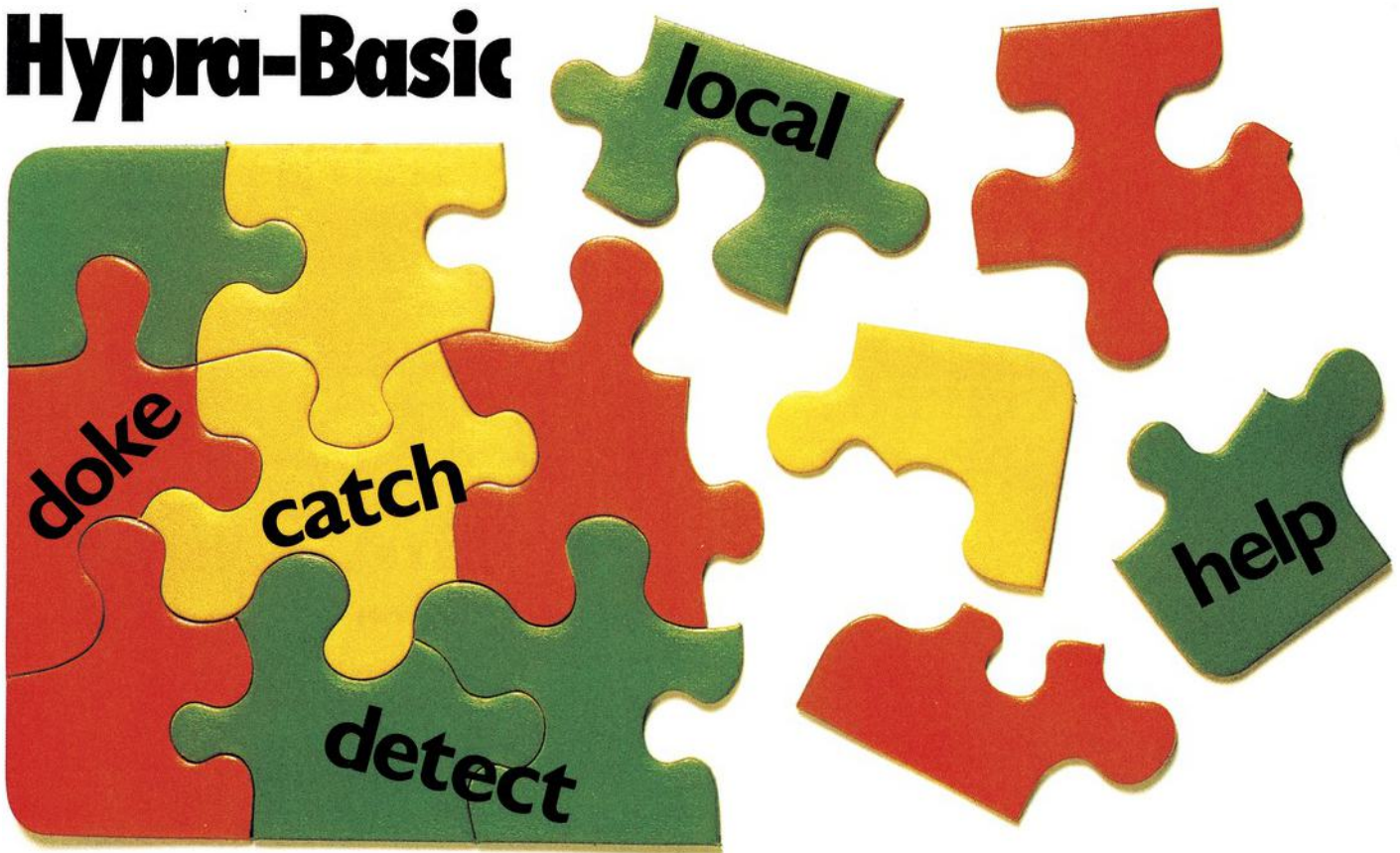


Hypra-Basic



Erstellen Sie sich Ihre eigene Befehlserweiterung mit den Befehlen, die Sie wirklich brauchen. Hypra-Basic ist eine universelle und ausbaufähige Basic-Erweiterung. Es überbietet die Mächtigkeit professioneller Erweiterungen und die Stärke teurer Toolkits bei weitem.

Hyptra-Basic ist mehr als nur eine einfache Erweiterung. Durch seinen modularen Aufbau kann man in Hypra-Basic problemorientierte Erweiterungen erstellen, die die meisten kommerziellen Programme in den Schatten stellen werden. Hypra-Basic ist keine in allen Einzelheiten und Programmsegmenten feststehende Basic-Erweiterung. Vielmehr dient es zum Verknüpfen einzelner Maschinensprache-Routinen. Es ist also als Linker (aus dem englischen »to link« = verbinden, verkettete) zu bezeichnen. Die Vorteile von Hypra-Basic liegen damit klar auf der Hand:

Auch mehrere Programme, die sich im Bereich ab \$C000 befinden, können so gleichzeitig im Speicher liegen. Hypra-Basic verschiebt diese Programme und paßt Adressen automatisch an.

Die erstellten Routinen können im Speicher fast frei verschoben werden.

Der Aufruf der einzelnen Routinen erfolgt, wie im normalen Basic, über von Hypra-Basic frei definierbare Namen (mit Ausnahme der Basic-Befehlsworte) und nicht über SYS-Aufrufe.

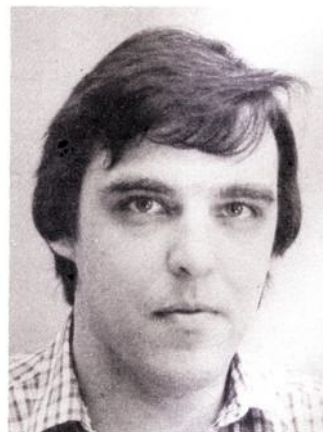
Die einzelnen Befehlserweiterungen können jeweils bestimmten Problemen angepaßt werden. Denkbar sind zum Beispiel Grafik- oder Soundprogramme genauso wie Tools oder genauere Rechenroutinen.

Im Laufe der Zeit können Sie sich eine umfassende Bibliothek anlegen, die Ihnen für alle Eventualitäten eine Lösung anbietet.

Natürlich werden wir in den nächsten Ausgaben noch viele Routinen unserer Leser abdrucken. Die vorgestellten Routinen machen den Anfang.

Ihre Aufgabe soll es sein, für Hypra-Basic Programm-Module zu schreiben. Die besten werden wir veröffentlichen.

(R. Aretz/og)



Lebenslauf

Mein Name ist Richard Aretz, ich erblickte am 12.11.1958 das Licht der Welt. Mein Geburtsort heißt Kempen. Das liegt zirka 30 km von der Stadt Aachen entfernt. Nach dem Besuch der Volksschule (1965 bis 1973) erlernte ich das Handwerk des Heizungsbauers. Im Anschluß an die Gesellenprüfung folgten sieben Jahre in meinem erlernten Beruf. In der Zwischenzeit holte ich auf dem sogenannten 2. Bildungsweg in Form von Abendkursen die mittlere Reife und die Fachhoch-

schulreife nach. Danach, seit 1983, Studium der Elektrotechnik an der Fachhochschule Aachen.

Im Verlauf des 1. Semesters kam ich zum ersten Mal mit Computern in Berührung und erlernte die Programmiersprache Fortran. Das faszinierte mich derart, daß es bis zum eigenen Computer kein weiter Weg war. Seit zirka zwei Jahren besitze ich einen C 64.

Es ist schon erstaunlich, was in einer solch »kleinen« Maschine alles steckt. Bei aller Begeisterung über diese schnell fortschreitende Technik sollte man eines nicht vergessen. Es ist eine Maschine, die keine Gefühle, Launen oder sonstige menschliche Regungen zeigen kann, sondern nur das ausführt, was ein Programmierer ihr vermittelt. Einige Filmemacher scheinen das vergessen zu haben. Solche Filme — Sie wissen sicher, über welche Art Filme ich hier spreche — tragen meines Erachtens nur zur Verunsicherung vieler Menschen bei, die nicht mit der Materie vertraut sind.

Richard Aretz

Hypra-Basic

Stricken Sie sich Ihre persönliche Erweiterung mit Hypra-Basic. Ohne überflüssigen Ballast mitzuführen, können Sie sich mit Hypra-Basic die Erweiterungen selbst herstellen, deren Befehle, Mächtigkeit und Umfang Sie selbst bestimmen können.

Geben Sie zunächst das Programm »BASIC .ASS« (Listing 1) mit dem MSE ein und speichern Sie es auf Diskette. Beachten Sie bitte das Leerzeichen vor dem Punkt. Dann geben Sie »HYPRABASIC« (Listing 2) sowie »DOKUMENT« (Listing 3) und »MG SPEICHER« (Listing 4) mit dem Checksummer ein und speichern diese ebenfalls.

Die Programme haben folgende Funktionen.

»MG SPEICHER« speichert ein Maschinenprogramm nach Angabe von Modulnummer, Start- und Endadresse.

»DOKUMENT« erstellt eine Auflistung der auf Diskette befindlichen Module mit allen wesentlichen Daten.

»HYPRABASIC« ist das Herz des Programms, das das menügesteuerte Verbinden der Module übernimmt.

»BASIC .ASS« enthält zum einen die Umrechnungsroutine, um Maschinenprogramme an andere Bereiche anzupassen. Dahinter befindet sich die Erweiterung für den Interpreter.

Eigene Programme einbinden

Wenn Sie eine eigene Routine in Hypra-Basic einbauen wollen, muß Ihr Unterprogramm folgende Voraussetzungen erfüllen: Start- und Endadresse müssen bekannt sein. Datenbytes, zum Beispiel für eine Fehlermeldung, müssen nach dem Maschinenprogrammteil stehen, also daran angehängt sein. Die Anzahl der Datenbytes muß ebenfalls bekannt sein. Die Befehlsworte der Module dürfen keine Basic-Anweisungen oder Funktionen enthalten. Verbotene Namen sind zum Beispiel »SPRITEON« (enthält das ON aus ON X GOTO/ GOSUB) oder »PRINTAT« (enthält PRINT). Module, die mehrere Befehle beinhalten, müssen in sich geschlossen sein. Es dürfen maximal 16 Befehle in einem Modul enthalten sein. Ein Filename muß folgendes Format haben: "xERW . ASS".

Dabei entspricht x der Nummer des Moduls. Beispiele für Namen sind »4ERW .ASS«, »38ERW .ASS« etc. Beachten Sie im Namen bitte wieder das Leerzeichen vor dem Punkt.

Da eine Diskette maximal 144 Einträge im Directory aufnehmen kann, und für jedes Modul zwei Einträge benötigt werden, sind auf einer Diskette 70 Module das Maximum. Bei höheren Modulnummern wird vom Computer eine andere Diskette verlangt. Bei den Modulnummern von 71 bis 140 also Disk 2, von 141 bis 210 Disk 3 etc.

Die Lage der erstellten Erweiterung im C 64: Die Befehlstablelle und die Interpreter-Erweiterung können nur im RAM-Bereich liegen, der ständig aktiv ist und nicht zu weit unten im Basic-Bereich liegt (dort liegt ja das Hypra-Basic). Diese Bereiche sind:

13200 (\$3390) bis 40959 (\$9FFF) und
49152 (\$C000) bis 53247 (\$CFFF).

Die Befehlsmodule selbst können auch im RAM unter dem ROM liegen. Dadurch erschließt sich ein Bereich von:
13200 (\$3390) bis 53247 (\$CFFF) und
57344 (\$E000) bis 65535 (\$FFFF).

Natürlich dürfen sich die beiden Bereiche nicht überlappen. Deshalb ist es vorteilhaft, wenn die Befehlsmodule hinter der Befehlstablelle liegen.

Starten des Programms: Laden Sie »HYPRABASIC« und starten Sie es. Das Maschinenprogramm »BASIC .ASS« wird daraufhin selbständig nachgeladen.

Eigene Befehle: Mit dem Programm »MG SPEICHER« oder mit einem Monitor auf Diskette gespeicherte Programme

müssen mit einem bestimmten Namen versehen sein (siehe oben). Mit dem Menüpunkt »Parameter eingeben« des Hauptprogrammes müssen Sie die zu jedem Modul folgenden Angaben machen: Die Anzahl der im Modul enthaltenen Befehle, den Namen, mit dem der Befehl aufgerufen wird, zu jedem Namen die Startadresse, die Endadresse und die Anzahl der Datenbytes. Die Parameter werden dann in einem File mit dem Namen »xMODUL« gespeichert. X ist dabei wieder die Nummer des Moduls.

Erweiterungen erstellen: Geben Sie einfach die Nummern der gewünschten Module nacheinander ein. Die Eingabe von »F« storniert das jeweils letzte Modul. Sind alle Erweiterungen im Speicher, beenden Sie die Eingaben mit »E«. Als nächsten Schritt geben Sie die Startadresse der Tabelle sowie der Befehle ein. Wenn die Befehle direkt nach der Tabelle liegen sollen, dann bestätigen Sie bei »Befehle« die vom Computer vorgeschlagene Adresse mit der RETURN-Taste. Daraufhin wird die Erweiterung generiert, jedoch noch nicht gespeichert. Sie steht jetzt im Speicher des Computers an den von Ihnen angegebenen Adressen.

Speichern einer Erweiterung: Mit der Taste F3 gelangen Sie in das Speicher-Menü. Zwei Angaben werden hier von Ihnen erwartet: Der Name, unter dem die Befehlstablelle und die Interpreter-Erweiterung gespeichert wird sowie der Name, unter dem die Befehle gespeichert werden. Wenn Tabelle und Befehle direkt hintereinander liegen, genügt zum Speichern natürlich ein Name. In diesem Fall geben Sie bei »Befehle« nichts ein, sondern tippen einfach nur RETURN. Daraufhin wird die Erweiterung auf Diskette gespeichert.

Protokoll: Alle wichtigen Daten der im Speicher des C 64 liegenden Erweiterung werden noch einmal gezeigt. Eventuell sollten Sie hier noch einmal prüfen, ob es zwischen der Tabelle und den Befehlen keine Überschneidung gegeben hat. Das Protokoll können Sie auf Drucker ausgegeben.

Starten einer Erweiterung: Bringen Sie Ihren C 64 in den Einschaltzustand und laden Sie Ihre Erweiterung mit LOAD "name",8,1. Danach geben Sie NEW ein, um alle Zeiger wieder richtig zu setzen. Ist die Tabelle von den Befehlen getrennt, so laden Sie diese nun nach und geben auch danach wieder ein NEW ein. Der Start der Erweiterung, also das Einbinden in die Interpreterschleife, geschieht durch den Befehl SYS (Startadresse der Tabelle). Nach dem READY des Computers sind die neuen Befehle aktiv.

Hinweise für Programmierer: Wenn Sie selbst in Maschinensprache programmieren und sich einige Module selbst anfertigen wollen, beachten Sie bitte folgende Punkte: Ein Modul muß, bevor es mit Namen aufgerufen wird, auch über einen SYS-Befehl zu erreichen sein. Ein entsprechender Aufruf wäre zum Beispiel: SYS (Adresse) P1,P2,P3.

Vor dem ersten Parameter darf also kein Komma stehen, denn dieser müßte ja sonst nach dem Befehlswort mit eingegeben werden. Achten Sie bitte darauf, daß ein Modul mit mehreren Befehlen in sich geschlossen lauffähig sein muß, also nicht in ein anderes Modul verzweigt.

Module

Die in dieser Ausgabe veröffentlichten Module (Listings 5 bis 13) sind Beispiele, die die Fähigkeiten des Hypra-Basic demonstrieren und den Umgang veranschaulichen sollen. Die Erklärungen und Parameter dazu sind jeweils beim Modul selbst in der Bildunterschrift enthalten.

Bestimmt werden auch Sie Module für Hypra-Basic entwickeln. Die besten Module, die uns zugeschickt werden, werden wir gegen ein entsprechendes Honorar abdrucken. Somit kann sich jeder in kurzer Zeit eine Bibliothek von Basic-Befehlen, Tools und sonstigen nützlichen Routinen anlegen. Erstrebenswert wäre ein Standard-Basic für den C 64, für das dann auch Listings veröffentlicht würden. In diesem Sinne wünschen wir Ihnen viel Erfolg mit Hypra-Basic.

(R. Aretz/og)


```

programm : basic .ass      3138 3383

3138 : 20 fd ae 20 8a ad 20 f7 8d
3140 : b7 a0 00 84 f7 84 f8 b1 c3
3148 : 14 85 02 b9 88 31 c5 02 04
3150 : f0 0a c8 c0 95 d0 f4 a9 96
3158 : 00 85 f7 60 84 a7 c0 1d e7
3160 : b0 07 a9 01 85 f7 4c 87 77
3168 : 31 c0 65 b0 07 a9 02 85 3a
3170 : f7 4c 87 31 a9 03 85 f7 4e
3178 : a9 01 85 f8 a0 01 b1 14 23
3180 : 85 f9 c8 b1 14 85 fa 60 84
3188 : 0a 4a 2a 6a 8a aa 98 a8 41
3190 : ba 9a 08 28 48 68 00 40 e7
3198 : 60 ea 18 38 58 78 b8 d8 58
31a0 : f8 88 c8 ca e8 69 29 c7 7a
31a8 : e0 c0 49 a9 a2 a0 09 e9 97
31b0 : 65 25 06 24 c5 e4 c4 c6 d2
31b8 : 45 e6 a5 a6 a4 46 05 26 8b
31c0 : 66 e5 85 86 84 75 35 16 40
31c8 : d5 d6 55 f6 b5 64 56 15 c1
31d0 : 36 76 f5 95 94 61 21 c1 ce
31d8 : 41 a1 01 e1 81 71 31 d1 72
31e0 : 51 b1 11 f1 91 10 30 50 87

31e8 : 70 90 b0 d0 f0 6d 2d 0e 32
31f0 : 2c cd ec cc ce 4d ee ad 46
31f8 : ae ac 4e 0d 2e 6e ed 8d 5b
3200 : 8e 8c 7d 3d 1e dd de 5d e2
3208 : fe bd bc 5e 1d 3e 7e fd 99
3210 : 9d 79 39 d9 59 b9 be 19 84
3218 : f9 99 20 4c 6c 20 8a ad bd
3220 : 20 f7 b7 a5 14 85 f7 a5 77
3228 : 15 85 f8 20 fd ae 20 8a 2d
3230 : ad 20 f7 b7 a5 14 85 f9 e7
3238 : a5 15 85 fa 20 fd ae 20 15
3240 : 8a ad 20 f7 b7 a5 14 85 ac
3248 : 9b a5 15 85 9c a0 00 a5 c6
3250 : 01 85 02 78 a9 35 85 01 00
3258 : b1 f7 91 9b e6 f7 d0 02 52
3260 : e6 f8 e6 9b d0 02 e6 9c e1
3268 : a5 f8 c5 fa d0 ea a5 f7 45
3270 : c5 f9 00 e4 a5 02 85 01 85
3278 : 58 20 79 00 a8 d0 9e 60 8b
3280 : 20 9e ad 20 82 b7 a6 22 23
3288 : a4 23 20 bd ff 20 fd ae d4
3290 : 20 9e b7 a0 01 20 ba ff fd
3298 : a9 00 20 d5 ff 86 f7 84 21
32a0 : f8 60 20 9e ad 20 82 b7 fa
32a8 : a6 22 a4 23 20 bd ff 20 1d
32b0 : fd ae 20 9e b7 20 ba ff 48

32b8 : 20 fd ae 20 8a ad 20 f7 0d
32c0 : b7 a5 14 85 f7 a5 15 85 0c
32c8 : f8 20 fd ae 20 8a ad 20 73
32d0 : f7 b7 a6 14 e8 d0 02 e6 ba
32d8 : 15 a4 15 a9 f7 20 d8 ff 9e
32e0 : 60 a9 ec 8d 08 03 a9 32 a5
32e8 : 8d 09 03 60 20 73 00 c9 f8
32f0 : 7f 90 06 20 79 00 4c e7 d6
32f8 : a7 a9 82 85 f7 a9 33 85 6a
3300 : f8 a0 00 b1 f7 85 f9 18 42
3308 : 69 05 85 fa e6 f7 d0 02 2a
3310 : e6 f8 b1 f7 f0 dd d1 7a 18
3318 : d0 07 c8 c4 f9 f0 13 d0 4b
3320 : f1 18 a5 f7 65 fa 85 f7 ba
3328 : a5 f8 69 00 85 f8 a0 00 46
3330 : f0 cf b1 f7 85 55 c8 b1 fd
3338 : f7 85 56 c8 18 a5 7a 65 04
3340 : f9 85 7a 90 02 e6 7b a9 45
3348 : 82 85 57 a9 33 85 58 a5 a4
3350 : 01 85 02 78 a9 35 85 01 00
3358 : b1 f7 85 59 c8 b1 f7 85 96
3360 : 5a c8 b1 f7 aa a0 00 b1 9d
3368 : 59 91 57 c8 d0 f9 e6 5a a6
3370 : e6 58 ca 10 f0 a5 02 85 86
3378 : 01 58 20 54 00 4c ae a7 a4
3380 : 00 00 00 00 00 02 da 00 fc

```

Listing 1. »BASIC .ASS« geben Sie bitte mit dem MSE ein

```

5 IF FB<>0 THEN 10 <156>
7 FG=1:LOAD "BASIC .ASS",8,1 <164>
10 DIM A$(100):AN=1:MD=1:POKE 53280,0:POKE <229>
53281,0:F=1:POKE 646,F:MP=12600 <218>
15 POKE 55,56:POKE 56,49 <200>
20 HL$="(39SPACE)" <218>
30 H$="*****" <241>
*** <235>
40 REM *** HAUPTMENUE *****
50 PRINT CHR$(147)TAB(9)"(RVSON,SPACE)HAUP <121>
TMENUE(SPACE,RVOFF,SPACE)HYPR-BASIC":P <189>
RINT H$:AN=1:MD=1 <134>
60 PRINT"FOLGENDE OPTIONEN SIND MOEGlich:" <209>
:PRINT H$:PRINT <171>
70 PRINT TAB(5)"(RVSON)[F1](RVOFF,SPACE)ER <089>
WEITERUNG ERSTELLEN <167>
80 PRINT TAB(5)"(DOWN,RVSON)[F3](RVOFF,SPA <108>
CE)ERWEITERUNG ABSPEICHERN <176>
90 PRINT TAB(5)"(DOWN,RVSON)[F5](RVOFF,SPA <225>
CE)PARAMETER EINGEBEN <047>
100 PRINT TAB(5)"(DOWN,RVSON)[F7](RVOFF,SP <054>
ACE)PARAMETER ANZEIGEN <114>
110 PRINT TAB(5)"(DOWN,RVSON)[F2](RVOFF,SP <063>
ACE)PROTOKOLL":PRINT H$ <198>
115 PRINT TAB(10)"BITTE WAELHEN SIE !!" <020>
120 POKE 198,0:WAIT 198,1:GET S$:S=ASC(S$) <154>
130 IF S<133 OR S>137 THEN 120 <028>
140 S=S-132 <246>
150 ON S GOSUB 210,410,600,800,11000 <010>
160 GOTO 50 <170>
200 REM ***** ERWEITERUNG ERSTELLEN *** <1044>
210 GOSUB 1010 <248>
220 POKE 211,8:POKE 214,5:SYS 58640:PRINT <170>
SIND SIE SICHER ? (J/N)" <194>
230 POKE 198,0:WAIT 198,1:GET S$:IF S<>"J <063>
"OR AN=0 THEN RETURN <094>
240 GOSUB 10010 <199>
250 PRINT CHR$(147)"SIE MUESSEN JETZT ZWEI <246>
SPEICHERBEREICHE" <050>
260 PRINT"AUSWAELHENEN":PRINT:PRINT"1) BER <177>
EICH FUER BEFEHLSTABELLE.DIESER" <077>
270 PRINT TAB(3)"DARF NUR IM 'NORMALEN RAM <051>
LIEGEN." <117>
280 PRINT:PRINT"2) BEREICH FUER DIE ERWEIT <077>
ERUNG.HIER" <051>
290 PRINT TAB(3)"IST DER SPEICHERBEREICH B <117>
ELIEBIG.":PRINT <077>
292 PRINT"FALLS BEIDE PROGRAMMTEILE NACHEI <077>
NANDER" <051>
293 PRINT"STEHEN SOLLN,BITTE BEI 'BEFEHLE <117>
NUR" <077>
294 PRINT"<RETURN> DRUECKEN.":PRINT H$ <077>
295 PRINT"DIE BEFEHLSTABELLE BRAUCHT":HG" <077>
BYTES":PRINT H$ <077>
300 INPUT"1) TABELLE:":SD$:SD=VAL(SD$):IF <077>
SD=0 THEN 250 <077>
305 PRINT TAB(12)SD+HG <077>
310 INPUT"(UP)2) BEFEHLE:":MA$:MA=VAL(MA$) <077>
:MS=MA <077>
320 IF MA=0 THEN RETURN <077>
330 PRINT CHR$(147)"ERWEITERUNG WIRD GENER <077>
IERT":PRINT <077>
340 LA=SD+162+MA:GOSUB 10410 <077>
350 FOR KI=1 TO AN:MOZ=A$(KI):PRINT MOZ, <077>
<117>

360 HM=INT(MOZ/70)+1:IF MD<>HM THEN GOSUB <037>
10210 <060>
370 GOSUB 10660:NEXT:RETURN <114>
400 REM ***** PROGRAMM ABSPEICHERN ** <081>
410 PRINT CHR$(147)TAB(10)"ERWEITERUNG SPE <026>
ICHERN":PRINT H$ <141>
420 PRINT"BEIM ABSPEICHERN KOENNEN ZWEI VE <120>
RSCHIE- DENE " <149>
430 PRINT"FILES ABGELEGT WERDEN.":PRINT <038>
440 PRINT"1) BEFEHLSTABELLE" <144>
450 PRINT"2) MODULE":PRINT <097>
460 PRINT"DIES IST ERFORDERLICH,WENN BEIDE <238>
PRO-" <109>
470 PRINT"GRAMMTEILE UNABHAENGIG VONEINAND <112>
ER" <183>
480 PRINT"EXISTIEREN.FALLS SIE BEIDE PROGR <197>
AMME" <070>
490 PRINT"GEMEINSAM GENERIERT HABEN,GEBEN <155>
SIE UN-" <153>
500 PRINT"TER MODULE EINEN LEERSTRING EIN. <088>
:PRINT H$ <220>
510 INPUT"TABELLE:":NA$ <087>
520 INPUT"BEFEHLE:":NB$:PRINT H$ <187>
525 PRINT"SIND SIE SICHER,DASS DIE RICHTIG <153>
E DIS- " <088>
527 PRINT"KETTE IM LAUFWERK IST? (J/N)" <220>
528 POKE 198,0:WAIT 198,1:GET A$:IF A$="N" <088>
THEN RETURN <220>
530 IF NB$=""THEN SYS(MP+362)"@: "+NA$,8,SD <087>
,MA+1:RETURN <155>
540 SYS(MP+362)"@: "+NA$,8,SD,BT+1 <153>
550 SYS(MP+229)MS,MA+1,MP+2000 <088>
560 CLOSE 1:OPEN 1,8,1,"@: "+NB$:PRINT#1,CH <220>
R$(MS-INT(MS/256)*256):CHR$(MS/256); <088>
570 N=MA+1-MS <220>
580 FOR I=0 TO N-1:PRINT#1,CHR$(PEEK(MP+20 <087>
00+I)):NEXT:CLOSE 1 <140>
590 RETURN <218>
600 REM ***** PARAMETER EINGEBEN *** <058>
610 PRINT CHR$(147)TAB(10)"PARAMETER EINGE <103>
BEN":PRINT H$ <158>
615 INPUT"MODULNUMMER(SSPACE)":MO$ <100>
620 INPUT"WIEVIELE BEFEHLE:":AB <048>
630 PRINT H$ <048>
640 FOR I=1 TO AB <048>
650 POKE 211,0:POKE 214,5:SYS 58640:PRINT <087>
I"(LEFT)) ":INPUT "NAME:":NA$(I):PRIN <168>
T H$ <146>
660 INPUT"(4SPACE)STARTADRESSE:":AS$(I) <016>
670 INPUT"(4SPACE)ENDADRESSE(2SPACE)":AE$ <135>
(I) <228>
680 INPUT"(4SPACE)DATENBYTES(2SPACE)":AD$ <216>
(I) <097>
690 PRINT H$:NEXT <206>
700 PRINT TAB(10)"SIND SIE SICHER (J/N)" <210>
710 POKE 198,0:WAIT 198,1:GET A$:IF A$="N" <005>
THEN RETURN <005>
720 CLOSE 1:OPEN 1,8,1,"@: "+MO$+"MODUL,S" <005>
730 PRINT#1,STR$(AB) <005>
740 FOR I=1 TO AB:PRINT#1,NA$(I):PRINT#1,A <005>
S$(I):PRINT#1,AE$(I):PRINT#1,AD$(I) <005>
750 NEXT:CLOSE 1:RETURN <005>
800 REM ***** PARAMETER ANZEIGEN ***** <005>
810 PRINT CHR$(147)TAB(10)"PARAMETER ANZEI <005>

```

Listing 2. »HYPR-BASIC«, das Hauptprogramm


```

GEN":PRINT H$ <135>
820 INPUT"VON MODULNUMMER:";M1 <135>
830 INPUT"BIS MODULNUMMER:";M2:PRINT H$ <231>
840 FOR K=M1 TO M2:POKE 214,5:SYS 58640 <027>
845 FOR I=1 TO 15:PRINT HL$:NEXT <029>
850 POKE 211,0:POKE 214,5:SYS 58640 <113>
860 MOX=K:GOSUB 10530:PRINT"MODULNUMMER:"K
:PRINT"ANZAHL DER BEFEHLE:"AB:PRINT H$ <085>
870 FOR I=1 TO AB <024>
880 POKE 214,9:SYS 58640:PRINT I"(LEFT) N
AME:"NA$(I);LEFT$(HL$,10):PRINT <122>
890 PRINT"STARTADRESSE:"AS(I);LEFT$(HL$,10
) <122>
900 PRINT"ENDADRESSE(2SPACE):"AE(I);LEFT$(
HL$,10) <010>
910 PRINT"DATENBYTES(2SPACE):"AD(I);LEFT$(
HL$,10) <138>
930 PRINT H$:PRINT TAB(6)"WEITER MIT BELIE
BIGER TASTE" <224>
940 POKE 198,0:WAIT 198,1:NEXT:NEXT:RETURN <108>
999 END <239>
1000 REM ***** MODULE AUSWAELHEN ***** <221>
1010 PRINT CHR$(147)TAB(10)"MODULE AUSWAEL
HEN:" <187>
1020 PRINT H$ <236>
1040 PRINT"BITTE MODULNUMMERN EINGEBEN:" <190>
1050 PRINT H$ <012>
1060 FOR I=1 TO 19:PRINT HL$:NEXT <250>
1070 PRINT H$:PRINT TAB(5)"(RVSON,SPACE)E(
SPACE,RVOFF,SPACE)= ENDE(9SPACE,RVSON
,SPACE)F(SPACE,RVOFF,SPACE)= FEHLER"; <222>
1080 POKE 211,30:POKE 214,2:SYS 58640:POKE
19,1:INPUT MN$:POKE 19,0:PRINT <029>
1090 IF LEFT$(MN$,1)="E"THEN AN=AN-1:RETUR
N <123>
1100 IF LEFT$(MN$,1)="F"AND AN>1 THEN AN=A
N-1:MN$="(3SPACE)":GOTO 1130 <137>
1110 IF LEN(MN$)>3 OR VAL(MN$)=0 THEN 1080 <235>
1120 AZ(AN)=VAL(MN$) <096>
1130 Z=INT(AN/10)+4:S=(AN-(Z-4)*10)*4:POKE
211,S:POKE 214,Z:SYS 58640:PRINT MN$ <046>
1150 IF MN$<"(3SPACE)"THEN AN=AN+1 <021>
1160 GOTO 1080 <085>
10000 REM **** LAENGE DECODE BESTIMMEN *
10010 HG=0:HA=0:AB=0:ZB=0:FOR JI=1 TO AN <204>
10015 HM=INT(AZ(JI)/70)+1:IF HM<MD THEN G
OSUB 10210 <228>
10020 CLOSE 1:OPEN 1,8,0,MID$(STR$(AZ(JI))
,2,4)+"MODUL,S" <054>
10030 INPUT#1,A$:H=VAL(A$):ZB=ZB+H <048>
10040 FOR J=1 TO H <124>
10050 INPUT#1,H1$,H2$(J),H3$(J),H4$(J):HA=
HA+LEN(H1$)+6 <236>
10060 NEXT <164>
10070 HB=(VAL(H3$(H))+VAL(H4$(H)))-VAL(H2$
(1)):IF HB>HG THEN HG=HB <110>
10090 CLOSE 1:NEXT <182>
10100 HG=INT((HG/256)+1)*256:HG=HG+HA+170:
RETURN <249>
10200 REM **** MODULDISK AUSGEBEN **** <061>
10210 H1=PEEK(211):H2=PEEK(214) <226>
10212 FOR X=1984 TO 2023:POKE X-1156,PEEK(
X):POKE X,32:X1=X+1:NEXT <105>
10215 POKE 211,0:POKE 214,24:SYS 58640 <169>
10220 PRINT TAB(3)"(GREY 2)BITTE MODULDISK
NR."HM" EINLEGEN!"; <206>
10230 MD=HM:POKE 646,F:POKE 198,0:WAIT 198
,1 <242>
10240 FOR X=1984 TO 2023:POKE X,PEEK(X-115
6):POKE X+54272,F:NEXT:POKE 211,H1 <141>
10250 POKE 214,H2:SYS 58640:RETURN <159>
10300 REM ***** PROGRAMM LADEN ***** <208>
10310 HI=INT(LA/256):LO=LA-256*HI <019>
10330 SYS(57812)NA$,8,0:POKE 780,0:POKE 78
1,LO:POKE 782,HI:SYS 65493 <071>
10340 RETURN <238>
10400 REM ***** DECODE ROUTINE HOLEN ** <183>
10410 H=LA:LA=SD:SYS(MP+229)13025,13184,SD <203>
10430 LA=H:H=SD+11:HI=INT(H/256):LO=H-256*
HI:POKE SD+1,LO:POKE SD+6,HI <104>
10440 H=SD+161:HI=INT(H/256):LO=H-256*HI:P
OKE SD+25,LO:POKE SD+29,HI <233>
10460 HI=INT(LA/256):LO=LA-256*HI:POKE SD+
103,LO:POKE SD+107,HI <246>
10470 POKE LA-1,0:BT=SD+161:RETURN <069>
10500 REM ***** PARAMETER LADEN ***** <182>
10530 FL$=MID$(STR$(MOX)+"MODUL,S,R",2,13) <058>
10540 CLOSE 2:OPEN 2,8,2,FL$ <120>
10550 INPUT#2,A$:AB=VAL(A$) <063>
10560 FOR JI=1 TO AB:INPUT#2,NA$(JI):INPUT
#2,A$:AS(JI)=VAL(A$) <174>
10570 INPUT#2,A$:AE(JI)=VAL(A$) <214>
10580 INPUT#2,A$:AD(JI)=VAL(A$) <222>
10590 NEXT <186>
10600 CLOSE 2:RETURN <065>
10610 REM **** BEFEHLSTABELLE BEARBEITEN** <208>
10615 YY=LEN(NA$(1)):POKE BT,YY <072>
10620 FOR K=1 TO YY:POKE BT+K,ASC(MID$(NA$
(I),K,1)):NEXT:BT=BT+K <247>
10630 HI=INT(PZ/256):LO=PZ-256*HI:POKE BT,
LO:POKE BT+1,HI <100>
10635 HI=INT(MA/256):LO=MA-256*HI:POKE BT+
2,LO:POKE BT+3,HI <244>
10636 LE=AE(AB)+AD(AB)-AS(1):POKE BT+4,INT
(LE/256):BT=BT+5 <070>
10640 RETURN <028>
10650 REM ***** VERSCHIEB ***** <218>
10660 GOSUB 10530:A$=STR$(MOX)+"ERW .ASS":
NA$=MID$(A$,2,13):GOSUB 10310:PZ=LA <239>

```

```

10670 N1=PZ:FOR I=1 TO AB:GOSUB 10610:N=AS
(I) <216>
10680 SYS MP,PZ:A=PEEK(247) <238>
10690 IF A<3 THEN 10740 <109>
10700 A1=PEEK(247)+256*PEEK(250) <072>
10710 IF A1<AS(1)OR A1>AE(AB)+AD(AB)THEN 1
0750 <184>
10720 A1=A1-AS(1):A1=N1+A1:HI=INT(A1/256):
LO=A1-256*HI <145>
10730 POKE PZ+1,LO:POKE PZ+2,HI <250>
10740 IF A=0 THEN A=1 <076>
10750 PZ=PZ+A:N=N+A:IF N<AE(1)+1 THEN 106
80 <009>
10760 PZ=PZ+AD(1):NEXT <187>
10770 SYS(MP+229)LA,PZ,MA <017>
10780 MA=MA+LE+1:RETURN <165>
11000 REM ***** PROTOKOLL ***** <118>
11010 PRINT CHR$(147)TAB(12)"PROTOKOLL":PR
INT H$ <002>
11020 GN=3 <130>
11025 CLOSE 1:OPEN 1,GN <077>
11030 PRINT#1,"ERWEITERUNG HYPER-BASIC VER
SION 1.1":PRINT#1,H$ <004>
11040 PRINT#1,"ANZAHL DER BEFEHLE:(9SPACE)
";ZB <112>
11050 PRINT#1,"STARTADRESSE BEFEHLSTABELLE
";SD <119>
11060 PRINT#1,"STARTADRESSE DER MODULE:(4S
PACE)";MS:PRINT#1,H$ <012>
11070 PRINT#1,"BYTES FUER TABELLE:(9SPACE)
";HG <008>
11080 PRINT#1,"BYTES FUER BEFEHLE:(9SPACE)
";MA-MS:PRINT#1,H$ <039>
11082 PRINT#1,"DIE ERWEITERUNG ENTHAELT FO
LGENDE MODULE":PRINT#1 <036>
11085 FOR I=1 TO KI-1:PRINT#1,AZ(I);NEXT:
PRINT#1 <053>
11090 IF FK=1 THEN FK=0:GN=3:CLOSE 1:RETUR
N <245>
11100 PRINT#1,H$:PRINT#1,"(2SPACE)AUSGABE
AUF DRUCKER (J/N)" <102>
11110 POKE 198,0:WAIT 198,1:GET A$:IF A$="
N"THEN CLOSE 1:RETURN <011>
11120 GN=4:FK=1:CLOSE 1:GOTO 11025 <181>

```

© 64'er

Listing 2. »HYPER-BASIC«, das Hauptprogramm (Schluß)

```

10 REM ***** DOKUMENT ***** <240>
20 POKE 53280,0:POKE 53281,0:DIM AN$(16),A
S$(16),AE$(16),AD$(16) <112>
30 INPUT"CLR,WHITE)GERAETENUMMER 3=BILD,4
=DRUCKER ";G <035>
40 INPUT"DOWN)STARTMODUL:";SM <149>
50 INPUT"DOWN)ENDMODUL:";EM:PRINT"CLR"; <233>
60 FOR I=SM TO EM:GOSUB 500 <094>
70 OPEN 1,G:I$=STR$(I)+"(2SPACE)" <231>
80 PRINT#1,LEFT$(I$,4):A$=" " <051>
100 FOR K=1 TO AB:IF K>1 THEN A$="(5SPACE
)" <050>
110 PRINT#1,A$:AN$(K)"(8SPACE)"AS$(K)" - "
AE$(K)"(5SPACE)"AD$(K) <227>
120 NEXT <130>
130 PRINT#1,"*****" <164>
*****
135 IF G=3 THEN PRINT"DOWN,9SPACE)WEITER
MIT TASTE !!!":POKE 198,0:WAIT 198,1:PR
INT <213>
140 NEXT <150>
150 IF G=3 THEN 60 <195>
160 END <162>
500 REM **** SUBR. PARAMETER HOLEN **** <205>
505 NA$=MID$(STR$(I),2,4)+"MODUL,S,R" <232>
510 OPEN 1,8,3,NA$ <098>
520 INPUT#1,AB$:AB=VAL(AB$) <213>
530 FOR K=1 TO AB <210>
540 INPUT#1,AN$(K),AS$(K),AE$(K),AD$(K):NE
XT:CLOSE 1:RETURN <004>

```

© 64'er

Listing 3. »DOKUMENT« zeigt die Module einer Diskette an

```

10 POKE 53280,0:POKE 53281,0:PRINT"CLR,WH
ITE)" <192>
20 INPUT"MODULNUMMER:";A$ <069>
30 INPUT"DOWN)STARTADRESSE:";AS <120>
35 PRINT"DOWN)ENDADRESSE DES LETZTEN BEFE
HLS + DATEN-":PRINT"BYTES DES LETZTEN B
EFEHLS!" <016>
40 INPUT"DOWN)";AE:AE=AE+1 <036>
45 INPUT"DOWN,3SPACE)SIND SIE SICHER (J/N
)(2SPACE)E=ENDE ";S$:IF S$="J"THEN 50 <241>
47 IF S$="E"THEN END <044>
48 GOTO 10 <226>
50 H1=INT(AS/256):L1=AS-256*H1 <095>
60 H2=INT(AE/256):L2=AE-256*H2 <248>

```



```

70 SYS(57812)"@:"+A$+"ERW .ASS",8:POKE 193
,L1:POKE 194,H1
80 POKE 174,L2:POKE 175,H2:SYS 62957

```

0 64'er

Listing 4. »MG SPEICHER«. Mit ihm kann man Maschinensprache-Programme auf Diskette bringen. Es geht aber auch mit einem Monitor oder dem MSE.

programm : 1erw .ass c000 c05b

```

c000 : a5 39 8d 4c c0 a5 3a 8d 6c
c008 : 4d c0 a5 7a 85 f7 a5 7b 14
c010 : 85 f8 a5 f7 85 7a a5 f8 2e
c018 : 85 7b a9 00 85 c6 20 87 e3
c020 : ea a5 c6 f0 f5 ad 77 02 5b
c028 : 85 02 4c 35 c0 20 79 00 5b
c030 : f0 e0 20 fd ae 20 8a ad ca
c038 : 20 f7 b7 20 fd ae 20 9e 59
c040 : b7 e4 02 d0 e8 a9 00 85 eb
c048 : c6 4c a3 a8 00 00 ad 4c 82
c050 : c0 85 14 ad 4d c0 85 15 a9
c058 : 4c a3 a8 ff 00 ff 00 fd 9c

```

Listing 5. Modulnummer: 1
Anzahl der Befehle: 2

1) KEY 49152-49227 2

Funktion: Sprung in angegebene Zeilennummer nach Tastendruck mit nachfolgendem ASCII-Code.

Syntax: KEY zeinr,ascii,(zeinr,ascii...)

Anwendung: Bei Menüsteuerungen, keine Fehleingabe möglich, da nur Tasten, die im Befehlsaufruf stehen, beachtet werden; alle anderen werden ignoriert.

Beispiel: KEY 12000,65,13000,66,14000,67

In diesem Beispiel wird beim Drücken der Taste 65 (A) in die Zeile 12000 verzweigt, bei 66 (B) nach 13000 etc.

2) RETKEY 49230-49242 0

Funktion: Erreicht das Programm diesen Befehl, wird in die Key-Zeile zurückgesprungen ähnlich einem RETURN.

Syntax: RETKEY keine Parameter

programm : 2erw .ass c000 c0c3

```

c000 : 20 9e b7 e0 19 30 03 4c 31
c008 : 85 c0 8e 3c 03 86 f7 20 9d
c010 : fd ae 20 9e b7 e0 28 10 83
c018 : 6c 8e 3d 03 20 fd ae 20 68
c020 : 9e b7 e0 19 10 5f e8 8e b2
c028 : 3e 03 20 fd ae 20 9e b7 85
c030 : e0 28 10 51 e8 8e 3f 03 59
c038 : ad 3c 03 cd 3e 03 30 02 3f
c040 : 10 43 ad 3d 03 cd 3f 03 a6
c048 : 30 02 10 39 ad 3d 03 85 80
c050 : d3 ad 3c 03 85 d6 20 10 19
c058 : e5 ac 3d 03 b1 d1 10 04 35
c060 : 29 7f 10 02 09 80 91 d1 0c
c068 : c8 cc 3f 03 d0 ee 18 a5 f7
c070 : d1 69 28 90 02 e6 d2 85 c0
c078 : d1 ee 3c 03 ad 3c 03 cd 94
c080 : 3e 03 d0 d5 60 a2 0e 4c 1b
c088 : 37 a4 20 00 c0 a9 00 85 7e
c090 : c6 85 a7 20 fd ae 20 9e 1a
c098 : b7 86 f8 a6 f8 a5 f7 8d 5d
c0a0 : 3c 03 a0 00 ea ea c8 d0 51
c0a8 : fb ca d0 f6 20 4c c0 e6 50
c0b0 : a7 a5 c6 f0 e6 a5 a7 4a c8
c0b8 : b0 08 a5 f7 8d 3c 03 20 dc
c0c0 : 4c c0 60 00 ff 00 ff 00 84

```

Listing 6. Modulnummer: 2
Anzahl der Befehle: 2

1) INVERS 49152-49289 0

Funktion: Invertieren eines rechteckigen Bildschirm-Ausschnitts.

Syntax: INVERS AZ,AS,EZ,ES

Beispiel: INVERS 2,3,10,25

Der Bildschirm-Bereich Zeile 2, Spalte 3 (obere linke Ecke) bis Zeile 10, Spalte 25 (untere Ecke) werden invertiert.

2) BLINK 49290-49346 0

Funktion: Rechteckiger Bildschirm-Ausschnitt blinkt mit angegebener Geschwindigkeit.

Syntax: BLINK AZ,AS,EZ,ES,SPEED

Beispiel: BLINK 2,3,10,25,100

Der Bildschirmbereich wie unter (1) blinkt mit der Geschwindigkeit 100 (1-255), je größer die Zahl SPEED, desto langsamer blinkt der Bildschirm.

programm : 3erw .ass c000 c16b

```

c000 : 20 9e b7 8e 3c 03 20 fd 87
c008 : ae 20 9e b7 8e 3d 03 20 84
c010 : fd ae 20 9e b7 8e 3e 03 2f
c018 : 20 fd ae 20 9e b7 86 f7 98
c020 : ad 3c 03 85 d6 ad 3d 03 33
c028 : 85 d3 20 10 e5 20 24 ea 67
c030 : ac 3d 03 a2 40 b1 d1 c9 fc
c038 : 5d d0 02 a2 5b 8a 91 d1 c6
c040 : a5 f7 91 f3 c8 a2 40 ce 04
c048 : 3e 03 d0 e9 60 20 9e b7 6a
c050 : 8e 3c 03 20 fd ae 20 9e d4
c058 : b7 8e 3d 03 20 fd ae 20 f3
c060 : 9e b7 8e 3e 03 20 fd ae cc
c068 : 20 9e b7 86 f7 ad 3c 03 7a
c070 : 85 d6 ad 3d 03 85 d3 20 5f
c078 : 10 e5 20 24 ea a2 5d 18 71
c080 : ad 3c 03 6d 3e 03 c9 1a 11
c088 : 10 29 ac 3d 03 b1 d1 c9 98
c090 : 40 d0 02 a2 5b 8a 91 d1 01
c098 : a5 f7 91 f3 18 a5 d1 69 e4
c0a0 : 28 90 04 e6 d2 e6 f4 85 31
c0a8 : d1 85 f3 a2 5d ce 3e 03 d8
c0b0 : d0 d8 60 a2 0e 4c 37 a4 c2
c0b8 : 20 9e b7 8e 3f 03 20 fd 70
c0c0 : ae 20 9e b7 8e 40 03 20 54
c0c8 : fd ae 20 9e b7 8e 41 03 f3
c0d0 : 20 fd ae 20 9e b7 8e 42 05
c0d8 : 03 20 fd ae 20 9e b7 86 23
c0e0 : f7 ad 3f 03 8d 3c 03 ad 00
c0e8 : 40 03 8d 3d 03 ad 41 03 5e
c0f0 : 8d 3e 03 20 20 c0 ad 42 a4
c0f8 : 03 8d 3e 03 20 6d c0 18 52
c100 : ad 3f 03 6d 42 03 aa ca 38
c108 : 8e 3c 03 ad 41 03 8d 3e 0a
c110 : 03 20 20 c0 ad 3f 03 8d 3f
c118 : 3c 03 18 ad 41 03 6d 40 f4
c120 : 03 aa ca 8e 3d 03 ad 42 24
c128 : 03 8d 3e 03 20 6d c0 ad ae
c130 : 3f 03 85 d6 ad 40 03 85 21
c138 : d3 20 10 e5 a4 d3 a9 70 4c
c140 : 91 d1 98 18 6d 41 03 48 60
c148 : a8 88 a9 6e 91 d1 18 ad d0
c150 : 3f 03 6d 42 03 aa ca 86 72
c158 : d6 20 10 e5 ac 40 03 a9 2b
c160 : 6d 91 d1 68 a8 88 a9 7d 88
c168 : 91 d1 60 ff 00 ff 00 ff fa

```

Listing 7. Modulnummer: 3
Anzahl der Befehle: 3

1) WLINE 49152-49228 0

Funktion: Zeichnet eine waagrechte Linie in einen LOW-RES-Bildschirm. Eventuelle Kreuzungen mit senkrechten Linien werden berücksichtigt.

Syntax: WLINE AZ,AS,Länge,Farbe

Beispiel: WLINE 2,4,10,1

Mit diesem Beispiel wird eine waagrechte Linie, die im Punkt (2,4) beginnt und eine Länge von 10 besitzt, mit der Farbe 1 (weiß) gezeichnet. Dieser wie auch die folgenden beiden Befehle beziehen sich auf einen LOW-Resolution-Bildschirm. Sie eignen sich besonders für eine Menüaufbereitung oder zur Anzeige von Tabellen.

2) SLINE 49229-49335 0

Funktion: Zeichnet senkrechte Linie in LOW-RES-Bildschirm.

Syntax: SLINE AZ,AS,Höhe,Farbe

Beispiel: SLINE 2,4,10,14

Es wird eine senkrechte Linie, Startpunkt (2,4) und Höhe 10 mit der Farbe 14 (hellblau) gezeichnet.

3) RELOW 49336-495140

Funktion: Zeichnet ein Rechteck in einem LOW-RES-Bildschirm.

Syntax: RELOW AZ,AS,Breite,Höhe,Farbe

Beispiel: RECLW 2,3,10,5,1

Zeichnen eines Rechtecks mit dem Startpunkt (2,3 = obere linke Ecke), der Breite 10, der Höhe 5 und der Farbe 1 (weiß).

programm : 4erw .ass c000 c063

```
c000 : 20 8a ad 20 f7 b7 a5 14 d1
c008 : 85 f7 a5 15 85 f8 20 fd 31
c010 : ae 20 8a ad 20 f7 b7 a5 12
c018 : 14 85 f9 a5 15 85 fa 20 cc
c020 : fd ae 20 8a ad 20 f7 b7 f9
c028 : a5 14 85 9b a5 15 85 9c fe
c030 : a0 00 a5 01 85 02 78 a9 f7
c038 : 35 85 01 b1 f7 91 9b e6 ef
c040 : f7 d0 02 e6 f8 e6 9b d0 d3
c048 : 02 e6 9c a5 f8 c5 fa d0 e4
c050 : ea a5 f7 c5 f9 d0 e4 a5 c8
c058 : 02 85 01 58 20 79 00 a8 87
c060 : d0 9e 60 3e 03 20 fd ae e6
```

Listing 8. Modulnummer: 4

Anzahl der Befehle: 1

1) TRANS 49152-49250 0

Funktion: Verschieben eines Speicherbereiches. Dieser Befehl bezieht sich nur auf das RAM, das heißt es kann auch das RAM unter dem Betriebssystem oder Basic-Interpreter benutzt werden.

Syntax: TRANS Staradr., Endadr., Startadresse neu

Beispiel: TRANS 1024,2023,40960

Der Bildschirmspeicher wird unter den Basic-Interpreter geschoben.

programm : 5erw .ass c000 c094

```
c000 : 20 76 c0 20 8a ad 20 f7 16
c008 : b7 20 fd ae 20 9e b7 86 07
c010 : 02 a2 00 20 73 00 f0 06 6e
c018 : 9d 3c 03 e8 d0 f5 a5 01 06
c020 : 85 9e 78 a9 35 85 01 a0 0c
c028 : 00 b1 14 9d 3c 03 e8 c8 cb
c030 : c4 02 d0 f5 a5 9e 85 01 4f
c038 : 58 20 83 c0 20 a5 a9 a5 ba
c040 : 9b 85 7a a5 9c 85 7b 60 96
c048 : 20 76 c0 20 8a ad 20 f7 5e
c050 : b7 20 fd ae 20 28 af 20 af
c058 : 82 b7 85 02 a5 01 85 9b 07
c060 : 78 a9 35 85 01 a0 00 b1 23
c068 : 33 91 14 c8 c4 02 d0 f7 11
c070 : a5 9b 85 01 58 60 a2 00 77
c078 : a9 00 9d 3c 03 e8 e0 c4 95
c080 : d0 f8 60 a5 7a 85 9b a5 27
c088 : 7b 85 9c a9 3c 85 7a a9 4f
c090 : 03 85 7b 60 5b 8a 91 d1 35
```

Listing 9. Modulnummer: 5

Anzahl der Befehle: 2

1) RAMVAR 49152-49223 0

Funktion: Bringt ASCII-Zeichen aus dem RAM in eine Stringvariable.

Syntax: RAMVAR Anfangsadresse, Länge, var\$=''

Beispiel: RAMVAR 40960,10,a\$=''

Der Inhalt der Speicherzellen 40960-40969 wird in die Variable a\$ gebracht. Dieser Befehl bezieht sich nur auf den RAM-Bereich.

2) VARRAM 49224-49269 30

Funktion: Umgekehrte Funktion wie (1). Der Inhalt der Variablen wird in Form der ASCII-Codes ins RAM befördert.

Syntax: VARRAM adresse,variable\$

Beispiel: VARRAM 40960,a\$

Der Inhalt der Variablen a\$ wird ab 40960 ins RAM geschrieben.

programm : 6erw .ass c000 c0fb

```
c000 : 20 28 af a5 47 85 f7 a5 a1
c008 : 48 85 f8 a0 00 b1 f7 99 06
c010 : 3c 03 c8 c0 05 d0 f6 20 0b
```

```
c018 : fd ae 20 28 af a0 00 b1 dd
c020 : 47 91 f7 c8 c0 05 d0 f7 ae
c028 : a0 00 b9 3c 03 91 47 c8 2a
c030 : c0 05 d0 f6 60 20 28 af 8d
c038 : a5 46 48 a5 45 48 20 c4 68
c040 : c0 a9 02 f0 26 a5 f7 85 ee
c048 : f9 a5 f8 85 fa 20 fd ae 09
c050 : 20 28 af 20 c4 c0 a9 02 71
c058 : f0 11 a0 00 68 91 f7 a5 37
c060 : 45 91 f9 c8 68 91 f7 a5 43
c068 : 46 91 f9 60 20 28 af a5 4f
c070 : 46 48 a5 45 48 20 c4 c0 06
c078 : a5 f7 85 f9 48 a5 f8 85 5a
c080 : fa 48 a5 02 f0 30 20 fd 55
c088 : ae 20 28 af 20 c4 c0 a0 b3
c090 : 02 b1 f7 85 9b c8 b1 f7 d0
c098 : 85 9c a0 00 a0 00 b1 f7 54
c0a0 : 91 f9 e6 f7 d0 02 e6 f8 91
c0a8 : e6 f9 d0 02 e6 fa c6 9b 98
c0b0 : d0 ea c6 9c 10 e6 68 85 1f
c0b8 : f8 68 85 f7 68 91 f7 c8 c9
c0c0 : 68 91 f7 60 a6 2f a5 30 d6
c0c8 : 86 f7 85 f8 c5 32 d0 04 03
c0d0 : e4 31 f0 1d a0 00 b1 f7 ed
c0d8 : c8 c5 45 d0 06 a5 46 d1 39
c0e0 : f7 f0 13 c8 b1 f7 18 65 33
c0e8 : f7 aa c8 b1 f7 65 f8 90 4c
c0f0 : d7 a9 00 85 02 60 a9 01 18
c0f8 : 85 02 60 03 20 6d c0 18 97
```

Listing 10. Modulnummer: 6

Anzahl der Befehle: 3

1) SWAPVA 49152-49204 0

Funktion: Vertauschen zweier Variablenwerte.

Syntax: SWAPVA var1,var2

Beispiel: SWAPVA as,k

Die Werte der Variablen as und k werden vertauscht.

2) SWAPAR 49205-49259 0

Funktion: Wie (1), nur für Arrays.

Syntax: SWAPAR arr1,arr2

Beispiel: SWAPAR d\$,g\$

Der gesamte Inhalt des Array d\$(n) wird mit dem des Array g\$(n) vertauscht. Es ist zu beachten, daß beide Arrays vorher dimensioniert sein müssen.

3) DUPEAR 49260-49347 55

Funktion: Erzeugt zwei identische Arrays mit angegebenem Namen. Array 1 ist das »Mutterarray«.

Syntax: DUPEAR array1,array2

Beispiel: DUPEAR as,m

Das Array as wird komplett in das Array m übertragen. Es ist darauf zu achten, daß beide Arrays dimensioniert sein und gleiche Größe haben müssen. Anwendung findet dieser Befehl zum Beispiel beim Aufruf von Unterprogrammen, die nachträglich an ein Hauptprogramm angepaßt werden müssen.

programm : 7erw .ass c000 c00b

```
c000 : a9 e4 8d 08 03 a9 a7 8d b7
c008 : 09 03 60 a0 00 b1 f7 99 60
```

Listing 11. Modulnummer: 7

Anzahl der Befehle: 1

1) OFF 49152-49162 0

Funktion: Schaltet Erweiterung ab.

Syntax: OFF keine Parameter

programm : 8erw .ass c000 c153

```
c000 : 20 9e b7 a9 00 48 4c 10 26
c008 : c0 48 20 fd ae 20 9e b7 8a
c010 : 68 a8 8a 99 a7 00 c8 98 71
c018 : c0 05 d0 ed a5 a8 85 d3 aa
c020 : a5 a9 85 d6 20 10 e5 20 30
c028 : 24 ea a5 a7 d0 48 18 a5 1b
c030 : d1 69 28 85 f7 85 f9 a5 4f
c038 : d2 69 00 85 f8 18 a5 f8 48
c040 : 69 d4 85 fa a4 a8 b1 f7 1a
c048 : 91 d1 b1 f9 91 f3 c8 c4 d3
c050 : aa 30 f3 18 a5 d1 69 28 f1
c058 : 85 d1 a5 d2 69 00 85 d2 dc
c060 : 20 24 ea e6 a9 a5 a9 c5 24
c068 : ab d0 c3 85 d6 20 10 e5 97
```



```

c070 : 20 fd ae 4c a4 aa c9 01 8d
c078 : d0 50 e6 aa a5 a8 85 d3 dd
c080 : a5 ab 85 d6 20 10 e5 20 91
c088 : 24 ea 38 a5 d1 e9 28 85 fc
c090 : f7 85 f9 a5 d2 e9 00 85 04
c098 : f8 18 a5 f8 69 d4 85 fa 6e
c0a0 : a4 a8 b1 f7 91 d1 b1 f9 66
c0a8 : 91 f3 c8 c4 aa 30 f3 38 6a
c0b0 : a5 d1 e9 28 85 d1 a5 d2 e0
c0b8 : e9 00 85 d2 20 24 ea c6 b9
c0c0 : ab a5 ab c5 a9 d0 c3 4c aa
c0c8 : 6b c0 c9 02 d0 3c e6 ab 28
c0d0 : a4 a8 c8 a5 d1 85 f7 18 08
c0d8 : a5 d2 69 d4 85 f8 b1 d1 66
c0e0 : 48 b1 f7 88 91 f7 68 91 ad
c0e8 : d1 c8 c8 c4 aa d0 ef 88 ea
c0f0 : a9 20 91 d1 18 a5 d1 69 11
c0f8 : 28 85 d1 a5 d2 69 00 85 8f
c100 : d2 e6 d6 a5 d6 c5 ab d0 9b
c108 : c7 60 c6 aa e6 ab a5 aa be
c110 : 85 d3 a5 a9 85 d6 20 10 cd
c118 : e5 a5 d1 85 f7 a5 d2 18 1d
c120 : 69 d4 85 f8 a4 aa 88 b1 99
c128 : d1 48 b1 f7 c8 91 f7 68 52
c130 : 91 d1 88 88 c4 a8 10 ef 8f
c138 : c8 a9 20 91 d1 18 a5 d1 27
c140 : 69 28 85 d1 a5 d2 69 00 ef
c148 : 85 d2 e6 d6 a5 d6 c5 ab 4a
c150 : d0 c7 60 42 03 a9 ca 86 22

```

Listing 12. Modulnummer: 8
Anzahl der Befehle: 1

1) SCROLL 49152-49490 0
Funktion: Scrollt rechteckigen Bildschirmausschnitt in beliebige Richtung.
Syntax: SCROLL mode,as,az,es,ez,(var\$)
Beispiel: SCROLL 1,2,4,5,10,a\$
Hier wird das angegebene Fenster nach unten gescrollt. In die frei gewordene erste Zeile wird der String a\$ ge-printed.
mo=0 nach oben
mo=1 nach unten

mo=2 nach links

Bei mo=2 und mo=3 entfällt der String am Schluß des Aufrufs.

programm : 9erw .ass c000 c0de

```

c000 : a5 2d 85 f7 a5 2e 85 f8 70
c008 : a0 00 84 c6 b1 f7 10 47 4c
c010 : 29 7f 20 d2 ff c8 b1 f7 58
c018 : 29 7f 20 d2 ff a9 25 20 85
c020 : d2 ff a9 3d 20 d2 ff c8 2e
c028 : b1 f7 85 f9 c8 b1 f7 85 7a
c030 : fa a5 f9 10 18 38 a9 ff 67
c038 : e5 fa 85 fa a9 ff e5 f9 81
c040 : 85 f9 a9 2d 20 d2 ff e6 38
c048 : fa d0 02 e6 f9 a5 f9 a6 0a
c050 : fa 20 cd bd 4c b8 c0 20 53
c058 : d2 ff c8 b1 f7 10 3a 29 cd
c060 : 7f 20 d2 ff a9 24 20 d2 86
c068 : ff a9 3d 20 d2 ff a9 22 a7
c070 : 20 d2 ff a0 02 b1 f7 f0 7d
c078 : 18 85 a7 c8 b1 f7 85 22 8b
c080 : c8 b1 f7 85 23 a0 00 b1 6a
c088 : 22 20 d2 ff c8 c6 a7 d0 72
c090 : f6 a9 22 20 d2 ff 18 90 96
c098 : 1f 20 d2 ff a9 3d 20 d2 26
c0a0 : ff 18 a5 f7 69 02 48 a5 27
c0a8 : f8 69 00 a8 68 20 a2 bb f3
c0b0 : 20 bc aa a9 91 20 d2 ff 73
c0b8 : 18 a5 f7 69 07 85 f7 a5 96
c0c0 : f8 69 00 85 f8 a9 0d 20 6f
c0c8 : d2 ff a5 f8 c5 30 d0 07 51
c0d0 : a5 f7 c5 2f 30 01 60 a5 a0
c0d8 : c6 f0 fc 4c 08 c0 b1 d1 d0

```

Listing 13. Modulnummer: 9
Anzahl der Befehle: 1

1) DUMP 49152-49373 0
Funktion: Zeigt aktuellen Variablenspeicher auf dem Bildschirm an. Nach jeder Ausgabe kann mit SPACE weitergemacht werden.
Syntax: DUMP keine Parameter

Schwarz auf weiß

Mit den folgenden Änderungen zum Programm »Database« (Sonderheft 7/85) ist es nun auch möglich, Datensätze auf dem Drucker auszugeben. Außerdem können Sie leicht eigene Ideen verwirklichen, zum Beispiel Etikettendruck oder ähnliches.

Um die erforderlichen Änderungen an Database vorzunehmen, gehen Sie bitte wie folgt vor:
1. Bitte laden Sie das Programm »DBII« (Datenpflege).
2. Anschließend geben Sie die in Listing 1 enthaltenen Programmzeilen ein. Bei den Zeilen 330 und 3500 müssen die einzelnen Befehle abgekürzt eingegeben werden (siehe auch Anhang D in der Bedienungsanleitung Ihres C 64).
3. Zum Schluß speichern Sie das geänderte Programm — nachdem Sie die alte Version gelöscht haben — mit »SAVE »DBII«, 8« wieder auf Diskette.

Die neuen Kommandos

Neu hinzugekommen sind drei Kommandos im Menüpunkt »Daten ausgeben« (im Teil »Datenpflege«).

Das erste Kommando erreichen Sie durch Drücken der Taste »C«. Mit ihm ist es möglich, den gerade im Speicher befindlichen Datensatz noch einmal einzulesen und auf dem Bildschirm auszugeben. Dabei kann auch zwischenzeitlich in einen anderen Programmteil verzweigt werden.

Ein Beispiel: Sie haben über die Indexsuche einen bestimmten Datensatz selektiert. Nun möchten Sie — aus wel-

chen Gründen auch immer — in den Programmteil »directory ausgeben« verzweigen. Wenn Sie dann anschließend wieder in die Datenausgabe zurückkehren und mit dem zuvor selektierten Datensatz weiterarbeiten wollen, blieb bisher nur die Möglichkeit, diesen nochmals suchen zu lassen. Jetzt drücken Sie einfach die »C«-Taste und der betreffende Datensatz wird wieder in Ihren C 64 geladen und auf dem Bildschirm ausgegeben!

Besonders interessant wird der »C«-Befehl auch im Zusammenhang mit dem zweiten Kommando, das die Ausgabe von Datensätzen auf den Drucker ermöglicht: Durch Drücken der Taste »P« wird die Druckerausgabe ein-, durch Drücken von »O« wieder ausgeschaltet. Außer im Programmteil »Daten ausgeben« kann die Druckerausgabe auch im Hauptmenü (mit denselben Tasten) ein- oder ausgeschaltet werden. Im Hauptmenü wird die eingeschaltete Druckerausgabe durch den Schriftzug »Ausgabe auf Drucker!« (im unteren Bildschirmbereich); im Teil »Daten ausgeben« durch ein »P« in der Kopfzeile (anstelle des »/« nach dem Dateinamen) angezeigt.

Wenn Sie jetzt — bei eingeschalteter Druckerausgabe — einen Datensatz in den Computer laden, so erfolgt die Ausgabe gleichzeitig auf Drucker und Bildschirm! Bei der Druckerausgabe wird dabei jedes Datenfeld in einer neuen Zeile zu Papier gebracht. Am Ende eines Datensatzes werden jeweils zwei Leerzeilen (zur Abtrennung mehrerer Datensätze voneinander) gedruckt.

Um auf den »C«-Befehl zurückzukommen: Wenn Sie sich nicht sicher sind, ob Sie einen bestimmten Datensatz ausdrucken wollen oder nicht, so laden Sie diesen zunächst bei ausgeschalteter Druckerausgabe. Haben Sie sich dann für einen Ausdruck entschieden, schalten Sie jetzt mit »P« die Druckerausgabe ein und drücken anschließend »C« und schon wird der Datensatz ausgedruckt! Danach können Sie mit »O« die Druckerausgabe wieder ausschalten, um nach weiteren Datensätzen zu suchen.