

CBM/PET NEWS

Informa Verlag AG
Postfach 1401
CH-6000 Luzern 15

3. Jahrgang
Erscheint 6 mal jährlich
ISSN 0251-2017

Commodore-Anwenderzeitschrift
Jahresbezugspreis inkl. Versand und Porto
SFr. 48.-/DM 55.-/öS 400

Lieber Commodore-Freund

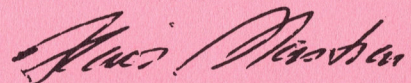
Erinnern Sie sich noch? Erst knapp drei bis vier Jahre ist es her, seit die "Kleincomputerei" bei uns so richtig losgegangen ist. Vielleicht gehören auch Sie zu jenen "Mikro-Fans der ersten Stunde", die nach dem Kauf ihres Kleincomputers feststellen mussten, dass sie bei den auftauchenden Fragen und Problemen so ziemlich ohne Beratung und Hilfe dastanden. Wissen Sie noch, wie Sie fast verzweifelt nach deutschsprachigen Unterlagen und Orientierungshilfen gesucht haben und sich dabei in kühnen Träumen sogar eine spezifische Fachzeitschrift gewünscht haben?

Ein erster Lichtblick, der diesen Informationsmangel einigermaßen erhellte, war immerhin z.B. die damalige Clubzeitschrift des Schweizer Computer Club, aus der heraus dann erstmals im Januar 1980 die "PET NEWS" als eigenständige Commodore-Anwenderzeitschrift entstanden ist. Inzwischen hat sich diese Publikation zu einer vielbeachteten und anerkannten Fachschrift entwickelt.

Die ständig steigende Nachfrage nach den CBM/PET NEWS als vertiefende und fundierte Information zu diesem heute wohl am weitest verbreiteten Kleincomputersystem im deutschsprachigen Raum, zeigt uns, dass wir mit unserem redaktionellen Konzept richtig liegen. Für Sie als Leser und Anwender kommt als wichtiges Kriterium hinzu, dass wir unsere Informationen und Tests völlig unabhängig von Hersteller und Lieferanten bearbeiten, was im Gegensatz zu diversen ähnlich gelagerten Zeitschriften nicht immer der Fall ist.

Quasi als Symbol für noch frischeren Wind haben wir zum Anlass der ersten Ausgabe im 3. Jahrgang einen neuen Schriftzug für die CBM/PET NEWS geschaffen. Als weitere Neuerung werden wir ab sofort in bescheidenem Rahmen Inserate aufnehmen, denn auch das ist schlussendlich Information für Sie. Und zum Schluss noch eine Bitte. Schreiben Sie uns, wenn Sie irgendwo nicht weiter wissen, vielleicht kann einer Ihrer Mitleser helfen. Auch an fertigen Manuskripten haben wir Interesse; Sie sind herzlich zur redaktionellen Mitarbeit eingeladen.

Viel Freude mit Commodore wünscht Ihnen im Namen der Redaktion



Heinz Kastien

Inhaltsverzeichnis

- | | | | |
|----|----------------------------------|----|----------------------------------|
| 3 | Listschutz CBM 8000 | 15 | Programm des Monats Multiplotter |
| 6 | Eingabeprogramm Lagerbuchhaltung | 19 | Wussten Sie schon... |
| 7 | Eingabeprogramm Adressverwaltung | 20 | Barcodeeinleseprogramm |
| 9 | Leichte Kost Wilhelm Tell | 24 | Neuheiten Hexadezimal |
| 13 | Barcodeleser | 25 | Inhaltsverzeichnis 1981 |

Einen ganzen Monat lang können Sie unsere Anlage kostenlos testen!

Exklusiv von Compu Life:
Schweizer Normtastatur!



Ganzes Bildschirm-
Textsystem inkl.
Wust bereits ab

Fr. 12'330.-

Bearbeiten und verarbeiten Sie ab sofort Ihre Texte auf dem bequemen und bedienungsfreundlichen Ein-Platz-System von Compu Life.

Computer, Typ CBM 8032 - TT
Commodore-Computer mit exklusiv von Compu Life entwickelter Schweizer Normtastatur.

Speichergerät (Doppel-Floppy)
Speicherkapazität min. 450 voll beschriebene A4-Seiten (ohne Diskettenwechsel).

Typenrad-Schönschreibdrucker
Mit Schweizer Textverarbeitungs-Modul für alle deutschen, französischen, italienischen und englischen Zeichen.

Software
Spezielles Software-Angebot für Industrie, Handel, Gewerbe (auch Kleingewerbe).

System Einführung

Gründliche System-Einführung inkl. Betriebsanleitung. Falls Sie bereits ein CBM 8032-Modell besitzen, können Sie dieses ohne grossen Aufwand durch uns mit unserer speziellen Programmier- oder Schweizer Normtastatur und mit einem augenschonenden Blendschutz ausstatten lassen!

Kommen Sie - vergleichen Sie!

Damit wir Ihnen in aller Ruhe zeigen können, was ein Commodore-Computer zu leisten vermag, ist eine telefonische Voranmeldung unerlässlich (Tel. 063 72 1113). Auf Wunsch senden wir Ihnen auch gerne detaillierte Informationen zu.

Konstruktionsänderungen vorbehalten.

Hard- + Software aus einer Hand

COMPU LIFE

Rüfenacht AG, 4950 Huttwil, Telefon 063 72 1113

Tricks und Tips

LISTSCHUTZ CBM 8000

O. W. Kuhn / Bonstetten

Im Heft 5/81 haben wir ein Programm vorgestellt, welches es erlaubt, Programme vor unberechtigtem Listen zu schützen. Heute können wir Ihnen eine erweiterte Version dieses Programms vorstellen, das auf dem CBM 8032 läuft. Es sperrt den Befehl LIST, SYS(xxxx), PRINT PEEK(xxxx) und POKE im Direkt Modus, LIST auch im Programm-Modus.

Das Sperren von PEEK und POKE im Direkt-Modus sperrt das Abfragen bzw. Aendern in der Zero-Page, um aus der Sicherheitsroutine herauszuspringen.

Der SYS Befehl im Programm-Modus konnte nicht gesperrt werden. Dadurch könnte natürlich die CHRGET-Routine auf ihren ursprünglichen Inhalt zurückgesetzt werden. Dies ist also die schwächste Stelle des Schutzprogramms. Man könnte sich jedoch mit der Aufzeichnung der Zero-Page ab \$ 70 behelfen, sodass das Programm bei Hex \$ 70 startet.

Der Stack muss die Startadresse der RUN-Routine enthalten, die bei einem RTS-Befehl sofort abgerufen wird und das Programm automatisch startet. Ein Inaktivieren der STOP-Taste würde sich ebenfalls erübrigen.

Das Schutzprogramm wird mit SYS (896) aktiviert.

```
50 PRINT"␣   DIESES PROGRAMM ERZEUGT DAS MASCHINEN-PROGRAMM DES LISTSCHUTZ-";
55 PRINT"PROGRAMMS.␣":PRINTTAB(30)"␣␣␣␣␣␣ BITTE WARTEN !! ␣␣␣␣␣␣"
60 F#="SUMMENFEHLER DER 'DATA'-ZAHLEN !! BITTE ZEILEN NOCHMALS PRUEFEN."
100 FORI=896TO1009:READX:POKEI,X:S=S+X:NEXT
110 IFS<>1419STHENPRINTTAB(5)F#:END
120 PRINTTAB(15)"ALLES KORREKT ! DAS MASCHINEN-PROGRAMM IST GELADEN.":END
1000 DATA165,119,133,17,165,120,133,18,162,23,189,153,211,149,112,202
1010 DATA208,248,165,17,133,119,165,18,133,120,169,76,133,121,169,167
1020 DATA133,122,169,3,133,123,96,201,155,240,23,72,165,55,201,255
1030 DATA240,3,76,214,3,104,201,158,240,8,201,153,240,4,201,151
1040 DATA208,21,72,138,72,186,189,4,1,201,183,208,7,189,3,1
1050 DATA201,126,240,10,104,170,104,201,58,176,203,76,125,0,169,234
1060 DATA160,3,32,29,187,160,122,76,188,245,76,73,83,84,73,78
1070 DATA71,0
READY.
```

```
0001 0000 ;*****
0002 0000 ;*
0003 0000 ;* -- L I S T S C H U T Z -- *
0004 0000 ;* ===== *
0005 0000 ;*
0006 0000 ;* ORIGINAL IDEA BY: S. CSOMOR *
0007 0000 ;*
0008 0000 ;* ADAPTED FOR 8032 CBM-COMPUTER *
0009 0000 ;*
0010 0000 ;* (C) BY O.W.KUHN, BONSTETTEN *
0011 0000 ;*
0012 0000 ;* NOVEMBER 1981 *
0013 0000 ;*
0014 0000 ;*****
0015 0000 ;
0016 0000 ;*****
0017 0000 ;* ACTIVATE WITH SYS(896) *
0018 0000 ;*****
0019 0000 ;
0020 0000 ;START = $0380 = 896
0021 0000 ;
0022 0000 ;
```



```

0023 0000 ;SYSTEM SUBROUTINE CALLS
0024 0000 ;
0025 0000 PRINT = #BB1D
0026 0000 ERROR = #F5BC
0027 0000 CHRGE = #D399
0028 0000 ;
0029 0000 ;
0030 0000 * = #0380
0031 0380 ;
0032 0380 A5 77 START LDA #77 ;GET BASIC POINTER
0033 0382 85 11 STA #11 ;SAVE AWAY HI-BYTE
0034 0384 A5 78 LDA #78 ;LO-BYTE OF POINTER
0035 0386 85 12 STA #12 ;SAVE IT TOO
0036 0388 A2 17 LDX ##17 ;LOAD OFFSET
0037 038A BD 99 D3 RESTOR LDA CHRGE,X ;GET CHRGET-ROUTINE
0038 038D 95 70 STA #70,X ;RESTORE CHRGET
0039 038F CA DEX ;NEXT BYTE
0040 0390 D0 F8 BNE RESTOR ;GO BACK IF NOT ALL DONE
0041 0392 A5 11 LDA #11 ;GET POINTER BACK
0042 0394 85 77 STA #77 ;HI-BYTE TO CHRGET
0043 0396 A5 12 LDA #12 ;LO-BYTE OF POINTER
0044 0398 85 78 STA #78 ;PUT IT TO CHRGET
0045 039A A9 4C LDA ##4C ;LOAD JMP-INSTRUCTION
0046 039C 85 79 STA #79 ;PUT IT INTO CHRGET
0047 039E A9 A7 LDA #<BEGIN ;HI-BYTE OF PROG-START
0048 03A0 85 7A STA #7A ;PUT IT INTO CHRGET
0049 03A2 A9 03 LDA #>BEGIN ;LO-BYTE OF PROG-START
0050 03A4 85 7B STA #7B ;PUT IT INTO CHRGET
0051 03A6 50 RETURN RTS ;BACK TO BASIC
0052 03A7 C9 9B BEGIN CMP ##9B ;IS IT 'LIST' ?
0053 03A9 F0 17 BEQ PROTEC ;YES, PROTECT IT
0054 03AB 48 PHA ;SAVE ACC
0055 03AC A5 37 LDA #37 ;DIRECT MODE FLAG
0056 03AE C9 FF CMP ##FF ;IS IT #FF ?
0057 03B0 F0 03 BEQ DIRECT ;YES, CHECK FURTHER
0058 03B2 4C D6 03 JMP CONT1 ;NO, GO BACK TO CHRGET
0059 03B5 68 DIRECT PLA ;GET TOKEN BACK
0060 03B6 C9 9E CMP ##9E ;IS IT 'SYS'
0061 03B8 F0 08 BEQ PROTEC ;YES, PROTECT IT
0062 03BA C9 99 CMP ##99 ;IS IT 'PRINT'
0063 03BC F0 04 BEQ PROTEC ;YES, PROTECT IT
0064 03BE C9 97 CMP ##97 ;IS IT 'POKE'
0065 03C0 D0 15 BNE CONT ;NO, GO BACK TO CHRGET
0066 03C2 48 PROTEC PHA ;SAVE TOKEN TO STACK
0067 03C3 8A TXA ;GET X-REGISTER
0068 03C4 48 PHA ;SAVE IT TO STACK
0069 03C5 BA TSX ;GET STACK POINTER
0070 03C6 BD 04 01 LDA #104,X ;READ STACK
0071 03C9 C9 B7 CMP ##B7 ;IS IT A #B7
0072 03CB D0 07 BNE RESTK ;NO, RESTORE STACK
0073 03CD BD 03 01 LDA #103,X ;READ NEXT STACK-BYTE
0074 03D0 C9 7E CMP ##7E ;IS IT A #7E
0075 03D2 F0 0A BEQ PMSG ;YES, PRINT PROTECT-MSG
0076 03D4 68 RESTK PLA ;GET X-REGISTER BACK
0077 03D5 AA TAX ;PUT IT BACK INTO X
0078 03D6 68 CONT1 PLA ;GET TOKEN BACK
0079 03D7 C9 3A CONT CMP ##3A ;IS IT A NUMBER ?
0080 03D9 B0 CB BCS RETURN ;RETURN TO BASIC
0081 03DB 4C 7D 00 JMP #7D ;GO BACK TO CHRGET
0082 03DE A9 EA PMSG LDA #<MSG ;HI-BYTE OF P-MSG
0083 03E0 A0 03 LDY #>MSG ;LO-BYTE OF P-MSG
0084 03E2 20 1D BB JSR PRINT ;PRINT IT
0085 03E5 A0 7A LDY ##7A ;GET ERROR-MSG POINTER
0086 03E7 4C BC F5 JMP ERROR ;GET AND PRINT IT
0087 03EA 4C 49 MSG .BYTE 'LISTING',#00
0088 03F1 00
0088 03F2 .END

```

ERRORS = 0000

SYMBOL TABLE

SYMBOL VALUE

| | | | | | | | |
|--------|------|--------|------|-------|------|--------|------|
| BEGIN | 03A7 | CHRGE | D399 | CONT | 03D7 | CONT1 | 03D6 |
| DIRECT | 03B5 | ERROR | F5BC | MSG | 03EA | PMSG | 03DE |
| PRINT | BB1D | PROTEC | 03C2 | RESTK | 03D4 | RESTOR | 038A |
| RETURN | 03A6 | START | 0380 | | | | |

END OF ASSEMBLY

LAGERBUCHHALTUNG ADRESSVERWALTUNG

EINGABEPROGRAMME

In den beiden letzten Ausgaben der CBM/PET NEWS haben wir Ihnen das Menü und die Diskferroutinen vorgestellt. Heute wollen wir nun das Programm zur Eingabe und Abspeicherung der Daten (Adressen und Artikel) näher erklären.

Zur Abspeicherung von Daten auf einer Disk sind vier Arten von Dateien gebräuchlich :

1. Sequentielle Dateien
2. Index-Sequentielle Dateien
3. Relative Dateien
4. Dateien mit Direktzugriff

Im Verlaufe unserer Fortsetzungsserie werden wir alle vier Möglichkeiten der Speicherung und des Abruf vorstellen, beginnend mit der einfachen, sequentiellen Datei.

Die sequentielle Datei ist vom Programmieraufwand und Verständnis her gesehen, die einfachste Art der Datenspeicherung. Die Daten werden bitweise genau in der Reihenfolge abgespeichert, wie sie an das Floppy übergeben werden. Das gleiche geschieht natürlich auch mit umgekehrtem Vorzeichen beim Datenabruf. Neben den genannten Vorteilen der sequentiellen Datei ist als grösster Nachteil die relativ lange Zugriffszeit des Floppys auf die Daten zu nennen. Diesem Uebelstand werden wir in einer der nächsten Fortsetzungen mit index-sequentuellen Dateien und solchen mit Direktzugriff abhelfen.

Die beiden vorgestellten Programme zur Eingabe und Abspeicherung von Artikeln und Adressen können ca. 5000 Artikel bzw. ca. 1500 Adressen auf einer Diskette verwalten. Der Unterschied liegt darin, dass das Adressverwaltungsprogramm, für die 3000er Serie geschrieben ist, dessen Floppy nur 180 kByte auf einer Disk hat, im Gegensatz zu 500 kByte des Floppy 8050.

Weiterhin ist es mit dem genannten Programm nicht möglich an eine bereits bestehende Datei zusätzliche Daten anzuhängen. Dies ist mit einer einfachen Erweiterung des Programms möglich, die wir ebenfalls in einer der nächsten Fortsetzungen bringen werden.

Die beiden Programme sind in ihrem Aufbau mehr oder weniger identisch und unterscheiden sich nur in der Abspeicher und Fehlererkennung sowie in der Einteilung des Bildschirms.

Nun noch kurz ein paar Worte zum Funktionsprinzip des Programms. Die eingetippten Namen (Artikel) werden zu einem String addiert und indexiert (b%). Sobald 25 Daten eingegeben sind, werden die Daten mit dem Filenamen " 1" abgespeichert und hinter die Datei ein "EOF" gesetzt. Nun wird f% (Filename) um 1 erhöht und die nächsten Daten können mit dem Filenamen " 2" abgespeichert werden. Werden weniger als 25 Daten abgespeichert, also das Programm in Zeile 1190 abgebrochen, werden die Daten abgespeichert, aber nun hinter die Datei ein "END" gesetzt. Vor der Dateneingabe ist selbstverständlich in Drive 1 eine leere, formatierte Diskette einzulegen.

ARTIKELEINGABE

```

900 rem vorbereitung
910 t$="#####** A r t i k e l e i n g a b e ***"
920 printt$
930 printtab(14)"#####Legen Sie eine formatierte Disk in Drive #
940 printtab(14)"#####Wenn Sie fertig sind, dann <RETURN> druecken."
950 setz$:ifz$<>chr$(13)then950
960 gosub9000
970 dimg$(25):i=0:fz=0:bz=0
1000 rem artikeleingabe
1010 printtab(19)t$
1020 printtab(19):"#####Artikelnummer      :";:a=6:gosub3000:a1$=x$
1030 printtab(19):"#####Artikelname        :";:a=25:gosub3000:a2$=x$
1040 printtab(19):"#####Lieferant          :";:a=20:gosub3000:a3$=x$
1050 printtab(19):"#####Einstandspreis     :";:a=7:gosub3000:a4$=x$
1060 printtab(19):"#####Verkaufspreis      :";:a=7:gosub3000:a5$=x$
1070 printtab(19):"#####Lagerbestand       :";:a=6:gosub3000:a6$=x$
1080 printtab(19):"#####Mininmalbestand  :";:a=4:gosub3000:a7$=x$
1090 printtab(19):"#####Stimmen die Daten ?"
1100 setz$
1110 if z$=""then1100
1120 ifz$="j"then 1150
1130 if z$="n"then1010
1140 goto1100
1150 printtab(19):"#####Wollen Sie weitere Artikel ?"
1160 setz$
1170 if z$=""then1160
1180 ifz$="j"thengosub2000:goto1010
1190 if z$="n"thene$="end":gosub2000:gosub8000:goto4000
1200 goto1160
2000 g$(bz)=a1$+a2$+a3$+a4$+a5$+a6$+a7$
2010 bz=bz+1
2020 ifbz>25thene$="eof":gosub8000:bz=0:fz=fz+1
2030 return
3000 x$="":an=0:print"  ";
3010 setzz$:ifzz$=""then3010
3020 zz=asc(zz$)
3030 ifzz=20then3090
3040 ifzz=13then3150
3050 x$=x$+zz$:anz=an+1
3060 ifanz>athenprint" ";:fori=1toan+1:print"  ";:next:goto3000
3070 printzz$:" ";
3080 goto3010
3090 ifanz=1thenx$="":an=0:goto3130
3100 ifan<1then3010
3110 anz=an-1
3120 x$=left$(x$,anz)
3130 print"  ";
3140 goto3010
3150 print" "
3160 fori=1toan-len(x$):x$=x$+" ";:next
3170 return
4000 print"#####load";chr$(34);"0:lagermenu";chr$(34);",8"
4010 print"#####run";:poke158,2:poke623,13:poke624,13:end
8000 rem abspeicherroutine
8010 dopen#1,(str$(fz)),d1,w
8020 fori=0tobz:print#1,g$(i):next
8030 print#1,e$:gosub9000
8040 dclose#1
8050 return

```



```

9000 rem fehlererkennung
9010 fs=ds
9020 ifds=0thenreturn
9030 print"#####";tab(24)ds$
9040 dclose#1:end

```

ADRESSEINGABE

```

900 rem vorbereitung
910 poke59468,14
920 t$="#####** A d r e s s e i n g a b e **"
930 printt$
940 print"#####Lesen Sie eine Disk in Drive # 1 !"
950 print"#####Sind Sie fertig ? Dann : <RETURN>"
960 getz$
970 ifz$<>chr$(13)then960
980 open15,8,15:print#15,"i1":gosub9000
990 dima$(25):i=0:b%=0:f%=0
1000 poke59468,14:rem adresseingabe
1010 printt$
1020 print"#####Anrede          :";a=3:gosub3000:a1$=x$
1030 print"#####Vorname        :";a=10:gosub3000:a2$=x$
1040 print"#####Name          :";a=15:gosub3000:a3$=x$
1050 print"#####Plz           :";
1055 a=4:gosub3000:a4$=x$:gosub5000:a5$=b$:goto1070
1060 print"#####Ort           :";a=20:gosub3000:a5$=x$
1070 print"#####Strasse & Nr. :";a=20:gosub3000:a6$=x$
1090 print"#####Stimmen die Daten ?"
1100 getz$
1110 if z$=""then1100
1120 ifz$="j"then 1150
1130 if z$="n"then1010
1140 goto1100
1150 print"#####Wollen Sie weitere Adressen ?"
1160 getz$
1170 if z$=""then1160
1180 ifz$="j"thengosub2000:goto1010
1190 if z$="n"thene$="end":gosub2000:gosub8000:goto4000
1200 goto1160
2000 g$(b%)=a1$+a2$+a3$+a4$+a5$+a6$
2010 b%=b%+1
2020 ifb%>=5thene$="end":gosub8000:b%=0:f%=f%+1
2030 return
3000 x$="":an=0:print " #";
3010 getzz$:ifzz$=""then3010
3020 zz=asc(zz$)
3030 ifzz=20then3090
3040 ifzz=13then3150
3050 x$=x$+zz$:an=an+1
3060 ifan>athenprint " ";fori=1toan+1:print"# #";:next:goto3000
3070 printzz$:"#";
3080 goto3010
3090 ifan=1thenx$="":an=0:goto3130
3100 ifan<1then3010
3110 an=an-1
3120 x$=left$(x$,an)
3130 print " ####";
3140 goto3010
3150 print " "
3160 fori=1toa-len(x$):x$=x$+" ";:next
3170 return
4000 print"#####load";chr$(34);"0:adressmenu";chr$(34);",8"
4010 print"#####run";poke158,2:poke623,13:poke624,13:end

```


Variablentabelle

| | |
|--------------------|--|
| a1\$ - a6\$ (a7\$) | Teilstring für Daten |
| g\$(b%) | indizierter String mit allen Teilstring |
| a | Zähler für vorgegebene Stellenzahl der Teilstring. |
| b% | Zähler für Stringindex. |
| f% | Zähler für Fileindex |

W i c h t i g

Es ist empfehlenswert bei der Eingabe des Programms die gleichen Zeilennummern und Variablen zu verwenden, da alle nachfolgenden Programme und Erweiterungen auf den hier genannten Zeilennummern und Variablen basieren.

Sie können nun bis zum Erscheinen des nächsten Heftes im April Ihre Adressen bzw. Lagerartikel eingeben und auf einer Disk abspeichern. Wir werden Ihnen dann den Abruf der Artikel von der Disk beschreiben.

LEICHTE KOST

WILHELM TELL

Wir haben Ihnen versprochen, von dieser Ausgabe an auch für die Benutzer des VC 20 interessante Beiträge zu veröffentlichen, leider ist unser Aufruf an die Leserschaft, uns VC 20-Beiträge zuzusenden, bisher nicht erhört worden. Trotzdem sollen Sie zu dem versprochenen Programm kommen.

Nachdem wir 1981 schon unseren Nationalsport, das Hornussen, vorgestellt haben, soll diesmal auch unser Nationalheld vorgestellt werden. Auch unseren Nachbarländern wird "Wilhelm Tell" sicherlich nicht unbekannt sein, denn schon Friedrich Schiller hat sich vor 200 Jahren mit dieser legendären Gestalt beschäftigt. Daher haben auch wir eine Anleihe bei Schiller gemacht und den "Apfelschuss" versucht in ein kleines Programm zu verpacken und daraus ein kurzweiliges Spiel zu machen.

Was lag näher, als dieses Programm auf dem VC 20 zu machen, denn durch die Möglichkeit der farbigen Gestaltung kommt dieses Spiel erst voll zur Geltung.

Die Bedienung des Programms ist denkbar einfach, es gilt einen hochgeworfenen Apfel mit dem Pfeil aus einer Armbrust zu treffen. Die Schwierigkeit liegt in der Entfernung der Armbrust zum Apfel und der Wurfhöhe des Apfels, die sich ständig ändern. Die Anzahl der Schüsse und die Treffer werden laufend gezählt.

Wir wünschen Ihnen bei Tells Apfelschuss recht viel Vergnügen.


```

1040 T=11
1070 PRINT "#####"
1080 PRINT "#####"AB$;
1090 PRINT "T#####"P$
1100 POKE36879,221:GOSUB 5000
1105 IF X<0 THEN 3000
1110 GET S$:IF S$<>" " THEN 1100
1115 S=S+1
1120 PRINT "#####"TAB(T)P$
1130 T=T-2
1140 IF T<=0 THEN 3000
1150 IF PEEK(8076+Z)<>81 THEN 1100
1160 IF PEEK(8076+Z+1)=32 THEN 3000
1165 IF X<0 THEN 3000
1170 GOTO 2000
1180 GOSUB 5000:GOTO 1120
2000 FOR F=25 TO 31 STEP 2
2010 POKE 36879,F
2020 FOR N=0 TO 200:NEXT N
2030 NEXT F
2100 TR=TR+1:W=W+1
2110 TR$=""
2120 GOTO 3100
3000 TR$="KLEIN"
3010 W=W+1
3100 POKE36879,142
3101 W$=STR$(W):SS$=STR$(S):T$=STR$(TR)
3102 IF LEN(W$)>=3 THEN 3105
3103 W$="" +W$
3104 GOTO 3102
3105 IF LEN(SS$)>=3 THEN 3108
3106 SS$="" +SS$
3107 GOTO 3105
3108 IF LEN(T$)>=3 THEN 3110
3109 T$="" +T$:GOTO 3108
3110 PRINT"##### "TR$" TREFFER##### "
3120 PRINT" "
3130 PRINT" ##### "W$"##### "SS$"##### "
3140 PRINT" #REPFEL##### #SCHUSS"
3150 PRINT" #####"
3160 PRINT" #####"
3170 PRINT" #####"
3180 PRINT" #####"
3190 PRINT" #####"
3200 PRINT" #####"
3210 PRINT" #####"
3220 PRINT" #####"
3230 PRINT" #####"
3240 PRINT" #####"
3250 PRINT" #####"
3260 PRINT" #####"
3270 PRINT" #####"
3280 PRINT" #####"
3290 PRINT" #####"
3300 PRINT" #####"
3310 PRINT" #####TREFFER:"T$"##### "
3330 FOR N=0 TO 4000:NEXT N
3340 H=0
3400 GOTO 1000

```



```

4999 REM   HOEHE BESTIMMEN
5000 IF H=A THEN 5020
5010 H=H+1:GOTO 5030
5020 H=H-1:A=H
5030 X=H*22
5040 POKE 8164-X+Z,32
5050 POKE 8142-X+Z,81
5060 POKE 8120-X+Z,85
5070 POKE 8098-X+Z,32
5080 POKE 38862-X+Z,2
5090 POKE 38840-X+Z,5
5100 RETURN
READY.

```

```

10 - 70 Textremarks
80     Farbgebung: weisser Rahmen mit hellblauem Hintergrund
90     Setzen der gelben Zeichenfarbe
100 - 250 Titelbild
255 - 265 FOR-NEXT Schleifen für Farbwechsel
270     Farbgebung: roter Rahmen mit weissem Hintergrund
280 - 310 Frage nach Erklärungen (blaue Schrift)
400 - 600 Erklärungen (blaue Schrift); das Wort "Space" wird in roter
        Reversschrift und die letzten drei Zeilen in purpurner Re-
        versschrift geschrieben
1000    Definieren der Höhenvariablen
1010    Definieren des Abstandes zur Armbrust
1020 - 1040 Variable für das Bild
1070 - 1090 Zeichnen des Spielfeldes
1100    Farbgebung: grüner Rahmen, hellgrünes Feld
1110    GET-Anweisung zur Auslösung des Schusses
1115    Schussanzahl um 1 erhöhen
1120    Pfeilbewegung auf dem Bildschirm
1140 - 1165 Frage nach Treffer und entsprechende Verzweigung.
2000 - 2030 Farbwechsel bei Treffer
2100 - 3010 Definieren der Zählvariablen bei Treffer oder Fehlschuss.
3100    Farbgebung: blauer Rahmen, oranges Feld
3101 - 3109 Stringlängen der Anzahl Äpfel, Schüsse und Treffer auf
        gleiche Werte setzen
3110 - 3330 farbiges Schlussbild zeichnen
5000 - 5100 Unterprogramm für den Apfelwurf

```

WUSSTEN SIE SCHON.

- dass der Befehl PRINT USING, der in vielen BASIC Dialekten enthalten ist und im COMMODORE BASIC fehlt, mit der unten stehenden kleinen Subroutine leicht umgangen werden kann.

```

100 REM PRINT-USING  A.SIMOES
110 INPUT X:GOSUB 60000
120 PRINT TAB(78-LEN(X#));X#
130 END
60000 X=INT(X*100+.5)/100
60010 X#=STR$(X):X#=RIGHT$(X#,LEN(X#)-1)
60020 IF X<1 THEN X#="00"+X#:GOTO 60040
60030 IF LEN(X#)<3 THEN X#=X#+".00":GOTO 60070
60040 IF MID$(X#,LEN(X#)-2,1)=". " THEN 60070
60050 IF MID$(X#,LEN(X#)-1,1)=". " THEN X#=X#+":GOTO 60070
60060 X#=X#+".00"
60070 IF X>ABS(X) THEN X#="-"+X#:RETURN
60080 X#=" "+X#:RETURN

```


Hardware

BARCODELESER

In dieser Ausgabe beginnen wir in der Rubrik "Maschinenprogrammierung" mit der Besprechung von Programmen zum Einlesen und zur Ausgabe von Programmen und Daten im Barcode (Strichcode).

Nun braucht es aber zu dieser Art von Uebermittlung eine spezielle Art von Leser, mit dem es möglich ist, den auf Papier gedruckten Strichcode in den Rechner zu laden.

Diese Strichcodeleser sind schon seit geraumer Zeit bei Registrierkassen bekannt und werden dort zum direkten Einlesen von Preisen und Artikelnummer direkt ab der Verpackung unter Umgehung der Tastatur benutzt.

Es handelt sich bei diesen Lesern um optoelektronische Systeme, in denen das Licht einer Leuchtdiode (LED) vom Strichcode reflektiert wird und von einem Phototransistor in ein elektrisches Signal umgesetzt und dem Rechner zugeführt wird.

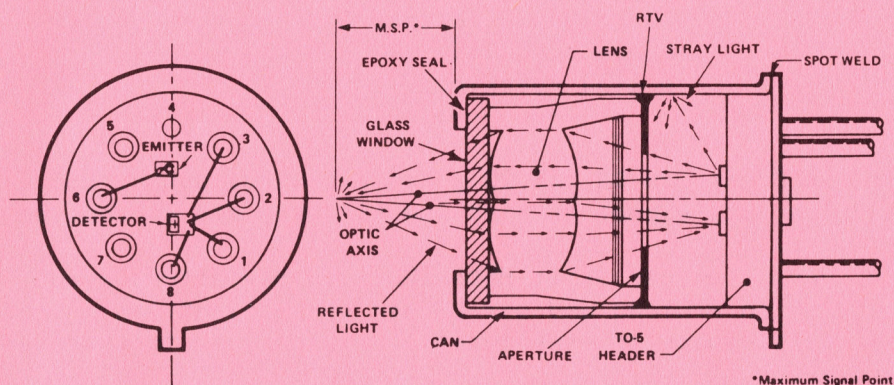
Von verschiedenen Herstellern werden heute optoelektrische Wandler angeboten, in denen die Leuchtdiode und der Phototransistor integriert sind und die darüberhinaus auch bereits das erforderliche Linsensystem enthalten. Der Selbstbau solcher Systeme mit LED und Phototransistoren ist theoretisch möglich, wir raten Ihnen jedoch davon ab, da die Justierung des LED und des Phototransistors ohne entsprechende Linsensysteme sehr schwierig ist. Die Elektronik der Leser ist denkbar einfach. Die LED wird mit einer möglichst konstanten Spannung von 5 V unter Vorschaltung eines Widerstandes von 180-270 Ohm betrieben. Der Phototransistor gibt die Helligkeitsänderungen an einen kleinen Verstärker weiter, der daraus ein TTL-Signal erzeugt, das an die USER-PORT-Erweiterung geleitet wird.

Zur Verstärkung werden meist Operationsverstärker eingesetzt, die dann direkt im Lesestift integriert sind.

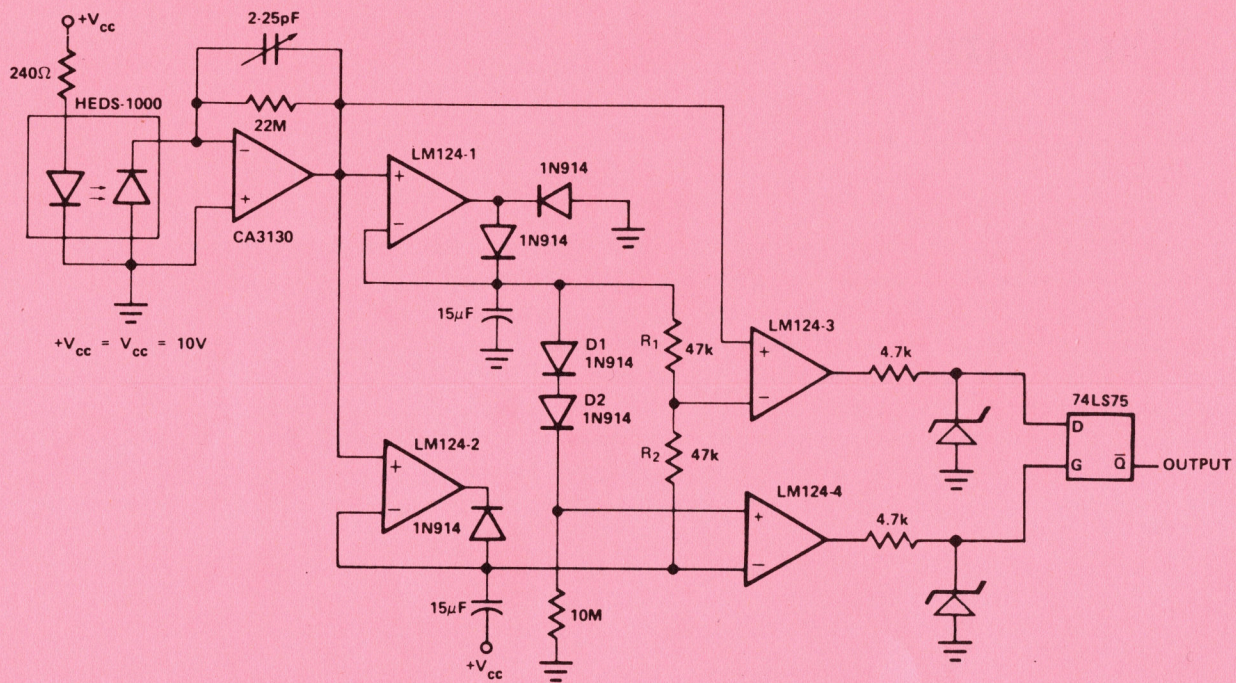
Sofern Sie bereits über einen solchen Lesestift verfügen, muss dieser am Anschluss P07 des USER-PORTS (PIN 10 der unteren Steckerleiste) angeschlossen werden. Wenn die Spannungsversorgung des Lesestiftes nicht über das Kassetteninterface erfolgt, muss noch eine Masseverbindung mit PIN 1,12,A oder N hergestellt werden.

Vom USER-PORT wird das Signal im Rechner selbst in die entsprechenden Daten oder Programme umgewandelt.

Das Schaltbild zeigt die Schaltung eines solchermaßen aufgebauten Barcodelesers.



Optik des Barcodelesers



Barcodeleser

Fertig montierte Barcodeleser können bei verschiedenen Distributoren gekauft werden, jedoch haben wir bei Tests festgestellt, dass diese, im Handel befindlichen Leser entweder sehr unhandlich sind, oder eine nur ungenügende Genauigkeit aufweisen. Von den handelüblichen Barcodelesern haben wir die unten aufgeführten Typen geprüft und für bedingt geeignet befunden.

In der Aprilausgabe dieses Heftes werden wir Ihnen dann einen eigenen Lesestift vorstellen, der sich auf Grund seiner Griffelform vor allem durch seine Handlichkeit auszeichnet und der speziell auf das CBM-System bzw. auf die USER-PORT-Erweiterung zugeschnitten ist.

Da wir Ihnen gleichzeitig auch die notwendige Software dazu anbieten, sind wir sicher, Ihnen ein komplettes System anzubieten, mit dem es möglich ist, den Austausch von Programmen und Daten auf rein drucktechnischem Weg unter Umgehung von Disk oder Kassetten zu ermöglichen. Für nähere Einzelheiten bitten wir Sie, sich vor allem die Rubrik "Maschinenprogrammierung" sehr genau durchzulesen.

Zum Lesen des Barcode wurde unter andern der Leserstift der Firma

Ing. Büro W. Kranis GmbH
Lindenberg 113
D 8134 Poecking

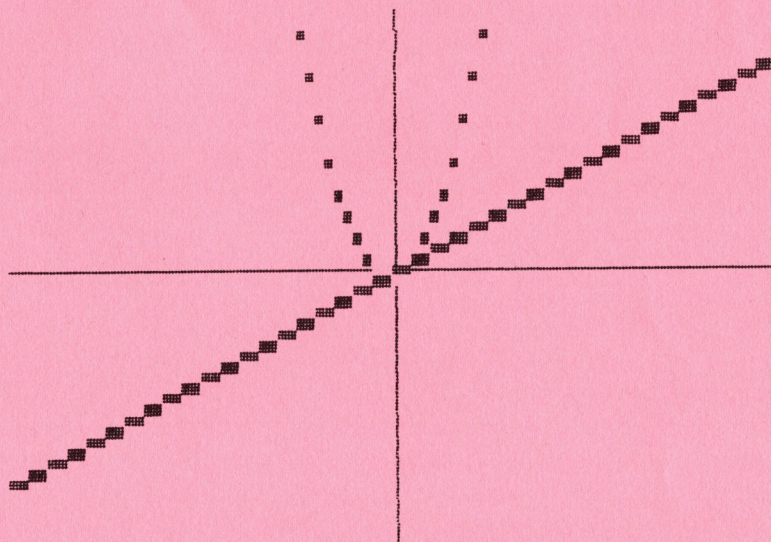
getestet, der zum Preis von DM 135.00 geliefert wird. Obwohl das Gerät relativ unhandlich ist, eignet er sich zum Einlesen des Barcode sehr gut.

Weitere Leser, vor allem der Stift HEDS 3000 von Hewlett-Packard befindet sich noch in Prüfung und wir werden voraussichtlich in der Aprilausgabe der News ein geeignetes Interface für diesen formschönen Lesestift vorstellen.

Programm des Monats BASIC-BASIC

MULTI PLOTTER

- Programme, die Funktionen oder andere graphische Darstellungen auf einem Printer oder auf dem Bildschirm plotten sind hinlänglich bekannt. Sie arbeiten alle mehr oder weniger nach dem gleichen System, wobei vielfach Maschinensprachen Subroutinen verwendet werden, um auf dem Bildschirm eine verbesserte Auflösung zu erreichen. Unter Zuhilfenahme solcher Subroutinen erreicht man auf dem Bildschirm eine Auflösung von 4000 Punkten, anstatt der sonst nur 1000 möglichen Punkte. Die Besonderheit dieses Programms liegt darin, dass gleichzeitig mehrere Funktionen geplottet werden können. Natürlich sind auch hier Grenzen gesetzt, da bei zu vielen Funktionen auf dem Bildschirm die Uebersichtlichkeit nicht mehr gewährleistet ist.



Hardcopyausdruck des Multiplotters

PROGRAMMBEDIENUNG

Zuerst fragt der Rechner nach der Anzahl der Funktionen, die geplottet werden sollen. Die Anzahl ist nicht beschränkt, jedoch wurde oben schon erwähnt, dass zuviele Funktionen gleichzeitig die Uebersichtlichkeit stark beeinträchtigen.

Als Nächstes gibt man die verschiedenen Funktionen in der Form

$$Y=F(X)$$

ein.

Es sind folgende Zeichen erlaubt :

(), /, *, ↑, +, -, ., X, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

wobei X immer das Argument der Funktion sein muss.

Folgende Funktionen werden akzeptiert:

ABS, ATN, COS, EXP, INT, LOG, RND, SIN, SGN, SQR, TAN auch π wird angenommen.

Beispiel: $Y=X^3$; $Y=EXP(1+(SIN(\pi+X)))$; $Y=INT(X)$

Danach gibt man die obere und untere X-Schranke ein, sie definieren den Bildschirmausschnitt in der Horizontalen.

Falls man auf die Frage : "Soll die X=Achse in die Bildschirmmitte gelegt werden " mit Ja antwortet, wird dies ausgeführt und die untere und obere Y-Schranke so gewählt, dass die Funktion verzerrungsfrei gezeichnet wird. Andernfalls kann man die untere Y-Schranke eingeben. Damit sind die Eingaben abgeschlossen. Der Computer zeichnet nun die Funktionen und wartet am Ende auf einen Tastendruck. Ist dies die \square -Taste, so beginnt das Programm von vorne, andernfalls kann der Bediener einen neuen Bildschirmausschnitt definieren. Bei Anschluss eines Verstärkers kündigt sich das Programmende durch einen kurzen Pfeifton an.

Programmdiskussion

-
- 9 - 18 Dimensionierung der Felder und DATA-Statements.
 - 99 - 102 Eingabe der Anzahl der zu plottenden Funktionen.
 - 110 - 130 Die Funktionen werden abgefragt und im Feld F\$ gespeichert.
 - 200 Einlesen der gespeicherten Funktionen.
 - 230 - 290 Einlesen der Zeichen, um die X-Achse und Funktionen darstellen zu können.
 - 299 - 390 Einlesen der linken und rechten Bildschirmbegrenzung und Fehlertest. Die Zeilen 331 und 341 verhindern, das Auftreten des X-Wertes 0, da für diesen Wert einige Funktionen nicht definiert sind z.B. $Y=1/X$; $LOG(X)$
 - 400 - 500 Definieren der X-Werte innerhalb des Bildschirms.
 - 520 - 575 Bestimmen der oberen und unteren Bildschirmbegrenzung inkl. Feldertest.
 - 580 Einsetzen der Funktion. Erhöhen des Funktionenpointers.
 - 610 - 620 Berechnen der einzelnen Funktionswerte.
 - 640 Y-Schrittweite.
 - 660 - 670 Berechnen der Bildschirmposition der Koordinatenachsen.
 - 675 - 712 Zeichnen der Koordinatenachsen. falls diese die erste zu plottenden Funktion ist und die Achsen im Bildschirmbereich sind.
 - 720 - 750 Bestimmen der Lage der zwei benachbarten Punkte, die zusammen zur gleichen Bildschirmspalte gehören.
 - 760 - 790 Plotten der nächsten Funktion bzw. warten auf Befehl für Neustart.
 - 840 - 880 Bestimmung ob die gleichen Plottpunkte zum gleichen Zeichen gehören, dh. es muss \square , \blacksquare , \blacktriangleright oder \blacktriangleleft geschrieben werden.
 - 890 - 900 ist dies der Fall:
schreiben eines Zeichens auf den Bildschirm.
 - 910 - 930 ist dies nicht der Fall:
schreiben zweier Zeichen auf den Bildschirm
 - 940 - 1110 Diverse kleine schon besprochene Unterprogramme.

59999 - 60006 Erzeugen der Funktionen aus einem String
 Der String wird Zeichen für Zeichen durchgegangen, jeder
 Befehl in seinen TOKEN (BASIC interner Code) umgewan-
 delt und mittels POKE in Zeile 7 geschrieben. Zuletzt
 wird ein Doppelpunkt und ein REM angehängt, damit eine
 alte, längere Funktion nicht interpretiert wird.

MULTI PLOTTER

```

0 GOTO10
1 : ** MULTI PLOTTER **
2 : (C) 11.07.1981
3 : PETER SOMM
4 : ALFRED HUGGENBERGERSTR. 19
5 : 8570 WEINFELDEN
6 :
7 Y=COS(X):REM:.....
:.....
8 RETURN
9 REM PROGRAMMVORBEREITUNG
10 DIMXX$(31),XX(31),X(79),F(79),M(80),S(1,1) 50,3,51
15 DATA(,40,.)41,/,173,*,172,↑,174,+ ,170,-,171,.,46,X,88,0,48,1,49,2,
16 DATA4,52,5,53,6,54,7,55,8,56,9,57,ABS,182,ATN,193,COS,190,EXP,189,
17 DATALOG,188,RND,187,SGN,180,SIN,191,SQR,186,TAN,192,π,255 INT,181
18 DATA98,127,255,226,123,70,126,68,108,71,124,72
99 REM DEFINITION DER FUNKTIONEN
100 PRINT"☐ WIEVIELE FUNKTIONEN SOLLN GEPLOTTET WERDEN ? ";
101 POKE14,1:INPUTPF$:PF=VAL(PF$):POKE14,0:PRINT:IFPF=0THEN100
102 DIMF$(PF)
110 FORT=1TOPF
120 POKE14,1:INPUT"☐ F = ";F$(T):POKE14,0:PRINT
130 NEXT
200 RESTORE:FORT=1TO31:READXX$(T),XX(T):NEXT
230 M(80)=2:READS(0,0),S(1,0),S(0,1),S(1,1)
240 FORK=1TO79STEP2
250 M(K)=1
260 NEXT
270 FORK=0TO3
280 READT(K),L(K)
290 NEXT
299 REM DEFINITION DES BILDAUSSCHNITTES
300 PRINT"☐ DEFINITION DES BILDAUSSCHNITTS:"
310 PRINT"☐ "
320 PRINT"EINGABE DES DARZUSTELLENDEN X-BEREICHES☐"
330 PRINT"☐ UNTERE X-SCHRANKE : ";POKE14,1:INPUTXU$:PRINT:POKE14,0:
331 IFXU=0THENXU=,00001 PRINT:XU=VAL(XU$)
340 PRINT"☐ OBERE X-SCHRANKE : ";POKE14,1:INPUTX0$:PRINT:POKE14,0:
341 IFX0=0THENX0=,00001 PRINT:X0=VAL(X0$)
350 IFX0<=X0THENPRINT"☐ FEHLER☐: FALSCHES X-SCHRANKEN☐":GOTO330
360 UK=X0-XU:X0=X0-UK:XU=XU-UK
370 TX=INT(-40/(X0-XU)*XU+.5)
380 IFTX<0THENTX=-1
390 IFTX>40THENTX=41
400 X(0)=X0
480 P=(X0-XU)/79
490 FORK=0TO78
500 X(K+1)=X(K)+P:NEXT
  
```



```

520 PRINT"SO LL DIE X-ACHSE IN DIE BILDMITTE GELEGT WERDEN ?
525 POKE158,0:REM 8K-PET : POKE525,0 !!!!!
530 GETRP#:IFRP#<>"J"ANDRP#<>"N"THEN530
540 IFRP#="J"THENY0=ABS(X0-XU)/40*25/2:YU=-Y0:GOTO575
550 PRINT"NEINGABE DES DARZUSTELLENDEN Y-BEREICHES"
560 PRINT"    UNTERE Y-SCHRANKE : ";POKE14,1:INPUTYU#:PRINT:POKE14,0:
    PRINT:YU=VAL(YU#)
561 PRINT"SO LL DIE DARSTELLUNG VERZEICHNUNGSFREI FOLGEN ?"
562 POKE158,0:REM 8K-PET : POKE525,0 !!!!!
563 GETRP#:IFRP#<>"J"ANDRP#<>"N"THEN563
564 IFRP#="J"THENY0=(ABS(X0-XU)/40*25)+YU:GOTO575
570 PRINT"    OBERE Y-SCHRANKE : ";POKE14,1:INPUTY0#:PRINT:POKE14,0:
    PRINT:Y0=VAL(Y0#)

572 IFY0<=Y0THENPRINT"FEHLER: FALSCH E Y-GRENZEN":GOTO550
575 PRINT" ";
579 REM BERECHNEN DER PLOTTPUNKTE
580 PR=PR+1:GOSUB60000:TY=25-INT(-25/(Y0-YU)*YU+.4)
590 IFTY<0THENTY=-1
600 IFTY>25THENTY=26
610 IFY0=Y0THENGOSUB960
620 IFY0>Y0THENGOSUB940
640 K1=49/(Y0-YU)
660 AX=INT(49*YU/(YU-Y0)+.5)
670 AY=INT(79*XU/(XU-X0)+.5)-79
675 IFPR>1THEN720
680 IFAX<0ORAX>49THEN710
690 FORK=33728-40*INT(AX/2)TO33728-40*INT(AX/2)+39
700 POKEK,L(M(AX)):NEXT
710 IFAY<0ORAY>79THEN720
711 FORK=32768+INT(AY/2)TO33728+INT(AY/2)STEP40
712 POKEK,L(2+M(AY)):NEXT
720 FORK=0TO78STEP2
730 P1=INT(K1*(F(K)-YU)+.5)
740 P2=INT(K1*(F(K+1)-YU)+.5)
750 GOSUB840:NEXT
760 IFPR<PFTHEN580
762 POKE59467,16:POKE59466,170:POKE59464,190:FORT=1TO300:NEXT:
765 POKE158,0:REM 8K-PET : POKE525,0 !!!!! POKE59467,0
770 GETP#:IFP#=""THEN770
780 IFF#<>"@"THENPR=0:GOTO200
790 RUN
840 C1=INT(P1/2):C2=INT(P2/2)
850 IFF1<0ORP1>79THENP1=80
860 IFF2<0ORP2>79THENP2=80
870 R1=M(P1):R2=M(P2)
880 IFC1-C2<>0THEN910
890 IFR1<2THENPOKE33728+K/2-40*C1,S(R1,R2)
900 RETURN
910 IFR1<2THENPOKE33728+K/2-40*C1,T(R1)
920 IFR2<2THENPOKE33728+K/2-40*C2,T(R2+2)
930 RETURN
940 FORK=0TO79:GOSUB1090:NEXT
950 RETURN
960 Y0=-1E38:YU=-Y0
970 FORK=0TO79:GOSUB1090
980 IFF(K)>Y0THENY0=F(K)
990 IFF(K)<Y0THENYU=F(K)
1000 NEXT
1010 RETURN
1090 X=X(K)

```



```

1100 GOSUB7
1110 F(K)=Y:RETURN
59999 REM GENERIEREN DER FUNKTION
60000 PP=1166:FORT=1:TLEN(F$(PR))
60001 HH$=MID$(F$(PR),T,1):HH=ASC(HH$):IFHH>64ANDHH<85THENHH$=
MID$(F$(PR),T,3):T=T+2
60002 FORTT=1:T031:IFXX$(TT)=HH$THENHH=XX(TT):TT=33
60003 NEXT:IFTT<33THEN T=100:NEXT:PRINT"NE FALSCHER FUNKTION":GOTO60005
60004 POKEPP,HH:PP=PP+1:NEXT:POKEPP,58:POKEPP+1,143:RETURN
60005 GETP$:IFP$=""THEN60005
60006 RUN
READY.

```

WUSSTEN SIE SCHON.

- dass nun auch ein Hardcopyprogramm für den CBM 8000 vorliegt. Das nachfolgende Listing ist eine Ergänzung der im Heft 2/81 beschriebenen Programme.

```

60000 REM          A.SIMOES
60010 NN=1:OPEN1,4:FORII=1:TONN
60020 JJ=0
60030 SL$="":JJ=JJ+1:FORKK=1:T080
60040 WW=0:XX=PEEK(32767+KK+80*(JJ-1))
60050 IFXX=320RXX=96THENSL$=SL$+CHR$(32):GOTO600220
60060 IFXX>127THENSL$=SL$+CHR$(18)
60070 IFPEEK(59468)=12THEN60130
60080 IFXX>0ANDXX<27THENSL$=SL$+CHR$(17)+CHR$(XX+64):WW=1
60090 IFXX>128ANDXX<155THENSL$=SL$+CHR$(17)+CHR$(XX-64):WW=1
60100 IFXX>64ANDXX<91THENSL$=SL$+CHR$(145)+CHR$(XX):WW=1
60110 IFXX>192ANDXX<219THENSL$=SL$+CHR$(145)+CHR$(XX-128):WW=1
60120 IFWW=1THEN60210
60130 IFXX<32THENSL$=SL$+CHR$(XX+64):GOTO600210
60140 IFXX>31ANDXX<64THENSL$=SL$+CHR$(XX):GOTO600210
60150 IFXX>63ANDXX<96THENSL$=SL$+CHR$(XX+128):GOTO600210
60160 IFXX>95ANDXX<128THENSL$=SL$+CHR$(XX+64):GOTO600210
60170 IFXX>127ANDXX<160THENSL$=SL$+CHR$(XX-64):GOTO600210
60180 IFXX>159ANDXX<192THENSL$=SL$+CHR$(XX-128):GOTO600210
60190 IFXX>191ANDXX<224THENSL$=SL$+CHR$(XX):GOTO600210
60200 IFXX>223THENSL$=SL$+CHR$(XX-64)
60210 IFXX>127THENSL$=SL$+CHR$(146)
60220 NEXTKK
60230 IFLEFT$(SL$,6)="READY."THEN60260
60240 PRINT#1,CC$:SL$
60250 IFJJ<25THEN60030
60260 FORPP=1:T010:PRINT#1:NEXTPP
60270 NEXTII:CC$="":CLOSE1

```

- dass es manchmal vorkommt, das nach einem BACKUP-Befehl kein Programm mehr ab Floppy 8050 mehr geladen werden kann. Beim BACKUP ist dann der Lesekopf verstellt worden. Probieren Sie dann im Direktmodus die Disk forciert zu initialisieren.

```

REM A.SIMOES
OPEN1,8,15:PRINT#1,"I0"
CLOSE1
REM FUER BEIDE DRIVE PRINT#1,"I"

```


Maschinensprache

BARCODEPROGRAMMIERUNG

Mit dem heutigen Artikel beginnt eine lose Folge von Theorie und Praxis über den Barcode, Lese- und Druckerprogramme.

Hier ist vorauszuschicken, dass die vorgestellten Programme speziell für Commodore Computer geschrieben sind. Wenn nichts anderes erwähnt wird, so gelten sie alle ohne Veränderung für die Serie 3000/4000 und 8000. Auf diese Universalität wurde grossen Wert gelegt.

1. Der Bar Code

Es ist wohl noch nicht allzu lange her, da war vom Barcode in der Öffentlichkeit nicht viel bekannt. Bei einigen Spezialanwendungen wurde dieses Verfahren zwar eingesetzt, einer grösseren Verbreitung standen aber noch etliche Hindernisse im Wege. Unter Vorbehalt auf Vollständigkeit und Gewichtung seien ein paar genannt: teure Leseköpfe (Sende/Empfangsdiode), schwieriger Anschluss an Kassenterminals und Computer, keine einheitliche genormte Darstellung der Information (Barcode), geringe eigene Möglichkeiten die nötigen Daten und Zeichen auszudrucken.

Diese Situation hat sich insofern geändert, als heute eine grosse Zahl billige und zuverlässige Leseköpfe resp. Lesestifte auf dem Markt erhältlich sind. Das Problem für eine weit verbreitete Anwendung liegt immer noch darin, die Daten auf einfache Art und billig ausdrucken zu können. Es ist zu hoffen, dass diesbezüglich noch einige gute Ideen und Lösungen gefunden werden.

Nun zurück in die Gegenwart. Der Barcode kann dazu benutzt werden, um Artikelnummern, Warenbezeichnungen, Preisangaben auf irgendwelchen Produkten auszudrucken, für die automatische Führung von Kassenzetteln, Verkaufsstatistiken, Lagerbeständen usw. Diese Barcodedarstellungen bestehen meist nur aus wenigen Zeichen.

Wir wollen aber den Barcode dazu benutzen, um grössere Datenmenge, sprich Programme, einzulesen. Dadurch werden Sie vom lästigen und mit Fehlern behafteten Eintippen von längeren Programmen befreit.

Wir haben uns entschlossen, Ihnen ein sauberes Einleseprogramm vorzustellen, das den Barcode verarbeitet, wie er in der Zeitschrift "mc" (BRD) verwendet wird.

Wie Sie vielleicht wissen, sind nicht alle Barcodes identisch. Je nach Anordnung der Striche definiert man ein eigenes Datenformat (von Firma zu Firma verschieden).

Wie ist nun der heute benutzte Barcode aufgebaut ?

Die Programme sind im ASCII-Code ausgedruckt. Jedes Zeichen belegt 8 Bit. Der BASIC-Befehl PRINTX4,Y4 braucht somit 10 Bytes, also 80 Bit. Jedes Bit wird durch einen dicken oder dünnen Strich dargestellt, dazwischen jeweils ein weisser konstanter Abstand (white gap). Ein dicker Strich bedeutet eine logische "1", ein dünner eine "0".

Speziell an diesem Code ist noch die Darstellung des ASCII-Charakters, dass nämlich das niederwertigste Bit links steht, also gerade umgekehrt als der normale ASCII-Code. Mit dem Lesestift wird von links nach die Zeile abgetastet. Das Verhältnis der Striche zueinander ist

Der Bildschirm wird gelöscht, der blinkende Cursor bleibt verschwunden. In der Mitte des Bildschirms erscheint 001. Diese Zahl gibt Ihnen immer wieder an, welche Barcodezeile Sie mit dem Lesestift abtasten müssen. Ist eine Zeile vollständig und fehlerfrei gelesen worden, so wird diese Nummer automatisch weiterschaltet. Die eingelesenen Zeichen werden auf der oberen Bildschirmzeile dargestellt und zwar solange, bis eine BASIC-Zeile komplett ist. Sie können somit den Aufbau des BASIC-Programms miterleben. Ist eine BASIC-Zeile komplett, so wird Sie automatisch in den BASIC-Speicher übertragen und auf dem Bildschirm gelöscht. Ein eventueller Rest von der nächsten Zeile wird ebenfalls auf dem Bildschirm dargestellt.

Sie brauchen also nichts weiter zu tun, als den gesamten Barcode mit dem Lesestift abzutasten. Alles weitere erledigt der Computer. Wenn Sie die letzte Barcodezeile eingelesen haben, sie besteht aus zwei dicken und 24 dünnen Strichen, so erscheint auf dem Bildschirm in Höhe der Zeilennummer "END", ebenfalls blinkt der Cursor wieder. Somit sind Sie wieder im BASIC-Mode und können mit LIST das Programm anschauen.

Vergessen Sie nicht das BASIC-Programm abzuspeichern.

Einige Punkte sollten Sie trotz der einfachen Handhabung beachten:

Möchten Sie während des Einlesevorganges aus irgend einem Grund abbrechen, so können Sie dies mit der STOP-Taste machen, alle anderen Tasten sind wirkungslos. Einen kurzen Moment die STOP-Taste drücken und der Rechner meldet BREAK. Ein Neustart mit SYS(31744) ist dann für eine Fortsetzung allerdings nötig.

Der Lesestift sollte kontinuierlich von links nach rechts über die Barcodezeile geschoben werden. Geschwindigkeitsänderungen spielen keine Rolle, nur nicht stehen bleiben. Sollte dies einmal passieren, so können Sie die gleiche Zeile noch einmal einlesen. Erscheint in der Mitte der Zeile ein %-Zeichen, so liegt ein Prüfsummenfehler vor. Sie müssen dann die Zeile nochmals lesen.

Achtung! Im Barcode sind die Zeilen nicht nummeriert, also keine Zeilen überspringen, oder böswillig Zeilen zweimal einlesen! Dies ist ein Mangel des Barcodeformats.

Wichtig! Mit dem Lesestift können Sie beliebig herumfahren, abheben, rückwärtsfahren, neu beginnen. Wenn Sie aber eine Zeile einlesen wollen, so müssen Sie den Lesestift vor der Zeile auf Weiss abstellen und mindestens eine Sekunde warten, bevor Sie losfahren. Dies wird zur eindeutigen Identifikation benötigt, damit der Rechner weiss, dass eine Zeile eingelesen wird. Auf diese Weise sollte das Einlesen keine Probleme mehr bieten.

In der nächsten Folge der CBM/PET NEWS werden wir Ihnen das Ausgabeprogramm zum Ausdruck eines BASIC-Programms im Barcode vorstellen. Wir beabsichtigen dann auch beide Programme auf Disk und Kassette zusammen mit einem Lesestift als Pack anzubieten, bis dahin müssen wir Sie allerdings noch um ein klein wenig Geduld bitten.


```

1000 REM -----
1010 REM ### * BARCODE - EINLESEPROGRAMM * ###
1020 REM ### VERSION FUER CBM 3000/4000/8000 ###
1030 REM ### MC-FORMAT JAN.82 E.BREU ###
1040 REM -----
1050 REM
1060 POKE 52,0: POKE 53,124
1070 REM
1080 FOR M=31744 TO 32289: READ X: POKE M,X: NEXT M
1090 REM
1100 PRINT "DATA'S GELADEN."
1110 REM
2000 DATA 216,169,0,133,218,133,219,169,49,133,0,169,48,133,1,133
2010 DATA 2,169,147,32,210,255,165,0,141,124,130,165,1,141,123,130
2020 DATA 165,2,141,122,130,88,32,228,255,240,7,201,3,208,3,76
2030 DATA 224,124,120,169,0,133,75,133,76,133,77,133,78,133,211,133
2040 DATA 62,133,63,32,235,125,176,221,165,60,133,64,165,61,133,65
2050 DATA 32,235,125,176,208,169,32,141,106,130,165,64,101,60,133,64
2060 DATA 165,65,101,61,133,65,56,165,64,233,6,133,64,165,65,233
2070 DATA 0,133,65,48,9,230,62,208,238,230,63,76,103,124,32,132
2080 DATA 125,176,39,132,211,152,240,29,32,132,125,176,29,152,24,101
2090 DATA 75,133,75,169,0,101,76,133,76,166,219,152,157,64,3,230
2100 DATA 219,198,211,208,227,32,132,125,144,7,165,218,133,219,76,22
2110 DATA 124,132,77,32,132,125,176,242,132,78,165,75,197,77,240,7
2120 DATA 169,37,141,106,130,208,227,165,76,197,78,208,243,165,75,5
2130 DATA 76,208,24,162,0,189,129,125,157,88,130,232,224,3,208,245
2140 DATA 169,32,141,106,130,165,219,133,218,88,96,162,48,230,0,165
2150 DATA 0,201,58,144,14,134,0,230,1,165,1,201,58,144,4,134
2160 DATA 1,230,2,166,218,189,64,3,32,210,255,201,13,240,10,232
2170 DATA 228,219,208,241,134,218,76,22,124,232,228,219,240,41,160,0
2180 DATA 189,64,3,153,64,3,200,232,228,219,208,244,132,219,162,0
2190 DATA 134,218,189,109,125,32,210,255,232,224,10,208,245,240,25,169
2200 DATA 147,32,210,255,76,3,125,162,0,134,218,134,219,189,119,125
2210 DATA 32,210,255,232,224,10,208,245,169,19,141,111,2,169,13,141
2220 DATA 112,2,141,113,2,141,114,2,169,4,133,158,96,83,89,83
2230 DATA 40,51,50,48,54,51,41,83,89,83,40,51,49,55,54,49
2240 DATA 41,5,14,4,162,8,160,0,32,146,125,176,4,202,208,248
2250 DATA 24,96,32,235,125,176,83,165,62,10,133,64,165,63,42,133
2260 DATA 65,56,165,60,229,64,165,61,229,65,48,41,169,0,133,209
2270 DATA 133,210,56,165,60,233,3,133,60,165,61,233,0,133,61,48
2280 DATA 9,230,209,208,238,230,210,76,179,125,165,209,133,60,165,210
2290 DATA 133,61,56,176,1,24,152,106,168,24,165,62,101,60,133,62
2300 DATA 165,63,101,61,74,133,63,102,62,24,96,169,0,133,60,133
2310 DATA 61,169,0,133,210,133,209,230,209,208,6,230,210,208,2,56
2320 DATA 96,44,79,232,16,241,230,60,208,4,230,61,240,241,44,79
2330 DATA 232,48,243,169,254,133,209,44,79,232,48,234,230,209,208,247
2340 DATA 24,96
2350 END

```

READY.

```

STARTADR : $7C00 31744
ENDADR : $7E21 32289
BYTES : 546

```

```

7C00 D8A90085DA85DBA9318500A930085018502A99320D2FFFA5008D7C82A5018D7B82
7C20 A5028D7A825820E4FFF007C903D0034CE07C78A900854B854C854D854E85D385
7C40 3E853F20EB7DB0DDA53C8540A53D854120EB7DB0D0A9208D6A82A540653C8540
7C60 A541653D854138A540E9068540A541E90085413009E63ED0EE63F4C677C2084
7C80 7DB02784D398F01D20847DB01D9818654B854BA900654C854CA5DE989D4003E6
7CA0 DBC6D3D0E320847D9007A5DA85DB4C167C844D20847DB0F2844EA54BC54DF007
7CC0 A9258D6A82D0E3A54CC54ED0F3A54B054CD018A200BD817D9D5882E8E003D0F5
7CE0 A9208D6A82A5DB85DA5860A230E600A500C93A900E8600E601A501C93A900486
7D00 01E602A6DABD400320D2FFC90DF00AE8E4DBD0F186DA4C167CE8E4DBF029A000
7D20 BD4003994003C8E8E4DBD0F484DBA20086DABD6D7D20D2FFE8E00AD0F5F019A9
7D40 9320D2FF4C037DA20086DA86DEBD777D20D2FFE8E00AD0F5A9138D6F02A90D6D
7D60 70028D71028D7202A904859E60535953283332303633329535953283331373631
7D80 29050E04A208A00020927DB004CAD0F8186020EB7DB053A53E0A8540A53F2A85
7DA0 4138A53CE540A53DE5413029A90085D185D238A53CE903853CA53DE900853D30
7DC0 09E6D1D0EEE6D24CB37DA5D1853CA5D2853D38B00118986AA818A53E653C853E
7DE0 A53F653D4A853F663E1860A900853C853DA90085D285D1E6D1D006E6D2D00238
7E00 602C4FE810F1E63CD004E63DF0F12C4FE830F3A9FE85D12C4FE830EAE6D1D0F7
7E20 1860

```


Neuheiten

HEXADEZIMALLISTEN

Gerade für den Beginner der Assembler oder Maschinensprachprogrammierung bereitet die Umrechnung des dezimalen in das hexadezimale Zahlensystem nicht unerhebliche Schwierigkeiten. Nun werden einige sicher mit Recht sagen, dass die Umrechnung leicht mit einem kleinen Programm vorgenommen werden kann. Nur wer kann schon mit seinem Rechner ein Maschinenprogramm erstellen und gleichzeitig mit dem gleichen Rechner die Umrechnung vornehmen. Selbstverständlich kann die Umrechnung auch mit einem Taschenrechner vorgenommen werden, dies ist jedoch vor allem eine Preisfrage.

Wir können Ihnen zu einem sehr günstigen Preis Listen anbieten, mit denen der ganze Bereich der Hexadezimalzahlen von 0000 - FFFF überstrichen wird und in denen die Zahlen direkt abgelesen werden können. Die Listen haben einen Umfang von 64 A 4 Seiten und sind direkt durch den Ersteller gegen Voreinzahlung von Fr. 10.-- zu beziehen:

Rolf Bornhauser
Straussenhaus
CH 9402 Mörschwil

Auszug aus einer Originalseite A4 (stark verkleinert)

| HEX / DEC | TABELLE | | | | | | | | | | | | | | | |
|-----------|-----------------|-------|-------|-------|-------|-------|-------|-------|-------------------|-------|-------|-------|-------|-------|-------|-------|
| | HEX 0000 - 03FF | | | | | | | | DEC 00000 - 01023 | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 0000 | 00000 | 00001 | 00002 | 00003 | 00004 | 00005 | 00006 | 00007 | 00008 | 00009 | 00010 | 00011 | 00012 | 00013 | 00014 | 00015 |
| 0010 | 00016 | 00017 | 00018 | 00019 | 00020 | 00021 | 00022 | 00023 | 00024 | 00025 | 00026 | 00027 | 00028 | 00029 | 00030 | 00031 |
| 0020 | 00032 | 00033 | 00034 | 00035 | 00036 | 00037 | 00038 | 00039 | 00040 | 00041 | 00042 | 00043 | 00044 | 00045 | 00046 | 00047 |
| 0030 | 00048 | 00049 | 00050 | 00051 | 00052 | 00053 | 00054 | 00055 | 00056 | 00057 | 00058 | 00059 | 00060 | 00061 | 00062 | 00063 |
| 0040 | 00064 | 00065 | 00066 | 00067 | 00068 | 00069 | 00070 | 00071 | 00072 | 00073 | 00074 | 00075 | 00076 | 00077 | 00078 | 00079 |
| 0050 | 00080 | 00081 | 00082 | 00083 | 00084 | 00085 | 00086 | 00087 | 00088 | 00089 | 00090 | 00091 | 00092 | 00093 | 00094 | 00095 |
| 0060 | 00096 | 00097 | 00098 | 00099 | 00100 | 00101 | 00102 | 00103 | 00104 | 00105 | 00106 | 00107 | 00108 | 00109 | 00110 | 00111 |
| 0070 | 00112 | 00113 | 00114 | 00115 | 00116 | 00117 | 00118 | 00119 | 00120 | 00121 | 00122 | 00123 | 00124 | 00125 | 00126 | 00127 |
| 0080 | 00128 | 00129 | 00130 | 00131 | 00132 | 00133 | 00134 | 00135 | 00136 | 00137 | 00138 | 00139 | 00140 | 00141 | 00142 | 00143 |
| 0090 | 00144 | 00145 | 00146 | 00147 | 00148 | 00149 | 00150 | 00151 | 00152 | 00153 | 00154 | 00155 | 00156 | 00157 | 00158 | 00159 |
| 00A0 | 00160 | 00161 | 00162 | 00163 | 00164 | 00165 | 00166 | 00167 | 00168 | 00169 | 00170 | 00171 | 00172 | 00173 | 00174 | 00175 |
| 00B0 | 00176 | 00177 | 00178 | 00179 | 00180 | 00181 | 00182 | 00183 | 00184 | 00185 | 00186 | 00187 | 00188 | 00189 | 00190 | 00191 |
| 00C0 | 00192 | 00193 | 00194 | 00195 | 00196 | 00197 | 00198 | 00199 | 00200 | 00201 | 00202 | 00203 | 00204 | 00205 | 00206 | 00207 |
| 00D0 | 00208 | 00209 | 00210 | 00211 | 00212 | 00213 | 00214 | 00215 | 00216 | 00217 | 00218 | 00219 | 00220 | 00221 | 00222 | 00223 |
| 00E0 | 00224 | 00225 | 00226 | 00227 | 00228 | 00229 | 00230 | 00231 | 00232 | 00233 | 00234 | 00235 | 00236 | 00237 | 00238 | 00239 |
| 00F0 | 00240 | 00241 | 00242 | 00243 | 00244 | 00245 | 00246 | 00247 | 00248 | 00249 | 00250 | 00251 | 00252 | 00253 | 00254 | 00255 |
| 0100 | 00256 | 00257 | 00258 | 00259 | 00260 | 00261 | 00262 | 00263 | 00264 | 00265 | 00266 | 00267 | 00268 | 00269 | 00270 | 00271 |
| 0110 | 00272 | 00273 | 00274 | 00275 | 00276 | 00277 | 00278 | 00279 | 00280 | 00281 | 00282 | 00283 | 00284 | 00285 | 00286 | 00287 |
| 0120 | 00288 | 00289 | 00290 | 00291 | 00292 | 00293 | 00294 | 00295 | 00296 | 00297 | 00298 | 00299 | 00300 | 00301 | 00302 | 00303 |
| 0130 | 00304 | 00305 | 00306 | 00307 | 00308 | 00309 | 00310 | 00311 | 00312 | 00313 | 00314 | 00315 | 00316 | 00317 | 00318 | 00319 |
| 0140 | 00320 | 00321 | 00322 | 00323 | 00324 | 00325 | 00326 | 00327 | 00328 | 00329 | 00330 | 00331 | 00332 | 00333 | 00334 | 00335 |
| 0150 | 00336 | 00337 | 00338 | 00339 | 00340 | 00341 | 00342 | 00343 | 00344 | 00345 | 00346 | 00347 | 00348 | 00349 | 00350 | 00351 |
| 0160 | 00352 | 00353 | 00354 | 00355 | 00356 | 00357 | 00358 | 00359 | 00360 | 00361 | 00362 | 00363 | 00364 | 00365 | 00366 | 00367 |
| 0170 | 00368 | 00369 | 00370 | 00371 | 00372 | 00373 | 00374 | 00375 | 00376 | 00377 | 00378 | 00379 | 00380 | 00381 | 00382 | 00383 |
| 0180 | 00384 | 00385 | 00386 | 00387 | 00388 | 00389 | 00390 | 00391 | 00392 | 00393 | 00394 | 00395 | 00396 | 00397 | 00398 | 00399 |
| 0190 | 00400 | 00401 | 00402 | 00403 | 00404 | 00405 | 00406 | 00407 | 00408 | 00409 | 00410 | 00411 | 00412 | 00413 | 00414 | 00415 |
| 01A0 | 00416 | 00417 | 00418 | 00419 | 00420 | 00421 | 00422 | 00423 | 00424 | 00425 | 00426 | 00427 | 00428 | 00429 | 00430 | 00431 |
| 01B0 | 00432 | 00433 | 00434 | 00435 | 00436 | 00437 | 00438 | 00439 | 00440 | 00441 | 00442 | 00443 | 00444 | 00445 | 00446 | 00447 |
| 01C0 | 00448 | 00449 | 00450 | 00451 | 00452 | 00453 | 00454 | 00455 | 00456 | 00457 | 00458 | 00459 | 00460 | 00461 | 00462 | 00463 |
| 01D0 | 00464 | 00465 | 00466 | 00467 | 00468 | 00469 | 00470 | 00471 | 00472 | 00473 | 00474 | 00475 | 00476 | 00477 | 00478 | 00479 |
| 01E0 | 00480 | 00481 | 00482 | 00483 | 00484 | 00485 | 00486 | 00487 | 00488 | 00489 | 00490 | 00491 | 00492 | 00493 | 00494 | 00495 |
| 01F0 | 00496 | 00497 | 00498 | 00499 | 00500 | 00501 | 00502 | 00503 | 00504 | 00505 | 00506 | 00507 | 00508 | 00509 | 00510 | 00511 |
| 0200 | 00512 | 00513 | 00514 | 00515 | 00516 | 00517 | 00518 | 00519 | 00520 | 00521 | 00522 | 00523 | 00524 | 00525 | 00526 | 00527 |
| 0210 | 00528 | 00529 | 00530 | 00531 | 00532 | 00533 | 00534 | 00535 | 00536 | 00537 | 00538 | 00539 | 00540 | 00541 | 00542 | 00543 |
| 0220 | 00544 | 00545 | 00546 | 00547 | 00548 | 00549 | 00550 | 00551 | 00552 | 00553 | 00554 | 00555 | 00556 | 00557 | 00558 | 00559 |
| 0230 | 00560 | 00561 | 00562 | 00563 | 00564 | 00565 | 00566 | 00567 | 00568 | 00569 | 00570 | 00571 | 00572 | 00573 | 00574 | 00575 |
| 0240 | 00576 | 00577 | 00578 | 00579 | 00580 | 00581 | 00582 | 00583 | 00584 | 00585 | 00586 | 00587 | 00588 | 00589 | 00590 | 00591 |
| 0250 | 00592 | 00593 | 00594 | 00595 | 00596 | 00597 | 00598 | 00599 | 00600 | 00601 | 00602 | 00603 | 00604 | 00605 | 00606 | 00607 |
| 0260 | 00608 | 00609 | 00610 | 00611 | 00612 | 00613 | 00614 | 00615 | 00616 | 00617 | 00618 | 00619 | 00620 | 00621 | 00622 | 00623 |
| 0270 | 00624 | 00625 | 00626 | 00627 | 00628 | 00629 | 00630 | 00631 | 00632 | 00633 | 00634 | 00635 | 00636 | 00637 | 00638 | 00639 |
| 0280 | 00640 | 00641 | 00642 | 00643 | 00644 | 00645 | 00646 | 00647 | 00648 | 00649 | 00650 | 00651 | 00652 | 00653 | 00654 | 00655 |
| 0290 | 00656 | 00657 | 00658 | 00659 | 00660 | 00661 | 00662 | 00663 | 00664 | 00665 | 00666 | 00667 | 00668 | 00669 | 00670 | 00671 |
| 02A0 | 00672 | 00673 | 00674 | 00675 | 00676 | 00677 | 00678 | 00679 | 00680 | 00681 | 00682 | 00683 | 00684 | 00685 | 00686 | 00687 |
| 02B0 | 00688 | 00689 | 00690 | 00691 | 00692 | 00693 | 00694 | 00695 | 00696 | 00697 | 00698 | 00699 | 00700 | 00701 | 00702 | 00703 |
| 02C0 | 00704 | 00705 | 00706 | 00707 | 00708 | 00709 | 00710 | 00711 | 00712 | 00713 | 00714 | 00715 | 00716 | 00717 | 00718 | 00719 |
| 02D0 | 00720 | 00721 | 00722 | 00723 | 00724 | 00725 | 00726 | 00727 | 00728 | 00729 | 00730 | 00731 | 00732 | 00733 | 00734 | 00735 |
| 02E0 | 00736 | 00737 | 00738 | 00739 | 00740 | 00741 | 00742 | 00743 | 00744 | 00745 | 00746 | 00747 | 00748 | 00749 | 00750 | 00751 |
| 02F0 | 00752 | 00753 | 00754 | 00755 | 00756 | 00757 | 00758 | 00759 | 00760 | 00761 | 00762 | 00763 | 00764 | 00765 | 00766 | 00767 |

INHALTSVERZEICHNIS 1981

Tricks und Tips

| | | |
|---|------|---|
| Listener | 1/81 | 2 |
| Software-Toolkit (Auto-Renumber) | 1/81 | 5 |
| Hardcopy-Programme | 2/81 | 2 |
| Graphikzeichen auf dem CBM 8000 | 2/81 | 6 |
| INPUT-Routinen | 3/81 | 2 |
| Listschutzprogramme | 4/81 | 2 |
| Lagerbuchhaltung, Adressverwaltung (Einführung) | 5/81 | 2 |
| Lagerbuchhaltung, Adressverwaltung (Menu) | 5/81 | 4 |
| Nachtrag Listschutz | 5/81 | 8 |
| Overlaytechniken | 6/81 | 2 |
| Fehlermeldung und Statusregister | 6/81 | 3 |

Hardware

| | | |
|----------------------------------|------|---|
| Digital-Analog-Wandler | 1/81 | 9 |
| USER-PORT-ERWEITERUNG Basisgerät | 2/81 | 7 |
| Verstärkerbaustein | 3/81 | 4 |
| Digital-Analog-Wandler | 4/81 | 6 |
| Analog-Digital-Wandler | 5/81 | 7 |
| Temperaturzusatz | 6/81 | 8 |

Programm des Monats

| | | |
|---|------|----|
| Muehle-Spiel | 1/81 | 11 |
| Nachtrag Skirennen | 1/81 | 15 |
| Wettervorhersage | 2/81 | 10 |
| Hornusse | 3/81 | 6 |
| Bildschirmausdruck zum Muehleprogramm 1/1 | 3/81 | 11 |
| Netzteil Berechnung | 4/81 | 8 |
| Trimmtrainer | 5/81 | 9 |
| Musikprogrammierung | 5/81 | 14 |
| Rubik-Wuerfel | 6/81 | 11 |

Maschinensprache

| | | |
|--|------|----|
| Einführung Assemblerprogrammierung | 1/81 | 16 |
| Assemblerprogrammierung (Assemblerarten) | 2/81 | 15 |
| Rundung von Zahlen | 2/81 | 19 |
| Assemblersprache | 3/81 | 14 |
| Assemblerprogrammierung PRINTOUT | 4/81 | 13 |
| Speicherbelegung | 5/81 | 18 |
| Musikprogrammierung in Maschinensprache | 6/81 | 20 |

Neuheiten

| | | |
|---|------|----|
| HRG-Soft-Toolkit | 1/81 | 19 |
| Nachtrag HRG-Soft-Toolkit | 2/81 | 9 |
| Toolkit für CBM 4000 und 8000 | 2/81 | 20 |
| BASIC-Erweiterung | 3/81 | 19 |
| Disk-0-Pro | 3/81 | 20 |
| ROM-Routinen, Code-Listen, Speicherbelegung | 4/81 | 19 |

Leichte Kost

| | | |
|------------------------|------|----|
| Miniinterrupt | 1/81 | 21 |
| Bildschirmcodeprogramm | 1/81 | 21 |
| Kalenderblatt | 5/81 | 20 |
| Surprise-Programm | 6/81 | 6 |

ERSCHEINUNGSDATEN

DER CBM/PET NEWS 1982

| | |
|-----------|------------|
| Heft 1/82 | 22.02.1982 |
| Heft 2/82 | 26.04.1982 |
| Heft 3/82 | 28.06.1982 |
| Heft 4/82 | 06.09.1982 |
| Heft 5/82 | 25.10.1982 |
| Heft 6/82 | 13.12.1982 |

CBM/PET NEWS

sind nur im Abonnement pro Kalenderjahr erhältlich. Bereits erschienene Nummern des Jahrgangs werden nachgeliefert.



Computer-Anlagen für
Klein- und Mittelbetriebe
Monroe, Commodore, ABC 80

Peter Nellen, EI.-Ing. HTL
Balfrinstr. 15, 3930 Visp
Tel. 028 46 41 21

Ihr Commodore-Wiederverkäufer im
Raum Basel

BD-Electronic

Grosse Auswahl an Software, wie z. B.
– Finanzbuchhaltung – Datenverwaltung
– Textverarbeitung – Lagerverwaltung

Gundeldingerstr. 313, Basel

Telefon 061 35 05 17



**Computer
Shop
Luzern**

- Grösste Auswahl an Geräten, Programmen, Büchern
- Kompetente Fachberatung
- Schnelle Serviceleistung, eigener technischer Dienst
- Grösstes Computer-Versandhaus
- Basic-Programmierkurse

Seeburgstrasse 18, 6002 Luzern, Tel. 041 31 45 45



Rüfenacht AG
CH-4950 Huttwil Tel. 063 72 11 13

Fertige Lösungen für Ihren
Klein- oder Mittelbetrieb

- Finanzbuchhaltung
- Liegenschaftsverw.
- Textverarbeitung
- Fakturierung/Debitoren
- Adressverwaltung
- Lagerverwaltung

LOGON AG

Baslerstrasse 145
8048 Zürich
Telefon 01 62 59 22

Zugerstrasse 69
8048 Zürich
Telefon 042 31 44 80

Instant-Soft

In Zürich exklusiv bei:

**fürer-
bürocomputer**

Tödistrasse 48, 8021 Zürich
Telefon 01 202 49 92

COMPUTER SHOP

Mikrocomputer + Software

- Schulung
- Beratung
- Service
- Verkauf

Dornacherstrasse 161
CH-4053 BASEL
Tel. 061 / 35 31 14

Ihr Fachhändler für
COMMODORE:

Helfenstein + Bucher AG

HB Hirschengraben 43,
6002 Luzern
Telefon 041 22 13 43

Komplettes **COMMODORE**-Programm
ab Februar 1982 in neuen, separaten
Demo-Räumen.

Standard-Software oder «mass-
geschneidert» auf alle speziellen Kunden-
wünsche.

Regionalvertretung für Elektriker-Software



Pius Schäfler
St. Gallerstr. 44, Ringstr. 5
Gossau, Tel. 071 85 13 87

DATATECHNIK

H. U. GURTNER
NEUGASSE 17
3800 INTERLAKEN

TEL. 036 22 10 21

Die Welt der Mikrocomputer ComputerLand®

We know small computers.

Zentral-/Zweierstr., 8036 Zürich
Tel. 01 - 35 62 10/11

Ihr Commodore-Spezialist in Thun

HBM electronic

Frutigenstr. 3, 3600 Thun

Tel. 033 22 66 88

CBM/PET NEWS

Verlag, Redaktion, Inserate
Informa Verlag AG
Postfach 1401, CH-6000 Luzern 15
Telefon 041 - 31 18 46, Tx 72 227 (dclch)
Postcheck-Konten: Luzern 60 - 27181,
Stuttgart 3786-709, Wien PSK 7975.035

Manuskripte

Mit der Zustellung von Manuskripten anerkennt
der Autor die Copyrightbestimmungen des
Verlages. Mit der Annahme von Manuskripten
durch die Redaktion und der Autor-Honorierung
durch den Verlag hat dieser das Recht zur
Veröffentlichung der entsprechenden Beiträge
in anderen verlageeigenen Publikationen und
zur Übersetzung in andere Sprachen erworben.
Für die Veröffentlichung wird keine Gewähr
oder Garantie übernommen, auch nicht dafür,
dass die verwendeten Schaltungen, Firmen-
namen und Warenbezeichnungen usw. frei von
Schutzrechten Dritter sind. Die Verwendung
der Informationen erfolgt auf eigenes Risiko.

© 1982 by Informa Verlag AG, Luzern, aber
Speicherung in Datenverarbeitungsanlagen für
den eigenen Gebrauch erlaubt. Nachdruck,
auch auszugsweise, sowie Vervielfältigungen
jedwelter Art nur mit schriftlicher Genehmi-
gung des Verlages und unter voller Quellen-
angabe.
ISSN 0251-2017/Printed in Switzerland

Alles unter einem Dach

- professionelle Kleincomputer
- Komplettangebot – vom Gerät bis zur
Arbeitsplatzgestaltung und
-organisation
- garantierte Wartung, auch nach
dem Kauf



Wettingen: 056 27 01 27

Nowak AG

Edig. konz. Fachgeschäft

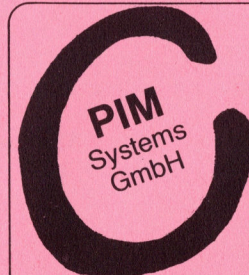
Technikumstrasse 46
8400 Winterthur
Telefon 052/22 08 03



LASYS AG

Reherstrasse 21
9014 St. Gallen

Tel. 071 24 96 76



053
45 45 0

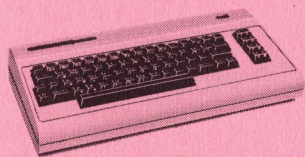
Textverarbeitung
Adressverwaltung
Lagerverwaltung

Finanzbuchhaltung

Lochstr. 18 8204 Schaffhausen

Einen Computer für die Buchhaltung, für das Lager, zum Textverarbeiten, zum Planen und Forschen, zum Studieren, zum Spielen?

Also einen Commodore.



Commodore Volkscomputer VC20

Preis: Fr. 795.-
 Speicher: 5k Bytes RAM, Erweiterung bis 32k RAM möglich
 Bildschirm: 22 Zeichen, 23 Linien
 Farben: 8farbig
 Ton: 3 Ton-Generatoren, 3 Oktaven
 Sprachen: Basic V 2.0, Maschinsprache 6502
 Anschluss: an jeden Fernseher anschliessbar



Commodore CBM 4032

Preis: Fr. 2'975.-
 Speicher: 32k Bytes RAM
 Bildschirm: 40 Zeichen, 25 Linien
 Sprachen: Basic 4, Maschinsprache 6502



Commodore CBM 8032

Preis: Fr. 3'475.-
 Speicher: 32k Bytes RAM
 Bildschirm: 80 Zeichen, 25 Linien
 Sprachen: Basic 4, Maschinsprache 6502



Commodore MMF

Preis: Fr. 4'975.-
 Speicher: 96k Bytes RAM, 48k Bytes ROM
 Bildschirm: 80 Zeichen, 25 Linien
 Sprachen: Basic, Fortran, Cobol, Pascal, APL, Assembler 6502/6809

Commodore Computer gibt es für alles und jedermann. Und weil wir unseren Kunden nicht nur eine breite Hardware-Palette, sondern auch ein umfassendes Software-Programm bieten, sind weltweit schon über eine Viertelmillion Commodore im Einsatz.

Unsere Computer werden aus gutem Grund nur über erfahrene Wiederverkäufer vertrieben: sie prüfen Ihr EDV-Problem und erarbeiten auf Wunsch gleich eine Gesamtlösung.

Eines ist sicher: wir haben auch für Sie eine effiziente und kostengünstige* Lösung. Senden Sie uns also heute noch den Coupon, damit wir Sie eingehend ins Bild setzen können.

*weitere Geräte aus unserem Hardware-Angebot: Floppy Disk (2 x 176k Bytes) Fr. 3'175.-, Drucker ab Fr. 1'750.-.

Informations-Gutschein:

- Ja, senden Sie mir bitte Unterlagen über Commodore.
 Ich möchte Commodore Computer persönlich kennenlernen.

Absender: _____

Bitte an Commodore AG, Aeschenvorstadt 57, 4010 Basel, senden.

commodore
 COMPUTER

Commodore AG, Aeschenvorstadt 57, 4010 Basel,
 Tel. 061 23 78 00, Tlx 64961 cbm ch

C/PN

2-82