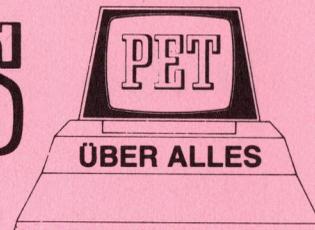


CBM/PET NEWS



AZ CH-6006 Luzern / Verlag SCC AG, Seeburgstrasse 12 / Erscheint 6mal jährlich

Lieber PET-Freund,

während auf dem deutsche Markt schon seit einigen Monaten der neue Commodore Computer VC 20 erhältlich ist, müssen wir uns hier noch einige Wochen gedulden, bis dieser lang erwartete "Volkscomputer", um auch einmal dieses Schlagwort zu gebrauchen, geliefert werden kann. Hoffentlich dauert es dann nicht mehr zu lange, bis auch die angekündigten Erweiterungen und die Peripherie zum Verkauf frei gegeben werden.

Gerade für den Beginner, der normalerweise mit finanziellen Mitteln nicht gerade reich gesegnet ist, füllt dieses Gerät mit seinem niedrigen Preis und seinen relativ grossen Ausbaumöglichkeiten, eine grosse Lücke im bisherigen Commodore Angebot. Nicht zuletzt durch die enormen Gestaltungsmöglichkeiten, die durch die Einführung der Programmierung in Farbe um ein vielfaches gesteigert worden sind. Während der grosse Bruder, der CBM 8000, der Computer für den Profi bleibt, ist der VIC 20 also in erster Linie dem Hobbyprogrammierer vorbehalten.

Wir möchten daher schon jetzt alle Besitzer eines VC 20, vor allem in der BRD, bitten, uns möglichst Ihre Erfahrungen, die Sie mit dem VC 20 gemacht haben aber auch Ihre Programme, die Sie auf diesem Gerät entwickelt haben, zuzusenden. Denn es ist unser Bestreben, in jeder Beziehung auf dem laufenden zu sein und vor allem auch Ihnen, immer einen aktuellen Inhalt unserer Zeitschrift anzubieten. Wir unsererseits werden bestrebt sein, auch möglichst sofort nach Erscheinen des Gerätes hier auf dem Markt, Ihnen mit Informationen und technischen Details aufzuwarten.

Im letzten Heft hatten wir angekündigt, dass wir in dieser Ausgabe mit detaillierten semiprofessionellen Programmen auf dem CBM 8000 beginnen werden. Wir hatten an diese Ankündigung die Bitte geknüpft, uns Wünsche und Anregungen zu diesem Themenkreis mitzuteilen, bisher leider ohne grossen Erfolg. Wir hoffen aber, dass nach dem Vorliegen dieser ersten Folge, Denkanstösse und Wünsche in grosser Zahl bei uns eintreffen werden, da die Programme, die wir Ihnen vorstellen, nur ein Schema darstellen können, zu dem jeder ernsthafte Programmierer seine eigenen Vorstellungen und Ideen hat und die gerade in den Feinheiten vom jeweiligen Benutzer entsprechend abgewandelt und ergänzt werden müssen. Aber gerade deshalb ist uns Ihre Meinung und Ihre Diskussion so enorm wichtig.

Die Redaktion

Heinz Kastien

Tricks und Tips

LAGERBUCHHALTUNG

ADRESSVERWALTUNG

In dieser Folge beginnen wir mit der Besprechung von Routinen kommerzieller Programme. Diese Programme erfordern in allen Fällen neben dem Rechner einen Printer und eine Floppy-Station. Die Besprechung der Programme nehmen wir jeweils am Beispiel eines CBM 3032 mit Floppy 3040 und eines CBM 8032 mit Floppy 8050 vor. Auf die Auswahl des Printers werden wir je nach den Erfordernissen zu gegebenem Zeitpunkt noch separat eingehen.

Bei der kommerziellen Datenverarbeitung, z.B. bei Programmen für Adressverwaltung, Lagerbuchhaltungen, Statistiken usw. muss man sich in jedem Fall vor Beginn der eigentlichen Programmierarbeit, wenn nicht sogar vor dem Kauf des Computers und der Floppy-Station, genaue Rechenschaft über einige wichtige Fakten ablegen. Details und Programmbeispiele sollen hier, als auch in den Fortsetzungen dieser Serie immer am Beispiel einer Lagerverwaltung und einer Adressverwaltung demonstriert werden, gleichgut lassen sich natürlich auch andere Beispiele wie Fakturierung oder Debitorenbuchhaltungen als Beispiel wählen. In jedem Fall stellen sich am Anfang folgende Fragen:

A. Was soll mein Programm alles können. Es ist also ein Pflichtenheft zu erstellen, in dem die Aufgaben, die das Programm zu bewältigen hat, klar definiert sind. Dieses Pflichtenheft ist enorm wichtig, da es relativ schwierig ist, in ein bestehendes Programm nachträglich andere Programmteile einzuflechten.

B. Die maximale Anzahl der Daten ist genau festzulegen. Es genügt aber nicht einfach zu sagen: "Ich habe 5000 Artikel", sondern es müssen auch die Details, wie Artikelnummer, Artikelname, Preis, Lagerbestand, Lieferant usw. festgelegt werden. Ja diese Forderung geht soweit, dass möglichst sogar die Anzahl Buchstaben pro Name definiert wird, da erst dann der genaue, erforderliche Speicherplatz berechnet werden kann. Daraus ergeben sich für ein Kleincomputersystem folgende Problemstellungen:

1. Der Speicherplatz dieser Systeme ist meist beschränkt, die Programme haben jedoch in der Regel einen recht grossen Umfang. Da aber noch eine gewisse Reserve an Speicherplatz vorhanden sein soll, um Daten in den Rechner zu übernehmen, ist es nicht möglich, gleichzeitig das gesamte Programm im Rechner selbst abzuspeichern. Man teilt daher das Programm in mehrere Programmteile ein und speichert Sie auf einer Diskette ab. Je nach Bedarf werden dann die einzelnen Programmteile über ein Menu abgerufen und in den Rechner geladen.

Es ist also zu überlegen, welche Programmteile sich in jedem Fall gleichzeitig im Rechner befinden sollen, um nicht durch unnötig viele Abspeicher und Ladevorgänge das Programm zu stark zu verlangsamen.

2. Es ist entscheidend, wieviele Daten verwaltet werden sollen und welche Daten sich möglichst gleichzeitig im Rechner befinden sollen, um auch hier durch ständige neue Ladevorgänge die Abarbeitung des Programms nicht zu verlangsamen.

Ebenso wichtig ist die genaue Definition der Daten, also Anzahl, numerisch, alphanumerisch oder gemischt, grösster und kleinster Wert und Anzahl der Stellen einer Variablen.

3. Wenn der Speicherplatz einer Diskette nicht ausreicht, um alle Daten aufzunehmen, so ist ebenfalls festzulegen, welche Daten zusammengehören und auf der gleichen Diskette abgespeichert werden sollten. Auch hier wird durch eine gute Einteilung bei der Programmausführung viel Zeit gewonnen.

Es hat sich herausgestellt, dass es in jedem Fall zweckmässig ist, Programm und Daten getrennt auf verschiedenen Disketten abzuspeichern, schon allein aus Gründen der Datensicherung.

4. Da die Geräte bei der kommerziellen Datenverarbeitung meist den ganzen Tag in Betrieb sind, aber durch das Floppy gewisse Programmteile öfters neu geladen werden, gehen bekanntlich bei jedem neuen Ladevorgang alle Variablen verloren, die nicht auf Diskette abgespeichert worden sind. Dies bezieht sich z.B. auf das Datum und die Uhrzeit, die also bei jedem neuen Programmteil frisch eingegeben werden müssen. Es hat sich daher als vorteilhaft erwiesen, diese Variablen, die während des ganzen Tages immer gleich benutzt werden, im 2. Kassettenbuffer oder, da dies beim CBM 8000 nicht möglich ist, im oberen Teil des RAM oder im ersten Kassettenbuffer abzulegen, aber vor dem Löschen und Ueberschreiben mit BASIC zu schützen.

5. Schliesslich spielt die Auswahl des Printers eine nicht unwesentliche Rolle.

Ein Nadeldrucker wie z.B. der Commodore 3022 ist relativ schnell, ergibt aber durch die Matrix und die fehlenden Unterlängen ein nicht allzu gut leserliches Schriftbild. Ein Typenraddrucker wie der Commodore 8026 ist langsam, dafür ist die Schrift sehr gut leserlich und durch Auswechseln des Typenrads sind verschiedene Schriften möglich. In vielen Fällen wird man daher sogar zwei verschiedene Printer wählen und von Fall zu Fall den einen oder anderen Printer nehmen.

Wie bereits im Abschnitt 1 erwähnt worden ist, geht man meist in der Art und Weise vor, dass das Programm in mehrere Teile unterteilt wird und über ein Menu die einzelnen Programmteile aufgerufen werden können. Für diese Menutechnik sind verschiedene Methoden bekannt. Eine dieser Methoden werden wir für eine Lagerbuchhaltung auf dem CBM 8032 und eine Adressverwaltung auf dem CBM 3032 beschreiben.

Es sollen die aufgezeigten Charakteristika am Beispiel einer Adressverwaltung praktiziert werden. Hierbei ist zu beachten, dass bei der Bedienung der Geräte meist Personen zum Einsatz gelangen, die in der Programmierung unkundig sind und von denen also keinerlei Gerätekenntnisse verlangt werden können. Daher muss die Bedienung der Geräte möglichst einfach sein und Eingabefehler müssen vom Rechner durch entsprechende Prüfung der Daten von vorneherein vermieden werden. Bei der Gestaltung der nachfolgenden Programme soll auf diese Tatsache möglichst grossen Wert gelegt werden.

Dies beginnt schon beim Einlegen der Diskette, indem man das erste Programm nur mit dem Befehl `LOAD"*",8` beim CBM 3000 oder mit `DLOAD` beim CBM 8000 lädt. Vom ersten geladenen Programm sollen also alle anderen Programme leicht zugänglich sein. Dies ist mit der oben beschriebenen Menutechnik leicht möglich.

Das nachfolgende Programm stellt die Eingabe und das Menu einer einfachen Adressverwaltung dar, mit der 1000 Adressen verwaltet werden können. Der Speicherbedarf für diese Adressmenge beträgt ungefähr 30 kB, die Daten haben also auf einer Disk des Floppy 3040 Platz.

Die abgespeicherten Adressen sollen folgenden Umfang aufweisen :

1. Anrede
2. Vorname
3. Geschlechtsname
4. Strasse und Hausnummer
5. Postleitzahl und Ort
6. Telefonnummer
7. Bemerkungen

Eine Adresse dieses Umfanges benötigt zur Abspeicherung auf Disk ca. 12 Byte, je nach Anzahl Buchstaben.

Das Programm seinerseits, soll die unten aufgeführten Arbeitsschritte zulassen:

1. Eingabe der Adressen
2. Abruf von Adressen auf dem Bildschirm
3. Ausdruck der Adressen auf Etiketten
4. Adressmutationen
5. Ausdruck einer gesamten Adressliste

Das nun folgende Listing zeigt das Menu und das Laden weiterer Programme. Es ist auf einem CBM 3032 zusammen mit einem Floppy 3040 programmiert. Eine genaue Besprechung des Programms erübrigt sich, da es bis auf die Cursorbewegungen und die Namen im Menu mit der nachfolgenden Lagerbuchhaltung identisch ist.

ADRESSVERWALTUNG MENU

```

1000 REM MENU
1010 POKE59468,14:PRINT"XXXXXXXXXXXXXXXXXXXX*** \ - / , ***"
1020 PRINT"XXXXXXXXXXXXXXXXXXXX1. # ADRESSEINGABE"
1030 PRINT"XXXXXXXXXXXXXXXXXXXX2. # ADRESSMUTATION"
1040 PRINT"XXXXXXXXXXXXXXXXXXXX3. # ADRESSABRUF"
1050 PRINT"XXXXXXXXXXXXXXXXXXXX4. # ADRESSAUFKLEBER"
1060 PRINT"XXXXXXXXXXXXXXXXXXXX5. # ADRESSLISTE"
1070 GETZ:IFZ=0ORZ>5THEN1070
1080 ONZGOTO1100,1110,1120,1130,1140,1150
1090 END
1100 P#="EINGABE":GOTO1200
1110 P#="MUTATION":GOTO1200
1120 P#="AUSGABE":GOTO1200
1130 P#="KLEBER":GOTO1200
1140 P#="LISTE":GOTO1200
1200 POKE59468,12:PRINT"LOAD";CHR$(34);"0:";P#;CHR$(34);",8"
1210 PRINT"RUN";:POKE158,2:POKE623,13:POKE624,13:END
READY.

```

Gleichzeitig zeigen wir die Menutechnik am Beispiel einer Lagerbuchhaltung, programmiert auf einem CBM 8032 mit Floppy 3050. Hier wird vor dem Laden des Menu das Datum eingegeben, auf seine Richtigkeit geprüft und im oberen Teil des RAM abgelegt. Erst dann wird das Menu geladen, von dem alle weiteren Programme zugänglich sind.

Das Lagerbuchhaltungsprogramm ermöglicht folgende Teilprogramme:

1. Eingabe der Artikel
2. Abruf einzelner Artikel
3. Mutation der Artikel
4. Ausdruck einer Artikelliste
5. Inventar

LAGERBUCHHALTUNG MENU

```

1000 REM MENU
1010 PRINT"#####13*** \ - / , ***"
1020 PRINT"#####131. # ARTIKELEINGABE
1030 PRINT"#####132. # ARTIKELMUTATION"
1040 PRINT"#####133. # ARTIKELAUSGABE
1050 PRINT"#####134. # INVENTAR
1060 PRINT"#####135. # ARTIKELLISTE
1070 GETZ:IFZ=0ORZ>5THEN1070
1080 ONZGOTO1100,1110,1120,1130,1140,1150
1090 END
1100 P$="EINGABE":GOTO1200
1110 P$="MUTATION":GOTO1200
1120 P$="AUSGABE":GOTO1200
1130 P$="INVENTAR":GOTO1200
1140 P$="LISTE":GOTO1200
1200 PRINT"#####LOAD";CHR$(34);"0:";P$;CHR$(34);",8"
1210 PRINT"#####RUN";:POKE158,2:POKE623,13:POKE624,13:END

```

LAGERBUCHHALTUNG CODE & DATUM

```

1000 POKE52,200:POKE53,123:DIMM(12):REM DATUM
1010 PRINT"#####INGABE DES DATUMS "
1020 PRINT"#####1301.08.1980#####";
1025 GOSUB2000:M$=X$:GOSUB3000
1030 A(1)=ASC(A1$):A(2)=ASC(A2$):A(3)=ASC(A3$):A(4)=ASC(A4$):
1040 A(6)=ASC(A6$):A(7)=ASC(A7$):A(8)=ASC(A8$)          A(5)=ASC(A5$)
1050 FOR I=1TO8:POKE 30999+I,A(I):NEXT
1060 PRINT"#####LOAD";CHR$(34);"0:MENU";CHR$(34);",8"
1070 PRINT"#####RUN";:POKE158,2:POKE623,13:POKE624,13:END
1080 END
2000 X$="":ANZ=0:PRINT"  _";
2010 GETZZ$:IFZZ$=""THEN2010
2020 ZZ=ASC(ZZ$)
2030 IFZZ=20THEN2090
2040 IFZZ=13THEN2150
2050 IFZZ<46ORZZ>57ORZZ=47THEN2010
2060 X$=X$+ZZ$:ANZ=ANZ+1
2070 PRINTZZ$;" _";
2080 GOTO2010
2090 IFANZ=1THENX$="":ANZ=0:GOTO2130
2100 IFANZ<1THEN2010
2110 ANZ=ANZ-1
2120 X$=LEFT$(X$,ANZ)
2130 PRINT"  _";
2140 GOTO2010
2150 PRINT" "
2160 RETURN
3000 IFLEN(M$)<>10THEN3160
3010 A1$=LEFT$(M$,1):A2$=MID$(M$,2,1):A3$=MID$(M$,4,1):
3015 A5$=MID$(M$,7,1):A6$=MID$(M$,8,1):A7$=MID$(M$,9,1):
3020 IFASC(A1$)<48ORASC(A1$)>57THEN3160          A4$=MID$(M$,5,1)
3030 IFASC(A2$)<48ORASC(A2$)>57THEN3160          A8$=RIGHT$(M$,1)
3040 IFASC(A4$)<48ORASC(A4$)>57THEN3160
3050 IFASC(A3$)<48ORASC(A3$)>57THEN3160

```

```

3060 IFVAL(RIGHT$(M$,4))<1000THEN3160
3070 IFVAL(MID$(M$,4,2))<10RVAL(MID$(M$,4,2))>12THEN3160
3080 MT=VAL(MID$(M$,1,2))
3090 MM=VAL(MID$(M$,4,2))
3100 RESTORE
3110 FORY=1TO12
3120 READM(Y):NEXT
3130 IFVAL(RIGHT$(M$,4))/4=INT(VAL(RIGHT$(M$,4))/4)THENM(2)=29
3140 IFMT>M(MM)THEN3160
3150 RETURN
3160 PRINT"7"#####" :GOTO 1010
3170 END
3190 DATA31,28,31,30,31,30,31,31,30,31,30,31

```

Programmbeschreibung

Zeile 1000 Reversation des Speicherplatzes für das Datum und Dimensionierung der Felder

Zeile 1010 - 1020 Titel

Zeile 1025 Aufruf Unterroutinen für GET-Routine und Datumroutine

Zeile 1030 - 1050 Umwandlung des Datums in ASCII-Code und Ablegen im Speicherplatz 31000-31008

Zeile 1060 Laden des Menu ab Floppy, das Laden erfolgt mit dem LOAD-Befehl auf dem Bildschirm und den beiden RETURN im Tastaturbuffer

Zeile 2000 - 2160 GET-Routine, siehe hierzu CBM/PET NEW's Heft 3/81

Zeile 3000 Prüfen des Datumstring auf richtige Länge.

Zeile 3010 - 3015 Zerlegen des Strings

Zeile 3020 - 3050 Prüfen des Tages und Monats auf Zahleneingabe

Zeile 3060 Prüfen der Jahreszahl auf realen Wert

Zeile 3070 Prüfen des Monats auf Zahl zwischen 1 und 12

Zeile 3080 - 3090 Umwandlung des Tages und Monates in num. Variable

Zeile 3100 - 3120 Einlesen der DATA-Statements (Anzahl der Tage im Monat)

Zeile 3130 Korekktur des Monats Februar bei Schaltjahren.

Zeile 3140 Vergleich des Tages mit der grösstmöglichen Zahl

Zeile 3160 Löschen des Datums bei falscher Eingabe und neue Abfrage.

Zeile 3190 DATA-Statements.

Eine Programmbeschreibung zu den beiden Menuprogrammen (Adressverwaltung auf CBM 3032 und Lagerbuchhaltung auf CBM 8032) erübrigt sich, da die Programme sehr einfach sind. Das Laden weiterer Programme erfolgt mit der gleichen Routine, wie im Datumprogramm beschrieben.

RUBIK - WUERFEL

Ab sofort ist das Lösungsprogramm für Rubik's Zauberwürfel nicht nur Apple-Besitzern und PASCAL-Bewanderten vorbehalten. Für den CBM/PET ist jetzt ein 6 kByte BASIC-Programm "RUBIK" erhältlich (läuft also auch auf dem 8 k-PET!).

Ueber den Würfel an sich sind keine grossen Worte mehr zu verlieren. Wichtig ist nur, dass sich nicht jede beliebige Stellung erdreuen lässt. Bei einer unmöglichen Kombination gibt der Computer früher oder später auf dem Bildschirm be kannt:

"Ihr Würfel ist falsch zusammengesetzt!"

Zusammen mit dem Programm wird eine ausführlich Programmbeschreibung geliefert. Erhältlich ist "RUBIK" beim SCC unter der Best. Nr. P 4600 für Fr. 24.--.

Der Kern der Schaltung ist ein integrierter A/D-Wandler MC 14433, der ein 3 1/2 stelliges Multiplex BCD-Signal erzeugt. Dieses Multiplexsignal wird mit einem vierfach Analogswitch CD 4016 und einem vierfach Latch CD 4042 in ein 8-bit-Signal umgewandelt, das direkt dem USER-PORT zugeführt wird. Zusätzlich wird ein ZN 4 58 zur Erzeugung der Referenzspannung sowie ein 7404 zur Erzeugung des Clock benötigt.

Die Schaltung besticht durch den geringen Aufwand an externen diskreten Bauelementen und durch nur eine Versorgungsspannung von + 5 Volt.

Der beschriebene Wandler weist folgende technische Daten auf :

Auflösung	3 1/2 stellig
max. Anzeige	1.999 V
Genauigkeit	0.1 %
Nullpunkt	automatisch bei Eingang Null
Polarität	positiv/negativ
Ueberspannung	"Overload"
Schutz	bis +/- 300 V

Zum Einbau des A/D-Wandlers muss ein mitgelieferter fünfter Stecker in die Basisplatine der USER-PORT Erweiterung eingelötet werden. Die acht Ausgänge und die Versorgungsspannung von 5 V bezieht der A/D-Wandler direkt aus dem Basisgerät. In die + 5 V Spannungsversorgung sollte ein einpoliger "Device Select"-Schalter gelegt werden, der es gestattet, den Wandler abzuschalten. Die Eingänge des A/D-Wandlers werden an zwei zusätzliche, in der Rückwand der USER-PORT-Erweiterung anzubringende Buchsen geführt.

Die Schaltung wird fertig montiert und abgeglichen unter der Nr. P 9108 zum Preis von Fr. 210.-- geliefert. Software auf Kassette ist im Preis enthalten.

LISTSCHUTZ AUS HEFT 4/81

Bei den Programmen zum "Listschutz" sind von den Autoren nachträglich noch einige Verbesserungen vorgenommen worden, die wir Ihnen nicht vorhalten möchten.

Security-Maschinenprogramm

Adresse 7946 muss von 00 auf hex. 20 geändert werden.

Das zweite inzwischen veränderte Porgramm bringen wir der Sicherheit halber komplett mit den DATA-Statements für das CBM 3000 und PET 2001 System.

```
19 FORA=800T0895:READB:POKEA,B:NEXT
20 DATA 165,119(201),133,17(8),165,120(202),133,18(9),162,23
25 DATA 189,249(181),224,149,112(194),202,208,248
28 DATA 165,17(8),133,119(201),165,18(9),133,120(202)
30 DATA 169,76,133,121(203),169,71,133,122(204),169,3
40 DATA 133,123(205),96,201,155,208,21,72,138,72
50 DATA 186,189,4,1,201,198,208,7,189,3
60 DATA 1,201,249(235),240,10,104,170,104,201,58
70 DATA 176,226,76,125(207),0,162,8,189,119,3
80 DATA 32,210,255,202,16,247,160,122,76,125(133)
90 DATA 245,71,78,73,84,83,73,76,63,17
READY.
```



Programm des Monats BASIC-BASIC

TRIMMTRAINER

Schon in den letzten CBM/PET News haben wir ein Programm veröffentlicht, das zur Erhaltung der Fitness beitragen soll, denn bekanntlich ist langes Sitzen vor dem Computer der Gesundheit nicht gerade förderlich.

Im genannten Programm hatten wir aber nur den Kalorienverbrauch berücksichtigt, ohne auf die eingerostete Muskulatur zu achten. Diesem Uebelstand soll nun mit diesem Fitnessprogramm abgeholfen werden.

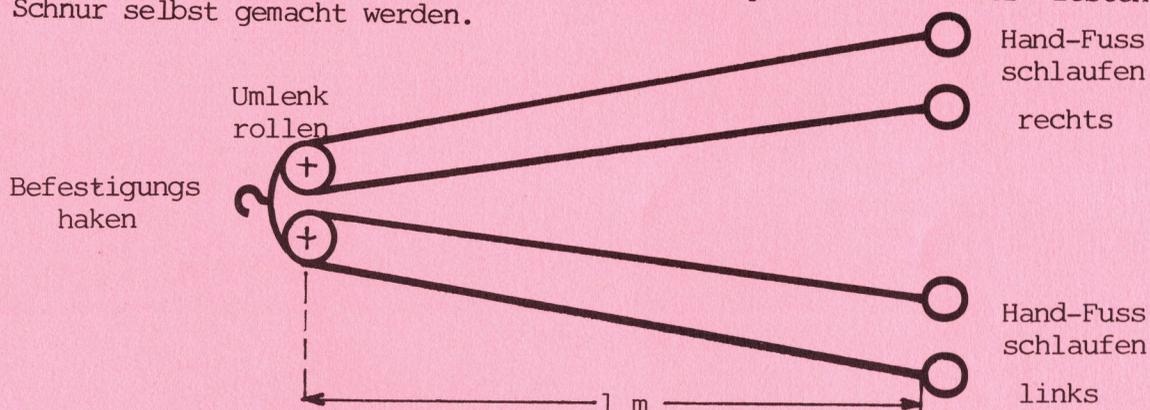
Dieses Programm kann unter Zuhilfenahme eines kleinen Gymnastikgerätes helfen, Ihren physischen Zustand auf ein möglichst hohes Niveau zu bringen oder zu halten.

Der Trimmtrainer gemäss dem Photo ist zum Preis von DM 89,-- durch den Verfasser des Programms erhältlich

Willi Gietmann
Ahornstr. 9
D 4174 Issum 1

Tel. 02835/2381
Kontonr. 435 214
Kreissparkasse Geldern

andererseits kann das Gerät aber auch nach untenstehender Zeichnung sehr leicht aus zwei Kunststoffrollen und ein paar Metern einer festen Schnur selbst gemacht werden.



Zu diesem Programm benötigt man zusätzlich noch einen Verstärker, der am Ausgang CB 2 angeschlossen wird, wie er in Heft 3/81 oder 3/80 beschrieben worden ist.

Das Programm kennt drei Schwierigkeitsstufen mit verschiedenen Trainingszeiten und ist für das schönere Geschlecht, das sich gelegentlich auch mit der Computerei beschäftigt, in einer ca. 20 % leichteren Variante möglich. Die Trainingszeiten liegen zwischen 18 und 45 Minuten und werden in ihrer Schwierigkeit langsam gesteigert.

Das Programm enthält einige Routinen, die in den vergangenen Heften bereits besprochen worden sind, so z.B. die Eingabe der Daten über die Tastatur, die hier als File angesprochen wird, um beim Drücken der RETURN-Taste ein Aussteigen des Rechners zu verhindern.

Auch das Musikprogramm, das zur Untermalung der Entspannungspausen dient, und die Takte werden über das frei laufende Schieberegister erzeugt und sind in Heft 3/80 bereits schon einmal besprochen worden.


```

1150 OPEN4,4
1160 FORI=1TO5:PRINT#4:NEXT
1170 PRINT#4,CHR$(1);CHR$(1);CHR$(1);"      TRIMMTALER"
1180 FORI=1TO2:PRINT#4:NEXT
1190 HE$="HERR  "
1200 IFFF<1THENHE$="FRAU  "
1210 PRINT#4,CHR$(1);HE$;VN$;" ";NA$;" HAT AM ";
1220 G1$=DA$:GOSUB1420:PRINT#4,G2$:PRINT#4
1230 PRINT#4,CHR$(1);"IN DER ZEIT VON ";
1240 G3$=UA$:GOSUB1440:PRINT#4,G4$;" BIS ";
1250 G3$=UE$:GOSUB1440:PRINT#4,G4$
1260 PRINT#4
1270 PRINT#4,CHR$(1);CHR$(1);"      DEN"
1280 PRINT#4:PRINT#4
1290 IFNN=1THENR$="  BRONZENEN"
1300 IFNN=3THENR$="  GOLDENEN"
1310 IFNN=2THENR$="  SILBERNEN"
1320 PRINT#4,CHR$(1);
1330 FORI=1TO3:PRINT#4,CHR$(1);:NEXTI
1340 PRINT#4,R$
1350 PRINT#4:PRINT#4:PRINT#4
1360 PRINT#4,CHR$(1);CHR$(1);" TRIMMTALER ERWORBEN."
1370 FORI=1TO18:PRINT#4:NEXT
1380 CLOSE4
1390 END
1400 PRINT#4:PRINT#4
1410 STOP
1420 REM---UNTERPROGRAMM STRECKEN---
1430 G2$=LEFT$(G1$,2)+". "+MID$(G1$,3,2)+". "+RIGHT$(G1$,2):RETURN
1440 G4$=LEFT$(G3$,2)+": "+MID$(G3$,3,2)+": "+RIGHT$(G3$,2):RETURN
1450 STOP
1460 POKE59464,0:POKE59464,0:POKE59464,R:FORT1=1TOT*T2-40:NEXT:
1470 POKE59464,R:FORT1=1TOT*T2-140:NEXT:RETURN
1480 POKE59466,D:FORI=1TON:FORJ=0TO20:GOSUB1500:NEXTJ:FORJ=20TO0
1490 GOSUB1500:NEXTJ:NEXTI:RETURN
1500 FORK=50TO60:POKE59464,J+K:NEXTK:RETURN
1510 A=INT(26*RND(1)):R$=R$(A):GOSUB1550:GOSUB1460:RETURN
1520 POKE59464,J:FORK=1TO40:NEXTK:RETURN
1530 READR,T:IFR=999THENRETURN
1540 GOSUB1460:GOTO1530
1550 IFR$="H0"THENR=251:RETURN
1560 IFR$="C"THENR=237:RETURN
1570 IFR$="C#"THENR=224:RETURN
1580 IFR$="D"THENR=211:RETURN
1590 IFR$="D#"THENR=199:RETURN
1600 IFR$="E"THENR=188:RETURN
1610 IFR$="F"THENR=177:RETURN
1620 IFR$="F#"THENR=167:RETURN
1630 IFR$="G"THENR=157:RETURN
1640 IFR$="G#"THENR=149:RETURN
1650 IFR$="A"THENR=140:RETURN
1660 IFR$="B"THENR=132:RETURN
1670 IFR$="H"THENR=124:RETURN
1680 IFR$="C1"THENR=117:RETURN
1690 IFR$="C1#"THENR=111:RETURN
1700 IFR$="D1"THENR=104:RETURN
1710 IFR$="D1#"THENR=99:RETURN
1720 IFR$="E1"THENR=93:RETURN
1730 IFR$="F1"THENR=88:RETURN
1740 IFR$="F1#"THENR=83:RETURN
1750 IFR$="G1"THENR=78:RETURN

```

```

1760 IFR$="G1#"THENR=73:RETURN
1770 IFR$="A1"THENR=69:RETURN
1780 IFR$="B1"THENR=65:RETURN
1790 IFR$="H1"THENR=61:RETURN
1800 IFR$="C2"THENR=57:RETURN
1810 IFR$="Z"THENR=0:RETURN
1820 RETURN
1830 DATABB
1840 DATA69,2,0,2,57,2,69,2,93,2,0,2,69,2,93,2,117,2,0,2,93,2
1850 DATA117,2,140,6,0,2,188,2
1860 DATA140,2,117,2,140,2,124,2
1870 DATA140,2,124,2,140,2,149,2,124,2,104,2,124,2,117,2,0,2,140,2,0,2,69,2
1880 DATA0,2,57,2,69,2,93,2
1890 DATA0,2,69,2,93,2
1900 DATA117,2,0,2,93,2,117,2,140,6,0,2,117,2,0,2,117,2,0,2,117,2
1910 DATA0,2,117,2,0,2,69,2
1920 DATA0,2,117,2,0,2,117,4,124,2,0,2,93,2,0,2,93,2,0,2
1930 DATA93,2,0,2,93,2,0,2,57,2,0,2,93,2,0,2,93,4,99,2,0,2
1940 DATA124,2,93,2,78,2,93,2,83,2,93,2,83,2,93,2,99,2,83,2
1950 DATA69,2,83,2,78,2,83,2,78,2,83,2,93,2,78,2,93,2,99,2
1960 DATA93,2,69,2,93,2,99,2,93,2,61,2,93,2,99,2,93,2
1970 DATA57,2,93,2,99,2,93,2
1980 DATA57,2,61,2,69,2,61,2,78,2,83,2,93,2,78,2,0,2,78,1
1990 DATA99,1,78,1,83,2
2000 DATA93,6,0,2,93,2
2010 DATA0,2,78,2,93,2,124,2,0,2,93,2,124,2,157,2,0,2,124,2,157,2,188,6
2020 DATA0,2,132,4,140,2,0,2
2030 DATA104,4,111,2,104,2,78,4,88,2,93,2,88,2,0,2,104,2,0,2,88,2,0,2
2040 DATA69,2,93,2,104,2,0,2,88,2,104,2,124,2,0,2,104,2,124,2,157,6,0,2
2050 DATA157,2,117,2,93,2,117,2,104,2,117,2,104,2,117,2,124,2,104,2,88,2
2060 DATA104,2,93,2,104,2,93
2070 DATA2,104,2,117,2,93,2,117,2,124,2,117,2,81,2,117,2,124,2,117,2
2080 DATA78,2,117,2,124,2,117,2,69
2090 DATA2,117,2,124,2,117,2,69,2,78,2,88,2,78,2,93,2,104,2,117,2
2100 DATA93,2,0,2,104,1,93,1
2110 DATA104,1,93,1,104,1,117,6,0,2,93,2,0,2,93,2,0,2,93,2,0,2
2120 DATA93,2,0,2,57,2,0,2
2130 DATA93,2,0,2,93,4,104,2,0,2,104,2,0,2,104,2,0,2,104,2,0,2
2140 DATA104,2,0,2,61,2
2150 DATA0,2,104,2,0,2,104,4,117,2,0,2,69,4,57,2,69,2,78,4,88,8
2160 DATA69,1,78,1,88,1,93,1
2170 DATA104,12,88,1,93,1,104,1,117,1,132,2,104,2,88,2,104,2,132,2
2180 DATA140,2,132,2,140,2
2190 DATA149,2,0,2,188,2,0,2,177,4,188,2,0,2,140,4,149,2,124,2,104,4,117,2
2200 DATA124,2,117,4,140,1,124,1
2210 DATA117,1,104,1,93,4,117,2,93,2,69,2,5,0,3,93,3,0,2,104,3
2220 DATA117,3,5,124,3,7,117,4
2230 DATA124,9,140,14,0,30,999,0
2240 DATAYY,D1,1,D1,1,D1,1,Z,1,D1,1,D1,1,D1,1,Z,1,F1#,1,Z,1,E1,1,Z,1,D1,2,Z,1
2250 DATAC1#,1,H,1,B,1,H,1,C1#,1,D1,1,Z,1,E1,1,Z,1,A,1,Z,5,F#,1,F,1,F#,1,Z,1
2260 DATAF#,1,F,1,F#,1,Z,1,F#,1,F,1,F#,1,Z,1,E,1,Z,1,D,1,Z,1,F#,1,F,1,F#,1,Z,1
2270 DATAF#,1,F,1,F#,1,Z,1,F#,1,F,1,F#,1,Z,1,H,1,Z,1,A,1,Z,1,A,1,F#,1,G,1,Z,1
2280 DATAH,1,Z,1,A,1,Z,1,A,1,E,1,F#,1,Z,1,H,1,Z,1,A,1,Z,1,A,1,F#,1,E,1,Z,1,C1#
2290 DATA1,Z,1,D,1,Z,1,H,1,Z,1,C#,1,Z,1,A,1,G#,1,A,1,H,1,A,1,G,1,F#,1,Z,1,F#,1
2300 DATAF,1,F#,1,Z,1,F#,1,F,1,F#,1,Z,1,E,1,Z,1,D,1,Z,1,F#,1,F,1,F#,1,Z,1,F#,1
2310 DATAF,1,F#,1,Z,1,F#,1,F,1,F#,1,Z,1,H,1,Z,1,A,1,Z,1,A,1,F#,1,G#,1,Z,1,F1#
2320 DATA1,Z,1,E1,2,Z,2,F#,1,Z,1,E1,1,Z,1,D1,2,Z,2,C1#,2,Z,1,H,1,A,1,G,1,F#,1
2330 DATAE,1,D,1,Z,1,D1,1,Z,1,D1,1,Z,1
2340 DATAXX,0
READY.

```

Im nächsten Heft werden wir Ihnen ein Programm zur Lösung des RUBIK Würfels präsentieren. Daneben veröffentlichen wir ein Programm zum Plotten von Funktionen, das gleichzeitig das Plotten mehrerer Funktionen gestattet.

Programmbeschreibung

Zeile 50 - 160 Maskieren des Bildschirms und Beschriften der Maske.
Zeile 180 Die Tastatur wird als File angesprochen, damit beim Drücken der RETURN-Taste der Rechner nicht aussteigt.

Zeile 190 - 350 Es werden mit INPUT die Variablen erfragt. Durch die Cursorbewegungen gelangt man immer an die richtige Stelle der Bildschirmmaske.

Zeile 380 - 420 Dem Geschlecht des Turners wird ein Faktor, 0.81 für weiblich und 1.0 für männliche Teilnehmer zugeordnet. Der Faktor wurde sportärztlich ermittelt.

Zeile 440 - 490 Eingabe des Schwierigkeitsgrades der Übung.
Zeile 500 Fehlermeldung bei falschen Eingabewerten und Neubeginn.

Zeile 510 - 560 Setzen der Variablen A,B und C und Aufforderung zum Beginn des Programms.

Zeile 570 Warteschleife

Zeile 580 - 830 Der USER-PORT wird als Ausgang geschaltet, CB 2 wird aktiviert, die Aktiv- und Pausenzeiten werden festgelegt. Die Formeln sind vom amerikanischen Cycletraining abgeleitet.

Zeile 840 - 860 Daten für die Berechnung. Eine Änderung dieser Werte stellt die Richtigkeit des Programms in Frage.

Zeile 880 Siehe spezieller Teil über POKE-Befehle

Zeile 890 - 1000 Text für die Sonate von Bach und den Radetzky Marsch

Zeile 1010 - 1090 Verarbeitung der Liederdaten der Zeilen 1830-2340

Zeile 1100 - 1410 Druckroutine für den Ausdruck des Brevet.

Zeile 1430 - 1450 Analyse von Datum und Uhrzeit

Zeile 1460 - 1820 Aktivieren des USER - PORTS und Ausgabe der Töne in Abhängigkeit der Tonhöhe, Tonlänge und Pausenzeit.

Zeile 1830 - 2340 Liederdaten

MUSIKPROGRAMMIERUNG

Im vorgestellten Programm wird mit einigen POKE-Befehlen gearbeitet, die wir etwas näher erklären wollen, ebenso die Erzeugung der einzelnen Töne.

POKE 59467,16

Dieser Befehl schaltet den Ausgangbaustein VIA 6522 als "freilaufenden Oscillator", der eine Rechteckspannung von 32768 Hz erzeugt. Solange dieser Befehl aktiv ist, kann nicht mit dem Kassettenrecorder gearbeitet werden. Sollte dies doch erfolgen, z.B. bei Unterbrüchen, muss mit POKE 59467,0 das Kassetteninterface wieder in Betrieb genommen werden.

POKE 59466,xxx

Mit xxx im Bereich von 1 - 255 wird die Klangfarbe der Wiedergabe bestimmt. Die Binärwerte dieser Zahlen bestimmen die Frequenzform am USER-PORT Ausgang CB2.

Dezimal Binär Frequenzform

1	00000001	
60	00111100	
85	01010101	
104	01101000	
170	10101010	
255	11111111	

keine Frequenz

Binärmuster in spiegelbildlichem Binärmuster ergeben die gleiche Klangfarbe.

POKE 59464,xxx

xxx im Bereich von 1 - 255. xxx tritt als Divisor der Basis-Frequenz auf. POKE 59464,100 erzeugt einen Ton von ca. 328 Hz, der somit in der Nähe des Tons E1 liegt. Die Divisoren von 1 - 4 sind für Musikprogramme ungeeignet, da sie Töne erzeugen, die in den Bereich des Ultraschall fallen. Theoretisch stehen die Frequenzen von 32768 - 128.5 Hz, also fast 5 1/2 Oktaven zur Verfügung. Praktisch lassen sich aber nur ca. 3 Oktaven mit hinreichender Genauigkeit mit POKE - Befehlen darstellen.

NOTE	FREQUENZ NORM	POKE-WERT SOLL	POKE-WERT IST	FREQUENZ IST	ABWEICHUNG IN %
C	130.82	250.47	250	131.07	+ 0.193
CIS	138.60	236.41	236	138.85	+ 0.179
D	146.84	223.15	223	146.94	+ 0.069
DIS	155.57	210.62	211	155.30	- 0.174
E	164.82	198.81	199	164.66	- 0.095
F	174.62	187.65	188	174.30	- 0.184
FIS	185.00	177.11	177	185.13	+ 0.070
G	196.00	167.18	167	196.22	+ 0.110
GIS	207.67	157.79	158	207.39	- 0.134
A	220.00	148.95	149	219.92	- 0.037
B	233.10	140.57	141	232.40	- 0.302
H	246.94	132.68	133	246.38	- 0.228
<hr/>					
C1	261.63	125.25	125	262.14	+ 0.196
CIS1	277.19	118.21	118	277.69	+ 0.182
D1	293.67	111.58	112	292.57	- 0.374
DIS1	311.14	105.31	105	312.08	+ 0.301
E1	329.63	99.41	99	330.99	+ 0.413
F1	349.23	93.83	94	348.60	- 0.182
FIS1	370.00	88.56	89	368.18	- 0.492
G1	392.00	83.59	84	390.19	- 0.486
GIS1	415.31	78.89	79	414.79	- 0.126
A1	440.00	74.47	74	442.81	+ 0.639
B1	466.17	70.29	70	468.11	+ 0.417
H1	493.88	66.35	66	496.49	+ 0.527
<hr/>					
C2	523.26	62.62	63	520.13	- 0.599
CIS2	554.37	59.10	59	555.39	+ 0.184
D2	587.34	55.79	56	585.14	- 0.374
DIS2	622.26	52.66	53	618.26	- 0.642
E2	659.26	49.70	50	655.36	- 0.592
F2	698.46	46.91	47	697.19	- 0.182
FIS2	740.00	44.28	44	744.73	+ 0.639
G2	784.00	41.80	42	780.19	- 0.486
GIS2	830.60	39.45	39	840.21	+ 1.156 *
A2	880.00	37.23	37	885.62	+ 0.639
B2	932.33	35.14	35	936.23	+ 0.418
H2	987.76	33.17	33	992.97	+ 0.527
<hr/>					
C3	1046.50	31.31	31	1057.03	+ 1.006



Man sieht, das die maximale Abweichung der Töne 1,16 % beträgt. Genau lässt sich also kein Ton erzeugen, was aber in den meisten Fällen nicht stört.

In der Praxis sieht das ganze wie folgt aus:

```
100 POKE 59467,16:POKE 59466,51:POKE 59464,100
110 FOR I=1 TO 200:NEXT I:POKE 59467,0
READY.
```

Diese zwei Zeilen erzeugen einen Ton von 328 Hz, der ca. 1/4 Sekunde dauert (Zeitschleife).

Wird ein Mehrsignal benötigt, so stellen die nachfolgenden Zeilen die beste Lösung dar

```
1000 POKE M1,16:POKE M2,Z1:POKE M3,Z2
1010 FOR I=1 TO Z3:NEXT I:POKE M1,0:RETURN
READY.
```

Die 3 Parameter Z1, Z2 und Z3 können beliebig bestimmt werden und mit GOSUB abgerufen werden.

```
100 POKE 59467,16:POKE 59466,51:POKE 59464,100
110 FOR I=1 TO 200:NEXT I:POKE 59467,0
1000 POKE 59467,16:POKE 59466,104
1010 FOR I=1 TO 3:FOR J=20 TO 255 STEP 5
1020 POKE 59464,J:NEXT J:NEXT I:POKE 59467,0
READY.
```

Als direktes Demonstrationsbeispiel der Musikprogrammierung sei das amerikanische Signal "Lichter Löschen ! Nachtruhe" gegeben.

```
100 POKE 59467,16:POKE 59466,104
110 READ A,B:IF A=999 THEN POKE 59467,0:END
120 POKE 59464,A:FOR I=1 TO 60*B:NEXT I:GOTO 110
500 DATA 250,2,0,,2,250,1,0,,2,188,4,0,1,250,2,0,,2,188,1,0,,2
510 DATA 149,4,0,1,250,2,0,,2,188,1,0,,2,149,3,0,,4,250,2,0,,2
520 DATA 188,1,0,,2,149,3,0,,4,250,2,0,,2,188,1,0,,2,149,5,0,1
530 DATA 188,2,0,,2,149,1,0,,2,125,5,0,,2,149,4,0,,2,188,3,0,,2
540 DATA 250,5,0,,2,250,4,0,,2,250,2,0,,4,177,5,999,0
READY.
```

In den Zeilen 100-120 steht das bereits mehrfach erwähnte Programm. In den Zeilen ab 500 sind die notwendigen Notenwerte, Notenzeiten, Pausen und Pausenzeiten als Data Statements enthalten, die in Zeile 110 immer als Paar mit READ A,B gelesen werden. Dieses Paar enthält immer den Noten bzw. Pausenwert und die Spieldauer, die eigentliche Spieldauer wird in der Zeitschleife gebildet.

```
100 N=4000:M=4000
110 READ A$,B:IF A$="XX" THEN 200
120 GOSUB 500:POKE M,A:POKE M+1,B:M=M+2:GOTO 110
200 FOR I=N TO M-2 STEP 2:A=PEEK(I):B=PEEK(I+1)
210 POKE 59467,16:POKE 59466,104:POKE 59464,A
```

```

220 T=TI+B
230 IF T>T THEN 230
240 NEXT I
250 POKE 59467,0:END
500 IF A$="C" THEN A=250:RETURN
510 IF A$="C#" THEN A=236:RETURN
520 IF A$="D" THEN A=223:RETURN
530 IF A$="D#" THEN A=211:RETURN
540 IF A$="E" THEN A=199:RETURN
550 IF A$="F" THEN A=188:RETURN
560 IF A$="F#" THEN A=177:RETURN
570 IF A$="G" THEN A=167:RETURN
580 IF A$="G#" THEN A=158:RETURN
590 IF A$="A" THEN A=149:RETURN
600 IF A$="B" THEN A=141:RETURN
610 IF A$="H" THEN A=133:RETURN
620 IF A$="C1" THEN A=125:RETURN
630 IF A$="C1#" THEN A=118:RETURN
640 IF A$="D1" THEN A=112:RETURN
650 IF A$="D1#" THEN A=105:RETURN
660 IF A$="E1" THEN A=99:RETURN
670 IF A$="F1" THEN A=94:RETURN
680 IF A$="F1#" THEN A=89:RETURN
690 IF A$="G1" THEN A=84:RETURN
700 IF A$="G1#" THEN A=79:RETURN
710 IF A$="A1" THEN A=74:RETURN
720 IF A$="B1" THEN A=70:RETURN
730 IF A$="H1" THEN A=66:RETURN
740 IF A$="C2" THEN A=63:RETURN
750 IF A$="P" THEN A=0:RETURN
760 PRINT A$," FEHLER !":END
900 DATA C,40,P,4,C,20,P,4,F,80,P,20,C,40,P,4,F,20,P,4,A,80,P,20
910 DATA C,40,P,4,C,20,P,4,A,60,P,8,C,40,P,4,F,20,P,4,A,60,P,80
920 DATA C,40,P,4,F,20,P,4,A,100,P,20,F,40,P,4,A,20,P,4,C1,100
930 DATA P,4,A,80,P,4,F,60,P,4,C,100,P,4,C,80,P,4,C,40,P,8
940 DATA F,100,XX,0
READY.

```

SARGON 2

Jetzt haben auch Sie die Möglichkeit, eine Partie Schach gegen den Vierzeweltmeister zu spielen. Mit S a r g o n 2 können Sie in verschiedenen Spielstärken spielen. Jede Spielstärke variiert in der Rechenzeit und der Suchtiefe.

Einige Besonderheiten von "Sargon 2" wollen wir Ihnen an dieser Stelle nicht vorenthalten.

- 7 Spielstufen
- div. Eröffnungsvarianten
- favorisierte Spielzüge werden angezeigt
- 2 Schachuhren
- Tonausgabe über USER-PORT möglich
- auf allen CBM 3000 oder 2000 mit neuem Betriebssystem

"Sargon 2" ist beim SCC unter der Best. Nr. P 4108 zu Fr. 78.-- erhältlich (Clubmitglieder erhalten 10 % Rabatt)



Maschinensprache

SPEICHERBELEGUNG

Leider können wir die Serie der Assemblerprogrammierung erst in der nächsten Ausgabe fortsetzen. Wir haben zwei neue Programme in Angriff genommen, die aber mehr Zeit in Anspruch nehmen, als wir gedacht hatten. Es handelt sich um ein wirklich gutes Programm zur Erzeugung komplexer Töne und ein weiteres zum Einlesen von Strichcode mit einem Barcodeleser über den USER-PORT. Trotzdem wollen wir nicht völlig aus dem Thema fallen und bringen daher einen Beitrag über das Arbeiten mit dem Betriebssystem.

Wie bei vielen Computern, gibt es auch beim PET/CBM einige Tricks und Routinen, mit denen man das Betriebssystem überlisten und so gewisse Vorteile erringen kann.

Oft erstellt man ein grösseres Programm, mit dessen Hilfe Daten ermittelt werden und stellt dann fest, dass der Speicherplatz des Computers mit seinen 8 Kilobyte nicht ausreicht, um die Daten zu verarbeiten.

Da den meisten Hobbyprogrammierern dann das Floppy fehlt, um die Daten in ein Hilfsfile zu schreiben, den zweiten Teil des Programms zu laden und die Daten wieder aus dem File zu lesen, scheint diese Situation ausweglos, da alle Variablen, Felder und Strings, die vorher festgelegt worden sind, beim Laden des neuen Programms gelöscht werden.

Dies geschieht, obwohl der "LOAD" Befehl kein "CLR" enthält. Das heisst, die Daten werden im Speicher nicht gelöscht oder überschrieben, sondern der Rechner weiss nur nicht mehr, wo er sie abgelegt hat, es sei denn, das neue Programm belegt mehr Speicherplatz als das alte Programm.

Zuerst aber einmal einige grundsätzliche Bemerkungen über das Abspeichern von Variablen, Feldern und Strings.

Im Speicher werden alle Daten in dieser Reihenfolge abgelegt :

Ende Betriebssystem (RAM)
Pointer 40,41 (1b,hb)
Anfang Programmspeicher

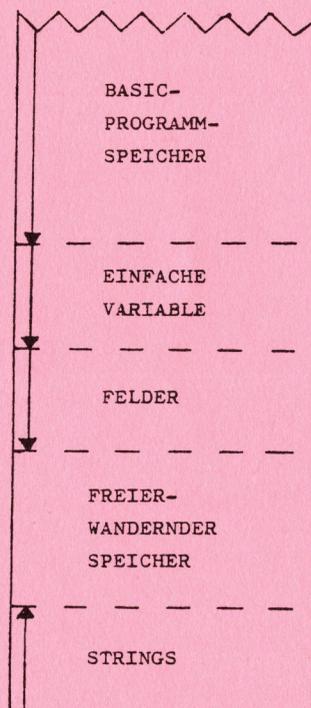
Ende Programmspeicher
Pointer 42,43 (1b,hb)
Anfang einfache Variable

Ende einfache Variable
Pointer 44,45 (1b,hb)
Anfang Felder

Ende Felder
Pointer 46,47 (1b,hb)
Anfang wandernder Speicher

Ende wandernder Speicher
Pointer 48,49 (1b,hb)
Ende Stings (rücklaufend)

Anfang Strings
Pointer 50,51 (1b,hb)
Ende RAM



Ab der Adresse 1025 beginnt das RAM (Arbeitsspeicher). An seinem Anfang wird das BASIC-Programm im Interpretercode notiert. Ein Zeiger (Pointer) weist auf das nächste freie Byte hinter dem Programm. Es ist in Byte 42 und 43 gespeichert und zwar in Low-Byte/High-Byte Form. Das bedeutet, setzt man die binäre Form, der in (42) gespeicherten Zahl hinter die binäre Form von (43), so ergibt sich eine Binärzahl mit 16 Stellen. Sie wird nach folgendem Schema umgerechnet:

B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
i	i	i	i	i	i	i	i	i	i	i	i	i	i	i	i
t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
Low - Byte								High-Byte							

ist zum Beispiel $PEEK(42)=3$ und $PEEK(43)=4$, so ergibt sich folgende Rechnung:

$$(4*256)+3=1027$$

$$00000100\ 00000011 = 1*1024+1*2+1*1 = 1027$$

In diesem Fall ist 1027 das erste freie Byte, was dem Einschaltzustand entspricht. Die ersten 3 Byte sind mit dem TIM Interrupt (Monitorauf-ruf) belegt. Auf das Ende der Variablen zeigt wieder ein Pointer, der gleichzeitig den Beginn der Felder markiert, er ist in Byte 44 und 45 nach dem gleichen Schema codiert.

Der Pointer auf das Ende der Felder steht in Byte 46 und 47. An das Ende der Felder schliesst sich ein sehr interessanter Speicherbereich an, der "wandernde" oder "freie" Speicher. In ihn können die BASIC-Programme, die Variablen und Felder absinken, wenn sich einer der drei Bereiche vergrössert.

Das heisst, wenn das BASIC-Programm erweitert wird, werden die Variablen und Felder vom Betriebssystem herabgesetzt, um Platz für die Programmanweisungen zu schaffen. Die Felder rücken soweit in den wandernden Speicher hinein, wie oben an Platz benötigt wird. Dabei sinken natürlich auch die Pointer, die ja zeigen, wo die Variablen, Felder etc. stehen, mit. Hier darf sinken aber nicht missverstanden werden.

Wenn sich die Pointer dem Ende des Speichers nähern, wird ihr Zahlenwert natürlich grösser und nicht kleiner. Auf das Ende des "wandernden" Speichers weist wiederum ein Pointer. Er steht in Byte 48 und 49.

Schliesslich werden vom Ende des Arbeitsspeichers an abwärts, die String notiert. Sie werden interessanterweise "rückwärts" geschrieben.

Das bedeutet, das erste Byte, des ersten Strings, ist das letzte des Speichers. So wurde erreicht, das auch die String in den Speicher hineinwachsen und das Betriebssystem entlastet wird. Wenn der (46)/(47)-Pointer gleich dem (48)/(49)-Pointer ist, erscheint die Fehlermeldung "OUT OF MEMORY", da dann der freie Speicher völlig belegt ist. Um dies zu verhindern oder hinauszuzögern, führt das System eine recht lange und zeitraubende Routine aus, die sogenannte "garbage-collection"-Routine (Müllsammlung).

Bei der Neuordnung eines String entstehen oft Leerstellen, gleich ungenutztem Speicher, wenn die neu zugeordneten Zeichenketten kürzer sind als die alten. Diese Leerstellen werden beseitigt, indem die anderen String aufrücken und die Pointer richtig gestellt werden.

Auch bei $PRINTFRE(0)$ wird diese Routine aktiviert.

Nun aber zurück zum Hauptproblem. Inzwischen ist bekannt, das die Daten nicht verlorengehen, sondern lediglich die Pointer verstellt werden,



weil das neue Programm kürzer als das alte ist. Ist das neue jedoch länger, so werden zumindest die einfachen Variablen und die Felder überschrieben. Im ersten Fall muss man also lediglich die Pointerstellung des alten Programms im Kassettenbuffer abspeichern und nach dem Einlesen des neuen, kürzeren Programms wiederherstellen. Dazu sind lediglich die beiden folgenden Zeilen hinter das alte Programm zu schreiben.

```
100 POKE826,PEEK(42):POKE827,PEEK(43):POKE828,PEEK(44):POKE829,PEEK(45)
110 POKE830,PEEK(46):POKE831,PEEK(47):POKE832,PEEK(48):POKE833,PEEK(49)
```

READY.

Und die beiden nächsten Zeilen vor das neue Programm.

```
200 POKE42,PEEK(826):POKE43,PEEK(827):POKE44,PEEK(828):POKE45,PEEK(829)
210 POKE46,PEEK(830):POKE47,PEEK(831):POKE48,PEEK(832):POKE49,PEEK(833)
READY.
```

Abschiessend noch einmal alle Pointer für alle PET/CBM-Systeme auf einen Blick.

Bezeichnung	BASIC 1.0	BASIC 2.0
Ende Betriebssystem/Anfang (BASIC)	122/123	40/41
Ende BASIC/Anfang Variable	124/125	42/43
Ende Variable/Anfang Felder	126/127	44/45
Ende Felder/Anfang freier Speicher	128/129	46/47
Ende freier Speicher/Ende Strings	130/131	48/49
Anfang Strings/Ende RAM	-----	50/51
Ende Speicher gesamt	132/133	52/53

LEICHTE KOST KALENDERBLATT

Uns wurde ein Programm zur Berechnung eines Kalenderblattes zugesandt. Mit diesem Programm ist es möglich festzustellen, auf welches Datum ein beliebiger Wochentag gefallen ist oder fallen wird. Das Programm soll vor allem zeigen, wie ein solches Problem angegangen werden muss.

Die heute gültige Zeitrechnung basiert auf dem Gregorianischen Kalender, der am 15. Oktober 1582 eingeführt worden ist.



KALENDERBLATT

```

100 REM *** KALENDERBLATT #1 ***
110 REM ----VERSION CBM 3001----
120 :
130 REM (C) 1981 HANS KAST
140 REM MUEHLACKERSTR. 43
150 REM 9436 BALGACH
160 REM -----
170 :
180 DIM FE(42)
190 PRINT"J": REM SCREEN CLR
200 X$="-----"
210 PRINTTAB(5)"  /"X$"\ "
220 PRINTTAB(5)"  I GREGORIANISCHER KALENDER I "
230 PRINTTAB(5)"  \"X$"/":PRINT
240 PRINT"PROGRAMMIERT 6/1981 VON H.KAST BALGACH"
250 PRINT"*****":REM 4 TIEF
260 PRINT"DIESES PROGRAMM ZEIGT IHNEN FUER JEDEN":PRINT
270 PRINT"BELIEBIGEN MONAT AB DEM JAHR 1582 EIN":PRINT
280 PRINT" VOLLSTAENDIGES KALENDERBLATT."
290 PRINT:PRINT:GOSUB860
300 PRINT"J":REM CLR
310 PRINT"EINGABE ":"PRINT"-----"
320 PRINT:POKE14,1:REM INPUT STEIGT NICHT AUS
330 INPUT"MONAT ? (ZAHL) ";E$
340 PRINT:M=INT(VAL(E$))
350 IFM<1ORM>12THEN330
360 PRINT:INPUT"JAHR ? (4-STELLIG) ";E$
370 PRINT:Y=INT(VAL(E$))
380 IFY>=1582ANDY<3333THEN400
390 PRINT:PRINT"ZWISCHEN 1582 UND 3333 !":GOTO360
400 POKE14,0:T=1:REM 1.TAG IM MONAT
410 RESTORE
420 REM ERMITTELN DES MONATSNAMEN (M$) UND ANZAHL TAGE (TM):
430 FORI=1TOM:READM$,TM:NEXT
440 REM --- SCHALTJAHR-TEST ---
450 IFM<>2THEN510
460 Y1=INT(Y/100):Y2=Y-Y1*100:IFY2=0THEN500
470 IFY2/4=INT(Y2/4)THENTM=29:REM JAHR DURCH 4 TEILBAR
480 GOTO510
490 REM 'HUNDERTER' JAHRE:
500 IFY1/4=INT(Y1/4)THENTM=29:REM HUNDERTER DURCH 4 TEILBAR
510 REM----BERECHNUNG DES FAKTORS----
520 REM (ANZAHL TAGE N. CHR.)
530 X=365*Y+T+31*(M-1)
540 IFM<=2THEN580
550 REM MAERZ BIS DEZEMBER
560 F=X-INT(0.4*M+2.3)+INT(Y/4)-INT(3/4*(INT(Y/100)+1))
570 GOTO600
580 REM JANUAR UND FEBRUAR
590 F=X+INT((Y-1)/4)-INT(3/4*(INT((Y-1)/100)+1))
600 REM----ZIFFER FUER WOCHENTAG----
610 REM (0=SAMSTAG, 1=SONNTAG, 2=MONTAG...)
620 W=F-INT(F/7)*7:REM REST VON GANZZAHLIGER DIVISION
630 IFW=0THENW=7
640 REM----KALENDERTABELLE FUELLEN----
650 REM ----( DATENFELD FE(I) )----
660 REM 1.TABELLENZEILE LOESCHEN:
670 FORI=0TOW-1:FE(I)=0:NEXT
680 REM GUELTIGE TAGE IM FELD AUFFUELLEN:
690 FORI=0TOTM-1:FE(I+W)=I+1:NEXT
700 REM RESTLICHES FELD LOESCHEN:
710 FORI=TM+WT042:FE(I)=0:NEXT

```

```

720 REM---- AUSGABE AUF BILDSCHIRM ----
730 PRINT"J":PRINTTAB(12);M$;Y:PRINT
740 PRINTTAB(5)"_____ "
750 PRINTTAB(5)"SO MO DI MI DO FR SA":PRINT
760 FORIW=0TO5:FORIT=1TO7:I=IW*7+IT
770 REM IW=INDEX ZEILE (WOCHE), IT=INDEX SPALTE (TAGE)
780 IFFE(I)=0THEN820:REM BEI LEER KEIN PRINT
790 REM POSITION FUER TABULATOR, 4 ZEICHEN PRO TAG :
800 P=4*IT:IFFE(I)<10THENP=P+1:REM EINSTELLIG
810 PRINTTAB(P);FE(I);
820 NEXTIT:PRINT:PRINT:NEXTIW:PRINT:PRINT
830 GOSUB860:GOTO190
840 :
850 :
860 POKE158,0:REM TASTATURPUFFER LEEREN
870 PRINTTAB(24)"_____ "
880 PRINTTAB(8)"FORTSETZUNG MIT SPACE"
890 GETE$:IFE$<>" "THEN890
900 RETURN
910 :
920 DATA JANUAR,31,FEBRUAR,28,MAERZ,31,APRIL,30
930 DATA MAI,31,JUNI,30,JULI,31,AUGUST,31
940 DATA SEPTEMBER,30,OKTOBER,31,NOVEMBER,30,DEZEMBER,31
READY.

```

Zeile 100 - 170 Programmkopf
 Zeile 180 Dimensionierung der Datenfelder FE(I) der Tageszahlen
 Zeile 190 - 290 Titel und Erklärungen auf dem Bildschirm
 Zeile 300 - 390 Eingabe des gewünschten Monats und Jahres incl. Gültigkeitstest . POKE 14,1 für CBM 3000, POKE 3,1 für PET 2001 und POKE 16,1 für CBM 8000
 Zeile 400 Wiederherstellung des Normalzustandes, siehe Zeile 300
 Zeile 410 - 430 Monatsnamen und Tageszahlen aus DATA-Statements holen
 Zeile 440 - 500 Schaltjahrtest: Schaltjahre sind durch 4 teilbar, Ausnahmen sind Jahreszahlen mit ganzen Hundertern, die nur dann Schaltjahre sind, wenn die ersten beiden Ziffern durch 4 teilbar sind
 Zeile 510 - 590 Anzahl Tage ab 1.1.0000 bis zum 1. Tag des gewünschten Monats
 Zeile 600 - 630 Ziffern des Wochentags bestimmen
 Zeile 640 - 710 Datenfeld FE(I) mit Tageszahlen für den Monat füllen 6 Wochen zu je 7 Tagen = 42 Daten.
 Zeile 720 - 820 Ausgabe der Tabelle, wieder 6 Wochen zu 7 Tagen.
 Zeile 830 Pause und Rückkehr zum Programmbeginn.
 Zeile 860 - 900 Subroutine PAUSE. Poke 158,0 bewirkt, dass während des Programmablaufs gedrückte Tasten keine Wirkung zeigen. POKE 525,0 für PET 2001 und POKE 158,0 für CBM 8000.
 Zeile 920 - 940 DATA - Statements

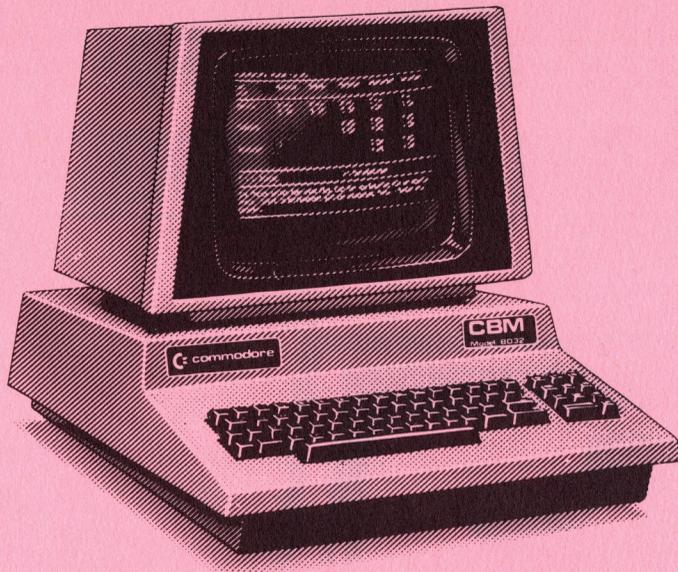
Variablentabelle

```

-----
FE( ) Tabellenfeld
M, M$ Monat Nummer und Namen
Y Jahreszahl
T, TM Tag(=1), Anzahl Tage im Monat von DATA
W Ziffer zur Wochentagsbestimmung
I Index, Zähler
IW, IT Index für Tabelle Wochen, Tage
P Tabellenposition (TAB)
X Zwischenresultat
F Faktor (Anzahl Tage n. Chr.)

```





commodore COMPUTER

P 2007	8 K	PET mit kleiner Tastatur inkl. Rekorder	1695.—
P 2118	16 K	CBM 4016 (exkl. Rekorder)	2450.—
P 2134	32 K	CBM 4032 (exkl. Rekorder)	2975.—
P 2037	32 K	CBM 8032, 80 Zeichen (exkl. Rekorder)	3975.—
P 2038		CBM 8050 Dual Floppy 1 Megabyte	4475.—
P 2140		CBM Dual Floppy 4040	3475.—
P 2025		CBM 8024 9 x 7 Matrixdrucker	4350.—
P 2026		CBM 8026 Typenrad-Drucker mit Tastatur	3950.—
P 2027		CBM 8026 Typenrad-Drucker ohne Tastatur	3250.—
P 2042		Grüner Filter zu PET-Bildschirm	24.95
P 2090		CBM Rekorder	195.—
P 2101		Kabel PET-IEEE	100.—
P 2102		Kabel IEEE-IEEE	115.—
P 9111		Musikzusatz mit Software	98.—
P 9112		grosser Musikzusatz mit D/A Wandler + Software	148.—
P 9113		RTTY Interface mit Software	240.—
P 9900		CBM Basic Umrüstsatz Basic 1 auf Basic 2	240.—
P 9901		BASIC-Toolkit ROM (TK 80 P) 8 K PET alt	185.—
P 9902		BASIC-Toolkit ROM (TK 160) neuer PET	125.—
P 9907		BASIC-Toolkit ROM (TK 8000) für 8000 Serie	206.—
P 9903		Space Maker (Platzmacher) WPP/Toolkit	69.—
P 9951		Spacemaker Compu/Toolkit (für P 9903 und P 9951 ist die gleiche Platine verwendbar!)	69.—
P 9950		Hochauflösende Grafik mit Einbau (HRG)	1280.—
P 9985		UTILITY EPROM	158.—
P 9986		HRG Soft (Eprom)	194.—
P 3512		CBM Newtim SV20 Monitor Assembler Editor Bei Bestellung unbedingt CBM-Type angeben!	400.—
P 9988		Monjana ROM zu PET	198.—
P 3501		Wordprocessor II Programm mit ROM	280.—
P 3503		Vorauslieferung deutsche Anleitung 15 Seiten	20.—
P 3502		Wordprocessor III	640.—
P 3511		Assembler für 3001	320.—
P 3531		CBM Pascal für 3001	350.—
P 3521		Assembler II für 8001	480.—
P 3532		Visicalc deutsch	580.—
P 3507		OZZ, Information Management System diverse Interface vorrätig	1300.—

CBM/PET

Systemsoftware

Assembler Pac	P 3300 K	60.—
Hardcopy	P 3313 K	18.—
Teletype	P 3315 K	18.—
Jana Monitor	P 3317 K	29.—
Fettdruck	P 3325 K	18.—
Programm in Maschinensprache	P 3375 K	39.80
Ein/Ausgabe-Programm mit PET	P 3377 K	49.80
EDASS 65	P 3379 K	80.—
CAD	P 3381 K	80.—
CBM ASS 65	P 3383 D	94.—

Programme des Monats aus CBM/PET News

Benzinverbrauchsrechner	P 3701	18.—
Software Uhr	P 3703	18.—
Software Schaltuhr	P 3705	18.—
Computer-Musik	P 3707	18.—
Kalorienplaner	P 3709	18.—
Graphic	P 3711	18.—
Skirennen	P 3713	18.—
Mühle	P 3715	18.—
Wettervorhersage	P 3717	18.—
Hornussen	P 3719	18.—

Small Business

Zinseszins	P 6111	18.—
Artikelverwaltung	P 6113	18.—
Adressbuch	P 6115	18.—

Commodore Operating System

Disk	P 6301	300.—
Finanzbuchhaltung 1, Disk	P 6303	2000.—
Finanzbuchhaltung 2, Disk	P 6305	4000.—
Textverarbeitung, Disk	P 6307	2500.—
Lagerkontrolle, Disk	P 6309	700.—
Fakturierung 4, Disk	P 6317	1700.—
Adresskartei, Disk	P 6319	500.—
Sort, Disk	P 6321	500.—
Listen-Generator, Disk	P 6323	700.—
Lagerbuchhaltung (2. Rec.)	P 6401	18.—

CBM-4000-Serie

Slot A	P 4800	18.—
Slot B	P 4801	18.—
Cover 18	P 4802	18.—
Osterdatum	P 4803	18.—
G Book A + B (2 Programme)	P 4804	28.—
Cover 19	P 4805	18.—
Godzilla	P 4806	18.—
Bets	P 4807	18.—
Joust	P 4808	18.—
Miner	P 4809	M 24.—
Frog	P 4810	M 24.—
Weather	P 4811	18.—
Rail	P 4812	18.—
Musik	P 4813	M 24.—
Lagerbuchhaltung	P 4814	18.—
Off-the-Wall	P 4815	18.—
Morsen	P 4816	18.—
Merke	P 4817	18.—
Piranha	P 4818	18.—
Primzahlen	P 4819	18.—
Reversi	P 4820	18.—
Rhino	P 4821	18.—
Plotroutine	P 4822	18.—
Mondlandung	P 4823	18.—
Reaktionstest	P 4824	18.—
QTH-Kenner	P 4825	18.—
Dromeda/Ufolandung	P 4826	M 24.—
Seekrieg	P 4827	18.—
Sea War	P 4828	18.—
Sheep	P 4829	18.—
Cover 20	P 4830	18.—
Mastermind	P 4831	18.—
Fussball	P 4832	18.—
Mayday	P 4833	18.—
Empire	P 4834	18.—
Encluing	P 4835	18.—
Magisches Quadrat	P 4836	18.—
Kurven-Plotter	P 4837	18.—
El Presidente	P 4838	18.—
Gleichungssysteme	P 4839	18.—
Vocabular 2	P 4840	18.—
BASIC-Kurs (2 Kassetten)	P 4841	28.—
Fahrtst	P 4842	18.—

M benötigen Musikzusatz zu PET,
siehe P 9111/P 9112

In der Programmbibliothek des SCC-Shop sind weit über 100 Programme für CBM/PET vorhanden. Verlangen Sie bitte unverbindlich den Spezialprospekt.



Korrespondenz und
Manuskripte bitte an

SCC PET NEWS
Seeburgstrasse 12
6006 Luzern

Die Beiträge stammen grösstenteils von Clubmitgliedern oder sind gekürzte Übersetzungen. Für die Veröffentlichung wird keine Gewähr oder Garantie übernommen, auch nicht dafür, dass die verwendeten Schaltungen, Firmennamen und Warenbezeichnungen frei von Schutzrechten Dritter sind. Die Verwendung der Informationen erfolgt auf eigenes Risiko.

Copyright by SCC Lucerne, aber Speicherung in Datenverarbeitungsanlagen für den eigenen Gebrauch erlaubt.

Commodore: Wegbereiter des Jedermann-Computer



C commodore
Commodore AG · Dufourstraße 9 · 4010 Basel · Tel. 061/23 78 00 · Telex 64 961

Autorisierte Commodore-Wiederverkäufer mit technischem Kundendienst

Aarau Dahms Computersysteme · Tel. (064) 22 77 66. **Basel** BD-Electronic · Tel. (061) 35 36 37. Geiger-Microcomputer · Tel. (061) 44 13 13. Leobag Computer AG · Tel. (061) 35 31 14. SYSAG Systems & Services AG · Tel. (061) 38 21 20. Thürlemann Discount · Tel. (061) 22 41 66. **Bern** Computerland AG · Tel. (031) 24 25 54. Radio TV Steiner AG · Tel. (031) 55 45 81. **Biel** EIM Computer AG · Tel. (032) 23 15 88. **Brugg** Megos AG · Tel. (056) 41 34 17. **Fontainemelon** Urs Meyer Electronic · Tel. (038) 53 43 43. **Freiburg** Sovitrel SA · Tel. (037) 22 78 37. **Genf** Centre Informatique Gesmarco · Tel. (022) 21 11 75. Egg-Telsa SA ·

Tel. (022) 20 06 00. Gesmarco SA (Thônex) · Tel. (022) 49 88 44. Irco Electronic · Tel. (022) 20 33 06. Radio TV Steiner AG · Tel. (022) 28 52 22. **Gossau** Pius Schäfler · Tel. (071) 85 13 87. **Huttwil** Compu-Life, Rüfenacht AG · Tel. (063) 72 11 13. **Interlaken** Datatechnik · Tel. (036) 22 10 21. **Lausanne** Mafioly SA · Tel. (021) 22 00 44. Schaefer Informatique · Tel. (021) 23 55 55. **Lugano** Luigi Chiodoni · Tel. (091) 23 23 09. **Luzern** Dialog Computer Treuhand AG · Tel. (041) 31 45 45. Helfenstein + Bucher AG · Tel. (041) 22 13 43. Schweizer Computer Club · Tel. (041) 31 45 45. **Magliaso** Marah SA · Tel. (091) 71 14 28. **Mellingen** Instant-Soft AG · Tel. (056) 91 20 21. **Niederrohrdorf** Nöthiger Electronic · Tel. (056)

96 28 96. **Rüti/ZH** Logon AG · Tel. (055) 31 72 30. **Schaffhausen** PIM Systems · Tel. (053) 45 45 0. Syntron Electronic · Tel. (053) 5 33 77. **Sion** Sphère Corporation · Tel. (027) 22 68 14. **St.Gallen** LASYS · Tel. (071) 28 39 05. **Thun** HMB electronic · Tel. (033) 22 66 88. **Wettingen** Elbatex AG · Tel. (056) 26 98 27. **Winterthur** Nowak AG · Tel. (052) 22 08 03. **Wohlen/AG** SYSAG Systems & Services AG · Tel. (057) 6 36 50. **Zürich** FÜRrer Büro-Computer · Tel. (01) 202 49 92. Hannes Keller AG · Tel. (01) 69 36 33. Logon AG · Tel. (01) 62 59 22. Microspot AG · Tel. (01) 241 20 30. Erhard Wipf AG · Tel. (01) 221 21 00.