

Negende Jaargang, Nr. 6
december 1986

DE 6502 KENNER

** DE 6502 KENNERS ** -- EEN CLUB VOOR 65xx GEBRUIKERS

De vereniging heeft leden in Nederland, België, Duitsland, Frankrijk, Spanje, Portugal, Amerika, Zambia. Het doel van de vereniging is: het bevorderen van de kennissuitwisseling tussen gebruikers van 65xx-computers, zoals KIM, JUNIOR, COMMODORE-64, APPLE, C64, PEARLCOM, AIM-65, SYM, PET, BBC ATARI, VIC-20, BASIS 108, PROTON-computers, ITT-2020, OSI, ACC 8000, ACORN ELECTRON, SYSTEM 65, PC-100, PALLAS, MINTA FORMOSA, ORIC-1, STARLIGHT, CV-777, ESTATE III, SBGS/68, NCS 6502, KEPAC System-4, Elektuur SAMSON-65 DOS computer, LASER, etc., etc.

De kennissuitwisseling wordt o.a. gerealiseerd door 5 maal per jaar DE 6502 KENNER te publiceren (1984 en 1985 6 maal zonder contributieverhoging), door het houden van clubbijeenkomsten, door het instandhouden van een cassette-bibliothek en door het verlenen van paperware-service. Regionale bijeenkomsten worden door leden georganiseerd.

Verschijningsdata DE 6502 KENNER 1985

derde zaterdag van februari, april, juni, augustus, oktober, december.

Inlichtingen over de regio-bijeenkomsten:

Gerard van Roekel
Van der Palstraat 11 - C
3135 LK Vlaardingen
Tel.: 010 - 351101

De vereniging is volledig onafhankelijk, is statutair ooggericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Voorzitter:
Rinus Vleesch-Dubois
F1. Nightingalestraat 212
2037 NG Haarlem
Tel.: 023 - 330993

Penningmeester:
John F. van Sprang
Tulp 71
2925 EW Krimpen/IJssel.
Tel.: 01807 - 20589

Leden:
Adri Hankel (05490 - 51151) Hardware/software
Erwin Visschedijk (05490 - 71416) Hardware/software
Promotie
Nico de Vries (010 - 502239) Hardware/software/PET
Erevoorzitter: Siep de Vries
Ereleden : Mm. H. de Vries - Van der Winden
Anton Mueller
Lidmaatschap : Hfl. 45,- per kalenderjaar, postrekening 3757649 t.n.v. Penningmeester KIM Gebruikers Club Ned., Krimpen/IJssel.
Lidmaatschap 86: Te voldoen uiterlijk in december 1985.
Advertenties : Tarieven op aanvraag bij de redactie.

Bijeenkomsten van de club

derde zaterdag van januari, maart, mei, september, november.

Redactie-adres en informatie over paperware etc.

Willem L. van Pelt
Jacob Jordaanstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

Redactie DE 6502 KENNER:
Willem L. van Pelt
Jacob Jordaanstraat 15
2923 CK Krimpen/IJssel.
Tel.: 01807 - 19881

** DE 6502 KENNER ** -- EEN BLAD VOOR 65xx GEBRUIKERS

DE 6502 KENNER is een uitgave van de KIM Gebruikers Club Nederland. Het blad wordt verstrekt aan leden van de club. DE 6502 KENNER wordt van kopij voorzien door leden van de club, bij de opeen van een publikatie bijgestaan door de redactie. De inzendingen van programma's dienen voorzien te zijn van commentaar in de listings en zo mogelijk door een inleiding voorafgegaan. Publikatie van een inzending betekent niet dat de redactie of het bestuur enige aansprakelijkheid aanvaardt voor de toepassing ervan. De inzendingen kunnen geschieden in assembly-source-listings, in Basic, in Basicode, Forth, Focal, Comal, Pascal, Fortran, Cobol, Logo Elan, etc. etc.

De leden schrijven ook artikelen over de door hen ontwikkelde hardware en/of aanpassingen daarop. Zij schrijven tevens artikelen van algemene aard of reageren op publicaties van andere inzenders.

DE 6502 KENNER IS EEN BLAD VAN EN DOOR DE LEDEN

Micro-ADE Assembler/Disassembler/Editor is een produkt van Micro Ware Ltd., geschreven door Peter Jennings en besteed voor alle 6502-computers. De KIM Gebruikers Club Ned. heeft de copyrights verworven nadat ons lid Sebo Woldringh de 4 K KIM-1 versie uitbreidde tot 8 K KIM-1 versie. Adri Hankel paste deze aan voor de JUNIOR. Willem L. van Pelt stelde een nieuwe 8 K source-listing voor de JUNIOR samen. De implementatie op andere systemen dan de KIM-1 en JUNIOR kan eenvoudig gebeuren door het aanzetten van de I/O-adressen, welke in de source-listing gemakkelijk te vinden zijn.

FATE Format-lister/cond. Assembler/Tape-utilities/Editor is de door ons lid Rob Baner geschreven source-listing van een 12 K universeel systeem voor de JUNIOR-computer aan de hand van het universele disk operating system van de fa. Proton Electronics te Naarden, nu geschikt voor werken met tapes. FATE wordt beschikbaar gesteld met toestemming van Proton.

In de edities van DE 6502 KENNER worden regelmatig mededelingen gedaan over de door de club georganiseerde bijeenkomsten. Ook worden bestuurlijke mededelingen gedaan, naast informatie over hetgeen in de handel te koop is. Leden die iets te koop hebben of iets zoeken kunnen dit in de edities van DE 6502 KENNER bekend maken. Ook worden wel brieven aan redactie gepubliceerd, evenals specifieke vragen van leden. De edities worden samengesteld op basis van een groot aantal prioriteiten, welke door een redactievergadering worden gehanteerd. Deze vergadering bestaat uit de vaste medewerkers zoals in de colofon vermeld. Het aantal inzendingen is groter dan in een enkele editie van minimaal 48 pagina's is te verwerken. Hierdoor kan het voorkomen dat een inzending eerst na enige tijd kan worden gepubliceerd.

DE CLUB HEEFT BEHOEFTE AAN MEER LEDEN. WIJ WILLEN MEER AAN KUNEN BIEDEN DAN NU AL HET GEVAL IS. WERF DAAROM EEN LIJF!

WILT U EEN PRIJSLIJST? STUUR EEN GEFRAKENDE ENVELOP AAN HET REDACTIE-ADRES.

Een onafhankelijke jury kent jaarlijks een aantal aanmoedigingspremies toe aan auteurs van gepubliceerde artikelen in DE 6502 KENNER.

DE 6502 KENNER

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

Adres voor het inzenden van en reacties op artikelen voor DE 6502 KENNER:
Willems L. van Pelt
Jacob Jordaanstraat 15
2923 CK Krieken a/IJssel
Tel.: 01807 - 19881

Vaste medewerkers:
Willems L. van Pelt
Gerard van Reekel
Frans Smeethuijzen

Freelance medewerkers:
Frans Bakx
Rob Banen
Fridus Jonkman
Gert Klein
Roger Langeveld
Anton Muller
Gert van Oosterhout
Ruud Uphoff

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) 1985 KIM
Gebruikers Club Nederland.

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

In verband met auteurswetgeving en andere maatregelen op het gebied van bescherming van software kan de redactie geen aansprakelijkheid aanvaarden voor inzendingen.

Inzendingen dienen afkeurig te zijn van de inzender, tenzij anders aangegeven.

INHOUDSOPGAVE DE 6502 KENNER NR. 41 DECEMBER 1985

1. Uitnodiging Bijeenkomst Ferbo Kraamemie/Assendelft	2.
2. Van de redactie	3.
3. DOS65 Corner ... Coen Kleipool, France	4.
4. Basic teksteditor Basicprogramma op de OHIO DOS JUNIOR ... Maarten van Hintum	6.
5. Screen-Editor V3.0 (SCRED 3.0) Een scheraredakteur voor Basic of Comal Voor Elektuur's JUNIOR computer met VDU-kaart ... B. de Bruine	15.
6. Graphics in Applesoft ... Hans Bosch, Twente University of Technology	29.
7. Aanmoedigingspremie artikelen 1984 ... Fred Harthoorn, Sebo Weldringh en Anton Muller veroraden de jury die de premies toekende aan J.J.A. Janssen en R. Kleipool	achterblad.

Jaap van Toledo, lid van de redactievergadering, heeft het redactiewerk moeten beëindigen. Jaap heeft het te druk met prive en studie. Zijn medewerking aan vergaderingen kenmerkte zich door de rust waarmee zaken onder de loupe nam, er oog voor houdend dat de vergaderingen meestal wat rommelig en ruweig en langdurig verlopen. Het gezelligheidsterrein overwint soos de drift om strak en doortastend te vergaderen. De gastvrijheid die wij bij hem thuis hebben genoten is van cruciale betekenis geweest voor het goede verloop ervan, en niet in de laatste plaats te danken aan zijn vrouw. Ik heb bewondering voor het feit dat zij niet nerveus werd van al dat ge-abacadabra over bitjesgemaier aangaande de digitale logica rond de krachtige instructieset van een uiterst geavanceerd processorontwerp en de additionele hardware daaromheen. Maar Jaap liet zich de wijs niet afbrengen als tussen het publiceerbaar materiaal een listing zat van welke gezegd kon worden dat er nog een deel aan ontbrak. Ik zal die opmerkingsgeest van hem niet licht vergeten.
Mag ik jullie vanaf deze plaats het beste toewensen namens ons allen.

VERGEET NIET OM LIDMAATSCHAP TE BETALEN IN DECEMBER

DON'T FORGET TO PAY IN DECEMBER YOUR 1986 SUBSCRIPTION
SEND EUROCHEQUE OF HFL. 45,- TO MR. W.L. VAN PELT

DE 6502 KENNER
INDUSTRIELENDE
HOLLAND
CODE NR. 740 65

DE 6502 KENNER

UITNODIGING BIJENKOMST

Datum : zaterdag 18 januari 1986
 Lokatie : nieuwe kantine FORBO-Krommenie
 Industrieweg 12 te Assendelft. Tel.: 075 - 280600
 ENTRÉE: FL. 10,00

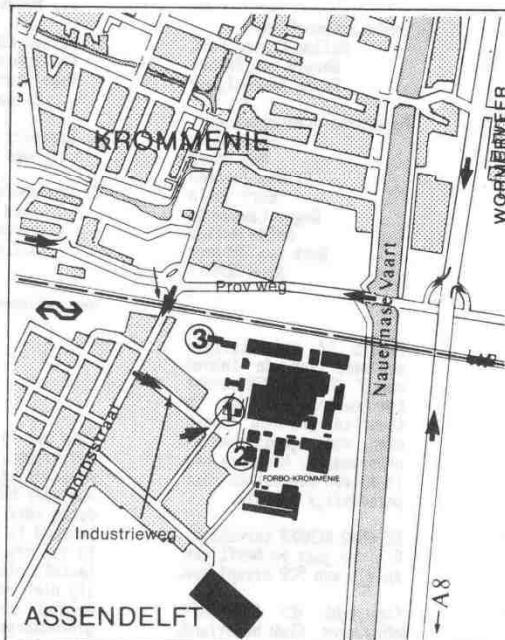
Reisroute per auto : - komende uit de richting Amsterdam Coentunnel helemaal afrijden. Aan het eind rechtsaf (water aan linker zijde). Dan 1e afslag rechts richting Uitgeest - Alkmaar. Doorrijden tot stoplichten. Linksaf spoorbaan over. Na 75 meter linksaf = Industrieweg. Links aanhoudende komt men op het FORBO-Krommenie terrein.

komende uit de richting Alkmaar Shnelweg Alkmaar - Haarlem. Afslag Uitgeest/Zaandam. Bij kruising linksaf. Bij 3e stoplichten rechtsaf spoorbomen over. Na 75 meter linksaf = Industrieweg. Links aanhoudende komt men op het FORBE-Krommenie terrein.

per trein : - Station Krommenie - Assendelft. Rechtsaf tot over spoorbomen. Na 75 meter linksaf = Industrieweg. Links aanhoudende komt men op het FORBO-Krommenie terrein.

PROGRAMMA :

- 09.30 - Zaal open, ontvangst met koffie.
- 10.00 - Opening door voorzitter en verwelkoming door de gastheer Co Filmer.
- 10.30 - Behandeling Jaarverslag 1985.
- 10.45 - Doorlopend draaiend 68000 Netwerk-systeem onder Unix.
Inleidingen over Unix en 'C'-compiler.
- 11.45 - Koffiepauze
- 12.00 - Forum
Aansluitend lunchpauze.
- 13.00 - MARKT. Ieder die iets aan te bieden heeft kondigt dit aan.
- 13.15 - INFORMEEL GEDEELTE.
In dit deel kunnen alle leden hun zelfontwikkelde hardware en software demonstreren.
Het illegaal gebruik en het kopieren van software waarop enige vorm van auteursrechten rusten is op onze bijeenkomsten ten strengste verboden!
BRENG UW EIGEN SYSTEEM MEE !!!
Het aanbieden van overbodige spullen is op eigen tafel(s) te regelen.
- 17.00 - SLUITING.



1	Portier Portier Portier Gatemanager Portero	2	Ontvangstcentrum Salle de réception Empfangsraum Reception building Sala de recepcion	3	Kantoor Bureaux Büros Offices Oficinas
----------	---	----------	---	----------	--

4	Centraal magazijn Magasin central Zentrallager Central warehouse Almacén central
----------	--

Treinverb. Amsterdam-Alkmaar (half uur dienst)
 Chemin de fer Amsterdam-Alkmaar (toutes les demi-heures)
 Eisenbahn Amsterdam-Alkmaar (jede halbe Stunde)
 Railway Amsterdam-Alkmaar (half hour service)
 Linea ferroviaria Amsterdam-Alkmaar (cada media hora)

FORBO-KROMMENIE BV
INDUSTRIEWEG 12
1566 JP ASSENDELFT - HOLLAND
075-280600

Forbo

DE 6502 KENNER

Geen editie gaat voorbij of de samenstelling ervan blijkt een hele opgave. Het streven is er steeds op gericht alle leden van onze club zoveel mogelijk tevreden te stellen. Dat valt daarom niet mee, omdat zij het in de eerste plaats zijn die moeten zorgen dat het materiaal ervoor in de copy-buffer aanwezig is. Willen we iedereen tevreden stellen, en zouden we daarvoor het materiaal hebben, dan legt het maximum aantal pagina's je weer beperkingen op. De hoeveelheid artikelen is dan groot, het aantal pagina's per artikel klein. Wat doe je dan met omvangrijke artikelen? Als je ze plaatst, dan wordt de variatie-mogelijkheid kleiner. Theoretisch zou je uit kunnen wijken naar de parware-service. Maar, is dat praktisch juist? We zijn van oordeel dat we in de eerste plaats moeten trachten alle inzendingen in de edities te publiceren. De consequentie daarvan is, dat in het geval van omvangrijke inzendingen dan nog nadrukkelijker gekeken moet worden naar de kwaliteit van de artikelen.

In deze editie denkt de redactie erin geslaagd te zijn een editie samen te stellen die, voor wat betreft het aspect kwaliteit, zeker representatief is voor wat in onze club allemaal mogelijk blijkt. De hier aangeboden routines vragen de aandacht voor zeer leerzame zaken. Wie dit niet ontgaat, zal de beperkte hoeveelheid artikelen niet als storend ervaren.

Het bestuur is al geruime tijd bezig met een bezinning op de toekomst. Dat is niet alleen een verdienste van het bestuur, het is ook een noodzaak. Centraal staat de gedachte dat het toch eigenlijk wel een beetje van de gekke is dat we leden van de club moeten zien vertrekken omdat zij zich aangetrokken voelen tot de uitdaging om ook van en met andere processoren kennis en ervaring op te doen. Vooral als het om een zeer bewuste keuze gaat en bovendien nog om mensen die voor onze club van blijvende betekenis zouden kunnen zijn, terwijl je toch realistisch moet blijven en constateren moet dat de toenemende belangstelling voor andere processoren niet te stuiven is, is een bezinning op het te voeren beleid een nuttige zaak. En als je goed door denkt, is het niet meer dan een consequent doortrekken van een lijn die zich jaren geleden al begon af te tekenen. Waren we eerst uitsluitend en alleen geïnteresseerd in machinetaal, via heel voorzichtige schreden op het gebied van Basic, kwamen we terecht in andere hogere programmeertalen als FORTH, en COMAL en nu ook PASCAL, terwijl we nog verwachten dat op het M65/DOS65-systeem straks ook al 'C' geïmplementeerd kan worden en ondersteuning nodig heeft. Machinetaal wordt daardoor niet verdronken. Integendeel! Door juist ruimte te geven aan hogere programmeertalen kun je ook later ontdekken welke beperkingen eraan verbonden zijn, naast de vele voordelen van gebruikersvriendelijkheid. Machinetaal kent geen beperkingen, zelfs niet qua gebruikersvriendelijkheid, want die bouw jezelf in. Hoe gebruikersvriendelijk een hogere programmeertaal ook is, er blijven altijd heel specifieke wensen onvervuld, die dan weer middels machinetaal worden voorzien. Het bestuur is momenteel van oordeel dat we ons best geleidelijk aan breder kunnen en misschien wel moeten oppellen v.w.b. de orientatie op processorgebied. Eigenlijk kunnen we niet heen om het feit dat een processor niets meer is dan een functie in een bepaald systeem. De Commodore 64 bijvoorbeeld kent geen 6502, maar een 6510. Je kunt je dan nog vastbijten in het feit dat hij 6502-taal-compatibel is. Maar zodra een bij uitstek 6502-machine als de APPLE haar taak feitelijk verricht middels (niet: met) een Z80, bijvoorbeeld voor onder CPM draaiende programma's en de behoefte aan meer en meer geheugen dwingt tot het gebruik van andere processoren, terwijl samenwerking met 68000 nog een kwestie van voldoende dalende prijzen is, dan zul je toch aan bijstelling van de eigen doelstelling niet ontkomen.

De keuze om geleidelijk af te tasten welke mogelijkheden er zijn in het 6X(C)XXX gebied, lijkt de enig juiste. vP.

For Elektor's JUNIOR computer Disk Operating System FDOS Koen van Nieuwenhove Introduction on the 8"/5" DOS with FDC 1771. With outlines and command summary. Dutch version, published aug. 1983	HFL. 10.00
English version	HFL. 10.00
Disk Operating System DOSAV2.0A v. Nieuwenhove Introduction on the SD+DD 5" DOS with FDC 1770. With outlines and command summary. Dutch version	HFL. 10.00
English version	HFL. 10.00
Disk Operating System DOSAV2.0A v. Nieuwenhove The complete documentation on the SD+DD 5" DOS with FDC 1770, a built-it-yourself project for Elektor's JUNIOR computer, easy to install on other computers for those who are familiar with programming in machine language. With a heavily commented source-listing, patch-routines to work with KB-9 Microsoft Basic, the assembler of C. Moser and Pascal, the base BYTE articles, etc. With outlines + command summary and introduction. English version	HFL. 97,50
FATE for DOSAV2.0A Optional: FORTH for DOSAV2.0A	
This Disk Operating System is implemented on the JUNIOR without VDU card of W.L. van Pelt and others, also not 65XX compatibles. You may wire-wrap it on VERO board 10-0581B. The con- troller is available: M.C.A. Tronix Route du Condroz 513 B-4200 LUIK, Belgium.	HFL. 22.50
DATBAS, a database-program for the JUNIOR com- puter with Elektor's VDU card and the OHIO OS- 650 V3.3 disk operating system, written by Jan van Heuven, translated by Fernando Lopez, Por- tugal into English, 15 p.	HFL. 20.00
FORTH-79 STANDARD with definition of terms, required word set, extension word sets, refe- rence word set, handy reference card, etc.	HFL. 65.00
fig-FORTH 6502 Assembly Source Listing, release 1.1, with compiler security and variable length names	HFL. 45.00
fig-FORTH Installation Manual, release 1, with compiler security and variable length names; also Glossary, Model and Editor	HFL. 20.00
FORTH-79 STANDARD CONVERSION version 1.1	HFL. 48.50
fig-FORTH Source listing for Elektor's JUNIOR- computer with Proton's Senior D.O.S.	HFL. 57.50
ASM65C 65(C)XXX-Cross-Assembler for JUNIOR- computer with Proton's Senior D.O.S., vers 1.0 Developed by Rob Banen with the FATE assembler English version, Area: \$3000-\$486A, 58 pages.	HFL. 15.00
DISASSEMBLER FOR THE ROCKWELL 65(C)02 for the JUNIOR computer with Proton's Senior D.O.S. Developed by Rob Banen with the FATE assembler English version, Area: \$2000-\$2477, 10 pages.	
Micro-ADE 8K Assembler/Disassembler/Editor for use with any 6502 computer with 2 tape-recor- ders (remote control). Very popular in our club on Elektor's JUNIOR computer. Parts of it were used in EC65 and OCTOPUS computers. Manual, based on 4K KIM-version, with command summary of the 8K version. English version. Heavily commented 8K Micro-ADE source-listing, English version.	HFL. 35.00
FATE 12K Format Lister/Assembler/Tape utility/ Editor for Elektor's JUNIOR computer without VDU card.	HFL. 65.00
Manual English version	HFL. 35.00
12K FATE source listing English version	HFL. 110.00
Send Eurocheque to Mr. W.L. van Pelt, Editorial Office. If not paying with eurocheque: HFL. 7.50 extra transfers!!	

16-Sep-85 21:00 KIM.ART1 Page 1

DATE: 26.08.85

DRAFT KIM-KENNER ARTICLE.

*** DOS-65 CORNER ***

BY: COEN KLEIPOOL, VAL DE PERIER, F-83310 COGOLIN. T.(33)94-544382

THE DOS EDITOR

IN THE FIRST ARTICLE OF THIS SERIES ON DOS-65 I BRIEFLY MENTIONED THE EDITOR, WHICH WE WILL NOW EXAMINE MORE IN DETAIL. IN FACT, THIS EDITOR IS SO EXTENSIVE THAT I COULD EASILY FILL SEVERAL PAGES.

TO ANSWER THE OBVIOUS QUESTION RIGHTAWAY, AN EDITOR SHOULD COME WITH DOS-65 TO ENSURE A PERFECT COORDINATION BETWEEN WRITING AND FILING AND TO MAKE SURE THAT RE-EDITING A FILE ALREADY WRITTEN ON DISK HAPPENS SMOOTHLY. THE FILE IS READ FROM DISK INTO MEMORY, CHANGED AND WRITTEN BACK IN SUCH A WAY THAT YOU HARDLY REALISE YOU ARE DEALING WITH A DISK-FILE. THEREFORE DOS AND EDITOR SOFTWARE SHOULD BE INTEGRATED.

MOST OF US WILL PROBABLY HAVE A LINE EDITOR INCORPORATED IN ASSEMBLER SOFTWARE. PERHAPS THIS IS A GOOD MOMENT TO EXPLAIN A FEW NOTIONS ABOUT EDITORS TO NEWCOMERS.

A LINE EDITOR STORES THE FILE IN MEMORY IN LINES, EACH LINE BEING OPENED BY A LINE NUMBER AND CLOSED BY AN END-OF-LINE CHARACTER. IF YOU WISH TO CHANGE A LINE, YOU FIRST HAVE TO GO INTO COMMAND MODE AND TYPE IN THE LINE NUMBER. THE COMPUTER WILL THEN SEARCH THROUGH MEMORY FOR THAT LINENUMBER AND PLACE THE LINE IN A BUFFER WHERE IT CAN BE EDITED, AFTER WHICH THE LINE IS PUT BACK IN ITS PROPER PLACE IN MEMORY BY USING ITS LINE NUMBER. IF THE EDITED LINE IS DIFFERENT IN LENGTH, THE COMPUTER MUST THEN PERFORM SOME BLOCKMOVING.

ALL THIS CAN BE USED, BUT IS NOT VERY PRACTICAL. REAL TROUBLE STARTS HOWEVER IF YOU TRY TO MOVE A BLOCK OF SOURCE TO A DIFFERENT LOCATION IN THE FILE. THE FIRST TIME I DID THIS IN MICRO-ADE, I THOUGHT MY COMPUTER HAD BROKEN DOWN ! IT TOOK HIM MORE THAN 20 MINUTES TO MOVE 30 LINES, AMPLE TIME TO GO DOWN TO THE LOCAL BISTRO AND HAVE A DRINK.

YOU WONT HAVE THESE PROBLEMS WITH A FULL-SCREEN EDITOR. THE FILE IS WRITTEN IN MEMORY AS A CONTINUOUS STRING AND THE SCREEN IS A WINDOW WHICH YOU CAN PLACE ANYWHERE OVER THE FILE TO READ A PAGE. YOU CAN SCROLL UP AND DOWN AS IF YOU ARE READING PAGES OF A BOOK. TO CHANGE THE TEXT, JUST PLACE THE CURSOR OVER A CHARACTER AND DELETE OR INSERT.

ANY EDIT COMMAND IS EXECUTED BY MANIPULATING THE CURSOR, WHICH IS LINKED BY SOFTWARE TO THE IMAGINARY CURSOR IN THE COMPUTER'S MEMORY.

I HAVE USED THE DOS-65 EDITOR FOR SOME TIME AND I AM STILL IMPRESSED BY ITS MANY FACILITIES. IT CAN BE USED AS A WORD PROCESSOR, JUST HAVE A LOOK AT THE FOLLOWING COMMANDS:

DE6502 KENNER

PASTE: ALLOWS TO PUT A PARAGRAPH OF THE FILE IN PASTE-BUFFER AND INSERT THIS PASTE IN ANOTHER PART OF THE FILE;

SEARCH & CHANGE: AFTER YOU HAVE DEFINED A SEARCH AND A CHANGE-STRING, THE COMPUTER WILL SEARCH THROUGH MEMORY AND REPLACE THE SEARCH STRING EVERYWHERE BY THE CHANGE-STRING.

MACRO: THE PROGRAM CAN LEARN A STRING OF EDIT-COMMANDS WHICH CAN BE USED MANY TIMES OVER.

FILL: REARRANGES THE LINES IN SUCH A WAY THAT EACH LINE BECOMES A STANDARD LENGTH.

SFILL: INSERTS SPACES TO PROVIDE A STRAIGHT RIGHHAND EDGE TO THE TEXT.

NEEDLESS TO SAY THAT ALL THESE FACILITIES MAKE THE DOS-65 EDITOR ALMOST A WORDPROCESSOR. BUT THERE IS MORE, THE REAL PURPOSE IS OBTAIN A POWERFUL TOOL TO WRITE SOURCE WHICH CAN BE USED WITH SEVERAL ASSEMBLERS. THIS IS NOT TOO OBVIOUS AS EACH ASSEMBLER USES A DIFFERENT SYSTEM.

AD BROUWER HAS WRITTEN A CONVERSION PROGRAM FOR BOTH MOSER AND MICRO-ADE WHICH MAKES THIS POSSIBLE. YOU HAVE TO WRITE THE SOURCE IN ACCORDANCE WITH THE MOSER OR MICRO-ADE FORMAT (AS FAR AS SPACES AND TABS GO) BUT THERE IS NO NEED TO WORRY ABOUT LINENUMBERS OR END-OF-LINE CHARACTERS. IF YOU USE THE NEW "GET" COMMAND IN MOSER, LINENUMBERS AND END-OF-LINE CHARACTERS WILL BE AUTOMATICALLY INSERTED IN THE MOSER SOURCE BUFFER BY THE CONVERSION PROGRAM. THE MOSER OR MICRO-ADE LINE-EDITOR IS NO LONGER USED.

TO DEBUG, THE SOURCE IS RE-EDITED IN DOS-65, AFTER WHICH YOU CAN MAKE ANOTHER ATTEMPT TO ASSEMBLE. IF MOSER SHOWS AN ERROR MESSAGE IN A CERTAIN LINE, DOS-65 CAN JUMP TO THAT LINE NUMBER ALTHOUGH IT WONT SHOW THE ACTUAL NUMBER.

THE "NEW" MOSER ALSO HAS SPECIAL COMMANDS FOR DOS REDIRECTS. THE INPUT REDIRECT MAKES IT POSSIBLE TO PUT ALL THE ASSEMBLY COMMANDS IN ONE INPUT FILE; RUNNING THAT FILE MAKES ASSEMBLY AUTOMATIC. AT THE SAME TIME THE ASSEMBLED LISTING IS WRITTEN ON THE OUTPUT FILE FOR FURTHER PROCESSING. YOU CAN IMAGE THAT IS IT QUITE A THRILL TO WORK WITH THIS PROFESSIONAL SYSTEM !

THERE IS A LOT MORE TO TELL, BUT THAT GOES BEYOND THE SCOPE OF AN INTRODUCTORY ARTICLE. I SUGGEST THAT YOU HAVE A GOOD LOOK AT DOS-65 AT THE NEXT KIM-CLUB MEETING !

NEXT ISSUE: DOS-65 MEMORY MAPPING.
A DOS-65 DISASSEMBLER.

4651 CHARACTERS, 96 LINES.



DE6502 KENNER

H I N D T E K S T

Basic teksteditor by M.R. van Hintum.
Wolborgenmate 130.
7006 DL Doetinchem.
08340-25320.

Deze teksteditor is geschreven in Microsoft Basic en draait bij mij op de OHIO-DOS-Junior.

De originele Basic is met bepaalde instructies uitgebreid die ik later zal verklaren en waarvoor ik equivalenten instructies zal geven, die waarschijnlijk in elke andere Basic voorkomen.

Basic heeft een nadeel wat betreft de snelheid, doch is naar ieders wens makkelijk aan te passen. Waar het mogelijk was heb ik zoveel mogelijk met de snelheid rekening gehouden.

De volgende instructie en/of tellers komen voor :

A - INV.
B - STO.
C - DOS.
D - DIS.
E - DIR.
F - DEL.
G - CNT.
H - INS.
I - MOV.
J - SCE.
K - EDI.
L - EIN.
M - *XXY
N - #XX#
O - ''
P - SPC=...
Q - BYT=...
R - UIT.
INV. staat voor invoeren van tekst (zie ook M,N en O). Hierbij verschijnt links in beeld het regelnummer. Tekst kan ingegeven worden, totdat de input-buffer vol is.

B.
STO staat voor stoppen en betekent alleen aan het begin van een regel, dat de invoer van tekst afgesloten moet worden.

C.
DOS staat voor Doe Out System en betekent, dat de aanwezige tekst in een file op diskette weggeschreven wordt, waarvan eerst de naam opgegeven dient te worden.

D.
DIS staat voor Doe In System en betekent, het tegenovergestelde als punt C.

E.
DIR geeft een directionaire van de geselecteerde diskdrive.

F.
DEL staat voor delete lines. Waarna de begin en eindregel opgegeven is, wordt het blok regels tussen deze waarden uit de tekstfile verwijderd.

G.

CNT staat voor continue. Wanneer men de aanwezige tekstfile wil uitbreiden met meer regels dient voor invoeren van tekst dit commando gebruikt te worden.
Hierbij verschijnen de laatste 3 regels van de uit te breiden tekst boven in het scherm, zodat men weet wat de laatste tekst was.

H.

INS. staat voor insert. Hiermee kunnen regels tekst tussen gevoegd worden. De laatste 3 regels verschijnen boven in beeld. Elke nieuwe ingevoerde regel is de laatste van deze 3 regels. Bij het gebruik hiervan dient het regelnummer opgegeven te worden, waarna tussengevoegd moet worden.

I.

MOV. staat voor move. Dit commando verplaatst een blok tekst voor een in te geven regelnummer. Voor het blok dient een begin- en een eindregel in te geven.

J.

SCE. staat voor scanen. Dit commando zoekt in de tekstfile naar een op te geven stuk tekst. Wanneer dit gevonden is verschijnt het regelnummer links, de regel tekst, en rechts op welke positie in de regel het begin van de string gevonden is.

K.

EDI. staat voor teksteditor. Hiermee kan een string door een ander string vervangen worden.
Dit kan voor de gehele of voor een gedeelte van de tekst gelden.
Wanneer dit geldt voor een gedeelte (wat natuurlijk sneller gaat) geldt dient het begin- en het eindregelnummer opgegeven te worden.
Ook is dit een handig commando om steeds terugkomende tekst eerst in te geven door een bepaald karakter, waarna met dit commando dit bepaalde karakter veranderd wordt door de steeds terugkomende tekst.
B.v. de string ''MINISTERIE VAN CULTUUR EN WETENSCHAPPEN'' wordt eerst ingegeven als ''%'', waarna met het commando EDI alle voor-komende strings ''%'' worden veranderd in ''MINIST -enz.''.
Deze commando kan ook gebruikt worden om een string te wissen.

L.

EIN. staat voor einde van het programma en komt terug met de OK-prompt van Basic.

M.

(*XXY staat voor XX maal het karakter Y. B.v. *10# vult een regel met 10 maal het karakter #.

N.

(#XX# staat voor tab(XX)). B.v. #10#Hallo betekend dat Hallo op de 10-e positie in de regel wordt gezet.

O.

'' betekend, dat voor het ingeven van een regel, waarvan de eerste karakters spaties zijn deze regel vooraf dient te worden gegaan door aanhalingsstekens. Deze zijn alleen maar softwarematig nodig, daar Basic leadingspace niet 'ziet' bij input.

P.

SPC= staat voor diskspace en is een teller hoeveel tracks de aanwezige tekstfile in de OHIO-DOS inneemt.

Q.

BYT= staat voor aantal bytes dat de tekstfile lang is. Het aantal aanslagen plus de CR's en LF's.

R.

UIT. staat voor uitvoer naar zowel beeldscherm als printer voor de gehele of gedeelteijke tekst en met of zonder regelnummers.

De printerinstructies gelden voor een Epson RX-80F/T.

Hieronder vallen niet de instructies :

- Auto page printing.
- More copy printing.
- Terug naar menu.
- Printer test.
- Start printer uitvoer.

Enkele instructie en eventuele alternatieven :

- Poke's zijn niet noodzakelijk, wel vervangen door RETURN.
- PRINT&(X,Y) ''Hallo''
print op de coördinaten X en Y de string ''Hallo''.
te vervangen door :
FORA=1TOX:PRINT:NEXT:PRINTTAB(Y) ''HALLO''
- PRINTUSING ''##''/5
print alle enen, tieners en honderdtallen onder elkaar.
te vervangen door :
I\$=STR\$(I):PRINTTAB(4-LEN(I\$))I\$
- PRINT!(49);
maakt het beeldscherm schoon.
te vervangen door :
FORX=1TO24:PRINT:NEXT
- PRINT!(52);
maakt de rest van het beeldscherm schoon.
- regel 2300 = print geen cursor.
- regel 2310 = print cursor.
- regel 2315 = print snelknipperende cursor.
- PRINT#4,
print in output-device t.b.v. de printer.

Ik hoop dat iedereen van dit programma evenveel plezier en gemak mag hebben als ik zelf.
Voor eventuele toevoegingen sta ik altijd open.

DE 6502 KENNER

@

```

1 GOTO10
2 P2=P2+LEN(A$(I))+2:P3=1+INT(P2/1048):POKE55296,10:POKE55297,32
3 PRINT&(32,1)"SPC=";:PRINTUSING"####"P3;"TRACKS."
4 PRINT&(32,2)"BYT=";:PRINTUSING"####"P2;"BYTES."
5 POKE55296,10:POKE55297,0:RETURN
10 PRINT!(49);:POKE2888,0:POKE8722,0:POKE2972,13:POKE2976,13
20 DIMA$(640):POKE2797,58:LA=1
30 FORY=1TO71:Y$=Y$+CHR$(184):NEXT:PRINT
40 REM DATA INVOEREN
50 GOSUB820
60 GOTO890
70 GOSUB2:PRINT&(50,2)"#XX#=TABULATIE XX."
80 PRINT&(50,1)"*XY=XX MAAL 'Y'.""
90 PRINT&(50,0)""'=LEADING SPACE."
100 PRINT&(1,0)"EIN=EINDE PROGR.":IFLA=1THENP=4:GOTO110
101 P=4:BR=LA-4:IFBR<1THENBR=1
102 FORI=BRTOLA-1:PRINT&(0,P)";:PRINTUSING"##"I;": ";A$(I):P=P+1:NEXT
110 FORI=LATO640
120 PRINT&(0,P)";:PRINTUSING"##"I;:INPUTA$(I)
130 IFLEFT$(A$(I),1)="#THENGOSUB1120
140 IFLEFT$(A$(I),1)="*THENGOSUB1190
145 GOSUB2
150 IFA$(I)="EIN"THENEND
160 IFA$(I)="STO"THENPRINT&(50,1)"INS=TUSSEN VOEGEN."
161 IFA$(I)<>"STO"THEN190
170 PRINT&(50,0)"SCE=ZOEKEN.":P2=P2-3
180 LA=I:PRINT&(50,2)"DIR=DIRECTORY."
181 GOSUB5040:GOTO890
190 P=P+1:IFF>22THEN210
200 NEXT
210 P=4:PRINT&(0,4)! (52);
220 FORT=I-3TO1:PRINT&(0,P)";:PRINTUSING"##"T;SPC(2);A$(T)
230 P=P+1:NEXT
240 GOTO200
250 REMUITVOER
260 BE=1:BL$=""
270 PRINT&(20,12)";:INPUT"(M)ET OF (Z)ONDER REGEL NUMMERS";LS$
280 PRINT&(20,13)";:INPUT"(A)LLES OF (G)EDEELTELIJK";GE$
290 IFGE$="A"THENBE=1:ST=LA-1:GOTO310
300 PRINT&(20,14)";:INPUT"GEEF BEGIN-, LAATSTE REGEL (BE,LA)":BE,ST
310 PRINT&(20,14)";:INPUT"OUTPUT OF (S)CHERM OF (P)RINTER":PR$
320 IFPR$="P"THENP2=0:P3=0:GOTO1440
330 PRINT&(0,4)";! (52);
340 IFST>LATHENST=LA-1
350 P=4:GOSUB2300:FORI=BETOST:IFLS$="Z"THENTX=0:GOTO370
360 TX=5:PRINT&(0,P)";:PRINTUSING"##"I;
370 PRINT&(TX,P)A$(I)
380 P=P+1:IFF<22THEN390
383 GOSUB2315
384 PRINT&(17,3)";:PRINTUSING"##"I;
385 PRINT&(17,3)";:INPUT"VOLGEND BLAD (J/N)":BL$"
386 GOSUB2300
390 PRINT&(0,3)Y$:IFBL$="N"THENGOSUB2310:GOSUB5040:GOTO890
400 IFBL$="J"THENPRINT&(0,4)! (52);P=4:BL$=""
410 NEXT:GOSUB2310
420 GOSUB2320:GOSUB5040:GOTO890

```

```
430 REM NAAR DISK
440 PRINT&(0,4)! (52);:PRINT&(20,12)"";:INPUT"GEEF BESTANDSNAAM ";DI$|
450 DISKOPEN,6,DI$:PRINT#6,LA:FORI=1TOLA
460 PRINT#6,CHR$(34);A$(I):NEXT:DISKCLOSE,6
470 GOTO890
480 REM VAN DISK
490 PRINT&(0,4)! (52);:PRINT&(20,12)"";:INPUT"GEEF BESTANDSNAAM ";DI$|
495 TRAP3000
496 PRINT&(0,4)! (52);
500 DISKOPEN,6,DI$:INPUT#6,LA:FORI=1TOLA:INPUT#6,A$(I):NEXT
510 DISKCLOSE,6:GOTO890
520 DISK!"DI":GOSUB2320:GOTO890
530 REM WISSEN
540 INPUT"GEEF BEGIN- EN LAATSTE REGEL NUMMER (BE,LA) ";ST,EN
550 VE=EN-ST:FORT=1TOLA:A$(ST)=A$(EN+1):ST=ST+1:EN=EN+1:NEXT:LA=LA-VE-1
560 GOTO890
570 PRINT&(50,2)"#XXX#=TABULATIE XX."
571 PRINT&(50,1)"*XXY=XX MAAL 'Y'."'
572 PRINT&(50,0)''' =LEADING SPACE.''
575 PRINT&(0,6)"";:INPUT"GEEF REGEL NUMMER VAN IN TE VOEGEN REGEL ";IN
580 FORT=LATOINSTEP-1:A$(T)=A$(T-1):NEXT:LA=LA+1
581 PRINT&(0,4)! (52);:FORP1=IN-3TOIN-1:PRINTUSING"###"P1;:PRINTA$(P1):NEXT
590 PRINT:PRINTUSING"###"IN;:INPUTA$(IN)
595 IFLEFT$(A$(IN),1)="#"THENHG=I:I=IN:GOSUB1120:I=HG
596 IFLEFT$(A$(IN),1)="*"THENHG=I:I=IN:GOSUB1190:I=HG
600 IFA$(IN)="STO"THEN620
610 IN=IN+1:GOTO580
620 FORT=(IN)TOLA:A$(T)=A$(T+1):NEXT:LA=LA-1
621 PRINT&(50,1)"INS=TUSSEN VOEGEN. "
622 PRINT&(50,0)"SCE=ZOEKEN. "
623 PRINT&(50,2)"DIR=DIRECTORY. ":GOTO890
630 REMVERPLAATSEN
640 PRINT&(1,6)"GEEF BEGIN- EN EIND REGEL VAN TE VERPLAATSEN BLOK"
650 INPUT"EN REGEL WAARVOOR BLOK MOET KOMEN TE STAAN (A,B,C) ";A,B,C
660 IFA>B OR A>LA OR B>LA OR C>LA THEN 890
670 IFC>ATHEN700
680 FORK=ATOB:SA$=A$(K):FORL=KTOCSTEP-1:A$(L)=A$(L-1):NEXT
690 A$(C)=SA$:C=C+1:NEXT:GOTO890
700 FORK=BTOASTEP-1:SA$=A$(K):FORL=KTOC:A$(L)=A$(L+1):NEXT
710 A$(C)=SA$:C=C-1:NEXT:GOTO890
720 INPUT" GEEF TE ZOEKEN TEKST ";W$
730 PRINT&(0,4)! (52);:GOSUB1050
740 GOSUB2320:GOTO890
750 PRINT&(50,0)''' =LEADING SPACE.'':PRINT&(0,4)"";
760 PRINT"GEEF TE VERVANGEN TEKST":INPUT";03$"
770 PRINT:PRINT"GEEF NIEUWE TEKST":INPUT";04$"
780 IA=LA-1:BR=1:PRINT:INPUT"ALLE REGELS EDITEN (J/N) ";BR$|
790 IFBR$="J"THEN810
800 PRINT:INPUT"GEEF BEGIN- EN EIND REGEL VOOR EDITEN (B,E) ";BR,IA
810 GOSUB1250:PRINT&(50,0)"SCE=ZOEKEN. ":GOTO420
820 PRINT! (49);:PRINT&(1,0)"INV=INVOER. ";&(17,0)"STO=EINDE INV. ";AB
830 PRINT&(32,0)"EDI=EDITOR. ";&(50,0)"SCE=ZOEKEN. "
840 PRINT&(1,1)"UIT=UITVOER. ";&(17,1)"DIS=DISK IN. ";
850 PRINT&(32,1)"DEL=WISSEN. ";&(50,1)"INS=TUSSEN VOEGEN. "
860 PRINT&(1,2)"CNT=TOEVOEGEN. ";&(17,2)"DOS=DISK UIT. ";
870 PRINT&(32,2)"MOV=VERPLAATSEN. ";&(50,2)"DIR=DIRECTORY. "
```

```

880 PRINT#;
890 PRINT&(0,4)""; !(52):PRINT&(20,12)"";:INPUT"UW KEUZE ";KE$;GOTO900
891 PRINT&(0,4)""; !(52):PRINT&(20,12)"";:INPUT"MENU (J/N) ";EK$;
892 GOSUB5040:GOT0890
900 POKE2972,13:PRINT&(0,4)""; !(52);
910 IFKE$="INV"THENLA=1:P2=0:P3=0:GOT070
920 IFKE$="UIT"THEN250
930 IFKE$="STO"THENEND
940 IFKE$="DOS"THEN430
950 IFKE$="DIS"THENP2=0:P3=0:GOT0480
960 IFKE$="DIR"THEN520
970 IFKE$="DEL"THENPOKE2972,58:GOT0530
980 IFKE$="CNT"THEN70
990 IFKE$="INS"THEN570
1000 IFKE$="MOV"THENPOKE2972,58:GOT0630
1010 IFKE$="SCE"THEN720
1020 IFKE$="EDI"THENPOKE2972,58:GOT0750
1030 IFKE$="EIN"THENEND
1040 GOT0890
1050 REM ZOEK IN #
1060 P=4
1070 FORI=1TOLA:FORT=1TOLEN(A$(I)):IFMID$(A$(I),T,LEN(W$))=W$THEN1110
1080 NEXTT
1090 NEXTI
1100 RETURN
1110 PRINT&(0,P)"";:PRINTUSING"##";I;A$(I):PRINT&(68,P)T:P=P+1:GOT01090
1120 REMTABULATIE
1130 A=VAL(MID$(A$(I),2,1))*10+VAL(MID$(A$(I),3,1))
1140 IFA>254THENA=0
1150 A$(I)=RIGHT$(A$(I),LEN(A$(I))-4)
1160 SP$="":FORSP=1TOA:SP$=SP$+" ":NEXT
1170 A$(I)=SP$+A$(I)
1180 RETURN
1190 REM XX MAAL 'Y' UITPRINTEN
1200 A=VAL(MID$(A$(I),2,1))*10+VAL(MID$(A$(I),3,1))
1210 SE$=MID$(A$(I),4,LEN(A$(I))-3)
1220 A$(I)=""
1230 FORU=1TOA:A$(I)=A$(I)+SE$:NEXT
1240 RETURN
1250 REM EDITOR
1260 PRINT&(0,4)!(52);
1270 P=4:GOSUB2300
1280 FORI=BRT0IA
1290 FORT=1TOLEN(A$(I)):IFMID$(A$(I),T,LEN(03$))=03$THEN1330
1300 NEXTT
1310 NEXTI
1320 GOSUB2310:RETURN;
1330 REM
1340 IFF>21THENP=4:PRINT&(0,P)!(52);
1350 IFT=1THEN01$="" :GOT01370
1360 01$=LEFT$(A$(I),T-1)
1370 IFT+LEN(03$)-1=LEN(A$(I))THEN02$="" :GOT01390
1380 02$=RIGHT$(A$(I),LEN(A$(I))-T-LEN(03$)+1)
1390 A$(I)=01$+04$+02$
1400 PRINT&(0,P)"";:PRINTUSING"###";I;A$(I):PRINT&(68,P)T:P=P+1
1410 IFF>21THENP=4:PRINT&(0,P)!(52);

```

```

1970 PRINT#4,CHR$(27); "0"; :RETURN
1980 PRINT#4,CHR$(27); "1"; :RETURN
1990 PRINT#4,CHR$(27); "2"; :RETURN
2000 PRINT&(35,18)""; :INPUT"GEEF DE WAARDE VOOR n (=n/216 INCH)";N
2010 PRINT&(35,18)FF$; :PRINT#4,CHR$(27); "3";CHR$(N); :RETURN
2020 PRINT&(35,18)""; :INPUT"GEEF DE WAARDE VOOR n (=n/72 INCH)";N
2030 PRINT&(35,18)FF$; :PRINT#4,CHR$(27); "A";CHR$(N); :RETURN
2040 PRINT&(35,18)""; :INPUT"GEEF AANTAL REGELS ";N
2050 PRINT&(35,18)FF$; :PRINT#4,CHR$(27); "C";CHR$(N); :RETURN
2060 PRINT&(35,18)""; :INPUT"GEEF AANTAL REGELS ";N
2070 PRINT&(35,18)FF$; :PRINT#4,CHR$(27); "N";CHR$(N); :RETURN
2080 PRINT&(35,18)""; :INPUT"GEEF POSITIE ";N
2090 PRINT&(35,18)FF$; :PRINT#4,CHR$(27); "L";CHR$(N); :RETURN
2100 PRINT&(35,18)""; :INPUT"GEEF POSITIE ";N
2110 PRINT&(35,18)FF$; :PRINT#4,CHR$(27); "Q";CHR$(N); :RETURN
2120 PRINT#4,CHR$(27); "8"; :RETURN
2130 PRINT#4,CHR$(27); "9"; :RETURN
2140 PRINT#4,CHR$(27); "U";CHR$(1); :RETURN
2150 PRINT#4,CHR$(27); "U";CHR$(0); :RETURN
2160 PRINT#4,CHR$(27); "S";CHR$(1); :RETURN
2170 PRINT#4,CHR$(27); "S";CHR$(0); :RETURN
2180 PRINT#4,CHR$(27);CHR$(64); :RETURN
2190 GOTO890
2200 PRINT#4,"#":PRINT#4,
2210 PRINT&(35,18)"";
2220 INPUT"JUIST (J/N) ";N$:IFN$="N"THEN2200
2230 PRINT&(35,18)FF$; :RETURN
2240 PRINT&(0,4)! (52);
2245 IFST>LATHENST=LÄ-1
2250 INPUT"KAN GEPRINT WORDEN (J/N) ";N$:IFN$="N"THEN2240
2260 FORI=BETOST:IFLS$="Z"THEN2280
2270 PRINTUSING"##";:PRINT#4,I;
2280 GOSUB2:PRINT#4,A$(I)
2290 NEXT:GOTO891
2300 POKE55296,10:POKE55297,32:RETURN
2310 POKE55296,10:POKE55297,0:RETURN
2315 POKE55296,10:POKE55297,64:RETURN
2320 GOSUB2310:GOSUB2315
2330 PRINT&(17,3)
2340 PRINT&(17,3)""; :INPUT"DRUK 'RETURN' VOOR MENU ";ME$
2350 PRINT&(0,3)Y$:GOSUB2310:RETURN
3000 PRINT&(0,4)! (52);
3001 PRINT&(17,7)"NAAM KOMT NIET VOOR !!!!!"
3010 GOSUB2320:GOTO890
4000 PRINT&(35,18)""; :INPUT"GEEF MODE (1-9) ";MO
4005 MO=0
4010 PRINT&(35,19)""; :INPUT"GEEF BEGIN REGEL ";BR:IF MO=20 THEN4200
4011 PRINT&(35,20)""; :INPUT"GEEF EINDE REGEL ";ER:IFER>BR THEN4200
4020 CR$=CHR$(27)
4030 ONMOGOTO4050,4060,4070,4080,4090,4100,4110,4120,4130
4050 MO$=CR$+"M":GOTO4500
4060 MO$=CR$+"P":GOTO4500
4070 MO$=CR$+"W"+CHR$(1):M=1:IFMO$=CR$+"W"+CHR$(0) THENM=0:ER=1:BR=1
4075 ME$=CR$+"W"+CHR$(0):ER=1:BR=1:IFMO$=CR$+"W"+CHR$(1) THENM=1:ER=1:BR=1
4076 GOTO4500
4080 MO$=CR$+CHR$(15):M=1:IFMO$=CR$+"W"+CHR$(0) THENM=0:ER=1:BR=1

```

```

1420 T=T+LEN(03$):GOTO1300
1430 REM PRINTER INITIALISEREN
1440 PRINT&(0,4)! (52) ::PRINT&(25,10)"ZET PRINTER AAN !!!"
1450 IFJK=1THEN1470
1460 PRINT#4,CHR$(27);CHR$(64)::JK=1
1470 REM PRINTER UITVOER
1480 FF$="
1490 PRINT&(0,4)! (52);
1500 PRINT" 1 = ELITE      MODE.";
1510 PRINTTAB(36)"19 = PAPER-END IGNORE."
1520 PRINT" 2 = PICA      MODE.";
1530 PRINTTAB(35)"20 = TOP OF FORM SETTING."
1540 PRINT" 3 = ENLARGED   MODE.";
1550 PRINTTAB(35)"21 = UNIDIRECTIONEL PRINTING.
1560 PRINT" 4 = CONDENSED   MODE.";
1570 PRINTTAB(35)"22 = BIDIRECTIONEL PRINTING.
1580 PRINT" 5 = UNDERLINED  MODE.";
1590 PRINTTAB(35)"23 =  50 CHAR/SEC. PRINTING.
1600 PRINT" 6 = EMPHASIZED  MODE.";
1610 PRINTTAB(35)"24 = 100 CHAR/SEC. PRINTING.
1620 PRINT" 7 = DOUBLE-STRIKE MODE.";
1630 PRINTTAB(35)"25 = PRINTER INITIALIZATION.
1640 PRINT" 8 = SUPERSRIPT  MODE.";
1650 PRINTTAB(35)"26 = TERUG NAAR MENU.
1660 PRINT" 9 = ALTERNATE    MODE.";
1670 PRINTTAB(35)"27 = PRINTER TEST.
1680 PRINT"10 = 1/8 INCH    LINE FEED.";
1690 PRINTTAB(35)"28 = MODE PER REGEL INVOEREN."
1700 PRINT"11 = 7/72 INCH    LINE FEED.";
1705 PRINTTAB(35)"29 = START PRINTER UITVOER."
1710 PRINT"12 = 1/6 INCH    LINE FEED.
1720 PRINT"13 = n/216 INCH   LINE FEED.
1730 PRINT"14 = n/72 INCH   LINE FEED.
1740 PRINT"15 = FORM LENGTH   SET.
1750 PRINT"16 = SKIPOVER PERFOR.  SET.
1760 PRINT"17 = LEFT MARGIN   SET.
1770 PRINT"18 = RIGHT MARGIN  SET.
1780 PRINT&(35,17)"";:INPUT"UW KEUZE ";KZ:GOSUB1800
1790 PRINT&(35,17)"           ";:GOTO1780
1800 IFKZ<11THEN1830
1810 IFKZ<21THEN1840
1820 IFKZ<30THEN1860
1830 ONKZGOTO1880,1890,1900,1910,1920,1930,1940,1950,1960,1970
1840 KZ=KZ-10
1850 ONKZGOTO1980,1990,2000,2020,2040,2060,2080,2100,2120,2130
1860 KZ=KZ-20
1870 ONKZGOTO2140,2150,2160,2170,2180,2190,2200,4000,22401
1880 PRINT#4,CHR$(27);"M";:RETURN
1890 PRINT#4,CHR$(27);"P";:RETURN
1900 PRINT#4,CHR$(27);"W";CHR$(1);:RETURN
1910 PRINT#4,CHR$(15);:RETURN
1920 PRINT#4,CHR$(27);"-";CHR$(1);:RETURN
1930 PRINT#4,CHR$(27);"E";:RETURN
1940 PRINT#4,CHR$(27);"G";:RETURN
1950 PRINT#4,CHR$(27);"S";CHR$(0);CHR$(27);"A";CHR$(5);:RETURN
1960 PRINT#4,CHR$(27);"4";:RETURN

```

```

4085 ME$=CR$+CHR$(18)
4086 GOTO4500
4090 MO$=CR$+"-":CHR$(1):M=1
4095 ME$=CR$+"-":CHR$(0)
4096 GOTO4500
4100 MO$=CR$+"E":M=1
4105 ME$=CR$+"F"
4106 GOTO4500
4110 MO$=CR$+"G":M=1
4115 ME$=CR$+"H"
4116 GOTO4500
4120 MO$=CR$+"S"+CHR$(0)+CHR$(A)+CHR$(5):M=1
4122 ME$=CR$+"T"+CHR$(A)+CHR$(15)
4123 GOTO4500
4130 MO$=CR$+"4":M=1
4135 ME$=CR$+"5"
4136 GOTO4500
4200 MO$=CHR$(12)
4210 A$(BR)=A$(BR)+MO$
4500 PRINT&(35,18)FF$:PRINT&(35,19)FF$:PRINT&(35,20)FF$
4505 IF MO=20 THEN RETURN
4510 A$(BR)=MO$+A$(BR):IFM=0THENRETURN
4520 A$(ER)=A$(ER)+ME$:RETURN
5040 PRINT&(32,1)"DEL=WISSEN."
5050 PRINT&(32,2)"MOV=VERPLAATSEN.":RETURN

```

With help of our member Rene Hettfleisch we found some important informations for users of the 8K Assembler/Disassembler/Editor Micro-ADE and COMAL*.KGN.

8K Micro-ADE on Elektor's JUNIOR computer with VDU-card.

Users of the JUNIOR computer with VDU-card that want to implement Micro-ADE on their system only need changing following addresses:

```

old: 2E9D 4C 00 30 INPUT JMP INPRTN ; THE KEYBOARD INPUT ROUTINE
new: 2E9D 4C AE 12 INPUT JMP RECCHA ; " " " "
old: 2EA0 4C 34 13 OUTPUT JMP PRCHA ; THE PRINTER OUTPUT ROUTINE
new: 2EA0 4C 8A 16 OUTPUT JMP NPRCHA ; THE NEW PRCHA ROUTINE AS
                                GIVEN IN ELEKTOR'S PAPERWARE 4,
                                INFORMATIONS AND SOURCES ABOUT
                                IMPLEMENTING THE VDU-CARD ON
                                JUNIOR WITH CASSETTE OR FLOPPY-
                                DISK INTERFACE

```

COMAL*.KGN on Elektor's JUNIOR computer with VDU-card.

To implement COMAL*.KGN on Elektor's JUNIOR computer using the VDU-card and Elektor's Paperware 4 software following addresses must be defined as follows:

```

old: 2553 20 AE 12 INPCH JSR RECCHA
new: 2553 20 AE 12 INPCH JSR RECCHA
old: 47FA 20 34 13 OUTCH JSR PRCHA
new: 47FA 20 8A 16 OUTCH JSR NPRCHA
old: 47F5 20 E8 11 CRLF JSR CRLF
new: 47F5 20 E8 11 CRLF JSR CRLF
old: 2C11 20 AE 12 INPCH JSR RECCHA
new: 2C11 20 AE 12 INPCH JSR RECCHA

```

DE6502 KENNER

SCREEN-EDITOR VOOR JUNIORCOMPUTER MET VDU-KAART

SCREEN-EDITOR V3.0

1. INLEIDING

Een 'schereredakteur' (screeneditor) onderscheidt zich van een regeleditor door de mogelijkheid om regels (bijvoorbeeld ten gevolge van een LIST-commando) die op het TV-scherm staan ter olaatse te wijzigen.

De hier beschreven editor kan niet zelfstandig werken, maar moet samenwerken met Basic of Comal. Dit maakt SCRED zeer geschikt voor systemen die met tapes werken.

Computers met diskdrives zullen veelal een onafhankelijke editor bezitten, waarmee ASCII-files aangemaakt worden. Deze files worden dan in no-time via disk weer in Basic (of wat dan ook) ingelezen worden. Met cassette-recorders zou dit een tijdrovende zaak worden. Vandaar dat SCRED toegevoegd wordt aan de programmeertaal.

De SCRED3.0 schermeditor kent drie verschillende gebruiksmogelijkheden:

- a. De gecombineerde edit/windowmode (overwritemode)
- b. De insertmode
- c. De commandmode

De edit/windowmode is de toestand waarin een programma wordt ingetikt met het toetsenbord. Eventuele fouten kunnen aangepast worden met de daarvoor bestemde toetsen.

Wil men een regeldeel tussenvoegen in een bestaande regel, dan moet men overschakelen naar de insertmode.

Voor speciale niet-schermgebonden bewerkingen is de commandmode. Momenteel kent de commandmode slechts het commando Quit, waarmee van Basic (of Comal) naar Junior-monitorniveau wordt gesorongen.

De meeste mensen ervaren een schermeditor als veel gebruikersvriendelijker dan een regeleditor. Vooral voor Basic (KB-9) en Comal (*KGN), die nauwelijks editmogelijkheden bezitten is deze editor een uitkomst.

2. HET PRINCIEP VAN DE SCREENEDITOR.

De hoofdrol in het screen-editorprogramma wordt gespeeld door pijltje-vooruit (→). Als de pijltje-vooruitroutine aangeroept wordt, gebeurt er het volgende:

- I. Lees de plaats in videoram waar de cursor zich bevindt (bijv. D600).
- II. Haal het karakter dat in de cursor staat (oo adres D600) in Accu en AHold.
- III. Sla dit karakter op in de buffer (van Basic of Comal of ...) en verhoog de bufferwijzer met 1.
- IV. Schuif de cursor 1 positie naar rechts.

Door herhaaldelijk (→) over een regel te verlaatsen, denkt Basic (of Comal) dat er een regel wordt ingetikt.

Met dit principe is inherent de beperking verbonden dat sommige editoperaties indirect gerealiseerd moeten worden (bij regeloverschrijding).

3. COMMANDO'S.

De besturing van de editfunkties vindt plaats op twee manieren:

- I. Direct.
Hieronder vallen de pijltjes, char-insert en char-delete.
- II. Via ESC of CNTR+N code.
(ESC) karakter betekent: Druk op de ESCAPE-toets, laat deze weer los en tyoe het gewenste karakter in.
(CNTR+N) karakter betekent: Druk gelijktijdig de CONTROL-toets en de N-toets in, laat deze weer los en tyoe het gewenste karakter in.

Bij full-screen-editing is het van groot belang dat ook de Elektuur PMV-commando's beschikbaar zijn. Ook deze commando's zijn in het commando-overzicht opgenomen.

Versie 3.0 bestaat uit de volgende commando's:

DE6502 KENNER

A. Schermeditorcommando's.

(Dit zijn commando's die alleen op een regel toepasbaar zijn.)

(↓)	Stuur de cursor 1 regel omlaag.	0.0V ROTIRE-KENNEN
(↑)	Stuur de cursor 1 regel omhoog.	
(ESC)V	Stuur de cursor 12 regels omlaag.	0.1013.31 .1
(ESC)6 of (ESC)†	Stuur de cursor 12 regels omhoog. (machtverheffingsteken)	
(ESC)1 of (ESC)2	Veeg het scherm schoon en positioneer de cursor in de linkerbovenhoek (Clear screen + Home).	
(ESC)3	Zet de cursor in de linkerbovenhoek (Home).	
(ESC)4	Veeg het scherm schoon vanaf de cursor (Screen-erase).	

B. Regel-edit commando's.

(Dit zijn commando's die alleen op een regel toepasbaar zijn.)

B1. Karaktergeoriënteerde commando's.

(→)	Ga naar het volgende karakter (cursor 1 positie rechts)
(←)	Ga naar het vorige karakter (cursor 1 positie links)
(CHR.INS)	Voeg een soatie tussen.
(CHR.DEL)	Verwijder het karakter waar de cursor op staat.

B2. Regelgeoriënteerde commando's.

(CR)	Sla de regel waarop cursor staat op in het geheugen.
(ESC)(<)	Ga met de cursor naar het begin van de regel.
(ESC)-->	Ga met de cursor naar het eind van de regel.
(ESC)A	Verandert alle letters tussen de cursor en het einde van de regel in HOOFDletters (capitals).
(ESC)a	Verandert alle letters tussen de cursor en het einde van de regel in kleine letters.
(ESC)u	Escape-soatiesbalk voegt 20 soaties tussen.
(ESC)I	Verleent toegang tot de insertmode.
(ESC)C	Verleent toegang tot de commandmode.
(ESC)8	Verwijderd het regeldeel vanaf de cursor tot aan het einde van de regel (Line erase).
(CNTR+N)	Verwijderd alle soaties tussen de cursor en het eerstvolgend karakter dat geen soatie is.
(CNTR+X)	Cancel; verwijder de tot nu toe ingetikte regel en zet de cursor op de volgende regel (alleen voor KB-9 Basic; was voorheen '18').

C. Commandomodecommando's.

(na (ESC)c)	
T (CR)	Zet in Basic de terminalwidth op 80 karakters.
Q	Verlaat Basic/Comal.
D	Verlaat Basic/Comal en start de disassembler op FC4E.
(CR)	Verlaat de commandmode.

4. GEBRUIKSAANWIJZING.

De werking van de meeste commando's is grotendeels al besproken in het commandoverzicht. Enkele commando's verdienen een gedetailleerde toelichting.

(CR) Slaat de regel waarop de cursor staat op in het geheugen. Het maakt niet uit op welke positie in de regel een (CR) wordt gegeven, omdat toch de gehele regel wordt afgetast (no erasure).

Als men bijvoorbeeld LIST intikt, betekent dit wel dat achter de LIST-instructie geen tekst meer mag staan. Is dit wel het geval, dan moet eerst een (ESC)8 en dan pas een (CR) worden ingetikt.

(CNTR+N) Kar. Voert de inversebewerking van (ESC) kar. uit. Ga over van overwritemode (ED/WI MODE) naar insertmode. Na (ESC)I wordt alle tekst tussengevoegd op de plaats waar de cursor zich bevindt.

DE 6502 KENNER

Op de eerste regel van het scherm verschijnt de insertmode-indicatie. Dit is om verwarring te voorkomen, als men bijvoorbeeld na een koffiepauze terug komt en niet meer weet in welke mode men zit.
De insertmode kent de volgende commando's:
(--) : Verwijder het laatste tussengevoegde karakter.
(--) : Verlaat de insertmode, en positioneer de cursor 1 positie naar rechts.
(CR) : Verlaat de insertmode, en sla de bewerkte regel op in het geheugen.

Verder werken uiteraard de volgende editcommando's nog steeds:
(CHR.INS), (CHR.DEL), (ESC) (-- , (ESC) --) , (ESC) , en (CNTR+N).

Opmerking : SCRED 3.0 start altijd op in de overwrite-mode (default).

(ESC)C Q Verlaat Basic/Comal in full-duplexmode.
Een break resulteert in een warme start van Basic/Comal.
Wordt een Quit midden in een regel gegeven (onder voorafgaande (RETURN)), dan wordt die regel niet in het geheugen opgeslagen.

5. MOGELIJKHEDEN EN BEPERKINGEN VAN SCRED 3.0

Mogelijkheid om regels te kopiëren. Door het regelnummer van een regel die al in het geheugen is opgeslagen te veranderen en een (CR) te geven, wordt de betreffende regel gekopieerd met het nieuwe regelnummer.

Suggestie: Maak veelvuldig gebruik van de PMV-videocommando's. Deze commando's maken het mogelijk een overzichtelijk scherm op te maken.

De vele editfunkties maken het plezierig om foutloos een file in te voeren, of te editeren. Toch kent SCRED enkele beperkingen:

a. Bij een toetsenbord met repeaterende toetsen, kan het voorkomen dat van een (↑) of een (↓) gebruikt in autorepeat, de code te snel bij de JUNIOR aankomt. Dit leidt tot een break bij de originele Elektuur RECCHA-routine. Dit gebeurt vooral als men kolom 80 van het scherm nadert.

Oplossing 1: Geef eerst een (ESC) --, en repeat dan in kolom 0. De SCRED-software hoeft dan niet de hele regel af te tasten, maar alleen kolom 0. Zodoende kan de Edit-software dit bijhouden.

Oplossing 2: Verlaag de repeatsnelheid van het keyboard.
Oplossing 3: Maak vooraf gebruik van (ESC)6 en (ESC)V of HOME, en stuur met (↑) of (↓) zonder te reoeteren naar de juiste regel.

Oplossing 4: Gebruik auto-repeat niet. Druk telkens de pijltjestoets in, kijk naar het beeldscherm, en herhaal dit totdat de cursor juist gepositioneerd is.

Oplossing 5: Vervang de routine op \$12AE door een interrupt-driven character-in routine.

b. De Basicbuffer kan maar 72 karakters bevatten (Comal 255 kar.). Zodra men de terminalbreedte (regel) overschrijdt en een (↑), (↓), (ESC)V of 6 geeft, of als men zich in de insert- of commandmode bevindt, ontstaat een Syntax-error na een (CR).

Oplossing 1: Overschrijdt de 80 karakters tellende VDU-regel in die gevallen niet.

Oplossing 2: Gebeurt dit toch, ga dan met (↑) naar de regel die begint met het regelnummer, geef een (ESC) --> en ga met (--) totdat het einde van de ingevoerd informatie is bereikt. Door nu (CR) in te tikken, wordt alles wel correct opgeslagen.

(NB: Men kan ook voldoende malen (ESC) -- geven, gevolgd door een (CR)).

VOORBEELD:

100 PRINT"Blabla.....bla"
lablabla.....bla"

(EOLN)

k80

DE 6502 KENNER

Ga met (↑) naar 100, geef (ESC) -->, (--) , (ESC) --> en druk op enter en alles is ok.

NB: Bij normale invoer, en de niet genoemde editcommando's hoeft men zich geen zorgen te maken.

6. SOURCELISTINGS (OP TAPE).

Load en Save hebben niets met de cassetterecorder te maken. Save bewaart de waarden van A,X,Y, en Load haalt die weer terug.

7. INSTALLATIE.

Met behulp van de sourcelisting is het in principe mogelijk om SCRED in iedere module te implementeren. In de praktijk zal blijken dat er toch steeds weer veranderingen aangebracht moeten worden, zowel in SCRED als in het programma waar men SCRED in wil bouwen.

7.1. INSTALLATIE IN KB-9 BASIC.

De volgende wijzigingen moeten in Basic aangebracht worden:

- Adres 2428 wordt JSR EDITOR 20 00 03 was 20 56 24
- " 2433 wordt CMP 08 C9 08 " C9 20
- " 243F wordt CMP 20 C9 20 " C9 5F
- " 25C8 wordt LDA 00 A9 00 " A9 20
- " 243B wordt CMP 18 C9 18 " C9 40

Het beste is om daarna de file (300,72A) te verplaatsen en te herlocaliseren vanaf adres 4261 (nieuwe editoradres). Vergeet dan niet het beginadres van vrij door Basic te gebruiken geheugen aan te passen (40CD/4147/4157/416A/417F).

7.2. INSTALLATIE IN KB-9 BASIC MET DE 'COLLECTED PATCHES' KOEN V. NIEUWENHOVE.

- Pas de Basic aan zoals beschreven onder KB-9 Basic.
- Move en relocate het geheel en voeg het toe aan de Basic volgens de beschrijving in de collected patches (ENDDAT 40CD/413D).
- Subroutine Input (43F0) dient uit het volgende statement te bestaan: 4C AE 12
- Zet de volgende data vanaf adres 41C6:
0D 0A 'Screen-editor V3.0 (C) 1985 B' 00 0D 0A 0C

NB: Autolinenumbering geeft soms problemen.

7.3. SCRED EN BASICODE.

Wijzig in het Basicodevertaalprogramma de volgende 3 bytes:
04C2 EA EA EA (was 20 34 13).

7.4. INSTALLATIE IN COMAL*KGN.

- Wijzig:
2553 20 00 03 (was 20 AE 12)
28FF 20 00 03
2C11 20 00 03
- 47E7 EA (was E8)
47E8 20 4E 04 JSR STOREBUF (was 9D 00 02)
- 3000 4C 50 07 (was 6C C8 47)

Zet dit programma vanaf 750:

750 A2 FF E8 DD 60 07 20 34 13 C9 03 D0 F5 6C C8 47
760 0A 53 43 52 45 45 4E 2D 45 44 49 54 4F 52 20 53
770 43 52 45 44 20 33 2E 30 0D 0A 28 43 29 20 31 39
780 38 35 20 42 59 20 42 2E 20 64 65 20 42 72 75 69
790 6E 65 0D 0A 0A 03

DE6502 KENNER

Handiger is het om het programma te herlocaliseren vanaf adres 4800. In dat geval moeten de geheugenplaatsen 4177/4179, 41A3/41A5 (begin van vrije ram) aangepast worden.

VERGEEET NIET OM DAN OOK JSR STOREBUF AAN TE PASSEN!!

7.5. INSTALLATE IN EEN WILLEKEURIGE MODULE.

Zoek de inoutroutine op. Hierin zit altijd een JSR RECCHA (12AE). Vervang dit door JSR EDITOR. Vindt de communicatie plaats in half-duplex, dan zal men een JSR PRCHA (1334) moeten vervangen door NOPjes (EA).

7.6. INSTALLATIE ALS STANDAARDEDITOR IN HET OPERATING SYSTEM.

Men kan de JUNIOR ook standaard met deze screeneditor uitrusten. Dan heeft men (net als bij de meeste hobbycomputers) een echte buffer. Wel moet dan iedere programmeertaal naar die buffer wijzen of er moet aan iedere taal apart een kopieroutine toegevoegd worden, die de data van de JUNIOR-buffer dan kooiert naar bijvoorbeeld de Basicbuffer. Dit is nodig, omdat Comal de buffer weer op een ander adres heeft dan Basic, ... enz. Daar ik altijd maar 1 programmeertaal tegelijk kan laden, heb ik hier vanaf gezien.

7.7. INSTALLATIE OP EEN ANDERE COMPUTER.

In het algemeen zal SCRED alleen geïmplementeerd kunnen worden op systemen met een memory-mapped VDU-systeem. Voor 'standalone'-terminals dient men heel anders te werk te gaan. Bezit men een memory-mapped VDU-computer, dan kunnen onderstaande aanwijzingen benut worden.

Men moet zelf de standaardroutines o zoeken van de betreffende computer, zoals duplex, home, clear-screen e.d. Bij Commodore-computers zijn de routines die hier als elementaire routines zijn geetiketteerd, standaard aanwezig. In plaats van de (ESC)-toets kan men bijvoorbeeld de Commodore-toets nemen.

Men dient er wel rekening mee te houden dat de char-delete-routine 'reversed' werkt.

7.8. UITBREIDINGEN.

SCRED 3.0 bevat de commando's die ikzelf het meest gebruik. Na een poosje met deze screeneditor gewerkt te hebben, kan men tot de conclusie komen dat men graag nog meer commando's wil bezitten, om nog sneller wijzigingen aan te brengen. Personen die suggesties of ideeën daarover hebben kunnen mij schrijven. Inmiddels ben ik al vergevorderd met een vernieuwde schermeditor. hierin zit een (ESC)I en (ESC)C die sneller werken, en waarmee men een 80 karakters tellende 80 karakters tellende regel zonder 'syntax-error' kan overschrijden.

Eveneens is een (ESC)P, (CNTR+N)P, en (ESC)Z routine gemaakt, zodat men nu een pointer kan zetten, en regeldelen kan deleten.

Tevens is een line-insert, line-delete en plak- en breekcommandoset toegevoegd. Er wordt nog gewerkt aan een Find /tekst/ en Change /tekst1/tekst2/ commando.

Verder is toegevoegd: Beep, (ESC)T, enz.

Voor het Find-commando is het noodzakelijk om subroutines uit de Basic-interpreter te gebruiken (o.a. om tokens op te sporen en om te zetten naar tekst en om het regelnummer te vinden in het geheugen).

Dit houdt in, dat het universele karakter van deze editor (kan in iedere taal geïmplementeerd worden) helaas verloren gaat.

Iedereen die mij inlichtingen wil geven, hoe genoemde zaken in Comal, Forth, TED/ASSM e.d. worden aangehandeld, kan mij veel werk besparen (schrijf!).

Suggesties over commando's die men onmisbaar acht zijn natuurlijk ook welkom.

Hoewel ik reeds ruim een half jaar met SCRED werk, is het niet uitgesloten dat er toch nog foutjes inzitten. Wilt U die s.v.o. (met Uw oloossing) aan de redactie melden?

B. de Bruine, O'land.

DE6502 KENNER

AS L

```

0010      LS
0020
0030      **** * * * * * * * * * * * *
0040      *   *   *   *   *   *   *   *   *
0050      *   *   *   *   *   *   *   *   *
0060      *   *   *   *   *   *   *   *   *
0070      *   *   *   *   *   *   *   *   *
0080
0090      :REALISATIE 21-1-85B
0100      :BEKNOPTE KGN-VERSIE /LAATSTE WIJZIGINGEN:070385B
0110      :EN 060785B
0120
0130      :Dit programma maakt gebruik van conditional assembly.
0140      :Indien het label BASIC of COMAL de waarde 0 heeft.
0150      :wordt voor die taal het programma gecompileerd.
0160
0170      :PROGRAMMEERTAALKEUZE (BASIC OF COMAL)
0180      :DE $00 =COMAL en negatieve waarde =BASIC
0190      :DE $FF=COMAL, 00=0, 01=BASIC
0200      :DE $00 =COMAL en negatieve waarde =BASIC
0210      :DE $FF=COMAL, 00=0, 01=BASIC
0220
0230      :PMV VIDEO ROUTINES/ADRESSEN
0240      UP      : CURSOR UP
0250      DOWN    : CURSOR DOWN
0260      RIGHT   : CURSOR RIGHT
0270      LEFT    : CURSOR LEFT
0275      ECURSOR  : VIRTUELLE CURSOR POSITIE
0280      INLINE   : CURSOR POSITION IN LINE
0290      COL     : CURSOR POSITION IN COLUMN
0300      AHOOLD  : TEMPS A.X.Y
0310      XHOLD   : DE $1A2C
0320      YHOLD   : DE $1A2B
0330      LOAD    : RESTORE A.X.Y
0340      DUPLEX  : DUPLEX-FLAG
0350      COMCOC  : RESET THE ESC-FLAG
0360      PRINT   : PRINT 1 ASCII CHAR ON SCREEN
0370      RECHA   : RECEIVE 1 ASCII CHAR FROM KEYBOARD
0380      MONITOR : RETURN TO LABEL LABJUN IN MONITOR
0390      HOCU    : HOME CURSOR AND JMP VIDEND
0400      HOME    : HOME CURSOR
0410      ERLNX   : ERASE TO END OF LINE
0420      ERSCRX  : ERASE TO END OF SCREEN
0430      RETU    : CARRIAGE RETURN
0440      FEED    : LINE FEED
0450      CURDN   : CURSOR DOWN
0460
0470      :CRTC ADRESSEN
0480      AR      : DE $DB00
0490      RFILE   : DE $DB01 : AR+1
0500
0510      :BASIC-LOCATIES
0520      IFE BASIC
0530      BUF     : DE $1B
0540      GET     : DE $2456
0550      ***
0560      :COMAL-LOCATIES
0570      :COMAL-LOCATIES
0580      IFE COMAL
0590      BUF     : DE $0200
0600      GET     : DE RECHA
0610      ***
0620      ***
0630      :LOCATIES GEBRUIKT DOOR SCRED EN BASIC
0640      TEMP   : DE $00
0650      VE     : DE $1A79
0655      CURCOP : DE $E2      : BEVAT COPIE CURSORPOSITIE
0660      :
0670      :VARIABELEN/CONSTANTEN VAN SCRED
0680      IFE BASIC
0690      BUFWIDTH : DI $48
0700      TERWIDTH : DI $48      : EIS: TERWIDTH (= BUFWIDTH)
0710      ***
0720      IFE COMAL
0730      BUFWIDTH : DI $F9

```

DE 6502 KENNER

0740 TERWIDTH	.DI \$4F		
0750	***		
0760			
0770	: A. Elementaire routines voor SCRED 3.0		
0780			
0790	.BA \$0300		
0800			
0300- 20 4D 04	0810 EDITOR	JSR HALF : HALF DUPLEX	
0303- 20 56 24	0820	JSR GET : HAAL KARAKTER OP VAN KEYBOARD	
0306- C9 1B	0830 EDIT	CMP #\$1B : ESC ?	
0308- D0 03	0840	BNE INVERSE : ZONIET, GA DOOR	
030A- 4C 91 06	0850	JMP ESCAPE : ZOJA, GA NAAR ESCAPEROUTINE	
030D- C9 0E	0860 INVERSE	CMP #\$0E : CNTR+N ?	
030F- D0 03	0870	BNE RETURN	
0311- 4C 0F 07	0880	JMP CNTRN : ZOJA, GA NAAR CONTROL/N ROUTINE	
0314- C9 0D	0890 RETURN	CMP #\$0D : <CR> ?	
0316- D0 0C	0900	BNE ALL	
0318- 20 70 03	0910 RET	JSR SAVE	
031B- 20 E7 03	0920	JSR ESC) : ESC-->	
031E- E8 0930		INX : PAS BUFFERWIJZER AAN	
031F- A9 0D 0940		LDA #\$0D : <CR> IN A	
0321- 20 70 03 0950		JSR SAVE	
0324- 20 34 13 0960	ALL	JSR PRINT : EXECUTE VIDEOCOMMANDO	
0327- 20 55 04 0970		JSR FULL : FULL-DUPLEX NOW	
032A- C9 0B	0980 DMH006	CMP #\$0B : --> ?	
032C- D0 18	0990	BNE OMLAAG	
032E- 20 70 03 1000		JSR SAVE	
0331- 20 5D 04 1010	UPSTAIRS	JSR POSCOL : POSITIE CURSOR	
0334- 20 70 03 1020		JSR SAVE	
0337- 20 5C 0F 1030		JSR UP : CURSOR 1 POS DMH006	
033A- 20 C0 16 1040		JSR LOAD	
033D- 20 22 04 1050		JSR ESC(: BEGIN V/D REGEL	
0340- 20 63 04 1060		JSR KOLON : POSITIONEER DE CURSOR IN BESPECIFIEERDE KOLOM	
0343- 4C C0 16 1070		JMP LOAD	
0346- C9 04 1080	OMLAAG	CMP #\$0A : LF ?	
0348- D0 0F 1090		BNE RECHTS	
034A- 20 70 03 1100		JSR SAVE	
034D- 20 5D 04 1110	DOWNSTAIRS	JSR POSCOL	
0350- 20 22 04 1120		JSR ESC(
0353- 20 63 04 1130		JSR KOLON	
0356- 4C C0 16 1140		JMP LOAD	
0359- C9 0C 1150	RECHTS	CMP #\$0C : --> ?	
035B- D0 1D 1160		BNE CHARIN	
035D- 20 70 03 1170		JSR SAVE	
0360- A0 00 1180	CHARUNDCUR	LDY #\$00 : HAAL KARAKTER WAAR DE CURSOR OP STAAT IN A	
0362- 20 D8 03 1190		JSR CURLOC : BEPAAL PLAATS KARAKTER IN VIDEOGEHEUGEN	
0365- B1 E2 1200		LDA (CURCOP),Y : HAAL KARAKTER IN A	
0367- BD 2D 1A 1210		STA AHOLD : EN IN AHOLD	
0368- 20 81 0F 1220		JSR RIGHT : VERPLAATS CURSOR 1 POS NAAR RECHTS	
036D- 4C C0 16 1230		JMP LOAD : GEEF A.X & Y HUN DUDE WAARDEN WEER TERUG	
0370- BD 2D 1A 1240	SAVE	STA AHOLD : BEWAAR A.X & Y	
0373- BE 2C 1A 1250		STX XHOLD	
0376- BC 2B 1A 1260		STY YHOLD	
0379- 60 1270		RTS	
037A- C9 11 1280	CHARIN	CMP #\$11 : CHAR-INSERT ?	
037C- D0 2E 1290		BNE CHRDCL	
037E- 20 70 03 1300	CHARINSERT	JSR SAVE	
0381- 18 1310		CLC	
0382- A9 4E 1320		LDA #\$4E : MAX. BEREIK 1 REGL (TERWIDTH-2)	
0384- ED 39 1A 1330		SBC COL : BEREKEN AANTAL TE VERSCHUIVEN KARAKTERS	
0387- AA 1340		TAX : X-AANTAL TE VERSCHUIVEN KARAKTERS	
0388- 20 D8 03 1350		JSR CURLOC	
0388- A0 00 1360		LDY #\$00	
038D- 20 8D 04 1370		JSR VIDPOS : BEREKEN EDLNRAMPOINTER	
0390- B1 E2 1380	INSLOOP	LDA (CURCOP),Y : KARAKTER IN A	
0392- 4B 1390		PHA	
0393- 20 98 04 1400		JSR VIDINC : POINTERS+1 A ALSET TOT 0000	
0396- 68 1410		PLA : ALSET TOT 0000	
0397- 91 E2 1420		STA (CURCOP),Y : KARAKTER IN A	
0399- 20 A9 04 1430		JSR VIDDEC : 2 POSITIES TERUG	
039C- 20 A9 04 1440		JSR VIDDEC : KARAKTER IN A	
039F- CA 1450		DEX	
03A0- 10 EE 1460		BPL INSLOOP : ALLES VERPLAATST ?	
03A2- 20 98 04 1470		JSR VIDINC : INSERT NOG EEN EXTRA SPATIE	

DE 6502 KENNER

03A5- A9 20	1480	LDA #\$20	
03A7- 91 E2	1490	STA (CURCOP),Y	
	1500	IFE COMAL	
	1510	LDA #\$00	: NIET-AFDRUKBAAR KARAKTER IN AHOLD
	1520	STA AHOLD	
	1530	***	
03A9- 4C C0 16	1540	JMP LOAD	
03AC- C9 04	1550	CMP #\$04	: CHAR-DELETE ?
03AE- D0 36	1560	BNE EXIT	
03B0- 20 70 03	1570	DELETE	JSR SAVE : ***DELETE*** VERWIJDERT KARAKTER ONDER CURSOR
03B3- A0 00	1580	LDY #\$00	
03B5- 18	1590	CLC	
03B6- A9 4F	1600	LDA #\$4F	: (TERWIDTH-1)
03B8- ED 39 1A	1610	SBC COL	
03BB- AA	1620	TAX	
03BC- 20 D8 03	1630	JSR CURLOC	
03BF- 20 98 04	1640	MOVEBACK	JSR VIDINC
03C2- B1 E2	1650	LDA (CURCOP),Y	
03C4- 48	1660	PHA	
03C5- 20 A9 04	1670	JSR VIDDEC	
03CB- 68	1680	PLA	
03C9- 91 E2	1690	STA (CURCOP),Y	
03CB- 20 98 04	1700	JSR VIDINC	
03CE- CA	1710	DEX	
03CF- 10 EE	1720	BPL MOVEBACK	
03D1- A9 20	1730	LDA #\$20	
03D3- 91 E2	1740	STA (CURCOP),Y	
	1750	IFE COMAL	
	1760	LDA #\$00	
	1770	STA AHOLD	
	1780	***	
03D5- 4C C0 16	1790	JMP LOAD	
	1800	:	
	1810	: ***CURLOC*** COPIEERT CURSOPPOSITIE NAAR	
	1811	CURSORLOCATIE \$E2/\$E3	
03D8- AD 36 1A	1820	JSR ECURSOR	: CURSOPPOSITIE IN VIDEORAM
03DB- 85 E2	1830	STA *CURCOP	: VIDEOWIJZER LO
03DD- AD 37 1A	1840	LDA ECURSOR+1	
03E0- 29 F7	1850	AND #\$F7	
03E2- 09 D0	1860	DRA #\$D0	: VIDEORAM VAN D000-D7FF
03E4- 85 E3	1870	STA *CURCOP+1	
03E6- 60	1880	RTS	
	1890	:	
	1900	: ***ESC--*** POSITIONEER CURSOR AAN EIND V/E REGEL	
03E7- A9 20	1910	ESC	LDA #\$20
03E9- 20 B2 04	1920		: BLOK 1 VERPLAATS CURSOR NAAR KOLOM 79
03EC- 20 70 03	1930	SHIFT	JSR CURSOR12 90 : GEEN CURSOR
03EF- 20 60 03	1940		JSR SAVE
03F2- 20 3C 04	1950		JSR CHARUNDCUR
03F5- AD 39 1A	1960		JSR STOREBUF
03F8- C9 47	1970		LDA COL
03FA- 30 F0	1980		CMP #TERWIDTH-1
03FC- 20 C0 16	1990		: EOLN
	2000		BMI SHIFT
03FF- 20 70 03	2010	LEFTBACK	JSR LOAD
0402- A0 00	2020		: BLOK 2 SCHUIF CURSOR NAAR LINKS.
0404- 20 D8 03	2030		JSR SAVE
0407- B1 E2	2040		: REDT A.X
0409- C9 20	2050		LDA #\$00
040B- D0 09	2060		JSR CURLOC
040D- 20 C2 0C	2070		LDA (CURCOP),Y
0410- 20 C0 16	2080		CMP #\$20
0413- CA	2090		: SPATIE ?
0414- 10 E9	2100		BNE EOLN
0416- E8	2110	EDLN	: GEEN SPATIE ? EOLN !
0417- 20 70 03	2120		JSR LEFT
041A- A9 00	2130	SKIP	: CURSOR NAAR LINKS
041C- 20 B2 04	2140		JSR LOAD
041F- 4C 60 03	2150		DEX
	2160		BPL LEFTBACK
	2170		INX
0422- A9 20	2180	ESC	: ***ESC(--*** ZET CURSOR AAN BEGIN NIEUWE REGEL
0424- 20 B2 04	2190		LDA #\$20
0427- 20 A8 0C	2200		JSR CURSOR
0428- A9 00	2210	KOLO	: CURSOR NAAR KOLOM 0
			LDA #\$00
			: CURSOR ZICHTBAAR

DE 6502 KENNER

042C- 20 B2 04	2220	JSR CURSOR						
	2230	IFE COMAL						
	2240	LDA #\$20	: ZET SPATIE					
	2250	JSR PRINT	: IN KOLOMO (VANWEGE '')' PROMPT)					
	2260	JSR LEFT	: EN WEER TERUG					
	2270	***						
042F- 20 35 04	2280	JSR BUFRES	: ZET BUFFERWIJZER OP 0					
0432- 4C C0 16	2290	JMP LOAD						
	2300		:B. SUBROUTINES VOOR CURSORBESTURING EN COMMANDODECODATIE					
	2310							
	2320							
	2330		: ***BUFRES*** RESET BUFFERPOINTER EN LAADT					
	2331		A.X.Y MET DORSPR. WAARDE					
	2340	IFE BASIC						
0435- A2 00	2350	LDX #\$00						
	2360	***						
	2370	IFE COMAL						
	2380	BUFRES	LDX #\$FF					
	2390	***						
0437- A9 00	2400	LDA #\$00						
0439- 4C 70 03	2410	JMP SAVE	: RESET A.X					
	2420							
	2430		: ***STOREBUF*** SLAAT KARAKTERS OP IN BUFFER					
	2440	IFE BASIC						
043C- E0 47	2450	CPX #BUFWIDTH-1	: BUFFER VOL ?					
043E- 10 03	2460	BPL BUFFUL	: ZOJA. NIETS MEER IN BUFFER SCHRIJVEN					
0440- 95 1B	2470	STA *BUF,X	: OPSLAAN IN BUFFER					
0442- EB	2480	INX	: X:=X+1 BUFFERWIJZER					
0443- 60	2490	RTS						
	2500	***						
	2510	IFE COMAL						
	2520	STOREBUF	CMP #\$00 : NIET AFDRUKBAAR KARAKTER ?					
	2530	BEQ INVALID						
	2540	INX						
	2550	STA BUF,X	: OPSLAG IN BUFFER					
	2560	INVALID	RTS					
	2570	***						
	2580							
	2590		: ***LEFTSHIFT*** CURSOR 1 POS NAAR LINKS. INCL. SAVE & LOAD					
0444- 20 70 03	2600	LEFTSHIFT	JSR SAVE					
0447- 20 C2 0C	2610	JSR LEFT						
0448- 4C C0 16	2620	JMP LOAD						
	2630							
	2640		: ***HALF*** ZET COMMUNICATIEKANAAL OP HALF-DUPLEX					
044D- 48	2650	HALF	PHA : RED A (COMMANDO)					
044E- A9 00	2660	LDA #\$00						
0450- 8D 41 1A	2670	STA DUPLEX						
0453- 68	2680	PLA	: HAAL COMMANDO VAN STACK					
0454- 60	2690	RTS						
	2700							
	2710		: ***FULL*** FULL-DUPLEX. AFDRUKKEN OP SCHERM					
0455- 48	2720	FULL	PHA : SAVE A					
0456- A9 01	2730	LDA #\$01						
0458- 8D 41 1A	2740	STA DUPLEX						
0458- 68	2750	PLA	: A=EDITCOMMANDO					
045C- 60	2760	RTS						
	2770							
	2780		: ***POSCOL*** ZET KOLOM WAAR CURSOR IN STAAT IN TEMP (\$00)					
045D- AD 39 1A	2790	POSCOL	LDA COL : IN WELKE KOLOM STAAT DE CURSOR ?					
0460- 85 00	2800	STA *TEMP						
0462- 60	2810	RTS						
	2820							
	2830		: ***KOLOM*** ZET CURSOR IN KOLOM GESPECIFICEERD DOOR POSCOL.					
	2840							
0463- 20 70 03	2850	KOLOM	JSR SAVE : SAVE X					
0466- AD 39 1A	2860	LDA COL	: IN WELKE KOLOM STAAT DE CURSOR ?					
0469- C5 00	2870	CMP *TEMP						
0468- F0 09	2880	BEQ STAND	: LAAT CURSOR DAAR STAAN					
046D- 20 60 03	2890	JSR CHARUNDCUR	: KARAKTER IN CURSOR + RIGHTSHIFT + LOAD					
0470- 20 3C 04	2900	JSR STOREBUF						
0473- BB	2910	CLV						
0474- 50 ED	2920	BVC KOLOM						
0476- A9 00	2930	STAND	LDA #\$00 : NIET AFDRUKBAAR KARAKTER IN A					
0478- 20 70 03	2940	JSR SAVE						

DE 6502 KENNER

047B- A9 4C	2950	RESTEMP	LDA #\$4C			047B- 8E 02	40 5A 02 -05 00
047D- 85 00	2960		STA *TEMP	: ZET WEER 4C OP \$00		047D- 8F	00 00
047F- 4C C0 16	2970		JMP LOAD			047E- 90 01	00 00
	2980			11TROB T80 1		047F- 91	00 00
	2990			11TROB T80 1		0480- 92	00 00
0482- 48	3000	CURSOR	:	***CURSOR*** PROGRAMMEERT HET FORMAAT V/D CURSOR		0481- 93	00 00
0483- A9 0A	3010		PHA	: SAVE MODE		0482- 94	00 00
0485- BD 00 DB	3020		LDA #\$0A	: ADRES VAN VDU-REG R10		0483- 95	40 5C 05 -05 00
0488- 68	3030		STA AR	: SELECTEER R10 IN DE 6845		0484- 96	00 00
0489- BD 01 DB	3040		PLA	: A=MODE		0485- 97	00 00
048C- 60	3050		STA RFILE	: MODE IN R10		0486- 98	00 00
	3060		RTS			0487- 99	00 00
	3070			:		0488- 9A	00 00
048D- 8A	3080	VIDPOS	TXA	: ***VIDPOS*** BEREKENT ABSOLUTE END.OF.LINE-1 POSITIE		0489- 9B	00 00
048E- 48	3090		PHA	: A=X=AANTAL TE VERSCHUIVEN KARAKTERS		048A- 9C	00 00
048F- 20 98 04	3100	LASTCHAR	JSR VIDINC			048B- 9D	00 00
0492- CA	3110		DEX			048C- 9E	00 00
0493- D0 FA	3120		BNE LASTCHAR			048D- 9F	00 00
0495- 68	3130		PLA			048E- A0	00 00
0496- AA	3140		TAX			048F- A1	00 00
0497- 60	3150		RTS			048G- A2	00 00
	3160			:		048H- A3	00 00
	3170			: ***VIDINC*** VERHOOGT DE VIDEOPOINTER MET 1 EN ZORGT DAT DE		048I- A4	00 00
	3180			VIDEORAMGRENZEN NIET OVERSCHREDEN WORDEN.		048J- A5	00 00
0498- E6 E2	3190	VIDINC	INC *CURCOP	: VERHOOG E2		048K- A6	74 03 -20 00
049A- A5 E2	3200		LDA *CURCOP			048L- A7	89 01 -05 00
049C- C9 00	3210		CMP #\$00			048M- A8	81 02 -05 00
049E- D0 08	3220		BNE INCIT			048N- A9	83 -5A 00
04A0- E6 E3	3230		INC *CURCOP+1			048O- A0	00 00
04A2- A5 E3	3240		LDA *CURCOP+1			048P- A1	00 00
04A4- 29 D7	3250		AND #\$D7	: VARIEERT VAN DO ... D7		048Q- A2	00 00
04A6- 85 E3	3260		STA *CURCOP+1			048R- A3	00 00
04A8- 60	3270	INCIT	RTS			048S- A4	00 00
	3280			:		048T- A5	00 00
04A9- C6 E2	3290	VIDDEC		: ***VIDDEC*** VERLAAGT DE VIDEOPOINTER MET 1		048U- A6	00 00
04AB- A5 E2	3300		DEC *CURCOP			048V- A7	00 00
04AD- C9 FF	3310		LDA *CURCOP			048W- A8	00 00
04AF- D0 OC	3320		CMP #\$FF			048X- A9	00 00
04B1- C6 E3	3330		BNE DEXIT			048Y- A0	00 00
04B3- A5 E3	3340		DEC *CURCOP+1			048Z- A1	74 03 -05 00
04B5- C9 CF	3350		LDA *CURCOP+1			048A- A2	89 01 -05 00
04B7- D0 04	3360		CMP #\$CF	: BUITEN VIDEORAM ?		048B- A3	81 00 03 -05 00
04B9- A9 D7	3370		BNE DEXIT			048C- A4	00 00
04BB- 85 E3	3380		LDA #\$D7	: BEGIN WEER BIJ D7FF		048D- A5	00 00
04BD- 60	3390		STA *CURCOP+1			048E- A6	91 14 00 -05 00
	3400	DEXIT	RTS			048F- A7	00 00
	3410			:		048G- A8	00 00
	3420			: EDIT-COMMANDO'S VOORAFGEGAAN DOOR ESC (\$1B) OF CNTR+N (\$0E)		048H- A9	00 00
	3430			: (CNTR+N) IS DE INVERSEFUNCTIE VAN (ESC)		048I- A0	00 00
	3440			:		048J- A1	00 00
	3450			: ***ESC 6.^ SCHUIFT DE CURSOR 12 REGELS OMHOOG		048K- A2	00 00
04BE- A2 0A	3460	ESC6	LDX #10			048L- A3	00 00
04C0- 8A	3470	UPSIDE	TXA			048M- A4	00 00
04C1- 48	3480		PHA			048N- A5	00 00
04C2- 20 5C 0F	3490		JSR UP	: CURSOR 1 REGL OMHOOG		048O- A6	00 00
04C5- 68	3500		PLA			048P- A7	00 00
04C6- AA	3510		TAX			048Q- A8	00 00
04C7- CA	3520		DEX			048R- A9	00 00
04C8- 10 F6	3530		BPL UPSIDE	: TOTDAT 11 POS. OMHOOG GEGAAN IS		048S- A0	00 00
04CA- 4C 31 03	3540		JMP UPSTAIRS	: EINDPOS.		048T- A1	00 00
	3550			:		048U- A2	00 00
	3560			: ***ESC V. v VERPLAATST DE CURSOR 12 POS. OMLAAG		048V- A3	00 00
04CD- A2 0B	3570	ESCV	LDX #11			048W- A4	00 00
04CF- 8A	3580	DOWNSIDE	TXA			048X- A5	00 00
04D0- 48	3590		PHA			048Y- A6	00 00
04D1- 20 D4 OC	3600		JSR DOWN	: REEDER SL.TASTER WELKOM T80 1		048Z- A7	00 00
04D4- 68	3610		PLA	: OVERFLOW STOPCONDITIE		048A- A8	00 00
04D5- AA	3620		TAX			048B- A9	00 00
04D6- CA	3630		DEX			048C- A0	00 00
04D7- 10 F6	3640		BPL DOWNSIDE	: 11 REGELS OMLAAG GEGAAN		048D- A1	00 00
04D9- 4C 4D 03	3650		JMP DOWNSTAIRS	: EINDPOSITIE		048E- A2	00 00
	3660			:		048F- A3	00 00
	3670			: ***ESC A *** WIJZIGT ALLE KLEINE LETTERS IN HOOFDLETTERS		048G- A4	00 00
04DC- 20 70 03	3680	ESCA	JSR SAVE			048H- A5	00 00

DE 6502 KENNER

04DF- 20 60 03 3690	JSR CHARUNDCUR	: HAAL KARAKTER ONDER CURSOR IN A
04E2- C9 61 3700	CMP #'a'	: FILTER KLEINE LETTERS
04E4- 30 06 3710	BMI UNCHANGED	
04E6- C9 7B 3720	CMP #\$7B	: ASCII 61 T/M 7A ZIJN KLEINE LETTERS
04E8- 10 02 3730	BPL UNCHANGED	
04EA- 29 DF 3740	AND #\$0F	: KLEINE LETTER WORDT HOOFDLETTER (BS=0)
04EC- 20 3C 04 3750	UNCHANGED	JSR STOREBUF
04EF- 20 44 04 3760	JSR LEFTSHIFT	: VERWIJDER OORSPR. KARAKTER
04F2- 20 34 13 3770	JSR PRINT	: PRINT NIEUWE KARAKTER
04F5- AD 39 1A 3780	LDA COL	
04FB- C9 46 3790	CMP #TERWIDTH-2	; TOT EDLN
04FA- D0 E0 3800	BNE ESCA	
04FC- 4C E7 03 3810	JMP ESC	
3820		
3830		: ***ESC a *** VERVANGT ALLE KLEINE LETTERS DOOR HOOFDLETTERS
3840		TUSSEN DE CURSOR EN HET ENDE VAN DE REGEL
04FF- 20 70 03 3850	ESCa	JSR SAVE
0502- 20 60 03 3860		JSR CHARUNDCUR
0505- C9 41 3870	CMP #'A'	: SELECTEER HOOFDLETTER (A..Z = 41 T/M 5A ASCII)
0507- 30 06 3880	BMI SAME	
0509- C9 5B 3890	CMP #\$5B	
050B- 10 02 3900	BPL SAME	
050D- 09 20 3910	ORA #\$20	: KLEINE LETTERS WORDEN HOOFDLETTERS (BS=1)
050F- 20 3C 04 3920	SAME	JSR STOREBUF
0512- 20 44 04 3930	JSR LEFTSHIFT	
0515- 20 34 13 3940	JSR PRINT	
0518- AD 39 1A 3950	LDA COL	
051B- C9 47 3960	CMP #TERWIDTH-1	
051D- D0 E0 3970	BNE ESCA	; TOT EDLN
051F- 4C E7 03 3980	JMP ESC	
3990		
4000		: ***ESC SPATIEBALK *** INSERT 20 SPATIES
0522- A9 14 4010	ESCSP	LDA #20
0524- 85 00 4020	SPACE	STA *TEMP
0526- 20 7E 03 4030		JSR CHARINSERT
0529- A5 00 4040		LDA *TEMP
052B- E9 01 4050	SBC #\$01	: 20 X VERLAGEN
052D- D0 F5 4060	BNE SPACE	
052F- 4C 7B 04 4070	JMP RESTEMP	; OORSPR. WAARDE TERUG IN TEMP
4080		
4090		: ***CNTR+N SPATIEBALK*** VERWIJDERT SPATIES TOT
4100		EERSTE NIET-SPATIE KARAKTER
0532- A9 48 4110	CNTRNSP	LDA #TERWIDTH
0534- 85 00 4120		STA *TEMP
0536- 20 70 03 4130	SPACEDEL	JSR SAVE
0539- 20 60 03 4140		JSR CHARUNDCUR
053C- C9 20 4150	CMP #\$20	: SPATIE ?
053E- D0 12 4160	BNE NOSPACE	: ZONEE, STOPPEN
0540- 20 44 04 4170	JSR LEFTSHIFT	
0543- 20 B0 03 4180	JSR DELETE	: VERWIJDER DIE SPATIE
0546- A5 00 4190	LDA *TEMP	
0548- E9 01 4200	SBC #\$01	
054A- 85 00 4210	STA *TEMP	
054C- F0 03 4220	BEQ CANTCONT	: OVERFLOW STOPCONDITIE
054E- 4C 36 05 4230	JMP SPACEDEL	: ELIMINEER NOG MEER SPATIES
0551- 18 4240	CLC	
0552- 4C 7B 04 4250	JMP RESTEMP	
4260		
4270		: ***ESC I *** SCHAKELT OVER VAN EDIT/WINDOWMODE NAAR DE INSERT
0555- AD 38 1A 4280	ESCI	LDA INLINE
0558- C9 02 4290		: WELKE REGEL ?
055A- 10 03 4300	CMP #\$02	: REGEL > 1 ?
055C- 20 31 06 4310	BPL DIRECT	: ZOJA. DIRECTE TOEGANG TOT INSERTMODE
055F- 20 5D 04 4320	JSR TOOHIGH	
0562- 20 2A 06 4330	JSR POSCOL	
0565- 20 D8 05 4340	JSR POSLIN	
0568- 89 FF 4350	JSR WINDOMMES	
056A- 20 E5 05 4360	LDA #\$FF	: INSERTMODE ALERT
056D- 20 1B 06 4370	JSR MESSAGE	
0570- 20 22 04 4380	JSR REGEI	: CURSOR TERUG OP DE JUISTE REGEL
0573- 20 63 04 4390	JSR ESC	
0576- 20 C0 16 4400	JSR KOLOM	: EN IN DE JUISTE KOLOM
0579- 20 4D 04 4410	JSR LOAD	
057C- 20 70 03 4420	JSR HALF	
	JSR SAVE	

DE 6502 KENNER

057F- 20 AE 12 4430	JSR RECHA	RECHA TOT RETURNCODE	PLA	RTS	RTS
0582- C9 0D 4440	CMP #\$0D	: (CR) ?	RTS	RTS	RTS
0584- F0 2B 4450	BEQ EDWIST	: TERUG NAAR ED/WI MODE. EN SLA REGEL OP IN GEHEUGEN	RTS	RTS	RTS
0586- C9 OC 4460	CMP #\$0C	: --)	RTS	RTS	RTS
0588- F0 2D 4470	BEQ CFDR		RTS	RTS	RTS
058A- C9 08 4480	CMP #\$08	: (-- ?	RTS	RTS	RTS
058C- D0 0A 4490	BNE EDKEY		RTS	RTS	RTS
058E- 20 34 13 4500	JSR PRINT	: EXECUTEER BACKSPACE	RTS	RTS	RTS
0591- 20 B0 03 4510	JSR DELETE	: EN TERUG MET SCHUIFDEEL (D.I. CURSOR T/M EDLN)	RTS	RTS	RTS
0594- CA 4520	DEX		RTS	RTS	RTS
0595- B8 4530	CLV		RTS	RTS	RTS
0596- 50 E4 4540	BVC NCHAR		RTS	RTS	RTS
0598- C9 20 4550 EDKEY	CMP #\$20		RTS	RTS	RTS
059A- 10 09 4560	BPL OVFLCHECK	: ASCII (20 IS EDITCOMMANDO	RTS	RTS	RTS
059C- 20 06 03 4570	JSR EDIT	: ZOJA, VOER EDITCOMMANDO UIT	RTS	RTS	RTS
059F- 20 4D 04 4580	JSR HALF		RTS	RTS	RTS
05A2- 4C 7C 05 4590	JMP NCHAR	: TERUG NAAR INPUT	RTS	RTS	RTS
05A3- 20 7E 03 4600 OVFLCHECK	JSR CHARINSERT		RTS	RTS	RTS
05A8- 20 3C 04 4610	JSR STOREBUF	: SLA OP IN BUFFER	RTS	RTS	RTS
05AB- 20 34 13 4620	JSR PRINT	: EN ZET OP SCHERM	RTS	RTS	RTS
05AE- 4C 7C 05 4630	JSR NCHAR		RTS	RTS	RTS
05B1- 20 BD 05 4640 EDWIST	JSR NOMENO		RTS	RTS	RTS
05B4- 4C 18 03 4650	JMP RET	: VOER RETURN UIT	RTS	RTS	RTS
05B7- 20 B0 05 4652 CFOR	JSR NOMENO	: CURSOR TERUG NAAR BEW. REGEL	RTS	RTS	RTS
05B8- 4C 60 03 4654	JMP CHARUNDCUR	: CURSOR 1 POSITIE NAAR RECHTS	RTS	RTS	RTS
05BD- 20 5D 04 4660 NOMENO	JSR POSCOL	: NORMAL-MESSAGE-NORMAL	RTS	RTS	RTS
05C0- 20 2B 06 4670	JSR POSLIN	: WELKE REGEL ?	RTS	RTS	RTS
05C3- 20 70 03 4680	JSR SAVE		RTS	RTS	RTS
05C6- 20 D8 05 4690	JSR WINDOWMES		RTS	RTS	RTS
05C9- 20 E0 0C 4700	JSR ERLNX	: VEEG DE EERSTE 2 REGELS V/H SCHERM SCHOON	RTS	RTS	RTS
05CC- 20 1B 06 4710	JSR REGEL	: ZET CURSOR TERUG OP LAATST BEWERKTE REGEL	RTS	RTS	RTS
05CF- 20 22 04 4720	JSR ESC(: BUFFERRESET + NWLN	RTS	RTS	RTS
05D2- 20 63 04 4730	JSR KOLOM	: ZET CURSOR IN DE JUISTE KOLOM	RTS	RTS	RTS
05D5- 4C C0 16 4740	JMP LOAD		RTS	RTS	RTS
05D6- 4750	:		RTS	RTS	RTS
05D8- 20 C8 OC 4770 WINDOWMES	JSR HOCU	: ***WINDOWMES*** VEEGT BOVENSTE REGEL V/H SCHERM SCHOON +CRLF	RTS	RTS	RTS
05DB- 20 E0 OC 4780	JSR ERLNX	: HOME CURSOR	RTS	RTS	RTS
05DE- 20 A8 OC 4790	JSR RETU	: LINE ERASE	RTS	RTS	RTS
05E1- 20 B3 OC 4800	JSR FEED	: CURSOR NAAR KO (KO=KOLOM 0)	RTS	RTS	RTS
05E4- 60 4810	RTS	: NIEUWE REGEL	RTS	RTS	RTS
05E5- 48 4820	:		RTS	RTS	RTS
05E6- A2 48 4830	JSR PRINT	: ***MESSAGE*** PRINT MEDEDELING EN ONDERSTREEP DIE	RTS	RTS	RTS
05E8- A9 87 4840	JSR FEED	: A=FF INSERTMODE	RTS	RTS	RTS
05E9- 4850	JSR FEED	: A=OB GIVE COMMAND:	RTS	RTS	RTS
05E5- 48 4860 MESSAGE	PHA	: RED MEDEDELINGAANWIJZER	RTS	RTS	RTS
05E6- A2 48 4870	LDX #TERWIDTH	: ONDERSTREEP DE MEDEDELING	RTS	RTS	RTS
05E8- A9 87 4880	LDA #\$07	: ONDERSTREEP-GRAFIC IN A	RTS	RTS	RTS
05EA- 20 34 13 4890 UNDERSCORE	JSR PRINT		RTS	RTS	RTS
05ED- CA 4900	DEX		RTS	RTS	RTS
05EE- D0 FA 4910	BNE UNDERSCORE		RTS	RTS	RTS
05F0- 20 C8 OC 4920	JSR HOCU		RTS	RTS	RTS
05F3- 6B 4930	PLA	: HAAL STRINGINDEX OP	RTS	RTS	RTS
05F4- AA 4940	TAX		RTS	RTS	RTS
05F5- EB 4950 STRING	INX		RTS	RTS	RTS
05F6- BD 01 06 4960	LDA TEXT,X		RTS	RTS	RTS
05F9- 20 34 13 4970	JSR PRINT		RTS	RTS	RTS
05FC- C9 03 4980	CMP #\$03		RTS	RTS	RTS
05FE- D0 F5 4990	BNE STRING		RTS	RTS	RTS
0600- 60 5000	RTS		RTS	RTS	RTS
0601- 5010	:		RTS	RTS	RTS
0602- 5020	JSR FEED	: TEKSTSTRINGS	RTS	RTS	RTS
0603- 49 6E 73 5030 TEXT	.BY 'Insert mode' \$03		RTS	RTS	RTS
0604- 65 72 74	JSR FEED	: BY 'GIVE COMMAND:' \$03	RTS	RTS	RTS
0607- 20 6D 6F	JSR FEED		RTS	RTS	RTS
0608- 64 65 03	JSR FEED		RTS	RTS	RTS
0609- 47 49 56 5040	JSR FEED	: BY 'GIVE COMMAND:' \$03	RTS	RTS	RTS
0610- 45 20 43	JSR FEED		RTS	RTS	RTS
0613- 4F 4D 4D	JSR FEED		RTS	RTS	RTS
0616- 41 4E 44	JSR FEED		RTS	RTS	RTS
0619- 3A 03 5050	JSR FEED	: ***REGEL*** POSITIONEERT DE CURSOR OP DE REGEL,	RTS	RTS	RTS
0620- 5060	JSR FEED		RTS	RTS	RTS

DE 6502 KENNER

0618- 20 87 OF	5080 REGEL	JSR HOME	GESPECIFICEERD DOOR VE. (VErticale positie)
061E- CE 79 1A	5090 DECRULE	DEC VE	; VERTIKALE REGEL -1
0621- 30 06	5100	BMI ORIGIN	; OMLAAG T.O.V HOMEPOSITIE
0623- 20 38 OF	5110	JSR CURDN	
0626- 18	5120	CLC	
0627- 90 F5	5130	BCC DECRULE	
0629- 60	5140 ORIGIN	RTS	
	5150		
	5160		; ***POSLIN*** BEWAART POSITIE CURATIEVE REGEL IN VE
062A- AD 38 1A	5170 POSLIN	LDA INLINE	
062B- 8D 79 1A	5180	STA VE	
0630- 60	5190	RTS	
	5200		
	5210		; ***TOOHIGH*** GENEREERT 2*OMHOOG+2*OMLAAG
0631- 20 5C OF	5220 TOOHIGH	JSR UP	
0634- 20 5C OF	5230	JSR UP	
0637- 20 B3 OC	5240	JSR FEED	
063A- 20 B3 OC	5250	JSR FEED	
063D- 60	5260	RTS	
	5270		
	5280		; ***ESC C*** VERLEENT TOEGANG TOT DE COMMAND-MODE
063E- AD 38 1A	5290 ESCC	LDA INLINE	; STAAT CURSOR OP REGEL 1 OF 2 ?
0641- C9 02	5300	CMP #\$02	
0643- 10 03	5310	BPL ENTERCM	; ZOJA, DIRECTE TOEGANG TOT COMMANDMODE
0645- 20 31 06	5320	JSR TOOHIGH	; VERSCHUIF (SCROLL) BEELDSCHERM 2 REGELS
0648- 20 5D 04	5330 ENTERCM	JSR POSCOL	
064B- 20 2A 06	5340	JSR POSLIN	
064E- 20 D8 05	5350	JSR WINDOWMES	
0651- A9 0B	5360	LDA #\$0B	: GIVE COMMAND
0653- 20 E5 05	5370	JSR MESSAGE	
0656- 20 35 04	5380	JSR BUFRS	
0659- 20 C0 16	5390	JSR LOAD	
065C- 20 00 03	5400 INPUT	JSR EDITOR	: INPUT COMMANDMODE
	5410	IFE BASIC	
065F- C9 54	5420 TERMINAL	CMP #'T'	: TERMINAL WIDTH = 80 (\$50)
0661- D0 04	5430	BNE QUIT	
0663- A9 50	5440	LDA #\$50	; TERMINALBREEDTE = 80 KARAKTERS
0665- 85 17	5450	STA *\$17	
	5460	***	
0667- C9 51	5470 QUIT	CMP #'Q'	; NAAR JUNIOR-MONITOR
0669- D0 09	5480	BNE DISAS	
066B- 20 E6 0C	5490	JSR ERSCRX	; 'SCREEN-ERASE'
066E- 20 7B 04	5500	JSR RESTEMP	; ZET OORSPRONKELIJKE WAARDE IN TEMP
0671- 4C 5F 10	5510	JMP MONITOR	
0674- C9 44	5520 DISAS	CMP #'D'	; GA NAAR DISASSEMBLER
0676- D0 09	5530	BNE WIMO	
0678- 20 E6 0C	5540	JSR ERSCRX	
0678- 20 7B 04	5550	JSR RESTEMP	
067E- 4C 4E FC	5560	JMP \$FC4E	; DISASM/ASCII-DUMP
0681- C9 0D	5570 WIMO	CMP #\$0D	; (CR)
0683- D0 D7	5580	BNE INPUT	
0685- 20 1B 06	5590	JSR REGEL	
0688- 20 22 04	5600	JSR ESC()	
068B- 20 63 04	5610	JSR KOLOM	
068E- 4C C0 16	5620	JMP LOAD	
	5630		
	5640		
	5650		; ***ESCAPE ROUTINE*** DECODEEREN VAN KOMMANDO'S
0691- 20 34 13	5660 ESCAPE	JSR PRINT	; ZET DE ESC-VLAG
0694- 20 70 03	5670	JSR SAVE	
0697- 20 AE 12	5680	JSR RECHA	; KARAKTER VAN TOETSENBORD
069A- 20 55 04	5690	JSR FULL	; FULL-DUPLEX
069D- C9 30	5700	CMP #'0'	; >=0 ? ASCII-CIJFER 0...9 ?
069F- 30 23	5710	BMI RESESC	; ZONEE, RESET DE ESCAPEVLAG
06A1- C9 36	5720	CMP #'6'	; (ESC)6 IS EEN SCREDCOMMANDO
06A3- F0 1F	5730	BEQ RESESC	
06A5- C9 39	5740	CMP #'9'	; (>=9 ?) = 9 IS EEN SCREDCOMMANDO
06A7- 10 1B	5750	BPL RESESC	
06A9- 20 34 13	5760	JSR PRINT	; VOER HET VIDEOCOMMANDO UIT
06AC- C9 34 13	5770	CMP #'4'	; REGEL OF SCHERMCOMMANDO ?
06AE- 10 06	5780	BPL CORR	
06B0- 20 35 04	5790	JSR BUFRS	; EN BUFFER AAN BEGIN NIEUWE REGEL
06B3- 4C C0 16	5800	JMP LOAD	

DE 65 02 KENNER

```

06B6- 20 60 03 5810 CORR      JSR CHARUNDCUR : /CORRECTIE OP PMV VIDEOCOMMANDO'S
06B9- 20 44 04 5820      JSR LEFTSHIFT ; VERWIJDER PMV-CIJFER, LEES BEELDSCHERM
06BC- A9 00 5830      LDA #$00
06BE- 8D 2D 1A 5840      STA AHOLD
06C1- 4C C0 16 5850      JMP LOAD
06C4- 48 5860 RESESC      PHA
06C5- 20 13 15 5870      JSR COMCOC : RESET DE VIDEOHANDLERS ESC-VLAG
06C8- 68 5880      PLA
06C9- C9 0C 5890      CMP #$0C : --?
06CB- D0 03 5900      BNE NEXT
06CD- 4C E7 03 5910      JMP ESC)
06D0- C9 08 5920 NEXT     CMP #$08 : (-- ?
06D2- D0 03 5930      BNE NEXT1
06D4- 4C 22 04 5940      JMP ESC(
06D7- C9 41 5950 NEXT1     CMP #'A' : A ?
06D9- D0 03 5960      BNE NEXT2
06DB- 4C DC 04 5970      JMP ESCA
06DE- C9 61 5980 NEXT2     CMP #'a' : a ?
06E0- D0 03 5990      BNE NEXT3
06E2- 4C FF 04 6000      JMP ESCa
06E5- C9 20 6010 NEXT3     CMP #$20 : SPATIE ?
06E7- D0 03 6020      BNE NEXT4
06E9- 4C 22 05 6030      JMP ESCSP
06EC- C9 5E 6040 NEXT4     CMP #$5E : --?
06EE- D0 03 6050      BNE NEXT5
06F0- 4C BE 04 6060 DEL6     JMP ESC6
06F3- C9 36 6070 NEXT5     CMP #'6' : 6 ?
06F5- F0 F9 6080      BEQ DEL6
06F7- 29 DF 6090      AND #$DF : kleine letter worden HOOFDLETTERS
06F9- C9 43 6100      CMP #'C' : C.c ?
06FB- D0 03 6110      BNE NEXT6
06FD- 4C 3E 06 6120      JMP ESCC
0700- C9 56 6130 NEXT6     CMP #'V' : V.v ?
0702- D0 03 6140      BNE NEXT7
0704- 4C CD 04 6150      JMP ESCV
0707- C9 49 6160 NEXT7     CMP #'I' : I.i ?
0709- D0 03 6170      BNE NEXT8
070B- 4C 55 05 6180      JMP ESCI
070E- 60 6190 NEXT8      RTS
0720- : +++CNTR+N COMMANDO-DEKODER+++
070F- 20 70 03 6220 CNTRN     JSR SAVE
0712- 20 4D 04 6230      JSR HALF
0715- 20 AE 12 6240      JSR RECHA : KOMMANDO IN A
0718- 20 55 04 6250      JSR FULL
071B- C9 20 6260      CMP #$20 : SPATIE ?
071D- D0 03 6270      BNE COMMAND2
071F- 4C 32 05 6280      JMP CNTRNSP
0722- 60 6290 COMMAND2      RTS
0730- : ALL COMMANDS ARE SHED-COMPATIBLE
0731- : PRINCIPE EL. ROUT /GOFFAU/
0732- : COPYRIGHT ALL ROUTINES BY B. DE BRUINE O'LAND
0733- : BASED ON THE UNIVERSITY OF SHEFFIELD EDITOR V3.2.3
0734- : MADE IN JANUARY 1985
0735- : (C)
0736- : //
0737- : //
```

//0000,0723,0723
DC PR#0

ACORN MON65

The MON65/DOS65 software and controllercard developed for Elektor's CPU/VDU-system by our two members of the board, Erwin Visschedijk and Adri Hankel, already runs by many of us. Now Erwin Visschedijk implemented MON65 on the ACORN ATOM computer, and wrote a manual of 34 pages. If you also want to implement a sophisticated monitor on your ACORN ATOM, then call Mr. Erwin Visschedijk, Drakensteyn 299, 7608 TR Almelo, The Netherlands, for more informations. We allready published about the MON65/DOS65 system in our earlier editions.

GRAPHICS IN APPLESOFT

By H. Bosch
Twente University of Technology
Postbox 217
7500 AE Enschede
The Netherlands

INTRODUCTION

It is often very useful to present the results of experiments or calculations in graphical form to assure readability. Increasing numbers of microcomputers are used in small scale automation projects where the need for 'easy-to-use' graphics is also evident.

In this paper, a machine language subroutine which carries out all the necessary functions to prepare a graph is described: scaling and plotting of the axes; and plotting of the functions to be represented. The text can be put anywhere. This subroutine may be called for either in direct mode or from the user's Applesoft program by using a statement which starts with the ampersand & command followed by a reserved word, the token of it specifying the particular action to be taken. The keywords allowable so far are &HLIN, &VLIN, &PLOT and &DRAW. The syntax and use of each of these commands will be described separately.

The extensive use of the monitor, floating point and graphic subroutines already present in the Apple's ROM made it possible to develop a fast and compact subroutine which occupies only about 1.5 k bytes of RAM space, and includes a shape table with a limited number of 32 characters for numerical notation and plotting purposes. Once the machine language subroutine has been implemented, any existing Applesoft programs can be extended very easily to give a graphical representation just by adding two or more of the &-commands mentioned above.

STRUCTURE OF THE PROGRAM

The complete machine language program is given in figure 1. Its main details will be discussed successively.

The & command causes an unconditional jump to an address pointed to by a vector in address 83F5. This ampersand vector is set in line #63 by BRUNning the program just once. This initialisation causes also the HIgh MEMemory pointer to be set to the start of the program and the pointer in 8E8 to be set to the end of the program where the shape table starts.

The ampersand command has to be followed by at least one option, which causes a subsequent jump to the appropriate part of the program, depending on the token for the keyword used. The program comprises of the following main subroutines.

&HLIN (&VLIN) causes a horizontal (vertical) axis to be scaled and drawn and supplied with a numeral value at each division. This part starts at line # 527.

&PLOT causes a particular symbol (one out of five possible) to be drawn at a given coordinate related to the axes already scaled and starts at line # 432.

&DRAW causes the desired text to be put at any place in the HIgh RESolution screen and starts at line # 468.

The labelling of the entry points of the monitor, floating point and graphic subroutines present in the Apple's ROM has been done according to W.F. Luebbert's "What's Where in the Apple?", MICRO INK inc., 1981. The wrong use of keywords and/or options results in the SYNTAX ERROR through line # 524. All other possible errors are dealt with by Apple's Error Handler.

After running the program you will be asked if you want to draw a picture. If you answer 'Y' or 'y' the program will start drawing.

'Graphics' is written in assembly language (Apple's Tool Kit) and can be run as is on 48K systems. To run on other systems the program should be re-ORGANIZED at \$4000, just above HIgh RESolution page 1.

The graphics calculator allows it over 100 x 80 pixels. It uses the Apple Basic SYNTAX OF THE HLIN STATEMENT

The syntax of the HLIN statement is

$\&HLIN <aexpr1, aexpr2>$ *
with $aexpr1 < aexpr2$ and $ABS(aexpr2) \geq .03$

* description following appendix N of the Applesoft Basic Programming Reference Manual.

Violation of the first limitation results in SYNTAX ERROR. The use of smaller values, as indicated by the second boundary equation, results only in insignificant amounts along the axes: due to a rounding-off procedure, the smallest increase amounts to .01. When smaller values are encountered, a suitable power of 10 will be effective.

A suitable number of divisions is calculated by the scaling routine in line ~~#~~ 327. Three parameters, J, SP and H, are chosen in such a way that the range of aexpr2-aexpr1 will cover at least 75% of the axis. J can amount to 1,2,4, or 5, SP ranges from 4 to 8, H being a power of 10. At a suitable combination of J, SP and H, the lowest value on the axes is calculated as $L_0 = J * H * \text{INT}(aexpr1/J/H)$. The highest value is then given by $H_1 = L_0 + J * SP * H$, SP being the number of parts in which the axis is divided. The method applied to find a suitable set of J,SP and H is not very sophisticated but is basically a matter of trial and error. An outline is given in figure 2. This method works well if $\text{SGN}(aexpr1) = \text{SGN}(aexpr2)$. If this is not true and if $\text{ABS}(aexpr1) << \text{ABS}(aexpr2)$, then the 75% condition might not be met, except for relatively high values of SP. That is why the maximum value of SP is held at 8; higher values result in poor readability.

The axes are drawn through DRAWAX (line ~~#~~ 576 in fig.1) and the relevant values are plotted along the axes (e.g. in figure 3) by SUBSCR (line ~~#~~ 240) and DRWSHP (line ~~#~~ 275). The parameter COUNT detects the end of the axis to put the highest value and the final transverse line there.

The use of &VLIN $aexpr1, aexpr2$ for drawing a vertical axis follows analogously. The coordinates of the origin are determined by the data in line ~~#~~ 85 and 86. The values can be changed by appropriate POKE's to the relevant addresses before one of the ampersand commands has been used. The same applies to the length of the axes as given in line ~~#~~ 83 and 84. The actual length depends on the number of scale parts and is accessible from the Applesoft program through PEEK's to 2428(x-axis) or 2429(y-axis). Once both a horizontal and a vertical axis have been drawn, an axis parallel to one and/or the other axis may be drawn by the statement &VLIN and/or &HLIN. An example is given in figure 4.

The user must set SCALE to 1 before applying &HLIN and &VLIN to ensure that the subscripts are drawn at the right size. This condition is not set in the program because other subroutines also make use of SHADDR (line ~~#~~ 298) with possible other value of SCALE.

SYNTAX OF THE DRAW STATEMENT

The DRAW statement causes text to be drawn on the HIGH RESOLUTION graphic page.

The syntax of the DRAW statement is

&DRAW String\$, X|Y|aexpr1, aexpr2

The option X (Y) puts the string right along the X (Y) axis. If neither X or Y is present, coordinates will be assumed which serve as the starting point for the first character in the string.

Applesoft does not allow one to float a string from the floating point into the string buffer (\$200 .. \$2FF) more than twice, otherwise the EXPRESSION TOO COMPLEX error results. To overcome this limitation, the appropriate error flag is reset in line # 510 each time the string buffer is used. This does not effect subsequent statements. In the example given in figure 5, SCALE was set to 1; other values may be used.

The PLOT statement is used to PLOT a symbol at any position in the High Resolution page. The syntax of the PLOT statement is

&PLOT aexpr1, aexpr2, 0|1|2|3|4
The first two values give the coordinates of the symbol to be plotted relative to the axes already scaled, while the last value serves as a plot code for the symbol to be used. Five choices are available. An application is given in figure 6a.

The plot-codes 2,3 and 4 are all based on the same shape # 30 (see next paragraph). The top line of the graph in figure 9 shows the result of using SCALE = 2. There is also the possibility of using the normal HGR commands as e.g. HPLOT in relation with the axes used. This needs the conversion from scaled coordinates into HGR coordinates. Assume a value H has to be expressed in a number of dots:

$H = H_0 - (H-L_0) * L / (H_1-L_0)$ where H_0 and H_1 mark the axis limits and L is the length of the axis.

H_0 = coordinates of origin in dots

L = length of axis in dots

L_0 = lowest value

H_1 = highest value

L_0 and $L / (H_1-L_0)$ are accessible and their 5 bits addresses may be read from lines # 188 and 189 (figure 1). For the Y-axis, the last value is stored with an opposite sign to satisfy the condition that the origin of the HGR-screen is in the upper left corner. Thus the equation above applies for both X- and Y-axes, see figure 6b for details. The values of the scale length may be PEEKed from address 788 and 789; if desired, see line # 46.

THE SHAPE TABLE

Deze sectie beschrijft de wijze van de vormtafel om een teken te maken.

A shape-table consists of a number of plot-codes defining one or more shapes.

The building of a shape-table and its use in Applesoft programs is well documented in the Applesoft manual. Thus, a shape-table has been constructed in such a way that, for the digits and letters, the end-point of the HRG-cursor coincides with the starting-point of the next symbol which will eventually be drawn. This makes the use of the shape-table from a machine language sub-routine very simple. Once the graphic cursor has been positioned by HPOSN (line # 271), subsequent DRAW statements (line # 309, called from DRWSHP in line # 275) result in a one-line text (the direction of it determined by the ROTation) without any further positioning of the cursor being necessary.

The construction of a shape-table is rather time consuming, especially when the shapes must hold their form at any value of the SCALE factor. That is why only a limited number (28) of ASCII characters are present in the shape-table as given in figure 7. To identify the characters present in the shape-table, their ASCII codes are given at the end of the table. The actual choice of characters has been made with respect to the Dutch language, the American user will probably miss e.g. the characters and in some special case also the characters and in x and y. The last few shapes are meant for plotting purposes and do not have an ASCII code.

The shape-number of all available symbols may be read from figure 8. Shape # 27 moves the cursor one place to the right at ROT = 0 (the corresponding ASCII code being 20), whereas the last symbol moves the cursor in the opposite direction. This last feature is used to create an INVERSE mode, as shown in figure 9, line # 420. The use of shapes # 29 and 30 does not change the cursor position.

The machine language program (figure 1) starts at \$ 9055 and has a length of \$ 431 hex.bytes. After assembling, the OBJect file is loaded. Enter the monitor by typing CALL-151.

After the appearance of the asterisk, the shape table (figure 7) can be entered into memory right behind the program, starting at \$ 9486. Consult page 44 of your Apple Reference Manual for details. Now BSAVE the program, including shape table and ASCII codes of symbols present, by

BSAVE name.OBJ, A\$9055,L\$5AB

From now on, the program and the tables are fixed together, the starting address of the latter determined by the address of the last label in the program, increased by one. Initialisation is carried out by the statement
`BRUN name.OBJ` either in direct or deferred mode. Adding this line to your HELLO-program results in automatic initialisation each time DOS is booted. Also the HIGH MEMORY pointer is set below the program to protect this part of RAM space. Finally, it should be remarked that the total length of 1451 bytes is small enough to store the program in an expansion (E)PROM on a peripheral card. Page 84 of the Apple's Reference Manual shows how to enable that memory space.

DEMONSTRATION OF THE FEATURES

A complete Applesoft program, including all the features described above, is given in figure 9. It consists of two parts. In the first part, a damped sinusoidal curve is calculated, its form being determined in line # 170 by the actual value returned by the RANDOM generator. The coordinates of the points to be plotted are printed on the screen by means of a simple formatting subroutine and stored in the two arrays given. In line # 250-260, the lowest and highest value of the ordinate are traced. The values along the X-axes range from zero to `XH`(line # 300).

These values are used in the second part to scale the graph (line # 330). The Y-axes is extended to show all possible plot codes in parallel curves (line # 370). The top curve in figure 9 is plotted with code 0, the bottom with code 4.

Lines # 390-410 are self-explanatory, but the use of shape 31 in line # 420 may need some amplification. The High GRaphic cursor starts where the next symbol would be DRAWn and ends 3 positions to the left on the same line. Thus, the cursor moves backwards, the number of subsequent DRAWings depending on the area to be covered. The use of XDRAW in stead of DRAW results in the INVERSE mode. Finally, line # 440 shows that some plot symbols are suitable to be enlarged.

Now you can try for yourself the examples given together with others, according to your own needs and interests.

Figure 1 Program listing

```

SOURCE FILE: GRAPHICS
0000:      1 ; BY H. BOSCH,
0000:      2 ; TWENTE UNIVERSITY OF TECHNOLOGY, DEPT OF CHEM. TECHN.,
0000:      3 ; POSTBOX 217, ENSCHEDE, THE NETHERLANDS
0000:      4 ;
0000:      5 ; MONITOR AND FLOATING POINT SUBROUTINES
FF4A:      6 IOSAVE EQU $FF4A    SAVE REGISTERS
FF3F:      7 IOREST EQU $FF3F    RESTORE REGISTERS
00B1:      8 ADVTP EQU $00B1    ADVANCE TEXTPOINTER
EB21:      9 MOV1F EQU $EB21    PACK FAC INTO TEMP1
EB1E:     10 MOV2F EQU $EB1E    PACK FAC INTO TEMP2
EB23:     11 MOVML EQU $EB23    PACK FAC INTO MEM(X), ZEROPAGE
EB2B:     12 MOVMF EQU $EB2B    PACK FAC INTO MEM(Y,X)
E5E2:     13 MOVSTR EQU $E5E2   MOVE STRING INTO $BUFFER
ED34:     14 FOUT EQU $ED34    VALUE(FAC) INTO STRINGBUFFER
E6FB:     15 CONINT EQU $E6FB   FAC INTO X REG.
EB93:     16 FLOAT EQU $EB93   INT(ACC)->FAC
EAF9:     17 MOVFM EQU $EAF9   MEM(Y,A)->FAC
EC23:     18 FPINT EQU $EC23   INT(FAC)->MANTISSA
E7BE:     19 FADD EQU $E7BE   MEM(Y,A)+FAC->FAC
E7A0:     20 FADHD EQU $E7A0   FAC + .5
EA66:     21 FDIV EQU $EA66   MEM(Y,A)/FAC->FAC
E97F:     22 FMULT EQU $E97F   MEM(Y,A)*FAC->FAC
E7A7:     23 FSUB EQU $E7A7   MEM(Y,A)-FAC->FAC
EA39:     24 MUL10 EQU $EA39   FAC (+-)*10->FAC
EA55:     25 DIV10 EQU $EA55   FAC (+)/10 ->FAC
EBAF:     26 ABS EQU $EBAF   ABS(FAC) ->FAC
EBB2:     27 FCOMP EQU $EBB2   FAC>=<MEM(Y,A)
DEBE:     28 CHKCOM EQU $DEBE   COMMA CHECK
DEC9:     29 SE EQU $DEC9    SYNTAX ERROR CHECK
DD7B:     30 FRMEVL EQU $DD7B   EVALUATE FORMULA (GEN)
DD67:     31 FRMNUM EQU $DD67   EVALUATE EXPRESSION (NUM)
F53A:     32 HLIN EQU $F53A   LINE TO X(X MSB, A LSB), Y(IN Y)
F411:     33 HPOSN EQU $F411   X IN Y(MSB) AND X(LSB), (Y IN A)
F457:     34 HPLOT EQU $F457   PLOT DOT AT HPOSN
F601:     35 DRAW EQU $F601   DRAW SHAPE IN X(LSB), Y(MSB)
0000:     36 ;
0000:     37 ;MEMORY USED
00EB:     38 LO EQU $EB      LOWEST VALUE, 5 BYTES
00FB:     39 HI EQU $FB      HIGHEST VALUE, 5 BYTES
00F9:     40 ROT EQU $F9      SHAPE ROTATION
0008:     41 SP EQU $08      SPX IN $08, SPY IN $09
00D7:     42 AX EQU $D7      1(Y-AXIS) OR 0(X-AXIS)
00FA:     43 COUNT EQU $FA    1 BYTE
0073:     44 HIM EQU $73    HIMEM
0100:     45 BUF EQU $100   STRING BUFFER
0314:     46 LEN EQU $314   LENGTH OF X(Y)-AXIS IN DOTS, 2BYTES
03F5:     47 AMPERS EQU $3F5  & VECTOR
0000:     48 ;
0000:     49 ;TEMPORY MEM USE
0093:     50 YA EQU $93      TEMP1, 5 BYTES
0098:     51 XA EQU $98      TEMP2
008A:     52 H EQU $8A      TEMP3
001E:     53 X1 EQU $1E      RUNNING X
001F:     54 Y1 EQU $1F      RUNNING Y
0006:     55 M EQU $06      1 BYTE
0007:     56 ML EQU $07      1 BYTE
0019:     57 SPTMP EQU $19    1 BYTE
00E3:     58 J EQU $E3      1 BYTE
0000:     59 ;
```

DE6502 KENNER

```

----- NEXT OBJECT FILE NAME IS GRAPHICS.OBJ0 200.H 78.1 70000
9055: 0000 60 ORG $9055
9055: 0000 61 ;
9055: 0000 62 ;*****BRUN BEFORE USE
9055:A9 AC 63 START LDA #>JMP_TBL 0000 00000
9055:8D F6 03 64 STA AMPERS+1 LSB JUMPTABLE 10000
905A:A9 93 65 LDA #<JMP_TBL 0000 10000
905C:8D F7 03 66 STA AMPERS+2 MSB 0000 17000
905F:A9 94 67 LDA #<EXIT 0000 17000
9061:85 E9 68 STA $E9 MSB SHAPE TABLE POINTER 17000
9063:A9 B6 69 LDA #>EXIT+1 0000 17000
9065:85 E8 70 STA $EB LSB SHAPE TABLE POINTER 17000
9067:D0 02 71 BNE IN2 0000 17000
9069:E6 E9 72 INC $E9 0000 17000
906B:A9 02 73 IN2 LDA #2 0000 17000
906D:85 FA 74 STA COUNT COUNTER TO DRAW E O AXIS (DRAWAX) 17000
906F:A9 55 75 LDA #START 0000 17000
9071:85 73 76 STA HIM 0000 17000
9073:85 6F 77 STA $6F 0000 17000
9075:A9 90 78 LDA #<START 0000 17000
9077:85 74 79 STA HIM+1 HIMEM TO BEGIN OF PROGRAM 17000
9079:85 70 80 STA $70 $POINTER TO HIMEM (NOT DONE BY PR#6 17000
907B:60 81 RTS 0000 17000
907C: 82 ;
907C:DC 83 RANGE DFB $DC RANGE X-AXIS=220 17000
907D:97 84 DFB $97 RANGE Y-AXIS=151 17000
907E:23 85 ORIGIN DFB $23 X COORDINATE=35 17000
907F:97 86 DFB $97 Y COORDINATE=151 17000
9080: 87 ;
9080: 88 ;ADDITIONAL SUBROUTINES 17000
9080: 89 ;CALCULATE MAX STRING LENGTH 17000
9080:A9 FB 90 MALEN LDA #HI HIGHEST SUBSCR 17000
9082:20 9E 90 91 JSR STRLEN 0000 17000
9085:86 07 92 STX ML L(HI)->ML 17000
9087:A9 EB 93 LDA #LO LOWEST SUBSCR 17000
9089:20 9E 90 94 JSR STRLEN 0000 17000
908C:E4 07 95 CPX ML L(LO)>L(HI)? 0000
908E:90 02 96 BCC MLE 0000 17000
9090:86 07 97 STX ML YES 0000 17000
9092:A9 BA 98 MLE LDA #H STEP 0000 17000
9094:20 9E 90 99 JSR STRLEN 0000 17000
9097:E4 07 100 CPX ML L(STEP)>ML? 0000
9099:90 02 101 BCC MLF 0000 17000
909B:86 07 102 STX ML YES, MAX $LENGTH 0000 17000
909D:60 103 MLF RTS 0000 17000
909E: 104 ;
909E: 105 ;CALCULATE STRING LENGTH 0000 17000
909E:A0 00 106 STRLEN LDY #0 0000 17000
90A0:20 F9 EA 107 JSR MOVFM MEM->FAC 0000
90A3:20 AF EB 108 JSR ABS 0000 17000
90A6:20 34 ED 109 JSR FOUT FAC TO STRING BUFFER 0000
90A9:A2 00 110 STRL2 LDX #0 0000 17000
90AB:EB 111 STR2 INX 0000 17000
90AC:BD 00 01 112 LDA BUF,X 0000 17000
90AF:D0 FA 113 BNE STR2 0000 17000
90B1:60 114 RTS STRING LENGTH IN X 0000 17000
90B2: 115 ;
90B2: 116 ;FLOATING POINT TIMES INTEGER 0000 17000
90B2: 117 ;INTEGER(X REG) * MEM(0,A REG) -> FAC 0000
90B2:85 19 118 FFTINT STA $19 0000 17000
90B4:86 1A 119 STX $1A SAVE A&X REGISTERS 0000 17000

```

DE 6502 KENNER

```

90B6:A9 00    120    LDA #0
90B8:20 93 EB  121    JSR FLOAT   H->FAC    P01 AB PAYACTP
90BB:A5 19    122 REST    LDA $19          P01 00 00 00:0001P
90BD:A0 00    123    LDY #0
90BF:20 BE E7  124    JSR FADD    FAC+MEM(0,A)->FAC    P01
90C2:C6 1A    125    DEC $1A          X REG -1    P01
90C4:D0 F5 00  126 BNE REST    H->H      P01
90C6:60      127    RTS
90C7:          128 ;
90C7:          129 ;X REG * INT IN MEM (1BYTE)
90C7:A0 00    130 INTMUO LDY #0
90CB:18 00 00  132 SPTMP CLC
90CC:65 19    133    ADC SPTMP A+A+SPTMP    P01 00 00 00:0001P
90CE:90 01    134 BCC IM3
90D0:C8      135 INY
90D1:CA      136 JIM3 DEX
90D2:D0 F7    137 BNE IM2
90D4:60      138 RTS
90D5:          139 ;
90D5:          140 ;CHANGE H-(*10 OR /10)
90D5:A0 00    141 CHNGH LDY #0
90D7:A9 BA    142 LDA #H
90D9:20 F9 EA  143 JSR MOVFM H->FAC
90DC:A6 E3    144 LDX J
90DE:CA      145 DEX
90DF:F0 05    146 BEQ DIV BRANCH ON J=1
90E1:20 39 EA  147 JSR MUL10 *10
90E4:D0 03    148 BNE SAV BRANCH
90E6:20 55 EA  149 DIV JSR DIV10 /10
90E9:A2 BA    150 SAV LDX #H
90EB:20 23 EB  151 JSR MOVML FAC->#H
90EE:60      152 RTS
90EF:          153 ;
90EF:          154 ;INTEGER DIVISION A REG / INT IN MEM(1BYTE)
90EF:A2 00    155 INTDIV LDX #0
90F1:38      156 SEC
90F2:E8      157 ID2 INX
90F3:E5 19    158 SBC SPTMP A=A-MEM(SPTMP)
90F5:B0 FB    159 BCS ID2
90F7:CA      160 DEX X=INT(A/MEM)
90FB:60      161 RTS
90F9:          162 ;
90F9:          163 ;FP MEM(0,X)-FP MEM(0,A) -> TEMP1
90F9:A0 00    164 SUBTMP LDY #0
90FB:86 46    165 STX $46
90FD:20 F9 EA  166 JSR MOVFM MEM(Y,A)->FAC
9100:A5 46    167 LDA $46
9102:20 A7 E7  168 JSR FSUB MEM(Y,A)-FAC
9105:20 21 EB  169 JSR MOV1F Y-A/->TEMP1
910B:60      170 RTS
9109:          171 ;
9109:          172 ;PARAM FOR CALC OF VALUES ALONG THE AXIS
9109:A2 FB    173 SUBSTP LDX #HI
910B:A9 EB    174 LDA #LO
910D:20 F9 90  175 JSR SUBTMP HI-LO->TMP1
9110:20 2A 91  176 JSR STORE
9113:A5 19    177 LDA SPTMP SP IN A
9115:20 93 EB  178 JSR FLOAT SP IN FAC
9118:A9 93    179 LDA #YA
911A:20 66 EA  180 JSR FDIV (HI-LO)/SP
911D:A2 BA    181 LDX #H
911F:20 23 EB  182 JSR MOVML ->TEMP3
9122:A2 EB    183 LDX #LO

```

DE 6502 KENNER

```

9124:A9 BA    184     LDA #H      TADW  R01    01 00 000000
9126:20 F9 90  185     JSR SUBTMR  LO-(HI-LO)/SP  R01    01 2A 000000
9129:60        186     RTS      01 00 000000
912A:           187 ;   01 00 000000
912A:           188 ;STORE LO IN $300(X) AND $30A(Y)
912A:           189 ;DOTS/(HI-LO) WITH PROPER SIGN IN $305, $30F
912A:A4 D7    190 STORE LDY AX    01 00 000000
912C:B9 14 03  191 LDA LEN,Y   01 00 000000
912F:A8        192 TAY      01 00 000000
9130:A9 00    193 LDA #0      01 00 000000
9132:20 F3 DE  194 JSR $DEF3   Y(LSB), A(MSB) INTO FAC 01 000000
9135:A9 93    195 LDA #YA    01 00 000000
9137:20 66 EA 196 JSR FDIV   LO-(HI-LO)/DOTS  R01    01 00 000000
913A:A0 E9    197 LDY #$E9    MSB FP 1.00  R01    00 000000
913C:A9 13    198 LDA #$13    LSB FP 1.00  R01    00 000000
913E:20 66 EA 199 JSR FDIV   DOTS/(HI-LO)  R01    01 00 000000
9141:A5 D7    200 LDA AX      01 00 000000
9143:D0 06    201 BNE ST2    BRANCH IF Y-AXIS  01 000000
9145:A5 A2    202 LDA $A2    01 00 000000
9147:49 80    203 EOR #$80    FACSIGN  01 00 000000
9149:85 A2    204 STA $A2    REVERSE SIGN IF X-AXIS 01 00 000000
914B:A9 05    205 ST2    01 00 000000
914D:20 63 91  206 JSR SET    01 00 000000
9150:AA        207 TAX      01 00 000000
9151:20 2B EB  208 JSR MOVFM  PACK FAC INTO $305(X) OR $30F(Y) 01 00 000000
9154:A9 EB    209 LDA #LO    01 00 000000
9156:20 F9 EA  210 HQJSR  MOVFM 01 00 000000
9159:A9 00    211 LDA #0      01 00 000000
915B:20 63 91  212 JSR SET    01 00 000000
915E:AA        213 TAX      01 00 000000
915F:20 2B EB  214 JSR MOVFM  PACK FAC INTO $300 ($30A) 01 00 000000
9162:60        215 RTS      01 00 000000
9163:           216 ;SET REGISTERS FOR DATA IN $300,$313 01 00 000000
9163:           217 ;SET REGISTERS FOR DATA IN $300,$313 01 00 000000
9163:18        218 SET CLC    01 00 000000
9164:A0 03    219 LDY #$03    01 00 000000
9166:A6 D7    220 LDX AX    01 00 000000
9168:F0 02    221 BEQ SETEX  01 00 000000
916A:69 0A    222 ADC #$0A    01 00 000000
916C:60        223 SETEX  RTS   01 00 000000
916D:           224 ; 01 00 000000
916D:           225 ;ROUND OFF TO 2 DECIMALS 01 00 000000
916D:A5 A2    226 RNDOFF LDA $A2    01 00 000000
916F:48        227 PHA      STORE SIGN 01 00 000000
9170:20 AF EB  228 JSR ABS    01 00 000000
9173:20 39 EA  229 JSR MUL10  01 00 000000
9176:20 39 EA  230 JSR MUL10  01 00 000000
9179:20 A0 E7  231 JSR FADDH  01 00 000000
917C:20 23 EC  232 JSR FPINT  01 00 000000
917F:20 55 EA  233 JSR DIV10  01 00 000000
9182:20 55 EA  234 JSR DIV10  01 00 000000
9185:68        235 PLA      01 00 000000
9186:85 A2    236 STA $A2    RESTORE SIGN 01 00 000000
9188:60        237 RTS      01 00 000000
9189:           238 ; 01 00 000000
9189:           239 ;CALCULATE VALUE AND ITS STARTING POINT 01 00 000000
9189:A0 00    240 SUBSCR LDY #0    01 00 000000
918B:A9 93    241 LDA #YA    TEMP1 ->FAC 01 00 000000
918D:20 F9 EA  242 JSR MOVFM 01 00 000000
9190:A9 8A    243 LDA #H    +TEMP3  01 00 000000
9192:20 BE E7  244 JSR FADD  01 00 000000
9195:20 6D 91  245 JSR RNDOFF  TO 2 DECIMALS 01 00 000000
9198:20 21 EB  246 JSR MOV1F  CURRENT SUBSCR IN TEMP1 01 00 000000

```

```

919B:20 34 ED 247 SC2    JSR   FOUT      FAC INTO STRING BUFFER
919E:A5 D7 248 LDA   AX
91A0:F0 18 249 BEQ   SC5      BRANCH ON X-AXIS
91A2:A5 07 250 LDA   ML      STRING LENGTH
91A4:2A 251 SC3    ROL   A      *4
91A5:2A 252 ROL   A      -X1
91A6:E5 1E 253 SBC   X1      -X1
91A8:49 FF 254 EOR   #$FF     X1-4L
91AA:AC 00 01 255 LDY   BUF
91AD:CO 2D 256 CPY   #$2D     VALUE<0?
91AF:D0 02 257 BNE   SC4
91B1:E9 04 258 SBC   #04     YES, X1-4
91B3:AA 259 SC4    TAX
91B4:A5 1F 260 LDA   Y1
91B6:69 02 261 ADC   #02
91B8:D0 0D 262 BNE   SC6
91BA:20 A9 90 263 SC5    JSR   STRL2
91BD:8A 264 TXA
91BE:E5 1E 265 SBC   X1      L-X1
91C0:49 FF 266 EOR   #$FF     X1-L1
91C2:AA 267 TAX
91C3:A5 1F 268 LDA   Y1
91C5:69 08 269 ADC   #08     Y1+8
91C7:A0 00 270 SC6    LDY   #0
91C9:20 11 F4 271 JSR   HPOSN
91CC:60 272 RTS
91CD:          273 ;
91CD:          274 ;DRAW CHARACTERS IN STRING BUFFER
91CD:A2 00 275 DRWSHP LDX   #0
91CF:18          SET UP POINTER TO ASCII TABLE
91DO:A5 E8 276 CLC   RR
91D2:69 5D 277 LDA   $EB
91D4:85 55 278 ADC   #$5D
91D6:A5 E9 279 STA   $55
91D8:69 01 280 LDA   $E9
91DA:85 56 281 ADC   #1
91DC:A0 00 282 STA   $56
91DE:BD 00 01 283 DR1    LDY   #0
91E1:F0 10 284 LDA   BUF,X
91E3:C8 286 DR2    INY
91E4:D1 55 287 CMP   ($55),Y
91E6:D0 FB 288 BNE   DR2
91E8:E8 289 INX
91E9:86 06 290 STX   M
91EB:98 291 TYA   A
91EC:20 F4 91 292 JSR   SHADDR
91EF:A6 06 293 LDX   M
91F1:D0 E9 294 BNE   DR1
91F3:60 295 SHFFIN RTS
91F4:          296 ;
91F4:          297 ;CALCULATE SHAPE ADDRESS
91F4:0A 298 SHADDR ASL   -A
91F5:AB 299 TAY
91F6:B1 E8 300 LDA   ($E8),Y
91FB:18 301 CLC   RR
91F9:65 E8 302 ADC   $E8
91FB:AA 303 TAX
91FC:C8 304 INY
91FD:B1 E8 305 LDA   ($E8),Y
91FF:65 E9 306 ADC   $E9
9201:A8 307 TAY
9202:A5 F9 308 LDA   ROT
9204:20 01 F6 309 JSR   DRAW

```

DE 6502 KENNER

ADDRESS	DATA	OPCODE	REGISTERS	ACCUMULATOR	STATUS	PC	PSW
9207:60	310	RTS	Y= ADD	Y=0	SD	0A1B01P	
9208:	311 ;	CD8	D32	Y=0	SD	0A1C01P	
9208:	312 ;	ROUND OFF SP-LENGTH (DOTS)	SP, Y	Y=0	SD	0A1D01P	
9208:B9 0B 00	313	PARAX	LDA SP, Y	Y=0	SD	0A1E01P	
9208:B5 19	314	STA SPTMP	RUNNING SP	Y=0	SD	0A1F01P	
920D:B9 7C 90	315	LDA RANGE, Y	RUNNING RANGE	Y=0	SD	0A2001P	
9210:20 EF 90	316	JSR INTDIV	SP=0	Y=0	SD	0A2101P	
9213:86 E3	317	STX J	J=INT(RANGE/SP)	Y=0	SD	0A2201P	
9215:60	318	RTS	Y=0	Y=0	SD	0A2301P	
9216:	319 ;	CD8	D32	Y=0	SD	0A2401P	
9216:A4 1F	320	LINA1	LDY Y1	Y=0	SD	0A2501P	
9218:A2 00	321	LDX #0	LDY Y0	Y=0	SD	0A2601P	
921A:A5 1E	322	LDA X1	LDY Y0	Y=0	SD	0A2701P	
921C:20 3A F5	323	JSR HLIN	FROM CURRENT CURSOR POSN TO X1,Y1				
921F:60	324	RTS	Y=0	Y=0	SD	0A2801P	
9220:	325 ;	CD8	D32	Y=0	SD	0A2901P	
9220:	326 ;SCALING	CD8	D32	Y=0	SD	0A2A01P	
9220:A9 01	327	SCALE	LDA #01	Y=0	SD	0A2B01P	
9222:20 93 EB	328	JSR FLOAT	1->FAC	Y=0	SD	0A2C01P	
9227:20 23 EB	330	JSR MOVML	FAC->H	Y=0	SD	0A2D01P	
922A:A9 03	331	BEGIN	LDA #03	Y=0	SD	0A2E01P	
922C:A6 D7	332	LDX AX	1->Y-AXIS, 0->X-AXIS	Y=0	SD	0A2F01P	
922E:95 08	333	STA SP, X	SP=3	Y=0	SD	0A3001P	
9230:A6 D7	334	INCSP	LDX AX	Y=0	SD	0A3101P	
9232:F6 08	335	INC SP, X	SP+1	Y=0	SD	0A3201P	
9234:A9 08	336	LDA #0B	Y=0	Y=0	SD	0A3301P	
9236:D5 08	337	CMP SP, X	Y=0	Y=0	SD	0A3401P	
9238:10 05	338	BPL JO	BRANCH ON SP<=8	Y=0	SD	0A3501P	
923A:20 D5 90	339	JSR CHNGH	H#10->#H	Y=0	SD	0A3601P	
923D:D0 EB	340	BNE BEGIN	Y=0	Y=0	SD	0A3701P	
923F:A9 00	341	JO	LDA #0	Y=0	SD	0A3801P	
9241:B5 E3	342	STA J	J=0	Y=0	SD	0A3901P	
9243:E6 E3	343	INC J	J+1	Y=0	SD	0A3A01P	
9245:A5 E3	344	LDA J	Y=0	Y=0	SD	0A3B01P	
9247:C9 03	345	CMP #03	J=3?	Y=0	SD	0A3C01P	
9249:D0 02	346	BNE J6	BRANCH ON J#3	Y=0	SD	0A3D01P	
924B:E6 E3	347	INC J	J=4	Y=0	SD	0A3E01P	
924D:C9 06	348	J6	CMP #06	J=6?	SD	0A3F01P	
924F:F0 DF	349	BEQ INCSP	BRANCH ON J=6	Y=0	SD	0A4001P	
9251:A6 E3	350	J1	LDX J	J=0	SD	0A4101P	
9253:A9 8A	351	LDA #H	H=Y	Y=0	SD	0A4201P	
9255:20 B2 90 B	352	JSR FFTINT	XA=J*H ->FAC	Y=0	SD	0A4301P	
9258:20 1E EB	353	JSR MOV2F	FAC->TEMP2, RESETS Y	Y=0	SD	0A4401P	
925B:A9 EB	354	LDA #LO	LSB LO	Y=0	SD	0A4501P	
925D:20 66 EA	355	JSR FDIV	LO/XA ->FAC	Y=0	SD	0A4601P	
9260:20 23 EC	356	JSR FPINT	INTEGER(FAC), RESETS Y	Y=0	SD	0A4701P	
9263:A9 98	357	LDA #XA	Y=0	Y=0	SD	0A4801P	
9265:20 7F E9	358	JSR FMULT	YA=XA*INT(LO/XA)	Y=0	SD	0A4901P	
9268:20 21 EB	359	JSR MOV1F	YA->TEMP1	Y=0	SD	0A4A01P	
926B:A4 D7	360	LDY AX	0->X AXIS, 1->Y AXIS	Y=0	SD	0A4B01P	
926D:B6 08	361	LDX SP, Y	SP=Y	Y=0	SD	0A4C01P	
926F:A9 98	362	LDA #XA	LSB(XA)	Y=0	SD	0A4D01P	
9271:20 B2 90	363	JSR FFTINT	SP*XA->FAC	Y=0	SD	0A4E01P	
9274:20 1E EB	364	JSR MOV2F	XA->ZP(X REG), RESETS Y	Y=0	SD	0A4F01P	
9277:A9 93	365	LDA #YA	Y=0	Y=0	SD	0A5001P	
9279:20 BE E7	366	JSR FADD	XA+YA(TEMP1) ->FAC	Y=0	SD	0A5101P	
927C:A0 00	367	LDY #0	AC1	Y=0	SD	0A5201P	
927E:A9 FB	368	LDA #HI	Y=0	Y=0	SD	0A5301P	
9280:20 B2 EB	369	JSR FCMP	FAC>HI?	Y=0	SD	0A5401P	
9283:C9 FF	370	CMP #\$FF	AC1	Y=0	SD	0A5501P	
9285:F0 BC	371	BEQ INCJ	BRANCH ON HI>XA+YA	Y=0	SD	0A5601P	
9287:A6 D7	372	LDX AX					

```

9289:A9 08      373    LDA #0B      XA 00I      000 00 00 00000000
928B:D5 08      374    CMP SP,X  C0D0D0  A0I      000 00 00 00000000
928D:F0 26      375    BEQ SAVLH  BRANCH ON SP=8 000 00 00 00000000
928F:A2 03      376    LDX #03      00 AXI      000 00 00 00000000
9291:A9 98      377    LDA #XA      00 YDI      000 00 00 00000000
9293:20 B2 90      378    JSR FPTINT  3**XA      000 00 00 00000000
9294:C6 9D      379    DEC $9D      FAC/2      000 00 00 00000000
9298:C6 9D      380    DEC $9D      FAC/2      000 00 00 00000000
929A:A9 EB 00      381    LDA #LO      T0000000 000 00 00 00000000
929C:20 BE E7      382    JSR FADD   LO+.75*XA->FAC 000 00 00 00000000
929F:AO 00      383    LDY #0      00000000 000 00 00 00000000
92A1:A9 FB      384    LDA #HI      00000000 000 00 00 00000000
92A3:20 B2 EB      385    JSR FCMP   LO+.75*XA>HI($FB)? 000 00 00 00000000
92A6:C9 01      386    CMP #01      00000000 000 00 00 00000000
92AB:D0 0B      387    BNE SAVLH  BRANCH ON FAC<HI 000 00 00 00000000
92AA:A9 FE      388    LDA #$FE      00000000 000 00 00 00000000
92AC:24 E3      389    BIT J       00000000 000 00 00 00000000
92AE:D0 80      390    BNE INCSP   BRANCH ON J>1 000 00 00 00000000
92B0:20 D5 90      391    JSR CHNGH  H/10 ->H 000 00 00 00000000
92B3:D0 9C      392    BNE J1      BRANCH WITH J=1 000 00 00 00000000
92B5:AO 00      393    SAVLH   LDY #0      00000000 000 00 00 00000000
92B7:A9 93      394    LDA #YA      00000000 000 00 00 00000000
92BC:A2 EB      396    LDX #LO      00000000 000 00 00 00000000
92BE:20 23 EB      397    JSR MOVML   FAC ->LO 000 00 00 00000000
92C1:A9 98      398    LDA #XA      00000000 000 00 00 00000000
92C3:20 BE E7      399    JSR FADD   +XA      000 00 00 00000000
92C6:A2 FB      400    LDX #HI      00000000 000 00 00 00000000
92C8:20 23 EB      401    JSR MOVML   XA+YA ->HI 000 00 00 00000000
92CB:60          402    RTS      00000000 000 00 00 00000000
92CC:          403    ;
92CC:          404    ;CALCULATE COORDINATES TO PLOT
92CC:20 B1 00      405    COORD1  JSR ADVTP  ADVANCETEXTPOINTER
92CF:20 67 DD      406    COORD2  JSR FRMNUM  VALUE INTO FAC 000 00 00 00000000
92D2:8A          407    TXA      ACCU=0 000 00 00 00000000
92D3:20 63 91      408    JSR SET      SET REG TO POINT TO MÉM 000 00 00 00000000
92D6:20 A7 E7      409    JSR FSUB   MEM-FAC 000 00 00 00000000
92D9:A9 05          410    LDA #5      SET REG 000 00 00 00000000
92DE:20 63 91      411    JSR SET      MEM*FAC 000 00 00 00000000
92DE:20 7F E9      412    JSR FMULT  FAC->TEMP1 000 00 00 00000000
92E1:20 21 EB      413    JSR MOV1F  TEMP1->REG 000 00 00 00000000
92E4:A6 D7          414    LDX AX      00000000 000 00 00 00000000
92E6:98          415    TYA      00000000 000 00 00 00000000
92E7:BC 7E 90      416    LDY ORIGIN,X 00000000 000 00 00 00000000
92EA:20 F3 DE      417    JSR $DEF3  Y(LSB),A(MSB) INTO FAC 000 00 00 00000000
92ED:A9 93          418    LDA #$93      00000000 000 00 00 00000000
92EF:20 BE E7      419    JSR FADD   MEM+FAC 000 00 00 00000000
92F2:20 A0 E7      420    JSR FADDH  FAC+.5 000 00 00 00000000
92F5:20 FB E6      421    JSR CONINT  FAC INTO X REGISTER 000 00 00 00000000
92F8:A4 D7          422    LDY AX      00000000 000 00 00 00000000
92FA:96 06          423    STX M,Y      M, M+1 CONTAIN COORDINATES 000 00 00 00000000
92FC:60          424    RTS      00000000 000 00 00 00000000
92FD:          425    ;
92FD:20 BE DE      426    VAL1  JSR CHKCOM  CHECK ON COMMA 000 00 00 00000000
9300:20 67 DD      427    VAL2  JSR FRMNUM  VALUE INTO FAC 000 00 00 00000000
9303:20 FB E6      428    JSR CONINT  FAC INTO X REGISTER 000 00 00 00000000
9306:60          429    RTS      00000000 000 00 00 00000000
9307:          430    ;
9307:          431    ;***** ENTRY &PLOT: PLOT SYMBOL DEPENDING ON CODE 00000000
9307:A9 00          432    SRPLOT LDA #0      00000000 000 00 00 00000000
9309:85 D7          433    STA AX      00000000 000 00 00 00000000
930B:20 CC 92          434    JSR COORD1  CALC X 000 00 00 00000000
930E:20 BE DE          435    JSR CHKCOM  CHECK ON COMMA 000 00 00 00000000

```

DE6502 KENNER

```

9311:E6 D7    436    INC AX      80A PCL    ZCF    80 04:080P
9313:20 CF 92  437    JSR COORD2 A CALC Y    ACC  80 04:080P
9316:20 FD 92  438    JSR VAL1 H PULL PLOTCODE INTO X    ACC  80 04:080P
9319:86 E3    439    STX J      80B X1    ACC  80 04:080P
931B:A0 00    440    LDY #0    80C ACC1    ACC  80 04:080P
931D:84 F9    441    STY ROT   80D ROT=0    ACC  80 04:080P
931F:A6 06    442    LDX M      80E 080    ACC  80 04:080P
9321:A5 07    443    LDA M+1  80F 081    ACC  80 04:080P
9323:20 57 F4  444    JSR HPLOT 810 PLOT DOT AT X=X, Y=ACCU 80 04:080P
9326:A6 E3    445    LDX J      811 081    ACC  80 04:080P
9328:F0 09    446    BEQ SRPEX 812 BRANCH IF CODE=0 80 04:080P
932A:CA      447    DEX      813 ACC1    ACC  80 04:080P
932B:8A      448    TXA      814 081    ACC  80 04:080P
932C:D0 06    449    BNE SRPCR 815 BRANCH IF PLOTCODE>1 80 04:080P
932E:A9 1D    450    LDA #$1D 816 H PCL    ACC  80 04:080P
9330:20 F4 91  451    JSR SHADDR 817 CODE=1, DRAW SQUARE 80 04:080P
9333:60      452    SRPEX  RTS    818 081    ACC  80 04:080P
9334:A5 E3    453    SRPCR  H PCL    819 081    ACC  80 04:080P
9336:29 01    454    AND #1   820 HOMING 80 04:080P
9338:D0 04    455    BNE SRF2 821 BRANCH IF CODE=3 80 04:080P
933A:A9 20    456    LDA #$20 822 ROT=-90 80 04:080P
933C:85 F9    457    STA ROT   823 FOR CODE=2 OR 4 80 04:080P
933E:A9 1E    458    SRP2   LDA #$1E 824 081    ACC  80 04:080P
9340:20 F4 91  459    JSR SHADDR 825 DRAW HALF CROSS 80 04:080P
9343:85 F9    460    STA ROT   826 RESET ROT=0 80 04:080P
9345:A5 E3    461    LDA J      827 081    ACC  80 04:080P
9347:29 03    462    AND #03 828 081    ACC  80 04:080P
9349:D0 E8    463    BNE SRPEX 829 BRANCH IF CODE=2 OR 3 80 04:080P
934B:C6 E3    464    DEC J      830 CODE=3 80 04:080P
934D:D0 EF    465    BNE SRP2 831 BRANCH 80 04:080P
934F:          466    ; 832 081    ACC  80 04:080P
934F:          467    ;***** ENTRY &DRAW: TEXT ALONG AXES 833 081    ACC  80 04:080P
934F:A2 00    468    SRDRAW  LDX #0 834 081    ACC  80 04:080P
9351:86 71    469    STX $71 835 081    ACC  80 04:080P
9353:E8      470    INX      836 081    ACC  80 04:080P
9354:86 72    471    STX $72 837 POINTER TO $BUFFER 80 04:080P
9356:20 B1 00  472    JSR ADVTP 838 ADVANCE TEXTPOINTER 80 04:080P
9359:20 7B DD  473    JSR FRMEVL 839 EVALUATE FORMULA 80 04:080P
935C:A6 9E    474    LDX $9E 840 081    ACC  80 04:080P
935E:A4 9F    475    LDY $9F 841 $POINTER, $L IN ACCU 80 04:080P
9360:20 E2 E5  476    JSR MOVSTR 842 $ ->BUF 80 04:080P
9363:AA      477    TAX      843 081    ACC  80 04:080P
9364:98      478    TYA      844 081    ACC  80 04:080P
9365:9D 00 01  479    STA BUF,X 845 ZERO AT $END 80 04:080P
9368:20 B1 00  480    JSR ADVTP 846 081    ACC  80 04:080P
936B:C9 58    481    CMP #$58 847 081    ACC  80 04:080P
936D:F0 13    482    BEQ TXTX 848 BRANCH IF X 80 04:080P
936F:C9 59    483    CMP #$59 849 081    ACC  80 04:080P
9371:F0 20    484    BEQ TXTY 850 BRANCH IF Y 80 04:080P
9373:20 00 93  485    JSR VAL2 851 FULL X COORD INTO X REG 80 04:080P
9376:86 1E    486    STX X1 852 081    ACC  80 04:080P
9378:20 FD 92  487    JSR VAL1 853 FULL Y COORD INTO X REG 80 04:080P
937B:86 1F    488    STX Y1 854 081    ACC  80 04:080P
937D:20 A4 91  489    JSR SC3 855 CALC START POSN 80 04:080P
9380:D0 22    490    BNE TXTDS 856 BRANCH 80 04:080P
9382:A9 00    491    TXTX 857 X-AXIS 80 04:080P
9384:85 F9    492    LDA #0 858 RESET ROT 80 04:080P
9386:A9 FB    493    LDA #$FB 859 081    ACC  80 04:080P
9388:85 1E    494    STA X1 860 081    ACC  80 04:080P
938A:A9 91    495    LDA #$91 861 081    ACC  80 04:080P
938C:85 1F    496    STA Y1 862 081    ACC  80 04:080P
938E:20 A4 91  497    JSR SC3 863 081    ACC  80 04:080P
9391:D0 0E    498    BNE TXTDR 864 BRANCH

```

```

9393:8A      499 TXTY RD STAN Y-AXIS    162  80 ORIGPAR
9394:0A      500 ASL A      032  262  80 ZEROPAR
9395:0A      501 ASL A      033  263  80 AF ORIGPAR
9396:69 20    502 ADC #$20    17 ATB  STAN A&C  81 ZEROPAR
9398:A2 05    503 LDX #$05  18 100  80  80 ZEROPAR
939A:A0 30    504 LDY #$30  18 100  80  80 ZEROPAR
939C:84 F9    505 STY ROT  17 100  ROT=-90  80  80 ZEROPAR
939E:20 C7 91  506 JSR SC6    CALC POSN  18 100  80 ZEROPAR
93A1:20 B1 00  507 TXTDR  JSR ADVTP  18 100  80  80 ZEROPAR
93A4:20 CD 91  508 TXTDS  JSR DRWSHP  DRAW CHAR'S IN $BUF  80 ZEROPAR
93A7:84 F9    509 STY ROT  17 100  RESET ROT=0  80  80 ZEROPAR
93A9:84 52    510 STY #$52  17 100  SUPPRESS 'FORMULA TOO COMPLEX' ERF
93AB:60      511 RTS     8TH   80  80 ZEROPAR
93AC:          512 ;      17 100  80 ZEROPAR
93AC:          513 ;***** 20010100 80 HTIM XA WORD  262  100  80 ZEROPAR
93AC:          514 ;***** MAIN ENTRY VIA JUMPTABLEWORD AND  09  80 ZEROPAR
93AC:C9 8E    515 JMPTBL CMP #$8E    17 100  80  80 ZEROPAR
93AE:F0 12    516 BEQ LINAX1  BRANCH IF HLIN 0&C  09  80 ZEROPAR
93B0:C9 8F    517 CMP #$8F    17 100  80  80 ZEROPAR
93B2:F0 0E    518 BEQ LINAX1  BRANCH IF VLIN 0&C  81 ZEROPAR
93B4:C9 94    519 CMP #$94    17 100  80  80 ZEROPAR
93B6:F0 97    520 BEQ SRDRAW  BRANCH IF DRAW 0&C  81 ZEROPAR
93B8:C9 8D    521 CMP #$8D    17 100  80  80 ZEROPAR
93BA:D0 03    522 BNE PRSE   BRANCH IF NOT PLOT  ADJPAR
93BC:4C 07 93  523 JMP SRPLOT  17 100  80  80 ZEROPAR
93BF:4C C9 DE  524 PRSE   JMP SE    ERROR    00  80 ZEROPAR
93C2:          525 ;      17 100  80 ZEROPAR
93C2:          526 ;***** ENTRY &HLIN AND &VLIN: DRAW X- OR Y-AXIS  80 ZEROPAR
93C2:38      527 LINAX1 SEC    17 100  80 ZEROPAR
93C3:E9 8E    528 SBC #$8E    TOKEN FOR HLIN 0&C  00 CALIBR
93C5:85 D7    529 STA AX    HLIN=0, VLIN=1  17 100  80 AF ORIGPAR
93C7:20 B1 00  530 JSR ADVTP  ADVANCE TEXTPOINTER  17 100  80 ZEROPAR
93CA:F0 36    531 BNE PRSE   PARALLEL AXIS W/O NUM.  80 ORIGPAR
93CC:20 67 DD  532 STA JSR FRMNUM  EVALUATE EXPRESSION 17 100  80 ZEROPAR
93CF:A2 EB    533 LDX #LO  17 100  80  80 ZEROPAR
93D1:20 23 EB  534 MOVML  17 100  STORE LO  80  80 ZEROPAR
93D4:20 BE DE  535 JSR CHKCOM  CHCK ON COMMA  17 100  80 ZEROPAR
93D7:20 67 DD  536 STA JSR FRMNUM  EVALUATE NEXT  09  80 81 ZEROPAR
93DA:A0 00    537 LDY #0    17 100  80  80 ZEROPAR
93DC:A9 EB    538 LDA #LO  17 100  80  80 ZEROPAR
93DE:20 B2 EB  539 JSR FCOMP  HI>LO?  100  80  80 ZEROPAR
93E1:C9 01    540 CMP #01    17 100  80  80 ZEROPAR
93E3:D0 DA    541 BNE PRSE   ERROR    00  80  80 ZEROPAR
93E5:A2 FB    542 LDX #HI  17 100  80  80 ZEROPAR
93E7:20 23 EB  543 JSR MOVML  STORE HI  00  80  80 ZEROPAR
93EA:20 20 92  544 JSR SCALE  CALCULATE LO AND HI  17 100  80 ZEROPAR
93ED:A4 D7    545 LDY AX    17 100  80  80 ZEROPAR
93EF:20 08 92  546 JSR PARAX LOAD SP, CALC INT(RANGE/SP)  17 100  80 ZEROPAR
93F2:20 C9 90  547 JSR INTMUL SP*J  17 100  80  80 ZEROPAR
93F5:99 14 03  548 STA LEN,Y  SCALE LENGTH IN # OF DOTS  17 100  80 ZEROPAR
93F8:20 09 91  549 JSR SUBSTP  LOWEST-STEP IN TEMP1, STEP IN TEMP  17 100  80 ZEROPAR
93FB:20 B0 90  550 JSR MALEN  CALC MAX STRING LENGTH  17 100  80 ZEROPAR
93FE:20 2D 94  551 JSR DRAWAX  DRAW AXIS WITH NUMERALS  17 100  80 ZEROPAR
9401:60      552 RTS     17 100  RETURN TO BASIC  09  80 81 ZEROPAR
9402:A9 00    553 LINAX2 LDA #0    17 100  80  80 ZEROPAR
9404:85 07    554 STA ML    NO NUMERALS FOR PARALLEL AXIS  80  80 ZEROPAR
9406:A5 D7    555 LDA AX    17 100  ATB  17 100  80  80 ZEROPAR
9408:A8      556 TAY    17 100  80  80 ZEROPAR
9409:49 01    557 EOR #01    REVERSE CURRENT AXIS  17 100  80 ZEROPAR
940B:AA      558 TAX    17 100  80  80 ZEROPAR
940C:AD 7F 90  559 LDA ORIGIN+1  17 100  80  80 ZEROPAR
940F:CO 01    560 CPY #1    17 100  80  80 ZEROPAR

```

DE 6502 KENNER

```

9411:F0 04      561    BEQ Y NXT2      BRANCH ON 2ND Y-AXIS      AB-1000P
9413:38      562    SEC      A 184 000      PULLUPP
9414:FD 14 03    563    SBC LEN,X      2ND X: Y0=Y0-LEN(Y)      AC-1000P
9417:85 1F      564    STA Y1      0048 004      C0 0000P
9419:AD 7E 90    565    LDA ORIGIN      0000 000      C0 0000P
941C:C0 00      566    CPY #0      0000 000      C0 0000P
941E:F0 04      567    BEQ NXT3      BRANCH ON 2ND X-AXIS      AB-1000P
9420:18      568    CLC      0000 000      C0 0000P
9421:7D 14 03    569    ADC LEN,X      2ND Y: X0=X0+LEN(X)      AB-1000P
9424:85 1E      570    STA X1      0000 000      C0 0000P
9426:20 08 92    571    JSR PARAX      SP, INT(RANGE/SP)      AB-1000P
9429:20 37 94    572    JSR LOOP      DRAW AXIS WITHOUT NUMERICALS      AB-1000P
942C:60      573    RTS      0000 000      C0 0000P
942D:          574    ;      0000 000      C0 0000P
942D:          575    ;DRAW AX WITH OR WITHOUT SUBSCRIPT      0000 000      C0 0000P
942D:AC 7E 90    576    DRAWAXB LDY ORIGIN      0000 000      C0 0000P
9430:84 1E      577    STY X1      0000 000      C0 0000P
9432:AC 7F 90    578    LDY ORIGIN+1      0000 000      C0 0000P
9435:84 1F      579    STY Y1      0000 000      C0 0000P
9437:A6 1E      580    LOOP #1 LDX X1      0000 000      C0 0000P
9439:A4 1F      581    LDY Y1      0000 000      C0 0000P
943B:A5 D7      582    LDA AX      0000 000      C0 0000P
943D:F0 04      583    BEQ XAX      0000 000      C0 0000P
943F:CA      584    DEX      0000 000      C0 0000P
9440:CA      585    DEX      0000 000      C0 0000P
9441:D0 02      586    BNE BOTH1      0000 000      C0 0000P
9443:C8      587    XAX      0000 000      C0 0000P
9444:C8      588    INY      0000 000      C0 0000P
9445:98      589    BOTH1 TYA      Y: Y1, X: Y1+2      AB-1000P
9446:A0 00      590    LDY#0      0000 000      C0 0000P
9448:20 4A FF    591    JSR IOSAVE      SAVE COORDINATES      AB-1000P
944B:A5 07      592    LDA ML      =0 FOR PARALLEL AXIS W/O NUM      AB-1000P
944D:F0 06      593    BEQ DRASEC      BRANCH ON 2ND AXIS      AB-1000P
944F:20 B9 91      594    JSR SUBSCR      CALCULATE VALUE & CURSOR POSN      AB-1000P
9452:20 CD 91      595    JSR DRWSHP      DRAW CHAR'S IN STRING BUFFER      AB-1000P
9455:20 3F FF    596    DRASEC JSR IOREST      RESTORE COORDINATES      AB-1000P
9456:20 11 F4    597    JSR HFOSN      CURSOR TO X1-2,Y1(Y) OR X1,Y1+2()      AB-1000P
945B:20 16 92    598    JSR LINA1      DRAW LINE TO X1,Y1      AB-1000P
945E:A5 19      599    LDA SFTMP      0000 000      C0 0000P
9460:F0 1B      600    BEQ FIN      BRANCH ON END OF AXIS      AB-1000P
9462:A4 D7      601    LDY AX      0000 000      C0 0000P
9464:B9 1E 00    602    LDA X1,Y      0000 000      C0 0000P
9467:CO 01      603    CPY #1      0000 000      C0 0000P
9469:F0 05      604    BEQ Y      BRANCH ON Y-AXIS      AB-1000P
946B:18      605    CLC      0000 000      C0 0000P
946C:65 E3      606    ADC J      ACCU=X1+J      AB-1000P
946E:90 03      607    BCC BOTH2      BRANCH      AB-1000P
9470:38      608    Y      SEC      0000 000      C0 0000P
9471:E5 E3      609    SBC J      ACCU=Y1-J      AB-1000P
9473:99 1E 00    610    BOTH2 STA X1,Y      0000 000      C0 0000P
9476:20 16 92    611    JSR LINA1      PLOT X1,Y1 TO X2,Y2      AB-1000P
9479:C6 19      612    DEC SFTMP      SF-1      AB-1000P
947B:DO BA      613    BNE LOOP      SP=0?      AB-1000P
947D:C6 FA      614    FIN DEC COUNT      YES, COUNT=COUNT-1      AB-1000P
947F:DO B6      615    BNE LOOP      BRANCH ON TEL=1      AB-1000P
9481:A9 02      616    LDA #2      0000 000      C0 0000P
9483:85 FA      617    STA COUNT      RESET COUNT=2      AB-1000P
9485:60      618    EXIT RTS      0000 000      C0 0000P
*** SUCCESSFUL ASSEMBLY: NO ERRORS

```

DE6502 KENNER

*9486..95E3

```

9486- 1F 00
9488- 40 00 4A 00 55 00 5E 00
9490- 6B 00 70 00 7A 00 82 00
9498- BC 00 94 00 9D 00 A3 00
94A0- AC 00 B5 00 C0 00 C9 00
94AB- D4 00 E1 00 ED 00 FB 00
94B0- 03 01 0E 01 18 01 20 01
94B8- 2C 01 3A 01 45 01 47 01
94C0- 49 01 4F 01 55 01 55 38
94CB- 24 24 C7 0A B5 52 09 00
94D0- 24 AD 3A 27 48 24 3F 96
94DB- 4A 09 00 9C 29 25 24 3C
94E0- B7 A9 4A 00 09 64 C0 F6
94EB- C0 36 2D 76 01 00 2D 24
94F0- 3F 24 2D 96 4A 00 2D 24
94FB- 3F 24 2D DE 36 4E 09 00
9500- 09 24 24 3F 56 B5 09 00
950B- 2D E4 27 2C 35 FE 36 6D
9510- 01 00 2D 24 24 3F 36 AD
951B- 4A 00 2D 24 24 3F 36 36
9520- 4D 01 00 41 3B 98 AD 4A
9528- 00 1A 2C 24 25 44 3E 96
9530- 4A 00 24 24 2D 36 BF 4A
953B- 64 52 00 24 24 2D 36 BF
9540- 2A 2D 24 97 49 00 2D 9C
9548- 23 24 2C 35 92 09 00 24
9550- 24 AD C1 97 97 AD C1 57
955B- 4B 00 45 6B C0 3F 37 36
9560- 2E 2D 1C 0B 76 01 00 24
956B- 24 35 AD 63 2C 36 F6 0A
9570- 20 51 00 24 64 98 2E 2E
957B- 2E 64 18 B6 9E 01 60 0A
9580- 00 24 24 2D 36 3F 76 49
958B- 00 24 24 2D 36 5F 4A 21
9590- BC 4A 01 00 2D 24 3F 24
959B- 2D 9E D2 4C 4A 00 21 E4
95A0- 2B B8 37 96 49 00 C0 40
95AB- 2E 36 35 25 25 44 3E 96
95B0- 49 00 E4 40 76 4C B8 57
95B8- E0 93 AD 8D E3 4C 52 00
95C0- 64 DB 40 72 98 B6 AD C1
95CB- 47 B9 00 09 00 0D 00 25
95D0- 3F 36 2D 1C 00 8C FF B8
95D8- AD 23 00 1A 27 24 24 2C
95E0- 36 36 3E 00
#95E4..95FF
95E4- 31 32 33 34
95EB- 35 36 37 38 39 30 2D 2F
95FO- 41 42 43 45 47 4D 4E 50
95FB- 52 53 54 56 4B 4C 20 2E

```

Figure 7 Shape table and ASCII codes of ASCII characters present in the shape table

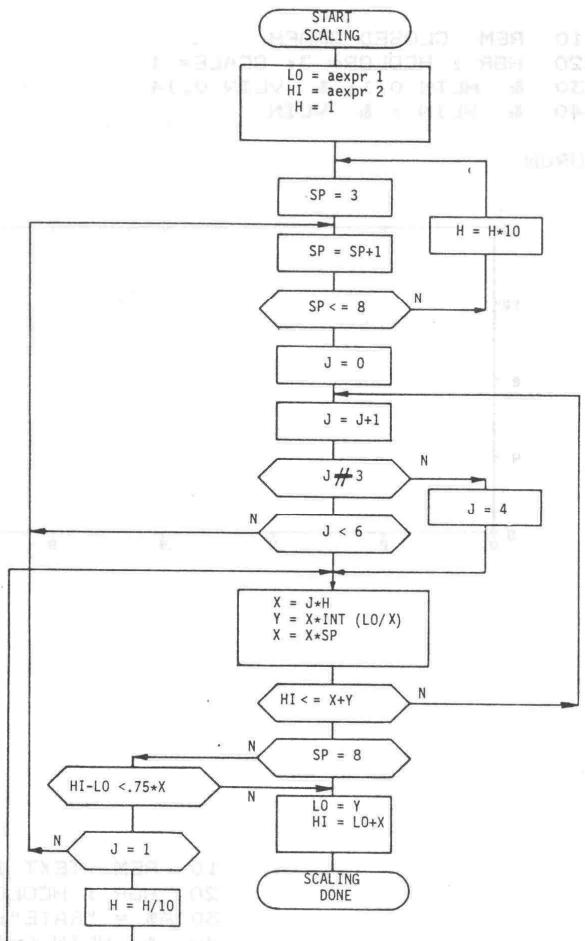


Figure 2 Calculation of the number of scale divisions and the lowest and highest numeral values along the axis.

```

10 REM SCALE AND DRAW HORIZONTAL AXIS
20 HGR : HCOLOR= 3: SCALE= 1
30 A = -.06
40 & HLIN A,1.9 / 4

```

URUN

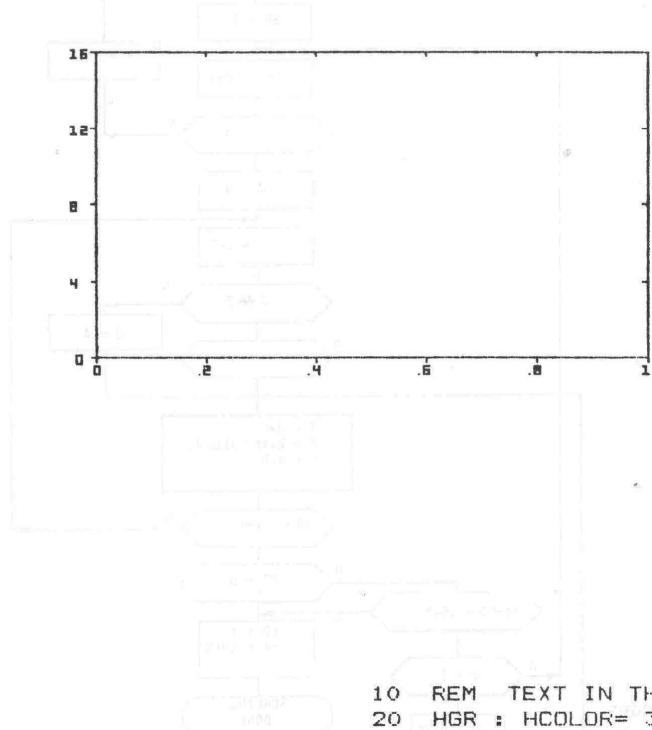
-1 0 .1 .2 .3 .4 .5

Figure 3

DE6502 KENNER

```
10 REM CLOSED GRAPH
20 HGR : HCOLOR= 3: SCALE= 1
30 & HLIN 0,1: & VLIN 0,14
40 & HLIN : & VLIN
```

URUN



```
10 REM CLOSER GRAPH
20 HGR : HCOLOR= 3: SCALE= 1
30 & HLIN 0,1: & VLIN 0,14
40 & HLIN : & VLIN
```

Figure 4

```
10 REM TEXT IN THE HI-RES PAGE
20 HGR : HCOLOR= 3: SCALE= 1: ROT= 0
30 A$ = "RATE": & DRAW A$,15,80
40 & HLIN 0,1: & DRAW "P ATA",X
50 ROT= 48: & DRAW "ROTATE",20,120
```

URUN

Figure 5

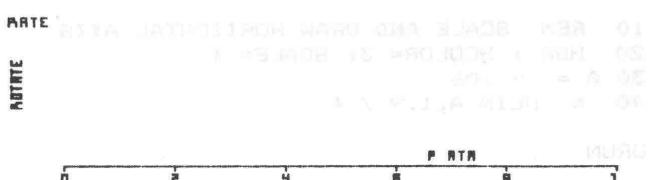


Figure 5

DE 6502 KENNER

```

20 HGR : HCOLOR= 3: SCALE= 1: ROT= 0
30 FOR I = 1 TO 2
40 SCALE= I:A = 32 - I
50 DRAW 1 AT 10,20 * I
60 FOR J = 2 TO A: DRAW J: REM CURSOR ENDS AT NEXT STARTING POINT
70 IF J > 27 THEN DRAW 27: DRAW 27: DRAW 27: REM CURSOR ENDS WHERE IT IS
    TARTS
80 NEXT J,I

```

URUN

1234567890-/ABCCEGMNPASTWKL...I

1234567890-/ABCCEGMNPASTWKL...I + MGR FORVAD + TRATE 001

Figure 8

FOR I=1 TO 20: DRAW 1 AT 10,20 * I: DRAW 27: DRAW 27: DRAW 27: I = I + 1: FOR O = 1 TO 20:

```

10 REM PLOT-CODES
20 HGR : HCOLOR= 3: SCALE= 1: ROT= 0
30 & HLIN 0,100: & VLIN 0,20
40 FOR I = 1 TO 5
50 & PLOT 20 * I,4 * I,I+1,I+2,I+3,I+4,I+5,I+6,I+7,I+8,I+9,I+10,I+11,I+12,I+13,I+14,I+15,I+16,I+17,I+18,I+19,I+20
60 NEXT I: FOR O = 1 TO 20: DRAW 1 AT 10,20 * O: DRAW 27: DRAW 27: DRAW 27: O = O + 1: FOR O = 1 TO 20:
```

URUN

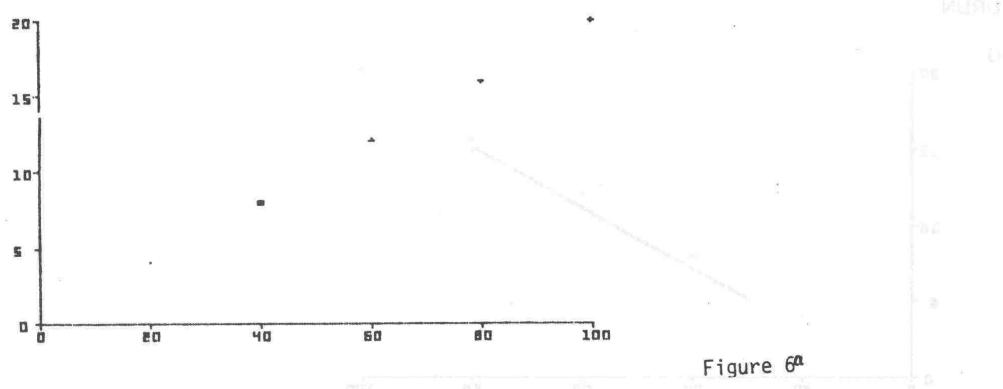


Figure 64

DE6502 KENNER

UPOKE1657,80

ULIST

```
10 REM PLOT CODES
20 HGR : HCOLOR= 3:: SCALE= 1: ROT= 0
30 & HLIN 0,100: & VLIN 0,20
40 FOR I = 1 TO 5
50 & PLOT 20 * I,4 * I,I - 1
60 NEXT : REM ALL PLOT CODES USED
70 REM
80 REM HPLOT LINE FROM 30,5TO80,15
90 DATA 30,5,80,15
100 START = 36990: REM XO IN $907E
110 FOR I = 0 TO 1:H(I) = PEEK (START + I): NEXT
120 REM CONVERT LO AND L/(HI-LO)
130 FOR I = 0 TO 3:START = 768 + 5 * I: GOSUB 200:C(I) = H: READ XY(I): NE
140 FOR I = 0 TO 3
150 J = I / 2 < > INT (I / 2): REM J=0 X; J=1 Y
160 A(I) = H(J) - (XY(I) - C(2 * J)) * C(2 * J + 1)
170 NEXT
180 HPLOT A(0),A(1) TO A(2),A(3) : =TOP : =BLINK : =10D,H : =BH : =O
190 END
200 N = 256:HH = - 1: REM CONVERSION
210 H = PEEK (START + 1) / N: IF H < .5 THEN H = H + .5:HH = 1: &
220 FOR J = 3 TO 5:N = N * 256:H = H + PEEK (START + J) / N: NEXT
230 H = HH * H * 2 ^ (PEEK (START) - 128)
240 RETURN
```

URUN

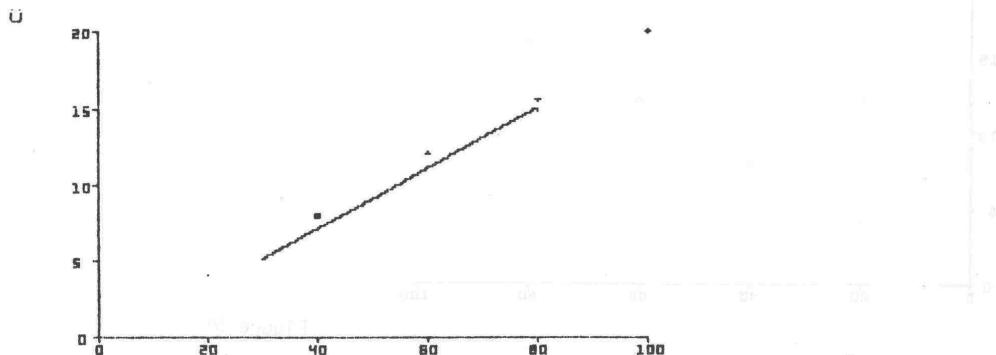


Figure 6b

DE 6502 KENNER

Aanmoedigingspremie 6502-Kenner artikelen over 1984.

Medio 1985 werd door het bestuur een jury samengesteld, dit jaar wederom bestaande uit Fred Harthoorn, Sebo Woldringh en ondergetekende.

De opdracht was uit de jaargang 1984 van de 6502-Kenner de twee beste artikelen te kiezen en hiervoor een eerste en een tweede prijs toe te kennen van respectievelijk f. 250,- en f. 100,-.

Deze prijzen zijn bedoeld als aanmoedigingspremie voor auteurs, of auteurs in spe, om zoveel en zo goed als mogelijk artikelen te doen publiceren in de 6502-Kenner.

Het belangrijkste criterium dat de jury bij haar beoordeling heeft gehanteerd is de praktische toepasbaarheid van het gebodene. Tevens werd scherp gelet op de leesbaarheid van het artikel.

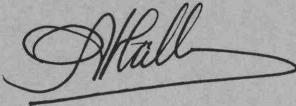
Als tweede is uit de bus gekomen het inleidende artikel over FORTH van Reinier Kleipool (Rotterdam), uit 6502-Kenner nr. 31 (p. 40-44). Een uitstekend verhaal dat naar onze mening best in een van de volgende 6502-Kenners mag worden vervolgd door deze auteur.

Als eerste is uit de bus gekomen het artikel van de heer J.J.A. Janssen (Nijmegen), eveneens uit 6502-Kenner nr. 31 (p. 17-26), betreffende de besturingsprogrammatuur voor de VDU kaart van Elektuur. Een goed voorbeeld, hoe men de 6845 videocontroller chip van Motorola kan aansturen.

Als opbouwende kritiek hierover het volgende. Wat wij missen in de bijbehorende programmatuur zijn een aantal functies, die we op de meeste professionele beeldschermen tegenkomen, zoals: character delete, character insert, line delete, line insert, alternate cursor (bijv. underscore), backward scroll, erase to end of line, erase to end of screen en uiteraard het smooth scrollen. Deze functies zijn ook allemaal mogelijk met de 6845, het moet alleen wel 'even' worden geprogrammeerd.

Ondanks de beperking van de mogelijkheden van deze programmatuur van de heer Janssen, is de jury desalniettemin van mening, dat hij door de omvang van het artikel, de geleverde prestatie en de uitgebreide beschrijving hiervan, in aanmerking komt voor de eerste prijs.

Namens de jury,



Anton Müller

DE 6502 KENNER

```

100 REM *** GRAPHICS IN APPLESOFT ***
110 REM BY H. BOSCH, TWENTE UNIVERSITY OF TECHNOLOGY
120 REM ENSCHEDE, THE NETHERLANDS
130 REM BEFORE USE 'BRUN GRAPHICS.OBJ', AT$9055.$9485, SHAF
135 REM 'BRUN GRAPHICS.OBJ" DONE IN HELLO-PROGRAM
140 TEXT : HOME : PRINT
150 PRINT : PRINT " X Y": PRINT : POKE 34,5
160 IM = 200: DIM X(IM),Y(IM)
170 A = RND (1):E = RND (A) * 10 ^ (3 - 2 * A): REM
180 YL = 0:YH = YL
190 I = I + 1:H = A * I: REM CALCULATE X
200 IF H - X(J) < A * 3 THEN 190: REM DISTANCE BETWEEN X-CORDINATES
210 J = J + 1:A$ = ""
220 IF PEEK (37) > 20 THEN HOME
230 GOSUB 470:X(J) = H: REM STORE X
240 H = E * SIN (H / 3) / (H + A): REM CALCULATE Y
250 IF YL > H THEN YL = H:A$ = " MIN": GOTO 270
260 IF YH < H THEN YH = H:A$ = " MAX"
270 GOSUB 470: PRINT A$
280 Y(J) = H: REM STORE Y
290 IF I < IM THEN 190
300 IM = J:XH = X(J)
310 REM ***** GRAPHICAL PART *****
320 HGR : VTAB 24: SCALE= 1: ROT= 0: HCOLOR= 3
330 & HLIN 0,XH: & VLIN YL -.8 * YH,YH: REM SCALE AND DRAW AXES
340 & HLIN : & VLIN : REM CLOSED GRAPH
350 FOR I = 1 TO IM
360 FOR J = 0 TO 4
370 & PLOT X(I),Y(I) - (J * YH) / 5,J: REM USE ALL PLOTCODES
380 NEXT J,I
390 & DRAW "TEKST LANGS AS",X: REM TEXT ALONG X-AXES
400 & DRAW "RATE CM/S",Y: REM TEXT ALONG Y-AXES
410 & DRAW "PROBE",200,20: REM TEXT AT 200,20
420 FOR I = 0 TO 11: XDRAW 31: NEXT : REM INVERSE MODE
430 ROT= 48: & DRAW "ROTATE",245,60
440 SCALE= 2: FOR J = 1 TO 4: & PLOT XH / (1 + J),YH,J: NEXT : REM EXTR
A PLOT FACILITIES
450 END
460 REM FORMATTING
470 H = INT (H * 100 + .5) / 100:H$ = STR$ (H):L = LEN (H$)
480 FOR K = 1 TO L: IF MID$ (H$,K,1) < > "." THEN NEXT K
490 PRINT " "; SPC (6 - K);H: TAB( 10);
500 RETURN

```

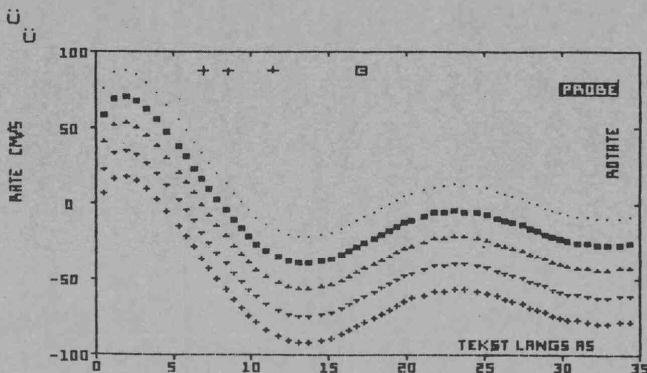


Figure 9