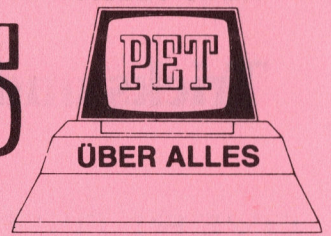


## CBM/PET NEWS



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

Lieber Pet - Freund

Nun liegt bereits das letzte Heft des 2. Jahrgangs vor Ihnen und wir glauben, dass wir die im letzten Jahr gemachten Versprechungen, so weit irgend möglich, eingehalten haben. Für die manchmal kleinen Terminverschiebungen bitten wir Sie um Verständnis. Es ist einfach nicht immer möglich, ein Programm oder einen Artikel immer genau auf das angekündigte Erscheinungsdatum fertig zu haben, da wir zum grössten Teil auf nebenberufliche Kräfte angewiesen sind.

Trotzdem haben wir uns auch für das Jahr 1982 Termine gesetzt, die wir nach besten Kräften einhalten wollen. Unser Programm für 1982 stellen wir Ihnen in seinen Grundzügen auf Seite 19 vor.

Wir möchten Sie in der nächsten Ausgabe mit einer neuen, zusätzlichen Rubrik bekannt machen, in der wir berufsspezifische Programme bringen werden, die Probleme nur einer Berufsgattung zum Inhalt haben und die darüberhinaus auch im Unterricht und in der Lehrlingsausbildung angewendet werden können. Die Reihe von Programmen soll zunächst in zwangloser Reihenfolge über physikalische, chemische und elektronische Probleme berichten. Je nach der Resonanz aus dem Leserkreis können dann auch andere Programme besprochen werden. Wir wissen, dass wir hiermit immer nur einen kleinen Leserkreis ansprechen, aber wir wollen ja für alle etwas Interessantes bringen.

Auch unter dem Titel "Maschinensprache und Assemblerprogrammierung" sollen wiederum einige aktuelle Programme und Beiträge zum Verständnis dieser etwas komplizierten Materie beitragen, aber auch dem Beginner sollen im Programm des Monats neue Ideen gegeben werden.

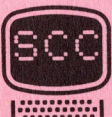
Nachdem wir in "Tricks und Tips" bereits mit der Veröffentlichung von kaufmännischen Programmen begonnen haben, die wir im nächsten Jahr natürlich fortsetzen werden, glauben wir, mit dem Inhalt der NEWS den meisten Wünschen gerecht zu werden.

Mit dieser Ausgabe ist nun auch Ihr Jahresabonnement abgelaufen. Wir würden uns natürlich riesig freuen, wenn wir Sie auch im dritten Jahrgang zu unseren treuen Lesern zählen dürften. Das Abonnement für weitere sechs Nummern beträgt 1982 Fr. 24.--/DM 29.-- für SCC-Mitglieder und Abonnenten von MIKRO- und KLEINCOMPUTER und Fr. 48.--/DM 55.-- für alle übrigen Bezüger. Bitte beachten Sie, die CBM PET NEWS sind nur im Jahresabonnement erhältlich.

Wir wünschen Ihnen ein frohes Weihnachtsfest, ein gutes Neues Jahr und natürlich weiterhin viel Freude und Erfolg mit Ihrem Commodore.

Die Redaktion

Heinz Kastien





# Tricks und Tips

## OVERLAYTECHNIKEN FLOPPY STATUS

Im Heft 81-5 wurde die Menutechnik zusammen mit der Overlaytechnik beschrieben, die es gestattet, durch ein laufendes Programm ab Floppy weitere Programme zu laden.

Speziell unser Menu Adressverwaltung wurde von Herrn A. Simoes mit berechtigter Kritik bedacht, die wir fairerweise unseren Lesern nicht vorenthalten wollen, daher drucken wir hier die geänderte Version ab.

```
1000 REM MENU
1010 REM BY A. SIMOES
1020 CLR
1030 SP=11
1040 POKE59468,14
1050 PRINT"┌"SPC(SP)"■■■■■■■■■■ \MENU ***"
1060 PRINTSPC(SP)"■■■■1. ■ ADRESSEINGABE"
1070 PRINTSPC(SP)"■■■■2. ■ ADRESSMUTATION"
1080 PRINTSPC(SP)"■■■■3. ■ ADRESSABRUF"
1090 PRINTSPC(SP)"■■■■4. ■ ADRESSAUFKLEBER"
1100 PRINTSPC(SP)"■■■■5. ■ ADRESSLISTE"
1110 POKE158,0
1120 WAIT158,1:GETZ$
1130 IFVAL(Z$)=0ORVAL(Z$)>5THEN1110
1140 ONVAL(Z$)GOTO1150,1160,1170,1180,1190
1150 P$="EINGABE":GOTO1200
1160 P$="MUTATION":GOTO1200
1170 P$="AUSGABE":GOTO1200
1180 P$="KLEBER":GOTO1200
1190 P$="LISTE"
1200 POKE59468,12:PRINT"└■■■■LOAD"CHR$(34);"0:"P$CHR$(34);",8"
1210 PRINT"■■■■RUN■■■■":POKE158,2:POKE623,13:POKE624,13:END
```

Der Unterschied zwischen dem hier vorgestellten Programm und dem der letzten Ausgabe liegt vor allem im GET der Zeilen 1110-1140.

Das Menu wird nicht mit GET Z sondern mit GET Z\$ abgerufen, dadurch wird erreicht, dass der Rechner nicht abbricht, wenn aus Versehen eine andere Taste als eine Zahl gedrückt wird. Gleichwohl werden nur Zahlen als Eingabe berücksichtigt (Zeile 1130). Weiterhin wird die sonst übliche Warteschleife (if z\$=""then...) hier durch WAIT ersetzt. Es wird solange gewartet, bis eine beliebige Taste gedrückt worden ist, da 158 die Adresse der Zeichenzahl im Tastaturbuffer ist.

## OVERLAYTECHNIKEN

In dieser Fortsetzung sollen einige weitere verfeinerte Overlaytechniken beschrieben, sowie Fehlermeldungen des Floppy besprochen werden.



```

1000 PRINT"LOAD";CHR$(34);"0:";PR$;CHR$(34);",8"
1010 FORV=0TO5:READX:POKE623+V,X:NEXT
1020 POKE158,6
1030 DATA19,13,82,85,78,13
READY.

```

Das Programm ist identisch mit den Zeilen 1200 und 1210 auf Seite 2. Allerdings werden hier das RUN und die beiden RETURN nicht über den Tastaturbuffer und den Bildschirmspeicher abgerufen, sondern in den Kassettenbuffer gepokt.

```

100 REM PROGRAMMSTART
110 REM BY J.KRUEGER
120 FORI=964TO992:READJ:POKEI,J:NEXT
130 LOAD"0:MENU",8
140 DATA165,201,133,42,133,44,133,46,165,202,133,43,133,45
150 DATA133,47,165,52,133,48,133,50,165,53,133,49,133,51,96

```

```

100 SYS964:PRINT"01 02 ENDE "
110 GETX$
120 IFX$="1"THEN PRINTX$:LOAD"0:PROGRAMM 1",8
130 IFX$="2"THEN PRINTX$:LOAD"0:PROGRAMM 2",8
150 IFX$="E"THEN PRINT"ENDE":END
160 GOTO110
READY.

```

Hier muss als erstes Programm immer der Programmteil Programmstart laufen. Danach muss vor jedem Programm mit SYS 964 dieser Programmteil aufgerufen werden. Mit diesem kleinen Programm werden die Pointer gesetzt, wie dies im Heft 5/81 unter der Rubrik "Maschinensprache" besprochen worden ist, es kann somit jedes Programm normal mit LOAD geladen werden.

## FEHLERMELDUNGEN & STATUSREGISTER

Beim Arbeiten mit der Floppy-Station können Fehler entstehen, die einerseits auf ein falsches Handling andererseits aber auch auf Fehler des Floppy zurückzuführen sind.

Es ist von grösster Wichtigkeit, diese Fehler zu erkennen und auf möglichst einfache Art zu beheben. Hier kommt uns die Intelligenz des Commodore Floppy zugute, durch die bereits eine komplette Fehleranalyse möglich ist. Sobald das Floppy durch Aufleuchten der roten Lampe zwischen den beiden Drive einen Fehler anzeigt, kann dieser Fehler durch ein kleines Hilfsprogramm in den Rechner eingelesen und eventuell durch geeignete Massnahmen behoben werden.

Diese Fehlererkennung wird zweckmässigerweise in eine Subroutine gekleidet, die von jeder Stelle des Programms aufgerufen werden kann. Es wäre allerdings des guten zuviel, z. B. nur beim Abruf eines Programms ab Floppy diese Routine zu benutzen, da hierbei meist keine Fehler auftreten und vor allem durch diese Routine das Programm verlangsamt wird.



Die möglichen Fehler beim Floppy sollen hier nicht aufgeführt werden, da diese in jedem Floppymanual nachzulesen sind.

```
1000 INPUT#15,EN$,EM$,ET$,ES$
1010 IFVAL(EN$)=0THENRETURN
1020 PRINTEM$,EN$,ET$,ES$
1030 CLOSE15:END
READY.
```

Durch diese Zeilen werden 4 Stringvariable eingelesen, die folgende Bedeutung haben:

1. EN\$ = Fehlercode
2. EM\$ = Name des Fehlers
3. ET\$ = Nummer des fehlerhaften Track
4. EB\$ = Nummer des Fehlerhaften Block

Durch Zeile 1010 wird erreicht, dass bei Abwesenheit eines Fehlers kein Ausdruck erfolgt, und das Programm unverändert weiterläuft.

Die Fehlerroutine kann natürlich auch dazu dienen, den erkannten Fehler zu beheben und das Programm fortzusetzen. Wird z. B. einmal vergessen, die Diskette zu initialisieren, so wird durch

```
1000 INPUT#15,EN$,EM$,ET$,ES$
1010 IFVAL(EN$)=0THENRETURN
1020 IFVAL(EN$)=29THENPRINT#15,"I0":RETURN
1030 PRINTEM$,EN$,ET$,ES$
1040 CLOSE15:RETURN
```

erreicht, dass die Initialisierung nachgeholt und das Programm erst dann fortgesetzt wird.

Während bei Floppy-Stationen der Serie 2040 und 3040 die oben erwähnte Routine angewendet werden muss, ist beim Floppy 8050 und 4040 die Erkennung eines Fehlers durch Aufruf der Fehlervariablen DS\$ oder DS möglich, ohne dass hierzu ein eigenes Unterprogramm geschrieben werden müsste. Um das Gleiche wie oben zu erreichen, sind nur die nachstehenden Zeilen im Programm erforderlich.

```
1000 IF DS = 0 THEN RETURN
1010 PRINT DS$
```

Eine ebenso wichtige Rolle wie die Fehlererkennung bildet die Statusvariable beim Arbeiten mit dem Floppy. Die Statusvariable wird bei allen INPUT/OUTPUT-Operationen gesetzt und durch sie ist es möglich, den Verkehr mit externen Geräten zu überwachen. Die Statusvariable ST hat folgende Bedeutung.

- 1 Zeitüberschreitung beim Lesen
- 2 Zeitüberschreitung beim Schreiben
- 4 zu kurzer Block
- 8 zu langer Block
- 16 nicht korrigierbarer Lesefehler
- 32 Prüfsummenfehler
- 64 Dateiende (End of File)
- 128 DEVICE NOT PRESENT (End of Tape)



#### ZEITUEBERSCHREITUNG (1 und 2)

Die Statusvariable wird dann gesetzt, wenn sehr langsame Peripheriegeräte am IEC-Bus angeschlossen sind, damit sich der Rechner an einem solchen Teilnehmer nicht aufhängt, bricht das Betriebssystem nach 65 ms ab. Der Abbruch wird dann mit ST 1 (PRINT) oder ST 2 (INPUT/GET) vermerkt. Diese Operation ist z.B. beim Einlesen von Daten eines Digitalinstrumentes mit IEC-BUS von grosser Bedeutung.

#### KURZER BLOCK (4)

Signalisiert einen unkorrekten Zwischenraum innerhalb eines FILE Records. Dies kann passieren, wenn man ein BASIC-Programm als DATA FILE einzulesen versucht.

#### LANGER BLOCK (8)

Zeigt zu lange Zwischenräume innerhalb eines DATEN Records an. Es kann die gleiche Ursache wie beim zu kurzen Block vorliegen.

#### NICHT KORRIGIERBARE LESEFEHLER (16)

Der Rechner wird gezwungen, das Lesen eines FILE zu verhindern und statt dessen die Statusvariable auf 16 zu setzen, dies erfolgt, wenn beim Laden eines Programms mehr als 31 Fehler in einer Datei oder einem Programm gefunden worden sind.

#### PRUEFSUMMENFEHLER (32)

Der Rechner liest alle Daten eines Bandes oder anderer INPUT-Geräte zweimal, falls die Summe der empfangenen Byte nicht miteinander identisch sind, wird das Statusregister auf 32 gesetzt. Wenn Sie das nächste mal einen VERIFY-ERROR bekommen, kontrollieren Sie ST, Sie werden dies dann besser verstehen.

Die Fehlermeldungen der Statusvariablen 4 - 32 können auch bei verschmutzten Leseköpfen des Bandgerätes oder Magnetisierungsfehlern des Kopfes entstehen. Versuchen Sie daher den Schreib/Lesekopf zu reinigen und zu entmagnetisieren. Ebenso kann eine Statusvariable angezeigt werden, die oben nicht vermerkt ist, dies tritt dann ein, wenn mehrere Fehler gleichzeitig vorliegen z.B.

ST = 56 zeigt den Status 32 + 16 + 8 an

#### DATEIENDE (64)

Es wird angezeigt, dass die Datei zu Ende ist. Dies ist vor allem dann wichtig, wenn nicht bekannt ist, wieviele Daten mit INPUT oder GET eingelesen werden können. Dieser Status kann bei Einlesen von Daten ab Floppy benutzt werden, wie das nachfolgende Beispiel zeigt:

```
100 OPEN1,8,3,"1:DATAFILE,S,R"
110 INPUT#1,A$(I)
120 RS=ST
130 PRINTA$(I)
140 IF RS AND 64 THEN 170
150 I=I+1
160 GOTO 110
170 CLOSE1
```



END OF TAPE (128)

Es wird dem Rechner mitgeteilt, dass keine Data File mehr vorhanden sind, und dass die FILE NOT FOUND ERROR generiert werden soll. Um eine solche Meldung an Tape Nr. 1 zu erhalten muss mit OPEN1,1,1 eröffnet werden.

Zur Abfrage des Statuswertes wird ST einer anderen Variablen zugewiesen:

RS = ST

Es wäre jedoch falsch, den Status mit

IF RS = 64 THEN ....

abzufragen, sondern vielmehr muss die Abfrage wie folgt geschehen

IF RS AND 64 THEN ....

Der Grund für diesen Befehl ist darin zu suchen, dass beim Setzen der Statusvariablen die Bits 0 - 7 gesetzt werden und ebenfalls wieder abgefragt werden, es wird also der binäre Zustand eines Registers ermittelt.

Beim Arbeiten mit dem Floppy ist nur die Statusvariable 64 von Bedeutung.

Nach dieser eingehenden Behandlung der Statusvariablen und der Floppy Fehlermeldungen soll unter Verwendung dieser Erkenntnisse in der nächsten Folge die Eingabe und das Abspeichern von Artikeln bei der Lagerverwaltung und die Abspeicherung von Adressen bei der Adressverwaltung demonstriert werden.

## LEICHTE KOST

## SURPRISE PROGRAMM

Dieses Programm soll in erster Linie eine kleine Weihnachtsüberraschung sein. Darüberhinaus soll aber auch gezeigt werden, wie es möglich ist, das gleiche Programm auf einen Rechner der 8000 und der 3000 Reihe unverändert laufen zu lassen.

Das Programm läuft in der vorliegenden Form auf der 8000 Serie, wird die Variable X = 0 und U = 40 gesetzt ist das gleiche Programm für die 3000 Serie geeignet. Um das Programm voll auskosten zu können sollte an den Ausgang CB 2 ein kleiner Verstärker oder Musikzusatz angeschlossen werden. Beim CBM 8000 ist dies nicht unbedingt erforderlich aber empfehlenswert.



## SURPRISE PROGRAMM

```

1000 REM SURPRISE PROGRAMM
1010 REM PROGRAMM FUER CBM 8000
1020 REM BEI CBM 3000 WIRD X=0 UND U=40 GESETZT
1030 X=20:A=9+X:U=80:XX=32768+X:POKE59468,12
1040 PRINT"#####TAB(A+11)"
1050 PRINT TAB(A+10)"^
1060 PRINT TAB(A+10)"^
1070 PRINT TAB(A+8)"###/^
1080 PRINT TAB(A+9)"^^
1090 PRINT TAB(A+7)"###/^
1100 PRINT TAB(A+9)"^^
1110 PRINT TAB(A+7)"///
1120 PRINT TAB(A+7)"///
1130 PRINT TAB(A+5)"###/^
1140 PRINT TAB(A+7)"///
1150 PRINT TAB(A+6)"///
1160 PRINT TAB(A+4)"###/^
1170 PRINT TAB(A+5)"///
1180 PRINT TAB(A+4)"///
1190 PRINT TAB(A+2)"###/^
1200 PRINT TAB(A+4)"///
1210 PRINT TAB(A+10)"##
1220 PRINT TAB(A+10)"##
1230 PRINT TAB(A)"
1240 I=42
1250 GOSUB 1460
1260 GOSUB 1350
1270 I=32
1280 GOSUB 1460
1290 FOR N=0 TO 1000:NEXT N
1300 PRINT"#####":PRINTTAB(2+A)"DER SCC WUENSCHT EINE"
1310 PRINT:PRINT:PRINT:PRINTTAB(5+A)"F R O E H L I C H E"
1320 PRINTTAB(5+A)" "
1330 PRINTTAB(5+A)" W E I H N A C H T "
1340 END
1350 POKE 59467,16:POKE 59466,104
1360 READ A1,B:IF A1=999 THEN POKE 59467,0:RETURN
1370 POKE 59464,A1:FOR I=1 TO 120*B:NEXT:GOTO 1360
1380 DATA 250,2,188,1,0,1,188,1,0,1,188,3,0,1,167,2,149,1
1390 DATA 0,1,149,1,0,1,149,3,0,1,149,1,167,1,149,1,141,2
1400 DATA 199,2,167,2,188,3,0,1,125,1,0,1,125,1,149,1
1410 DATA 112,3,0,1,125,1,0,1,125,1,141,1,0,1,141,3,0,1
1420 DATA 141,1,0,1,141,1,167,1,125,3,141,1,0,1,141,1
1430 DATA 149,1,0,1,149,3,0,1,250,2,188,1,0,1,188,1,0,1
1440 DATA 188,3,0,1,167,2,149,1,0,1,149,1,0,1,149,3,0,1
1450 DATA 149,1,167,1,149,1,141,2,199,2,167,2,188,4,999,0
1460 FOR N=0 TO 300:NEXT N
1470 POKE XX+2*U+20,I
1480 POKE XX+5*U+17,I
1490 POKE XX+5*U+22,I
1500 POKE XX+7*U+16,I
1510 POKE XX+8*U+24,I
1520 POKE XX+11*U+14,I
1530 POKE XX+11*U+21,I
1540 POKE XX+11*U+25,I
1550 POKE XX+12*U+19,I
1560 POKE XX+14*U+13,I
1570 POKE XX+15*U+25,I
1580 POKE XX+16*U+18,I
1590 POKE XX+17*U+11,I
1600 POKE XX+18*U+26,I
1610 RETURN

```

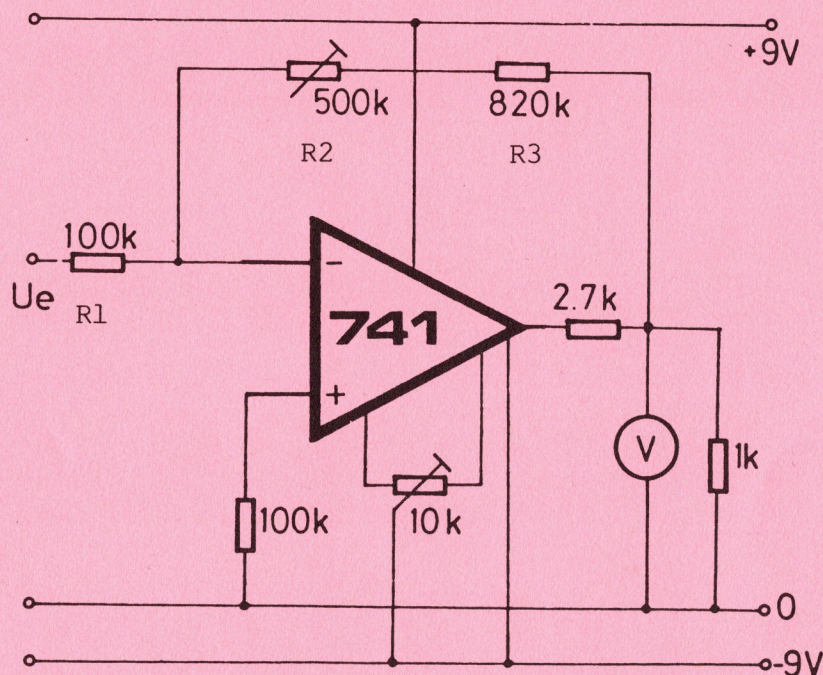


# Hardware

## TEMPERATURMESS ZUSATZ

Nachdem in der letzten Ausgabe ein Analog-Digital-Wandler beschrieben worden ist, können nun auch analoge Daten verarbeitet werden. Dies ist enorm wichtig, da nahezu alle Informationen, die wir von Messfühlern und Messwandler erhalten, analogen Charakter haben.

Der Temperaturmesszusatz ist nur eine der vielen Möglichkeiten, Umweltinformationen in den Rechner einzugeben und zu verarbeiten. Hierbei ist die eigentliche Schaltung, von kleinen Anpassungen abgesehen, immer die gleiche, lediglich der Messfühler oder Messwandler ist ein anderer. Grundsätzlich enthält eine Schaltung zur Messung von Temperaturen immer einen Operationsverstärker, der das Signal des Fühlers auf einen genügend grossen Pegel bringt. Operationsverstärker zeichnen sich durch einen hohen Verstärkungsfaktor bei sehr guter Linearität und kleinem Aufwand an externen Bauelementen aus. Das nachfolgende Schema soll eine solche Schaltung deutlich machen.



Der Operationsverstärker arbeitet hier als invertierender Verstärker, d.h., das am Eingang  $U_e$  liegende Signal erscheint mit umgekehrter Polarität am Ausgang des Verstärkers. Der Verstärkungsfaktor wird durch den Eingangswiderstand  $R_1$  und den Rückführungswiderstand  $R_2$  und  $R_3$  bestimmt. Der Verstärkungsfaktor ergibt sich aus der Formel:

$$V = \frac{R_2 + R_3}{R_1}$$



Im gezeigten Beispiel beträgt der Verstärkungsfaktor 1000. 1 mV am Eingang ergibt also eine Ausgangsspannung von 1 V. Mit dem Potentiometer P1 kann die Verstärkung genau abgeglichen werden.

Das Potentiometer P 2 dient zur Einstellung der Offsetspannung. Hier kann also der Verstärker auf Null abgeglichen werden. Am Ausgang der Schaltung kann ein niederohmiges Messinstrument angeschlossen werden, da die Stromstärke gross genug ist, um ein Zeigerinstrument zu speisen. Die Eingangsempfindlichkeit der Schaltung liegt bei 1 MOhm/V.

Als Messfühler für Temperaturen kommen NTC-Widerstände, Transistoren, Dioden und Thermoelemente in Frage.

NTC - Widerstände sind die billigste Lösung des Problems, haben aber mit Abstand auch die grössten Fehlerquellen. Ein NTC-Widerstand ist ein Halbleiter mit einem negativen Temperaturkoeffizient, d.h., je höher die Temperatur ist, desto kleiner ist der Widerstand des Bauelementes. Der NTC-Widerstand hat eine relativ grosse Masse und ist somit für sehr genaue Messaufgaben zu träge. Darüberhinaus ist er nur in einem relativ kleinen Temperaturbereich linear und nur für Temperaturen unter 150 Gr. C geeignet. Da der Strom, der durch den Widerstand fliesst nicht nur von seinem Widerstandswert, also der Temperatur, abhängt sondern auch von der angelegten Spannung, ist eine gut stabilisierte Versorgungsspannung erforderlich. Wir wollen hier aus diesem Grund auf den NTC-Widerstand als Messfühler verzichten.

Weiterhin können normale Transistoren und Dioden als Temperaturmessfühler benutzt werden, da auch bei diesen Bauelementen der Sperrstrom von der Temperatur abhängt. Da hier ähnliche Verhältnisse vorliegen wie bei den NTC-Widerständen, soll diese Art von Messfühlern ebenfalls unberücksichtigt bleiben.

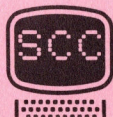
Schliesslich können als sehr gute Messfühler die Thermoelemente angesprochen werden.

Thermoelemente bestehen aus zwei verschiedenen Metallen, die miteinander verschweisst sind. Wird diese Schweissstelle erwärmt, so wird eine sehr kleine Thermospannung erzeugt. Diese Spannung ist für eine Direktmessung völlig ungeeignet und muss daher verstärkt werden. Bei Platin-Platinrhodiumelementen beträgt die Spannung  $0,06 \cdot 10^{-4}$  V/grad-1 und bei Kupfer-Konstantan  $0,42 \cdot 10^{-4}$  V/grad-1. Das Thermoelement ist von einer externen Versorgungsspannung unabhängig, da das von ihm erzeugte Potential direkt verstärkt wird. Durch die sehr kleine Masse der Schweissstelle ist das Element nicht träge und es kann durch Auswahl der verschiedenen Messfühler eine Temperaturmessung im Bereich von -100 bis +500 Grad Celsius vorgenommen werden.

Obwohl die oben beschriebene Schaltung sich auch zur Verstärkung des Thermolementstroms eignen würde, wurde bewusst ein fertig vergossener Messblock gewählt. Bei diesem fertig gelieferten Block ist die Linearität des Verstärkers bei unterschiedlicher Umgebungstemperatur hervorragend und vor allem auch die Nullpunktstabilität mit 0.04 Grad /Grad Celsius in einem absolut tragbaren Rahmen. Daneben ist die Eichung, die vor der ersten Benutzung vorgenommen werden muss, sehr einfach zu bewerkstelligen.

#### S C H A L T P L A N S E R V I C E

Unser Schaltplanservice erfreut sich einer grossen Beliebtheit. Leider wird nicht berücksichtigt, dass der zu überweisende Betrag von Fr. 2,-- pro Schaltplan 3 internationalen Postantwortscheinen entspricht. Der Betrag deckt gerade die Portokosten, wenn es sich um Auslandssendungen handelt. Wir bitten daher genügend Postantwortscheine der Anfrage beizulegen, da wir ansonsten Ihre Wünsche nicht mehr erfüllen können.





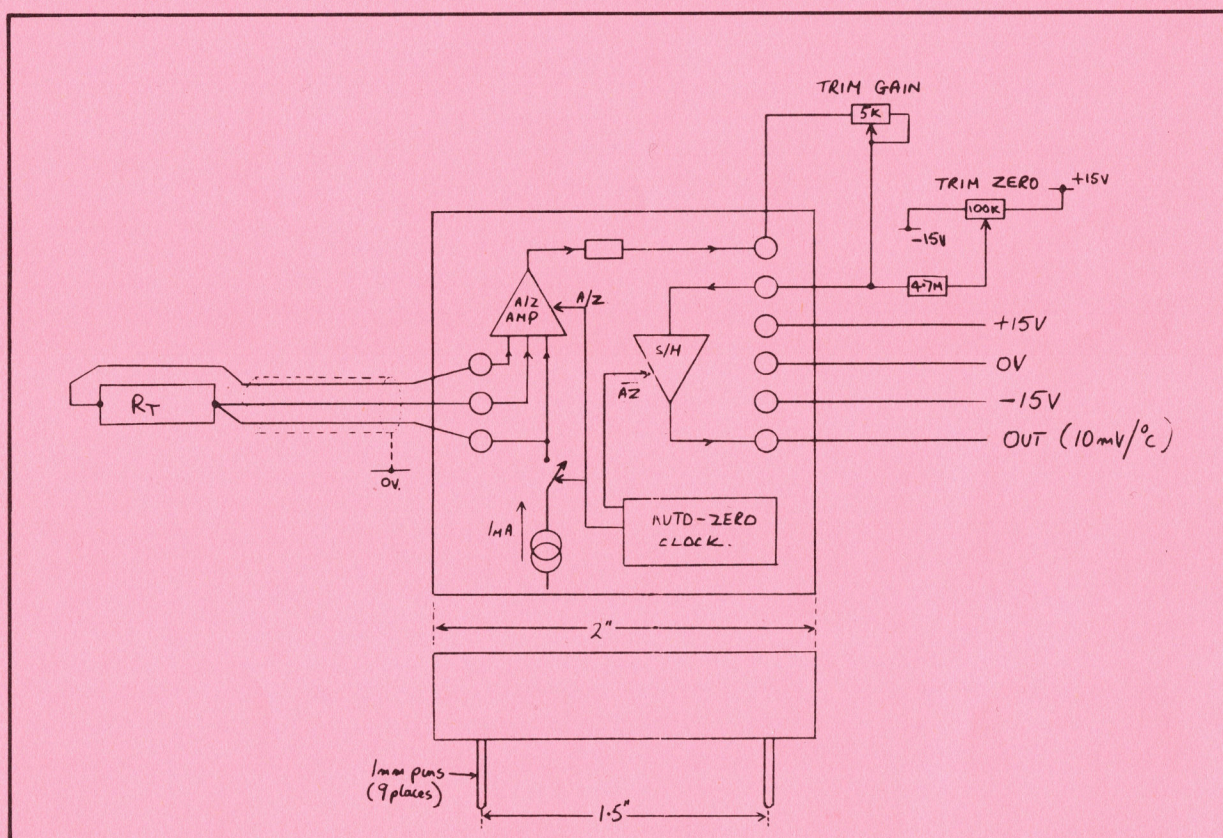
Der Temperaturmessverstärker benötigt eine Versorgungsspannung von  $\pm 15$  V und 10 mA Strom, die an der USER-PORT-Erweiterung zur Verfügung stehen.

Die Ausgangsspannung des Verstärkers beträgt 10 mV/Grad Celsius, d. h., der Analog-Digital-Wandler, dessen max. Eingangsspannung ohne vorgeschalteten Spannungsteiler 2 V beträgt, ist in der Lage Temperaturen zwischen  $-200$  und  $+200$  Grad Celsius zu messen.

Zu diesem Messverstärker gehört das Thermoelement, das in seiner einfachsten Ausführung Temperaturen zwischen  $-50$  und  $+150$  Grad Celsius zu messen vermag.

Die fertig montierte und abgegliche Schaltung kann als Platine direkt in die USER-PORT-Erweiterung eingesteckt werden, wobei lediglich die Anschlüsse für den Messfühler auf eine externe Buchse und der Ausgang des Verstärkers auf den Analog-Digitalwandler gegeben werden muss.

Das Programm zeigt ein Beispiel für die Messung und Anzeige der Raumtemperatur.



Blockschaltbild des Temperaturmessverstärkers

Die Platine ist zusammen mit einem Platinenstecker und 1 Fünfpolbuchse sowie einem Device-Selectschalter unter der Nummer P 2260 zum Preis von Fr. 595.-- durch den SCC erhältlich. Der Preis erscheint im ersten Moment ein wenig hoch, es ist jedoch zu bedenken, dass der Verstärker praktisch keine Nullpunktdrift aufweist und eine hervorragende Temperaturstabilität aufweist. Darüberhinaus kann an den Eingang der Schaltung natürlich auch ein anderer Messfühler z.B. für die Luftfeuchtigkeit oder den Luftdruck angeschlossen werden und auf diese Art, die Anwendungsmöglichkeiten um ein Vielfaches erweitert werden.



# Programm des Monats BASIC-BASIC

## RUBIK — WUERFEL

Der RUBIK - Würfel darf wohl als bekannt voraus gesetzt werden. Auch Programme zur Lösung des Problems sind in hinlänglicher Zahl in einschlägiger Literatur zitiert worden und sind auch beim SCC direkt erhältlich.

Wir möchten Ihnen hier ein Programm vorstellen, dass zwar das Würfelproblem nicht löst, jedoch bei der Lösung eine echte Hilfestellung bietet.

Beim Arbeiten mit dem Würfel ergibt sich das Problem, das man spätestens nach der dritten Drehung nicht mehr weiss, wie sich die Einzelflächen verschoben haben.

Unser Programm soll eine Würfelsimulation darstellen, mit dem alle Drehungen auf dem Bildschirm möglich sind, ohne überhaupt den Würfel in die Hand zu nehmen, die Darstellung des Würfels erfolgt auf dem Bildschirm von vorne und von hinten.

Die Besonderheit des Programms besteht im Ausdruck des Würfelbildes, da nach jedem Zug das gesamte Würfelbild auf einem Printer ausgedruckt werden kann. Hinzu kommt auch der Ausdruck der Art der Drehung.

Es ist also möglich, die Veränderung der einzelnen Flächen über den gesamten Spielverlauf zu dokumentieren und somit Drehungen auch gezielt rückgängig zu machen oder zu reproduzieren.

### Bedienungsanweisung

-----

Nach der Eingabe des Programms folgt die Frage nach Ausdruck (M/O), sowie die Frage ob man selbst die Anordnung treffen möchte (J/N).

Antwortet man auf die letzte Frage mit "N" (nicht selbst anordnen), erscheint das Würfelbild in der Ausgangsstellung auf dem Bildschirm, anschliessend die Frage nach dem Befehl.

Wird dagegen mit "J" geantwortet, müssen die 6 Flächen des Würfels zu je 9 Feldern selbst eingegeben werden. Bei der Frage nach den Seitenfeldern bedeuten die Buchstaben V, H, O, U, R, L, die Seite, z.B. V(2,2) für die linke obere Ecke der Vorderseite. Stellt man bei der Eingabe Fehler fest, kann mit **G** korrigiert werden. Entweder noch auf der gleichen Seite, oder die ganze Anordnung löschen.

Anschliessend erfolgt wiederum die Frage nach dem Befehl.

Jetzt wird die Art und die Seiten der Drehung eingegeben, für Drehungen nach rechts und oben +, für alle anderen -.

z.B. Drehung der oberen Seite nach rechts + O  
Drehung der rechten Seite nach links - R

Gibt man auf die Frage nach dem Befehl nur S ohne Vorzeichen ein, wird die Anordnung speziell gespeichert und kann später wieder zurückgeholt werden, z.B. nach mehrmaligen erfolglosen Versuchen mit Eingabe "R".

Hat man auf die Frage nach dem Ausdruck mit M geantwortet, wird jeweils der Befehl und die daraus resultierende Anordnung ausgedruckt.

Hat man mit O (ohne Ausdruck) geantwortet, so ist ein Ausdruck einzelner Züge trotzdem mit dem Befehl D möglich. Nach jeder zehnten Drehung wird automatisch die Druckroutine aufgerufen.



# RUBIK-WUERFEL

```

1500 REM RUBIK WUERFEL
1510 REM PROGRAMMIERT DURCH R. KONZ OKT. 1981
1520 REM NACH EINER IDEE VON J. W. PEHOUSHEK
1530 REM DAS PROGRAMM LAUFT NUR AUF CBM 3000 ODER 2000
1540 REM ERFORDERLICHER SPEICHERPLATZ 10K
1550 REM ZUM AUSDRUCK IST CBM PRINTER ERFORDERLICH
1560 REM PRINTERADRESSE 4
1570 PRINT"##### R U B I K - W U E R F E L ####"
1580 ZZ=0
1590 DIM R(5,5),L(5,5),F(5,5),B(5,5),U(5,5),D(5,5)
1600 DIM R1(5,5),L1(5,5),F1(5,5),B1(5,5),U1(5,5),D1(5,5)
1610 DIM T1(5,5),T2(5,5),C$(30)
1620 PRINT:PRINT"MOECHTEN SIE MIT ODER OHNE DRUCKER (M/O)"
1625 GETD$:IFD$=""THEN1625
1626 IFD$="M"ORD$="O"THEN1630
1627 GOTO1625
1630 PRINT:PRINT"MOECHTEN SIE SELBST ANORDNEN ?"
1635 GET A$:IF A$="J" THEN GOSUB 5030:GOTO 1700
1640 IF A$<>"N" THEN 1635
1650 REM INITIALISIEREN DER SEITENWERTE
1660 FOR R=2 TO 4:FOR C=2 TO 4
1670 F(R,C)=71:L(R,C)=87:U(R,C)=89
1680 B(R,C)=66:R(R,C)=82:D(R,C)=79
1690 NEXT C:NEXT R
1700 GOSUB 4920
1710 PRINT"J":PRINT:PRINT:PRINTTAB(8);"INITIALISIERTE ANORDNUNG"
1720 GOSUB 4160
1725 IF L>=30 THEN GOSUB 10000:C$="":GOTO 1740
1730 IF D$="M" THEN GOSUB 10000
1740 INPUT"BEFEHL =      #####";A$
1750 IF A$="S" THEN GOSUB 5420:GOTO 1720
1760 IF A$="D" THEN GOSUB 10000:GOTO 1720
1770 IF A$="R" THEN GOSUB 5610:GOTO 1720
1780 D=-1
1790 IF LEFT$(A$,1)="+" THEN D=1
1800 IF LEFT$(A$,1)="-" THEN D=0
1810 IF D<>-1 THEN 1840
1820 PRINT"FEHLER - NOCHMALS EINGEBEN"
1825 FORI=1TO500:NEXT:PRINT"J"
1830 GOTO 1740
1840 S=0
1850 A$=LEFT$(A$,2)
1860 IF RIGHT$(A$,1)="U" THEN S=1
1870 IF RIGHT$(A$,1)="O" THEN S=2
1880 IF RIGHT$(A$,1)="H" THEN S=3
1890 IF RIGHT$(A$,1)="V" THEN S=4
1900 IF RIGHT$(A$,1)="L" THEN S=5
1910 IF RIGHT$(A$,1)="R" THEN S=6
1920 IF S=0 THEN 1820
1930 C$=C$+A$+CHR$(32):L=LEN(C$)
1950 ON S GOSUB 2030,2350,2670,2990,3310,3630
1960 GOTO 1720
1970 END
1980 REM
1990 REM
2000 REM SUBROUTINEN
2010 REM
2020 REM BEWEGUNG DER R-SEITE
2030 IF D=2 GOTO 2230
2040 FOR R=1 TO 5:FOR C=1 TO 5

```

TT"



```

2670 IF D=2 GOTO 2870
2680 FOR R=1 TO 5:FOR C=1 TO 5
2690 T2(R,C)=B(R,C):NEXT C:NEXT R
2700 IF D=1 THEN GOSUB 3950
2710 IF D=0 THEN GOSUB 4010
2720 FOR R=1 TO 5:FOR C=1 TO 5
2730 B(R,C)=T1(R,C):NEXT C:NEXT R
2740 REM B-FACE (5*5) HAS BEEN ROTATED  NOW EQUATE EDGES
2750 R(1,4)=B(1,2):U(2,4)=B(1,2)
2760 U(2,3)=B(1,3)
2770 L(1,2)=B(1,4):U(2,2)=B(1,4)
2780 R(2,4)=B(2,1):U(2,5)=B(2,1)
2790 L(2,2)=B(2,5):U(2,1)=B(2,5)
2800 R(3,4)=B(3,1)
2810 L(3,2)=B(3,5)
2820 R(4,4)=B(4,1):D(5,2)=B(4,1)
2830 L(4,2)=B(4,5):D(1,2)=B(4,5)
2840 R(5,4)=B(5,2):D(4,2)=B(5,2)
2850 D(3,2)=B(5,3)
2860 L(5,2)=B(5,4):D(2,2)=B(5,4)
2870 R(2,5)=B(2,2):U(1,4)=B(2,2)
2880 U(1,3)=B(2,3)
2890 L(2,1)=B(2,4):U(1,2)=B(2,4)
2900 R(3,5)=B(3,2)
2910 L(3,1)=B(3,4)
2920 R(4,5)=B(4,2):D(4,1)=B(4,2)
2930 D(3,1)=B(4,3)
2940 L(4,1)=B(4,4):D(2,1)=B(4,4)
2950 RETURN
2960 REM
2970 REM
2980 REM  F-FACE ROTATE
2990 IF D=2 GOTO 3190
3000 FOR R=1 TO 5:FOR C=1 TO 5
3010 T2(R,C)=F(R,C):NEXT C:NEXT R
3020 IF D=1 THEN GOSUB 3950
3030 IF D=0 THEN GOSUB 4010
3040 FOR R=1 TO 5:FOR C=1 TO 5
3050 F(R,C)=T1(R,C):NEXT C:NEXT R
3060 REM F-FACE (5*5) HAS BEEN ROTATED  NOW EQUATE EDGES
3070 L(1,4)=F(1,2):U(4,2)=F(1,2)
3080 U(4,3)=F(1,3)
3090 R(1,2)=F(1,4):U(4,4)=F(1,4)
3100 L(2,4)=F(2,1):U(4,1)=F(2,1)
3110 R(2,2)=F(2,5):U(4,5)=F(2,5)
3120 L(3,4)=F(3,1)
3130 R(3,2)=F(3,5)
3140 L(4,4)=F(4,1):D(1,4)=F(4,1)
3150 R(4,2)=F(4,5):D(5,4)=F(4,5)
3160 L(5,4)=F(5,2):D(2,4)=F(5,2)
3170 D(3,4)=F(5,3)
3180 R(5,2)=F(5,4):D(4,4)=F(5,4)
3190 L(2,5)=F(2,2):U(5,2)=F(2,2)
3200 U(5,3)=F(2,3)
3210 R(2,1)=F(2,4):U(5,4)=F(2,4)
3220 L(3,5)=F(3,2)
3230 R(3,1)=F(3,4)
3240 L(4,5)=F(4,2):D(2,5)=F(4,2)
3250 D(3,5)=F(4,3)
3260 R(4,1)=F(4,4):D(4,5)=F(4,4)
3270 RETURN
3280 REM

```



```

2050 T2(R,C)=D(R,C):NEXT C:NEXT R
2060 IF D=1 THEN GOSUB 3950
2070 IF D=0 THEN GOSUB 4010
2080 FOR R=1 TO 5:FOR C=1 TO 5
2090 D(R,C)=T1(R,C):NEXT C:NEXT R
2100 REM R-SEITE (5*5) HAT GEDREHT ; ECKEN AUSGLEICHEN
2110 B(4,5)=D(1,2):L(4,2)=D(1,2)
2120 L(4,3)=D(1,3)
2130 F(4,1)=D(4,1):L(4,4)=D(1,4)
2140 B(4,4)=D(2,1):L(4,1)=D(2,1)
2150 F(4,2)=D(2,5):L(4,5)=D(2,5)
2160 B(4,3)=D(3,1)
2170 F(4,3)=D(3,5)
2180 B(4,2)=D(4,1):R(4,5)=D(4,1)
2190 F(4,4)=D(4,5):R(4,1)=D(4,5)
2200 B(4,1)=D(5,2):R(4,4)=D(5,2)
2210 R(4,3)=D(5,3)
2220 F(4,5)=D(5,4):R(4,2)=D(5,4)
2230 B(5,4)=D(2,2):L(5,2)=D(2,2)
2240 L(5,3)=D(2,3)
2250 F(5,2)=D(2,4):L(5,4)=D(2,4)
2260 B(5,3)=D(3,2)
2270 F(5,3)=D(3,4)
2280 B(5,2)=D(4,2):R(5,4)=D(4,2)
2290 R(5,3)=D(4,3)
2300 F(5,4)=D(4,4):R(5,2)=D(4,4)
2310 RETURN
2320 REM
2330 REM
2340 REM U-FACE ROTATE
2350 IF D=2 GOTO 2550
2360 FOR R=1 TO 5:FOR C=1 TO 5
2370 T2(R,C)=U(R,C):NEXT C:NEXT R
2380 IF D=1 THEN GOSUB 3950
2390 IF D=0 THEN GOSUB 4010
2400 FOR R=1 TO 5:FOR C=1 TO 5
2410 U(R,C)=T1(R,C):NEXT C:NEXT R
2420 REM U-FACE (5*5) HAS BEEN ROTATED NOW EQUATE EDGES
2430 L(2,1)=U(1,2):B(2,4)=U(1,2)
2440 B(2,3)=U(1,3)
2450 R(2,5)=U(1,4):B(2,2)=U(1,4)
2460 L(2,2)=U(2,1):B(2,5)=U(2,1)
2470 R(2,4)=U(2,5):B(2,1)=U(2,5)
2480 L(2,3)=U(3,1)
2490 R(2,3)=U(3,5)
2500 L(2,4)=U(4,1):F(2,1)=U(4,1)
2510 R(2,2)=U(4,5):F(2,5)=U(4,5)
2520 L(2,5)=U(5,2):F(2,2)=U(5,2)
2530 F(2,3)=U(5,3)
2540 R(2,1)=U(5,4):F(2,4)=U(5,4)
2550 L(1,2)=U(2,2):B(1,4)=U(2,2)
2560 B(1,3)=U(2,3)
2570 R(1,4)=U(2,4):B(1,2)=U(2,4)
2580 L(1,3)=U(3,2)
2590 R(1,3)=U(3,4)
2600 L(1,4)=U(4,2):F(1,2)=U(4,2)
2610 F(1,3)=U(4,3)
2620 R(1,2)=U(4,4):F(1,4)=U(4,4)
2630 RETURN
2640 REM
2650 REM
2660 REM B-FACE ROTATE

```



```

3290 REM
3300 REM L-FACE ROTATE
3310 IF D=2 GOTO 3510
3320 FOR R=1 TO 5:FOR C=1 TO 5
3330 T2(R,C)=L(R,C):NEXT C:NEXT R
3340 IF D=1 THEN GOSUB 3950
3350 IF D=0 THEN GOSUB 4010
3360 FOR R=1 TO 5:FOR C=1 TO 5
3370 L(R,C)=T1(R,C):NEXT C:NEXT R
3380 REM L-FACE (5*5) HAS BEEN ROTATED NOW EQUATE EDGES
3390 B(1,4)=L(1,2):U(2,2)=L(1,2)
3400 U(3,2)=L(1,3)
3410 F(1,2)=L(1,4):U(4,2)=L(1,4)
3420 B(2,4)=L(2,1):U(1,2)=L(2,1)
3430 F(2,2)=L(2,5):U(5,2)=L(2,5)
3440 B(3,4)=L(3,1)
3450 F(3,2)=L(3,5)
3460 B(4,4)=L(4,1):D(2,1)=L(4,1)
3470 F(4,2)=L(4,5):D(2,5)=L(4,5)
3480 B(5,4)=L(5,2):D(2,2)=L(5,2)
3490 D(2,3)=L(5,3)
3500 F(5,2)=L(5,4):D(2,4)=L(5,4)
3510 B(2,5)=L(2,2):U(2,1)=L(2,2)
3520 U(3,1)=L(2,3)
3530 F(2,1)=L(2,4):U(4,1)=L(2,4)
3540 B(3,5)=L(3,2)
3550 F(3,1)=L(3,4)
3560 B(4,5)=L(4,2):D(1,2)=L(4,2)
3570 D(1,3)=L(4,3)
3580 F(4,1)=L(4,4):D(1,4)=L(4,4)
3590 RETURN
3600 REM
3610 REM
3620 REM R-FACE ROTATE
3630 IF D=2 GOTO 3830
3640 FOR R=1 TO 5:FOR C=1 TO 5
3650 T2(R,C)=R(R,C):NEXT C:NEXT R
3660 IF D=1 THEN GOSUB 3950
3670 IF D=0 THEN GOSUB 4010
3680 FOR R=1 TO 5:FOR C=1 TO 5
3690 R(R,C)=T1(R,C):NEXT C:NEXT R
3700 REM R-FACE (5*5) HAS BEEN ROTATED NOW EQUATE EDGES
3710 F(1,4)=R(1,2):U(4,4)=R(1,2)
3720 U(3,4)=R(1,3)
3730 B(1,2)=R(1,4):U(2,4)=R(1,4)
3740 F(2,4)=R(2,1):U(5,4)=R(2,1)
3750 B(2,2)=R(2,5):U(1,4)=R(2,5)
3760 F(3,4)=R(3,1)
3770 B(3,2)=R(3,5)
3780 F(4,4)=R(4,1):D(4,5)=R(4,1)
3790 B(4,2)=R(4,5):D(4,1)=R(4,5)
3800 F(5,4)=R(5,2):D(4,4)=R(5,2)
3810 D(4,3)=R(5,3)
3820 B(5,2)=R(5,4):D(4,2)=R(5,4)
3830 F(2,5)=R(2,2):U(4,5)=R(2,2)
3840 U(3,5)=R(2,3)
3850 B(2,1)=R(2,4):U(2,5)=R(2,4)
3860 F(3,5)=R(3,2)
3870 B(3,1)=R(3,4)
3880 F(4,5)=R(4,2):D(5,4)=R(4,2)
3890 D(5,3)=R(4,3)
3900 B(4,1)=R(4,4):D(5,2)=R(4,4)

```



```

3910 RETURN
3940 REM DREHUNG NACH RECHTS
3950 FOR R=1 TO 5:FOR C=1 TO 5
3960 T1(C,6-R)=T2(R,C):NEXT C:NEXT R
3970 RETURN
4010 FOR R=1 TO 5:FOR C=1 TO 5
4020 T1(6-C,R)=T2(R,C)
4030 NEXT C:NEXT R
4040 RETURN
4130 REM
4140 REM
4150 REM WUERFEL VON BEIDEN SEITEN
4160 PRINT"XXXXXXXXXX"
4170 PRINT TAB(4);
4180 FOR C=2 TO 4
4190 PRINT CHR$(U(2,C));
4200 NEXT C
4210 PRINT TAB(18);
4220 FOR C=4 TO 2 STEP -1
4230 PRINT CHR$(U(4,C));
4240 NEXT C
4250 PRINT:PRINT TAB(4);
4260 FOR C=2 TO 4:PRINT CHR$(U(3,C));:NEXT C
4270 PRINT TAB(18);
4280 FOR C=4 TO 2 STEP -1:PRINT CHR$(U(3,C));:NEXT C
4290 PRINT:PRINT TAB(4);
4300 FOR C=2 TO 4
4310 PRINT CHR$(U(4,C));
4320 NEXT C
4330 PRINT TAB(18);
4340 FOR C=4 TO 2 STEP -1
4350 PRINT CHR$(U(2,C));
4360 NEXT C
4370 PRINT:PRINT
4380 FOR A=1 TO 6
4390 FOR C=2 TO 4
4400 IF A=1 THEN PRINT CHR$(L(2,C));
4410 IF A=2 THEN PRINT CHR$(F(2,C));
4420 IF A=3 OR A=4 THEN PRINT CHR$(R(2,C));
4430 IF A=5 THEN PRINT CHR$(B(2,C));
4440 IF A=6 THEN PRINT CHR$(L(2,C));
4450 NEXT C:PRINT SPC(1);
4460 IF A=3 THEN PRINT SPC(2);
4470 NEXT A
4480 PRINT
4490 FOR C=2 TO 4:PRINT CHR$(L(3,C));:NEXT C:PRINT SPC(1);
4500 FOR C=2 TO 4:PRINT CHR$(F(3,C));:NEXT C:PRINT SPC(1);
4510 FOR C=2 TO 4:PRINT CHR$(R(3,C));:NEXT C:PRINT SPC(3);
4520 FOR C=2 TO 4:PRINT CHR$(R(3,C));:NEXT C:PRINT SPC(1);
4530 FOR C=2 TO 4:PRINT CHR$(B(3,C));:NEXT C:PRINT SPC(1);
4540 FOR C=2 TO 4:PRINT CHR$(L(3,C));:NEXT C
4550 PRINT
4560 FOR A=1 TO 6
4570 FOR C=2 TO 4
4580 IF A=1 THEN PRINT CHR$(L(4,C));
4590 IF A=2 THEN PRINT CHR$(F(4,C));
4600 IF A=3 OR A=4 THEN PRINT CHR$(R(4,C));
4610 IF A=5 THEN PRINT CHR$(B(4,C));
4620 IF A=6 THEN PRINT CHR$(L(4,C));
4630 NEXT C:PRINT SPC(1);
4640 IF A=3 THEN PRINT SPC(2);
4650 NEXT A

```



```

4660 PRINT:PRINT
4670 PRINT TAB(4);
4680 FOR R=2 TO 4
4690 PRINT CHR$(D(R,4));
4700 NEXT R
4710 PRINT TAB(18);
4720 FOR R=4 TO 2 STEP -1
4730 PRINT CHR$(D(R,2));
4740 NEXT R
4750 PRINT:PRINT TAB(4);
4760 FOR R=2 TO 4:PRINT CHR$(D(R,3));:NEXT R
4770 PRINT TAB(18);
4780 FOR R=4 TO 2 STEP -1:PRINT CHR$(D(R,3));:NEXT R
4790 PRINT:PRINT TAB(4);
4800 FOR R=2 TO 4
4810 PRINT CHR$(D(R,2));
4820 NEXT R
4830 PRINT TAB(18);
4840 FOR R=4 TO 2 STEP -1
4850 PRINT CHR$(D(R,4));
4860 NEXT R
4870 PRINT:PRINT:PRINT
4880 RETURN
4890 REM
4900 REM
4910 REM 5*5-REIHE INITIALISIEREN
4920 D=2
4930 GOSUB 2030
4940 GOSUB 2350
4950 GOSUB 2670
4960 GOSUB 2990
4970 GOSUB 3310
4980 GOSUB 3630
4990 RETURN
5000 REM
5010 REM
5020 REM EIGENE WUERFELANORDNUNG
5030 PRINT:PRINT"DIE SEITEN WERDEN FOLGENDERMASSEN GEKENNZEICHNET:
5040 PRINT"      V ( 2,2 ) = OBERE LINKE ECKE"
5050 PRINT"      V ( 2,4 ) = OBERE RECHTE ECKE"
5060 PRINT"      V ( 4,2 ) = UNTERE LINKE ECKE"
5070 PRINT"      ETC.
5080 PRINT:PRINT"GEBEN SIE DEN FARBBUCHSTABEN EIN":PRINT
5090 PRINT:PRINT
5100 FOR R=2 TO 4:FOR C=2 TO 4
5110 PRINT"SEITE V (":R;C:") =":INPUT A$:S=1:GOTO 5800
5120 F(R,C)=ASC(A$):NEXT C:NEXT R
5130 FOR R=2 TO 4:FOR C=2 TO 4
5140 PRINT"SEITE H (":R;C:") =":INPUT A$:S=2:GOTO 5800
5150 B(R,C)=ASC(A$):NEXT C:NEXT R
5160 FOR R=2 TO 4:FOR C=2 TO 4
5170 PRINT"SEITE L (":R;C:") =":INPUT A$:S=3:GOTO 5800
5180 L(R,C)=ASC(A$):NEXT C:NEXT R
5190 FOR R=2 TO 4:FOR C=2 TO 4
5200 PRINT"SEITE R (":R;C:") =":INPUT A$:S=4:GOTO 5800
5210 R(R,C)=ASC(A$):NEXT C:NEXT R
5220 FOR R=2 TO 4:FOR C=2 TO 4
5230 PRINT"SEITE O (":R;C:") =":INPUT A$:S=5:GOTO 5800
5240 U(R,C)=ASC(A$):NEXT C:NEXT R
5250 REM D-SEITE HAT ANDERE REIHENFOLGE
5260 FOR R=2 TO 4:FOR C=2 TO 4
5270 PRINT"SEITE U (":R;C:") =":INPUT A$:S=6:GOTO 5800

```



```

5280 D(C,6-R)=ASC(A$):NEXT C:NEXT R:PRINT
5290 RETURN
5410 REM  SPEICHER SUBROUTINE
5420 INPUT"SICHER ";A$
5430 IF A$<>"J" THEN RETURN
5440 FOR R=2 TO 4:FOR C=2 TO 4
5450 F1(R,C)=F(R,C):L1(R,C)=L(R,C):U1(R,C)=U(R,C)
5460 B1(R,C)=B(R,C):R1(R,C)=R(R,C):D1(R,C)=D(R,C)
5470 NEXT C:NEXT R
5500 PRINT"NEUE ANORDNUNG IST IM SPEICHER"
5520 RETURN
5600 REM  LADEN DER GESP. ANORDNUNG
5610 INPUT"SICHER ";A$
5620 IF A$<>"J" THEN RETURN
5630 FOR R=2 TO 4:FOR C=2 TO 4
5640 F(R,C)=F1(R,C):L(R,C)=L1(R,C):U(R,C)=U1(R,C)
5650 B(R,C)=B1(R,C):R(R,C)=R1(R,C):D(R,C)=D1(R,C)
5660 NEXT C:NEXT R
5670 GOSUB 4920
5680 RETURN
5800 IF A$="G" THEN GG=GG+1:X=GG:GOTO 5900
5810 IF A$="R" THEN RR=RR+1:X=RR:GOTO 5900
5820 IF A$="B" THEN BB=BB+1:X=BB:GOTO 5900
5830 IF A$="W" THEN WW=WW+1:X=WW:GOTO 5900
5840 IF A$="Y" THEN YY=YY+1:X=YY:GOTO 5900
5850 IF A$="O" THEN OO=OO+1:X=OO:GOTO 5900
5860 IF A$="@" THEN INPUT"FEHLER IN DIESER SEITE ";F$
5865 IF F$="N" THEN F$="":GOTO 5100
5870 IF F$="J" THEN F$="":ON S GOTO 5100,5130,5160,5190,5220,5260
5880 PRINT"DIE FARBEN SIND G,R,B,W,Y,O !"
5890 GOTO 5920
5900 IF X>9 THEN PRINT:PRINT"FARBE "A$" IST GESETZT !":GOTO 5920
5910 ON S GOTO 5120,5150,5180,5210,5240,5280
5920 ON S GOTO 5110,5140,5170,5200,5230,5270      ZZ=1:GOTO 10020
10000 OPEN1,6:IF ZZ=0 THEN PRINT#1,CHR$(1)"RUBIK-WUERFEL":PRINT#1:
10010 PRINT#1:PRINT#1,CHR$(1)ZZ:". DREHUNG : ";A$:PRINT#1:ZZ=ZZ+1
10020 J=-1
10040 FOR P=33048 TO 33528
10050 J=J+1:IF J=40 THEN J=0:PRINT#1
10060 Y=PEEK(P)
10070 IF Y>64 THEN Y=32+Y
10080 IF Y>255 THEN Y=32
10090 IF Y<32 THEN Y=64+Y
10100 PRINT#1,CHR$(Y);
10110 NEXT
10120 PRINT#1
10130 CLOSE 1
10140 RETURN
READY.

```

#### Programmbeschreibung

```

1500 - 1580 Erklärungen.
1590 - 1610 Felder dimensionieren.
1620 - 1625 Frage nach dem Ausdruck.
1630 - 1640 Frage nach der Anordnung.
1650 - 1690 Festsetzen der Seitenwerte.
1720      Sprung zur Routine für den Bildschirmausdruck.
1730      Verzweigung für die Printeroutine.

```



1740           Eingabe des Rotierbefehls.  
 1750 - 1820 Analyse des Befehls und Verzweigung.  
 1840 - 1920 Richtiger Befehl analysieren und Seitenvariable S setzen.  
 2020 - 3910 Die Variablen für die Seiten werden bezeichnet.  
 2040 - 2090 Drehung der Seiten.  
 2110 - 2300 Ecken sowie Seiten anpassen.  
 2310 - 3910 Programm ist analog den Zeilen 2040 - 2300.  
 3950 - 3970 Drehung der Seiten nach rechts bzw. nach oben.  
 4010 - 4040 Drehung der Seiten nach links bzw. nach unten.  
 4160 - 4880 Zeichnen des Würfels von beiden Seiten auf den Bildschirm.  
 4170 - 4350 Zeichnen der oberen Seite.  
 4380 - 4650 Zeichnen der senkrechten Seiten.  
 4670 - 4860 Zeichnen der unteren Seite.  
 4920 - 4990 Aufruf der Subroutinen zur Drehung der Seiten.  
 5030 - 5070 Erklärung zur eigenen Zusammenstellung des Würfels.  
 5100 - 5290 Eingabeprogramm.  
 5440 - 5520 Routine zur Abspeicherung der Anordnung.  
 5630 - 5680 Routine zum Abruf der gespeicherten Anordnung.  
 5800 - 5850 Variable für gebrauchte Farben setzen.  
 5860 - 5880 Korrekturen der Eingabe.  
 5900           Es wird verhindert, dass mehr als 9 Felder der gleichen Farbe gesetzt werden.  
 10000          Hardcopysubroutine zur Ausgabe auf dem Printer.

## VORSCHAU 1982

### Tricks und Tips

-----

Lagerbuchhaltung  
 Adressverwaltung  
 Programmiertips  
 Floppy-Hilfsroutinen

### Hardware

-----

Joystick  
 Barcodeleser  
 programmierbarer Dimer  
 Light-Pen

### BASIC - BASIC

-----

Multiplotter  
 Adressverwaltung Cassette  
 Morse-Trainer  
 diverse Spiele

### Maschinensprache

-----

Barcodeleser  
 Barcodedrucker  
 Mehrstimmige Musik



# Maschinensprache

## MUSIK — PROGRAMMIERUNG

In der heutigen Ausgabe "Assembler Programmierung" wollen wir Ihnen zeigen, wie ein Programm aufgebaut ist, das selbständig und ohne BASIC verschiedene Schwingungen erzeugt.

Mit dem im CBM eingebauten Baustein VIA 6522 können diese Schwingungen über den Ausgangskanal CB2 mit einem an den USER-PORT angeschlossenen Musikzusatz hörbar gemacht werden.

Als Grundlage für die Notengebung diente die im Heft 5/81 Seite 15 erschienene Tabelle.

Das Maschinenprogramm ist im Prinzip einfach und teilt sich in folgende Funktionsblöcke auf:

- a) die Hardware für die Tonausgabe vorbereiten.
- b) den Notenwert aus der Notentabelle lesen und ausgeben.
- c) die Tondauer aus der Tabelle lesen und entsprechend lange warten.
- d) ist die Notentabelle abgearbeitet, so wird die Hardware wieder in den Originalzustand zurückgesetzt und das Maschinenprogramm verlassen. Im anderen Fall wird die Notentabelle weiter bearbeitet.

Dieses Programm ist noch nicht für die komplexe und mehrstimmige Musik ausgelegt, die in einer der nächsten Ausgaben dieses Heftes folgen soll.

Nun die einzelnen Abschnitte im Detail:

a) Damit der universelle Baustein VIA 6522 hohe Frequenzen abgibt, muss im Controlregister ACR das 'FRE RUNNING BIT' ACR 4 gesetzt werden. Durch diese Massnahme wird das interne Schieberegister angesteuert, welches dann die verschiedenen Frequenzen über die Ausgangsleitung CB 2 zum USER-PORT abgibt (Grundfrequenz 32 kHz).

In BASIC schreibt man: POKE 59467,16

Hier gilt: LDA #16  
STA OSZI ; Oscillator einschalten

wobei alle Adressen im Programmkopf definiert sind.  
Die Klangfarbe der Töne kann hier in engen Grenzen variiert werden. Dies wird erreicht, indem das Schieberegister mit verschiedenen Bitmustern geladen wird (8 Bit). Dieses Bitmuster wird vom Schieberegister des Bausteins dauernd ausgegeben.

In BASIC schreibt man: POKE 59466,k  
interessante Töne ergeben die Werte k=60,85,104 .....

Hier gilt: LDA #104  
STA KFARB

b) An dieser Stelle werden die Noten, resp. die verschiedenen Frequenzen, ebenso die Tondauer aus der Notentabelle gelesen und ausgegeben.

In der ersten Spalte sind die Noten, in der zweiten Spalte findet man die Spieldauer für die Note.

Der Notenwert 0 gibt eine Pause, die Spieldauer 0 am Ende der Tabelle



beendet das Programm in der Ausführung.

Die Notenwerte in unserem Maschinenprogramm sind als Beispiel gedacht und basieren auf der kleinen Mozart Sonate. (Die Noten sind: A C# E G# A B A F# E A E D C#)

Das Ausgeben der Noten ab der Notentabelle erfolgt im Programm in den Zeilen 19 - 34.

In BASIC schreibt man: POKE 59464,n  
wobei n die Werte aus der Notentabelle sind.

Hier schreibt man: LDY #0 ; Offset setzen  
LDA (NTI),Y; Notenwert aus Tabelle lesen.  
STA FREQ ; und ausgeben.

Mit dem letzten Befehl wird erreicht, dass die Grundfrequenz durch den Tabellenwert geteilt wird. Dies ergibt den gewünschten Ton.

z.B. 32768/149 ergibt die Frequenz von ca. 220 Hz.  
Dies entspricht der Note A.

c) Für die Tondauer wurde als Zeitschleifenwert 70 gewählt (Programmzeile 26). Sollen die Töne schneller ausgegeben werden, so ist ein kleinerer Wert zu wählen (experimentiere!).

In der Subroutine INCNT wird jeweils der Tabellenzeiger NTI (2Byte) in der Tabelle weitergeschaltet.

d) Erkennt das Programm das Tabellenende mit den Zeilen :

LDA (NTI).Y; Tondauer aus der Tabelle lesen  
STA CNT ; sen und in Counter setzen  
BEQ L7 ; Springe, wenn Musikabbruch

so wird der Tonoscillator abgestellt und das Musikprogramm verlassen. Auf diese Art ist es also möglich, Tonfolgen, Sirenen, Geräusche usw. mit einem reinen Maschinenprogramm zu erzeugen, das zudem sehr wenig Speicherplatz benötigt.

Hier stellt sich die Frage, kann das Maschinenprogramm auch als Subroutine von einem BASIC-Programm aus aufgerufen werden? Natürlich ist dies möglich, sogar ohne irgendwelche Änderungen.

Es kann direkt aufgerufen werden mit:

SYS(830) : REM TONAUSGABE

Es können auch Varianten eingeführt werden:

Sie wollen das Maschinenprogramm dazu benutzen, um bei einem Spielprogramm verschiedene Geräusche oder Pfeiftöne abzugeben, (z.B. jeder Tastendruck gibt ein kurzer Pieps), so brauchen Sie nur die Notentabelle entsprechend aufzubauen. Dazu wird jeweils vor dem Aufruf der Notenpointer NTP auf das gewünschte Geräusch gerichtet. Der Notenpointer befindet sich auf der Adresse 898 und 899 (dezimal). Möchte man z.B. nur die zweite Hälfte der Mozartsonate hören, so müsste man schreiben :

POKE 898,148 : REM Lo Byte Pointer NTP  
POKE 899,3 : REM Hi Byte Pointer NTP setzen  
SYS(830)

Die Werte 148 und 3 zeigen jetzt  $3 \cdot 256 + 148 = \text{916}$ , entspricht \$0394

Will man die Geschwindigkeit der Tonausgabe beschleunigen, so braucht man nur zu schreiben :



Dies verringert den Zähler der Zeitschleife an der Stelle \$0363.  
Durch dauerndes Aendern dieser Parameter können Sie mit einem BASIC-  
Programm die interessantesten Toneffekte erzeugen.

```

-----
* CBM * ASSEMBLER 6502 *                                MUSIK LST                                PAGE 1
-----
033E      1  **830
033E      2  !
E84B      3  OSZI = 59467 ;ADRESSEN VOM BAUSTEIN VIA 6522
E84A      4  KFARB = 59466 ;FREILAUFENDER OZILLATOR
E848      5  FREQ = 59464 ;KLANGFARBENEINSTELLUNG
033E      6  !
033E      7  ! ;ZEROPAGE ADRESSEN FUER:
004D      8  NTI = 77 ;NOTENTABELLEN ZEIGER INDIREKT
004F      9  CNT = 79 ;COUNTER FUER ZEITSCHLEIFE
033E     10  !
033E AD8203 11  START LDA NTP ;DEN NOTENTABELLEN ZEIGER
0341 854D 12  STA NTI ;FESTLEGEN
0343 AD8303 13  LDA NTP+1
0346 854E 14  STA NTI+1
0348 A910 15  LDA #16
034A 8D4BE8 16  STA OSZI ;OSZILLATOR EINSCHALTEN
034D A968 17  LDA #104 ;KLANGFARBE FESTLEGEN
034F 8D4AE8 18  STA KFARB
0352 A000 19  L1 LDY #0 ;OFFSETWERT SETZEN
0354 B14D 20  LDA (NTI),Y ;NOTENWERT AUS TABELLE LESEN
0356 8D48E8 21  STA FREQ ;UND AUSGEBEN
0359 207B03 22  JSR INCNT ;DEN ZEIGER WEITERSCHALTEN
035C B14D 23  LDA (NTI),Y ;TONDAUER AUS DER TABELLE LESEN
035E 854F 24  STA CNT ;UND IN COUNTER SETZEN
0360 F012 25  BEQ L7 ;SPRINGE WENN MUSIK ABRUCH
0362 A246 26  L4 LDX #70 ;ZAEHLER FUER ZEITSCHLEIFE
0364 88 27  L5 DEY
0365 D0FD 28  BNE L5
0367 CA 29  DEX
0368 D0FA 30  BNE L5 ;SPRINGE WENN ZEIT NICHT FERTIG
036A C64F 31  DEC CNT
036C D0F4 32  BNE L4 ;SPRINGE WENN COUNTER NICHT FERTIG
036E 207B03 33  JSR INCNT
0371 4C5203 34  JMP L1 ;NAECHSTE NOTE VERARBEITEN
0374 35  ;
0374 8D48E8 36  L7 STA FREQ ;TON UND OSZILLATOR
0377 8D4BE8 37  STA OSZI ;ABSTELLEN
037A 60 38  RTS ;MUSIK SUBROUTINE VERLASSEN
037B 39  ;
037B E64D 40  INCNT INC NTI ;SUBROUTINE ZUR WEITER-
037D D002 41  BNE L9 ;SCHALTUNG DES TABELLENZEIGERS
037F E64E 42  INC NTI+1
0381 60 43  L9 RTS
0382 44  ;
0382 8403 45  NTP .WORD NOTE
0384 46  ;
0384 9509 47  NOTE .BYTE 149,9
0386 EC04 48  .BYTE 236,4
0388 C704 49  .BYTE 199,4
038A 9E05 50  .BYTE 158,5
038C 9501 51  .BYTE 149,1
038E 8D01 52  .BYTE 141,1
0390 950F 53  .BYTE 149,15
0392 0002 54  .BYTE 0,2
0394 B109 55  .BYTE 177,9
0396 C704 56  .BYTE 199,4
0398 9504 57  .BYTE 149,4
039A C704 58  .BYTE 199,4
039C DF04 59  .BYTE 223,4
039E EC09 60  .BYTE 236,9
03A0 0000 61  .BYTE 0,0
03A2 62  ;
03A2 63  END

```

14 SYMBOLS 0 ERRORS 100 BYTES

```

100 REM -----
110 REM #    MUSIK    CBM 3001    #
120 REM -----
130 REM
140 FOR M=830 TO 929: READ A: POKE M,A: NEXT M
150 PRINT: PRINT"DATA'S GELADEN."
160 REM
10000 DATA 173,130,3,133,77,173,131,3,133,78,169,16,141,75,232,169
10010 DATA 104,141,74,232,160,0,177,77,141,72,232,32,123,3,177,77
10020 DATA 133,79,240,18,162,70,136,208,253,202,208,250,198,79,208,244
10030 DATA 32,123,3,76,82,3,141,72,232,141,75,232,96,230,77,208
10040 DATA 2,230,78,96,132,3,149,9,236,4,199,4,158,5,149,1
10050 DATA 141,1,149,15,0,2,177,9,199,4,149,4,199,4,223,4
10060 DATA 236,9,0,0
10070 END

```



# Schweizer Computer Club ☎ 041 - 31 45 45

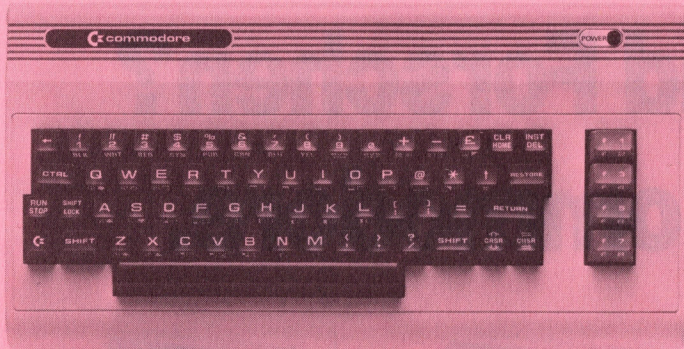


Sofort lieferbar!

## VC=20

### Der Vollprofi-Kleincomputer für den Ersteinsteiger oder als Zweitgerät!

Kompakt wie eine Reiseschreibmaschine und ebenso problemlos zu handhaben, präsentiert sich der VC 20, der Volkscomputer von Commodore. Basic-Interpreter, Betriebssystem, Arbeitsspeicher und Videoanschluss für das Fernsehgerät (auch Farbe) sind eingebaut. Mit 20 KB ROM im Festwertspeicher ist er kein «Spielzeug» mehr, sondern ein vollwertiger Computer, der zu einem System ausgebaut werden kann.



P 2160	SK	VC 20	795.—
P 2090		Kassettenrekorder mit Zählwerk	195.—
P 2141		Single Floppy 175 K	1675.—
P 2165		Speichererweiterung 3 K	99.—*
P 2166		Speichererweiterung 8 K	149.—*
P 2167		Speichererweiterung 16 K	ca. 250.—*
		Drucker, Software usw. auf Anfrage	
		* erst Anfang 1982 lieferbar	

### Gelegenheit!

Pet-2001-Platinen 8 K, Basic 1,0, geprüft,  
3 Monate Garantie

**Fr. 650.—**

## commodore COMPUTER

### CBM/PET

P 2118	16 K	CBM 4016 (exkl. Rekorder)	2475.—
P 2134	32 K	CBM 4032 (exkl. Rekorder)	2975.—
P 2037	32 K	<b>CBM 8032, 80 Zeichen (exkl. Rekorder)</b>	<b>3475.—</b>
P 2038		<b>CBM 8050 Dual Floppy 1 Megabyte</b>	<b>3975.—</b>
P 2028	Neu	Tractor zu CBM 8026/27	595.—
P 2096	96 K	CBM 8096	4675.—
P 2140		CBM Dual Floppy 4040	3475.—
P 2025		CBM 8024 9 x 7 Matrixdrucker	ab 4350.—
P 2026		CBM 8026 Typenrad-Drucker mit Tastatur	3750.—
P 2027		CBM 8026 Typenrad-Drucker ohne Tastatur	2750.—
P 2044		<b>CBM Drucker 4022</b>	<b>1750.—</b>
P 2064		<b>CBM Add-On Speichererweiterung für 8032 auf 64 K inkl. Einbau</b>	<b>1400.—</b>

### Öffnungszeiten im SCC-Shop

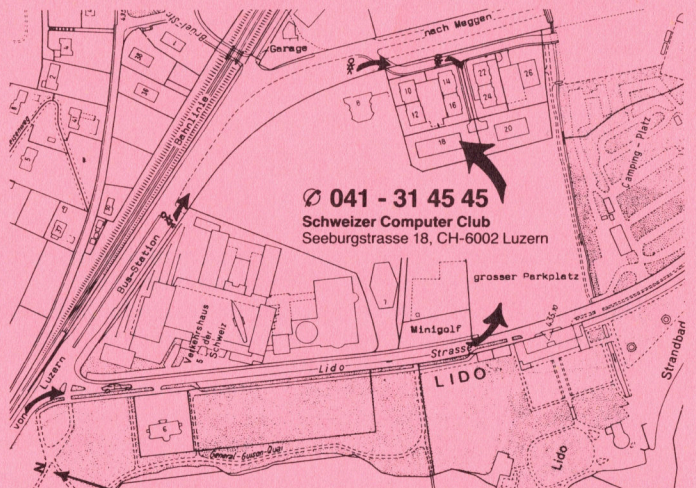
Montag bis Freitag am Nachmittag  
von 13.30 bis 18.00 Uhr

**Samstag von 8.30 bis 11.45 Uhr**

Telefonauskunft:

Dienstag bis Freitag jeweils  
zwischen 9.00 und 11.00 Uhr

P 2091		Speichererweiterung für Pet 2001 kl. Tastatur auf 32 K inkl. Einbau	750.—
P 2042		Grüner Filter zu PET-Bildschirm	25.—
P 2090		CBM Rekorder	195.—
P 2101		Kabel PET-IEEE	100.—
P 2102		Kabel IEEE-IEEE	115.—
P 9113		RTTY Interface mit Software	240.—
P 9898		CBM Basic Umrüstsatz Basic 1 auf Basic 2	240.—
P 9899		CBM Basic Umrüstsatz Basic 2 auf Basic 4	240.—
P 9900		CBM Floppy Umrüstsatz Dos. 1.0 auf Dos. 2.0	240.—
P 9920	Neu	<b>Basic Compiler Rom 3001</b>	<b>700.—</b>
P 9921	Neu	<b>Basic Compiler Rom 4001</b>	<b>750.—</b>
P 9922	Neu	<b>Basic Compiler Rom 8001</b>	<b>800.—</b>
P 9901		BASIC-Toolkit ROM (TK 80 P) 8 K PET alt	185.—
P 9902		BASIC-Toolkit ROM (TK 160 ) Serie 3001	125.—
P 9907		BASIC-Toolkit ROM (TK 8000) für 8000 Serie	206.—
P 9908		<b>BASIC-Toolkit TK 4000 für Serie 4000</b>	<b>206.—</b>
P 2065	Neu	<b>CBM HRG, High Resolution Graphic zu 8032</b>	<b>1475.—</b>
P 9950		<b>Hochauflösende Grafik mit Einbau (HRG)</b>	<b>1280.—</b>
P 9985		<b>UTILITY EPROM</b>	<b>158.—</b>
P 9986		<b>HRG Soft (Eprom)</b>	<b>194.—</b>
P 3512		CBM Newtim SV20 Monitor Assembler Editor	400.—
		Bei Bestellung unbedingt CBM-Type angeben!	
P 9988		<b>Monjana ROM zu PET</b>	<b>198.—</b>
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	1300.—
		<b>diverse Interface vorrätig</b>	



Sie finden den SCC in einem modernen Neubau in der Nähe des Verkehrshauses der Schweiz. Fahren Sie bitte mit der Buslinie Nr. 2 ab Bahnhof Luzern bis Haltestelle Verkehrshaus oder Brühlstrasse. Wenn Sie mit dem eigenen Wagen kommen, benutzen Sie bitte den grossen Parkplatz (ganz am Ende) gegenüber der Badeanstalt Lido, direkt hinter dem Verkehrshaus. **Wir freuen uns auf Ihren Besuch.**

# Schweizer Computer Club ☎ 041 - 31 45 45



# Einen Computer zum Spielen, zum Studieren, für die Buchhaltung, zum Textverarbeiten, für das Lager, zum Planen und Forschen? Also einen Commodore.



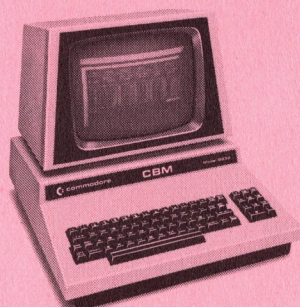
## Commodore Volksc Computer VC20

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, Maschinensprache 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, Maschinensprache 6502



## Commodore CBM 8032

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