voer het aantal in".FOUTEN%.NFOUT%
130 UNTIL NFOUT%<=NLEERLING%
131 FOUTEN%(0..VRAAG) :=NFOUT%
132 FOUTEN%(1..VRAAG) :=INT(100*NFOUT%/NLEERLING%+0.5)
133 ENDFOR
134 REPEAT

```

```

135 PRINT "0 = absoluut afdrukken"
136 PRINT "1 = relatief afdrukken"
137 PRINT "2 = einde"
138 PRINT
139 PRINT " wat wilt u ? "
140 GET N$
141 PRINT
142 KEUZE:=INT(VAL(N$))
143 IF KEUZE=0 OR KEUZE=1 THEN
144   EXEC: "diagram afdrukken"
145 ENDIF
146 UNTIL KEUZE=2
147 END.
148 //
149 //
150 //
151 PROC "diagram afdrukken"
152 PRINT "zet de printer aan a.u.b."
153 // wacht totdat de printer aan staat
154 REPEAT
155   A:=PEEK(STATUS)
156 UNTIL A=ONLINE
157 PRINT
158 EXEC: "init printer"
159 IF KEUZE=0 THEN
160   NREGELS%:=NLEERLING%
161 ELSE
162   NREGELS%:=34
163 ENDIF
164 FOR REGEL:=NREGELS% TO 1 STEP -1
165   EXEC: "construeer reuel"
166   IF KEUZE=0 THEN
167     PRINT RIGHT$(" "+STR$(REGEL).2)":
168   ELSE
169     KARAKTER%(0):=240
170     EXEC: "print".KARAKTER%(0)
171     PRINT " ":
172   ENDIF
173   EXEC: "print reuel"
174 ENDFOR
175 EXEC: "print onderlijn"
176 PRINT CHR$(10)
177 // short line
178 PRINT CHR$(27)"B"
179 // wacht totdat de printer uit staat
180 REPEAT
181   A:=PEEK(STATUS)
182 UNTIL A>ONLINE AND A<BUSY
183 ENDPROC
184 //
185 PROC "voer het aantal in",PR$.N%
186 REPEAT
187   PRINT
188   PRINT "Het aantal"PR$"a.u.b. ? "
189   INPUT N$
190   N%:=INT(VAL(N$))
191 UNTIL VAL(N$)=N% AND N%>0
192 ENDPROC
193 //
194 PROC "init printer"
195   IF NLEERLING%<30 THEN
196     // short line
197     PRINT CHR$(27)"B":
198   ELSE
199     // long line
200     PRINT CHR$(27)"A":
201   ENDIF
202   // 6 regels per inch
203 PRINT CHR$(27)"6":
204   // groot afdrukken
205 PRINT CHR$(31):
206 ENDPROC
207 //

```

=====  
voor Junior met VDU-kaart  
\*\*\*\* PACMAN \*\*\*\*

=====  
De redactie hoopt binnenkort een serie publikaties uit te kunnen brengen van de inwendels in de edities reeds bekende Phons Bloemen. Het betreft hier een artikel in 3 delen, buiten de tekst zo'n 3 x 20 K source omvattend. Het bekende soel kan alleen gedraaid worden als men de beschikking heeft over Ramgeheugen tot \$A000. Het gehele programma, inclusief de coding voor beeldscherm, de zoektabellen en het doolhof beslaat het geheugen van \$8000 tot en met \$9FFF. Ook de karaktergenerator EPROM van Elektuur (ESS 523) dient aanwezig te zijn.

Dat er mensen zijn die het vele typewerk niet zien zitten, en omdat de laatste publicatie pas in april 1985 is te verwachten, heeft de redactie de routine op band beschikbaar gesteld.

Het staat in het bekende KIM en JUNIOR Hypertape formaat en kan met de monitor worden binnengehaald of met behulp van Micro-ADE.

De prijs van de cassette bedraagt:

FL. 15.00

Voor de allerongeduldigsten stellen we de gehele sourcelisting, inclusief de inleiding en schema voor joystick ook beschikbaar via de paperware service. LET OP: het is precies hetzelfde als gepubliceerd zal worden. Hef is dus niet nodig het te bestellen, maar het kan wel, als er teveel ongeduld is.

De prijs van de papers bedraagt, inclusief verpakking en verzendkosten:

FL. 45,00

Overschrijven op postgiro 841433 t.n.v. W.L. van Pelt te Krimpen a.d. IJssel, of op bankrekening 44.11.06.471 van de AMRO-bank te Krimpen a.d. IJssel (giro 3050). Een Eurocheque mag ook.

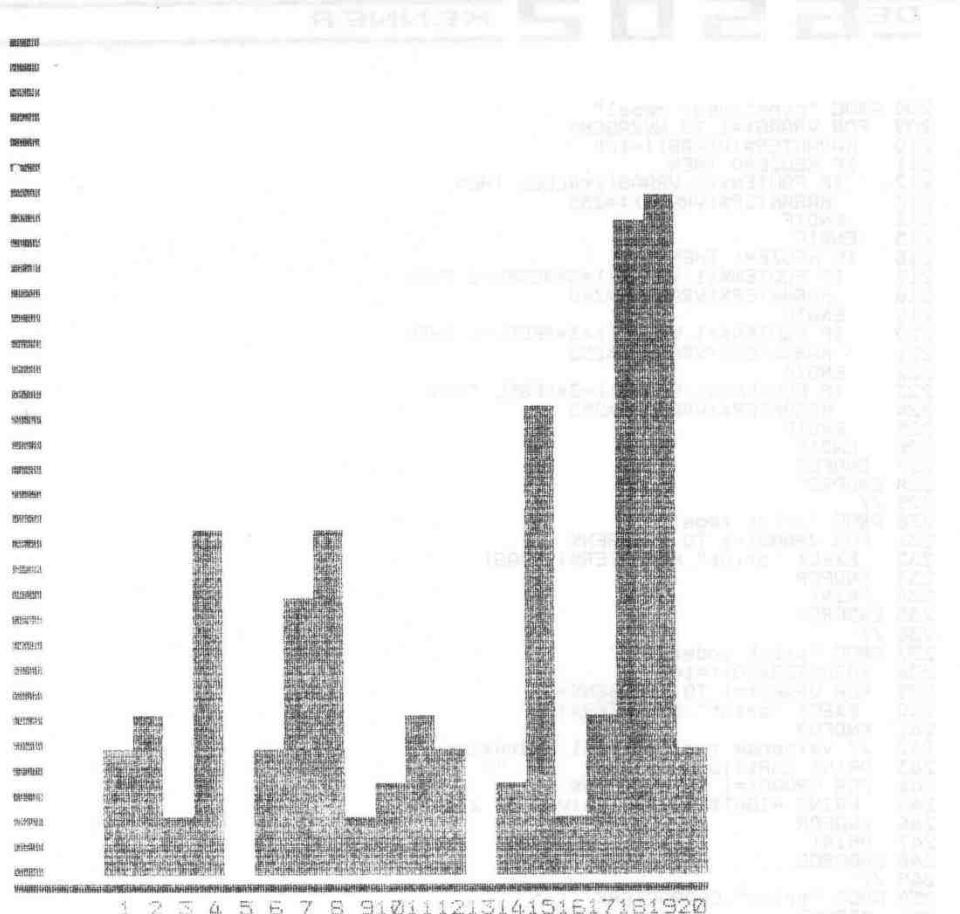
# DE 6502 KENNER

```

208 PROC "construeer regel"
209 FOR VRAAG:=1 TO NVRAGEN%
210 KARAKTER%(VRAAG) :=128
211 IF KEUZE=0 THEN
212 IF FOUTEN%(0,VRAAG)=REGEL THEN
213 KARAKTER%(VRAAG) :=255
214 ENDIF
215 ENDIF
216 IF KEUZE=1 THEN
217 IF FOUTEN%(1,VRAAG)=3*REGEL-2 THEN
218 KARAKTER%(VRAAG) :=240
219 ENDIF
220 IF FOUTEN%(1,VRAAG)=3*REGEL-1 THEN
221 KARAKTER%(VRAAG) :=252
222 ENDIF
223 IF FOUTEN%(1,VRAAG)=3*REGEL THEN
224 KARAKTER%(VRAAG) :=255
225 ENDIF
226 ENDIF
227 ENDFOR
228 ENDPROC
229 //
230 PROC "print regel"
231 FOR VRAAG:=1 TO NVRAGEN%
232 EXEC: "print".KARAKTER%(VRAAG)
233 ENDFOR
234 PRINT
235 ENDPROC
236 //
237 PROC "print onderlijn"
238 KARAKTER%(0) :=140
239 FOR VRAAG:=1 TO NVRAGEN%+3
240 EXEC: "print".KARAKTER%(0)
241 ENDFOR
242 // voldende reel normaal afdrukken
243 PRINT CHR$(10)CHR$(30) "
244 FOR VRAAG:=1 TO NVRAGEN%
245 PRINT RIGHT$(" "+STR$(VRAAG),2):
246 ENDFOR
247 PRINT
248 ENDPROC
249 //
250 PROC "print".CHARACTER%
251 REPEAT
252 A:=PEEK(STATUS)
253 UNTIL A>BUSY
254 POKE PIAPB,CHARACTER%
255 ENDPROC
256 //
)

```

Bin Number	Frequency
1	4
2	5
3	1
4	11
5	4
6	5
7	9
8	11
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	20
19	1
20	4



## PATCHES FOR SENIOR MONITOR      FATE 65XX ASSEMBLER V1.0 PAGE: 0001

```

0001 0000      .TIT 'PATCHES FOR SENIOR MONITOR'
0002 0000      .OPT GEN
0003 0000      .OPT PRI
0004 0000      .OPT SYM
0005 0000      :
0006 0000      :
0007 0000      : ****
0008 0000      : *   *
0009 0000      : * PATCHES ON SENIOR MONITOR V4.6 *
0010 0000      : *
0011 0000      : ****
0012 0000      :
0013 0000      :

-----  

DEFINITIONS  

0015 0000      : CHARACTER DEFINITIONS
0016 0000      :
0017 0000      :
0018 0000      LF      =$0A          : LINE FEED
0019 0000      :
0020 0000      :
0021 0000      : SENIOR MONITOR LOCATIONS
0022 0000      :
0023 0000      TRFLG    =$0531        : TRACE FLAG
0024 0000      TRMCNT   =$0545        : HISTORY COUNTER
0025 0000      VFLG     =$058C        : VERIFY FLAG
0026 0000      MAXLIN   =$058F        : MAX. LINES/PAGE FOR PRINTER
0027 0000      :

```

# DE6502 KENNER

PATCHES FOR SENIOR MONITOR FATE 65XX ASSEMBLER V1.0 PAGE: 0002

```

0028 0000 : SENIOR MONITOR ROUTINES
0029 0000
0030 0000
0031 0000 INTSET =$F6BA : INITIALISE INTERFACES
0032 0000 RST =$E237 : RESET ENTRY POINT SENIOR
0033 0000
0034 0000

```

## PATCHES

```

0036 0000 : NR. 1: ON ELEKTERMINAL MAXIMUM 16 LINES/SCREEN
0037 0000
0038 0000
0039 0000
0040 0000 *= $E83E
0041 E83E A918 LDA #$18 : REPLACE BY 18. 14 LINES ON SCREEN
0043 E840
0044 E840
0045 E840 : NR. 2: MAX 62 LINES/PAGE ON PRINTER.
0046 E840 : NEW FORM FEED STRING
0047 E840
0048 E840 *= $E26D
0049 E26D
0050 E26D A006 LDY #$06
0051 E26F B9CAFF INLS6 LDA INTAB.Y : SET MAX. LINES/PAGE
0052 E272 998F05 STA MAXLIN.Y : ON PRINTER AND FORM
0053 E275 88 DEY : FEED STRING
0054 E276 10F7 BPL INLS6
0055 E278 A900 LDA #$00
0056 E27A 8D4505 STA TRMCNT : CLEAR HISTORY COUNTER
0057 E27D 8D3105 STA TRFLG : CLEAR TRACE FLAG
0058 E280 A937 LDA #$37
0059 E282 8D8C05 STA VFLG : SET VERIFY FLAG
0060 E285
0061 E285 *= $$FFCA
0062 FFCA
0063 FFCA 3E INTAB .BYT $3E : MAX LINES/PAGE
0064 FFCB 0A .BYT LF, LF, LF, LF, LF, $00
0064 FFCC 0A
0064 FFCD 0A
0064 FFCE 0A
0064 FFCF 0A
0064 FFDO 00
0065 FFD1
0066 FFD1
0067 FFD1 : NR. 3: ON QUME DRIVES HEADS ARE ALWAYS LOADED. SO
0068 FFD1 : THE HEAD LOAD TIME CAN BE 0 MSEC.
0069 FFD1 : ON QUME DRIVES THE HEAD STEPPING TIME IS
0070 FFD1 : BETTER THEN 15 MSEC.
0071 FFD1
0072 FFD1 *= $$FC64
0073 FC64
0074 FC64 A900 LDA #0 : HEAD-TIME = 0 MSEC.
0075 FC66
0076 FC66 *= $$FDAF
0077 FDAF
0078 FDAF A903 LDA #3 : 15 MSEC. STEP TIME
0079 FDB1
0080 FDB1 .END

```

2000 16000 0,2V RECHERCHER X250 STAR ROTRONIC ADINER R05 SCHOTAR

## TIPS & TRICKS

2000 16000 0,2V RECHERCHER X250 STAR ROTRONIC ADINER R05 SCHOTAR  
2000 16000 0,2V RECHERCHER X250 STAR ROTRONIC ADINER R05 SCHOTAR  
2000 16000 0,2V RECHERCHER X250 STAR ROTRONIC ADINER R05 SCHOTAR

### OHIO DOS V3.X SPECIAL VOOR DE JUNIOR

Er zijn enkele commando's die beginnen met de escape-toets gevolgd door een ander karakter. ( b.v. ESC 1 = beeld wissen ) De OHIO-basic negeert de escape-toets. Door nu te zessen PRINT !(n) is dit opgelost. ( n is de decimale waarde van het karakter dat na de escape-toets komt) Dus PRINT!(49) is gelijk aan print CHR\$(27);CHR\$(49).

Bij gebruik van de CTRL 'X' heb je grote kans dat het systeem hangt. Om dit te voorkomen zet dan vast in BEXEC de volgende instructies:

POKE 9593,234:POKE9594,234  
Terugzetten in de oude stand gebeurt met:  
POKE 9593,20:POKE9594,24

Na het invoeren van onderstaande instructies in de BEXEC kan de printer aan en uitgeset worden met CTRL 'G' (lo wordt op ,09 gezet)  
POKE 9610,201:POKE9611,7:POKE9618,9

Voor diegene die DOS V3.X hebben en de editor willen gebruiken, hij zit op track 15. Type in DISK!"lo 15", run het en als het programma zich meldt met 'EDIT ENABLED' type dan het woord "HALLO" in, daarna CTRL 'H' gevolgd door CTRL 'P'. Nu is de editor geinitialiseerd.

Zo ook met de RUBOUT functie in DOS V3.X. Deze kun je gebruiken als men onderstaande instructies invoert:

POKE1394,127:POKE1419,127  
POKE2820,127:POKE1386,128

#### HIER VOLGEN NOG ENKELE ROUTINES IN DOS:

\$2D73 STRING OUT JSR \$2D73 gevolgd door een tekststring wordt uitgeprint op het ingestelde device. De tekststring moet eindigen op 00.

\$2761 UNLOAD HEAD Gemakkelijk als het systeem hangt en de kop er nog op staat. ( Sommige drives gaan dan ook niet open! )

\$267C 1 mSEC TELLER

\$2678 10 mSEC TELLER

#### BELANGRIJKE ADRESSEN:

\$2300 (8960D) HIMEM Als achter in het RAM-gebied wat ruimte gereserveerd moet worden (b.v. in BASIC), bevat het HI-adres het laatste beschikbare RAM-gebied.

\$2322 (8994D) OUTPUT IO DEVICE

\$2DC4 (1271ED) DIR TRACK Hierin staat op welke track de directory te vinden is.

\$2EA3 (9891D) TRACKACC Track-to-track acces tijd in msec en decimaal. (standaard 28)

\$2EBC Inhoud van \$31. Bij 2 Mc processors is dit \$62

Met dank aan W.v.Dinther

# DE 6502 KENNER

CHTAPE DE 6502 KENNER

PAGE 01

```

0010: 0200      CHTAPE ORG $0200
0020:
0030: ****
0040: * C H E C K T A P E *
0050: ****
0060:
0070: Phons Bloemen
0080: Vinkelalaan 200
0090: 5702 LX Helmond
0100:
0110: This program reads a tape and gives
0120: the ID (identificationnumber)
0130: the SA (startaddress)
0140: the EA (endaddress)
0150: the CHK (checknumber)
0160:
0170: So it serves everyone's not knowing what
0180: is on it.
0190: To leave the program press (RST).
0200: Parts of the program you'll find in TM.
0210:
0220: ZERO PAGE
0230: 23 00 PRPNT * $0023
0240: FA 00 POINT * $00FA
0250: SUBROUTINES FROM TM
0260: C2 0B RDBIT * $0BC2 READS 1 BIT FROM TAPE
0270: E8 0B BTWEEN * $0BE8 DISPLAYS THE BETWEEN CHAR
0280: F3 0B RD BYT * $0BF3 READS 1 HEX BYTE = 2 ASCII CHAR FROM TAPE
0290: 36 0C READCH * $0C36 READS AN ASCII CHAR FROM TAPE
0300: 4B 0C CHKSUM * $0C48 COMPUTES CHECKSUM OF RECEIVED DATA
0310: 5D 0C CHARVU * $0C5D OUTPUTS CHAR TO 7-SEGMENT DISPLAY
0320: 64 0C VU * $0C64
0330: SUBROUTINES FROM PM
0340: E8 11 CRLF * $11E8 PRINT (CR) & (LF)
0350: F3 11 PRSP * $11F3 PRINT SPACE
0360: 8F 12 PRBYT * $128F CONVERTS BYTE INTO TWO ASCII CHARS
0370: 34 13 PRCHA * $1334 TRANSMIT ASCII CHAR TO PRINTER
0380: BC 14 RESTTY * $148C
0390: ALLOCATIONS IN PIA-RAM
0400: 58 1A CTRL * $1A58
0410: 69 1A SY * $1A69 SYN COUNTER
0420: 68 1A CHAR * $1A6B CHARACTER FROM TAPE
0430: 6E 1A CHKL * $1A6E CHECKSUM LOW
0440: 6F 1A CHKH * $1A6F CHECKSUM HIGH
0450: 70 1A SA * $1A70 STARTADDRESS
0460: 78 1A GANG * $1A78 TEMP OF PBD-BITS
0470: 79 1A ID * $1A79
0480: IN / OUT DEFINITIONS
0490: 81 1A PADD * $1A81 DATA DIR. REG. OF PORT A
0500: 82 1A PBDD * $1A82 DATA REG. OF PORT B
0510: 83 1A PBDD * $1A83 DATA DIR. REG. OF PORT B
0520:
0530: MAIN PROGRAM
0540:
0550: 0200 A9 32 START LDAIM $32 INPUT RECORDER ON
0560: 0202 BD 82 1A STA PBD OUTPUT RECORDER OFF
0570: 0205 BD 78 1A STA GANG
0580: 0208 A9 7E LDAIM $7E PBO..PB7 INPUT
0590: 020A BD 83 1A STA PBDD
0600: 020D A9 7F LDAIM $7F PA0..PA6 OUTPUT
0610: 020F BD 81 1A STA PADD 7-SEGMENTS DISPLAY ON
0620: 0212 A9 00 LDAIM $00
0630: 0214 BD 6E 1A STA CHKL RESET CHK
0640: 0217 BD 6F 1A STA CHKH

```

# DE 6502 KENNER

CHTAPE DE 6502 KENNER

PAGE 02

0650:	021A	A9 FF	SYNC	LDAIM \$FF	RESET INCOMING	0000 10100
0660:	021C	8D 6B 1A		STA CHAR	CHARACTER	10000
0670:	021F	20 C2 0B	SYNCA	JSR RDBIT		10000
0680:	0222	6E 6B 1A		ROR CHAR	SHIFT IN CHAR	10000
0690:	0225	AD 6B 1A		LDA CHAR	CHAR = SYNC (\$16) ?	10000
0700:	0228	20 E8 0B		JSR BTWEN	DISPLAY BETWEENCHAR	10000
0710:	022B	C9 16		CMPIM \$16	SYNC?	10000
0720:	022D	D0 F0		BNE SYNC	GO ON UNTIL	10000
0730:	022F	A0 0A		LDYIM \$0A	10 SYNCs ON ROW	10000
0740:	0231	8C 69 1A		STY SY	SYNC COUNTER	10000
0750:	0234	20 36 0C	TENSYN	JSR READCH	READ CHARACTER	10000
0760:	0237	20 5D 0C		JSR CHARVU	SYNCCHARACTER	10000
0770:	023A	C9 16		CMPIM \$16	STILL SYNC ?	10000
0780:	023C	D0 DC		BNE SYNC	NO, AGAIN	10000
0790:	023E	CE 69 1A		DEC SY	10 SYNCs ALREADY ?	10000
0800:	0241	D0 F1		BNE TENSYN		10000
0810:	0243	20 36 0C	STAR	JSR READCH	READ CHARACTER	10000
0820:	0246	20 5D 0C		JSR CHARVU	SYNCCHARACTER	10000
0830:	0249	C9 2A		CMPIM '*' STAR ?		10000
0840:	024B	F0 06		BEQ STARA		10000
0850:	024D	C9 16		CMPIM \$16	SYNC?	10000
0860:	024F	F0 F2		BEQ STAR		10000
0870:	0251	D0 AD		BNE START	NOTHING? THEN AGAIN	10000
0880:	0253	20 5D 0C	STAR A	JSR CHARVU	GOT IT	10000
0890:	0256	20 F3 0B		JSR RDBYT	READ BYTE (ID)	10000
0900:	0259	8D 79 1A		STA ID	SAVE IT	10000
0910:	025C	20 F3 0B		JSR RDBYT	READ BYTE (SAL)	10000
0920:	025F	20 4B 0C		JSR CHKSUM	ADJUST CHK	10000
0930:	0262	85 FA		STAZ POINT		10000
0940:	0264	8D 70 1A		STA SA		10000
0950:	0267	20 F3 0B		JSR RDBYT	READ BYTE (SAH)	10000
0960:	026A	20 4B 0C		JSR CHKSUM	ADJUST	10000
0970:	026D	85 FB		STAZ POINT	+01	10000
0980:	026F	8D 71 1A		STA SA	+01	10000
0990:	0272	20 F3 0B	PNTUP	JSR RDBYT	READ BYTE (DATA)	10000
1000:	0275	30 29		BMI BADTAP	ILLEGAL ?	10000
1010:	0277	F0 0F		BEQ CHECK	ENDCHARACTER / ?	10000
1020:	0279	20 4B 0C		JSR CHKSUM	ADJUST CHK	10000
1030:	027C	E6 FA		INCZ POINT	POINT +1	10000
1040:	027E	D0 02		BNE PNTJUM		10000
1050:	0280	E6 FB		INCZ POINT	+01	10000
1060:	0282	20 64 0C	PNTJUM	JSR VU	GOT IT	10000
1070:	0285	4C 72 02		JMP PNTUP		10000
1080:	0288	20 F3 0B	CHECK	JSR RDBYT	READ BYTE (CHKL)	10000
1090:	028B	CD 6E 1A		CMP CHKL	CHK CONTROL	10000
1100:	028E	D0 10		BNE BADTAP		10000
1110:	0290	20 F3 0B		JSR RDBYT	READ BYTE (CHKH)	10000
1120:	0293	CD 6F 1A		CMP CHKH		10000
1130:	0296	D0 08		BNE BADTAP		10000
1140:	0298	A9 00		LDAIM \$00	SET CTRL FLAG	10000
1150:	029A	8D 58 1A		STA CTRL		10000
1160:	029D	4C A5 02		JMP SCREEN		10000
1170:	02A0	A9 01	BADTAP	LDAIM \$01	SET CTRL FLAG	10000
1180:	02A2	8D 58 1A		STA CTRL		10000
1190:	02A5	20 BC 14	SCREEN	JSR RESTTY	I/O FOR PM	10000
1200:	02A8	20 E8 11		JSR CRLF		10000
1210:	02AB	20 1F 03		JSR PRINT		10000
1220:	02AE	49		= 'I'		10000
1230:	02AF	44		= 'D'		10000
1240:	02B0	3A		= ':'		10000
1250:	02B1	20		= ','		10000
1260:	02B2	20		= '\$'		10000
1270:	02B3	03		= \$03		10000
1280:	02B4	AD 79 1A		LDA ID	PRINT ID	10000

# DE 6502 KENNER

CHTAPE DE 6502 KENNER

PAGE 03

1290: 02B7 20 8F 12	JSR PRBYT
1300: 02BA AD 58 1A	LDA CTRL
1310: 02BD F0 0E	BEQ GOOD
1320: 02BF 20 1F 03	JSR PRINT
1330: 02C2 20	= ,
1340: 02C3 20	= ,
1350: 02C4 42	= 'B
1360: 02C5 41	= 'A
1370: 02C6 44	= 'D
1380: 02C7 20	= ,
1390: 02C8 20	= ,
1400: 02C9 03	= \$03
1410: 02CA 4C 00 02	JMP START
1420: 02CD 20 1F 03	GOOD JSR PRINT
1430: 02D0 20	= ,
1440: 02D1 20	= ,
1450: 02D2 4F	= 'O
1460: 02D3 4B	= 'K
1470: 02D4 20	= ,
1480: 02D5 20	= ,
1490: 02D6 20	= ,
1500: 02D7 53	= 'S
1510: 02D8 41	= 'A
1520: 02D9 3A	= :
1530: 02DA 20	= ,
1540: 02DB 20	= ,
1550: 02DC 24	= '\$
1560: 02DD 03	= \$03
1570: 02DE AD 71 1A	LDA SA +01 PRINT SA
1580: 02E1 20 8F 12	JSR PRBYT
1590: 02E4 AD 70 1A	LDA SA
1600: 02E7 20 8F 12	JSR PRBYT
1610: 02EA 20 1F 03	JSR PRINT
1620: 02ED 20	= ,
1630: 02EE 20	= ,
1640: 02EF 45	= 'E
1650: 02F0 41	= 'A
1660: 02F1 3A	= :
1670: 02F2 20	= ,
1680: 02F3 20	= ,
1690: 02F4 24	= '\$
1700: 02F5 03	= \$03
1710: 02F6 A5 FB	LDAZ POINT +01 PRINT EA,
1720: 02FB 20 8F 12	JSR PRBYT WAITING IN POINT
1730: 02FB A5 FA	LDAZ POINT
1740: 02FD 20 8F 12	JSR PRBYT
1750: 0300 20 1F 03	JSR PRINT
1760: 0303 20	= ,
1770: 0304 20	= ,
1780: 0305 43	= 'C
1790: 0306 48	= 'H
1800: 0307 4B	= 'K
1810: 0308 3A	= :
1820: 0309 20	= ,
1830: 030A 20	= ,
1840: 030B 24	= '\$
1850: 030C 03	= \$03
1860: 030D AD 6F 1A	LDA CHKH PRINT CHK
1870: 0310 20 8F 12	JSR PRBYT
1880: 0313 AD 6E 1A	LDA CHKL
1890: 0316 20 8F 12	JSR PRBYT
1900: 0319 20 F3 11	JSR PRSP
1910: 031C 4C 00 02	JMP START
1920:	

```

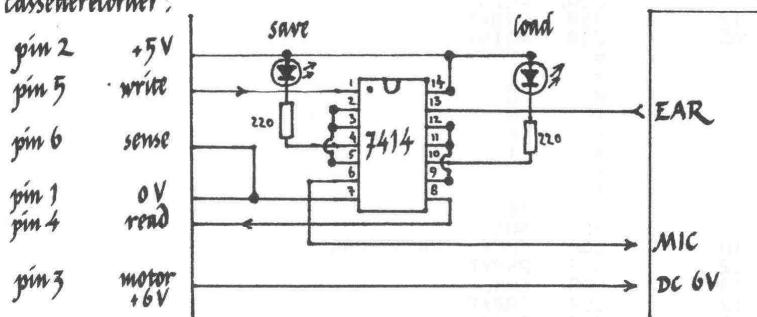
1930:      END PROGRAM
1940:
1950:
1960:      PRINT
1970: 031F 68   See Junior Paperware 1 or 6502-K nr 25
1980: 0320 85 23 PRINT PLA
1990: 0322 68   STAZ PRPNT
2000: 0323 85 24 PLA
2010: 0325 E6 23 PRTA INCZ PRPNT
2020: 0327 D0 02 BNE PRTB
2030: 0329 E6 24 INCZ PRPNT +01
2040: 032B A0 00 PRTB LDYIM $00
2050: 032D B1 23 LDAIY PRPNT
2060: 032F C9 03 CMPIM $03
2070: 0331 F0 06 BEQ PRTC
2080: 0333 20 34 13 JSR PRCHA
2090: 0336 4C 25 03 JMP PRTA
2100: 0339 A5 24 PRTC LDAZ PRPNT +01
2110: 033B 48   PHA
2120: 033C A5 23 LDAZ PRPNT
2130: 033E 48   PHA
2140: 033F 60   RTS

```

## Cassette-interface: simpel

Na in een vorig nummer getoond te hebben met welke interface-schakeling ik mijn normale audio-cassettorecorder op de CBM 64 heb aangesloten, ben ik nog wat aan het experimenteren geslagen. Zelf heb ik 'Realistic' CTR 39 recorders die op 6 Volt voedingsspanning werken. Dat is wat de 64 levert en dus de moeite van het uitproberen waard.

Met één TTL-IC blijkt het mogelijk betrouwbaar te werken met een normale cassetterecoder:



Cassette port.

Recorder.

Voor de 64 lijkt het nu net alsof er altijd een toets op de recorder is ingesleut, dus zelf op de toetsen letten bij LOAD en SAVE! Fer Weber

Fer Weber

# DE 6502 KENNER

```
0 LIST
SCR # 0
0 -->
1 DEZE FORTH WOORDEN MAKEN HET MOGELIJK OM EEN HEXDUMP
2 EN EEN ASCIIDUMP AF TE DRUKKEN
3
4 HEXDUMP    ( ADR1 ADR2 -- )
5      DRUKT DE HEXDUMP TUSSEN ADRES ADR1 EN ADR2
6      AF.
7 ASCDUMP     ( ADR1 ADR2 -- )
8      DRUKT DE ASCIIDUMP TUSSEN ADRES ADR1 EN ADR2
9      AF.
10
11 GERT KLEIN
12 DIEDENWEG 119
13 6706 CM WAGENINGEN
14 02-06-1984
15

OK
1 LIST
SCR # 1
0 ( HEXDUMP 1           GKL JUNI 84 )
1 0 VARIABLE POINT
2 0 VARIABLE ENDAD
3 HEX
4 : U. 0 D. : ( PRINT AN UNSIGNED NUMBER )
5 : FETCHBYTE POINT @ C@ : ( FETCH BYTE FROM POINT )
6 : .0 0 DO 30 EMIT LOOP : ( PRINT N NULLSZEROES )
7 : .POINTER POINT @ ( PRINT POINTER WITH LEADING ZEROES )
8 DUP 10 < OVER FFFF > AND IF 3 .0 ENDIF
9 DUP 100 < OVER OF > AND IF 2 .0 ENDIF
10 DUP 1000 < OVER FF > AND IF 1 .0 ENDIF
11 U. :
12 ( PRINT TOP ROW 0 TO F )
13 : .ROW 5 SPACES 10 0 DO I . SPACE LOOP CR :
14 : ?ASCII DUP 7F < OVER 20 > AND :( VALID ASCII ? )
15 DECIMAL -->

OK
2 LIST
SCR # 2
0 ( HEXDUMP 2           GKL JUNI 84 )
1 : HEXDUMP ENDAD ! POINT ! CR CR .ROW ( SAVE POINTERS )
2   BEGIN CR .POINTER 16 0 ( PRINT 16 BYTES )
3   DO FETCHBYTE DUP 16 < ( LEADING ZERO ? )
4     IF ( YES )
5       48 EMIT      ( PRINT ZERO )
6     ENDIF
7     .1 POINT +! ( INCREMENT POINT )
8     POINT @ ENDAD @ = ( DONE ? )
9     IF ( YES )
10    QUIT      ( TERMINATE PROGRAM )
11   ENDIF
12   LOOP      ( NEXT BYTE )
13 AGAIN : ( ALWAYS )
14
15 -->

OK
3 LIST
SCR # 3
0 ( ASCII DUMP          GKL JUNI 84 )
1 : ASCDUMP ENDAD ! POINT ! CR CR .ROW ( SAVE POINTERS )
2   BEGIN CR .POINTER 16 0 ( 16 CHARACTERS )
3   DO FETCHBYTE ?ASCII ( PRINTABLE ASCII ? )
4     IF ( YES )
5       SPACE EMIT SPACE ( OUTPUT IT )
6     ELSE ( NOT PRINTABLE )
7       DROP 3 SPACES ( OUTPUT BLANKS )
8     ENDIF
9     .1 POINT +! ( POINTER + 1 )
10    POINT @ ENDAD @ = ( DONE ? )
11    IF ( YES )
12    QUIT
13   ENDIF
14   LOOP
15 AGAIN ( ALWAYS ) : HEX ;S
```

# DE6502 KENNER

## SYM-1 Basic

Tokenized Microsoft Basic Keywords and addresses                    W.L. van Pelt  
Analogous to the publication for Commodore-64 by A. Mueller,  
DE 6502 KENNER, December 1983, pages 5-8.

COMMANDS					
KEYWORDS CORRESPONDING TO C089					
ADDRESSES CORRESPONDING TO C003					
THE ADDRESSES OF ROUTINES FOR COMMANDS ARE THE ADDRESSES					
MINUS 1, BECAUSE THE ROUTINES ARE INVOKED THROUGH RTS.					

KEYWORD	TOKEN	ADDR-1			
END	80	C623	WAIT	92	D5E2
FOR	81	C534	LOAD	93	C6B6
NEXT	82	CAD7	SAVE	94	C675
DATA	83	C781	VERIFY	-	-
INPUT#	-	-	DEF	95	D16B
INPUT	84	C9B8	POKE	96	D5D9
DIM	85	CE54	PRINT#	-	-
READ	86	C9E4	PRINT	97	C8BD
LET	87	C82E	CONT	98	C64A
GOTO	88	C72E	LIST	99	C4AC
RUN	89	C706	CLR	9A	C471
IF	8A	C7B1	CMD	-	-
RESTORE	8B	C609	SYS	-	-
GOSUB	8C	C711	OPEN	-	-
RETURN	8D	C758	CLOSE	-	-
REM	8E	C7C4	GET	9B	D001
STOP	8F	C621	NEW	9C	C455
ON	90	C7D4			
NULL	91	C664			

MISCELLANEOUS KEYWORDS					
KEYWORDS CORRESPONDING TO C0F9					

KEYWORD	TOKEN
TAB(	9D
TO	9E
FN	9F
SPC(	A0
THEN	A1
NOT	A2
STEP	A3



# DE 6502 KENNER

BASIC-SAVE- AND LOAD

PROTON 650X ASSEMBLER V4.4 PAGE: 0001

```

0001 0000      .TIT 'BASIC SAVE- AND LOAD'
0002 0000      .OPT GEN
0003 0000      .OPT SYM
0004 0000      ****
0005 0000      ****
0006 0000      ****
0007 0000      UITBREIDING OP DE SYM-1 BASIC VOOR KIM-1 EN JUNIOR
0008 0000
0009 0000      *** SAVE- EN LOAD ROUTINES NAAR FLOPPY-DISK ***
0010 0000
0011 0000
0012 0000
0013 0000      AUTEUR: F.J.M. SMEEHUIZEN
0014 0000      LIJPEDAL 19
0015 0000      2904 CL CAPELLE AAN DEN IJSEL
0016 0000      TEL: 010-512507
0017 0000
0018 0000      DE NU VOLGENDE ROUTINES ZIJN GESCHREVEN VOOR EEN SYSTEEM
0019 0000      DAT GEBRUIK MAAKT VAN DE PROTON-MONITOR.
0020 0000      VANDAAR EEN KORTE UITLEG VAN DE GEBRUIKTE SUBROUTINES.
0021 0000
0022 0000      - MET DE ROUTINE 'OUTFLD' WORDT HET OUTPUT-DEVICE BEPAALD
0023 0000      WAARNAAR GESCHREVEN MOET WORDEN.
0024 0000      - MET DE ROUTINE 'INFLD' WORDT EEN OUTPUT-DEVICE BEPAALD
0025 0000      WAARVAN GELEZEN MOET WORDEN.
0026 0000      - MET DE ROUTINE 'FLOPO' WORDT GEVRAAGD NAAR DE FILENAME
0027 0000      VAN HET WEG TE SCHRIJVEN BESTAND.
0028 0000      - MET DE ROUTINE 'FLOPI' WORDT GEVRAAGD NAAR DE FILENAME
0029 0000      VAN HET IN TE LEZEN BESTAND.
0030 0000      - MET DE ROUTINE 'CLOSED' WORDT HET OUTPUT-DEVICE AF-
0031 0000      GESLOTEN.
0032 0000      - MET DE ROUTINE 'OUTALL' WORDT DE ACCU-INHOUD NAAR HET
0033 0000      OUTPUT-DEVICE GESCHREVEN.
0034 0000      - MET DE ROUTINE 'INALL' WORDT DE INHOUD VAN HET VAN TE
0035 0000      VOREN OPGEGEVEN INPUT-DEVICE OPGEHAald.
0036 0000      - MET DE ROUTINE 'CLOSEI' WORDT HET INPUT-DEVICE
0037 0000      AFGESLOTEN.
0038 0000      - MET DE ROUTINE 'HEAD' WORDT DE LEES/SCHRIJFKOP VAN DE
0039 0000      FLOPPY VAN DE SCHIJF GELICHT EN DE MOTOR GESTOPT.
0040 0000
0041 0000      DE SAVE-ROUTINE WORDT OPGESTART DOOR HET INTOETSSEN VAN HET
0042 0000      KOMMANDO - #SAVE -
0043 0000      WAARNA VEROVOLGENS NAAR ADRES $99E8 WORDT GESPRONGEN.
0044 0000
0045 0000      DE LOAD-ROUTINE WORDT OPGESTART DOOR HET INTOETSSEN VAN HET
0046 0000      KOMMANDO - #LOAD -
0047 0000      WAARNA VEROVOLGENS NAAR ADRES $9A1A WORDT GESPRONGEN.
0048 0000
0049 0000      ZIE VOOR BEIDE KOMMANDO'S DE BESCHRIJVING VAN HET TOEVOEGEN
0050 0000      VAN NIEUWE BASIC KOMMANDO'S IN K.K. 29.
0051 0000
0052 0000
0053 0000      *** ZERO PAGE DEFINITIONS ***
0054 0000      .EX1      ; BASIC WARM START
0055 0000      .EX2      ; BASIC WARM START
0056 0000
0057 0000      ==$0050
0058 0060
0059 0060      TEMP1    ; TEMPORARY ADDRESS POINTER 1
0060 0062      TEMP2    ; TEMPORARY ADDRESS POINTER 2
0061 0078      BOBL     ; BEGIN OF BASIC SOURCE LOW
0062 007D      EOBL     ; END OF BASIC SOURCE LOW
0063 007D      BODL     ; BEGIN OF BASIC DATA LOW
0064 0081      EODL     ; END OF BASIC DATA LOW
0065 0083      BOSL     ; BEGIN OF BASIC STRING AREA LOW
0066 0087      EOSL     ; END OF BASIC STRING AREA LOW
0067 0088      PCIM     ; BASIC WARM START
0068 0088      ==$C27E
0069 C27E
0070 C27E
0071 C27E
0072 C27E
0073 99E8
0074 99E8
0075 99E8      *** SAVE-ROUTINE VOOR DE SYM-1 BASIC ***

```

BASIC SAVE- AND LOAD

PROTON 650X ASSEMBLER V4.4 PAGE: 0002

0001 0002 0003 0004

```

0076 99E8 20ED9A SAVEP    JSR OUTFLO   ; OPEN OUTPUT DEVICE
0077 99E8 20F09A          JSR FLOPO    ; ASK FOR FILENAME
0078 99EE 20359A          JSR PRGOUT  ; FILL POINTERS BASIC SOURCE AREA
0079 99F1 20BE9A          JSR DATOUT  ; SAVE PROGRAM
0080 99F4 20679A          JSR NULL    ; FILL POINTERS ZERO PAGE AREA
0081 99F7 20BE9A          JSR DATOUT  ; SAVE ZERO PAGE
0082 99FA 20F39A          JSR CLOSED   ; CLOSE FILE
0083 99FD 20FF9A          JSR HEAD    ; DESELECT FLOPPY
0084 9A00 60               RTS
0085 9A01 20ED9A SAVED   JSR OUTFLO   ; OPEN OUTPUT DEVICE
0086 9A04 20F09A          JSR FLOPO    ; ASK FOR FILENAME
0087 9A07 20849A          JSR BDATA   ; FILL POINTERS BASIC DATA AREA
0088 9A08 20BE9A          JSR DATOUT  ; SAVE DATA AREA
0089 9A0D 20A19A          JSR BSTRIN  ; FILL POINTERS BASIC STRING AREA
0090 9A10 20BE9A          JSR DATOUT  ; SAVE STRING AREA
0091 9A13 20F39A          JSR CLOSED   ; CLOSE FILE
0092 9A16 20FF9A          JSR HEAD    ; DESELECT FLOPPY
0093 9A19 60               RTS
0094 9A1A
0095 9A1A ;*** LOAD-ROUTINE VOOR DE SYM-1 BASIC ***
0096 9A1A
0097 9A1A 20029B LOADP  JSR INFLOP  ; OPEN INPUT DEVICE
0098 9A1D 20059B          JSR FLOPI   ; ASK FOR FILENAME
0099 9A20 20529A          JSR PRGIN   ; LOAD START ADDRESS BASIC SOURCE
0100 9A23 20CB9A          JSR DATAIN  ; LOAD DATA
0101 9A26 20529A          JSR PRGIN   ; LOAD START ADDRESS ZERO PAGE
0102 9A29 20CB9A          JSR DATAIN  ; LOAD DATA
0103 9A2C 20FC9A          JSR CLOSEI  ; CLOSE INPUT DEVICE
0104 9A2F 20FF9A          JSR HEAD    ; DESELECT FLOPPY
0105 9A32 4C7EC2          JMP WARM
0106 9A35
0107 9A35 ;*** SUBROUTINE TO WRITE START- AND END ***
0108 9A35 ;*** ADDRESS OF BASIC SOURCE TO FLOPPY ***
0109 9A35
0110 9A35 A57B PRGOUT  LDA BOBL   ; BEGIN ADDRESS LOW
0111 9A37 8560 STA TEMPB  ; COPY ADDRESS
0112 9A39 20F69A JSR OUTALL ; WRITE TO FLOPPY
0113 9A3C A57C LDA BOBL+1 ; BEGIN ADDRESS HIGH
0114 9A3E 8561 STA TEMPB+1 ; COPY ADDRESS
0115 9A40 20F69A JSR OUTALL ; WRITE TO FLOPPY
0116 9A43 A57D LDA EABL   ; END ADDRESS LOW
0117 9A45 20F69A JSR OUTALL ; WRITE TO FLOPPY
0118 9A48 8562 STA TEMPB  ; COPY ADDRESS
0119 9A4A A57E LDA EABL+1 ; END ADDRESS HIGH
0120 9A4C 20F69A JSR OUTALL ; WRITE TO FLOPPY
0121 9A4F 8563 STA TEMPB+1 ; COPY ADDRESS
0122 9A51 60               RTS
0123 9A52
0124 9A52 ;*** SUBROUTINE TO READ START- AND END ***
0125 9A52 ;*** ADDRESS OF BASIC SOURCE FROM FLOPPY ***
0126 9A52
0127 9A52 20F99A PRGIN  JSR INALL  ; READ FROM FLOPPY
0128 9A55 8560 STA TEMPB  ; STORE START ADDRESS LOW
0129 9A57 20F99A JSR INALL  ; READ FROM FLOPPY
0130 9A5A 8561 STA TEMPB+1 ; STORE START ADDRESS HIGH
0131 9A5C 20F99A JSR INALL  ; READ FROM FLOPPY
0132 9A5F 8562 STA TEMPB  ; STORE END ADDRESS LOW
0133 9A61 20F99A JSR INALL  ; READ FROM FLOPPY
0134 9A64 8563 STA TEMPB+1 ; STORE END ADDRESS HIGH
0135 9A66 60               RTS
0136 9A67
0137 9A67 ;*** SUBROUTINE TO SAVE ZERO-PAGE POINTERS ***
0138 9A67
0139 9A67 A965 NULL    LDA #$65  ; START OF ZERO-PAGE POINTERS
0140 9A69 8560 STA TEMPB ; WRITE TO FLOPPY
0141 9A6B 20F69A JSR OUTALL ; WRITE TO FLOPPY
0142 9A6E A900 LDA #$00
0143 9A70 8561 STA TEMPB+1 ; WRITE TO FLOPPY
0144 9A72 20F69A JSR OUTALL ; END OF ZERO-PAGE POINTERS
0145 9A75 9EA  LDA #$EA ; WRITE TO FLOPPY
0146 9A77 8562 STA TEMPB ; WRITE TO FLOPPY
0147 9A79 20F69A JSR OUTALL ; WRITE TO FLOPPY
0148 9A7C A900 LDA #$00
0149 9A7E 8563 STA TEMPB+1 ; WRITE TO FLOPPY
0150 9A80 20F69A JSR OUTALL ; WRITE TO FLOPPY
0151 9A83 60               RTS
0152 9A84
0153 9A84 ;*** WRITE START- AND ENDADDRESS OF ***

```

BASIC SAVE- AND LOAD

PROTON 650X ASSEMBLER V4.4 PAGE: 0003

0301 031 -3902 01291

```

0154 9A84 ; *** BASIC DATA AREA TO FLOPPY ***
0155 9A84
0156 9A84 A57D BDATA LDA B0DL ; BEGIN OF BASIC DATA AREA LOW
0157 9A86 8560 STA TEMPB
0158 9A88 20F69A JSR OUTALL ; WRITE TO FLOPPY
0159 9A8B A57E LDA B0DL+1 ; BEGIN OF BASIC DATA AREA HIGH
0160 9A8D 8561 STA TEMPB+1
0161 9A8F 20F69A JSR OUTALL ; WRITE TO FLOPPY
0162 9A92 A581 LDA E0DL ; END OF BASIC DATA AREA LOW
0163 9A94 8562 STA TEMPB
0164 9A96 20F69A JSR OUTALL ; WRITE TO FLOPPY
0165 9A99 A582 LDA E0DL+1 ; END OF BASIC DATA AREA HIGH
0166 9A9B 8563 STA TEMPB+1
0167 9A9D 20F69A JSR OUTALL ; WRITE TO FLOPPY
0168 9A9D 60 RTS
0169 9AA1 ;
0170 9AA1 ; *** WRITE START- AND ENDADDRESS OF ***
0171 9AA1 ; *** BASIC STRING AREA TO FLOPPY ***
0172 9AA1
0173 9AA1 A583 BSTRIN LDA B0SL ; BEGIN OF BASIC STRING AREA LOW
0174 9A83 8560 STA TEMPB
0175 9A85 20F69A JSR OUTALL ; WRITE TO FLOPPY
0176 9A88 A584 LDA B0SL+1 ; BEGIN OF BASIC STRING AREA HIGH
0177 9A8A 8561 STA TEMPB+1
0178 9AAC 20F69A JSR OUTALL ; WRITE TO FLOPPY
0179 9AAF A587 LDA E0SL ; END OF BASIC STRING AREA LOW
0180 9AB1 8562 STA TEMPB
0181 9AB3 20F69A JSR OUTALL ; WRITE TO FLOPPY
0182 9AB6 A588 LDA E0SL+1 ; END OF BASIC STRING AREA HIGH
0183 9AB8 8563 STA TEMPB+1
0184 9ABA 20F69A JSR OUTALL ; WRITE TO FLOPPY
0185 9ABD 60 RTS
0186 9ABE ;
0187 9ABE ; *** OUTPUT DATA TO FLOPPY ***
0188 9ABE
0189 9ABE A000 DATOUT LDY #$00
0190 9AC0 B160 LDA (TEMPA),Y
0191 9AC2 20F69A JSR OUTALL ; WRITE TO FLOPPY
0192 9AC5 20D89A JSR ADRES ; CALCULATE END OF DATA
0193 9AC8 B0F4 BCS DATOUT
0194 9AC8 60 RTS
0195 9ACB ;
0196 9ACB ; *** INPUT DATA FROM FLOPPY ***
0197 9ACB
0198 9ACB A000 DATAIN LDY #$00
0199 9ACD 20F99A JSR INALL ; READ FROM FLOPPY
0200 9AD0 9160 STA (TEMPA),Y
0201 9AD2 20D89A JSR ADRES ; CALCULATE END OF DATA
0202 9AD5 B0F4 BCS DATAIN
0203 9AD7 60 RTS
0204 9AD8 ;
0205 9AD8 ; *** INCREMENT DATA POINTER ***
0206 9AD8
0207 9AD8 18 ADRES CLC
0208 9AD9 A901 LDA #$01 ; ADD 1 TO TEMPORARY POINTER
0209 9AD8 6560 ADC TEMPB
0210 9ADD 9002 BCC ADR1
0211 9ADF E661 INC TEMPB+1
0212 9AE1 8560 ADRI STA TEMPB
0213 9AE3 38 SEC ; COMPARE BEGIN- AND END POINTER
0214 9AE4 A562 LDA TEMPB
0215 9AE6 E560 SBC TEMPB
0216 9AE8 A563 LDA TEMPB+1
0217 9AE9 E561 SBC TEMPB+1
0218 9AEC 60 RTS
0219 9AED ;
0220 9AED ; *** SENIOR MONITOR ROUTINES ***
0221 9AED
0222 9AED 6C38E0 OUTFLO JMP ($E038)
0223 9AFO 6C3CEO FLOPO JMP ($E03C)
0224 9AF3 6C0CEO CLOSED JMP ($E00C)
0225 9AF6 6C08E0 OUTALL JMP ($E008)
0226 9AF9 6C06E0 INALL JMP ($E006)
0227 9AFC 6C0AE0 CLOSEI JMP ($E004)
0228 9AFF 6C28E0 HEAD JMP ($E028)
0229 9B02 6C36E0 INFLOP JMP ($E036)
0230 9B05 6C3AE0 FLOPI JMP ($E03A)
0231 9B08 ;

```

# DE6502 KENNER

ASS L

```

0010 : WALLBREAKER FOR JUNIOR WITH OHIO-DOS
0020 :
0030 :
0040 : THIS PROGRAM IS A LITTLE DERIVATION OF THE
0050 : VIDEOGAME 'BREAKOUT'.
0060 : BREAK THE WALL. FOR EACH STONE YOU'LL GET
0070 : POINTS.
0080 :
0090 : STARTADDRESS=$3A7E. AS SOON AS THE CANON
0100 : GOES ON AND ON. YOU MAY SHOOT WITH <BRK>.
0110 :
0120 : THE ROUTINE FOR JUNIOR WAS PUBLISHED BEFORE
0130 : IN THE 6502 KENNER. EDITION NR. 22. AUGUST
0140 : 1982. BY W.J. KUTTERM.
0150 : NOW MODIFIED FOR JUNIOR WITH OHIO-DOS BY:
0160 :
0170 : HENK MOCKING
0180 : 't ZAND 42
0190 : 3451 GS VLEUTEN
0200 :
0210 PRCHA . DE $FEA3 : PRINT CHAR FROM ACCU
0220 CNTD . DE $FAF7 : COUNT 1024T NO IRQ
0230 RDFLAG . DE $FAD5 : FLAG-REGISTER
0240 LFEED . DE $2D6F : PRINT LINEFEED
0250 CRLF . DE $2D6A : PRINT A <CR/LF>
0260 BRKL . DE $FA7C : BREAK VECTOR
0270 BRKH . DE $FA7D :
0280 PRBYT . DE $2D92 : PRINT BYTE FROM ACCU
0290 POINTL . DE $00FA : ADDRESS POINTER
0300 POINTH . DE $00FB :
0310 PNT1 . DE $0000 : 'STAND' REGISTER
0320 PNT2 . DE $0001 :
0330 WINST . DE $0002 : 'WINST' REGISTER
0340 DWNEXT . DE $0003 : EXTRA <LF> REGISTER
0350 TEMPA . DE $FA60 : TEMPX IN PRCHA
0360 :
0370 . BA $3A7E
0380 :
3A7E- A9 0C 0390 WALL LDA #$0C : CLEAR SCREEN
3A80- 20 A3 FE 0400 JSR PRCHA
3A83- A9 B3 0410 LDA #$B3 : 132 ms TILL SCREEN IS CLEAR
3A85- 8D F7 FA 0420 STA CNTD
3A88- 2C D5 FA 0430 DELA BIT RDFLAG : WHEN DELA IS OVER. CONTINUE
3A8B- 10 FB 0440 BPL DELA
3A8D- A2 40 0450 LDX #$40 : FROM HERE ON WE ARE
3A8F- 86 FB 0460 STX *POINTH : GOING TO FILL LINES
3A91- A2 00 0470 VUL LDX #0 : WITH ' == ', AND
3A93- 86 FA 0480 STX *POINTL : ALSO STORE IT IN
3A95- 86 00 0490 STX *PNT1 : MEMORY. FROM $4001...$403F
3A97- 86 01 0500 STX *PNT2 : $4101...$413F/$4201...$423F
3A99- A9 3D 0510 VL LDA ##3D : $4301...$433F/$4401...$443F
3A9B- A0 00 0520 LDY #0 : ALSO SCORE TOTAL IS
3A9D- E6 FA 0530 INC *POINTL : SET TO 0000.
3A9F- 91 FA 0540 STA (POINTL).Y
3AA1- 20 A3 FE 0550 JSR PRCHA
3AA4- A5 FA 0560 LDA *POINTL
3AA6- C9 3F 0570 CMP #$3F : 3F HEX = 63 DEC. ELEKTERMINAL
3AA8- D0 EF 0580 BNE VL : MAX. 64 CHAR. ON A LINE. ON LAST
3AAA- 20 6A 2D 0590 JSR CRLF : POS. (64) NO "="""
3AAD- E6 FB 0610 INC *POINTH
3AAF- A5 FB 0620 LDA *POINTH
3AB1- C9 45 0630 CMP ##45 : WALL BUILD ?
3ABB- D0 DC 0640 BNE VUL
0650 :
0660 : THE WALL SHOULD BE BUILD NOW. AND STORED IN MEMORY
0670 :
3AB5- A2 0A 0680 CURDWN LDX ##0A : IF WALL BUILD. GO TO LINE 16
3AB7- 20 6F 2D 0690 LF JSR LFEED
3ABA- CA 0700 DEX
3ABB- DO FA 0710 BNE LF : ARE WE ON LINE 16 ?

```

# DE 6502 KENNER

```

3ABD- A9 E7    0720      LDA #$E7
3ABF- AD 7C FA  0730      STA BRKL
3AC2- A9 3A    0740      LDA #$3A
3AC4- BD 7D FA  0750      STA BRKH
3AC7- A2 01    0770      LDX #1      : CANON MOVES ACROSS LINE 16
3AC9- A9 0D    0780      LDA #$0D      : GO TO START OF LINE 16
3ACB- 20 A3 FE  0790      JSR PRCHA
3ACE- A9 55    0800      LP          : 0700
3AD0- 20 A3 FE  0810      LDA #$55      : 'U' IS CANON
3AD3- A9 08    0820      JSR PRCHA
3AD5- 20 A3 FE  0830      LDA #8       : REMOVE 'U' FROM SCREEN
3AD8- A9 20    0840      JSR PRCHA
3ADA- 20 A3 FE  0850      LDA #$20      : 0800
3ADD- E8      0860      JSR PRCHA
3ADE- E0 40    0870      CONT       : NEXT POSITION OF CANON AND REG.X
3AE0- F0 E5    0880      INX        : 0800
3AE0- F0 E5    0880      CPX #$40      : DID WE REACH END OF LINE OR FIRE
3AE0- F0 E5    0880      BEQ LOOP1    : BUTTON (BRK) HIT ?
3AE2- B0 28    0890      BCS SHOOT    : IF SO. ***FIRE!!***  

3AE4- 4C CE 3A  0900      JMP LP
3AE7- AE 60 FA  0920      SETX
3AEA- E0 40    0930      LDX TEMPA   : BRK SET TO THIS ADDRESS
3AEC- B0 04    0940      CPX #$40      : AT THE END OF PRCHA THERE
3AEE- B6 FA    0950      BCS RTN     : IS A TEST IF BRK IS PUSHED
3AF0- A2 41    0960      STX *POINTL : IF SO. THEN THIS SUBROUTINE
3AF2- 60      0970      LDX #$41      : IS THE END OF A JSR PRCHA
3AF3- A9 0B    0990      RTS
3AF5- 20 A3 FE  1000      OMHOOG   : PERFORM THE SHOT
3AF8- A9 2A    1010      JSR PRCHA
3AFA- 20 A3 FE  1020      LDA #$2A      : '*' IS THE BULLIT
3AFF- A9 08    1030      JSR PRCHA
3B02- A9 20    1050      LDA #$20      : BULLIT GOES UP
3B04- 20 A3 FE  1060      JSR PRCHA
3B07- A9 08    1070      LDA #8       : TILL IT REACHES
3B09- 4C 3A    1080      JSR PRCHA
3B0C- A9 08    1100      JMP PRCHA
3B0E- 20 A3 FE  1110      SHOOT     : THE WALL
3B11- A9 55    1120      LDA #$55      : BACK TO LAST PRINTED 'U'
3B13- 20 A3 FE  1130      JSR PRCHA
3B16- A9 08    1140      LDA #8       : THIS HERE MOVES
3B18- 20 A3 FE  1150      JSR PRCHA
3B1B- A0 0A    1160      LDY #$0A      : BULLIT UP
3B1D- 20 F3 3A  1170      JSR OMHOOG : ONE LINE
3B20- 88      1180      DEY
3B21- D0 FA    1190      BNE FIRE
3B23- A9 01    1200      LDA #1      : BOTTOM LINE
3B25- 85 02    1210      STA *WINST : WORTH ONE POINT
3B27- 85 03    1220      STA *DWNEXT : ALSO EXTRA <LF>
3B29- A9 44    1230      LDA #$44      : BOTTOM LINE IS ON
3B2B- 85 FB    1240      STA *POINTH : PAGE 44 IN MEMORY
3B2D- 20 F3 3A  1250      GET      : MOVE INTO LINE OF WALL
3B30- A0 00    1260      JSR OMHOOG
3B32- B1 FA    1270      LDY #0
3B34- C9 3D    1280      LDA (POINTL).Y
3B36- F0 2E    1290      CMP #$3D      : IS THERE A STONE ('=')
3B36- F0 2E    1290      BEQ STAND    : YES. PRINT TOTAL
3B38- E6 02    1300      INC *WINST
3B3A- E6 02    1310      INC *WINST : NO. MORE POINTS TO WIN
3B3C- A5 02    1320      LDA *WINST
3B3E- C9 09    1330      CMP #9       : ONLY 1.3.5.7.10 POINTS
3B40- D0 05    1340      BNE OK      : TO WIN
3B42- 18      1350      CLC          : I DIDN'T WANT TO USE
3B43- 69 07    1360      ADC #7      : DECIMAL CALCULATION
3B45- 85 02    1370      STA *WINST
3B47- C6 FB    1380      DEC *POINTH : LOOK ON NEXT LINE
3B49- A5 FB    1390      LDA *POINTH : FOR STONE ('=')
3B4B- C9 3F    1400      CMP #$3F      : I.E. CHECK MEMORY
3B4D- F0 05    1410      BEQ STRPNT : NO. MINUS POINT
3B4F- E6 03    1420      INC *DWNEXT : NEXT LINE MEANS EXTRA <LF>
3B51- 4C 2D 3B  1430      JMP GET
3B54- A9 00    1440      LDA #0      : THIS MEANS A HOLE

```

**DE 6502 KENNER**

```

3B56- 85 02      1450     STA *WINST : IN THE WALL SO 1
3B58- 38          1460     SEC : MINUS POINT
3B59- F8          1470     SED
3B5A- A5 00      1480     LDA *PNT1
3B5C- E9 01      1490     SBC #1
3B5E- 85 00      1500     STA *PNT1
3B60- A5 01      1510     LDA *PNT2
3B62- E9 00      1520     SBC #0
3B64- 85 01      1530     STA *PNT2
3B66- A9 20      1540     STAND    LDA #$20 : THIS HERE PRINTS
3B68- 91 FA      1550     STA (POINTL),Y
3B6A- F8          1560     SED : THE TOTAL POINTS
3B6B- 18          1570     CLC : YOU HAVE WON
3B6C- A5 00      1580     LDA *PNT1
3B6E- 65 02      1590     ADC *WINST
3B70- 85 00      1600     STA *PNT1
3B72- A5 01      1610     LDA *PNT2
3B74- 69 00      1620     ADC #0
3B76- 85 01      1630     STA *PNT2
3B78- D8          1640     CLD
3B79- A0 0A      1650     LDY #$0A : GO DOWN TO POSITION
3B7B- 20 6F 2D    1660     DWN      JSR LFEED : WHERE CANON WAS LEFT
3B7E- 88          1670     DEY
3B7F- D0 FA      1680     BNE DWN
3B81- 20 BF 3B    1690     JSR PRSTND : PRINT 'STAND'
3B84- A5 01      1700     LDA *PNT2
3B86- 20 92 2D    1710     JSR PRBYT
3B89- AD 00 00    1720     LDA PNT1
3B8C- 20 92 2D    1730     JSR PRBYT
3B8F- A0 08      1740     LDY #8
3B91- A9 FF      1750     DELAY   LDA #$FF : WAIT FEW SECONDS
3B93- 8D F7 FA    1760     STA CNTD : TO READ TOTAL
3B96- 2C D5 FA    1770     DEL      BIT RDFLAG
3B99- 10 FB      1780     BPL DEL
3B9B- 88          1790     DEY
3B9C- D0 F3      1800     BNE DELAY
3B9E- A9 1A      1810     BLNK    LDA ##$1A : REMOVE 'STAND: ....'
3BA0- 20 A3 FE    1820     JSR PRCHA
3BA3- A9 OD      1830     LDA ##$0D
3BA5- 20 A3 FE    1840     JSR PRCHA
3BA8- A4 03      1850     LDY *DWNEXT : EXTRA <LF> TO GO
3BA9- 20 6F 2D    1860     ZAK      JSR LFEED : TO LINE 16
3BAD- 88          1870     DEY
3BAE- D0 FA      1880     BNE ZAK
3BB0- A4 FA      1890     LDY *POINTL
3BB2- A9 20      1900     POSIS    LDA #$20 : RETURN TO POSITION
3BB4- 20 A3 FE    1910     JSR PRCHA
3BB7- 88          1920     DEY : OF THE CANON
3BB8- D0 F8      1930     BNE POSIS
3BB8- A6 FA      1940     LDX *POINTL
3BBC- 4C DD 3A    1950     JMP CONT
3BBC- 4C DD 3A    1960 :
3BBF- A9 OD      1970     PRSTND LDA ##$0D : THIS SHOULD EXPLAIN
3BC1- 20 A3 FE    1980     JSR PRCHA : IT SELF
3BC4- A9 53      1990     LDA ##$53
3BC6- 20 A3 FE    2000     JSR PRCHA
3BC9- A9 54      2010     LDA ##$54
3BCB- 20 A3 FE    2020     JSR PRCHA
3BCE- A9 41      2030     LDA ##$41
3BD0- 20 A3 FE    2040     JSR PRCHA
3BD3- A9 4E      2050     LDA ##$4E
3BD5- 20 A3 FE    2060     JSR PRCHA
3BD8- A9 44      2070     LDA ##$44
3BDA- 20 A3 FE    2080     JSR PRCHA
3BDD- A9 3A      2090     LDA ##$3A
3BDF- 20 A3 FE    2100     JSR PRCHA
3BE2- A9 20      2110     LDA ##$20
3BE4- 4C A3 FE    2120     JMP PRCHA
3BE4- 4C A3 FE    2130 :
3BE4- 4C A3 FE    2140     .EN

```

# DE 6502 KENNER

```
10 REM MAANLANDER
20 REM AFKOMSTIG VAN M. VAN HINTUM
30 PRINT "U ZIT IN UW MAANLANDER OP EEN HOOGTE"
40 PRINT "VAN 2500 METER VAN DE MAAN. UW SNELHEID"
50 PRINT "BEDRAAGT 500 M/SEC, EN U HEEFT NOG 800 LITER"
60 PRINT "BRANDSTOF. DOOR GEBRUIK TE MAKEN VAN BRANDSTOF, "
70 PRINT "DIENT U UW SNELHEID ZODANIG TE VERLAGEN, "
80 PRINT "DAT U MET EEN VEILIGE SNELHEID OP HET"
90 PRINT "MAANOPPERVLAK TERECHTKOMT. SUCCES!!!!!!"
100 LETS=500
110 LETH=2500
120 LETK=600
130 PRINT
140 PRINT "HOEVEEL LITER BRANDSTOF WILT U GEBRUIKEN ";
150 INPUTI
160 IF I=0 THEN 220
170 PRINT
180 PRINT "HO, HO, U KUNT GEEN BRANDSTOF AAN DE"
190 PRINT "RUIMTE ONTTREKKEN ! WE VERONDERSTELLEN"
200 PRINT "DUS DAT U GEEN BRANDSTOF WILT GEBRUIKEN."
210 LETI=0
220 IF I<=75 THEN 280
230 PRINT
240 PRINT "LAST VAN HEIMWEE, HM ? TERUG NAAR MOEDER"
250 PRINT "AARDE ? NOU, DAT GAAT NIET DOOR, WE GEVEN"
260 PRINT "U MAXIMAAL 75 LITER PER KEER !"
270 LETI=75
280 IF I<K THEN 330
290 PRINT
300 PRINT "ZOVEEL BRANDSTOF HEEFT U NIET MEER !"
310 PRINT "WE ZULLEN U ALLES GEVEN, WAT U HEEFT !"
320 LETI=K
330 LETK=K-1
340 LETH=H-S+(I-5)/2
350 LETS=S-I+5
360 IF INT(H+.5)<0 THEN 380
370 IF S<=5 THEN 460
380 IF INT(H+.5)>0 THEN 510
390 PRINT
400 PRINT "KWAF ! BOEM ! CRASH ! KLEUN !!!!!!!"
410 PRINT "HELAAS, UW STOFFELIJKE RESTEN LIGGEN, "
420 PRINT "TESAMEN MET DE OVERBLIJFSELEN VAN UW MAANLANDER"
430 PRINT "VERSPREID OVER HET MAANOPPERVLAK. "
440 PRINT "U KWAM MET EEN SNELHEID VAN ";INT(S+.5); " M/SEC NEER. "
450 GOTO 850
460 PRINT
```

```
70 PRINT"MIJN GELUKWENSEN, EEN GESLAAGDE LANDING !"
80 PRINT"U KUNT HET ANGSTZWEET NU VAN UW VOORHOOFD VEGEN"
90 PRINT"EN EEN WANDELINGETJE MAKEN OP HET MAANOPPERVLAK."
00 GOTO 890
10 IFK()>0THEN660
20 PRINT
30 PRINT"HELAAS, U HEEFT AL UW BRANDSTOF GEBRUIKT."
40 PRINT"UW BOORDCOMPUTER VOORSPELT, DAT U MET EEN"
50 PRINT";INT(SQR(S*S+10*H)+0.5);"
60 PRINT"M/SEC TE PLETTEN ZULT VALLEN !"
70 PRINT"HET WAS LEUK U GEKEND"
80 PRINT"TE HEBBEN. MAAR U MOET MAAR ZO DENKEN: AAN ALLES"
90 PRINT"KOMT EEND. VAARWEL, RUST IN VREDE!....."
00 FORJ=1TO1000
10 LETA=2.34**0.5
20 NEXTJ
30 PRINT
40 PRINT"KWAFF! BOEM! CRASH! KLEUN!.....(STILTE)"
50 GOTO 850
60 IFINT(S+0.5)>0THEN810
70 IFINT(S+0.5)=0THEN750
80 PRINT
90 PRINT"BANG GEWORDEN, HM ? U HEEFT ZO STERK AFGEREMD,"
00 PRINT"DAT U ZICH OP HET OGENBLIK MET EEN Snelheid"
10 PRINT";INT(S+0.5); M/SEC VAN HET MAANOPPERVLAK"
20 PRINT"VERWIJDERT. U BEVINDT ZICH NU OP ";INT(H+0.5);
30 PRINT"Meter HOOGTE."
40 GOTO130
50 PRINT
60 PRINT"U DENKT ZEKER: 'VOORZICHTIGHEID IS DE MOEDER VAN' "
70 PRINT"DE PORSELEINKAST', HM ?"
80 PRINT"U HANGT NAMELIJK STIL OP ";INT(H+0.5);
90 PRINT"Meter HOOGTE !"
00 GOTO130
10 PRINT
20 PRINT"Uw Snelheid IS NU ";INT(S+0.5); " M/SEC."
30 PRINT"Uw HOOGTE BEDRAAGT ";INT(H+0.5); " METER."
40 GOTO 130
50 PRINT
60 PRINT"DIT WAS PRUTSWERK NATUURLIJK, VOOR IEMAND"
70 PRINT"VAN UW INTELLIGENTIE. MAAR GOED, WE ZULLEN"
80 PRINT"U NOG EEN KANS GEVEN."
90 PRINT
100 PRINT"WILT U HET OPNIEUW PROBEREN (JA/NEE) "
110 INPUTQ$
120 IFQ$="JA"THEN30
130 STOP
140 END
```

# DE 6502 KENNER

JUNIOR 8 K OF JUNIOR 64 K  
\*\*\*\*\*

door : Hans Mooij  
Scharstraat 37  
1317 PL Almere

## 1.0

De JUNIOR-computer is in het begin als 8K systeem opgebouwd met 2 platen welke ik samen in een kastje heb ondergebracht.

Hiervoor heb ik verschillende software, zoals schaakprogramma (DE 6502 KENNER nr.20), One Armed Bandit (DE 6502 KENNER nr.14) en andere kleine programma's.

Later zijn er uitbreidingen gekomen welke allemaal op eurokaart formaat zijn.

Ik heb mijn JUNIOR dusdanig aangepast dat, als ik hem op het 19 inch-rek plaats, ik een 64 K systeem heb. Haal ik de stekker van de uitbreidingskaart, dan ontstaat weer mijn oude JUNIOR.

Tevens heb ik de I/O-addressering wat economischer gedecodeerd en daar-

door dan een I/O-mac (zoals ik dit noem) van F800 t/m FBFF.

intern: De PIA 6532 op de hoofdprint = FA00 - FA7F RAM  
extern: De VIA 6522 op de interfacekaart = FA80 - FB00 I/O / Timer  
De Floppy Disk kaart = F800 - F80F I/O / Timer  
De Video kaart = F900 - F90F

## 1.1

### Onderbouw

Op de 2C heb ik standaard bus signaal SEL aangedaan, dat F800 - F9FF decodeert. Van de 512 plaatsen extern zijn er nog vele onbezett. en voor elkaar toepassingen bruikbaar.

Het memory heb ik zoveel mogelijk met RAM bezet zodat er snel van systeem programma (Basic, Forth, e.d.) gewisseld kan worden. De meeste hoofdprogramma's heb ik op cassette op de adressen \$2000 - \$3FFF. Ik ben van plan hier een EPROM te gaan gebruiken met bankswitching, onder te verdelen in 8K statische RAM + 8K EPROM, 0000 - 1FFF, reso. 2000 - 3FFF. Het video-geheugen van 2K RAM is bij mij 4K geworden door 2 x 6116 op elkaar gesoldoerd op de kaart te plaatsen. Pin 18 heb ik dan voor verdere decodering gebruikt.

## 2.0

### Hardware aanpassingen: Upgrade

#### Onderdelen:

1 x 74LS12 1 x 74LS30 3 x 4K7 weerstanden

- 2.1) Doorverbinden 2a intern 2a extern van de 41612 DIN connector op interfacekaart.  
Noot: interne connector is de connector die naar de hoofdprint gaat  
externe connector is de connector die naar de buskaart gaat
- 2.2) Veranderingen hoofdprint:  
IC 2 vervalt en er komt een 2716 of 2732 voor in de plaats.  
2716 : 1K voor het gecorrigeerde MON-programma  
1K voor DOS Paperware 1 (Elektuur)  
2732 : 1K als 2716 MON (gecorrigeerd)  
1K als 2716 DOS  
1K MON voor evt. 6809E  
1K DOS voor evt. 6809E

Verwijder C 12 en C13. Kras 18a en 17c los: dit zit aan de bovenkant

# DE 6502 KENNER

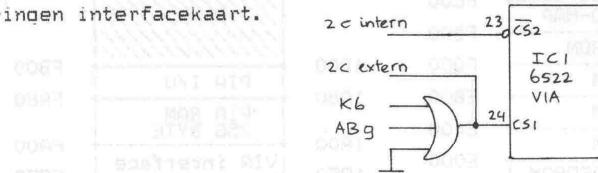
waar de toetsen zich bevinden.

Verbindt de -kant van C 13 (welke er uit is) aan +5V van IC 8 pin 14 (voor gebruik van 2732 een pull-up weerstand van 4K7).

Verbindt de +kant van C 12 (welke er uit is) met een pull-up weerstand van 4K7 aan +5V van IC 8 pin 14.

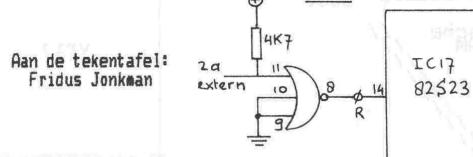
FIG.2

## 2.3) Veranderingen interfacekaart.



Haal K6 van pin 23 IC 1 los door vlakbij IC 13 pin 2 door te kraspen.  
Leg een verbinding van IC 1 pin 23 naar 2c interne connector 41612 DIN

FIG.3



Haal IC 13 er uit en kras verbinding van pin 6 naar pin 11 door. Kras vlakbij IC 13 pin 8 de verbinding door en leg een draadverbinding van pin 8 IC 13 naar R.

IC 13 pin 11 met een draad naar externe connector 41612 pin 2a. Leg tevens een pull-up weerstand van 2K2 van pin 2a naar +5V. 1a + 1c connector 41612 DIN extern.

## 2.4) Maak nu een printje met de 2 IC's 74LS30 en 74LS12 en verbindt deze op de hoofdprint.

FIG.4

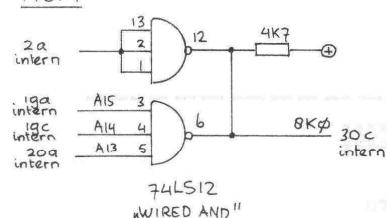
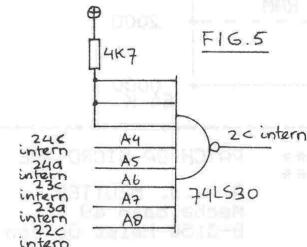


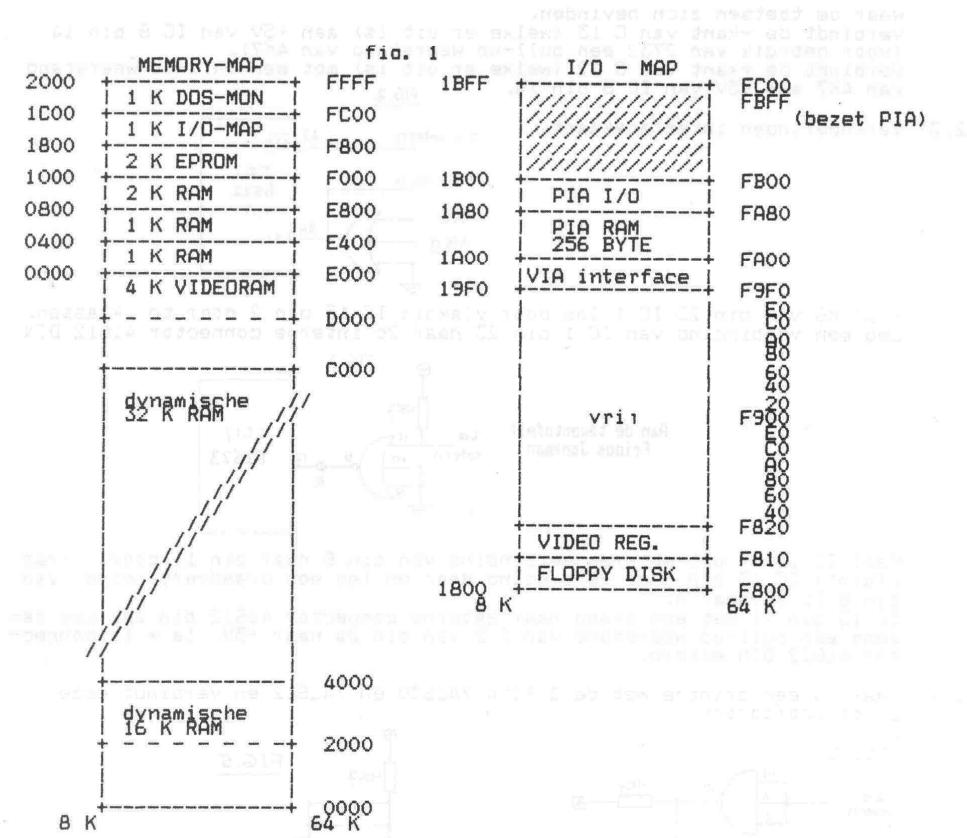
FIG.5



Bij het verwijderen van alle uitbreidingskaarten zal de JUNIOR als 8K systeem werken (zoals vroeger).  
De eerste uitbreidingskaart zal dehouden op plaats 0000 t/m 0200 moeten hebben. anders is er op deze plaats niet de nodige RAM aanwezig.

1FFF / +-----+ 8 K STANDAARD FFFF   1 K MON SYSTEME	AB CD EFGH
+-----+ 30000000 00000000	AB CD EFGH
I/O PIA+VIA   30000000 00000000	AB CD EFGH
+-----+ 30000000 00000000	AB CD EFGH
2 K EPROM TM   30000000 00000000	AB CD EFGH
+-----+ 30000000 00000000	AB CD EFGH
2 K RAM 6116   30000000 00000000	AB CD EFGH
+-----+ 30000000 00000000	AB CD EFGH
1 K RAM   30000000 00000000	AB CD EFGH
+-----+ 30000000 00000000	AB CD EFGH
1 K RAM   30000000 00000000	AB CD EFGH
+-----+ 30000000 00000000	AB CD EFGH
0000 /  -----  MEMORY MAP DOOO  -----  (64 K) / 8 K	AB CD EFGH

# DE 6502 KENNER



\*\*\*\*\* PATCH OP MICRO-ADE \*\*\*\*\*  
 \*\*\*\*\* A. v.d. MEUTTER \*\*\*\*\*  
 Mechelbaan 49  
 B-3150 Heist Op den Berg

Op Micro-ADE heb ik een kleine patch gemaakt, en wel voor LIST.  
 Normaal kan je LT gebruiken voor listino zonder regelnummer. Je  
 kan alleen terug regelnummers krijgen door PASS 2 uit te voeren.  
 Dit is niet langer zo!

Nu werk je als volgt : LI of LIST geeft regelnummers  
 LT geeft zonder nummers  
 L blijft zoals laatste keuze.

Je kan dus op ieder moment kiezen tussen LI en LT.

```

30EE EA          ;00 00 0000 0000 0000 0000
30EF EA          ;00 00 0000 0000 0000 0000
30F0 AD 01 01    ;00 00 0000 0000 0001 0001
30F3 C9 54      ;00 00 0000 0000 0001 0001
30F5 F0 0A      ;00 00 0000 0000 0001 0001
30F7 C9 49      ;00 00 0000 0000 0001 0001
30F9 F0 06      ;00 00 0000 0000 0001 0001
30FB EA         ;00 00 0000 0000 0001 0001
30FC EA         ;00 00 0000 0000 0001 0001
30FD 20 67 23   ;00 00 0000 0000 0001 0001
3100 60         ;00 00 0000 0000 0001 0001
3101 85 4D      ;00 00 0000 0000 0001 0001
3103 F0 F8      ;00 00 0000 0000 0001 0001

```

De enige wijziging is van 30F0 tot 30FC.

**6502 KENNER**

C. B. M. 64 KERNEL TEST ROUTINE

G. v. Roekel  
v. d. Palmstraat 11C  
3135 LK VLAARDINGEN

**Met een hulproutine is het mogelijk om de BASIC en MONITOR ROM te kopiëren naar RAM. Hieraan is wel een maar verbonden. Men kan wel alleen de BASIC ROM naar RAM kopiëren, maar niet alleen de MONITOR van ROM naar RAM. Voor het kopiëren van de MONITOR ROM dient u eerst de BASIC ROM naar RAM te kopiëren. Met onderstaand programma wordt de KERNAL routine gezet, waarbij u de keuze heeft om alleen de BASIC ROM te kopiëren of BASIC en MONITOR ROM.**

## KERNEL ROUTINE

```
100 PRINT"1 A000-BFFF IN RAM"
110 PRINT"2 A000-BFFF + E000-FFFF IN RAM"
120 PRINT"MAAK U KEUZE":INPUTA
130 PRINT"EVEN GEDULD A.U.B."
140 IFA=1THENGOSUB170
150 IFA=2THENGOSUB170:GOSUB190
160 END
170 FORJ=40960 TO 49151:POKEJ, PEEK(J)
180 NEXTJ:POKE1, 54:RETURN
190 FORJ=57344 TO 65535:POKEJ, PEEK(J)
200 NEXTJ:POKE1, 53:RETURN
```

TEST KERNEL



Indien u bij het eerste programma heeft gekozen voor nummer 2 dan zal na het 'RUNNEN' van het tweede programma het woord 'READY' vervangen zijn door het woord 'BLAAR'. En als u het woord 'LOAD' intoetst zal op het scherm de tekst 'DRUK PLAY TOETS IN' i.p.v. 'PRESS PLAY ON TAPE'. Mocht blijken dat het niet werkt, controleert u dan of beide programma's goed zijn ingetoetst. Anders is het noodzakelijk om met uw COMMODORE 64 terug te gaan naar de winkeliers. Het is namelijk begrepen dat er met een aantal machines geen KERNAL routine mogelijk was. Dit uitte zich over het algemeen in het niet goed functioneren van een aantal machinetaal programma's welke gebruik maakte van deze KERNAL routine.

# DE 6502 KENNER

## Versneld laden van BASIC-programma's op de SENIOR

Door: Gert van Dobroek  
Hooglanden 20  
9801 LB Zuidhorn  
Tel.: 05940 - 5627.

Bij het wegschrijven van BASIC-programma's naar schijf d.m.v. het commando "SAVE" wordt er een ASCII file aangemaakt. Bij het inlezen van een dergelijke file d.m.v. het commando "LOAD" vindt er tijdens het inlezen een omzetting plaats. Het gevolg hiervan is dat het laden van grote programma's (>16K) vele minuten vergt. Dit is te ondervangen door buiten BASIC op een memory dump van het programma te maken. Dit kan als volgt gedaan worden:

1. Vraag (binnen BASIC) het eindadres +1 van het programma op d.m.v. de odrachten:  
"CLEAR"  
"PRINT HEX\$ (&nmm-FRE(0))"  
waarbij mmm het hexadecimale adres van de hoogste RAM-locatie is. Indien BASIC dit adres bij de koude start zelf bepaald heeft, is dit maximaal HEX BFFF.  
Het beginadres van het programma is HEX 4000.
2. Ga uit BASIC met de odracht "MON". Schrijf nu het onder punt 1. gevonden geheugenobied weg de monitoropdracht "D". Schrijf bovendien pagina 0 (HEX 0000 t/m HEX 00FF) in dezelfde file weg.

Het inlezen van een op deze manier weggeschreven BASIC-programma gaat nu als volgt:  
A. Laad BASIC, doch start deze niet (of verlaat BASIC met de odracht "MON").  
B. Laad het BASIC-programma met behulp van het monitorcommando "L".  
C. Start BASIC met een warme start d.m.v. het monitorcommando "6".

## ELEKTUUR SAMSOM-65 DOS COMPUTER

In het voorjaar 1985 brengt Elektuur haar SAMSOM-65 DOS-computer uit. Op maandagavond 3 sept. 1984 werd ons deze nieuwe zelfbouw computer gedemonstreerd. De SAMSOM-65 is opgebouwd uit reeds bestaande eurokaarten, zoals de CPU- en VDU-kaart, dynamische Ramkaart en/of universele geheugenkaart, en de voor het OS65D Disk Operating System V3.3 bedoelde controllerkaart. Vergelijken met de JUNIOR, waarvoor erg veel sets boeken werden verkocht, zijn er twee in het oog lopende aangetrekkelijke verschillen. Ten eerste is de computer opgebouwd uit eurokaarten, zodat het opeen gemakkelijk kan worden ondergebracht in een 19" rek. Ten tweede is er gezorgd voor gebruikersvriendelijke software voor het monitorgebeuren. Hierin staan full screen editing en wordprocessing centraal, naast een aantal aantal utilities. De "Kenner" zullen in deze software, die tegen geringe vergoeding door Elektuur op uw diskette wordt geschreven, delen van de populaire Micro-ADE herkennen. Voor de SAMSOM-65 heeft Elektuur al specifieke toepassingen in ontwikkeling. De redactie zal de komst van deze computer alvast ondersteunen met een publicatie die het OS65D in belangrijke mate gebruikersvriendelijker zal maken. We zullen de toekomstige gebruikers ervan met de ervaringen die we in huis hebben een goed onderdaak kunnen verschaffen.

## KORREKIE

IN HET ARTIKEL "USING FORTH WITH THE 6502" UIT NR. 30 STAAT EEN FOUT. DIE IK HIERBIJ DANK ZIJ EEN BETER INZIETH IN FORTH (VIA DE METHODE VALLEN EN OPSTAAN VERKREGEN) KAN KORRIGEREN. EEN DEEL VAN DE TEKST UIT HET ARTIKEL IS OPNIEUW AFGE-DRUKT:

HET TOEGEVOEGEN VAN WOORDEN AAN EEN BESTAANDE VOCABULARY. WAARBIJ DEZE IN HET SYSTEEM "GELOCKED" MOETEN WORDEN. GAAT ALS VOLGT:

FORTH DEFINITIONS DECIMAL  
LATEST 12 +ORIGIN !  
HERE 28 +ORIGIN !  
HERE 30 +ORIGIN !  
HERE FENCE !

WANNEER EEN AANTAL WOORDEN. DIE EERDER ZIJN "GELOCKED" IN HET SYSTEEM. MOETEN WORDEN VERWIJDERD. DAN MOET HET VOL-DE WORDEN UITGEVOERD:

FORTH DECIMAL  
' AAA NFA 12 +ORIGIN !  
YYY DEFINITIONS  
' BBB NFA 28 +ORIGIN !  
' BBB NFA 30 +ORIGIN !  
' BBB NFA FENCE !  
' ZZZ 6 + 32 +ORIGIN !  
FORGET BBB

WAARIN: AAA HET LAATSTE WOORD IS VAN DE FORTH-VOCABULARY  
BBB HET EERSTE WOORD IS DAT UIT HET SYSTEEM VER-  
DWIJNT

YYY DE VOCABULARY IS WAARIN BBB VOORKOMT  
(DEZE REGEL IS ALLEEN NODIG INDIEN BBB VAN EEN ANDERE VOCABULARY DEEL UITMAAKT DAN FORTH:  
HET WOORD DEFINITIONS IS ALLEN NODIG BIJ GEBRUIK VAN FORTH VOLGENS DE 79-STANDARD)

ZZZ DE LAATST GEDIFINEERDE VOCABULARY IS DIE IN HET SYSTEEM BLIJFT

(DEZE REGEL IS ALLEEN NODIG WANNEER EEN OF MEER VOCABULARIES GEHEELE VERDwijnen).

WEL DIENT BEDACHT TE WORDEN DAT ALLEN EEN OF MEER WOORDEN. DIE EEN AANNEENSLUITENDE REEKSEN VORMEN EN DIE HET LAATST AAN HET SYSTEEM ZIJN TOEGEVOEGD OP DEZE WIJZE VERWIJDERD KUNNEN WORDEN.

FRIDUS JONKMAN  
E

## BUG IN DOS UITBREIDING VAN ELEKTUUR. SEPTEMBER 1984.

Gert Klein. Diedenweg 119. 6706 CM Wageningen meldt ons: Bij het OHIO DOS moet je eerst een file in de directory creeren alvorens je de betreffende file kunt wegschrijven. De door Elektuur gepubliceerde patch probeert dit te omzeilen. In een bepaald en helaas veel voorkomend geval leidt dit tot tamelijk ruïneuze resultaten. Wat is nu het geval: wanneer je nu een file wilt wegschrijven en er is nog geen entry in de directory aanwezig dan creert de genoemde patch zie zelf. De ruimte is echter orecies groot genoeg om de file op floo te schrijven. Wanneer je later de file gaat uitbreiden, dan maakt de patch de benodigde ruimte niet groter. Helaas is de OHIO DOS niet slim genoeg om dat zelf te ontdekken en probeert dus de file weg te schrijven in een te kleine ruimte. Daar zich is dat nog niet zo erg, maar het resultaat is wel dat de file op de floo grondig in de vernieling wordt geholpen.

STUUR DE REDAKTIE UW ZELF ONTWIKKELDE SOFTWARE/HARDWARE !

# **DE 6502 KENNER**

## **Micro-ADE Assembler/Disassembler/Texteditor for 6502 (8K)**

### **SYSTEM DESCRIPTION**

The Micro-ADE system is designed for use with any 6502 microcomputer and consists of three major programs as well as a number of utility programs. The major programs are an assembler, a disassembler, and a text editor.

The assembler is used to create machine executable code for the 6502 from a symbolic input source program. Small programs can be created and tested directly in memory. Larger programs may be written using cassette tapes for source input and object output.

The disassembler is used to list executable 6502 machine code in the symbolic assembler source format. Symbols are generated if they are defined in the symbol table.

The text editor is used to create source programs in the format required for the assembler. It contains the necessary routines for easy manipulation of text data in memory or from cassette files.

The minimum system configuration for full use of all Micro-ADE features consists of a 6502 CPU, 8K of random access memory, 2 cassette recorders with start/stop control, and an ASCII input/output device. It is possible to use all parts of the system in a restricted way with less memory and a single manually operated cassette recorder.

For some months past our club has obtained copyrights on Micro-ADE of Peter Jennings from Micro-Ware Ltd. We are allowed to place Micro-ADE at the disposal of the members of our club.

The original 4K Micro-Ade has been patched up to 8K by Sebo Woldrinah, with commands like Append Source Files, Insert Lines, Set/Reset Form Feed Flag, Choose Lines, List, Move Lines, Load ASCII-format Files, Set/Reset Page Mode, List Used Memory, Duplicate Files, Save Source Files, Print Symbol/XREF-tables, Set/Reset Page-per-File/EJECT Flag, Search Line, Print/Change String, Assemble/Execute, Set Lines/Screen and Lines/Page, Disassemble, List without line numbers, Save chosen Lines, Save ASCII-format Files, etc.

From these patches W.L. van Pelt composed an 8K source-listing, complete with comments.

Manual+Command Review: Fl. 35,00. Source Listing 8K-version: Fl. 65,00.  
Original 4K Source-Listing: Fl. 30,00. Cassettes in KIM/JUNIOR hypertape:  
4K KIM = Fl. 12,50, 8K KIM = Fl. 12,50, 8K JUNIOR = Fl. 12,50.

To be paid with Eurocheque, or on the postal account 841433 from W.L.van Pelt, Krimpen a.d. IJssel, The Netherlands. No other payments allowed.