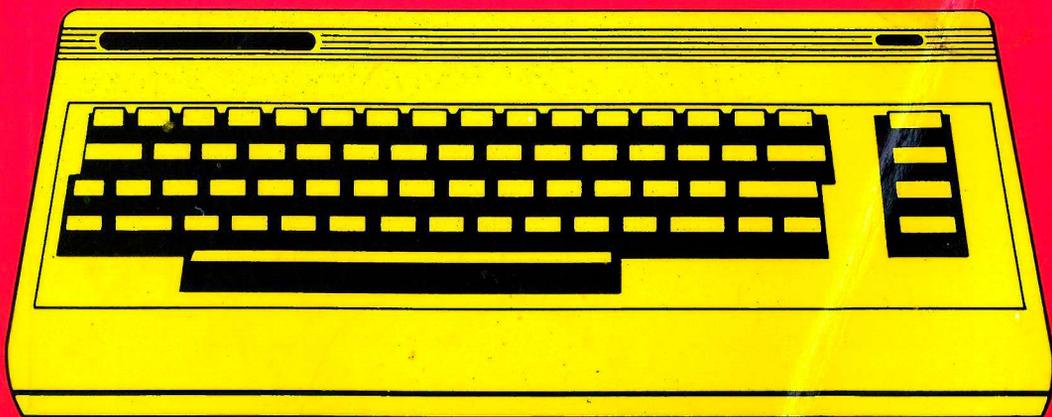


W. Hofacker

**PROGRAMME  
FÜR  
VC 20**

**Spiele  
Utilities**

**ERWEITERUNGEN**



Es kann keine Gewähr dafür übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen, sowie Programmlistings frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben auch keinen Aufschluß über eventuelle Verfügbarkeit oder Liefermöglichkeit. In jedem Falle sind die Unterlagen der Hersteller zur Information heranzuziehen.

Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen verboten. Programmlistings dürfen weiterhin nicht in irgendeiner Form vervielfältigt oder verbreitet werden. Alle Programmlistings sind Copyright der Fa. Ing. W. Hofacker GmbH. Verboten ist weiterhin die öffentliche Vorführung und Benutzung dieser Programme in Seminaren und Ausstellungen. Irrtum, sowie alle Rechte vorbehalten.

**COPYRIGHT by Ing. W. HOFACKER © 1982**

Tegernseer Str. 18, 8150 Holzkirchen / Obb.

1. Auflage 1982

Gedruckt in der Bundesrepublik Deutschland — Printed in West-Germany —  
Imprime'en RFA.

## Vorwort

Lieber Leser !

Mit dem Kauf eines VC-20 von Commodore haben Sie sicher eine kluge Entscheidung getroffen. Wie wir jedoch aus unserer jahrelangen Erfahrung auf diesem Gebiet wissen, besteht ein großes Interesse der Anwender, über das mitgelieferte Handbuch hinaus, sofort weitere Informationen zu erhalten.

Dieses Buch hat sich zur Aufgabe gemacht Sie mit vielen Tricks, Tips, Anleitungen zum Ausbau Ihres Systems und vor allem mit guten Programmen zu versorgen. Wie immer haben wir neben vielen Spielen auch ernsthafte Dinge wie Wortprozessor, Speichererweiterung, Ein-/Ausgabe-Programmierung u. v. a. mehr für Sie bereitgestellt. Alle Programme wurden bei uns sorgfältig getestet.

Wir wünschen Ihnen mit diesem Buch und Ihrem Computer viel Spaß bei der Programmierung und sind sicher, daß sich die gewonnenen Erfahrungen für Sie bald auch in beruflicher Hinsicht positiv auswirken werden.

Holzkirchen, Frühjahr 1982

Winfried Hofacker

## Quellenverzeichnis

1. Personal Computing on the VIC-20. A friendly Computer Guide. Published by Commodore International Ltd. Computer Systems Division 950 Rithenhouse Road Norristown, PA 19403, USA
2. Programmierhandbuch PET, Hofacker Verlag / Holzkirchen
3. Creative Software P. O. Box 4030, Mountain View, CA 94040
4. ELCOMP – Die Fachzeitschrift für Microcomputer Hofacker Verlag / Holzkirchen
5. VIC Memory Map, Kernal and Microsoft BASIC von Jim Butterfield, Canada and Andy Finkel / USA
6. United Microware Industries, Pomona California
7. Programmieren in Maschinensprache mit 6502 von E. Flögel Hofacker-Verlag / Holzkirchen

Dieses Buch ist eine Produktion des Ing. W. Hofacker GmbH Verlages. Die Produktion erfolgte unabhängig von Commodore Business Machines und ihren weltweiten Niederlassungen.

VC-20 ist ein eingetragenes Warenzeichen der Fa. Commodore

# Inhaltsverzeichnis

ELCOMP-Wortprozessor .....	1
Einfaches Zahlenratespiel .....	9
Reaktionszeit-Test. ....	11
Rechentest mit VC-20 .....	39
TIC-TAC-VIC. ....	16
Luftkrieg (mit Ton) .....	20
Ganzzahlige Teiler .....	24
Schießbude .....	26
U-Boot-Jagd .....	30
Eine Uhr für Ihren VC-20 .....	33
Uhr mit Wecker für den VC-20 .....	36
Programmieren in Maschinensprache .....	39
Befehlsliste der 6502 CPU .....	40
Bitmusterverteilung der 6502 CPU .....	41
Übernahme von Maschinenprogrammen in ein BASIC-Programm .....	47
Die Funktionsweise der USR(X)-Funktion .....	51
HEX/DEC Conversion .....	53
ASCII-Tabelle .....	54
Zweier Potenzen .....	54
Ein einfacher Monitor für VC-20 .....	55
Basisumrechnung mit dem VC-20 .....	62
Universal Experimentierplatine für VC-20 .....	67
VC-20 Speichererweiterung .....	74
Erweiterung des VC-20 um weitere 8K RAM/ROM oder EPROM .....	82
Die 16 K Byte RAMROM-Karte nach dem Bytewide-Konzept .....	85
Ein-/Ausgabe-Programmierungsexperimente mit dem VC-20 .....	104
Ein- und Ausgabeprogrammierung II .....	114
Logische Symbole .....	119
Code für Widerstände .....	119
Code für Kondensatoren .....	119
Ausgabeinterface für Netzverbraucher mit VC-20 .....	120
Anwenderport (USER I/O-Port) .....	121
Tricks für Ihren Volkscomputer VC-20 .....	127

Joystickprogrammierung mit dem VC-20 .....	127
Lesen des Joysticks .....	129
Dual Joystick für den VC-20 .....	135
Erstellung eines Interface für den Dualjoystick .....	137
Adressen für Zeichen am Bildschirm .....	144
Adressen für Farben am Bildschirm .....	144
Farbenkombination von Bildschirm und Bildschirmrand .....	145
PEEKs and POKES .....	145
Tabelle für Musiknoten .....	146
Lautsprecher-Kommando .....	146
BASIC TOKENS .....	147
RS-232 Register .....	148
Speicherbelegung des VC-20 .....	149
VC-20 MEMORY MAP KERNAL und Microsoft BASIC .....	150
Unterroutinen des Betriebssystems .....	158
ANHANG Interessante und praktische Produkte für unsere Leser .	159

# ELCOMP- Wortprozessor

## ELCOMP-Wortprozessor

Der nachfolgend aufgelistete Wortprozessor benötigt einen VC-20 mit 3K Speichererweiterung, zwischen 0400 Hex und 0FFF Hex. Für Text verbleiben dann noch ca. 3,5K Speicher. Hier können Sie etwa bis zu zwei Schreibmaschinenseiten ablegen.

Der Wortprozessor ist Menuegesteuert und erlaubt folgende Funktionen:

- EDITIEREN
- LADEN VON TEXT VON CASSETTE
- SPEICHERN VON TEXT AUF CASSETTE
- AUSDRUCKEN AUF BILDSCHIRM ODER DRUCKER (VC1515), AUF DEVICE 4 GESTELLT

Ein Wortprozessor ist nur sinnvoll in Verbindung mit einem Drucker. Ohne Drucker dürfte es Probleme geben.

Nach dem Einlesen des Programmes starten Sie mit RUN. Zur Eingabe von Text geben Sie N ein. Jetzt können Sie ohne Rücksicht auf die Zeilenlänge Ihren Text hintereinander eingeben. Achten Sie bitte darauf, daß zuerst die SHIFT-LOCK-Taste nicht gedrückt ist und nach Drücken von N (also bei der Eingabe von Text die SHIFT-LOCK-Taste gedrückt sein muß. Ansonsten erscheinen Grafikzeichen. Man kann diese jedoch immer mit der DEL-Taste wieder löschen. Wenn Sie die Eingabe beendet haben, lösen Sie die SHIFT-LOCK-Taste und drücken Sie jetzt für die Rückkehr ins Menü die £ (Pfund) Taste. Jetzt können Sie den eingegebenen Text mit "E" editieren oder den Text unter einem Namen auf Cassette ablegen. Die Editierfunktion erlaubt

das Korregieren, Einfügen und Umschreiben. Drucken Sie jetzt einmal die Taste. Der Cursor steht nun am Anfang des Textes. Sie können diesen Cursor jetzt mit Hilfe der Cursorsteuerung an jede beliebige Stelle im Text bringen. Wenn Sie an das Ende des Textes fahren, merken Sie, daß nach dem der Cursor das letzte Zeichen des gesamten Textes verlassen hat, die Frage ENDE und FERTIG gestellt wird. Wollen Sie jetzt etwas an den Text einfügen antworten Sie mit "N" für Nein. Wollen Sie wieder zurück ins Menü antworten Sie mit "J" für Ja.

Sie haben jetzt die Möglichkeit an den vorangegangenen Text neuen Text anzufügen.

Wollen Sie direkt an den bereits im Computer befindlichen Text neuen Text anschließen, so fahren Sie mit dem Cursor bis zum letzten Zeichen des letzten 170-Zeichen Blockes. Dann gehen Sie mit der Cursor-Vorwärts-Taste über dieses Zeichen hinaus. Sofort erscheint auf dem Bildschirm die Frage ENDE—FERTIG. Antworten Sie jetzt mit "N" und der Bildschirm wird gelöscht. In der linken oberen Ecke erscheint unser blinkender Cursor und wir können jetzt exakt an den vorangegangenen Text anschließen. Vergessen Sie die SHIFT-LOCK-Taste nicht ! Ist der Text fertig eingegeben, kehren wir mit £ wieder zurück ins Menü und können uns mit B den Text auf dem Bildschirm ansehen. Mit D können wir den Text mit einer gewünschten Zeilenlänge auf dem Bildschirm oder auf dem Drucker ausgeben.

Im Edit-Modus kann bereits im Computer befindlicher Text korregiert werden. Wir starten vom Menü aus mit E und fahren mit dem Cursor (hier wirkt sich die Dauerfunktion positiv aus) an die zu korregierende Stelle im Text. Beachten Sie bitte, daß in diesem Modus immer nur 170 Zeichen maximal angezeigt werden. Mit Cursor vorwärts jedoch erreichen Sie auch die nächsten 170-Zeichen-Blöcke. Die SHIFT-LOCK-Taste ist hierbei gelöst. An der zu korregierenden Stelle drucken wir die Taste mit dem Pfeil nach oben. Die Stelle wird jetzt negativ dargestellt und am Ende des Textes erscheint der blinkende Cursor. Sie geben jetzt Ihre Änderung hier ein. Achtung bei der Eingabe des neuen Textes muß die SHIFT-LOCK-Taste wieder gedrückt sein. Nach Eingabe der Korrektur muß die SHIFT-LOCK-Taste wieder gelöscht werden.

Wurde nur ein Zeichen eingegeben, ist das alte Zeichen überschrieben.

Werden mehrere Zeichen eingegeben, ist das anfänglich negativ gezeichnete Zeichen verschwunden und die zusätzlich eingegebenen Zeichen sind eingefügt worden. Wollen Sie die restlichen alten Zeichen löschen, gehen Sie mit dem Cursor an die Stelle und löschen diese mit der DEL-Taste. Ins Menü können Sie wieder zurück, wenn Sie mit der Cursor-Taste so lange nach unten gehen, bis die Frage Ende auftaucht und Sie mit "J" wieder an den Anfang gehen. Mit B können Sie sich jetzt den korrigierten Text ansehen.

Es dauert eine Weile, bis man mit dem Wortprozessor vertraut ist. Üben Sie daher zuerst an einigen kleinen Beispielen.

### **Speichern von Text auf Cassette:**

Geben Sie im Menü "S" ein. Drucken Sie eine beliebige Taste und geben Sie einen Filenam ein (nicht mehr als 5 Buchstaben). Dann folgt die Frage für diese Text. Jetzt brauchen Sie nur noch irgend eine Taste zu drücken und den Recorder mit Play und Record zu starten.

### **Laden von Textfiles:**

Mit L aus dem Menü wird ein vorher abgespeichertes Textfile wieder eingeladen. Auf dem Bildschirm erscheint der Filename.

↑ = Pfeil nach oben erlaubt bei der Texteingabe die Bildung von Paragraphen. Wenn möglich keine Worte trennen!

Ausdruck 55 Zeichen pro Zeile

### **BESCHREIBUNG**

DER EINFACHE WORTPROZESSOR VON ELCOMP IST NATUERLICH KEIN WORDSTAR, ABER FUER DASS WAS ER KOSTET UND DIE HARDWARE, DIE MAN DAZU BENOETIGT, LEISTET ER ERSTAUNLICHES. BEINAHE DAS WAS MAN VOR EINIGEN JAHREN NUR FUER VIELE HUNDERTAUSEND DEUTSCHE MARK KAUFEN KONNTE. DER ELCOMP WORTPROZESSOR KENNT NUR ZWEI DIREKTIVEN, DEN PFEIL NACH OBEN UND DAS PFUND ZEICHEN RECHTS OBEN AN DER TASTATUR. LADEN SIE DIE CASSETTE IN IHREN VC 20 WIE GEWOHNT UND STARTEN SIE MIT RUN. ES ERSCHEINT EIN MENUE, UM EINGABEN IN DAS MENUE ZU

MACHEN, MUSS DIE SHIFT LOCK TASTE GELOEST SEIN. WENN SIE TEXT EINGEBEN WOLLEN, MUSS DIE SHIFT LOCK TASTE GEDRUECKT SEIN. IN DIESER VERSION IST NUR GROSSSCHREIBUNG MOEGLICH. BEI ZAHLEN UND SATZZEICHEN MUSS DIE SHIFT LOCK TASTE GELOEST WERDEN. DER PROZESSOR FASST IHREN TEXT IMMER IN BLOECKE VON 170 ZEICHEN AUF. WENN SIE AENDERN WOLLEN, KOENNEN SIE WAEREND DER R TEXTEINGABE DIE DEL TASTE VERWENDEN. WENN SIE GERADE AM ENDE EINES BLOCKES SIND KANN DIE DEL TASTE NICHT IN DEN VOHERGEHENDEN BLOCK ZURUECKGEFUEHRT WERDEN. SIE MUESSEN DANN IM EDIT MODE DIESEN FEHLER AENDERN. IM EDIT MODE, IN DEN SIE UEBER DEN BEFEHL E AUS DEM MENUE EINSPRINGEN KOENNEN, HABEN SIE DIE CURSORSTEUERUNG FUER IHR FEHLERKORREKTUR ZUR VERFUEGUNG. SIE FAHREN MIT DEM CURSOR AN DIE GEWUENSCHTE STELLE UND DRUECKEN DIE TASTE "PFEIL NACH OBEN". DANN SPRINGT DER CURSOR AUTOMATISCH AN DAS ENDE DES TEXTES UND SIE KOENNEN DANN AB DORT EINFUEGEN UND UEBERSCHREIBEN. NACH DRUECKEN DES PFUND ZEICHENS WIRD DER GEAENDERTE TEXT DANN EINGEFUEGT. ES BENOETIGT EINIGE UEBUNG SICH MIT DEN EIGENSCHAFTEN DES WORTPROZESSORS VERTRAUT ZU MACHEN. AM BESTEN IST ES SIE PROBIEREN EINEN PROBETEXT. VOR DEM RUECKSPRUNG INS MENUE MUSS NATURLICH VOR DEM DRUECKEN DES PFUNDZEICHENS IMMER DIE SHIFT LOCK TASTE GELOEST WERDEN. NACH DEM RUECKSPRUNG SEHEN SIE AUCH WIEVIEL SPEICHERPLATZ SIE NOCH FUER WEITEREN TEXT ZUR VERFUEGUNG HABEN. BEI DER ANZEIGE IM EDITIERMODE SEHEN SIE WIE UNS DER COMPUTER IMMER NUR DIE 170 ZEICHEN TEXT ZUR GLEICHEN ZEIT ANZEIGEN KANN. MIT HILFE DER CURSORSTEUERUNG UND DER DAUERFUNKTIONEN LAESST SICH DOCH AUF EINFACHE WEISE DER GESAMTE TEXT ANSEHEN. WENN EINE AENDERUNG ERFOLGEN SOLL FAHREN SIE MIT DEM CURSOR ZUM ZEICHEN UND GEBEN DEN PFEIL EIN. IST DIE KORREKTUR ERFOLGT, MUSS ZUERST DIE SHIFT LOCK TASTE GELOEST WERDEN, DAMIT DIE PFUND TASTE AKTIV WIRD. WIR DRUECKEN DANN DIE PFUND TASTE, WELCHE UNS WIEDER IN DEN NORMALEN EDITIERMODUS ZURUECKBRINGT. VON DORT AUS KOENNEN WIR JETZT ALTEN TEXT MIT DEL LOESCHEN ODER MIT DEM CURSOR ZUR NAECHSTEN AENDERNG FORTFAHREN. DIESER TEXT WURDE MIT DEM WORTPROZESSOR GESCHRIEBEN UND EDITIERT.

```

10 DIM T$(32):Z$=""
100 PRINT"34 WORTPROZESSOR"
120 PRINT"BITTE WAERHLEN"
122 PRINT
125 PRINTSPC(5)"EDITIEREN"
130 PRINTSPC(5)"LADEN VON CASS."
140 PRINTSPC(5)"SPEICHERN A.CASS."
145 PRINTSPC(5)"AUSDRUCKEN"
147 PRINTSPC(5)"ABBILDSCHIRMAUSGABE"
150 X$="ODER NEU BEGINNEN":GOSUB290:IFC$="E"GOTO2000
160 IFC$="L"GOTO5000
170 IFC$="S"GOTO6000
180 IFC$="A"GOTO7000
190 IFC$="N"GOTO3000
195 IFC$="B"GOTO100
200 PT=0:LL=20:K=40:N=K:S$="":GOTO7110
290 PRINTX$+"?":GOTO310
300 PRINT"MIT LEERTASTE WEITER"
310 POKE198,0:WAIT198,1:GETC$:PRINT"3":RETURN
400 X$="":J=0
410 GETC$:IFC$=" "THENPRINT"CHR$(157)" "CHR$(157)":GOTO410
420 IFC$="E"THENRETURN
430 C=ASC(C$):IFC>64ANDC<91THENC=C+256
440 IFC>127THENC=C-128
450 L=LEN(X$):IFC=130RC=170RC=180RC=290R(C=20ANDL=0)GOTO410
460 IFC=20GOTO520
490 C$=CHR$(C):PRINTC$:X$=X$+C$:IFC=34THENPRINTC$" "
500 IFL<170GOTO410

```

BESCHREIBUNG  
 DER EINFACHE WORTPROZESSOR  
 VON ELCOMP IST NATUERLICH  
 KEIN WORDSTAR, ABER FUER DASS  
 WAS ER KOSTET UND DIE  
 HARDWARE, DIE MAN DAZU  
 BENOETIGT, LEISTET ER  
 ERSTAUNLICHES. BEI NAHE DAS WAS  
 MAN VOR EINIGEN JAHREN NUR  
 FUER VIELE HUNDERTAUSEND  
 DEUTSCHE MARK KAUFEN KONNTE.

Ausdruck 30 Zeichen pro Zeile

```

510 RETURN
520 IFL=1THENX$="":GOTO550
540 X$=LEFT$(X$,L-1)
550 PRINTCHR$(C):GOTO410
600 I=0:REM EDITING LOGIC
610 PRINT"J":IFKTHENPRINT"J"RIGHT$(T$(K-1),20)
620 PRINT:PRINTX$:LL=LEN(X$)-1
630 GETC:L=7724+I:POKEL,PEEK(L)+128:IFC$="^"GOTO750
640 IFC$="|"THENI=I+1
645 IFC$="J"ANDI>21THENI=I-22
650 IFC$="M"THENI=I+22
655 IFC$="N"THENI=0
660 IFC$="":GOTO690
670 U=ASC(C$):IFU=13THENI=191
675 IFU=20THENGOSUB700:GOTO610
680 IFC$="|"THENI=I-1:IFI<0THENI=0
690 POKEL,PEEK(L)-128:IFI>LLTHENRETURN
695 FORJ=0TO9:NEXTJ:GOTO630
700 L$="":IFI THENL$=LEFT$(X$,I)
710 R$="":IFI<LL THENR$=RIGHT$(X$,LL-I)
720 X$=L$+R$:RETURN
750 GOSUB700:GOSUB400
760 IFLEN(L$)+LEN(X$)+LEN(R$)<171THENX$=L$+X$+R$:GOTO610
770 FORJ=32TOK+2STEP-1:T$(J)=T$(J-1):NEXTJ
780 J=LEN(R$)+LEN(X$)-170:IFJ>0THENL$=L$+LEFT$(X$,J):X$=RIGHT$(X$,LEN(X$)-J)
790 T$(K+1)=X$+R$:X$=L$:GOTO610
800 POKE37148,252:POKE162,0:WAIT162,16:RETURN
2000 REM

```

```

2100 FORK=0:T031:IFT$(K)>" "THENX$=T$(K):GOSUB600:T$(K)=X$:NEXTK:GOTO100
2200 PRINT"ZUGENDE":X$="AFERTIG?":GOSUB290:IFC$="J"GOTO100
2210 GOTO3010
3000 K=0
3010 X$=T$(K):GOSUB400:T$(K)=X$:IFC$<>"E"ANDK<31THENK=K+1:GOTO3010
3020 FORJ=K+1:T032:T$(J)="":NEXTJ:GOTO100
5000 PRINT"DATENBAND EINLEGEN"
5002 PRINT"IRGEND EINE TASTE DRUECKEN"
5004 POKE198,0:WAIT 198,1:GETZX$
5008 CLR:OPEN:PRINT"LESEN: ";
5010 DIM T$(32):Z$="":T$=" "
5020 INPUT#1,X$:PRINT"X$ "
5030 GET#1,C$:IFASC(C$)<>13THEN$(I)=T$(I)+C$:GOTO5030
5050 IFT$(I)<>"E"THENI=I+1:T$(I)="":GOTO5030
5060 CLOSE:T$(I)="":GOTO100
6000 REM
6002 PRINT"MCASSETTE EINLEGEN"
6004 PRINT"BELIEBIGE TASTE WENN FERTIG"
6006 POKE198,0:WAIT198,1:GETZX$
6010 INPUT"FILENAME":L$:X$="AFUER DIESEN TEXT"
6020 GOSUB290:IFC$="J"THENSAVE"ELCOMP"
6030 GOSUB800:OPEN:1,1,L$:GOSUB800:PRINT#1,L$:GOSUB800
6040 FORI=0:T031:IFT$(I)>" "THENPRINT#1,T$(I):GOSUB800:NEXTI
6050 I=0:PRINT#1,"E":GOSUB800:CLOSE1:GOTO100
7000 REM
7010 X$="LEERRAUME IN EINEM TAB":GOSUB290
7015 T$="  ":IFVAL(C$)THEN$=LEFT$(Z$,VAL(C$))
7020 PT=0:X$="AUF BILDSCHIRM ODER DRUECKER":GOSUB290:IFC$="D"THENPT=2

```

```

7030 DS=0: X$="DOPPELTER ZEILENABST": GOSUB290: IFC$="J" THEN DS=1
7040 N=22*(1+PT): IFPTTHENOPEN4, 4: GOT07060
7050 INPUT "ZEILEN/SEITE": ILL
7060 INPUT "ZEICHEN/ZEILE": K: PRINT "J"
7100 N=(PT+1)*22: IFK>N THEN PRINT "ZU VIELE ZEICHEN": GOT07060
7105 S$="": IFK<N-1 THEN S$=LEFT$(Z$, (N-K)/2)
7110 TF=0: L=1: X$="": FOR J=0 TO 31: X$=X$+T$(J)
7115 IF TF THEN X$=T$+X$: TF=0
7120 M=LEN(X$): IFASC(X$)=32 THEN X$=MID$(X$, 2): GOT07120
7125 IF J<31 AND M<K THEN NEXT J
7130 FOR U=1 TO K: IF MID$(X$, U, 1)="^" THEN TF=1: GOT07150
7135 NEXT U: IF J=31 AND M<K THEN GOT07250
7140 FOR U=K TO 1 STEP -1: IF MID$(X$, U, 1)<>" " THEN NEXT U: PRINT "ZU WENIG": GOT07060
7150 L$="": IF U>1 THEN L$=LEFT$(X$, U-1)
7160 R$="": IF U<M THEN R$=RIGHT$(X$, M-U)
7170 IF DS AND (PT=0) THEN L$=L$+CHR$(13)
7180 IF DS AND PT THEN L$=L$+CHR$(12)
7190 X$=R$: IFPT THEN PRINT#4, S$+L$: GOT07115
7200 PRINTS$+L$: L=L+1+DS: IFL>LL THEN L=1: GOSUB300
7210 GOT07115
7250 IFPT THEN PRINT#4, S$+X$: CLOSE4: GOT0100
7260 PRINTS$+X$: GOSUB300: GOT0100
7300 END
8000 OPEN 4, 4
8010 PRINT#4, "HELLO"
8020 CLOSE 4

```

READY.

# Einfaches Zahlenratespiel

## Einfaches Zahlenratespiel

Das nachfolgende Programm gehört zu den Standardprogrammen, die jeder BASIC-Anfänger bei seinen ersten "Gehversuchen" durchprobiert und dann selbst ausbaut.

Der Computer "denkt" sich eine Zahl, die dann vom VC-20 Besitzer geraten werden muß. Nach Eingabe Ihrer geratenen Zahl, erhalten Sie einen Tip über das Verhältnis der geratenen Zahl und der Zahl, die sich der Computer "ausgedacht" hat.

```
100 REM *** NUMMER *** ZAHLENRATESPIEL*
110 REM *** COPYRIGHT: ING.W.HOFACKER *
120 REM *** TEGERNSEERSTR. 19
125 REM *** D-8150 HOLZKIRCHEN/OBB. **
200 REM *** PRINT INSTRUCTION ON HOW
205 REM *** TO PLAY
210 PRINT "ICH DENKE MIR EINE"
215 PRINT "ZAHL ZWISCHEN"
217 PRINT "1 UND 100"
220 PRINT "VERSUCH MEINE ZAHL ZU"
223 PRINT "ERRATEN. NACH JEDEM "
225 PRINT "VERSUCH WERDE ICH DIR"
227 PRINT "SAGEN, OB ES RICHTIG"
240 PRINT "WAR ODER NICHT, ZU"
245 PRINT "HOCH ODER ZU NIEDRIG
300 REM*** COMPUTER DENKT SICH
305 REM*** EINE ZAHL X
310 LET X=INT(100*RND(1))+1
320 PRINT
330 PRINT "OK, ICH HABE EINE ZAHL, "
335 PRINT "FANGE AN ZU RATEN"
```

```
400 REM *** DER MENSCH BEGINNT ZU
405 REM *** RATEN
410 PRINT
420 PRINT "WAS RATEST DU";
430 INPUT G
440 IF G=X THEN 500
450 IF G>X THEN 480
460 PRINT "ZU NIEDRIG, VERSUCH EINE"
465 PRINT " GROESSERE ZAHL"
470 GOTO 410
480 PRINT "ZU HOCH, VERSUCH EINE"
485 PRINT " NIEDRIGERE ZAHL"
490 GOTO 410
500 REM *** DER MENSCH HAT DIE ZAHL
505 REM *** DES COMPUTERS ERRATEN
510 PRINT
520 PRINT "RICHTIG, WILLST DU NOCH"
525 PRINT "EINMAL RATEN?"
530 PRINT
540 GOTO 300
999 END
```

READY.

# Reaktionszeit-Test

## Reaktionszeit-Test Der Party-Knüller

Dieser lustige Reaktionszeit-Test eignet sich bestens dazu auf einer Party, die Leistung Ihres Heimcomputers zu demonstrieren. Sicherlich läßt sich aus den Ergebnissen auch ein Rückschluß auf den Alkoholgehalt des Blutes und die Stimmung schließen.

Das Spiel um die Reaktionszeit kann von zwei Teilnehmern gleichzeitig ausgeführt werden. Der erste Spieler verwendet für seine Eingaben die A-Taste, während sich der 2. Spieler der L-Taste bedient. Erscheint auf dem Bildschirm ein ●, so muß der erste Spieler seine A-Taste drücken. Der 2. Spieler darf bei diesem Zeichen nicht reagieren. Er bekommt sonst Strafpunkte. Erscheint ein O auf dem Bildschirm, muß der 2. Spieler die L-Taste drücken. Jetzt darf der 1. Spieler seine Taste nicht drücken.

Zwischendurch erscheint manchmal ein Cursorzeichen "▣". Jetzt sind beide Spieler zum Drücken ihrer Taste aufgefordert. Wer seine Taste zuerst gedrückt hat, bekommt die Punkte.

```
1 PRINT"▣"  
10 PRINT"REAKTIONSZEITTEST"  
20 PRINT"FUER ZWEI SPIELER"  
30 PRINT"DER 1. SPIELER BENUTZT"  
40 PRINT"DIE 'A'TASTE, DER 2.  
50 PRINT"SPIELER DIE 'L' TASTE  
60 PRINT"ERSCHEINT AUF DEM BILD-"  
70 PRINT"SCHIRM EIN'●', SO MUSS"
```

```

72 PRINT"DER ERSTE SPIELER DIE"
73 PRINT" A -TASTE DRUECKEN"
75 FOR V=1 TO19000:NEXT
76 PRINT"J"
80 PRINT"IST ES EINE 'O'MUSS"
90 PRINT"DER ZWEITE SPIELER
100 PRINT"SEINE TASTE DRUECKEN"
110 PRINT"BEI VERWECHSLUNG GIBTS"
120 PRINT"2 MINUSPUNKTE"
130 PRINT"MANCHMAL WIRD EIN'※"
140 PRINT"ANGEZEIGT"
141 PRINT"WER JETZT ZUERST SEINE"
142 PRINT"TASTE DRUECKT ERHAELT"
143 PRINT"EINEN PUNKT"
144 PRINT"WER ZUERST 15 PKTE.
145 PRINT"HAT,GEWINNT"
150 PRINT"※DRUECKE EINE TASTE ZUM STARTEN DES SPIELES"
160 GETA$: IFA$=""GOTO160
165 PRINT"J"
170 GOSUB200:REM DELAY
180 GOSUB300:REM PUT UP TARGET
190 GOSUB400:REM SCORE
197 GOTO170
200 REM DELAY SUBROUTINE
210 FORI=1TORND(2)*60+50:A=SIN(25):NEXT
220 RETURN
300 I=INT(RND(2)*20+1):A=INT(RND(2)*21+1)
310 PRINT"※";:FORI=1TOD:PRINT"※";:NEXT
320 FORI=1TOA:PRINT"||";:NEXT
330 T=INT(RND(2)*3+1)
335 GETA$: IFA$<>""THENGOTO335:REM CLR BUFFER
340 IFT=1THENPRINT"●";
350 IFT=2THENPRINT"○";
360 IFT=3THENPRINT"※";
370 RETURN
400 XM=TI
401 IFTI-60>XMTHENGOTO460
402 REM
403 REM ZEIT
405 GETA$: IFA$=""THEN GOTO401
410 IFT=1AND A$="A"THENS1=S1+1
415 IFT=1AND A$="L"THENS2=S2-2
420 IFT=2AND A$="L"THENS2=S2+1

```

```

425 IFT=2AND A$="A" THEN S1=S1-2
430 IFT=3AND A$="A" THEN S1=S1+1
435 IFT=3AND A$="L" THEN S2=S2+1
460 FOR I=1 TO 20: X=SIN(2): NEXT
470 GET B$: IF B$<>" " THEN GOTO 470: REM CLR BUFFER
480 PRINT "PLAYER #1  PLAYER #2": PRINT " "; S1; "          "; S2
481 PRINT " ";: FOR I=1 TO 10: PRINT " ";: NEXT
482 FOR I=1 TO 8: PRINT " ";: NEXT
483 PRINT " ";: REM ERASE TARGET
490 RETURN

```

READY.

# Rechentest mit VC-20

## Rechentest mit VC-20

Dieses kleine Programm ermöglicht es den VC-20 als Lehrer einzusetzen, der niemals Ungeduld an den Tag legen wird. Je nach Wahl kann das Malnehmen oder das Zusammenzählen geübt werden. Bei richtigen Antworten erfolgt ein Lob, bei mehreren falschen Antworten hilft der Computer mit der Bekanntgabe des Ergebnisses etwas nach.

```
1 REM RECHENTESTPROGRAMM
10 FOR I=1 TO 80 : J=SIN(2) : NEXT
11 PRINT "DAS IST EIN RECHENTEST"
15 PRINT
16 PRINT "BIS JETZT HABEN SIE";R;"RICHTIG UND";W;"FALSCH"
17 PRINT
20 V=0
30 I=0
35 Z=0
40 PRINT "(1)FUER MULTIPLIKATION"
50 PRINT
60 PRINT "(2)FUER ADDITION"
70 PRINT
80 INPUT I
90 PRINT
100 IF I=1 GOTO 200
110 IF I=2 GOTO 350
120 IF I=0 GOTO 500
130 GOTO 600
200 LET X=INT(RND(12)*12+1)
205 LET Y=INT(RND(12)*12+1)
210 IF X<10 THEN PRINT " ";X
220 IF X>=10 THEN PRINT " ";X
```

```

260 IF Y<10THEN PRINT" X";Y
270 IF Y>=10THEN PRINT"X";Y
300 PRINT"_____"
310 LET Q=X*Y
320 INPUT D
330 GOTO 120
350 X=INT(RND(50)*50+1)
355 Y=INT(RND(50)*50+1)
360 IFX<10THENPRINT" ";X
370 IFX>=10THENPRINT" ";X
410 IFY<10THENPRINT"+ ";Y
420 IFY>=10THENPRINT"+";Y
450 PRINT"_____"
460 Q=X+Y
470 INPUTD
480 GOTO120
500 PRINT"DU HAST RECHT!!!":R=R+1
505 PRINT
508 Z=Z+1
509 IFZ<3GOTO512
510 GOTO10
512 IFI=1GOTO200
514 IFI=2GOTO350
600 PRINT"FALSCH,NOCH EINMAL...":W=W+1
610 PRINT
620 V=V+1
630 IFV=3GOTO650
640 IFI=1GOTO210
645 IFI=2GOTO360
650 PRINT"DIE RICHTIGE ANTWORT IST";Q
655 FORJ=1TO50:JX=SIN(2):NEXT
660 PRINT
670 GOTO10

```

READY.

# TIC – TAC – VIC

## TIC-TAC-VIC

TIC-TAC-VIC ist eine TIC-TAC-TOE Version für den VC-20 von Commodore. Geben Sie das Programm in Ihren Computer ein und achten Sie peinlich auf alle Leerzeichen und Cursorsteuer-Symbole. Stimmt die Eingabe nicht bis auf das letzte Steuerzeichen mit dem Listing überein, gibt es Probleme.

Dies liegt in erster Linie an der unsinnigen Steuerzeichenhandhabung des Commodore-BASIC. Andere BASIC-Versionen verwenden hier SET oder DRAWTO-Befehle. Hier kommt man mit herkömmlichen Zeichen aus und muß nicht identifizieren, ist das nun Cursor nach links, rechts oder gar ein Graphikzeichen ?

Also aufgepasst bei der Eingabe !

Starten Sie mit RUN. Der Computer fragt Sie ob Sie "X" oder "O" haben wollen. Wer X wählt beginnt mit der ersten Platzierung. Wenn Sie "O" wählen, beginnt der Computer das Spiel.

Ziel des Spieles ist es, drei Zeichen gleicher Art in eine Reihe zu bekommen (entweder waagrecht, senkrecht oder diagonal).

```
2 REM
3 PRINT"3":PRINT:PRINT
4 FORX=1TO6:PRINT"      3  3      3  3":NEXTX
5 PRINT"7      1  3  3      2  3  3      3"
6 PRINT"3
7 FORX=1TO6:PRINT"      3  3      3  3":NEXTX
8 PRINT"7      4  3  3      5  3  3      6"
9 PRINT"3
10 FORX=1TO6:PRINT"      3  3      3  3":NEXTX
11 PRINT"7      7  3  3      8  3  3      9"
50 PRINT"WILLST DU 'X' OR 'O'"
```

```

52 GETC$: IFC$="" THEN 52
55 IFC$="X" THEN 475
60 P$="0": Q$="X"
100 G=-1: H=1: IFS(5) <> 0 THEN 103
102 S(5)=-1: GOT0195
103 IFS(5) <> 1 THEN 106
104 IFS(1) <> 0 THEN 110
105 S(1)=-1: GOT0195
106 IFS(2)=1 AND S(1)=0 THEN 181
107 IFS(4)=1 AND S(1)=0 THEN 181
108 IFS(6)=1 AND S(9)=0 THEN 189
109 IFS(8)=1 AND S(9)=0 THEN 189
110 IF G=1 THEN 112
111 GOT0118
112 J=3*INT((M-1)/3)+1
113 IF 3*INT((M-1)/3)+1=M THEN K=1
114 IF 3*INT((M-1)/3)+2=M THEN K=2
115 IF 3*INT((M-1)/3)+3=M THEN K=3
116 GOT0120
118 FOR J=1 TO 7 STEP 3: FOR K=1 TO 3
120 IFS(J) <> G THEN 130
122 IFS(J+2) <> G THEN 135
126 IFS(J+1) <> 0 THEN 150
128 S(J+1)=-1: GOT0195
130 IFS(J)=H THEN 150
131 IFS(J+2) <> G THEN 150
132 IFS(J+1) <> G THEN 150
133 S(J)=-1: GOT0195
135 IFS(J+2) <> 0 THEN 150
136 IFS(J+1) <> G THEN 150
138 S(J+2)=-1: GOT0195
150 IFS(K) <> G THEN 160
152 IFS(K+6) <> G THEN 165
156 IFS(K+3) <> 0 THEN 170
158 S(K+3)=-1: GOT0195
160 IFS(K)=H THEN 170
161 IFS(K+6) <> G THEN 170
162 IFS(K+3) <> G THEN 170
163 S(K)=-1: GOT0195
165 IFS(K+6) <> 0 THEN 170
166 IFS(K+3) <> G THEN 170
168 S(K+6)=-1: GOT0195
170 GOT0450

```

```

171 IFS(3)=GANDS(7)=0THEN187
172 IFS(9)=GANDS(1)=0THEN181
173 IFS(7)=GANDS(3)=0THEN183
174 IFS(9)=0ANDS(1)=GTHEN189
175 IFG=-1THENG=1:H=-1:GOTO110
176 IFS(9)=1ANDS(3)=0THEN182
177 FORI=2TO9:IFS(I)<>0THEN179
178 S(I)=-1:GOTO195
179 NEXTI
181 S(1)=-1:GOTO195
182 IFS(1)=1THEN177
183 S(3)=-1:GOTO195
187 S(7)=-1:GOTO195
189 S(9)=-1
195 PRINT"DER COMPUTER GEHT NACH..."
202 GOSUB1000
205 GOTO500
450 IFG=1THEN465
455 IFJ=7ANDK=3THEN465
460 NEXTK,J
465 IFS(5)=GTHEN171
467 GOTO175
475 P$="X":Q$="0"
500 PRINT"WO WILLST DU HIN?"
501 GETC$:IFC$=""THEN501
502 M=VAL(C$)
505 PRINT"!"
506 IFM=0THENPRINT"DAS WAR KLUG! DANKE FUERS SPIEL ":
GOTO2001
507 IFM>9THEN509
508 IFS(M)=0THEN510
509 PRINT"DIESES FELD IST BELEGT":PRINT:PRINT:GOTO500
510 G=1:S(M)=1
520 GOSUB1000
530 GOTO100
1000 GOSUB3000
1095 FORI=1TO7STEP3
1100 IFS(I)<>S(I+1)THEN1115
1105 IFS(I)<>S(I+2)THEN1115
1110 IFS(I)=-1THEN1350
1112 IFS(I)=1THEN1200
1115 NEXTI:FORI=1TO3:IFS(I)<>S(I+3)THEN1150
1130 IFS(I)<>S(I+6)THEN1150

```

```

1135 IFS(I)=-1THEN1350
1137 IFS(I)=1THEN1200
1150 NEXTI:FORI=1TO9:IFS(I)=0THEN1155
1152 NEXTI:GOTO1400
1155 IFS(5)<>GTHEN1170
1160 IFS(1)=GANDS(9)=GTHEN1180
1165 IFS(3)=GANDS(7)=GTHEN1180
1170 RETURN
1180 IFG=-1THEN1350
1350 PRINT"ICH HABE GEWONNEN":GOTO2001
1400 PRINT"DAS WAR GUT,DANKE DIR"
2001 PRINT"NOCH EIN SPIEL?"
2002 PRINT"STARTEN BITTE MIT RUN"
2004 END
3000 FORI=1TO9
3005 PRINT"場":IFI<4THENAB=0:GOTO3040
3010 IFI>6THENAB=14:GOTO3040
3020 AB=7
3040 FORX=1TOAB+1:PRINT:NEXTX
3050 IFI=10RI=4ORI=7THENBA=0:GOTO3071
3060 IFI=20RI=5ORI=8THENBA=7:GOTO3071
3070 BA=14
3071 IFP$="X"THEN3080
3072 IFS(I)=-1THENGOSUB4020
3073 IFS(I)=1THENGOSUB5020
3074 NEXTI:RETURN
3080 IFS(I)=-1THENGOSUB5020
3090 IFS(I)=1THENGOSUB4020
3095 NEXTI:RETURN
4020 PRINTTAB(BA)"  ▾ ▾ ▾"
4050 PRINTTAB(BA)"  ▾ ▾"
4060 PRINTTAB(BA)"  ▾"
4070 PRINTTAB(BA)" ▾ ▾ ▾ ▾ ▾"
4080 PRINTTAB(BA)"  "
4090 RETURN
5020 PRINTTAB(BA)" ▾ ▾"
5050 PRINTTAB(BA)" ▾ ▾ ▾ ▾"
5060 PRINTTAB(BA)" ▾ ▾ ▾ ▾"
5070 PRINTTAB(BA)" ▾ ▾"
5090 RETURN

```

READY.

# Luftkrieg (mit Ton)

## Luftkrieg (mit Ton)

Dieses Spiel kann von zwei Spielern über die Tasten A und 0 gespielt werden.

Verschiedene Flugkörper überqueren das Spielfeld in unterschiedlichen Höhen. Der Abschluß erfolgt durch rechtzeitigen und gezielten Tastendruck. Der rechte Spieler genießt einen kleinen Vorteil. Daher sollte in bestimmten Abständen der Platz gewechselt werden.

Der linke Spieler verwendet zum Abfeuern eines Schusses die A-Taste und der rechte Spieler die Null-Taste. Das Spiel ist in BASIC und arbeitet daher etwas langsam. Führt man jedoch die Bewegungen und die Tonerzeugung in Maschinensprachenprogramme über, läßt sich noch einiges aus diesem Programm herausholen.

Ziel des Programmes ist es, dem Anfänger zu demonstrieren, wie er seine eigenen Spielprogramme schreiben kann.

```
3 REM ELCOMP MICROCOMPUTER
4 REM*MAGAZIN*
8 PRINT"J";
10 PRINT"*FLUGZEUGABSCHIESSEN*"
15 PRINT"4 VERSCHIEDENE ZIELE KOENNEN ZU"
16 PRINT"VERSCH.ZEITEN IN VERSCH.HOEHEN ER-"
17 PRINT"SCHEINEN,JE NACH HOEHE FEuern DIE"
18 PRINT:PRINT"KANONEN UNTERSCHIEDLICH SCHNELL."
20 PRINT:PRINT"SPIELER-DRUECKT A UM DIE LINKE KANONE"
25 PRINT"ZU FEuern.DER 2.SPIELER DRUECKT 0 FUER"
26 PRINT:PRINT"DIE RECHTE KANONE,WER 21 HAT GEWINNT!!":PRINT
30 PRINT:PRINT"BELIEBIGE TASTEN DRUECKEN!"
35 GETZ$:IFZ$=""GOTO35
90 PRINT"J":"ELCOMP (R) LUFTKRIIEG":PRINT:AS=0:BS=0
```

```

91 FOR I=1TO19:PRINT:NEXTI
92 PRINT"  *";TAB(17);"*"
93 PRINTTAB(3);"——";TAB(16);"——"
94 PRINT"A=FEUER";TAB(14);"0=FEUER"
95 PRINT"☺"
100 X=INT(RND(1)*10)+3:S=0
105 X3=7702+X*22
110 FORI=1TOX:PRINT"☺";:NEXTI
115 X1=INT(RND(1)*4)+1
120 ON X1 GOSUB 200,300,400,460,460
125 PRINT"☺"
130 GOTO100
200 FORI=1TO18
201 : IFS=1GOTO230
205 : PRINT"☺  1  11111111";
210 : PRINT"——☺  11111111";:E=4
211 : IFA$="A"GOTO220
212 : IFB$="0"GOTO220
213 : T=TI
215 : IFTI-T<16-XGOTO215
220 : PRINT"  1  1";:X3=X3+1
225 : GOSUB800
230 NEXTI
240 PRINT"  11111111  "
250 RETURN
300 FORI=1TO18
301 : IFS=1GOTO330
305 : PRINT"☺  11111111";
310 : PRINT"——☺  11111111";:E=4
311 : IFA$="A"GOTO320
312 : IFB$="0"GOTO320
313 : T=TI
315 : IFTI-T<15-XGOTO315
320 : PRINT"  1  1";:X3=X3+1
325 : GOSUB800
330 NEXTI
340 PRINT"  11111111  "
350 RETURN
400 FORI=1TO18
401 : IFS=1GOTO430
405 : PRINT"☺  11111111";
410 : PRINT"——☺  11111111";:E=4
411 : IFA$="A"GOTO420

```

```

412 : IFB$="0"GOTO420
413 : T=TI
415 : IFTI-T<15-XGOTO415
420 : PRINT" 7";:X3=X3+1
425 : GOSUB800
430 NEXTI
440 PRINT"      "
450 RETURN
460 FORI=1TO18
461 : IFS=1GOTO490
465 : PRINT" ";
468 FOR Q=1 TO 20:NEXTQ
470 : PRINT"->7";:E=2
471 : IFA$="A"GOTO480
472 : IFB$="0"GOTO480
473 : T=TI
475 : IFTI-T<5-XGOTO475
480 : PRINT" 7";:X3=X3+1
485 : GOSUB800
490 NEXTI
495 PRINT"      "
496 RETURN
600 IFZ$="A"THENA$="A"
610 IFZ$="0"THENB$="0"
620 RETURN
700 REM FEUERROUTINE
701 P=INT(RND(1)*3)+1
702 FORD=1TOP
705 : IFA$<"A"GOTO720
706 : IFY=0GOTO708
707 POKE38822,2:POKE8102-Y,32
708 POKE38800,2:POKE8080-Y,43:Y=Y+22
709 : IF8102-Y-X3>E0GOTO715
710 : IF8102-Y-X3>=0 GOTO900
715 IFY>418 THEN A$="":POKE38800,2:POKE8102-Y,32:Y=0
718 NEXTD
720 P=INT(RND(1)*3)+1
722 FORD=1TOP
723 : IFB$<"0"GOTO760
735 : IFZ=0GOTO750
740 : POKE8115-Z,32
750 : POKE8093-Z,43:Z=Z+22
752 : IF8115-Z-X3>E0GOTO755

```

```

753 : IF8115-Z-X3>=0 GOTO950
755 : IFZ>418 THEN B$="":POKE8115-Z,32:Z=0
759 NEXTD
760 RETURN
800 IF A$="A" THEN GOSUB700
805 IF B$="0" THEN GOSUB700
810 GETZ$: IFZ$="" THEN RETURN
820 IFZ$="A" THEN A$="A": GOSUB7000: RETURN
830 IFZ$="0" THEN B$="0": GOSUB7002: RETURN
840 RETURN
900 REM TARGET HIT
910 PRINT"*****XXXXXXXXXXXXXXXXX": GOSUB7005
920 A$="1": AS=AS+1: Y=0: S=1: POKE59467,0
921 PRINT"XXXXXXXXX ";
925 PRINT"X": PRINTTAB(0): "TREFFER": AS
930 IFAS=21 THEN N=9: GOTO1000
940 RETURN
950 PRINT"*****XXXXXXXXXXXXXXXXX": GOSUB7005
960 B$="1": BS=BS+1: Z=0: S=1: POKE59467,0
961 PRINT"XXXXXXXXX ";
965 PRINT"X": PRINTTAB(11): "TREFFER": BS
970 IFBS=21 THEN N=24: GOTO1000
975 RETURN
1000 PRINT"X";
1005 FORI=1 TO10
1010 : PRINTTAB(N): "SIEGER": T9=TI
1015 : IFTI-T9<50 GOTO1015
1016 : PRINTTAB(N): "XXXXXXXXX2SIEGERXXXXXXXXX";
1020 NEXTI
1025 PRINT
1030 GOTO90
7000 POKE36878,15: POKE36876,220: FORL=1 TO5: NEXTL
7001 POKE36876,0: FORL=1 TO500: NEXTL: POKE36878,0: RETURN
7002 POKE36878,15: POKE36876,200: FORL=1 TO5: NEXTL
7003 POKE36876,0: FORL=1 TO500: NEXTL: POKE36878,0
7004 RETURN
7005 POKE36877,220
7006 FORL=6 TO0 STEP-1: POKE36878,L: FORM=1 TO12
7007 NEXTM: NEXTL: POKE36877,0
7008 POKE36878,0: RETURN
7010 IFTI-T9<30 GOTO7010
7015 RETURN

```

# Ganzzahlige Teiler

## Ganzzahlige Teiler

Dieses kleine Programm ist als Hilfe für die Hausaufgaben der Schüler gedacht. Es ermittelt aus einer Reihe von Zahlen die ganzzahligen Teiler.

Beispiel:

Durch Eingabe von 40 und 200 durch Komma getrennt (40,200) ermittelt es alle ganzzahligen Teiler der Zahlen von 40 bis einschließlich 200.

```
0 DIM F(32)
1 GOSUB 50
2 IF (10000-P)*(P-1)<0 THEN 7
3 IF P<>INT(P) THEN 7
4 IF (10000-Q)*(Q-1)<0 THEN 7
5 IF Q<>INT(Q) THEN 7
6 GOTO 8
7 GOTO 1
8 IF Q=>P THEN 12
9 LET P9=P
10 LET P=Q
11 LET Q=P9
12 PRINT
13 PRINT
14 FOR J=P TO Q
15 LET F=0
16 LET M=0
17 LET R=SQR(J)
18 PRINT
19 PRINT J; TAB(8);
20 FOR K=1 TO R
21 LET X=J/K
```

```

22 IF X<>INT(X) THEN 34
23 LET G= LEN(STR$(K))+2
24 IF M+G<=64 THEN 28
25 PRINT
26 PRINT TAB(8);
27 LET M=0
28 PRINT K;
29 LET M=M+G
30 IF X<>R THEN 32
31 GOTO 35
32 LET F=F+1
33 LET F(F)=X
34 NEXT K
35 FOR K=F TO 1 STEP -1
36 LET G=LEN(STR$(F(K)))+2
37 IF M+G<=64 THEN 41
38 PRINT
39 PRINT TAB(8);
40 LET M=0
41 PRINT F(K);
42 LET M=M+G
43 NEXT K
44 PRINT
45 NEXT J
46 PRINT
47 PRINT
48 PRINT
49 GOTO 1
50 PRINT" :DIESES PROGRAMM BE-"
60 PRINT"RECHNET DIE GANZ-      ZÄHLIGEN"
70 PRINT"TEILER VON DEN EIN-
80 PRINT"GEgebenEN WERTEN"
90 PRINT"EINGABE IN EINER ZEILE"
92 PRINT"DURCH KOMMA GETRENNT"
100 INPUT P,Q
101 IF P=0 THEN END
110 RETURN

```

READY.

# Schießbude

## Schießbude

Dies ist ein lustiges Abschießspiel mit Ton. Links im Bild bewegt sich eine Abschlußbasis ständig von unten nach oben. Rechts baut sich eine Mauer mit bis zu drei Löchern auf.

Ziel ist es, mit dem Ball in die Löcher in der Mauer zu treffen. Ein Treffer bringt 100 Punkte. Man startet mit 100 Punkten und kann 5 Spiele mit je 10 Schuß absolvieren. Schießt man daneben, werden 10 Punkte abgezogen. Am Schluß werden alle erreichten Punkte ausgegeben. Will man noch eine Runde spielen, gibt man "Ja" ein. Mit "Nein" verlassen Sie das Programm.

Bitte auch hier unbedingt auf alle Cursorsteuerungen im Listing achten. Fehlt nur ein Zeichen, arbeitet das Programm nicht.

```
100 REM SCHIESSBUDE
110 REM
120 REM                               ING.W.HOFACKER
130 REM
190 REM NACH PET USER NOTES MONTGOMMERYVILLE USA
200 CLR
210 DIM Z(23)
220 PRINT":ZUM START BITTE"
225 PRINT:PRINT"IRGEND EINE TASTE"
230 PRINT"DRUECKEN" PRINT"DRUECKE 1 FUER FEUER"
240 PRINT"MICROCOMP MICROCOMPUTER          MAGAZIN"
250 GETO$: IFO$=""GOTO250
260 PRINT"
270 A$(0)="
280 A$(1)="
290 D$(0)="X"
```

```

300 D$(2)=" "
310 G$(0)="§":G$(1)=" "
320 POKE59467,0
330 PRINT"§"
340 T=T+S
350 K=1:X=1:S=100:C=10:R=R+1
360 R$=STR$(R)
370 IFR=5GOTO1040
380 FORI=0TO 18:PRINT" ":Z(I)=0:NEXT
390 FORI=1TO3
400 Z(INT(RND(1)*17+2))=1
410 NEXT
420 PRINT"§"
430 FORI=1TO19
440 PRINTTAB(19);G$(Z(I))
450 NEXT
460 PRINT" §RUNDE §COUNT §PUNKTE■
470 PRINT" ";R$;" 10 100 "
490 PRINT"§"
500 K=1
510 K=ABS(K-1):X=1
520 I=1
530 PRINTA$(K)
540 GET A$
550 IFA$="1"THENGOSUB640
560 IFC=0GOTO330
570 I=I+1
580 IFI=19GOTO510
590 GOTO530
600 POKE36877,220
610 FORL=6 TO0STEP-1:POKE36878,L:FOR M=1TO12
615 NEXTM:NEXTL:POKE36877,0
617 POKE36878,0
620 RETURN
640 PRINT"§§§";
650 Y=18
660 IFK=0THENW=I+2:GOTO680
670 W=20-I
680 IFZ(W)>0THENY=20
690 PRINT"§§ §";
700 X=X+1
710 IFX<>YGOTO690
720 IFX=18THEN PRINT"§§ §§§";GOSUB600:GOTO 840

```

```

730 Z(W)=Z(W)+1
740 FORJ=1T04
750 PRINT "||50||";
760 POKE59467,16
770 POKE59464,150:POKE59466,31
780 FORM=1T020:NEXT
790 PRINT "||0";
820 NEXT
825 GOSUB1300
830 S=S+120
840 S=S-5*(20-X)
850 POKE59467,0
860 S=S-Z(W)*10
870 C=C-1
880 FORJ=1T022-W
890 PRINT
900 NEXT
910 IFS<0THENS=0
920 S#=STR$(S):C#=STR$(C)
930 PRINTTAB(8)"T";C#;TAB(15);RIGHT$(S#,3)
940 PRINT "X"
950 FORJ=1TOW:PRINT "X";:NEXT
960 PRINT "T#####";D$(20-X);
970 IFX<20GOTO1010
990 POKE 59464,67
1000 POKE59466,35
1010 POKE59467,0
1020 X=1:PRINT "|| ||T"
1030 RETURN
1040 PRINT "T"
1050 POKE59467,16
1060 POKE59466,1
1070 PRINT "X000"
1080 FORJ=1T08
1090 PRINTTAB(3);"** SPIEL IST AUS"
1100 FORH=1T03
1110 POKE59464,INT(RND(1)*50+110)
1120 NEXT
1130 PRINTTAB(3);"**SPIEL IST AUS**T"
1140 FORH=1T03
1150 POKE59464,INT(RND(1)*50+40)
1160 NEXT
1170 NEXT

```

```
1180 PRINT:PRINT
1190 FORH=1TO39:PRINT"◆";:NEXT:PRINT
1200 POKE59467,0:POKE59466,0
1210 PRINT"MIT 40 SCHUSS AUF 3 "
1215 PRINT"ZIELE"
1220 PRINT"DIE GESAMTPUNKTZAHL "
1225 PRINT"IST";T;"PUNKTE"
1260 POKE59467,0:POKE59466,0
1270 PRINT"NOCH EINMAL?"
1280 INPUT A$
1290 IF A$="JA" THEN 200
1295 END
1300 REM KLINGEL
1310 POKE 36878,15
1320 FOR L=1TO5:FOR M=1TO5 :POKE 36876,220
1330 FOR N=1TO5: NEXT N:POKE 37876,0:NEXT M
1340 FOR M=1 TO 10 :NEXT M:NEXT L
1350 POKE 36878,0
1360 RETURN
2000 PRINT"!"
2010 FORL=1TO11:PRINT"":NEXT
```

READY.

# U-Boot-Jagd

## U-Boot-Jagd

Das Programm U-Boot-Jagd arbeitet mit der Cursorsteuerung Ihres VC-20 (die beiden Tasten rechts unten im Tastenfeld).

Nach dem Start des Programmes mit RUN erscheint ein Kästchen mit danebenstehender Punktzahl auf dem Bildschirm. Die Aufgabe des Spielers ist es so schnell wie möglich, mit dem Cursor (Ball auf dem Bildschirm) in die Mitte dieses Kästchens zu fahren. Ist der Ball in der Mitte des Kästchens (U-Boot) angekommen, erfolgt eine Explosion. Die Zeit wird links oben im Bild angezeigt. Wer in der vorgegebenen Zeit von 120 Sekunden, die meisten Punkte erreicht hat, hat gewonnen. Dieses Spiel arbeitet mit Ton.

Wer besonders gute Ergebnisse erzielt erhält einen Zeitbonus. D. h. er kann noch länger Punkte sammeln. Auch dieses Programm kann vom Leser erweitert und ausgebaut werden. Z. B. als Ziel besonders schöne Grafikgebilde und Tonausgaben bei der Cursorbewegung in verschiedene Richtungen usw.

```
1 REM U-BOOT JAGD-ING W HOFACKER GMBH
50 GOSUB11000:REM ERLAEUTERUNGEN
90 REM
99 AT=120
100 GOSUB1000:REM FELDAUFBAU
105 CX=3:CY=2:PRINT"3";
110 PRINT"110000";
200 GOSUB2000:REM UHR
300 GOSUB3000:REM ZIELOBJEKT
400 GOSUB4000:REM CURSORBEWEGUNG
500 GOSUB5000:REM TREFFER
600 GOTO300:REM WIEDERHOLUNG BIS ZUM ENDE
1000 PRINT"7";
```



```

4600 OK=1:M$="II ●":DX=1:DY=0:RETURN
4800 OK=1:M$="II II●":DX=0:DY=-1:RETURN
5000 REM
5005 PRINT"II";
5010 PRINT"IIII  IIII  XXXIII  ■";
5020 FORI=1TO20:X=SIN(2):NEXT
5025 GOSUB 5500
5030 PRINT"TTTTII  XXXII  XXXII  XXXXXXXII  ";REM
      LOESCHEN
5040 SC=SC+DS
5045 PRINT"§";
5050 FORI=1TOCX-1:PRINT"II":NEXT
5060 FORI=1TOCY:PRINT"X":NEXT
5070 PRINT"●";
5100 RETURN
5500 REM EXPL
5510 POKE36877,220
5520 FORL=6TO0STEP-1:POKE36878,L:FORM=1TO12
5525 NEXTM:NEXTL:POKE36877,0:POKE36878,0
5530 RETURN
6000 PRINT"XXXXXXXXXXXXXXXXSPIEL VORBEI NOCH EINMAL";
6100 INPUTYN$:IFLEFT$(YN$,1)="Y"THENGOTO90
6210 END
10000 REM BONUS
10100 AT=180
10110 PRINT"XXXXXXXXXXXXXXXXBONUS"
10115 PRINT"§";
10120 RETURN
11000 PRINT"J":PRINT"U-BOOT JAGD EINFACH"
11010 PRINT"DU ERHAELST PUNKTE,
11020 PRINT"WENN DU IN DIE MITTE"
11025 PRINT"DES ZIELES TRIFFST."
11030 PRINT"BEWEGE DAS ZIELGERAET"
11040 PRINT"MIT DEN CURSORTASTEN."
11050 PRINT"ANFANGS ERHAELST DU"
11060 PRINT"120 SEKUNDEN ZEIT."
11070 PRINT"DU MUSST ABER AUCH DEIN ZIEL RICHTIG TREFFEN."
11080 PRINT"UEBER 5000 PUNKTE GIBT ES EINEN"
11090 PRINT"BONUS VON 30 SEKUNDEN"
11100 PRINT"DRUEKE IRGEND EINE TASTE,  WENN DU ZUM SPIELEN
      BEREIT BIST...."
11110 GETA$:IF A$=""THEN GOTO 11110
11120 RETURN

```

# Eine Uhr für Ihren VC-20

## Eine Uhr für Ihren VC-20

Das nachfolgende Programm verwandelt Ihren VC-20 in eine Uhr, die Ihnen Stunden, Minuten und Sekunden auf dem Bildschirm anzeigt.

Das besondere an dieser Uhr ist, daß Sie auch noch ein Tickzeichen über den Lautsprecher ausgibt. Achten Sie bitte bei der Eingabe der Grafik- und Steuerzeichen in den Zeilen 1000 – 1010 auf die jeweilige Bedeutung.

Die Uhr wird gestellt, indem die Stunden, Minuten und Sekunden als sechsstellige Zahl eingegeben werden. Z. B.

195352

für 19 Uhr dreiundfünfzig und zweiundfünfzig Sekunden. Auch dieses Programm kann noch ausgebaut und verbessert werden. Z. B. Trennung der einzelnen Blöcke durch Darstellung in verschiedenen Farben. Stunden rot, Minuten blau und Sekunden grün mit einem Punkt dazwischen zur Trennung.

Wurde die Uhr bereits gestellt, kann als Ersatz für die momentane Zeitangabe einfach die "RETURN"-Taste gedrückt werden.

```
1 DIMA(6):DIMB(6):GOSUB3000
2 PRINT"0";
4 FORI=1TO6:B(I)=-1:NEXT
10 FORI=1TO6:A(I)=VAL(MID$(TI$,I,1)):NEXTI
30 FORM=1TO6
35 PRINT"#####";
40 IFB(M)=A(M)THENGOTO100
50 PRINTTAB((M-1)*3+1):GOSUB1010:REM ERASE
```

```

55 PRINT"XXXXXXXXXXXXXXXXX";
60 PRINTR((M-1)*3+1);
70 B(M)=R(M):X=R(M)
99 ON(X+1)GOSUB1009,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009
100 NEXTM
101 GOSUB 4000
200 GOTD10
1000 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1001 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1002 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1003 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1004 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1005 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1006 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1007 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1008 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1009 PRINT"XXXXXXXXXXXXXXXXX":RETURN
1010 FORI=1TO7:PRINT"XXXXXXXXX":NEXT:RETURN
3000 REM CLOCK SET ROUTINE
3010 PRINT"XXXXXXXXX DAS IST DIE VIC UHR
3020 PRINT"XDIE UHRZEIT WIRD ALS"
3030 PRINT"6-STELLIGE DIGITALZAHL"
3035 PRINT"EINGEGEBEN.WENN SIE
3036 PRINT"DIE UHR SCHON GESTELLT
3040 PRINT"HABEN,GEBEN SIE RETURN";
3045 PRINT"XDWIEVIEL UHR IST ES"
3050 INPUT N
3055 IFN=0THENGOTO3070

```

```
3056 IFN<100000THENTI$="0"+RIGHT$(STR$(INT(N)),5)
3060 IFN>=100000THENTI$=RIGHT$(STR$(INT(N)),6)
3070 RETURN
4000 REM CLOCK
4001 POKE36878.15
4002 REM FOR L=1TO10
4003 POKE36875.200
4004 FOR M=1TO10
4005 NEXT M
4006 POKE36875.0
4007 FOR M=1TO 300
4008 NEXT M
4009 POKE36878.200
4010 FOR M=1TO 10
4020 NEXT M
4030 POKE36874.0
4040 FOR M=1TO300
4050 NEXT M
4060 REM NEXTL
4070 POKE36878.0
4080 RETURN
```

READY.

# Uhr mit Wecker für den VC-20

## Uhr mit Wecker für den VC-20

Dieses Programm ähnelt dem Programm "Eine Uhr für Ihren VC-20". Es ist jedoch noch um eine Weckeinrichtung erweitert worden.

Nach dem Programmstart werden Sie gefragt, ob Sie auch einen Wecker wollen. Antworten Sie jetzt mit "Ja", können Sie die Zeit eingeben, zu der Sie geweckt werden wollen. Danach fragt der Wecker nach der momentanen Uhrzeit. Falls Sie diese noch nicht vorher eingegeben haben, müssen Sie jetzt die Zeit eingeben. Wenn die Zeit einmal eingegeben ist, kann auf die Frage "Wieviel Uhr ist es" mit "RETURN" geantwortet werden.

```
1 DIMA(6):DIMB(6):GOSUB3000
2 PRINT"?";
4 FORI=1TO6:B(I)=-1:NEXT
10 FORI=1TO6:A(I)=VAL(MID$(TI$,I,1)):NEXTI
30 FORM=1TO6
35 PRINT"#####";
40 IFB(M)=A(M)THENGOTO100
50 PRINTTAB((M-1)*3+1):GOSUB1010:REM ERASE
55 PRINT"#####";
60 PRINTTAB((M-1)*3+1);
70 B(M)=A(M):X=A(M)
```

```

99 ON(X+1)GOSUB1009,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009
100 NEXT M
101 GOSUB 4000
200 GOT010
1000 PRINT " ":RETURN
1001 PRINT " ":RETURN
1002 PRINT " ":RETURN
1003 PRINT " ":RETURN
1004 PRINT " ":RETURN
1005 PRINT " ":RETURN
1006 PRINT " ":RETURN
1007 PRINT " ":RETURN
1008 PRINT " ":RETURN
1009 PRINT " ":RETURN
1010 FOR I=1 TO 7:PRINT " ":NEXT I:RETURN
3000 PRINT " ":GOTO 5000
3010 PRINT "DRS IST DIE VIC UHR"
3020 PRINT "DIE UHRZEIT WIRD ALS"
3030 PRINT "6-STELLIGE DIGITALZHL"
3035 PRINT "EINGEGEBEN.WENN SIE"
3036 PRINT "DIE UHR SCHON GESTELLT"
3040 PRINT "HABEN,GEBEN SIE RETURN";
3045 PRINT "WIEVIEL UHR IST ES"
3050 INPUT N
3055 IF N=0 THEN GOTO 3070
3056 IF N<10000 THEN I$="0"+RIGHT$(STR$(INT(N)),5)
3060 IF N>=10000 THEN I$=RIGHT$(STR$(INT(N)),5)
3070 RETURN
4000 GOSUB 7000

```

```
4001 POKE36878,15
4002 REM FOR L=1TO10
4003 POKE36875,200
4004 FOR M=1TO10
4005 NEXT M
4006 POKE36875,0
4007 FOR M=1TO 300
4008 NEXT M
4009 POKE36878,200
4010 FOR M=1TO 10
4020 NEXT M
4030 POKE36874,0
4040 FOR M=1TO300
4050 NEXT M
4060 REM NEXTL
4070 POKE36878,0
4080 RETURN
5000 PRINT"WOLLEN SIE AUCH EINEN"
5010 PRINT"WECKER?"
5020 INPUT A$
5030 IF A$="JA" THEN 5050
5040 IF A$="NEIN" THEN3010
5050 PRINT"GEBEN SIE DIE ZEIT"
5060 PRINT"EIN, AN DER GEWECKT"
5070 PRINT"WERDEN SOLL SSMSS"
5080 PRINT"6-STELLIG"
5090 INPUT W$
5100 GOTO 3010
7000 IF TI$=W$ THEN 8000
7010 RETURN
8000 POKE36878,15
8010 FOR L=1 TO5
8020 FOR M=1 TO 50
8030 POKE36876,220
8040 FOR N=1TO5
8050 NEXT N
8060 POKE36876,0
8070 NEXT M
8080 FOR M=1 TO 3000
8090 NEXT M
8100 NEXT L
8110 POKE36878,0
8120 GOTO 3070
```

# Programmieren in Maschinensprache

## Programmieren in Maschinensprache

Im nachfolgenden Teil wollen wir Ihnen einen kleinen Überblick über die Möglichkeiten der Programmierung in Maschinensprache mit dem VC-20 geben. Wie Sie sicher aus den Erfahrungen, die Sie bei der Durcharbeitung der Kapitel Programmexperimente und Ein-/Ausgabeprogrammierung erfahren haben, stößt man irgend wann einmal an eine Grenze, wo BASIC einfach zu langsam ist. Sei es bei der Entwicklung einer eigenen Schnittstelle, bei der Abfrage externer Analog-Digitalwandler usw.

Natürlich können wir Ihnen im Rahmen dieses Büchleins keinen ausführlichen Kurs über Programmieren in Maschinensprache anbieten. Dazu gibt es spezielle Bücher, wie z. B. das Buch Nr 118 aus dem Hofacker Verlag.

Trotzdem wollen wir versuchen Ihnen die Grundlagen zu zeigen und das Interesse zu wecken. Vielleicht freunden Sie sich später doch mit diesem Gebiet noch etwas mehr an.

Der VC-20 enthält einen 6502-Microprozessor, jedoch standardmäßig keinen Maschinensprachenmonitor. Ein Maschinensprachen-Monitor ist die Grundvoraussetzung für die Programmierung in Maschinensprache (PEEK und POKE von BASIC her ist sehr mühsam).

Aus diesem Grunde haben wir in diesem Buch einen kompletten Maschinensprachen Monitor als Listing abgedruckt (siehe ab Seite 55).

Als Unterlagen für unsere jetzigen Experimente benötigen wir eine Liste mit allen 6502 Befehlen sowie wenn möglich ein Codierblatt für Maschinensprache und eine HEX-DEC, DEC-HEX Umrechnungstabelle.

Befehls	syml. Code	Wirkung	ADRESSIERUNGSARTEN																							
			UNM	ABS	ABS,X	ABS,Y	S0	S0,X	S0,Y	(IND,X)	(IND),Y	REL	IND	ACCU	IMPL	N	Z	C	I	D	V					
Transport	LDA	M → A	A9	AD	BD	B9	A5	B5		A1	B1										X	X	-	-	-	-
	LDX	M → X	A2	AE		BE	A6		B6												X	X	-	-	-	-
	LDY	M → Y	A0	AC	BC		A4	B4													X	X	-	-	-	-
	STA	A → M		8D	9D	99	85	95		81	91										-	-	-	-	-	-
	STX	X → M		8E			86		96												-	-	-	-	-	-
	STY	Y → M		8C			84	94													-	-	-	-	-	-
	TAX	A → X																	AA		X	X	-	-	-	-
	TAY	A → Y																	A8		X	X	-	-	-	-
	TXA	X → A																	BA		X	X	-	-	-	-
	TYA	Y → A																	98		X	X	-	-	-	-
	TXS	X → S																	9A		-	-	-	-	-	-
	TSX	S → X																	BA		X	X	-	-	-	-
	PLA	S+1 → S, Ms → A																	68		X	X	-	-	-	-
	PHA	A → Ms, S-1 → S																	48		-	-	-	-	-	-
	PLP	S+1 → S, Ms → P																	28		-	-	-	-	-	-
	PHP	P → Ms, S-1 → S																	08		-	-	-	-	-	-
	Arithmetische	ADC	A+M+C → A	69	8D	7D	79	65	75		61	71									X	X	X	-	-	-
SBC		A-M-C → A	E9	ED	FD	F9	E5	F5		E1	F1									X	X	X	-	-	-	X
INC		M+1 → M		EE	FE		E6	F6												X	X	-	-	-	-	-
DEC		M-1 → M		CE	DE		C6	D6												X	X	-	-	-	-	-
INX		X+1 → X																	E8		X	X	-	-	-	-
DEX		X-1 → X																	CA		X	X	-	-	-	-
INY		Y+1 → Y																	C8		X	X	-	-	-	-
DEY		Y-1 → Y																	88		X	X	-	-	-	-
Logische	AND	A ∧ M → A	29	2D	3D	39	25	35		21	31									X	X	-	-	-	-	-
	ORA	A ∨ M → A	09	0D	1D	19	05	15		01	11									X	X	-	-	-	-	-
	EOR	A ⊕ M → A	49	4D	5D	59	45	55		41	51									X	X	-	-	-	-	-
Vergleichs-	CMP	A-M	C9	CD	DD	D9	C5	D5		C1	D1									X	X	X	-	-	-	-
	CPX	X-M	E0	EC			E4													X	X	X	-	-	-	-
	CPY	Y-M	C0	CC			C4													X	X	X	-	-	-	-
	BIT	A ∧ M		2C			24													7	X	X	-	-	-	6
Verzweigungs-	BCC	BRANCH ON C=0											90							-	-	-	-	-	-	-
	BCS	BRANCH ON C=1											80							-	-	-	-	-	-	-
	BEQ	BRANCH ON Z=1											F0							-	-	-	-	-	-	-
	BNE	BRANCH ON Z=0											D0							-	-	-	-	-	-	-
	BMI	BRANCH ON N=1											30							-	-	-	-	-	-	-
	BPL	BRANCH ON N=0											10							-	-	-	-	-	-	-
	BVC	BRANCH ON V=0											50							-	-	-	-	-	-	-
	BVS	BRANCH ON V=1											70							-	-	-	-	-	-	-
JMP		4C																		-	-	-	-	-	-	
JSR		20																		-	-	-	-	-	-	
Schiebe-	ASL		0E	1E		06	16											0A		X	X	X	-	-	-	-
	LSR		4E	5E		46	56											4A		X	X	X	-	-	-	-
	ROL		2E	3E		26	36											2A		X	X	X	-	-	-	-
	ROR		6E	7E		66	76											6A		X	X	X	-	-	-	-
Status-Register-	CLC	C=0																		18	-	-	0	-	-	-
	CLD	D=0																		D8	-	-	-	0	-	-
	CLI	I=0																		58	-	-	0	-	-	-
	CLV	V=0																		88	-	-	-	0	-	-
	SEC	C=1																		38	-	-	1	-	-	-
	SED	D=1																		F8	-	-	-	1	-	-
	SEI	I=1																		78	-	-	1	-	-	-
Versch.	NOP	NO OPER																		EA	-	-	-	-	-	-
	RTS	RETURN F. SUB																		80	-	-	-	-	-	-
	RTI	RETURN F. INT																		40	-	-	-	-	-	-
	BRK	BREAK																		00	-	-	1	-	-	-

### Befehlsliste der 6502 CPU

LSD	0	1	2	3	4	5	6	7
MSD								
0	BRK	ORA-IND, X				ORA-Z PAGE	ASL-Z PAGE	
1	BPL	ORA-IND, Y				ORA-Z PAGE, X	ASL-Z PAGE	
2	JSR	AND-IND, X			BIT-Z Page	AND-Z PAGE,	ROL-Z PAGE	
3	BMI	AND-IND, Y				AND-Z PAGE, X	ROL-Z PAGE, X	
4	RTI	EOR-IND, X				EOR-Z PAGE	LSR-Z PAGE	
5	BVC	EOR-IND, Y				EOR-Z PAGE, X	LSR-Z PAGE, X	
6	RTS	ADC-IND, X				ADC-Z PAGE	ROR-Z PAGE	
7	BVS	ADC-IND, Y				ADC-Z PAGE, X		
8		STA-IND, X			STY-Z Page	STA-Z PAGE	STX-Z PAGE	
9	BCC	STA-IND, Y			STY-Z Page, X	STA-Z PAGE, X	STX-Z PAGE, Y	
A	LDY-IMM	LDA-IND, X	LDX-IMM		LDY-Z Page	LDA-Z PAGE	LDX-Z PAGE	
B	BCS	LDA-IND, Y			LDY-Z Page, X	LDA-Z PAGE, X	LDX-Z PAGE, Y	
C	CPY-IMM	CMP-IND, X			CPY-Z Page	CMP-Z PAGE	DEC-Z PAGE	
D	BNE	CMP-IND, Y				CMP-Z PAGE, X	DEC-Z PAGE, X	
E	CPX-IMM	SBC-IND, X			CPX-Z Page	SBC-Z PAGE	INC-Z PAGE	
F	BEQ	SBC-IND, Y				SBC-Z PAGE, X	INC-Z PAGE, X	

LSD	8	9	A	B	C	D	E	F
MSD								
0	PHP	ORA-IMM	ASL-A			ORA-ABS	ASL-ABS	
1	CLC	ORA-ABS, Y				ORA-ABS, X	ASL-ABS, X	
2	PSP	AND-IMM	ROL-A		BIT-ABS	AND-ABS	ROL-ABS	
3	SEC	AND-ABS, Y				AND-ABS, X	ROL-ABS, X	
4	PHA	EPR-IMM	LSR-A		JUMP-ABS	EOR-ABS	LSR-ABS	
5	CLI	EOR-ABS, Y				EOR-ABS, X	LSR-ABS, X	
6	PLA	ADC-IMM	ROR-A		JMP-IND	ADC-ABS	ROR-ABS	
7	SEI	ADC-ABS, Y				ADC-ABS, X		
8	DEY		TXA		STY-ABS	STA-ABS	STX-ABS	
9	TYA	STA-ABS, Y	TXS			STA-ABS, X		
A	TAY	LDA-IMM	TAX		LDY-ABS	LDA-ABS	LDX-ABS	
B	CLV	LDA-ABS, Y	TSX		LDY-ABS, X	LDA-ABS, X	LDX-ABS, Y	
C	INY	CMP-IMM	DEX		CPY-ABS	CMP-ABS	DEC-ABS	
D	CLD	CMP-ABS, Y				CMP-ABS, X	DEC-ABS, X	
E	INX	SBC-IMM	NOP		CPX-ABS	SBC-ABS	INC-ABS	
F	SED	SBC-ABS, Y				SBC-ABS, X	INC-ABS, X	

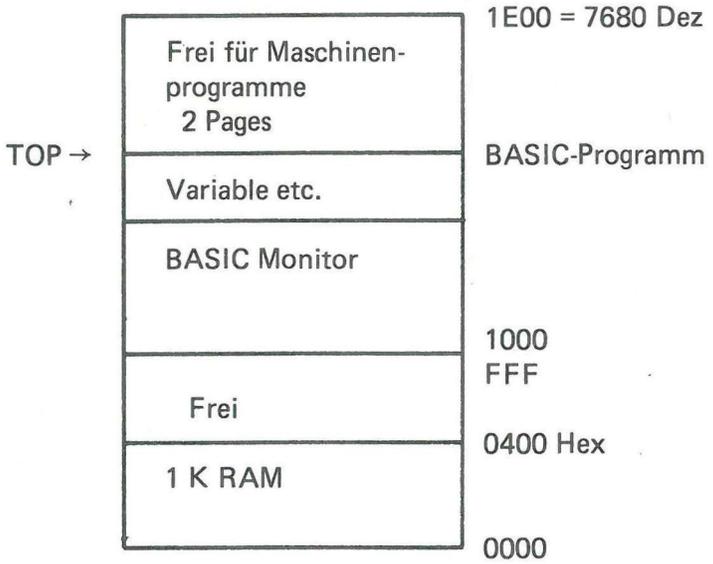
### Bitmusterverteilung der 6502 CPU

Wir wollen gleich mit einem Beispiel beginnen.

Geben Sie den Monitor in den VC-20 ein und testen Sie diesen.

Geben Sie jetzt PRINT FRE(0) im Direktmode ohne Zeilenzahl ein und notieren sich den freien Speicherplatz. In unserem Falle erhalten wir 986 Bytes frei.

Dies bedeutet, daß wir ca. 2 Pages = 510 Byte für unsere Versuche verwenden können. In der VC-20 Grundversion sieht unser Speicher jetzt wie folgt aus:



**Achtung !**

Wenn Sie eine Speichererweiterung 3 K zwischen 400 und FFF Hex eingesteckt haben ändern sich die Adressen. Ist oberhalb 2000 auch durch Speicher erweitert, sehen die Verhältnisse wieder anders aus.

Durch ?PEEK(55) und ?POKE(56) erfahren wir den oberen Punkt unseres BASIC Speicherbereiches.

$$55 = 0 \text{ Dez} = 00 \text{ Hex}$$

$$56 = 30 \text{ Dez} = 1E \text{ Hex}$$

Da die Darstellung wie immer in der Reihenfolge als Lower Byte/

Higher Byte erfolgt, liegt unsere oberste Speichergrenze bei 1E00 Hex = 7680 Dezimal.

Um den oberen Bereich von BASIC zu schützen, damit wir dort ein Maschinenprogramm ansiedeln können, muß die Grenze durch zwei POKE-Befehle in den Zellen 55 und 56 geändert werden.

Am günstigsten verfahren wir, wenn wir uns etwas an die Page-Grenzen halten (1 Page = 255 Byte) und uns in der Nähe von 500 Bytes unterhalb 7680 Dez an die nächstgelegene Pagegrenze legen.

Diese Page-Grenze ist: 1C00 = 7170 Dez.

Dies ergibt POKE 55,0 und POKE 56,28.

Für unsere BASIC-Variablen bleiben dann noch 425 Byte frei wobei uns 2 Pages = 510 Byte für unser Maschinenprogramm bleiben.

Wir geben jetzt folgendes Programm über den Monitor mit dem Befehl C ein:

Anfangsadresse:	1C10	A900	START	LDA 0
	1C12	A209		LDX 09
	1C14	9DFF1C	LOOP	STA \$1CFF,X
	1C17	CA		DEX
	1C18	10FA		BPL LOOP
	1C1A	60		RTS

7184 Dez = 1C10 / 7194 Dez = 1C1A

Nachdem Sie Ihr Programm eingegeben haben, sehen Sie sich einige Zellen ab 1CFF Hex einmal an. Diese sollten FF enthalten.

Unser kleines Programm soll nämlich ab Zelle 1CFF einige Zellen zu Null machen.

Starten Sie jetzt das Programm an Adresse 1C10 mit dem Befehl G1C10. Der Befehl RTS in Zelle 1C1A Hex bringt uns wieder in den Ausgangspunkt zurück. Geben Sie jetzt C ein, Anfangsadresse 1CFF RETURN und drücken Sie mehrmals die Leertaste. Und siehe da die Zellen 1CFF – 1D08 sind mit dem Wert 00 gefüllt.

Sie haben jetzt Ihr erstes Maschinenprogramm geschrieben und bereits erfolgreich getestet. Was macht eigentlich unser Programm im einzelnen:

Wir haben hier zuerst einmal den Akkumulator mit 00 geladen. LDA0. Dann haben wir das X-Register auf 09 gesetzt. jetzt laden wir die Nullen aus dem Akkumulator in die Speicherzelle 1CFF Hex. Der Befehl STA \$1CFF,X bietet die Möglichkeit der indizierten Adressierung. Der Wert X (Wert 09 im X-Register) wird vom Computer verwendet, um eine effektive Adresse zu berechnen. Der Inhalt des X-Registers wird deshalb zur Adresse 1CFF dazugezählt. So beginnt also der Akku zuerst seine ersten beiden Nullen in Zelle  $1CFF + 9 = 1D08$  abzulegen. Der DEX-Befehl zählt das X-Register um eins herunter und wir erhalten so die nächste effektive Adresse 1D07. So wird nun das X-Register bis auf 00 heruntergezählt. Dies geschieht über eine Schleife durch BPL. Hat das Register den Wert 00 überschritten, beginnt wieder FF. Diesen Wert sieht der 6502 Computer als negativ an und springt über den BPL-Test dann nach Zeile 1C1A Hex.

Der BPL-Befehl ist ein bedingter Testbefehl, der so lange eine Verzweigung bestehen läßt, solange ein positiver Wert dort vorhanden ist, wo wir durch BPL hinspringen.

Der Wert (Opcode) nach BPL ergibt sich aus der Entfernung. Er errechnet sich in unserem Beispiel wie folgt:

F1  
F2  
F3  
F4  
F5  
F6  
F7  
F8  
F9  
FA  
FB  
FC  
FD  
FE  
FF = entspricht Sprung 1 Zelle rückwärts



Von 1C1A bis 1C14 sind es insgesamt 6 Adressen. Von FF nach unten abgezählt, ergibt es den Wert FA. Er wird als Opcode hinter den BPL-Befehl gesetzt.

Der Befehl RTS (Return from Subroutine), Rückkehr aus dem Unterprogramm bringt uns wieder ins BASIC-Programm zurück. Wenn Sie einen BRK-Befehl noch vor den RTS-Befehl setzen, arbeitet zwar Ihr Programm, aber BASIC bringt die Fehlermeldung: “? ILLEGAL QUANTITY ERROR IN 10”.

Wenn Sie diese kleine Programmierübung bis jetzt durchgeführt haben, sollten Sie noch einige kleine Übungen durchführen.

1. Ändern Sie das Maschinenprogramm so, daß in die Zellen 1C10 bis 1D08 die Zahl 44 eingeschrieben wird.
2. Ändern Sie das Programm so, daß der Wert 1 in Zellen geschrieben wird, die zum eigentlichen 3,5K Arbeitsspeicherbereich des VC-20 gehören (1000 – 1BFF).

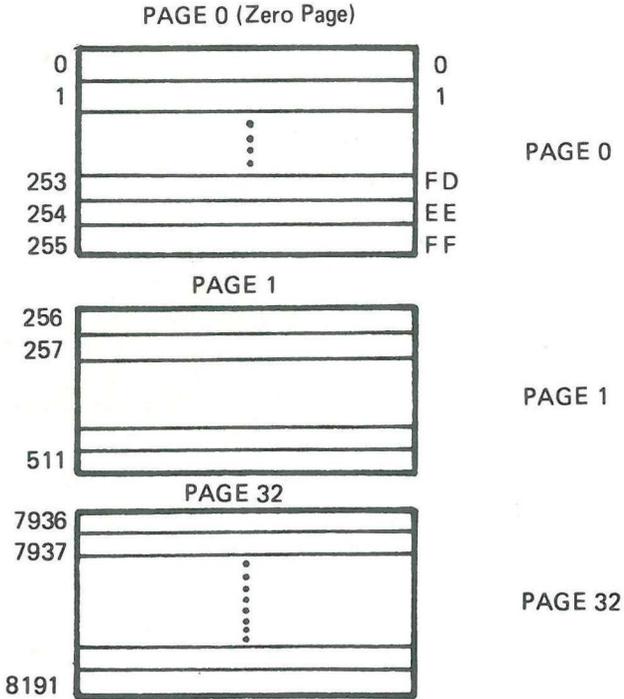
Die Übung 2 führt uns zu unserem nächsten Beispiel. Wie wir wissen, müssen wir unser Maschinensprachenprogramm, wenn wir es nicht an einer geschützten Stelle (z. B. Cassettenpuffer) abspeichern, vor der Überschreibung durch das BASIC schützen. Die Speicherzelle 56 dezimal enthält im VC-20 eine Adresse, die den verfügbaren Speicherraum nach oben begrenzt. Sehen Sie einmal durch ?PEEK (56) nach. Sie finden die Zahl 28. Dies bedeutet, daß der Speicherbereich Ihres PETs bei PAGE 28 endet (bei unseren Experimenten).

Der Hauptspeicher (RAM) des VC-20 ist in PAGES aufgeteilt, die sich wie folgt aufbauen.

28 Pages mit je 256 Worten Speicherplatz ergeben die 7168 Worte – den Bereich zwischen 400 und FFF und den für das Betriebssystem verwendete RAM-Teil ergeben die 3581 Byte freies RAM.

Wenn sich der VC-20 nach dem Einschalten meldet, zeigt er einen geringeren Wert. Dies liegt daran, daß ein kleiner Teil des RAM zur Speicherung von Zwischenwerten etc. reserviert wurde.

# PAGE-Aufbau im VC-20



Unser kleines Programm steht jetzt im Speicher und kann mit dem Monitor auf Cassette abgelegt werden.

### Achtung !

Beim Wiedereinladen muß die Page-Grenze wieder mit POKE 56,28 entsprechend gesetzt werden. Übrigens bleibt der Wert in Zelle 28 solange bis der Computer abgeschaltet wird, oder der Wert durch einen neuen POKE-Befehl geändert wird.

## Übernahme von Maschinenprogrammen in ein BASIC-Programm

Nicht jeder hat einen Maschinensprachenmonitor VC-Mona und kann dann Ihr eben geschriebenes Maschinenprogramm wieder in den VC-20 einlesen. Aus diesem Grunde wollen wir Ihnen jetzt ein kleines BASIC-Programm vorstellen, welches ein im Speicher stehendes Maschinenprogramm in DATA-Statements verwandelt und vom BASIC her, dann mit POKE-Befehlen in dezimal in den Speicher gepoked werden kann.

### Beschreibung POKER MAKER

Dieses Programm läuft auf der Grundversion des VC-20. POKER MAKER generiert ein Programm, welches Dezimalwerte in den Speicherbereich bringt. Geben Sie Anfangs- und Endadresse in dezimaler Form ein. Nach einigen Sekunden steht Ihr gewünschtes Programm im Speicher.

Achtung !

Vorher auf Cassette speichern, da sich das POKER MAKER Programm dabei löscht !!

Es empfiehlt sich Maschinenprogramme ab 1C00 Hex Anfangsadresse zu legen.

```
502 PRINT:PRINT"IDAS PROGRAMM GENERIERT"  
504 PRINT"EIN BASIC PROGRAMM"  
506 REM DIESES PROGRAMM ERZEUGT EIN MASCHINEN-  
507 REM PROGRAMM IN DATA ZEILEN,WELCHE  
508 REM DANN EINGEPOKED WERDEN KOENNEN.  
510 REM ACHTEN SIE DARAUF,DASS SIE SINN-  
511 REM VOLLEN CODE IN DEN RAM ZELLEN HABEN.  
512 INPUT "ANFANGSADRESSE,DEZ";L1  
514 INPUT "ENDADRESSE DEZ";L2  
516 IF L2<=L1 THEN 500  
518 PRINT "7000"  
520 PRINT "20FORI=";L1;"TO";L2;" : READD C:POKEI,DC:NEXTI"  
522 PRINT "L1=";L1;" :L2=";L2;" :GOTO528"  
524 POKE 198,10:FOR N=0 TO 9:POKE 631+N,13:NEXT N  
526 PRINT "7000":END  
528 ML=L1:LN=22  
530 PRINT "7000"
```

```

532 PRINT LN;"DATA";
534 D1=PEEK(ML):D1$=STR$(D1)
536 DC$=RIGHT$(D1$,LEN(D1$)-1)
538 PRINT DC$;:ML=ML+1
540 IF POS(0)<18 THEN 544
542 PRINT:PRINT "LN=";LN;" +2:ML=";ML;" :L2=";L2;" :GOTO530
   :GOTO 548
544 IF ML>L2 THEN PRINT:PRINT "GOTO554":GOTO 548
546 PRINT CHR$(44);:GOTO 534
548 PRINT "☺"
550 POKE 198,10:FOR N=0 TO 9:POKE 631+N,13:NEXT N
552 PRINT "☺":END
554 I=500:J=502:K=504
556 REM
558 REM
560 PRINT "☺☺☺":PRINT I:PRINT J:PRINT K
562 PRINT "I=";I+6;" :J=";J+6;" :K=";K+6;" :GOTO560"
564 PRINT "☺":POKE 198,10:FOR N=0 TO 9:POKE 631+N,13:NEXT N

```

Es wäre sehr praktisch, wenn Sie sich den Monitor VC-Mona noch von Ballast befreien und den POKER MAKER mit in den VC-Mona einbauen würden.

Nachdem POKER MAKER abgelaufen ist, erfolgt eine Fehlermeldung, welche Sie ignorieren können. Geben Sie List ein und Sie erhalten das gewünschte Programm.

Das nachfolgende BASIC-Programm wurde mit diesem POKER MAKER Programm generiert.

```

20 FORI= 7184 TO 7194 :READDC:POKEI,DC:NEXTI
22 DATA169,0,162,9
24 DATA157,255,28
26 DATA202,16,250
28 DATA96
507 REM PROGRAMM IN DATA ZEILEN,WELCHE
511 REM VOLLEN CODE IN DEN RAM ZELLEN HABEN.

```

Es poked uns das vorher besprochene Maschinenprogramm in den Speicher ab Adresse 7184 in den VC-20 Speicher.

## Achtung !

Ganz wichtig. Bevor Sie später dieses generierte Programm starten, müssen Sie zuerst wieder den oberen Speicherbereich vor dem Übergriff durch BASIC schützen. Dies geschieht, wie wir bereits gesehen haben mit POKE 56,28. Sie könnten auch die Zeile 10 POKE 56,28 noch vor das Programm setzen.

Das obenstehende BASIC-Programm können Sie jetzt auf Cassette ablegen, oder in ein anderes BASIC-Programm als Maschinenunterprogrammgenerator einbauen. Sie können es nach RUN mit dem Befehl SYS (7184) aufrufen. Dieser Befehl führt dann das Maschinenprogramm aus und kehrt wieder nach BASIC zurück.

Mit den nachfolgenden beiden Zeilen (100 + 200) können Sie von BASIC hier die Zellen 7184 bis 7194 ändern und diese z. B. alle auf 1 legen.

```
100 FOR I=7184 TO 7194:POKE I,0:NEXT I
200 FOR I=7184 TO 7194:PRINTPEEK(I):NEXT I
```

Die Zellen, die durch unser Programm zu Null gemacht wurden, liegen ab 1CFF Hex (7423 Dez) bis 1D08 Hex (7432 Dez).

Wir sehen uns dies jetzt mit den folgenden Zeilen an:

```
300 FOR M=7423 TO 7432:PRINT PEEK(M):NEXT M
350 END
400 FOR N=7423 TO 7432:POKE N,255:NEXT N
```

RUN 300 zeigt uns die Zellen, die wir später zu Null machen wollen. RUN 400 schreibt uns etwas anderes (255) ein, damit wir es später testen können.

```
20 FOR I= 7184 TO 7194 :READDC:POKEI,DC:NEXTI
22 DATA169,0,162,9
24 DATA157,255,28
26 DATA202,16,250
```

```
28 DATA96
29 END
100 FOR I=7184 TO 7194:POKE I,1:NEXT I
200 FOR I=7184 TO 7194:PRINTPEEK(I):NEXT I
300 FOR M=7423 TO 7432:PRINT PEEK(M):NEXT M
350 END
400 FOR N=7423 TO 7432:POKE N,255:NEXT N
507 REM PROGRAMM IN DATA ZEILEN.WELCHE
511 REM VOLLEN CODE IN DEN RAM ZELLEN HABEN.
```

Nachdem wir RUN 400 gegeben haben, starten wir mit RUN das Hauptprogramm an Zeile 20. Dann geben wir im Direktmode ohne Zeilenzahl SYS 7184 (Return) ein und starten das Maschinenprogramm. Anschließend sehen wir mit RUN 300 nach ob unser Maschinenprogramm auch gearbeitet hat.

Natürlich bleibt es jetzt Ihnen überlassen, welche Experimente und Versuche Sie noch selbst ausführen wollen. Da wäre z. B. eine Explosionsroutine in Maschinensprache oder schnelle Grafik auf dem Bildschirm usw. Oder Frequenzgeneratoren über Port B des User Ports. Ihrer Fantasie sind jetzt kaum noch Grenzen gesetzt.

## Die Funktionsweise der **USR(X)**-Funktion

Mit dem **SYS**-Befehl wird eine Startadresse eines Maschinenprogramms aus einem **BASIC**-Programm aufgerufen. Diese Startadresse ist bekanntlich der in Klammern zu setzende Operand des **SYS**-Befehls. Der Sprung ins Maschinenprogramm ist unmittelbar und es werden keine Daten aus dem **BASIC**-Programm mitgenommen. Entweder braucht man keine solchen Daten aus dem bisherigen **BASIC**-Programm-Verlauf für die angesprungene Maschinenroutine, z.B., weil diese Routine selbst Daten vom **User-Port** holen soll, um sie an Stellen abzuspeichern, die dann mit **PEEK** vom **BASIC** abgefragt werden können.

Oder man hat vor dem Sprung in das Maschinenprogramm die benötigten Daten von **BASIC** mit **POKE** in bestimmte Speicherstellen gebracht, die dann nach den Regeln der Maschinenprogrammierung im Verlauf des Maschinen-Programms angesprochen werden. Was in diesen oder anderen Speicherstellen am Ende des Maschinenprogramm-Laufes steht, kann man nach dem Rücksprung zum **BASIC**-Programm mit **POKE** herausholen. Der Rücksprung selbst wird bewirkt durch den Maschinenbefehl **RTS** (opcode hex 60) am Schluß des Maschinen-Programms. **BASIC** arbeitet dann weiter ab dem Befehl, der nach dem **SYS**-Befehl steht, der den Sprung ins Maschinenprogramm bewirkte.

Prinzipiell kann man in der beschriebenen Weise mit **PEEK** und **POKE** auch beliebig komplizierte Daten (Variablenwerte) zwischen **BASIC** und Maschinenprogramm vorher und nachher rangieren, womit auch jede Funktion von **USR(X)** restlos simuliert werden kann. Sehr praktisch ist dies aber allgemein nicht! Wenn die zu übertragenden Daten zwischen den Programmarten einfache ganze Zahlen sind, die sich auch noch in einem Byte unterbringen lassen (Dezimalzahlen bis 255), dann ist ein **POKE**, gefolgt von **SYS**, oder entsprechender **PEEK** nach **RTS** (z.B., wenn ein Datenbyte etwa vom **User-Port** im Maschinenprogramm geholt wurde), die auch im praktischen Sinn richtige Programmierweise.

Dieses Verfahren wird aber zumindest außerordentlich schwerfällig, sehr unübersichtlich und erfordert sehr zeitraubende Byte-Umrechnungen seitens des Benutzers, wenn die von **BASIC** an das Maschinenprogramm zu übergebenden Zahlenwerte Dezimalzahlen mit vielen Vor- und Nachkommastellen, bzw. Mantissen voller Präzision mit Exponenten sind, und auch die Vorzeichen von Mantisse und Exponent im Maschinenprogramm gebraucht werden. Dies alles mit einer Reihe von

POKEs für die entsprechenden Bytes zu bewerkstelligen ist, wie gesagt, selbstverständlich möglich, aber hochgradig unbequem. Der `USR(X)`-Befehl führt diese Aufgabe übersichtlich und schnell aus, gleichzeitig mit dem Sprung in das Maschinenprogramm, den er wie auch `SYS` bewirkt. Den Zahlenwert in voller Präzision "schleppt er dabei mit", und führt die erforderliche Übersetzung von der Dezimaldarstellung (egal, ob mit Festkommadarstellung oder in Form von Mantisse und Exponent zuvor als Wert der zugeordneten Variable in BASIC gewesen) in die sogenannte Fließkomma-Binärdarstellung, mit der und nur mit der das Maschinenprogramm arbeitet, gleichzeitig aus. Die dabei gewonnenen Bytes bringt der `USR(X)`-Befehl in die ihnen zugeordneten Stellen des sogenannten Software-Akkumulators auf der Seite 0 des VC-20-Speichers (0–FF).

Beim VC-20 finden wir diese Adressen ab 97 dezimal (61 Hex). Siehe dazu die Speicherbelegungsliste in diesem Buch.

`USR` ist eine BASIC-Funktion und kein Befehl wie `PRINT`, `GOTO` etc. Die Funktion  $Y = \text{USR}(X)$  ermöglicht es, ein Maschinenprogramm im Speicher zu starten, dessen Adresse in den ersten beiden Speicherplätzen des Rechners steht.

Die Adresse muß mit `POKE 1, Higher Byte` und  
`POKE 2, Lower Byte`

vor dem Start des `USR(X)` Programmes in den Speicher gebracht werden. Das Argument `X` wird in den Fließkomma-Akkumulator ab 97 dezimal gebracht. Sie können diesen Wert dann in Ihrem BASIC-Programm verwenden oder nicht. Wenn nach Ausführung der  $Y=\text{USR}(X)$ -Funktion Ihr Programm dann ins BASIC zurückkehrt, kann sich ein anderer Wert im FP-Akku befinden. Wenn ein Wert zurückgelassen wird, kann man diesen auch über die Variable `Y` finden.

Es ist besser, auf diese Weise nach dem Wert zu sehen, als in BASIC mit `PRINT PEEK` in den FP-Akku zu sehen.

Mit dem nachfolgenden Beispielprogramm können Sie die Funktion `USR(X)` ausprobieren. Vergessen Sie nicht, vorher

```
POKE 1, 16  
POKE 2, 28
```

für die Anfangsadresse des Maschinenprogrammes einzugeben.

```

10 FOR I=7423 TO 7433:POKEI,255:NEXT I
15 END
20 FOR I=7423 TO 7433:PRINT PEEK(I):NEXT I
25 END
50 Y=USR(X)
60 END

```

Im Gegensatz zu SYS kann mit der USR-Funktion eine ganze Zahl oder ein ganzzahliger Ausdruck von BASIC mit in ein Maschinenprogramm genommen werden und dort für Berechnungen weiterverwendet werden.

Die Verwendung von Maschinenunterprogrammen in BASIC ist in vielen Anwendungen unumgänglich. Sie werden in Ihrer zukünftigen Programmierpraxis sicher bald auf Fälle wie Tonausgabe, Grafik etc. stoßen, wo Sie den Einsatz von Maschinenunterprogrammen nicht mehr umgehen können.

### Hier noch einige Hilfstabellen

V A L U E	HEX/DEC CONVERSION (<64K)				POSITION
	16 <sup>3</sup>	16 <sup>2</sup>	16 <sup>1</sup>	16 <sup>0</sup>	
0	0	0	0	0	
1	4096	256	16	1	
2	8192	512	32	2	
3	12288	768	48	3	
4	16384	1024	64	4	
5	20480	1280	80	5	
6	24576	1536	96	6	
7	28672	1792	112	7	
8	32768	2048	128	8	
9	36864	2304	144	9	
A	40960	2560	160	10	
B	45056	2816	176	11	
C	49152	3072	192	12	
D	53248	3328	208	13	
E	57344	3584	224	14	
F	61440	3840	240	15	

## Hex-Dezimal

Lokalisieren Sie für jedes Hex-Digit das entsprechende dezimale Äquivalent und addieren Sie die Werte.

## Dezimal-Hex

Suchen Sie in der Tabelle die größte Zahl, die kleiner als Ihre Dezimalzahl ist. Notieren Sie den HEX-Wert in der ersten Position. Ziehen Sie den vorangegangenen Wert immer von der zu verwandelnden Zahl ab und wiederholen Sie für jeden Rest.

## ASCII-Tabelle

FIRST DIGIT →

		0		1		2		3		4		5		6		7	
SECOND DIGIT ↓	0	NUL	0	DLE	16	SP	32	0	48	@	64	P	80	\	96	p	112
	1	SOH	1	DC1	17	!	33	1	49	a	65	Q	81	a	97	q	113
	2	STX	2	DC2	18	"	34	2	50	B	66	R	82	b	98	r	114
	3	ETX	3	DC3	19	£	35	3	51	C	67	S	83	c	99	s	115
	4	EOT	4	DC4	20	\$	36	4	52	D	68	T	84	d	100	t	116
	5	ENQ	5	NAK	21	%	37	5	53	E	69	U	85	e	101	u	117
	6	ACK	6	SYN	22	&	38	6	54	F	70	V	86	f	102	v	118
	7	BEL	7	ETB	23	'	39	7	55	G	71	W	87	g	103	w	119
	8	BS	8	CAN	24	(	40	8	56	H	72	X	88	h	104	x	120
	9	HT	9	EM	25	)	41	9	57	I	73	Y	89	i	105	y	121
	A	LF	10	SUB	26	*	42	:	58	J	74	Z	90	j	106	z	122
	B	VT	11	ESC	27	+	43	;	59	K	75	[	91	k	107		123
	C	FF	12	FS	28	,	44	<	60	L	76	]	92	l	108		124
	D	CR	13	GS	29	-	45	=	61	M	77	^	93	m	109		125
	E	SO	14	RS	30	.	46	>	62	N	78	^	94	n	110	-	126
	F	SI	15	US	31	/	47	?	63	O	79	_	95	o	111	DEL	127

## Zweier Potenzen

4096	$2^{12}$	1	$2^0$
8192	$2^{13}$	2	$2^1$
16384	$2^{14}$	4	$2^2$
32768	$2^{15}$	8	$2^3$
65536	$2^{16}$	<hr/>	
131072	$2^{17}$	16	$2^4$
262144	$2^{18}$	32	$2^5$
524288	$2^{19}$	64	$2^6$
1048576	$2^{20}$	128	$2^7$
<hr/>		<hr/>	
2097152	$2^{21}$	256	$2^8$
4194304	$2^{22}$	512	$2^9$
8388608	$2^{23}$	1024	$2^{10}$
16777216	$2^{24}$	2048	$2^{11}$

# Ein einfacher Monitor für VC-20

## Ein einfacher Monitor für VC-20

Nachfolgend finden Sie ein Listing für einen einfachen Monitor für den Maschinensprachenprogrammierer. Das Programm arbeitet auf der Grundversion des VC-20 ohne Speichererweiterung.

- C = Change Memory-Location. Ändern des Inhaltes einer Speicherzelle
- M = Ausgabe einer Speicherzelle auf dem Bildschirm
- G = Start eines Maschinenprogrammes an einer bestimmten Adresse
- S = SAVE (Speichern von Maschinenprogrammen auf Cassette)
- R = Lesen von Maschinenprogrammen auf Cassette

Der sehr einfache Monitor ist in erster Linie für den ersten Kontakt mit Ihrem VC-20 von Commodore gedacht. Er erlaubt dem Anwender kleine Expeditionen im ROM- und RAM-Bereich.

Bei Arbeiten mit dem Monitor hat sich gezeigt, daß man kleine Maschinensprachenexperimente ab der Adresse 1D00 durchführen kann. Der Speicherbereich des VC-20 in der Grundauführung sieht wie folgt aus:

8K	8000
8K	6000
8K	4000
8K	2000
Bildschirm- speicher	8164 Dez. 1FFF
Verfügbares RAM	7680 Dez. 1E00
	1000
	0FFF
	0400 H.
	03FB H.
	0000 H.

Von 0000 Hex bis ca. 03FB befinden sich die BASIC-Vektoren und Adressen, die der VC-20 für seinen eigenen Betrieb benötigt. Von 400 Hex bis 0FFF ist kein RAM vorhanden. Bei den PET- und CBM-Versionen beginnt ab 400 Hex der BASIC Textbereich. Beim VC-20 beginnt der BASIC-Textbereich ab 1000 Hex. bzw. 1001 Hex. Damit stehen im VC-20 ca. 3,5K RAM ab Adresse 1000 für Anwenderprogramme zur Verfügung. Der obere Teil des 4K Blockes wird vom Bildschirmspeicher belegt (1E00 – 1FFF).

Das Programm läßt dem Anwender noch ca. 750 Byte. Mit dem C-Befehl können Sie Speicherzellen ansehen und evtl. ändern. Nach Eingabe der Anfangsadresse können Sie die Adressen durch Drücken der Space-Taste erhöhen und mit der Taste  (Pfeil nach oben) die Adressen erniedrigen.

Wichtig ist, daß Anfangs- und Endadresse immer vierstellig in HEX eingegeben werden. Auch die Ausgabe erfolgt in Hex. Zur Beendigung der Eingabe mit C drücken Sie RETURN. Sie können dann zurück in

den Monitor. Erneutes Drücken von RETURN bringt Sie wieder in den BASIC-Interpreter.

Bei all Ihren Maschinensprachenexperimenten mit diesem Monitor müssen Sie also darauf achten, daß Sie nicht mit dem eigentlichen BASIC-Programm des Monitors oder mit dem Bildschirmspeicher in Berührung kommen. Der Bereich um 1D00 Hex dürfte hier ganz gut geeignet sein.

Der Monitor erlaubt Ihnen weiterhin, Maschinenprogramme auf Cassette abzulegen und wieder einzulesen. Beim Abspeichern (SAVE) muß die Anfangs- und Endadresse sowie ein Name eingegeben werden. Beim Wiedereinladen muß der Name eingegeben werden. Nach der Aufforderung: "REWIND und DRUECKE TASTE", müssen Sie irgendeine Taste auf der VIC-Tastatur drücken. Am Bildschirm erscheint dann: "PRESS PLAY ON TAPE". Sie können jetzt das vorher gespeicherte Programm wieder einladen.

```
8020 PRINT"C=SPEICHER AENDERN"  
8030 PRINT"M=SPEICHER DUMP"  
8040 PRINT"G=STARTE AN ADRESSE"  
8060 PRINT"S=SAVE AUF CASSETTE"  
8070 PRINT"R=READ VON CASSETTE"  
8080 PRINT"RETURN=RUECKKEHR BASIC"  
8090 PRINT  
8091 PRINT"GEBEN SIE EIN BEFEHL"  
8095 PRINT  
8096 PRINT"* ";  
8110 GET C$:IF C$="" GOTO 8110  
8120 PRINTC$  
8130 IF C$="C" GOTO 8400  
8140 IF C$="M" GOTO 8700  
8150 IF C$="G" GOTO 9300  
8170 IF C$="S" GOTO 9700  
8180 IF C$="R" GOTO 9900  
8190 IF ASC(C$)=13 THEN END  
8200 PRINT"BEFEHL IST UNBEKANNT"  
8210 PRINT  
8220 PRINT
```

```

8230 GOTO 8096
8400 REM
8410 PRINT"ANFANGSADRESSE";
8420 INPUT HEX4$
8430 GOSUB 9000
8440 S=DEC4+1
8450 PRINT
8460 S=S-1
8470 GOSUB 9100
8480 PRINT " ";
8490 SO=PEEK(S)
8500 N=3:GOSUB 9150
8510 PRINT" - ";
8520 DEC2=0
8530 FOR I=2 TO 1 STEP-1
8540 GET C$:IF C#="" GOTO 8540
8550 IF C#="^" GOTO 8450
8560 IF C#=" " GOTO 8640
8570 TEMP=ASC(C#)
8580 IF TEMP=13 GOTO 8095
8590 PRINTC#;
8600 F=48:IF TEMP>58 THEN F=55
8610 DEC2=DEC2+(TEMP-F)*I*I*I*I
8620 NEXT I
8630 POKE S,DEC2
8640 S=S+1
8650 PRINT
8660 GOTO 8470
8700 REM
8705 PRINT
8710 PRINT"ANFANGSADRESSE";
8720 INPUT HEX4$
8730 GOSUB 9000
8740 L=DEC4
8745 PRINT
8750 PRINT"ENDADRESSE";
8760 INPUT HEX4$
8765 PRINT
8770 GOSUB 9000
8780 H=DEC4
8790 D=H-L
8800 FOR I=1 TO D STEP 8
8810 S=L-1+I

```

```

8820 GOSUB 9100
8830 PRINT " ";
8840 FOR J=1 TO 8
8850 S0=PEEK(S-1+J)
8860 N=3:GOSUB 9150
8870 PRINT " ";
8880 NEXT J
8890 PRINT " "
8900 NEXT I
8910 GOTO 8095
9000 REM
9010 DEC4=0:MULT=4096
9020 FOR I=1 TO 4
9030 TEMP=ASC(MID$(HEX4$,I,1))
9040 F=48:IF TEMP>58 THEN F=55
9050 DEC4=DEC4+(TEMP-F)*MULT
9060 MULT=MULT/16
9070 NEXT I
9080 RETURN
9100 REM
9110 N=1
9120 S(1)=INT(S/4096)
9130 S(2)=INT((S-S(1)*4096)/256)
9140 S0=(S-S(1)*4096)-S(2)*256
9150 S(3)=INT(S0/16)
9160 S(4)=(S0-S(3)*16)
9170 FOR K=N TO 4
9180 F=48:IF S(K)>9 THEN F=55
9190 PRINTCHR$(S(K)+F);
9200 NEXT K
9210 RETURN
9300 REM
9305 PRINT
9310 PRINT"ANFANGSADRESSE";
9320 INPUT HEX4$
9330 GOSUB 9000
9340 SYS(DEC4)
9350 GOTO 8095
9700 REM
9705 PRINT
9710 PRINT"ANFANGSADRESSE";
9720 INPUT HEX4$
9730 GOSUB 9000

```

```

9740 L=DEC4
9745 PRINT
9750 PRINT"ENDADRESSE";
9760 INPUT HEX4$
9770 GOSUB 9000
9780 H=DEC4
9790 D=H-L
9791 PRINT
9792 FS=-1:FD=-1:FC=191
9793 PRINT"FILENAME";
9794 INPUT F$
9795 OPEN1,1,1,F$
9797 J=LEN(STR$(D))
9799 FS=FS+J
9810 PRINT#1,D
9820 FOR J=0 TO D
9830 I=PEEK(L+J)
9840 PRINT#1,I
9841 K=LEN(STR$(I))
9842 FE=INT(-(FS+K)/FC)
9843 IF FE=FD GOTO 9849
9844 POKE 59411,53
9845 T=TI
9846 IF(TI-T)<5 GOTO 9846
9847 POKE 59411,61
9848 FS=-1
9849 FS=FS+K
9850 NEXT J
9860 CLOSE 1
9865 PRINT
9870 PRINT"GESPEICHERT"
9880 GOTO 8095
9900 REM
9905 PRINT
9906 PRINT"FILENAME";
9907 INPUT F$
9908 PRINT
9910 PRINT"REWIND UND DRUECKE TAS
TE"
9920 GET A$:IF A$=""THEN 9920
9930 OPEN1,1,0,F$
9940 INPUT#1,D
9945 PRINT"3"

```

```
9950 PRINT"ANFANGSADRESSE";
9960 INPUT HEX4$
9970 GOSUB 9000
9974 PRINT
9975 L=DEC4
9980 FOR I=0 TO D
9990 INPUT#1,X
9991 PRINT"X";
9992 POKE L+I,X
9993 NEXT I
9994 CLOSE1
9995 PRINT:PRINT
9996 PRINT"ENDE DES FILES"
9997 GOTO8095
READY.
```

# Basisumrechnung mit dem VC-20

## Basisumrechnung mit dem VC-20

Fast jeder, der sich intensiv mit seinem Personalcomputer beschäftigen will, wird eines Tages mit Zahlen verschiedener Basiswerte konfrontiert.

Das nachfolgende Programm erlaubt es dem Anwender Zahlen von und in Hexadezimal (Sedezimal)

Dezimal  
Oktal  
Binär und  
Oktal-Split  
zu verwandeln.

Eine große Hilfe wird Ihnen dieses Programm beim Programmieren in Maschinsprache sein.

```
2 DIM B(20)
5 PRINT"3"
6 PRINTTAB(10)"BASIS UMWANDLUNG"
8 PRINT"XXXX"
12 PRINT"ICH WANDLE ZAHLEN IN"
13 PRINT"DEZIMAL,OKTAL,OKTAL -"
14 PRINT"SPLIT,HEX UND BINAER"
20 PRINT"MEINGABE BASIS,ZAHL"
21 PRINT"UND GEWUENSCHTE UM-"
22 PRINT"RECHNUNG"
25 PRINT:PRINT
26 PRINT"OKTAL-SPLIT='S'"
30 PRINT"DEZIMAL='D' OKTAL='O'"
31 PRINT"HEXADEZIMAL='H' BINAER='B'"
```

```

32 PRINT"UNUR 7STELLIGE ZAHLEN"
33 GOTO 100
35 PRINT "EINE ANDERE ZAHL(J,N)?";
36 GET A$: IF A$="" THEN GOTO 36
37 IF A$="J" THEN GOTO 120
38 PRINT"J":PRINT"UNIBASIS UMRECHNGN":PRINT
39 POKE 59468,12: END
40 PRINT "UNIVON WELCHER BASIS (D,O,S,H,B)?";
42 GET Z$: IF Z$="" THEN GOTO 42
45 PRINTZ$
50 IF Z$="H" THEN 1400
55 PRINT"UN"
60 INPUT "UMZUWANDELNDE ZAHL";A
65 PRINT"UN"
70 PRINT "ZU BASIS.... (D,O,S,H,B) ? ";
74 GET Y$: IF Y$="" THEN GOTO 74
76 PRINT Y$
80 PRINT"UN"
90 IF Z$="S" THEN 1260
92 IF Z$="D" THEN 490
94 IF Z$="O" THEN 580
96 IF Z$="B" THEN 640
98 GOTO 640
100 PRINT"UN(DRUECKE EINE TASTE)";
110 GET R$: IF R$="" THEN GOTO 110
120 PRINT"J"
130 PRINTTAB(1)"ELCOMP ZAHLEN"
140 PRINT"UN";
150 GOTO 40
490 IF Y$="O" THEN 540
495 IF Y$="H" THEN 1600
500 IF Y$="S" THEN 1135
510 Z=2
520 Y$="BINAER ="
530 GOSUB 800
535 GOTO 35
540 Z=8
545 PRINT
550 Y$="OKTAL ="
560 GOSUB 800
565 GOTO 35
580 Z=8
590 IF Y$="D" THEN 620

```

```

595 IFY$="S"THEN GOSUB920:A=G:GOTO1135
596 IFY$="H"THEN GOSUB 920:A=G:GOTO1600
600 GOSUB 920
610 A=G: GOTO 510
620 GOSUB 920
630 GOTO 35
640 Z=2
670 IF Y$="D" THEN 700
675 IFY$="S"THEN GOSUB920:A=G:GOTO1135
676 IFY$="H"THEN GOSUB920:A=G:GOTO 1600
680 GOSUB 920
690 A=G: GOTO 540
700 GOSUB 920
710 GOTO35
800 N=0
810 N=N+1
830 B=INT(A/Z)
840 C=A-(B*Z)
850 B(N)=INT(C+.5)
860 A=B
870 IF B>0 THEN 810
872 IF Y$="H" THEN 912
875 IF Y$="S" THEN 912
880 PRINT Y$;
885 GOTO 900
890 N=N-1
895 IF N=0 THEN 911
900 PRINT B(N);
910 GOTO 890
911 PRINT
912 RETURN
920 C=1:E=(-1)
925 G=0
930 B=A/C
935 E=E+1
940 IF B<Z THEN 970
950 C=C*10
960 GOTO 930
970 D=INT(A/C)
980 H=(Z↑E)*D
1000 D=D*C
1010 A=A-D
1020 G=G+H

```

```

1025 C=C/10
1040 E=E-1
1050 IF E=>0 THEN 970
1055 IF Z$="S" THEN 1070
1060 PRINT"DEZIMAL =";G
1065 PRINT
1070 RETURN
1135 PRINT"OKTAL SPLIT =";
1140 B(3)=0: B(2)=0: B(1)=0
1150 B=INT(A/256)
1160 C1=A-(B*256)
1165 X=0
1170 A=B
1180 Z=8
1190 GOSUB 800
1200 N=3
1210 PRINTB(N);
1220 N=N-1
1230 IF N>0 THEN 1210
1240 X=X+1
1245 IF X>1 THEN PRINT: GOTO 35
1250 A=C1: B(3)=0: B(2)=0: B(1)=0: GOTO1180
1260 REM OS
1265 PRINT
1270 C=1000: Z=8
1280 B=INT(A/C)
1290 C1=A-(B*C)
1294 A=B
1300 GOSUB 920
1310 G1=G*256
1320 A=C1
1330 GOSUB 920
1340 G=G+G1
1350 PRINT"DEZIMAL =";G
1355 A=G
1360 IF Y$="B" THEN PRINT:GOTO 510
1362 IF Y$="H" THEN 1600
1364 IF Y$="O" THEN 540
1366 GOTO 35
1400 PRINT"X00": INPUT "UMZUWANDELNDE ZAHL";X$:PRINT:PRINT
1401 PRINT"MIN WELCHE BASIS(D,O,S,H,B) ? ";
1402 GET Y$: IF Y$="" THEN GOTO 1402
1403 PRINT Y$:PRINT

```

```

1405 HL=0
1406 G=0
1407 HJ=LEN(X$)
1410 FOR I=1 TO HJ
1415 XX$=MID$(X$,I,1)
1420 H=ASC(XX$)
1425 HL=HJ-I
1430 IF H<=57 THEN 1445
1435 B(I)=(H-55)*(16^HL)
1440 GOTO 1450
1445 B(I)=(H-48)*(16^HL)
1450 NEXT I
1455 FOR I=1 TO HJ
1460 G=G+B(I)
1465 NEXT I
1468 PRINT:PRINT:PRINT
1470 PRINT"DEZIMAL =" ;G
1475 PRINT
1480 A=INT(G+.5)
1485 IF Y$="0" THEN 540
1490 IF Y$="B" THEN 510
1495 IF Y$="S" THEN 1135
1500 GOTO 35
1600 Z=16
1601 PRINT:PRINT"HEXADEZIMAL = ";
1610 GOSUB 800
1611 GOTO 1630
1615 N=N-1
1625 IF N=0 THEN 1660
1630 IF B(N)=>10 THEN 1650
1635 A1=B(N)+48
1636 PRINT CHR$(A1);
1640 GOTO 1615
1650 A1=B(N)+55
1655 GOTO 1636
1660 PRINT
1661 GOTO 35
1670 END

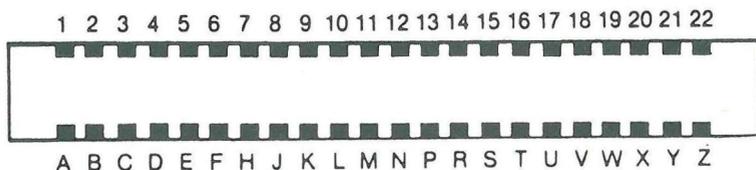
```

READY.

# Universal Experimentierplatine für VC-20

## Universal Experimentierplatine für VC-20

Die Universal Experimentierplatine erlaubt dem Anwender externe Erweiterungen (Speicherzusatz, Porterweiterungen, etc.) direkt über den "Memory Expansion"-Stecker anzuschließen. Die Platine enthält, genau wie der Speichererweiterungsstecker 2 x 22 Anschlüsse, welche übereinander angeordnet sind. Die Bezeichnungen sind von hinten gesehen wie folgt (von links nach rechts):

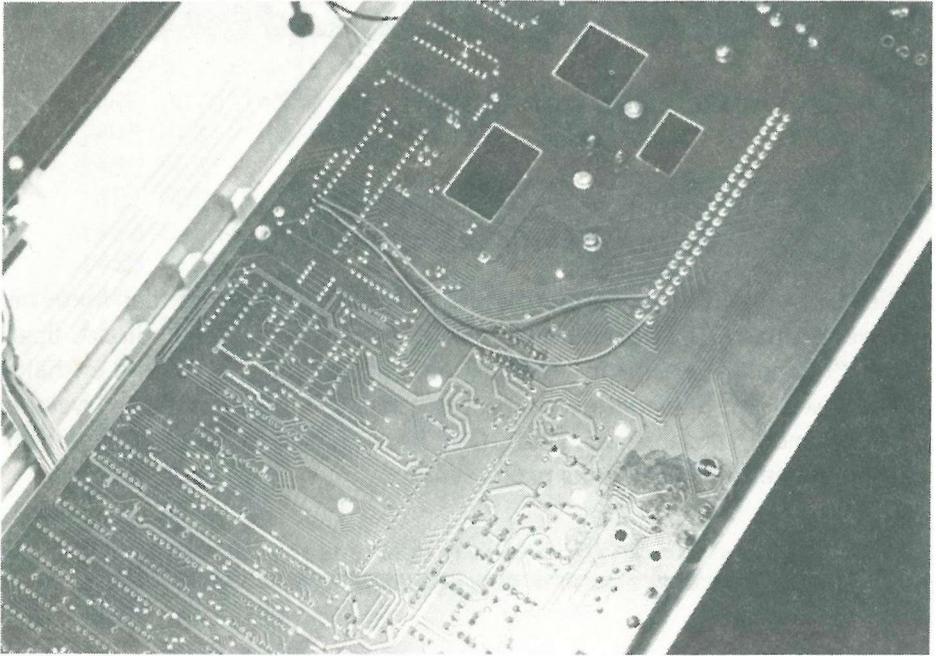


Dieser Bus wird in den USA auch mit S44 Standardbus bezeichnet (Standardbus der amerikanischen Industrie). Nur erfolgt dort die Numerierung umgekehrt. D. h. A beginnt rechts. Die Platine ist beiderseitig beschichtet, wobei die Anschlüsse A – Z und 1 – 22 so gelegt wurden, daß man sie einfach durch Einlöten eines Drahtes auf die gegenüberliegende Platinenseite herüberholen kann. Achtung beim Experimentieren – auf der Platine ist die Numerierung umgekehrt !

Am "Memory Expansion"-Ausgang des VC-20 sind alle Datenleitungen ungepuffert, direkt von der 6502A CPU angeschlossen. Weiterhin stehen die Adressen CA0 – CA13 zur Verfügung. Der ernsthafte Experimentierer wird als erstes die beiden fehlenden Adressen CA14

und CA15 von der CPU (6502A) an die noch freien Anschlüsse Y und 20 legen.

Dann öffnen Sie das Gehäuse und nehmen die Mutterplatine heraus. Wenden Sie die Platine so, daß sie wie im Bild unten vor Ihnen liegt.



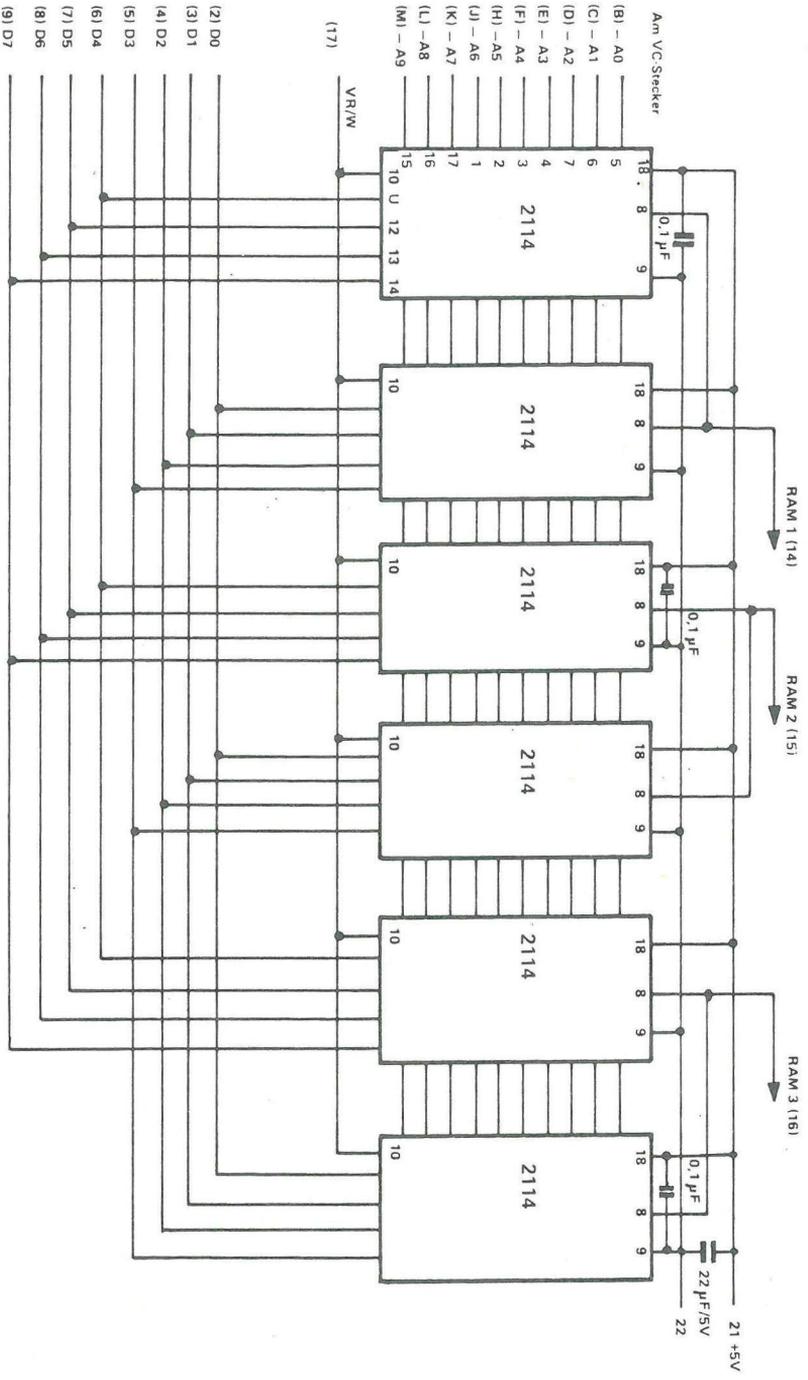
Löten Sie jetzt diese beiden Verbindungen ein.

Y nach Pin 24 am 6502A  
20 nach Pin 25 am 6502A

Neben Adress- und Datenleitungen sind noch die Leitungen  $\overline{\text{RAM1}}$ ,  $\overline{\text{RAM2}}$ ,  $\overline{\text{RAM3}}$  herausgeführt.

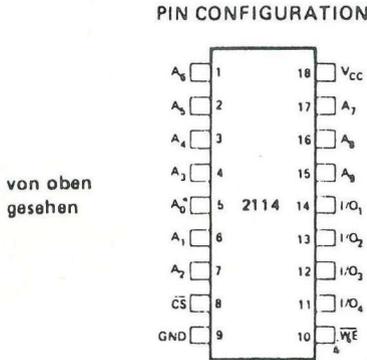
Diese dienen dazu die drei 1K Blöcke zwischen 400 und FFF Hex zu dekodieren. Eine 3K Speichererweiterung läßt sich also ohne externe Dekodierung aufbauen. Alles was wir dazu zusätzlich benötigen sind 6 Speicherbausteine 2114 oder besser L2114 (die Version mit der geringeren Leistungsaufnahme), drei Kondensatoren 0,1  $\mu\text{F}/60\text{V}$ , 6 Stück 18 polige IC-Sockel.

### 3K Speichererweiterung 0400 — 1FFF für den VC-20



Die IC-Sockel werden so auf der Experimentierplatine angeordnet, daß die Adreßleitungen A0 – A9 an jeden Sockel herausgeführt werden können.

Da die 2114 RAMs zu 1024 x 4 Bit organisiert sind, werden immer zwei Bausteine für einen 1K x 8 Bit Block benötigt. Wir legen deshalb die obere Hälfte des Datenbusses an den ersten Baustein und die zweite Hälfte D0 – D3 an den zweiten Baustein. Die Anschlüsse am Baustein sind immer D0 – D3 bezeichnet (siehe dazu auch Pinout des Bausteins 2114).



Jetzt brauchen wir nur noch die Stromversorgung, welche auch vom VC-20 an Klemmen 21 und 22 abgenommen werden kann. Sie reicht gut für 3K RAM.

Das Read/Write Signal wird bei jedem Baustein an Anschluß 10 gelegt. Die Anschlüsse RAM1, RAM2 und RAM3 werden dem jeweiligen 1K x 8 Block zugeführt (jeweils zwei Chips zusammen).

Für eine Erweiterung über diese 3K hinaus, müssen Sie eine externe Dekodierung durchführen. Die Anschlüsse BLK1, BLK2, BLK3 und BLK5 welche an die Steckeranschlüsse 10 bis 13 herausgeführt sind, können für diese Dekodierung mit benutzt werden.

Ein 74LS138 im VC-20 (Baustein UB1) erzeugt an seinen Ausgängen die BLK0 – BLK7 Signale, von denen vier (wie vorher beschrieben) herausgeführt sind.

BLK0 wählt intern den unteren 8K Block aus

BLK1 wählt den 2. 8K Block (2000 – 3FFF)

BLK2 wählt den 3. 8K Block (4000 – 5FFF)

BLK3 wählt den 4. 8K Block (6000 – 7FFF)

Bis hier sind die 32K des unteren Bereiches festgelegt. Ab 8000 hat der VC-20 seinen Zeichengenerator, Farbregister, Ein-/Ausgabe Ports usw. Diese Funktionen werden intern mit BLK4 ausgewählt, welcher natürlich nicht herausgeführt ist (8000 – 9FFF = Block 5).

BLK5 wählt den 6. Block (A000 – BFFF). Dieses signal ist herausgeführt und speziell für die ROM-Steckkarten geeignet. Hier können die sogenannten Autostart-ROMs angesiedelt werden. D. h. steckt ein entsprechend programmierter ROM in einem Sockel mit dieser Adresse, springt der VC-20 sofort nach dem Einschalten von selbst in dieses Programm.

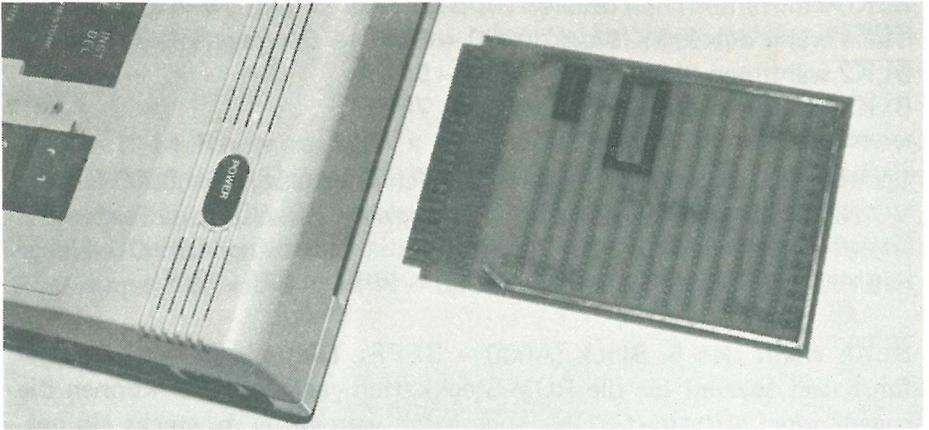
Für ein Autostart ROM müssen die ersten Zeilen im ROM wie folgt aussehen:

\$A004	\$41
\$A005	\$30
\$A006	\$C3
\$A007	\$C2
\$A008	\$CD

Die Adresse \$A000 und \$A001 enthalten die Zieladresse, wo das Programm nach RESET oder nach dem Einschalten hinspringen soll. Die Adressen \$A002 und \$A003 legen den nicht maskierbaren Interruptvektor des VC-20 fest.

Während eines Autostartes werden entweder das Betriebssystem, die Ausgangsports oder die VC-20 Sonderbausteine initialisiert. Also müssen die ersten fünf Befehle in einem Autostart-Programm wie folgt aussehen:

```
JSR $FD8D ; RAM Test
JSR $FD52 ; Betriebssystemvektoren setzen
JSR $FDF9 ; Ein/Ausgabe initialisieren
JSR $E518 ; Bildschirm initialisieren
CLI ; Unterbrechungen werden erlaubt
```



### Experimentierplatine

Sie sehen, daß Sie sich mit Hilfe dieser Informationen Ihren VC-20 auf einfachste Weise bis auf 32K RAM und noch zusätzliche 8K EPROM aufrüsten können.

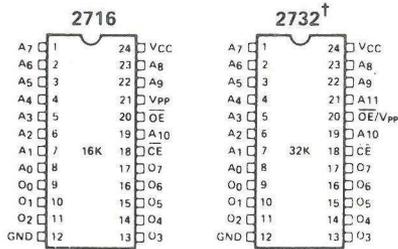
Verwenden Sie unsere RAM/ROM-Karte Nr. 615, so könnten Sie z. B. die Anfangsadresse von einem der beiden 8K Blöcke z. B. an 2000 Hex legen, indem Sie den Punkt A (oder B) auf der Platine mit BLK1 am Memory Expansion Stecker verbinden. Wenn Sie dann noch die Dekodierung innerhalb des entsprechenden Blockes auf 2K x 8 RAMs auslegen, haben Sie eine preiswerte Aufrüstung um 8K RAM ab 2000 Hex (siehe hierzu – Speichererweiterung für VC-20).

#### **Aufbau eines EPROM Sockels auf der Experimentierplatine**

Natürlich können Sie auch eine EPROM-Erweiterung für 2716 oder 2732 auf der Experimentierplatine (Best.-Nr. 652) von ELCOMP aufbauen.

Die Dekodierung kann mit Hilfe des BLK5 an der Adresse A12 bei 2K und 4K EPROMs und die Adresse A12 bei 8K Byte EPROMs muß jetzt auf die Experimentierplatine und an die entsprechenden Sockel gelegt werden. Alle Datenleitungen D0 – D7 müssen auch herangeführt werden (siehe Pinout).

## PIN CONFIGURATION



†Refer to 2732  
data sheet for  
specifications

BLK5 wird an Pin 20 (OE) gelegt. Pin 21 wird an +5V und Pin 18 an Masse gelegt (für 2K und 4K EPROMs). Bei 8K EPROMs müssen hier die Adressen A11 und A12 angelegt werden (A12 an Pin 21 und A11 an Pin 18).

Universal-Experimentierplatine für VC-20 ist vom ELCOMP-Platinen-service erhältlich. Best.-Nr. 4844 129,— DM

# VC-20

## Speichererweiterung

### VC-20 Speichererweiterung

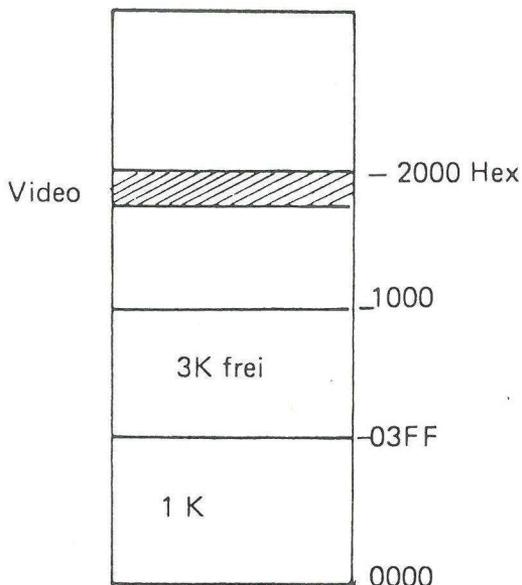
Der VC-20 von Commodore wird mit einer RAM-Kapazität von 5K in der Standardversion ausgeliefert.

Bald stellt sich jedoch heraus, daß man für anspruchsvolle Programme etwas mehr Speicher benötigt.



Um den Speicher des VC-20 sowohl im RAM- als auch im EPROM-Bereich zu erweitern, hat ELCOMP eine 16K RAM/ROM-Karte entwickelt. Auf dieser Karte können Sie 1K x 8 Bit, 2K x 8 Bit RAMs

oder auch 2K x 8 Bit EPROMs verwenden. Die Speicheraufteilung der Grundversion des VC-20 sieht wie folgt aus:



Um auf einen 8K RAM-Speicher aufzurüsten, muß der freie Speicherbereich von 0400 hex bis 0FFF mit 3K RAM (1 Baustein 2K x 8 Bit und 1 Baustein 1K x 8 Bit) bestückt werden.

Wir verwenden dazu den Block A auf unserer Karte. Für die 3K RAM benötigen wir

- 1 x RAM 1K x 8 4118 v. Mostek im Sockel A1
- 1 x RAM 2K x 8 M58725P im Sockel A2

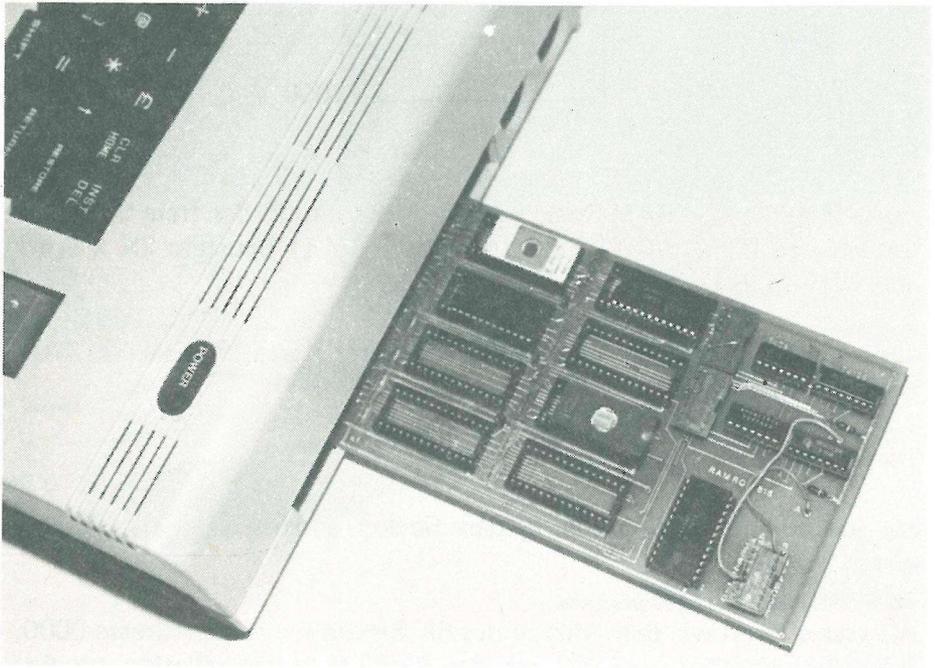
Wir dekodieren jetzt diese beiden Sockel entsprechend den Drahtverbindungen.

Als erstes legen wir den Anfang des 8K-Blocks A auf die Adresse 0000. Dies geschieht dadurch, daß wir den Punkt A auf der Platine mit Anschluß 0 auf der Plattform verbinden. Diese Verbindung erlaubt uns jetzt den Bereich von 0000 – 1000 hex = 4K mit Speicherbaustein zu

belegen.

0000 Hex	-----	
		1K RAM, bereits belegt im VC-20
0400	-----	
		1K RAM 4118
07FF	-----	} 1 x 2K RAM M58725P
0FFF	-----	

Wie wir jetzt die einzelnen Sockel im Feld A (1A – 4A) belegen, bleibt uns überlassen. Dadurch, daß wir aber nicht bei 0000 hex, sondern bei 0400 hex mit unserem ersten Speicherbaustein im Sockel 1A beginnen müssen, muß dies in den Verdrahtungsplattformen festgelegt werden. Die für den Block A zuständige Verdrahtung erfolgt auf der zugehörigen Plattform (siehe Bestückungsplan).

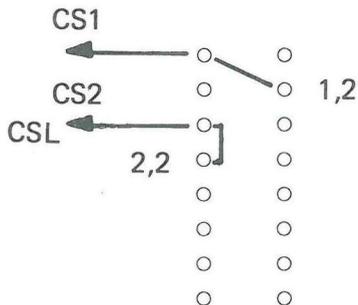


Hier werden jetzt die dekodierten Signale auf die Anschlüsse verteilt.

Socket 1A soll mit der Adresse 0400 hex beginnen, hierfür müssen wir CS1 mit 1,2 verbinden. 1,2 legt den Bereich von 0400 – 7FF fest und erlaubt uns ein 1K x 8 RAM zu verwenden.

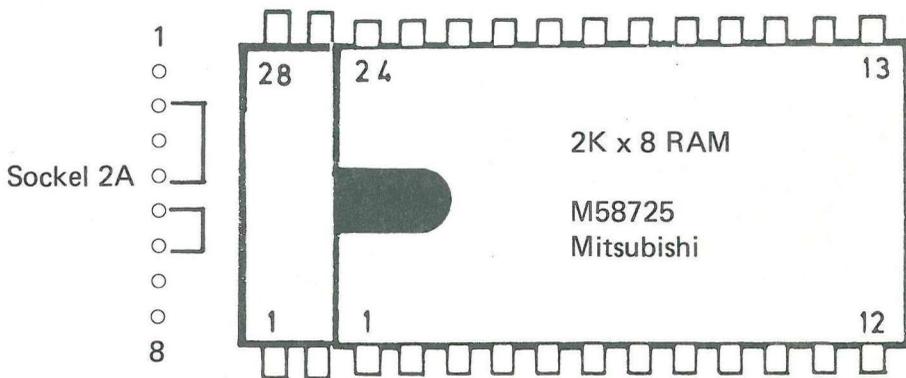
In Socket 2A soll nun ein 2K x 8 RAM gesteckt werden (Bereich 0800 – 0FFF). Dies bedeutet, daß wir CS2 mit 2,2 verbinden müssen (siehe Abbildung 3 unter der Überschrift "Die 16 K Byte RAMROM-Karte nach dem Byte-wide-Konzept").

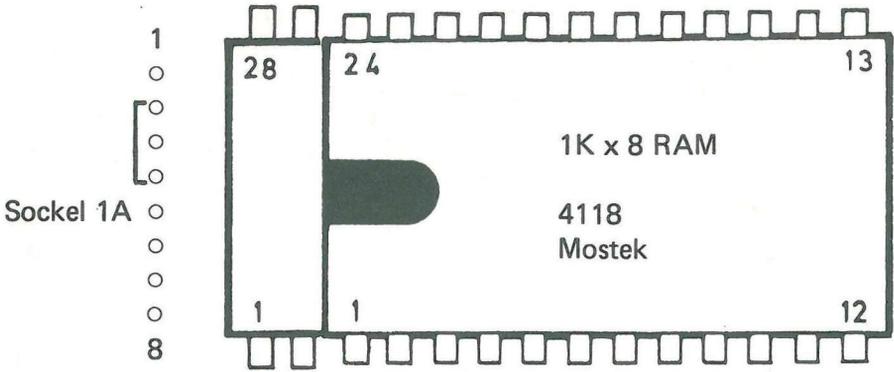
Ansicht der Lötplattform von oben



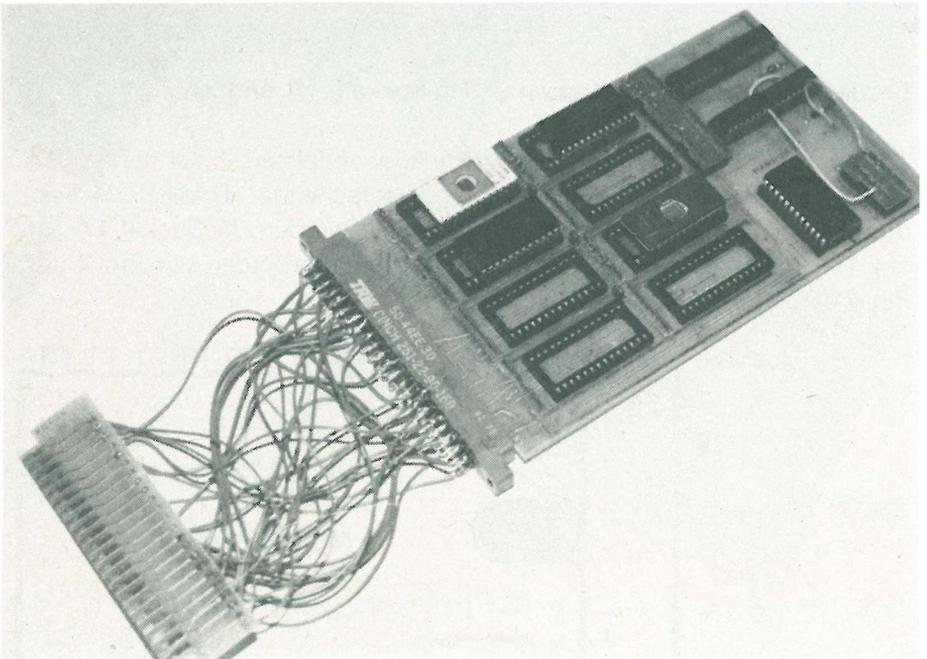
Festlegung der Anfangsadressen in den Sockeln 1A und 2A.

Als nächstes muß nun an den Typen-Auswahlleisten, links neben dem jeweiligen RAM/ROM-Sockel die entsprechende RAM-ROM-Type durch Einlöten einer Drahtbrücke festgelegt werden. In Socket 1A soll ein 1K x 8 Bit RAM gesteckt werden. Dazu verbinden wir Pin 2 und Pin 4 der Auswahlleiste.





Im Sockel 2A wollen wir ein 2K RAM verwenden. Hier wird nun Pin 2 mit Pin 4 und Pin 5 mit Pin 6 verbunden. Jetzt wäre unsere Speicherkarte für eine 3K RAM-Erweiterung präpariert. Was wir jetzt noch benötigen, ist ein Adapterkabel zum Anschluß der Platine an den VC-20 Erweiterungsstecker.



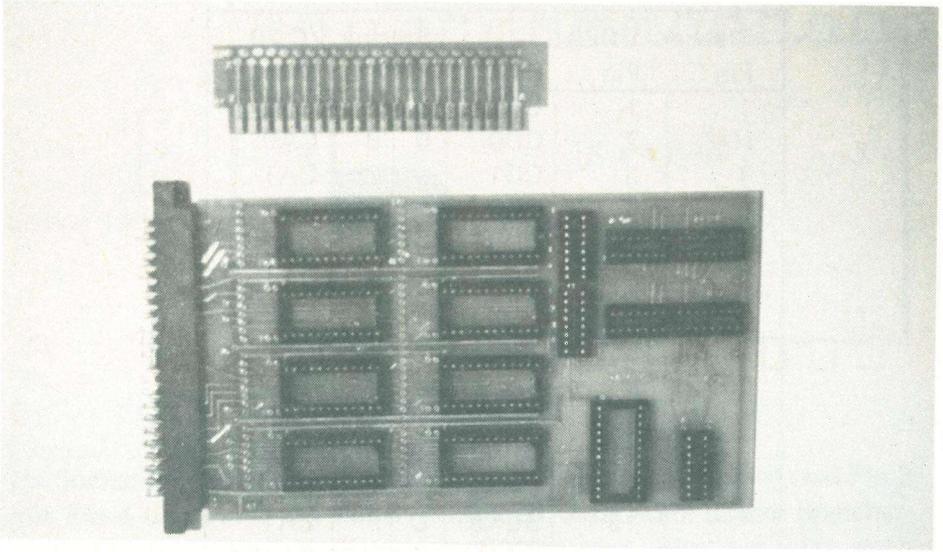
Stecker VC20			Stecker VC-20		
Pin	Pin	Beschr.	Pin	Pin	Beschr.
1	1	GND	A	A	GND
H	2	CD0	6	B	CA0
7	3	CD1	5	C	CA1
8	4	CD2	F	D	CA2
J	5	CD3	4	E	CA3
T	6	CD4	D	F	CA4
S	7	CD5	E	H	CA5
15	8	CD6	W	J	CA6
16	9	CD7	V	K	CA7
	10	<u>BLK1</u>	19	L	CA8
	11	<u>BLK2</u>	17	M	CA9
	12	<u>BLK3</u>	18	N	CA10
	13	<u>BLK5</u>	U	P	CA11
	14	<u>RAM1</u>	C	R	CA12
	15	<u>RAM2</u>	X	S	CA13
	16	<u>RAM3</u>		T	I/O2
	17	VR/W		U	I/O3
12	18	R/W CR/W	9	V	$\phi 2$
	19	<u>IRQ</u>		W	<u>NMI</u>
Y	20	CA15		X	<u>RESET</u>
20	21	+5V	B	Y	CA14
22	22	GND	Z	Z	GND

Die RAM/ROM-Karte benötigt eine  $\phi 2$  !

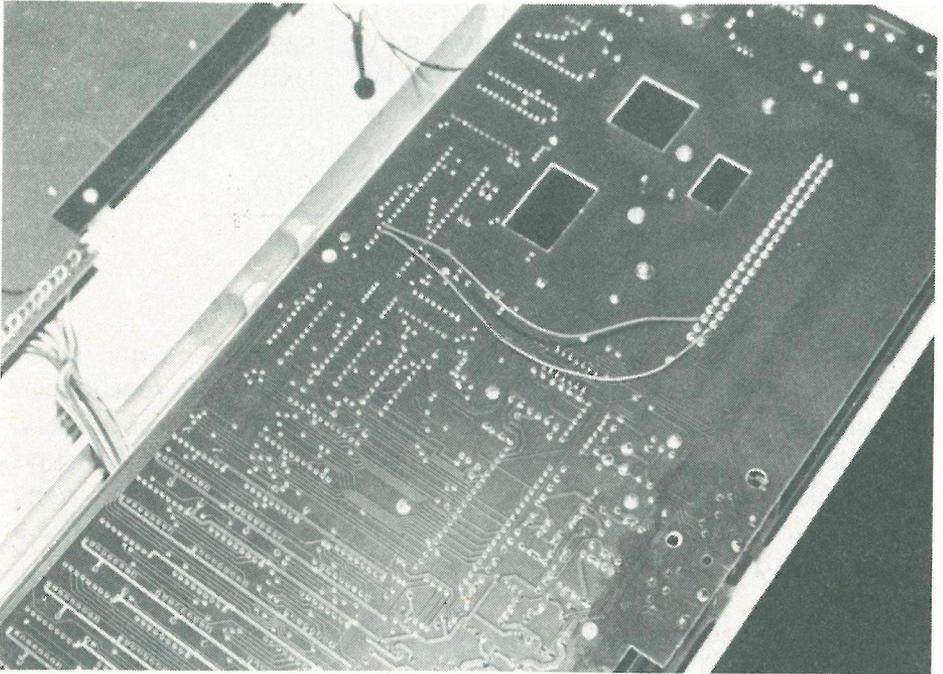
Dieser Adapter besteht aus einer kleinen Steckerplatine und einen 44poligen Stecker für die Karte. Man verbindet jetzt die zugehörigen Anschlüsse vom VC-20 mit dem entsprechenden Pin am Stecker (siehe Tabelle). Da beim VC-20 Expansionsstecker die Adressen CA14 und CA15 nicht auf die Leiste geführt sind, holen wir das jetzt nach. Dazu öffnen wir vorsichtig das Gehäuse (drei Schrauben), dann lösen wir die Platine von der Computergrundplatte.

Auf dem Expansionsstecker sind auch noch zwei Anschlüsse unbelegt, Pin Y und Pin 20.

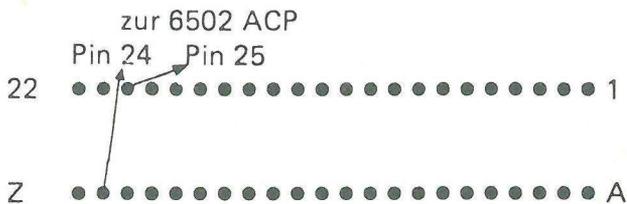
Von unten gesehen befinden sich diese in der rechten Hälfte der Platine.



**16K-Karte (oben im Bild der Adapter)**



**VC-20 geöffnet — Hauptplatine von unten gesehen**



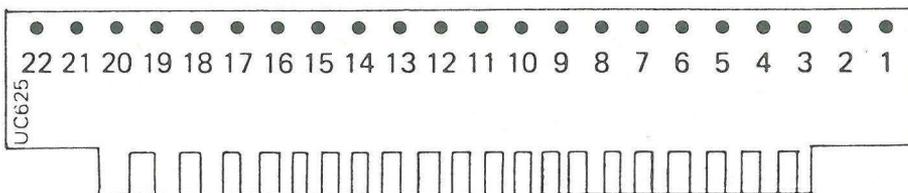
Leiterplatte von unten gesehen (Lötseite)

Wir verbinden Anschluß Y mit Pin 24 und Anschluß 20 mit Pin 25 am DIP-Gehäuse der 6502A CPU (siehe Bild). Jetzt sind alle Adressen und Datenleitungen am 44-poligen Erweiterungsstecker verfügbar und es lassen sich jetzt viele Erweiterungen durchführen, die vorher nicht so ohne weiteres möglich waren.

### Erstellung der Adapter-Platine

Bei der Zusammenschaltung von Platine und neuem Sockel ist besondere Vorsicht geboten.

Die kleine Adapterplatine VC625 wird so eingesteckt, daß sich die Zahlen oben befinden. Von links nach rechts 22–1. Im VC-20 ist die Numerierung genau entgegengesetzt.



Wir beginnen jetzt mit der Zählung auf der Adapterplatine mit 1 auf der rechten Seite oben. Wir löten jetzt die 30 Verbindungen nach folgender Checkliste ein:

VC-20 Adapter Pin	Platine Stecker Pin (S-44)	VC-20 Adapter Platine Pin	Platine Stecker Pin (S-44)
1	1	A	A
22	22	B	6
2	H	C	5
3	7	D	F
4	8	E	4
5	J	F	D
6	T	H	E
7	S	J	W
8	15	K	V
9	16	L	19
18	12	M	17
20	Y	N	18
21	20	P	U
		R	C
		S	X
		Y	B
		Z	Z
		V	9

Nachdem alle Verbindungen hergestellt sind, sollte alles noch einmal überprüft werden.

### Erweiterung des VC-20 um weitere 8K RAM/ROM oder EPROM

Der A-Block (8K) unserer Speicherkarte ist jetzt ausgeschöpft, da der noch zu verbleibende Bereich im VC-20 selbst belegt ist. Man könnte jetzt den Anfang des 8K Blockes B nach 2000 legen und von dort bis 3FFF um weitere 8K RAM oder EPROM aufrüsten. Eine andere Möglichkeit wäre ab A000 Hex mit einem 8K EPROM zu beginnen.

Programme, die bei A000 Hex beginnen, können als Autostartprogramme ausgelegt werden. Dies bedeutet, daß bei eingestecktem EPROM oder ROM nach dem Einschalten oder RESET der Computer an die Adresse \$A00 springt. In A000 wird das lower Byte und in A001 das higher Byte der Zieladresse abgelegt. (Nichtmaskierbarer

Interrupt Vektor des VC-20.)

Weiterhin müssen die Speicherzellen \$A004 bis \$A008 wie folgt belegt werden:

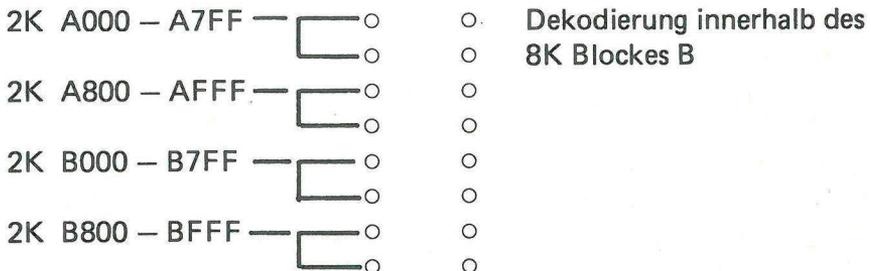
\$A004 ← \$41  
\$A005 ← \$30  
\$A006 ← \$C3  
\$A007 ← \$C2  
\$A008 ← \$CD

Die ersten fünf Befehle eines Autostart-Programmes müssen dann wie folgt aussehen:

JSR \$FD8D ; RAM Test  
JSR \$FD52 ; Setzen des Betriebssystemvektors  
JSR \$FDF9 ; I/O-Initialisierung  
JSR \$E518 ; Bildschirminitialisierung  
CLI ; Clear Interrupt

### Setzen des B-Blockes auf die Anfangsadresse A000

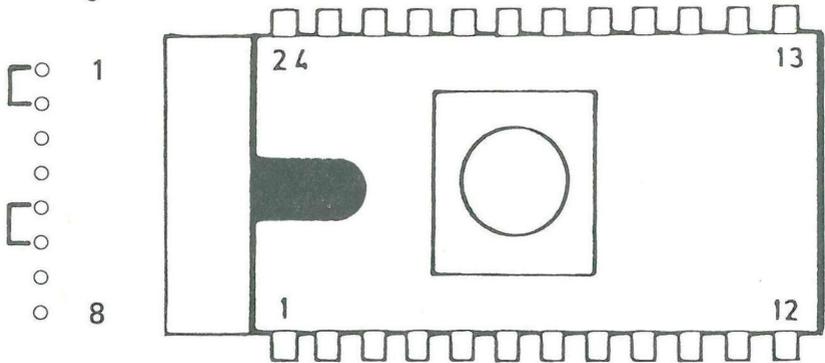
Wir verbinden den Anschluß B auf der Platine mit Anschluß A auf der Plattform für die Anfangsadresse (rechts neben dem 74154). Damit ist die Anfangsadresse für Sockel 1B mit A000 festgelegt. Angenommen, wir wollen in alle vier Sockel 1B, 2B, 3B, 4B ein 2716 EPROM einstecken, so müssen folgende Brücken auf der Chip-Selekt Plattform für B (obere Plattform) eingelötet werden.



Jetzt sind die Anfangsadressen der 4 EPROMs festgelegt. Damit wir 2716 in die einzelnen Sockel einstecken können, müssen

jetzt noch die entsprechenden Brücken links vor den Sockeln eingelötet werden.

Für 2716 gilt:



Diese Brücken müssen vor jedem der Sockel eingelötet werden, indem im Block B ein 2716 EPROM eingesteckt werden soll.

Dies sind nur einige Möglichkeiten, die Ihnen die RAMROM 615 Karte bietet. Sie könnten jedoch ebenso ab A000 Hex auch RAM-Bausteine einsetzen. Die Brücken müßten dann wie folgt links neben den RAMs angeordnet werden:



Wollten Sie die Konfiguration nicht ab Adresse A000 Hex, sondern ab 2000, so brauchen Sie nur die Verbindung von B nach 2, anstelle nach A zu legen. Alles andere bleibt wie gehabt.

# Die 16 K Byte RAMROM-Karte nach dem Bytewide-Konzept

## Die 16 K Byte RAMROM-Karte nach dem Bytewide-Konzept

Der Bau von Speichererweiterungen war noch nie so interessant wie heute. Gerade in diesen Tagen werden von einigen Herstellern Personalcomputer angeboten, die bereits nach dem ersten Einschalten schon nach einer Erweiterung des RAM-Bereiches "schreien".

Unsere Erfahrungen haben weiterhin gezeigt, daß wenn der "RAM-Hunger" dann gestillt ist, auch schnell der Appetit auf EPROMs aufkommt.

Aus diesem Grunde haben wir eine Speichererweiterungsplatine mit ganz besonderen Eigenschaften entwickelt.

Die neue im Europaformat aufgebaute Speichererweiterungsplatine erlaubt die Aufnahme von 16 K RAM oder 16 K EPROM oder auch EPROMs und RAMs gemischt. Als Philosophie liegt das Bytewide-Konzept zugrunde, welches es erlaubt 2716 EPROM-kompatible RAM-Bausteine wie den 4802 von Mostek usw. neben 2716 EPROMs im gleichen Sockel anwenden zu können. Weiterhin besteht die Möglichkeit, 4 K und 8 K EPROMs (2732 und 2764) einzusetzen.

Dies macht jede Speichererweiterung sehr flexibel, da Sie z. B. neben einer RAM-Erweiterung von 8 K noch ein Programmentwicklungspaket mit Monitor, Editor, Assembler etc. in EPROM auf der gleichen Platine mit unterbringen können. Natürlich können Sie diese Konfiguration später wieder völlig umwerfen und vielleicht 16 K EPROM oder 16 K RAM auf der gleichen Platine installieren.



Die doppelseitig beschichtete Platine verwendet den US – S44 Industrie-Standardbus. 44 Anschlüsse sind mit 1–22 und A–Z bezeichnet (ähnlich KIM, SYM, AIM, VC-20 ATARI etc.). Sie eignet sich zum direkten Einstecken in die ELCOMP–1 Erweiterungsplatine Nr. 606 zu 129,— DM. Über die Erweiterungsplatine können Sie die Karte mit fast allen gängigen 6502 Microcomputern verbinden.

Die Karte kann jedoch auch ohne die Erweiterungsplatine direkt mit einem 6502 Rechner verbunden werden. Hierbei müssen die entsprechenden Leitungen des Prozessors mit der an der Anschlußleiste verbunden werden.

Die 16 K Byte RAMROM-Karte ist in 2 8K Byte Blöcke A und B aufgeteilt. Jedem dieser Blöcke kann eine andere Anfangsadresse zugeordnet werden. Diese Anfangsadresse wird über einen 74154 aus den Adressleitungen A12 bis A15 dekodiert. Somit sind dafür die Adressen \$0000, \$1000, . . . , \$F000 festgelegt. Innerhalb dieser 8K Blöcke werden die Adressen über einen 74LS138 und einem 7408 in 1K und 2K Stufen aus den Adressleitungen A10 bis A12 und der ausgewählten Anfangsadresse dekodiert.

In Abbildung 1 ist die Dekodierung der RAMROM-Karte gezeigt. Die Dekodierung des 8K Blocks ist für A und B gleich. Die Verteilung der Sockel auf die Blöcke A und B, und die Numerierung zeigt Abbildung 2. Die dekodierten Signale für die 4K Grenzen können über eine DIL-Plattform unten rechts, die 2K und 1K dekodierten Signale über 2 DIL-Plattform zwischen Dekodierung und Speicher entnommen werden. Die obere gilt für Block B, die untere für Block A. Die Verteilung der Signale auf diese Anschlüsse zeigt Abbildung 3.

Dabei sind CS1 bis CS4 die Chip Select-Eingänge an den Sockeln 1 bis 4 (Pin 18 für 24 pol. IC, Pin 20 für 28 pol. IC), 2,1 bis 2,4 die 4 x 2K und 1,1 bis 1,8 die 8 x 1K dekodierten Adressen.

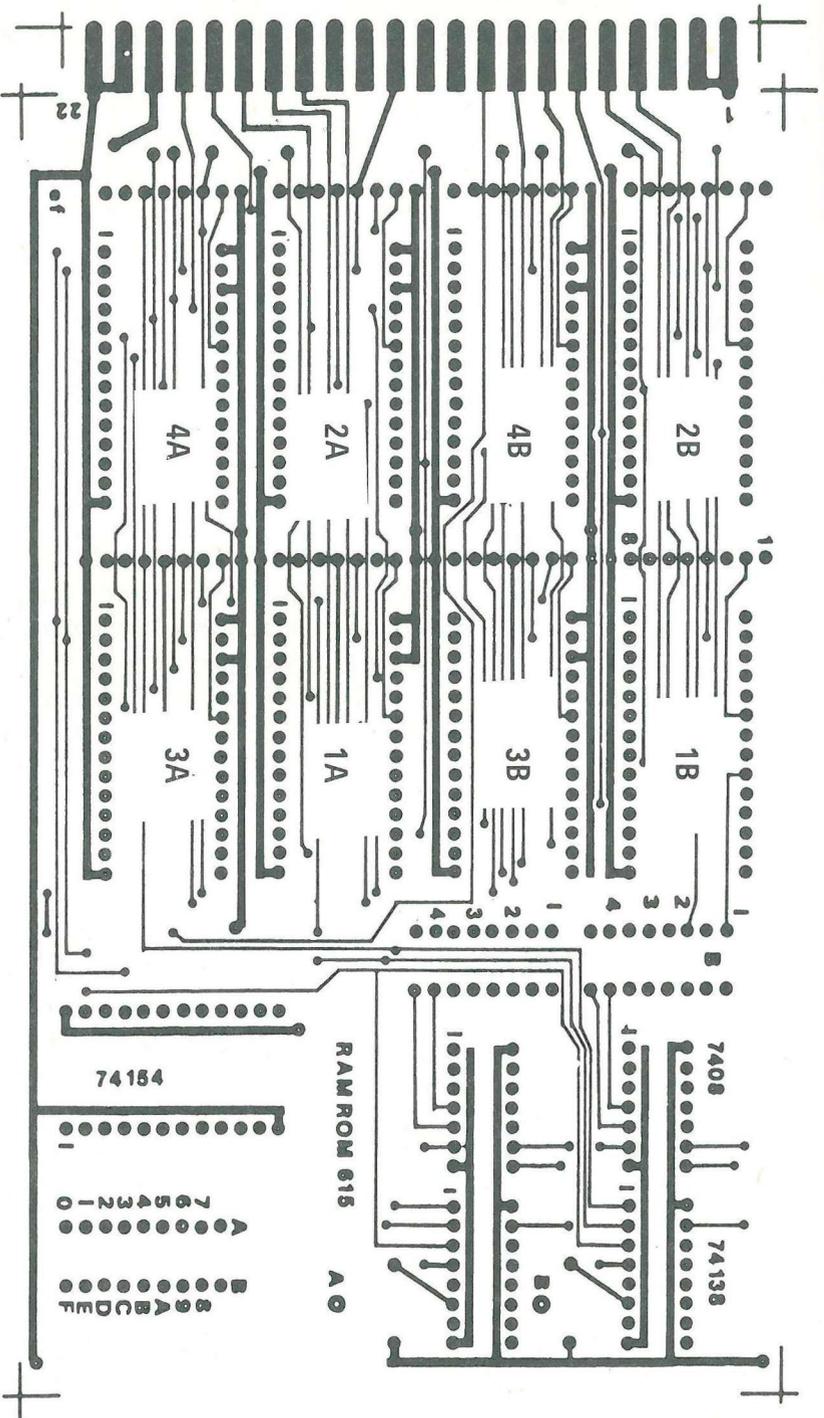
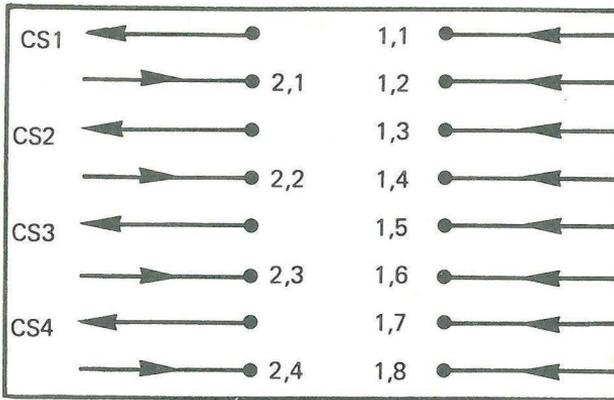


Abbildung 2  
Verteilung der Sockel auf die Blöcke A und B, und ihre Numerierung



**Abbildung 3**  
Verteilung der dekodierten Signale auf die Anschlüsse

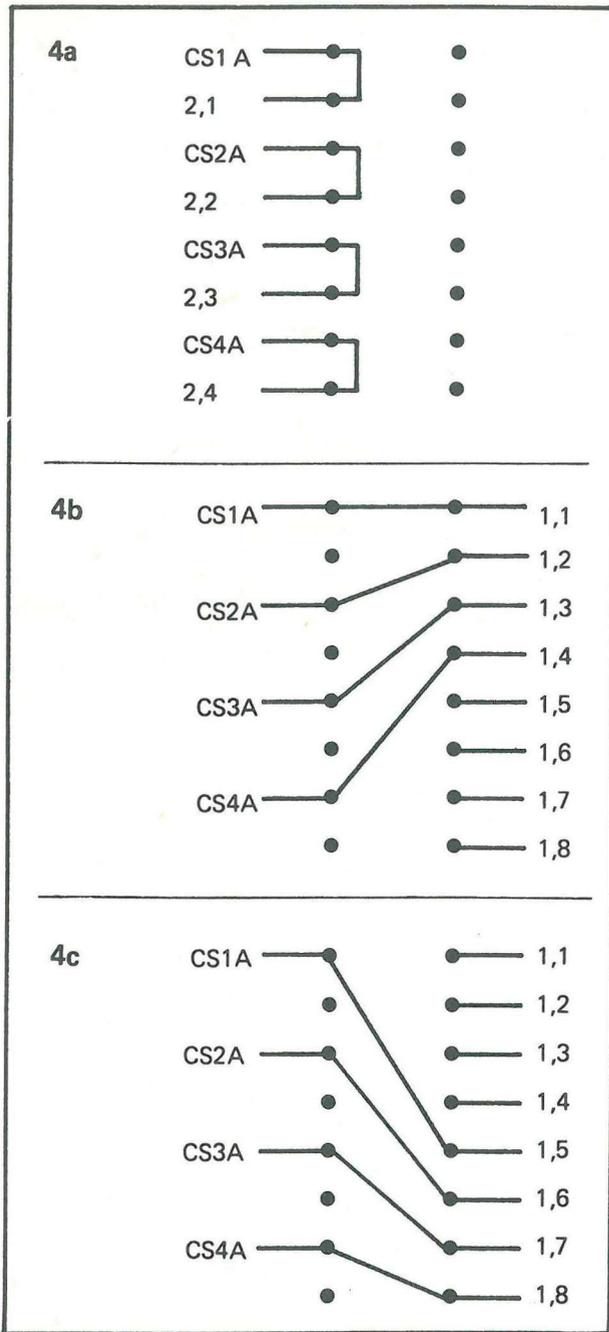
Beispiel für Block A:

Eingang A ist mit den dekodierten Adressen 2 und 3 verbunden. Dann werden an den Anschlüssen 2,1 bis 2,4 die Adressen

2,1 =	\$2000	—	\$27FF
2,2 =	\$2800	—	\$2FFF
2,3 =	\$3000	—	\$37FF
2,4 =	\$3800	—	\$3FFF

und an den Anschlüssen 1,1 bis 1,8 die Adressen

1,1 =	\$2000	—	\$23FF	
1,2 =	\$2400	—	\$27FF	
1,3 =	\$2800	—	\$2BFF	
1,4 =	\$2C00	—	\$2FFF	
1,5 =	\$3000	—	\$33FF	
1,6 =	\$3400	—	\$37FF	
1,7 =	\$3800	—	\$3BFF	
1,8 =	\$3C00	—	\$3CFF	dekodiert.



**Abbildung 4**  
 Verbindungen der  
 dekodierten Signale  
 mit den Chip-Select  
 Eingängen

Belegt man nun die Sockel mit 2 x 4K EPROM ode RAM, dann sind folgende Brücken einzulöten (Abbildung 4a). Die Anfangsadresse von Sockel 1A ist dann \$2000, von Sockel 2A = \$2800, von Sockel 3A = \$3000 und von Sockel 4A = \$3800. Belegt man diese Sockel mit 4 x 1K RAM, dann sind folgende Verbindungen einzulöten (Abbildung 4b). Dann liegen die Anfangsadressen für die Sockel bei

- 1A = \$2000
- 2A = \$2400
- 3A = \$2800
- 4A = \$2C00

Sollen die Adressen zwischen \$3000 und \$3FFF liegen, so sind die Verbindungen nach Abbildung 4c einzulöten.

Bei den Sockeln liegen die ungeraden Adressen immer bei den unteren Anschlüssen (2,3 2,4 und 1,5 bis 1,8), die geraden Adressen immer bei den oberen Anschlüssen (2,1 2,2 und 1,1 bis 1,4). Verbindet man also A mit den Anschlüssen 5 und 6 des Dekodierfeldes so erhält man folgende Anfangsadressen

- |              |              |
|--------------|--------------|
| 2,1 = \$6000 |              |
| 2,2 = \$6800 |              |
| 2,3 = \$5000 |              |
| 2,4 = \$5800 | und          |
| 1,1 = \$6000 | 1,5 = \$5000 |
| 1,2 = \$6400 | 1,6 = \$5400 |
| 1,3 = \$6800 | 1,7 = \$5800 |
| 1,4 = \$6C00 | 1,8 = \$5C00 |

In Abbildungen 5, 6, 7, 8, 9 sind nun die Verbindungen der Sockel für verschiedene Beispiele angegeben.

Beispiel 1 (Abbildung 5):

Sockel 1A–4A 4 x 2K EPROM/RAM, Anfangsadresse \$4000

Sockel 1B–4B 4 x 2K EPROM RAM, Anfangsadresse \$8000

- |                     |             |             |
|---------------------|-------------|-------------|
| Adressenverteilung: | 1A = \$4000 | 1B = \$8000 |
|                     | 2A = \$4800 | 2B = \$8800 |
|                     | 3A = \$5000 | 3B = \$9000 |
|                     | 4A = \$5800 | 4B = \$9800 |



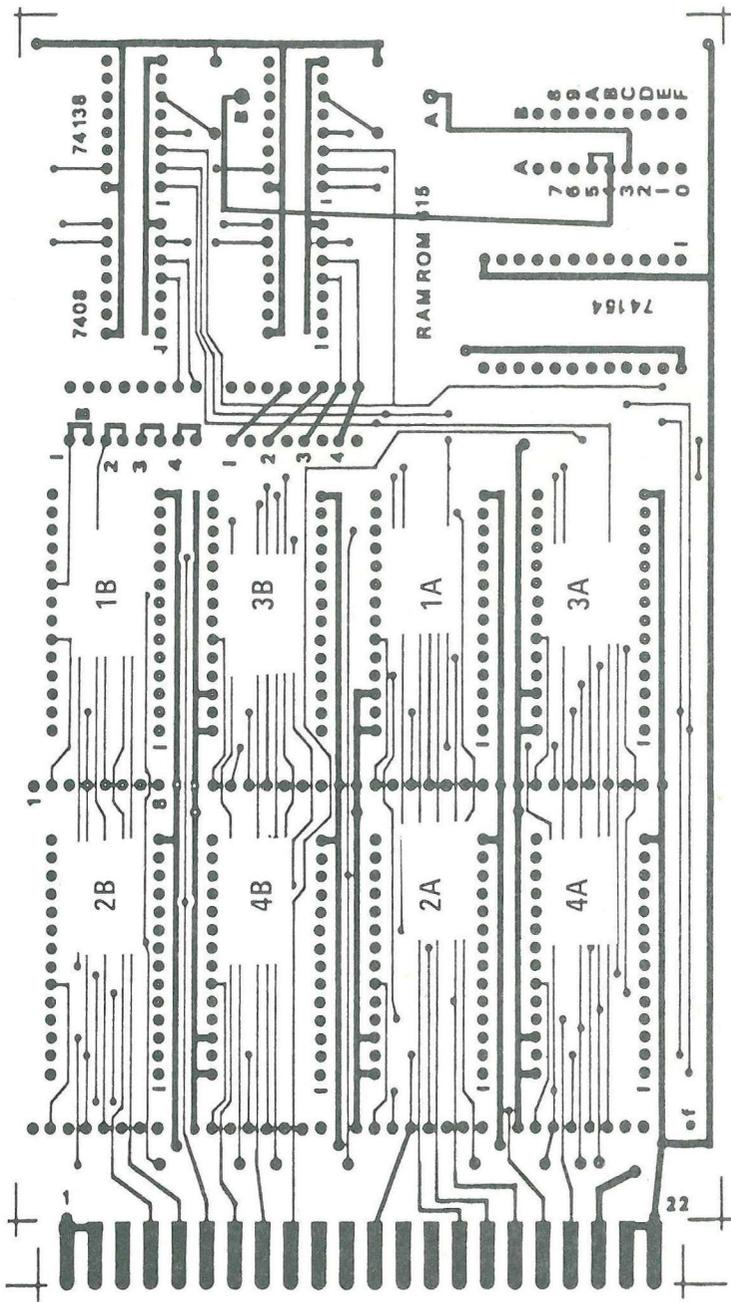


Abbildung 6

Beispiel 2 (Abbildung 6):

Socket 1A—4A 4 x 1K RAM, Anfangsadresse \$3000

Socket 1B—4B 4 x 2K EPROM/RAM, Anfangsadresse \$4000

Adressenverteilung:

1A = \$3000	1B = \$4000
2A = \$3400	2B = \$4800
3A = \$3000	3B = \$5000
4A = \$3800	4B = \$5800

Beispiel 3 (Abbildung 7):

Socket 1A—4A 4 x 2K EPROM/RAM, Anfangsadresse \$3000

Socket 1B—4B 4 x 2K EPROM/RAM, Anfangsadresse \$5000

Adressenverteilung:

1A = \$4000	1B = \$6000
2A = \$4800	2B = \$6800
3A = \$3000	3B = \$5000
4A = \$3800	4B = \$5800

Ist ein 8K-Block mit 4 x 2K RAMs bestückt, so ist es unerheblich, ob die dekodierte Anfangsadresse z. B. \$5000 am Socket 1A oder 3A anliegt. Bei EPROMs ist dies anders. Hier muß sichergestellt sein, daß der dekodierte Adressbereich am Socket mit dem Adressbereich des Programms im EPROM übereinstimmt, damit dies lauffähig bleibt.

Beispiel 3 (Abbildung 8):

In Beispiel 3, Version 2 (Abbildung 8) sind die Socket in aufsteigender Reihenfolge adressiert.

Adressenverteilung:

1A = \$3000	1B = \$5000
2A = \$3800	2B = \$5800
3A = \$4000	3B = \$6000
4A = \$4800	4B = \$6800

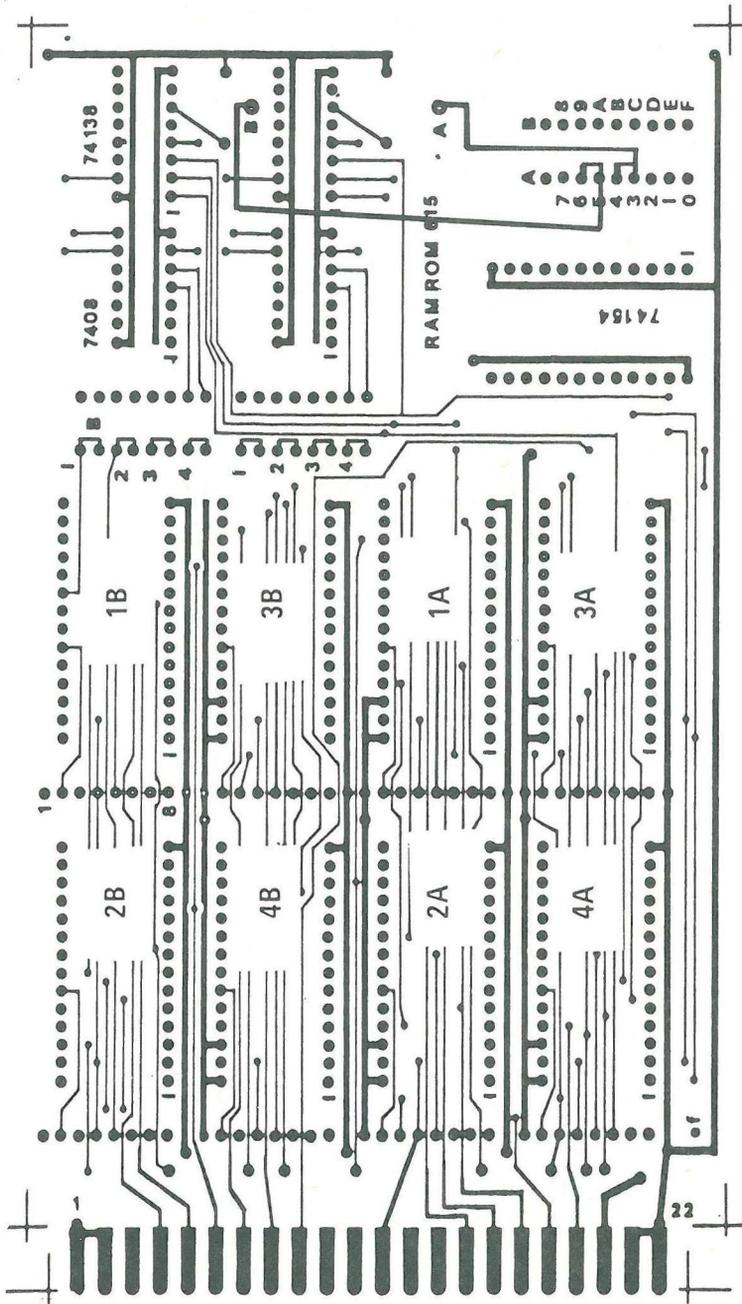


Abbildung 7



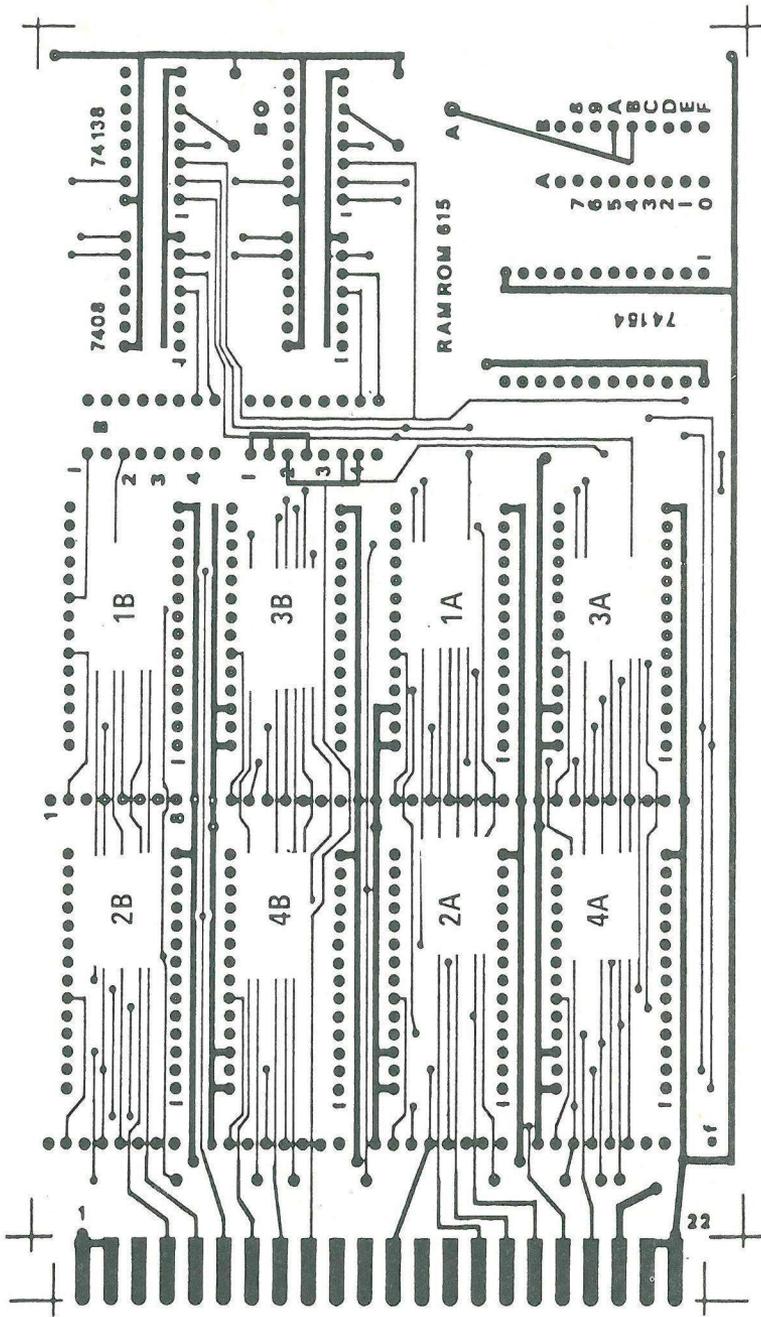


Abbildung 9

Beispiel 4 (Abbildung 9):

Socket 1A, 2A 2 x 4K EPROM, Anfangsadresse \$A000, \$B000

Adressverteilung:     1A = \$A000  
                           2A = \$B000

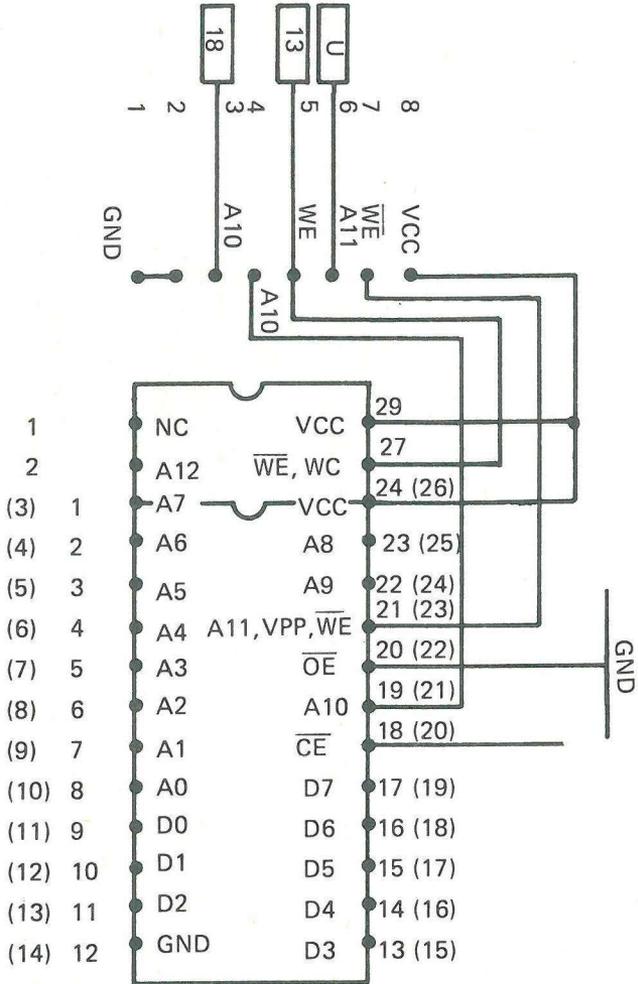


Abbildung 10  
 Verteilerleiste mit Socket

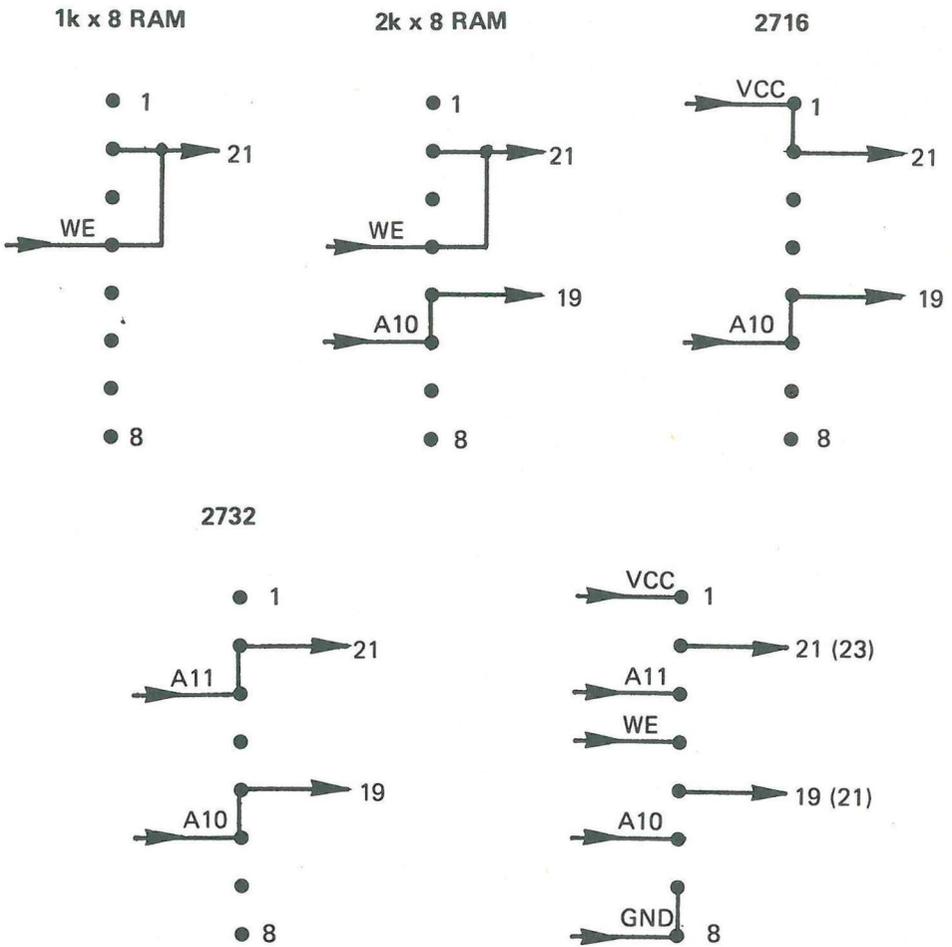
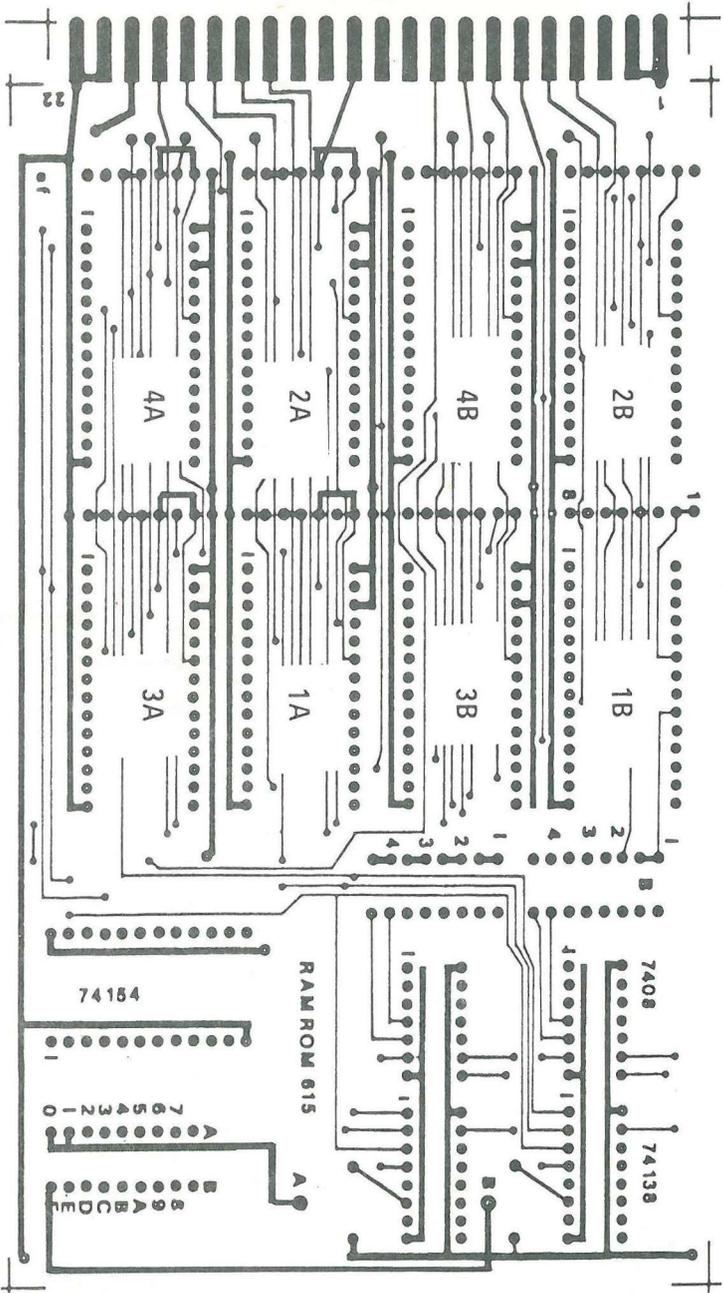


Abbildung 11

Verbindungen der Verteilerleiste mit dem Sockel für die verschiedenen Speicherarten

Bei jedem einzelnen Sockel muß nun noch durch Verdrahtung bestimmt werden, welches Bauteil (1K, 2K RAM oder 2K, 4K EPROM) er aufnehmen kann. Dazu dient eine Verteilerleiste, die bei jedem Sockel vorhanden ist. In Abbildung 10 ist ein Sockel mit der Verteilerleiste herausgezeichnet. An die Leiste gelangen von außen die Signale A11, WE, A10 und von Sockel die Anschlüsse 19 (21) und 21



Socket 1A-4A 4 x 2K RAM  
 Socket 1B 1 x 2K EPROM 2716

Abbildung 12

(23). Die Angabe in Klammern bezieht sich auf den 28 poligen IC. In Abbildung 11 sind nun die Verbindungen für 1K x 8, 2K x 8 RAM und für 2716, 2732 EPROM zusammengestellt.

Alle OE-Leitungen sind an Masse gelegt.

Als letztes Beispiel ist noch eine Speicherbelegung mit 8K RAM von \$000 bis \$3FFF und 2K EPROM von \$F800 – \$FFF angegeben. Sockel 1A bis 4A enthalten RAM, Sockel 1B einen 2716 (Abbildung 12).

In der letzten Abbildung ist noch die Belegung der Anschlußleiste angegeben. Sie entspricht in etwa dem S-44E Bus und ist in dieser Form auch auf der ELCOMP-1 Platine vorhanden.

### Anschlußbelegung der RAMROM-Karte

	S--44E		S--44E
1	GND	A	GND
2	GND	B	A14
3	NC	C	A12
4	A3	D	A4
5	A1	E	A5
6	A0	F	A2
7	D1	H	D0
8	<u>D2</u>	J	D3
9	ø2	K	NC
10	NC	L	NC
11	NC	M	NC
12	R/W	N	NC
13	NC	P	NC
14	NC	R	NC
15	D6	S	D5
16	D7	T	D4
17	A9	U	A11
18	A10	V	A7
19	A8	W	A6
20	VCC	X	A13
21	GND	Y	A15
22	GND	Z	GND

## **Bestückungsplan und Bestückungsanleitung**

Im folgenden Bild sehen sie die 16K RAMROM-Karte und die Lage der Bauteile auf der Platine. Eine unbestückte, gebohrte und durchkontaktierte Leiterkarte kann vom ELCOMP Platinenservice unter der Best.-Nr. 615 für 149,- DM incl. MwSt. bezogen werden.

Zur Verwendung von EPROMs und EPROM-kompatiblen RAMs sind 28 polige Sockel vorgesehen.

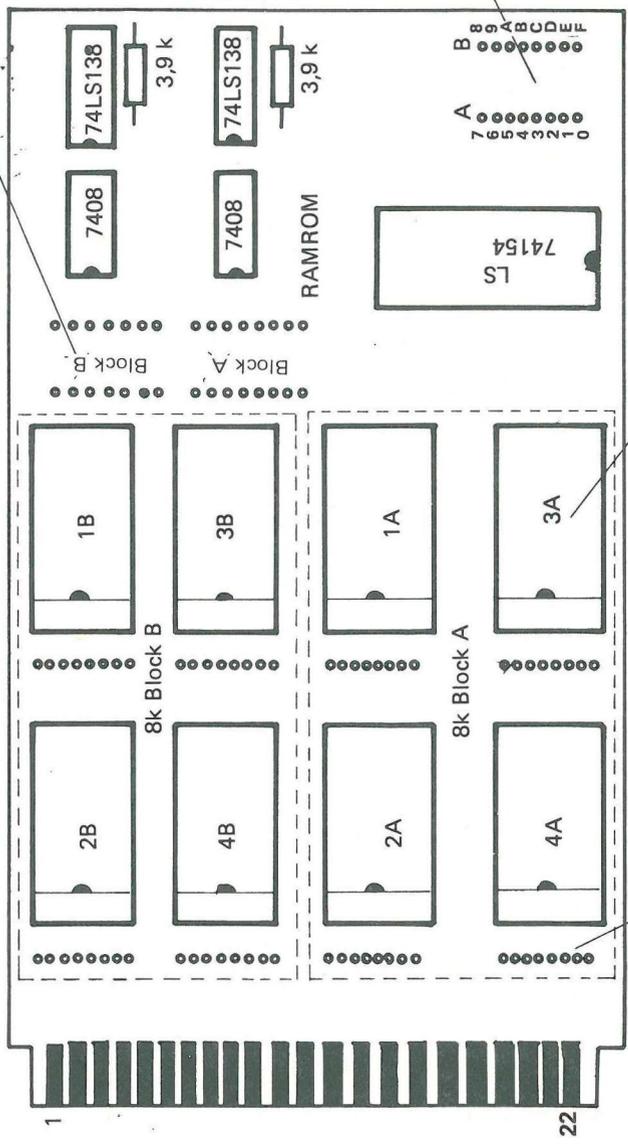
Werden 2716 oder andere 24 Pin Typen eingesteckt, bleiben links auf dem Sockel vier Pins frei. Links vor jedem Sockel sehen Sie acht Lötunkte. Hier kann eine Leiste eingesteckt werden, die dann je nach verwendetem IC durch Brücken codiert wird (siehe entsprechende Hinweise und Beispiele in diesem Artikel).

Für die Dekodierung der jeweiligen 8K Blöcke befinden sich rechts oben hinter den 28 Pin Sockeln zwei Reihen mit je 16 Lötunkten. Hier können Sie zwei 16 Pin DIL Sockel einlöten. In diese Sockel steckt

Hier können Sie zwei 16 Pin DIL-Sockel einlöten. In diese Sockel steckt man am besten eine DIL-Plattform. Auf diese Plattform (Adapter) können dann die gewünschten Brücken eingelötet werden.

Rechts unten in der Ecke befindet sich die 1K Grenzen Dekodierung. Auch hier empfiehlt es sich, einen DIL-Sockel einzulöten und die Decodierung auf einer Plattform vorzunehmen.

Dekodierung innerhalb der jeweiligen 8k Blöcke



Die 7408 Bausteine müssen als Standard-TTL-Bausteine ausgeführt werden

1k Grenzen Dekodierung

EPROM 24 Pin  
EPROM 28 Pin

Auswahlleiste für die verschiedenen RAM- und EPROM-Typen

Bestückungsplan: Ansicht von der Beschriftungsseite her

# Ein-/Ausgabe - Programmierexperimente mit dem VC-20

## Ein-/Ausgabe-Programmierexperimente mit dem VC-20

Der VC-20 bietet mit seinem, von außen leicht zugänglichen Ein-/Ausgabeport ideale Möglichkeiten für Programmierexperimente. Dieser Vorteil ist unter anderem dem recht interessanten Interfacebaustein 6522 zu verdanken, der sich im VC-20 befindet. Auf diesen Baustein wollen wir jetzt kurz eingehen.

Der 6522, ein universeller Interface-Adapter-Baustein (VIA), ist eine Weiterentwicklung des 6520 Peripheral Interface Adapters. Der 6520 ist voll hardwarekompatibel zum bekannten 6820 von Motorola.

Der 6522 besitzt die gleichen Eigenschaften wie der 6520, hat jedoch zusätzlich zwei leistungsfähige Zeitgeber, ein internes Schieberegister (seriell/parallel und parallel/seriell) sowie zwischengespeicherte Eingänge. Die Steuerung peripherer Geräte kann durch zwei bidirektionale Ports erfolgen (PA2 – PA7 und PB0 – PB7). Die Ports PB0 – PB7 sind beim VC-20 von außen über eine Steckerleiste leicht zugänglich. Als Stecker können verwendet werden: CHINCH 251-12-90-160 für User Port. PA2 – PA7 sind auch auf den User Port herausgeführt. Sie werden jedoch für den Joystick, Lichtgriffel und Cassetten Ein/Aus benötigt. PA7 wird für SERIALATN OUT zuerst über einen 7406 Puffer geführt und kann nur in eine Richtung verwendet werden.

Jeder dieser Ports kann vom Programm her als Eingabe oder Ausgabe kanal geschaltet werden. Einzelne Ein-/Ausgabeleitungen können weiterhin durch die Zeitgeber gesteuert werden, da man vom Programm

her, sehr leicht Rechteckwellen erzeugen kann. Man kann aber auch auf die gleiche Weise von außen kommende Impulse zählen und ins Programm übernehmen.

Die Kontrolle dieser Funktionen geschieht durch interne Register des 6522 Bausteins (näheres hierzu im Buch Nr. 118 vom Hofacker-Verlag).

Wir können jetzt schon erkennen, welche Möglichkeiten es hier gibt, sich bei Programmexperimenten "auszutoben". Natürlich sind die I/O-Ports des 6522 CMOS und TTL-compatibel. D. h., Sie können direkt TTL-Gatereingänge oder CMOS-Gatereingänge ansteuern.

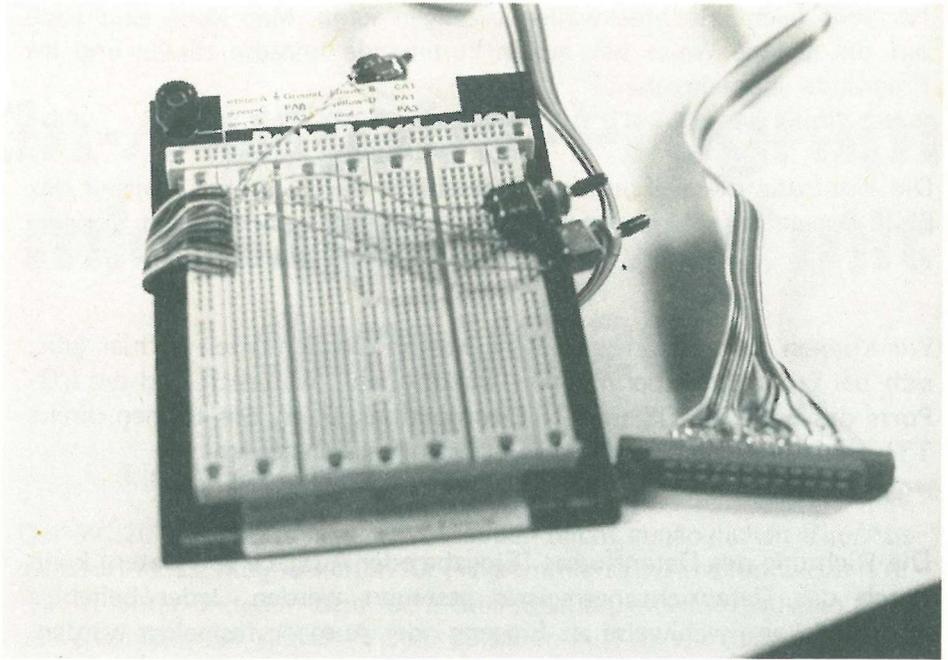
Die Richtung des Datenflusses (Eingabe oder Ausgabe von Daten) kann durch das Datenrichtungsregister gesteuert werden. Jeder beliebige Anschluß kann wahlweise als Eingang oder Ausgang festgelegt werden.

Am User-Port Ihres VC-20 finden Sie dann an der unteren Klemmleiste noch die Anschlußbezeichnungen CB1 und CB2. Diese sind auch direkte Verbindungen zum VIA # 1 6522 im VC-20.

CB1 ist eine Handshake Leitung (Quittungsleitung) für eingehende Signale. Sie kann auch als Flankendetektor verwendet werden. CB2 arbeitet ähnlich wie CB1, kann aber zusätzlich noch als Eingang oder Ausgang des Schieberegisters im 6522 programmiert werden. Sie werden über das periphere Kontrollregister (PCR) gesteuert. Wie man mit diesem Ausgabeport recht interessante Experimente durchführen kann, wollen wir im folgenden Teil zeigen. Es soll nur eine Anregung und ein Anstoß für Ihre eigenen weiteren Experimente sein.

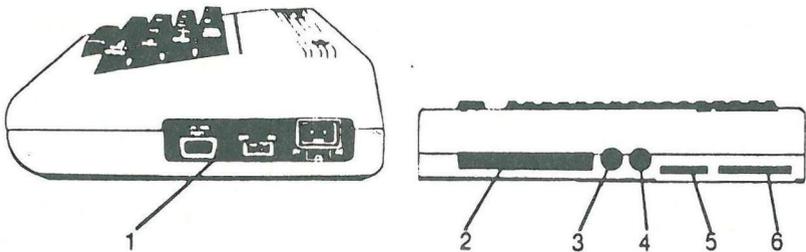
### **Anschluß einer Universal-Experimentierplatine**

Für den externen Aufbau von kleinen Experimenten eignet sich ein Experimentierboard (s. Bild) mit einem Anschlußkabel recht gut.



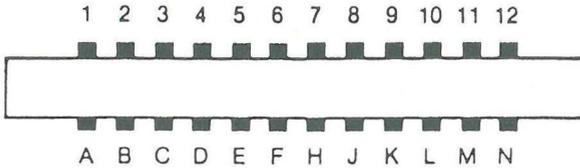
Über ein Flachbandkabel schließt man das Board an den User-Port an (PB0 – PB7).

#### Anschlüsse des VC-20



- 1) Joystick
- 2) Speichererweiterung
- 3) Audio und Video
- 4) Serielle Schnittstelle
- 5) Cassette
- 6) User Port für unsere Experimente

## USER I/O



PIN #	TYPE	NOTE	PIN #	TYPE	NOTE
1	GND	100mA MAX.	A	GND	
2	+5V		B	CB1	
3	RESET		C	PB0	
4	JOY0		D	PB1	
5	JOY1		E	PB2	
6	JOY2		F	PB3	
7	LIGHT PEN		H	PB4	
8	CASSETTE SWITCH	J	PB5	100mA MAX.	
9	SERIAL ATN IN	K	PB6		
10	+9V	L	PB7		
11	GND	M	CB2		
12	GND	N	GND		

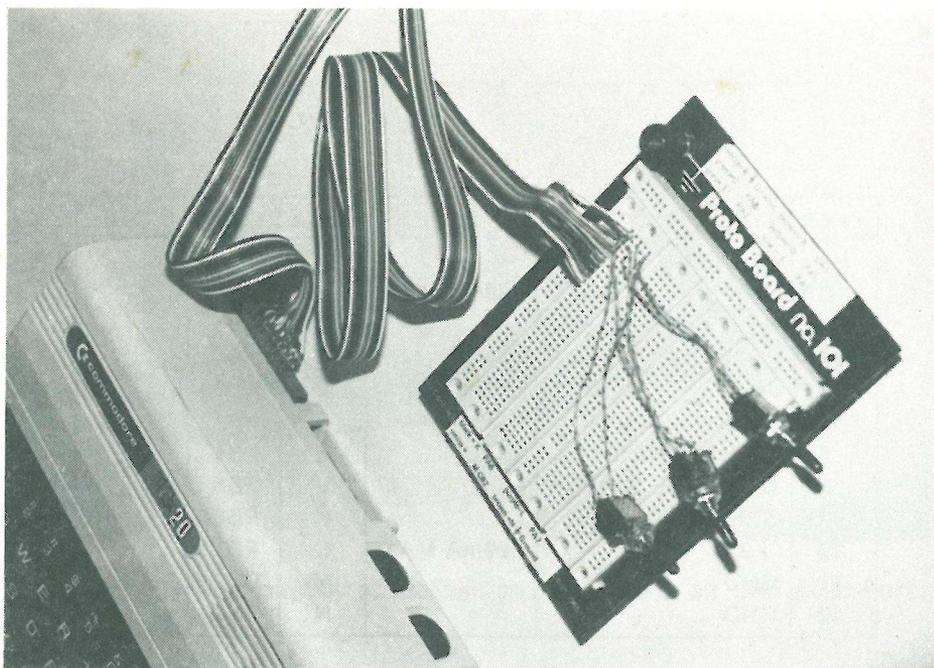
Der User Port ist von außen zugänglich. Die Kontakte sind auf der Leiterbahn oben und unten angebracht. Für Verbindungen sollte man auf jeden Fall einen Stecker aufstecken (CINCH 251-12-90-160 Typ 50 24EE-30). Dort kann man dann das Flachbandkabel anlöten und mit dem Universal Experimentierboard verbinden.

**Achtung ganz wichtig !**

Jegliches Aufstecken und Abziehen von Steckern an sämtlichen externen Anschlüssen des VC-20 darf nur bei abgeschaltetem Gerät erfolgen !!!

Wir haben uns die Anschlußbezeichnungen mit Farbe des Verbindungsdrahtes oben am Board vermerkt. So wissen wir auf den ersten Blick, mit welchem Anschluß wir es zu tun haben. Wir können jetzt Schalter anbringen und von außen Informationen in den VC-20 geben. In umgekehrter Richtung können wir jedes gewünschte 8 Bit Muster vom

VC-20 her über den Port nach außen geben.



Hierzu haben wir ein kleines Programm geschrieben, welches Ihnen die Arbeit mit dem Experimentierboard erleichtert.

### **Programmbeschreibung:**

Das kleine Hilfsprogramm erlaubt nun das Abfragen des Ports und die Informationsauswahl. Nach dem Einlesen und Eingabe von RUN wird gefragt, ob wir die acht, als Eingänge geschalteten I/O-Leitungen, abfragen wollen oder ob wir über alle 8 Leitungen Informationen hinausgeben möchten. Entscheiden Sie sich für das Erste, zeigt Ihnen der Computer die offenen Eingänge (log. 1) mit  $255 = 11111111$ . Jetzt können Sie beim Experimentieren die Eingänge verändern und haben die Information auf dem Bildschirm. Im zweiten Falle können Sie eine gewünschte Information (8 Bit binär) an die Ports ausgeben. Die Eingabe erfolgt dezimal. Hiermit lassen sich dann recht praktische Steuerschaltungen programmieren.

Ein weiteres, wichtiges Hilfsmittel, welches Sie bei solchen Experimenten gut gebrauchen können, ist ein Logiktester. Mit diesem Tester können Sie leicht die einzelnen Logikpegel in der Experimentierschaltung und am Port nachkontrollieren.

Wir wünschen Ihnen beim Experimentieren viel Erfolg !

```
5 REM COPYRIGHT BY
7 REM WINFRIED HOFACKER
10 PRINT"||DIESES PROGRAMM"
12 PRINT"ZEIGT DEN STATUS"
14 PRINT"DES USER PORTS IN"
16 PRINT"DEZIMAL UND BINAER"
20 PRINT"|"
39 S$(0)="0":S$(2)="1"
40 GOTO 200
41 POKE 37136,0
60 PRINT"DEZIMAL      ";PEEK(37136)
61 PRINT:PRINT"BINAER"
62 P=PEEK(37136)
63 GOTO 90
65 FOR I=1 TO 8
70 PRINT P AND 1;:P=P/2
80 NEXT I
81 PRINT"BINAER"
90 PRINT"||"ABS((PAND128)=128);
91 PRINTABS((PAND64)=64);
92 PRINTABS((PAND32)=32);
93 PRINTABS((PAND16)=16);
94 PRINTABS((PAND8)=8);
95 PRINTABS((PAND4)=4);
96 PRINTABS((PAND2)=2);
97 PRINTABS((PAND1)=1);
98 FOR V=1 TO 3000:NEXT
110 PRINT"||INHALT DES DATENRICHTUNGS"
111 PRINT"REGISTERS DR";PEEK(37138)
112 FOR X=1 TO 3000:NEXT
120 GOTO 60
200 PRINT"ABFRAGEN='1',AUSGABE='2'"
210 INPUT N
220 IF N=1 THEN 41
```

```

230 IF N=2 THEN 300
300 REM AUSGABE VON INFORMATIONEN
305 POKE 37138,255
310 PRINT"WELCHE AUSGANGSBITS"
330 INPUT K
340 IF K>255 THEN 310:GOTO 300
350 POKE 37136,K
370 L=PEEK(37136)
380 GOTO 60

```

READY.

### Aufbau und Arbeitsweise

Zuerst verbinden Sie das Board (MKS Bekatron) mit einem Flachkabel und dem CINCH-Stecker 251-12-90-160/50-24 EE-30 für den User Port.

Dann können Schalter und die Aufkleber mit den einzelnen Bezeichnungen angebracht werden. VC-20 ausschalten und Stecker so auf den User Port aufsetzen, daß die Anschlußreihe mit der Zahlenbezeichnung 1 - 12 nach oben zeigt. Nun den VC-20 einschalten und Programm eingeben. Sie können jetzt zwischen den Betriebsarten "Abfragen" oder "Ausgabe" wählen. Im ersten Falle können Sie den Port abfragen, welche Signale von außen anstehen. (Offen = log. 1 = High). Kurzschluß des Portausganges nach Masse = log. 0 = Low). In der Betriebsart "Ausgabe" können Sie Signale über die Ports per Programm ausgeben. Es wird eine externe 5V-Stromversorgung empfohlen. Auf dem Experimentierboard können Sie jetzt Ihre eigenen Schaltungen leicht aufbauen. Im Notfall können +5V an Pin 21 des Expansions-Port abgezweigt werden. Masse liegt dort an Pin 1.

### Ein-/Ausgabeprogrammierung mit dem VC-20

Der VC-20 besitzt, wie die meisten Anwender wissen, einen User Port zum Anschluß externer Geräte. Für provisorische und schnelle Ein-/Ausgabeprogrammierung eignet sich dieser parallele Ein-/Ausgabeport bestens. Er läßt sich sehr einfach handhaben und programmieren.

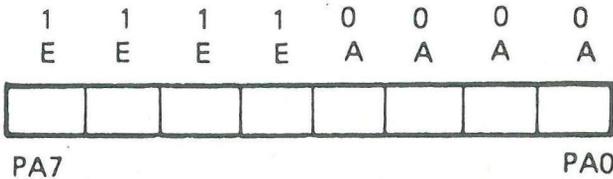
Der Ein-/Ausgabeport wird durch das Datenrichtungsregister gesteuert.

Es kann jede einzelne Leitung wahlweise als Eingang oder als Ausgang durch das Programm, festgelegt werden. Dies geschieht durch den Befehl:

POKE 37138,0 Alle Ports werden Eingänge  
POKE 37138,255 Alle Ports werden Ausgänge

Will ich nun vier der acht I/O-Leitungen zu Eingängen programmieren und die restlichen vier sollen Ausgänge bleiben, so gebe ich ein:

POKE 37138,240



Jetzt sind die Ports PBO – PB3 Eingänge und die Ports PB4 – PB7 Ausgänge. 240 ist das dezimale Äquivalent der Binärzahl 11 11 000. So können Sie jede beliebige Kombination zwischen 0 und 255 poken und bestimmte Ports zu Ein- oder Ausgängen programmieren.

Damit wäre erst einmal die Richtung des Datenflusses festgelegt. Übrigens, diese Konfiguration bleibt so lange erhalten, bis Sie geändert wird oder der VC-20 abgeschaltet wird.

Nach dem Einschalten des VC-20 schaltet dieser automatisch alle I/O-Leitungen auf Eingang. Sie können dies leicht nachprüfen, indem Sie

10 ?PEEK(37138)  
RUN

eingeben.

Nachdem wir jetzt die Richtung des Datenflusses festlegen können, wollen wir versuchen, Daten auf die Ports zu geben. Hierzu gibt es, im VC-20 eine besondere Speicherzelle mit der dezimalen Adresse 37136.

Alles was Sie nach POKE 37138,255 in diese Zelle hineinschreiben, erscheint am Ausgangsport PBO — PB7. Alles was Sie nach POKE 37138,0 aus dieser Speicherzelle herauslesen, wurde von außen an den Port angelegt (5V — TTL oder C MOS Pegel). Im offenen Zustand finden Sie eine log. 1, bei nach Masse verbundenen Ports wird eine log. 0 erkannt.

So, jetzt wollen wir zum besseren Verständnis etwas mit diesen Ausgangsleitungen "spielen".

```
10 REM MINIDEMO VON ELCOMP
20 POKE 37138,255
30 FOR I=0 TO 255
40 POKE 37136,I
60 FOR J=0 TO 255
65 FOR Y=1 TO 1000:NEXT Y
70 PRINT"DER INHALT DES"
72 PRINT"EIN/AUSGABE"
73 PRINT"REGISTERS IST: ";PEEK(37136)+1
80 NEXT I:NEXT J
```

READY.

Dieses Programm setzt die Ausgangsports P0 — P7 nacheinander auf die verschiedenen möglichen Werte. Nach dem Durchlaufen der Schleife bleibt das Ausgaberegister auf 255 = 11111111 stehen.

Sie können jetzt dieses kleine Programm so erweitern, daß Sie jeden gewünschten Ausgangswert oder bestimmte Folgen auf dem Port erzeugen können. Wir wollen jetzt einmal versuchen, ein Lauflicht zu programmieren. Hierzu verwenden wir das obenstehende Programm als Grundlage. Im obigen Programm wird in Zeile 30 durch die FOR-NEXT-Schleife jede ganze Zahl zwischen 0 und 255 erzeugt. D. h., auf dem Port erscheinen innerhalb eines Durchlaufes alle möglichen Kombinationen.

Beim Lauflicht soll aber immer nur ein Ausgang aktiviert werden. Wir erreichen dies, indem wir nur die Zahlen 2, 4, 8, 16, 32, 64 und 128 poken. Auf diese Weise entsteht zwischen den Ausgängen eine lauflichtartige Weiterbewegung.

```

100 REM MINIDEMO 2
110 REM LAUFLICHT
120 REM DIE ZAHL 900
130 REM IN ZEILE 165
140 REM BESTIMMT DIE
141 REM GESCHWINDIGKEIT
142 POKE 37138,255
143 FOR I=1 TO 7
144 X=2↑I
145 PRINT X
146 POKE 37136,X
147 FOR J=0 TO 255
165 FOR Y=1 TO 900:NEXT Y
170 PRINT"DER INHALT DES"
172 PRINT"E/A-REGISTERS"
173 PRINT"IST: ";PEEK(37136)
180 NEXT I
195 GOTO 143

```

Die gewünschten Werte für POKE können auch aus Datastatements gelesen werden. Hierzu soll das nachfolgende Beispiel dienen.

```

200 REM MINIDEMO3
210 READ NUM
220 IF NUM>128 THEN RESTORE:GOTO210
225 FOR G=1 TO 900:NEXTG
230 POKE 37136,NUM
235 PRINT PEEK(37136)
240 GOTO 210
300 DATA 1,2,4,8,16,32,64,128,255
310 END

```

Zum Schluß:

Jetzt haben Sie einige Grundlagen der Ein-/Ausgabeprogrammierung kennengelernt. Sie können jetzt selbst die verschiedensten Konfigurationen und Steuerungen entwerfen und ausführen. Sie brauchen dazu keinen LötKolben. Alles kann über die Tastatur geschehen.

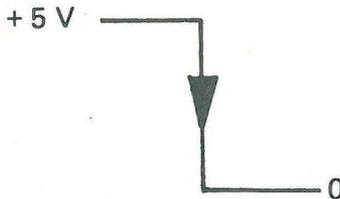
# Ein- und Ausgabe- programmierung II

## Ein- und Ausgabeprogrammierung II

Wir haben bis jetzt nur die 8 Port Ausgänge am VC-20 User Port betrachtet. Neben diesen beiden 8 Leitungen, die wahlweise als Eingänge oder als Ausgänge geschaltet werden können, haben wir dort am User Port noch die Klemmen CB1 und CB2. Diese Anschlüsse sind sog. "Handshake Lines" und dienen zum Quittungsaustausch mit externen Geräten.

CA1 und CA2 werden für den Cassettenrecorder und die die RESTORE Taste unten benötigt.

Im Ausgangszustand verhält sich CB1 so, daß auf eine negative Flanke getriggert wird, d. h. ein Impulsübergang von log. "1" auf log. "0" wird erkannt und registriert.



Wenn Ihr externes System eine positive Flanke erzeugt, also ein Impulsübergang von log. "0" nach log. "1" stattfindet, so können Sie auch diesen Impuls erkennen, wenn Sie das Quittungsverhalten des CB1 über BASIC ändern.

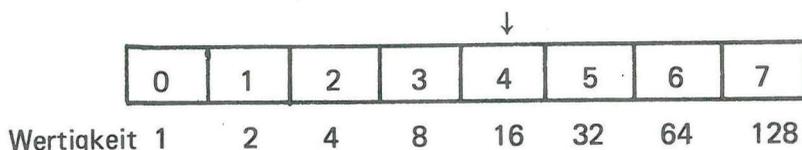
Die Änderung geschieht im Peripheral Control Register, dessen Bit 4 (CB1-Control), das Interrupt Bit enthält. Wenn dieses Bit eine log. "0" ist, so wird eine negative Flanke erkannt, ist es log. "1", wird eine positive Flanke erkannt. Entsprechend wird ein Unterbrechungsflag gesetzt.

Es kann durch POKE 37148, PEEK (37148) OR16 auf "1" gesetzt werden.

Wenn an Pin CB1 = Pin B am User Port ein positiver Impuls erscheint und das Unterbrechungsflag gesetzt wird, so bedeutet dies, daß ein externes Gerät oder Schaltung bereit ist, Daten auf die 8 Eingangsports zu geben.

Jetzt wird im Interrupt-Flag-Register im 6522 ein Bit gesetzt.

Im Interrupt Flag-Register wird Bit 4 im Zusammenhang mit dem CB1 Pin verändert.



### Interrupt-Flag-Register

Das Interrupt-Flag-Register hat die dezimale Adresse 37149. Vom BASIC her können Sie auf dieses Bit durch den Befehl: WAIT 37149,16 aufprüfen. Dieser Befehl stoppt die Programmausführung in BASIC solange, bis der Inhalt des Registers 37149 mit dem Wert logisch UND verknüpft einen Wert von 1 ergibt.

Die logische UND-Verknüpfung führt jedoch nur zu einer log. "1" am Ausgang, wenn beide Werte log. "1" sind, d. h. der Inhalt des Flag-Registers (37149) wird mit dem Wert 16 = binär 0001 0000 logisch verknüpft und auf den Wert 1 geprüft. Im Normalfalle befindet sich im Bit 4 des Kontrollregisters eine 0. Mit WAIT 37149,16 ergibt die logische UND-Verknüpfung eine Null. D. h. BASIC arbeitet nicht. Erst wenn eine positive Flanke erscheint, und das Bit 4 des Kontrollregisters zu 1 macht, ergibt die logische UND-Verknüpfung

eine 1 und der BASIC-Interpreter beginnt wieder zu arbeiten. Seien Sie deshalb vorsichtig, wenn Sie WAIT 37149,16 eingeben, da das Programm dann BASIC verläßt und auf einen Impuls an CB1 wartet. Sie können dann nur wieder ins BASIC zurück, wenn Sie abschalten oder eine aufsteigende Flanke von 0 auf +5V an CB1 anlegen.

```
10 A$="  "  
100 A$="  "  
120 FOR J=1 TO 72  
130 WAIT 37149,16  
140 N=PEEK(37136)AND127  
150 IF N=13 THEN 180  
160 A$=A$+CHR$(N)  
170 NEXT J  
180 PRINT A$
```

Nach Ausführung des WAIT-Befehles können die Daten an den Ports mit PA0 bis PA7 eingelesen werden.

N = PEEK 37136

Diese Leseoperation setzt den CB1-Pin wieder zurück, so daß auf den nächsten Impuls gewartet werden kann.

37136 ist hier die Adresse für das Ein-/Ausgabe-Register B mit Quittungsbetrieb (mit Hanshaking). Der Inhalt des Registers wird nun gelesen und als Variable N mit einem dezimalen Wert zwischen 0 und 255 versehen (je nachdem, was von außen angelegt wurde). Wie schon erwähnt, wird das CB1-Flag durch Lesen wieder zurückgesetzt und kann sofort wieder zum Erkennen nachfolgender Daten verwendet werden. Jetzt haben Sie die Funktion des CB1 Einganges etwas näher kennengelernt. Wir empfehlen Ihnen, einige Experimente damit durchzuführen. Sie können jetzt z. B. ganze ASCII-Zeichen in eine String-Variable einlesen.

```
10 DRR=37138  
20 ACR=37147
```

```
30 FR=37149
40 PCR=37148
100 POKE DRR,0
110 POKE ACR,28
130 POKE PCR,16
135 PRINT"BETAETIGEN"
140 WAIT FR,16
142 N=PEEK(37136)
145 PRINT PEEK(37136)
150 PRINT"AAA"
```

Mit diesem Programm können Sie bis zu 72 ASCII-Zeichen eingelesen werden. Die Eingabe ist auf 7 Bit begrenzt. Werden andere Zeichen eingegeben, kann das Programm durch ein Wandler-Programm in BASIC abgeändert werden.

Für Ihre Experimente mit dem VC-20 und dem Ein-/Ausgabeport können wir Ihnen zur Vereinfachung Ihrer Versuche unbedingt eine Experimentierplatine für den lötfreien Aufbau (von Bekatron) mit einem Verbindungskabel empfehlen.

Man besorgt sich am besten weiterhin noch:

- 1 Messinstrument (vielfach) handelsüblich
- 10 Leuchtdioden
- 10 Widerstände ca. 180 – 220  $\Omega$ /0,25W
- 10 Ein-/Aus Schalter
- 1 Stecker für User Port (24 polig)
- 1 12 – 14 poliges Bandkabel

An die Schalter lötet man ca. 10 cm lange Drähte, so daß man sie einfach in die Experimentierplatine stecken kann. Wer nicht direkt an den Ausgängen der # PIA 6522 im VC-20 arbeiten will, da er eine evtl. Beschädigung ausschließen will, kann zwei 7404 oder 7407 integrierte TTL-Bausteine dazwischenschalten. Falls Sie die Stromversorgung am User Port +5 und Masse nicht mehr als 100 mA belasten, können Sie diese zum Betrieb heranziehen. Will man jedoch größere Aufbauten vornehmen, sollte ein externes +5 Netzteil verwendet werden.

Mit der oben beschriebenen Anordnung lassen sich dann unzählige

Experimente aus der Ein-/Ausgabe Steuertechnik entwickeln und testen. Auch die zugehörige Software kann damit gut ausgetestet werden.

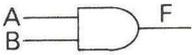
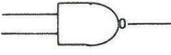
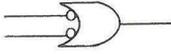
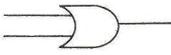
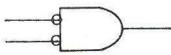
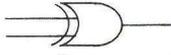
Da die beiden kleinen Programmbeispiele nur in BASIC geschrieben sind, ist die Angelegenheit in gewisser Weise natürlich begrenzt. Zum Verstehen der Zusammenhänge ist dies jedoch mehr als ausreichend.

Wer noch weiter einsteigen will, muß sich dann mit der Programmierung in Maschinsprache dieser Ports beschäftigen.

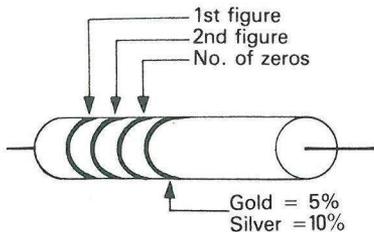
Als Zusatzlektüre für diesen Abschnitt der Ein-/Ausgabeprogrammierung möchte ich Ihnen noch das Buch von E. Flögel "Programmieren in Maschinsprache mit 6502" aus dem Hofacker-Verlag empfehlen.

Achten Sie bitte bei Experimenten mit CB1, daß keine prellenden Impulsflanken entstehen.

## Logische Symbole

AND gate		$F = A \cdot B$
NAND gate (negated output)		$F = \overline{A \cdot B}$
NAND gate (negated inputs)		$F = \overline{A} + \overline{B}$
OR gate		$F = A + B$
NOR gate (negated output)		$F = \overline{A + B}$
NOR gate (negated inputs)		$F = \overline{A} \cdot \overline{B}$
EXCLUSIVE – OR gate		$F = A \cdot \overline{B} + \overline{A} \cdot B$
Inverter gate.		$F = \overline{A}$

## Code für Widerstände



1,000 Ohms = 1K $\Omega$   
1,000,000 Ohms = 1M $\Omega$

0 Black	5 Green
1 Brown	6 Blue
2 Red	7 Violet
3 Orange	8 Grey
4 Yellow	9 White

## Code für Kondensatoren

### Colour Codes:

- as for resistors
- first three bands
- units are pF (pico-Farads)

### Number of Markings:

- mF : micro Farads.
- nF : nano Farads.
- pF : pico-Farads
- : (Russian 'P')
- pico Farads
- H : (Russian 'N')
- nano Farads

### Numerals only:

- first two digits are the value.
- Third digit is number of following zero's.

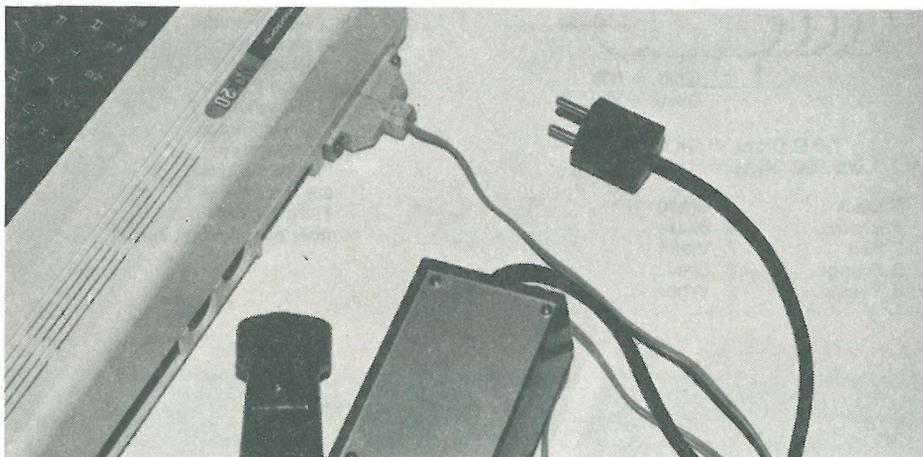
# Ausgabeinterface für Netzverbraucher

## Ausgabeinterface für Netzverbraucher mit VC-20

Für diejenigen, die Ihren VC-20 für Steuer- und Überwachungsschaltungen im Haushalt einsetzen wollen, haben wir ein einfaches Schaltinterface entwickelt. Es ermöglicht Ihnen mit Hilfe Ihres VC-20 einen freiprogrammierbaren Zeitschalter für die unterschiedlichsten Anwendungen aufzubauen.

Sie können damit Ihre Stereoanlage automatisch ein- und ausschalten, einen sekundengenauen Radiowecker zusammenstellen, genaue Zeitschalter (Timer) aufbauen oder Ihren VC-20 sogar als Sicherheitssystem in Ihrer Wohnung oder Haus verwenden.

Das Interface kann an den USER I/O (hinten links am VC-20) angeschlossen werden.



In der Schaltung sind zwei Relais vorgesehen, so daß Sie also bis zu zwei verschiedene Wechselstromverbraucher bedienen können. Die Stromaufnahme und die Spannung richten sich nach dem verwendeten Relais.

Wir haben ein 220V/4A Schaltrelais für Leiterplattenaufbau verwendet (Siemens V23027-A0001-A101).

### Anwenderport (USER I/O-Port)

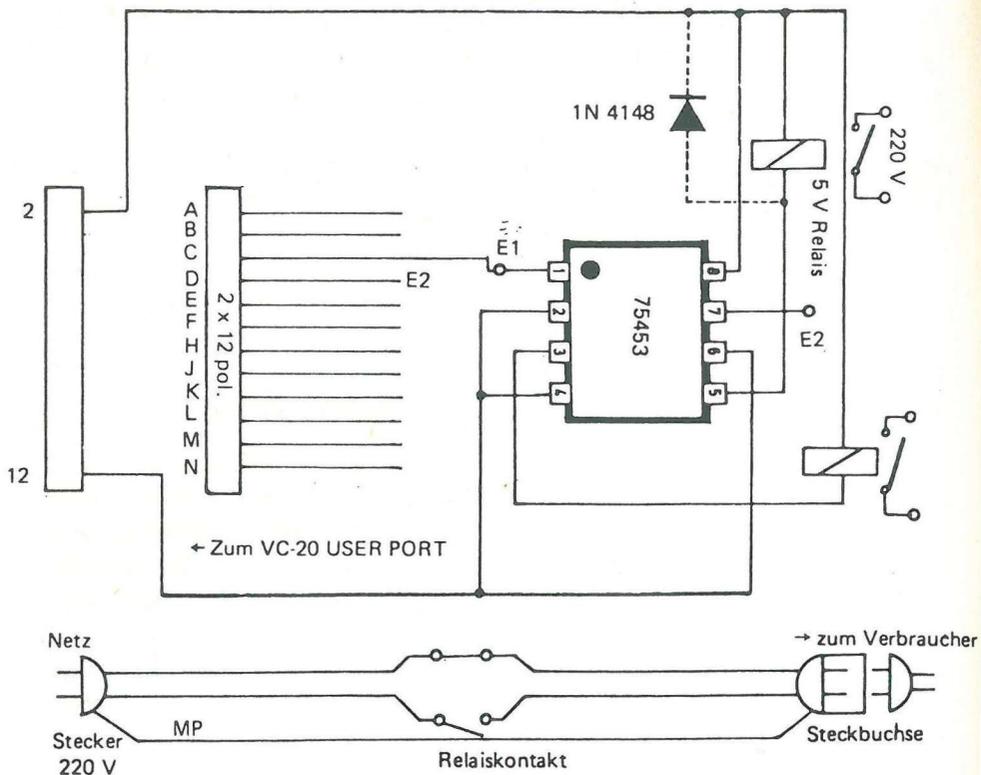
Der für den Experimentierer sehr interessante Ausgabeport befindet sich beim VC-20 hinten am Gehäuse links außen (von vorne gesehen).

Der 24 polige Stecker ist mit dem Ausgangsport B des peripheren Interface-Adapters VIA 6522 verbunden. PB0 – PB7, CB1 und CB2 sind herausgeführt. Die Anschlüsse des Port A vom gleichen Baustein sind als Joystick, Lichtgriffel und Cassettenrecorderschalter auch nach außen geführt. 5V und Masse sind auch vorhanden. Achtung die Belastung darf hier nicht mehr als 100 mA betragen. An diesen Anschluß schalten wir über einen TRW CINCH Stecker TRW 251-12-30-160/50-24A-30 unser Schalterinterface an.

E1 Eingang für Relais 1 an PB0 = C

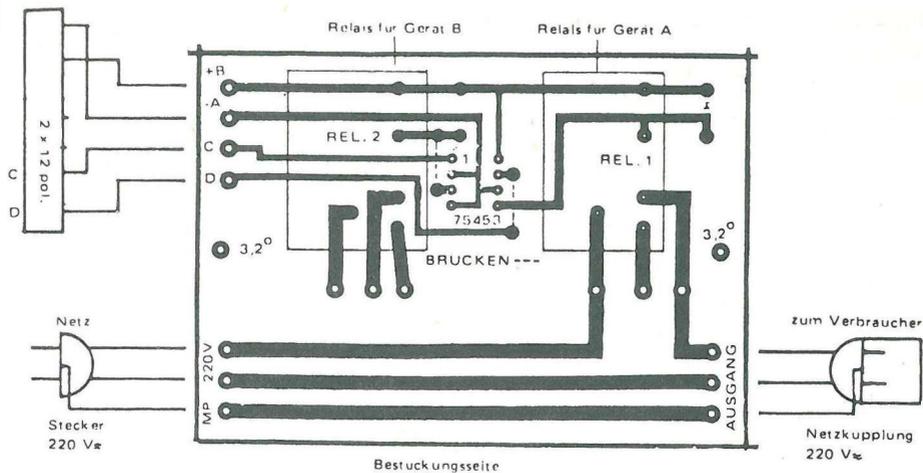
E2 Eingang für Relais 2 an PB1 = D

Der Eingang von Relais 1 (E1) wird mit PB0 am Stecker (Pin C) verbunden. E2 kann mit PB1 (Pin D) verbunden werden. Sind beide angeschlossen, lassen sich zwei Verbraucher gleichzeitig steuern. Die nachfolgende Schaltung zeigt eine Version des Netzschalters, mit der nur ein Verbraucher bedient werden kann. Zur Erweiterung kann jedoch leicht noch ein zweites Netzkabel installiert und die Software entsprechend abgeändert werden.

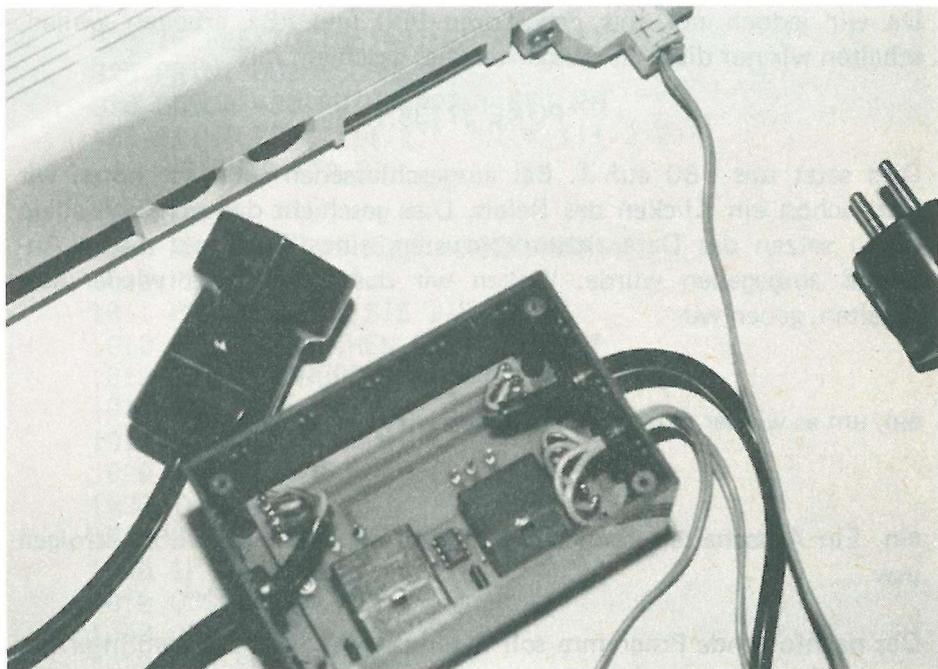


Ein Bausatz mit allen Bauteilen und Platinen unter Best.-Nr. 4846  
zu 199,- DM vom ELCOMP-Platinenservice

Das nachfolgende Bild zeigt uns den Aufbau der Bauteile auf einer Leiterplatte. Die Ansicht erfolgt von der Bestückungsseite her. Es ist nur Relais 1 angeschlossen und über einen Netzsteckersatz herausgeführt. Für Relais 2 sind die entsprechenden Anschlüsse vorgesehen (siehe links im Schaltbild).



Und so sieht der fertige Adapter, mit geöffneten Gehäusedeckel aus. Es empfiehlt sich dringend die Schaltung in einen Plastikgehäuse aufzubauen, da wir es doch mit 220V/50 Netzspannung zu tun haben. Aus diesem Grunde auch beim Experimentieren höchste Vorsicht walten lassen.



Für die softwaremäßige Behandlung der Zusatzschaltung gibt es ein kleines BASIC-Programm. Mit diesem Programm können Sie folgende Steuerfunktionen programmieren.

1. Ein-/Aus-Wechsel in bestimmten Abständen (auf die Sekunde genau)
2. Ein-Aus-Wechsel zu einer bestimmten Zeit
3. Stellen der internen Uhr
4. Einschalten über die Tastatur
5. Ausschalten über die Tastatur
6. STOP

Nach dem Einschalten des VC-20 und nach jedem RESET sind alle Leitungen unseres Tores B am User I/O auf Eingang geschaltet. Wir erkennen dies, indem wir das Datenrichtungsregister von TOR B in der PIA1 6522 durch PRINT PEEK (37138) untersuchen. Die Null in diesen Register sagt uns, daß alle Portleitungen von TOR B als Eingang geschaltet sind. Wollten wir alle zu Ausgängen machen, müssten wir POKE 37138,255 eingeben (alles im Direktmode des VC-20). Da wir jedoch nur mit den Toren PBO und PB1 arbeiten wollen, schalten wir nur diese als Ausgänge. Dies geschieht mit

POKE 37138,1

Dies setzt uns PBO auf 1. Bei ausgeschlossenen Interface hören wir jetzt schon ein Klicken des Relais. Dies geschieht dadurch, daß allein durch setzen des Datenrichtungsregisters eine "Eins" auf diesen Anschluß ausgegeben wurde. Wollen wir das Relais jetzt wieder ausschalten, geben wir

POKE 37136,1

ein, um es wieder einzuschalten geben wir

POKE 37136,0

ein. Ein Ausschalten kann dann wieder mit POKE 37136,1 erfolgen usw.

Das nachfolgende Programm soll Ihnen das Arbeiten mit dem Interface etwas erleichtern.

Es ist für Relais 1 ausgelegt. Sie können sich dieses Programm jedoch auch so umsetzen, daß das Relais 2 auch mitbedient wird. Auch läßt sich die Timer Funktion noch weiter ausbauen. Der Anfangszustand läßt sich mit Hilfe der Funktion 4 und 5 um Menue festlegen. Von diesem Zustand aus arbeitet dann der Schalter. Haben Sie zum Beispiel nach dem Programmstart Funktion 4 = Einschalten gewählt, dann bleibt das Relais eingeschaltet, bis Sie entweder Funktion 5 wählen oder über TIMER (1) oder Schaltuhr (2) einen Schaltvorgang ausgelöst haben.

```

10 POKE 37138,255
20 GOTO 3000
100 REM EINSCHALTEN
110 POKE 37136,0
120 RETURN
200 REM AUSSCHALTEN
210 POKE 37136,1
220 RETURN
300 REM SCHALTEN
310 IF PEEK(37136)=1 THEN POKE 37136,0
    :PRINT"EIN " :GOSUB 400:PRINT:RETURN
320 IF PEEK(37136)=0 THEN POKE 37136,1
325 PRINT"AUSGESCHALTET";
330 GOSUB 400:PRINT:PRINT:RETURN
400 PRINTLEFT$(TI$,2)":"MID$(TI$,3,2)"
    : "RIGHT$(TI$,2)
410 RETURN
1000 REM EIN-AUS ALLE N SEKUNDEN
1010 PRINT"TIMER"
1011 PRINT"GEBEN SIE DIE ZEIT"
1012 PRINT"ZWISCHEN DEN SCHALT-"
1013 PRINT"VORGAENGEN AN !"
1014 INPUT N
1015 GOSUB 300
1020 N=N*60
1030 T=TI+N
1035 GET C$: IF C$<>"" THEN RETURN
1040 IF T<T THEN 1035
1050 GOSUB 300
1060 GOTO 1030
2000 REM SCHALTUHR

```

```

2005 GOSUB 400
2010 PRINT"WANN SOLL GE-"
2011 PRINT"SCHALTET WERDEN ?"
2020 PRINT"GEBE 000000 EIN"
2021 PRINT"WENN SIE FERTIG SIND"
2030 FOR MX=1TO9
2040 PRINT"ZEIT ";MX
2050 INPUT T$
2060 IF LEN(T$)<>6 THEN PRINT"6 STELLIG
      BITTE":GOTO2040
2070 T(MX)=VAL(T$)
2080 IFT(MX)=0THEN 2100
2090 NEXT MX
2100 FOR J=1TO MX:PRINT T(J):NEXT J
2110 I=1
2115 PRINT
2130 IF T(I)<VAL(TI$)THEN 2200
2140 PRINT"#####";:GOSUB 400
2150 GOTO 2130
2200 GOSUB 300
2210 I=I+1
2220 IF T(I)=0 THEN RETURN
2230 GOTO 2115
2800 PRINT"WIEVIEL UHR IST ES?"
2810 INPUT TIME$:RETURN
2999 PRINT"AUFWIEDERSEHEN":END
3000 PRINT"0 WAEHLN SIE BITTE"
3010 PRINT"1.TIMER"
3020 PRINT"2.SCHALTUHR"
3030 PRINT"3.STELLEN DER UHR"
3040 PRINT"4.EINSCHALTEN"
3050 PRINT"5.AUSSCHALTEN"
3060 PRINT"6.STOP"
3100 GET C$
3110 IF C$=""THEN 3100
3200 C=VAL(C$)
3210 ON C GOSUB 1000,2000,2800,100,200,
      2999
3220 GOTO 3000

```

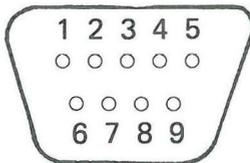
# Tricks für Ihren Volkscomputer VC-20

## Tricks für Ihren Volkscomputer VC-20

Im folgenden Teil wollen wir über Dinge berichten, die Sie vergeblich im mitgelieferten Handbuch suchen werden. Einige Anrufe von VC-20 Besitzern haben uns diesbezüglich schon erreicht.

### Joystickprogrammierung mit dem VC-20

Ihr VC-20 hat von vorne gesehen rechts einen neunpoligen Sockel zum Anschluß eines Steuerknüppels.



von vorne gesehen  
Controlport

An die Anschlüsse 1 – 4 sind Leitungen der zwei Ein-/Ausgabe-Bausteine 6522 im Inneren des Bausteins angeschlossen. Diesem Baustein sind zwei bestimmte Adressen aus dem Speicherbereich zugewiesen worden. (Näheres finden Sie hierüber im Buch Nr. 118 "Programmieren in Maschinensprache", von E. Flögel, S. 209).



ATARI-Joystick angeschlossen an den VC-20

Anschlüsse am  
Controller-Stecker  
und deren  
Bedeutung:

Pin	Schalterbez. am Joystick
1	= JOY 0
2	= JOY 1
3	= JOY 2
4	= JOY 3
6	= Schußknopf
8	= Masse

Jeder der beiden 6522 VIA-Bausteine hat zwei Ein-/Ausgabe-Tore. TORA und TORB. Jedes TOR kann mit Hilfe eines Datenrichtungsregisters als Eingabeport oder als Ausgabeport geschaltet werden. Nach Einschalten des Rechners sind alle Portleitungen als Eingänge geschaltet (DDR = 0).

Die Schalter 0, 1, 2 und der Schußknopf werden über die VIA # 1 und Schalter 3 muß über die VIA # 2 gelesen werden.

Die Adressen dieser beiden TORE sind:

VIA # 1 = \$9110 hex = 37136 dez.

VIA # 2 = \$9120 hex = 37152 dez.

Dies sind die jeweiligen Anfangsadressen, wobei entsprechend dem 6522 Datenblatt (siehe Buch Nr. 118) die folgenden 15 Adressen dann den gesamten Ein-/Ausgabeblock bilden.

### Lesen des Joysticks

Um den Joystick zu lesen, müssen zuerst die Datenrichtungsregister mit 0 geladen werden. Die Schalterstellungen müssen dann mit zwei PEEK-Befehlen gelesen werden (siehe hierzu Listing Nr. 1).

```
9000 DIM SK(2,2):D1=37139:D2=37154
9005 POKED1;0
9010 PA=37137:PB=37152
9020 FOR I=0TO2:FORJ=0TO2:READSK(J,I):NEXTJ,I
9030 DATA 7,0,1,6,8,2,5,4,3
9040 GOSUB 9500:PRINT SK(X+1,Y+1):GOTO9040
9500 POKED2,127:S3=-((PEEK(PB)AND128)=0):POKED2,255
9510 P=PEEK(PA):S1=-((PAND8)=0):S2=((PAND16)=0):S0=((PAND4)=0)
9520 FR=-((PAND32)=0):X=S2+S3:Y=S1+S0:RETURN
```

### Listing 1

D1 und D2 sind die beiden Datenrichtungsregister (Zeile 9000). In 9005 wird das Datenrichtungsregister D1 des einen VIA auf Eingabe geschaltet. PA und PB sind die beiden Portadressen. Die entsprechenden Werte für die Richtungen werden in ein Feld gelesen. Ab Zeile 9500 folgt das eigentliche Unterprogramm, welches Sie auch in Ihren eigenen Programmversuchen verwenden können. Nun erfolgt die Abfrage der einzelnen Bits am Port. Dies geschieht dadurch, daß das entsprechende Bit mit einem Wert, der der Zahl hoch zwei entspricht, abgefragt wird (Beispiel: Soll Bit 2 geprüft werden (Joyo), so wird mit 4 eine logische AND-Verknüpfung durchgeführt.

```

0 0 0 0 0 0 0 0  PORT A
0 1 2 3 4 5 6 7
    ↑
    Bit 2

```

In Listing 2 sehen Sie eine weitere Ausbaustufe unseres kleinen Hilfsprogrammes. Hier werden die Werte abgefragt und die Richtungen ausgedruckt.

```

0 DIM SK(2,2):D1=37139:D2=37154
1 POKED1,0
2 PA=37137:PB=37152
3 FOR I=0TO2:FORJ=0TO2:READSK(J,I):NEXTJ,I
4 DATA 7,0,1,6,8,2,5,4,3
10 REM STEUERKNUEPPEL
20 REM UTILITY PROGRAMM
30 GOSUB 9500
40 LET CC=SK(X+1,Y+1)
50 IF CC=8 THEN PRINT"RUHESTELLUNG"
55 IF CC=7THENPRINT"NORDWESTEN"
60 IF CC=0 THEN PRINT"NACH NORDEN"
65 IF CC=1THENPRINT"NORDOSTEN"
70 IF CC=4 THEN PRINT"NACH SUEDEN"
75 IF CC=5THEN PRINT"SUEDWESTEN"
80 IF CC=6 THEN PRINT"NACH WESTEN"
85 IF CC=3THEN PRINT"SUEDOSTEN"
90 IF CC=2 THEN PRINT"NACH OSTEN"
95 IF FR=1THEN PRINT"***FEUER**"
100 GOTO 30

```

```

9500 POKED2,127:S3=-((PEEK(PB)AND128)=0):POKED2,255
9510 P=PEEK(PA):S1=-((PAND8)=0):S2=((PAND16)=0):S0=((PAND4)=0)
9515 POKE D1,255
9520 FR=-((PAND32)=0):X=S2+S3:Y=S1+S0:RETURN

```

### Listing 2

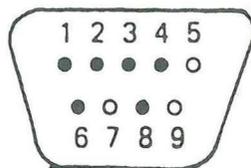
Die Variable FR sagt uns mit 1, daß der Schußknopf gedrückt wurde.

Listing 3 ist ein Zusatz zu Listing 2. Wenn Sie diese Zeilen noch einfügen, können Sie den Schußknopf testen. Über den Lautsprecher Ihres Fernsehgerätes wird dann der Ton ausgegeben.

```

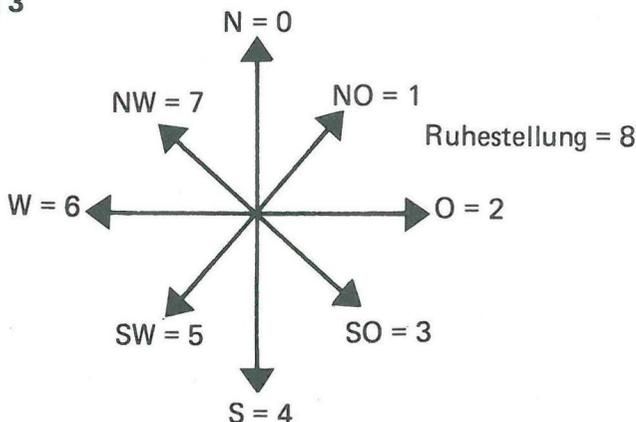
95 IF FR=1 THEN GOSUB 500
100 GOTO 30
500 POKE36877,220
510 FOR L=15 TO 0 STEP -1
520 POKE36878,L
530 FOR M=1 TO 25
540 NEXT M
550 NEXT L
560 POKE36877,0
570 POKE36878,0
580 RETURN

```



Stecker von hinten gesehen

### Listing 3



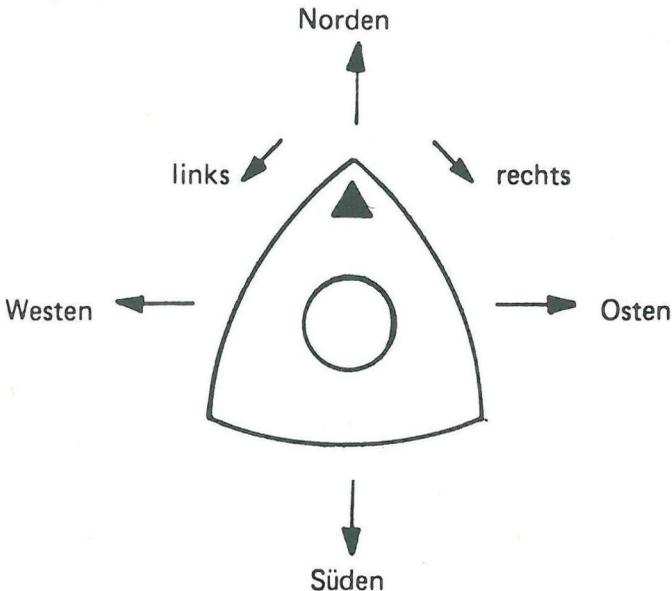
Zur Fertigstellung des Joysticks müssen die Anschlüsse des Steuerknüppels mit dem 9 poligen Stecker verbunden werden.

Schneiden Sie deshalb den Plastikstecker am Joystick ab und entfernen Sie die Isolierung an den zu verlötenden Drähten, verbinden Sie die Lötflächen am 9 poligen Stecker mit den Drähten wie folgt:

Pin	Farbe
1	Gelb
2	Grün
3	Blau
4	Grau
8	Weiß
6	Schwarz

Die verbleibenden Drähte können entfernt werden oder für eigene Konfigurationen verwendet werden.

### Fairchild Joystick





Fairchild-Joystick angeschlossen an den VC-20

Beim Joystick ist

Pin	Bezeichnung
8	= Masse
6	= Schussknopf
1	= Norden
2	= Süden
3	= Westen
4	= Osten

## Fairchild Joystick

Farbe, Anschlußdraht	Bezeichnung
Weiß	Masse
Schwarz	drücken nach unten
Gelb	Norden drücken
Grün	Süden drücken
Blau	Westen drücken
Grau	Osten drücken
Braun	ziehen
Orange	links drehen
Rot	rechts drehen



PB4

### Dual Joystick für VC-20

Anschluß

PB3

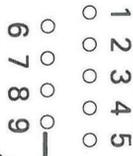
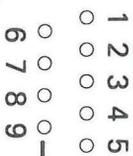
PB2

PB1

PB0

rechter  
Joystick

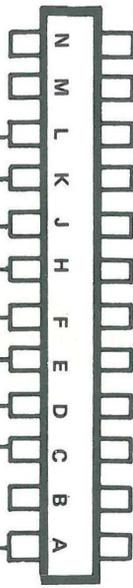
linker  
Joystick



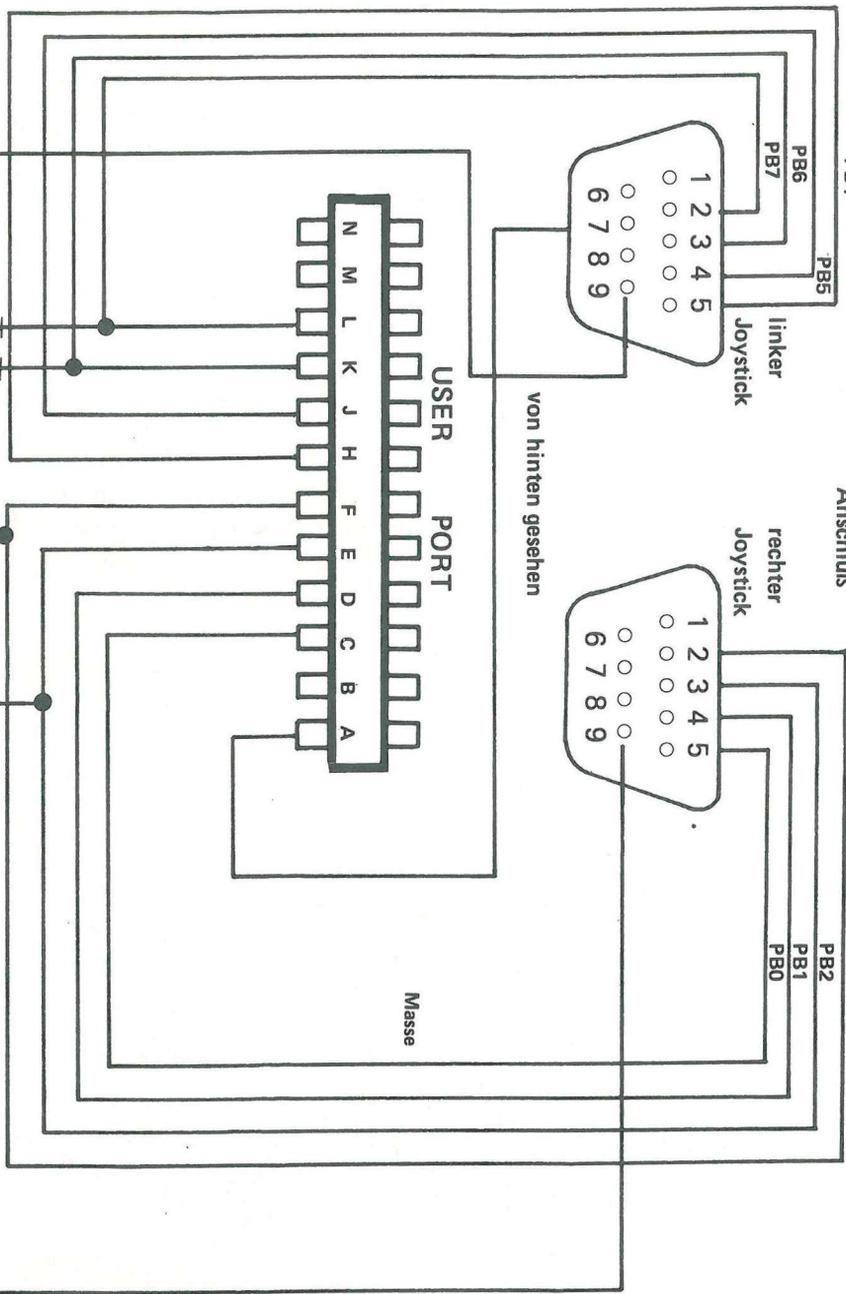
von hinten gesehen

Masse

USER PORT



4 x 1N4148



Im nachfolgenden Teil wollen wir Ihnen zeigen, wie man einen zusätzlichen Dualjoystick an den VC-20 anschließen kann. Sie können dann also bis zu drei Joysticks am VC-20 betreiben. Am Schluß geben wir Ihnen dann ein Listing eines aufregenden Schlangenspiels mit Tonausgabe und farbiger Grafik. Es läuft in der 3,5K Grundversion des VC-20.

### Erstellung eines Interface für den Dualjoystick

Zum Anschluß von zwei Joysticks bauen wir uns zuerst ein kleines Interface (siehe Schaltbild). Alles was wir dazu benötigen sind:

- 1 CINCH TRW—Stecker CARDCON  
251—12—50—170 50—24SN—9 8124
- 2 Dioden 1N4148 oder ähnlich
- 2 Joysticks ATARI oder Fairchild  
Best.-Nr. 4845 Joystick von Hofacker
- 2 9 polige Stecker für den Anschluß der Joysticks

Die beiden Joysticks werden über dieses Interface an den User Port des VC-20 hinten links (von vorne gesehen) angeschlossen.

Über Port B des PIA # 1 werden beide Joysticks abgefragt. Nachdem Sie das Interface aufgebaut haben, können Sie es mit folgendem kleinen Testprogramm prüfen:

```
10 POKE 37138,0
15 N=PEEK(37136)
20 PRINT N
25 GOTO 15
```

Schließen Sie beide Joysticks an und geben Sie RUN ein. Auf der linken Seite des Bildschirmes laufen jetzt Zahlen von oben nach unten durch. Je nach dem welchen Joystick Sie jetzt in eine Richtung bewegen, erhalten Sie einen bestimmten Zahlenwert auf dem Bildschirm. Der Wert N kann einen Wert zwischen 0 und 255 einnehmen. Er muß bei den entsprechenden Richtungen wie folgt aussehen:

## Rechter Joystick

Bewegung nach Norden 255-1

Bewegung nach Süden 255-2

Bewegung nach Westen 255-4

Bewegung nach Osten 255-8

## Linker Joystick

Bewegung nach Norden 255-16

Bewegung nach Süden 255-32

Bewegung nach Westen 255-64

Bewegung nach Osten 255-128

Wenn Sie später Ihre eigenen Programme für diese beiden Joysticks schreiben wollen, müssen Sie auch noch wissen, wie die Schußknöpfe abgefragt werden. Wenn der rechte Joystick gedrückt wird, wird vom jeweiligen Wert 1 abgezogen. Wird der Knopf beim linken Joystick gedrückt, wird 16 abgezogen.

Beispiel: Rechter Joystick in Ruhestellung liefert 254.

Üben Sie einige Beispiele und stellen Sie sicher, daß Sie die Zusammenhänge vor der Programmierung genau verstanden haben.

Beachten Sie bitte folgendes:

Wenn Z. B. der rechte Joystick nach Nordosten gedrückt wird, ergibt sich ein Wert von  $255-1-8 = 264$ .

Während das Programm weiterläuft drücken Sie jetzt den Feuerknopf auf dem rechten Joystick. Der angezeigte Wert sollte jetzt 243 sein. Beachten Sie bitte, daß der gleiche Wert auch angezeigt würde, wenn sich der rechte Joystick gleichzeitig nach Osten und Westen bewegen würde. Da dieses jedoch gleichzeitig nicht möglich ist, muß über Programm gewährleistet werden, daß der Computer erkennt, daß der Schußknopf gedrückt wurde.

Ihr Programm muß "wissen", daß es nicht länger eine der beiden Richtungen erkennen kann, während der Feuerknopf gedrückt ist. Um dies anschaulich zu machen, drücken Sie den Feuerknopf und bewegen Sie den Joystick nach Osten. Der Wert von N ändert sich nicht.

Ähnlich verhält es sich, wenn Sie den Joystick nach Westen bewegen. Während jetzt das Programm weiterläuft, drücken Sie den Feuerknopf am Linken Joystick. Der Wert von N wird nun 63. Der linke Joystickknopf verhält sich ähnlich wie der rechte Joystick. Drücken des Feuerknopfes hat den gleichen Effekt, wie eine Bewegung nach Ost und West zur gleichen Zeit. Auf diese Weise ergibt sich ein Wert von  $N = 255 - 64 - 128 = 63$ .

Per Programm muß man also nun feststellen, welcher der Joysticks sich gerade scheinbar nach Osten und Westen gleichzeitig bewegt. Wird dies dann festgestellt, wurde der Schußknopf betätigt.

Zum Abschluß wollen wir Ihnen noch ein recht lustiges und aufregendes Programm, für welches Sie zwei Joysticks benötigen, zeigen.

Es kann von zwei Spielern gegeneinander auf dem Bildschirm ausgetragen werden. Ziel des Spiels ist es, die Schlange so lange wie möglich auf dem Bildschirm umherzusteuern und dabei nirgends anzustoßen. Auch eine Rückwärtsbewegung führt zum "CRASH".

Da das Programm die 3,5K RAM der Grundversion voll ausnutzt, wurden die Erläuterungen im Programm etwas gestutzt.

Geben Sie das Programm sorgfältig ein. Es muß arbeiten, da es von uns getestet wurde. Achten Sie bitte genau auf die Grafik und die Cursorsteuerzeichen.

Spieler II entspricht dem linken Joystick (von vorne gesehen)  
Spieler I entspricht dem rechten Joystick (von vorne Gesehen)

Starten Sie mit RUN. Es erfolgt die Frage, ob Sie Erläuterungen bzw. eine Anleitung wünschen. Antworten Sie mit "J" für Ja oder mit "N" für Nein.

Das Programm bietet weiterhin die Möglichkeit, wenn kein Spielpartner vorhanden ist, gegen den Computer zu spielen. Hierzu werden Sie gefragt: "Wollen Sie gegen einen Freund oder gegen den Computer spielen".

Antworten Sie hier, je nach Wunsch mit "F" oder mit "C".

Wie bereits angedeutet, ist der Vorspann aus Speicherplatzgründen etwas zusammengestutzt worden. Das Spiel arbeitet mit Tonausgabe über den FS-Lautsprecher.

```
1 PRINT"J":PRINT:PRINT:POKE36879,27
2 PRINTSPC(4);"■ VC-SCHLANGE ■":GOTO 13
3 PRINT"SPIELEN, VERMEIDEN SIE EINEN"
4 REM
6 PRINT"FREUND ODER C FUER"
7 PRINT"COMPUTER EIN"
8 GOTO 26
9 REM COPYRIGHT C 1982
10 REM WINFRIED HOFACKER
13 PRINT "ANLEITUNG?"
14 GETZZ$
15 IFZZ$="N"THENPRINT:GOTO4
16 IFZZ$<>"J"THEN14
20 PRINT:PRINT"DIESE SPIEL KOENNEN"
22 PRINT"SIE GEGEN DEN COMPUTER"
23 PRINT "ODER GEGEN EINEN FREUND"
24 GOTO 3
26 GETOP$: IFOP$=""THEN26
27 IFOP$<>"C"ANDOP$<>"F"THEN26
28 IFOP$="F"THEN46
30 PRINT:PRINT"DA SIE GEGEN DEN"
32 PRINT"COMPUTER SPIELEN, VER"
33 PRINT"WENDEN SIE JOYSTICK#1"
35 PRINT"SIE STARTEN OBEN LINKS"
46 PRINT:PRINT "DRUECKEN SIE EINE TASTE"
47 GET ZZ$: IF ZZ$="" THEN 47
50 DIM IX(4),DY(4),DC(4),MV(15)
60 FORI=1TO4
70 READ IX(I),DY(I),DC(I)
80 NEXT
85 FORN=1TO15:READMV(N):NEXT
90 US(1)=0:US(2)=0:CS=0:N=.92:W=.97:V=1:L=2:R=3:U=4:K=.5
100 GOSUB10000
105 PRINT"■";
110 TL=20:UL(1)=214:UL(2)=219:FT=40:S=32:EN=0:CL=219:CD=2
140 UD(1)=3:UD(2)=2:M=50
160 PZ(1)=7773 :PZ(2)=8092 :CA=8092
165 POKE36879,40
```

```

210 TI$="000000"
500 TZ=TI
505 ZU=PEEK(37136)
510 PRD=MV((ZUAND240)/16):IFPRD<>0THENU(1)=PRD
515 I=1:GOSUB600:IFEN=1THEN5000
517 IFOP$="C"THEN1000
520 PLD=MV(ZUAND15):IFPLD<>0THENU(2)=PLD
530 I=2:GOSUB600:IFEN=1ANDOP$="F"THEN5000
540 IFEN=1ANDOP$="C"THEN6000
550 GOTO1900
600 A=PZ(I)+DX(UD(I))+DY(UD(I))
660 IFPEEK(A)<>STHENEN=1:RETURN
750 POKEPZ(I),UL(I)
755 GOSUB 8000
760 POKEA,DC(UD(I))
800 PZ(I)=A
805 RETURN
1000 REM PET MOVE
1030 X=DX(CD):Y=DY(CD)
1040 IF(PEEK(CA+X+X+Y+Y)=S)AND(PEEK(CA+X+Y)=S)THEN1600
1080 IFCD=LORCD=RTHEN1200
1090 IFRND(V)>KTHENH=L:G=R:GOTO1110
1100 H=R:G=L
1110 X=DX(H):Y=DY(H)
1120 IF(PEEK(CA+X+X+Y+Y)=S)AND(PEEK(CA+X+Y)=S)THEN1590
1140 IFH<>GTHENH=G:GOTO1110
1150 GOTO1300
1200 IFRND(V)>KTHENH=V:G=U:GOTO1110
1220 H=U:G=V
1240 GOTO1110
1300 X=DX(CD):Y=DY(CD)
1320 IFPEEK(CA+X+Y)=STHEN1600
1400 IFCD=LORCD=RTHEN1500
1420 IFRND(V)>KTHENH=L:G=R:GOTO1460
1440 H=R:G=L
1460 X=DX(H):Y=DY(H)
1470 IFPEEK(CA+X+Y)=STHEN1590
1480 IFH<>GTHENH=G:GOTO1460
1490 GOTO1590
1500 IFRND(V)>KTHENH=V:G=U:GOTO1460
1520 H=U:G=V
1540 GOTO1460
1590 CD=H

```

```

1600 IFRND(V)>WTHEN1080
1650 A=CA+DX(CD)+DY(CD)
1660 IFPEEK(A)<>STHEN6000
1750 POKECA,CL
1760 POKE A,DC(CD)
1800 CA=A
1900 IFTI-TZ<MTHEN1900
1940 M=M*N
1945 IFM<1THEN M=1
1950 GOTO500
5000 REM CRASH SPIELER
5005 IFI=1THENWW=2
5006 IFI=2THENWW=1
5010 POKEPZ(I),UL(I)
5020 POKEA,DC(UD(I))
5030 US(WW)=US(WW)+1
5035 IFOP$="P"THENC$=CS+1
5040 PRINT"SP SPIELER#"I;
5050 PRINT"PENG! "
5055 GOSUB 7000
5060 PRINT"TTSP.#1:"US(1);
5062 IFOP$="P"THENPRINT"    PET:"CS:GOTO5070
5065 PRINT"SP.#2:"US(2)
5070 TT=TI
5080 GETA$
5090 IFTI-TT<4*60THEN5080
5100 GOTO100
6000 REM CRASH VC-20
6010 POKECA,CL
6020 POKEA,DC(CD)
6030 US(1)=US(1)+1:N=N*.98:IFN<.5THENN=.5
6040 PRINT"VC VC-20";
6050 GOTO5050
7000 POKE36877,220
7010 FOR P=15 TO 0 STEP-1
7020 POKE 36878,P
7030 FOR H=1TO 10
7040 NEXT H
7050 NEXT P
7060 POKE 36877,0
7070 POKE 36878,0
7080 RETURN
8000 POKE36878,15

```

```

8010 POKE36876,220
8020 FOR P=1T05
8030 NEXT P
8040 POKE36878,0
8050 FOR P=1T050
8060 NEXT P
8070 POKE36876,200
8080 FOR P=1 TO 5
8090 NEXT P
8100 POKE36876,0
8110 FOR P=1T050
8120 NEXT P
8130 POKE36878,0
8140 RETURN
10000 REM ZEICHNE RAND
10010 PRINT"□";A$="■"
10020 FORI=1T022:PRINTA$;:NEXT
10040 FORI=1T021:PRINTA$;SPC(20);A$;:NEXT
10060 FORI=1T021:PRINTA$;:NEXT:PRINT"■■■■";RIGHT$(A$,1);
10100 RETURN
11000 ZU=PEEK(37136)
11005 IFZU=251THEND=1:GOTO570
11010 IFZU=247THEND=2:GOTO570
11015 IFZU=239THEND=3:GOTO570
11020 IFZU=253THEND=4:GOTO570
11025 GOTO600
50000 DATA 0,22,20,-1,0,60,1,0,62,0,-22,30
50005 DATA0,0,0,0,0,3,0,0,0,2,0,1,4,0

```

READY.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
7680																							
7702																							
7724																							
7746																							
7768																							
7790																							
7812																							
7834																							
7856																							
7878																							
7900																							
7922																							
7944																							
7966																							
7988																							
8010																							
8032																							
8054																							
8076																							
8098																							
8120																							
8142																							
8164																							

Adressen für Zeichen am Bildschirm

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
38400																							
38422																							
38444																							
38466																							
38488																							
38510																							
38532																							
38554																							
38576																							
38598																							
38620																							
38642																							
38664																							
38686																							
38708																							
38730																							
38752																							
38774																							
38796																							
38818																							
38840																							
38862																							
38884																							

Adressen für Farben am Bildschirm

**Farbenkombination von  
Bildschirm und Bildschirmrand**

**POKE 36879,X**

	Bildschirmrand		Bildschirm						
	Schwarz	Weiß	Rot	Cyan	Purpur	Grün	Blau	Gelb	
Schwarz	8	9	19	11	12	13	14	15	
Weiß	24	25	26	27	28	29	30	31	
Rot	40	41	42	43	44	45	46	47	
Cyan	56	57	58	59	60	61	62	63	
Purpur	72	73	74	75	76	77	78	79	
Grün	88	89	90	91	92	93	94	95	
Blau	104	105	106	107	108	109	110	111	
Gelb	120	121	122	123	124	125	126	127	

	Bildschirmrand		Bildschirm						
	Orange	H.-Orange	Rosa	H.-Cyan	H.-Purpur	H.-Grün	H.-Blau	H.-Gelb	
Schwarz	136	152	168	184	200	216	232	248	
Weiß	137	153	169	185	201	217	233	249	
Rot	138	154	170	186	202	218	234	250	
Cyan	139	155	171	187	203	219	235	251	
Purpur	140	156	172	188	204	220	236	252	
Grün	141	157	173	189	205	221	237	253	
Blau	142	158	174	190	206	222	238	254	
Gelb	143	159	175	191	207	223	239	255	

**PEEKs and POKES**

<b>Farbe</b>	
POKE 646,X	Set current color code
POKE 36879,X	Set screen and border color

<b>Lichtgriffel</b>	
PEEK (36870)	Horizontal position of light pen
PEEK (36871)	Vertical position of light pen

<b>Tastatur</b>	
POKE 650,128	Make all keys repeat
PEEK (203)	Which key pressed (64 if none)
PEEK (198)	# of character in keyboard buffer

<b>Ton</b>	
POKE 36874,X	Frequenz des Oszillators (niedrig)
POKE 36875,X	Frequenz des Oszillators (mittel)
POKE 36876,X	Frequenz des Oszillators (hoch)
POKE 36877,X	Frequenz des Rauschgenerators
POKE 36878,X	Lautstärke

<b>Anwender Port</b>	
POKE 37138,0	Alle Bits Eingänge
POKE 37138,X	Alle Bits Ausgänge
POKE 37136,X	Data Register (Ein-/Ausgabe)

<b>Cassettenrecorder</b>	
POKE 37148,252	Einschalten des Recorders
POKE 37148,254	Ausschalten des Recorders

**Tabelle für Musiknoten**

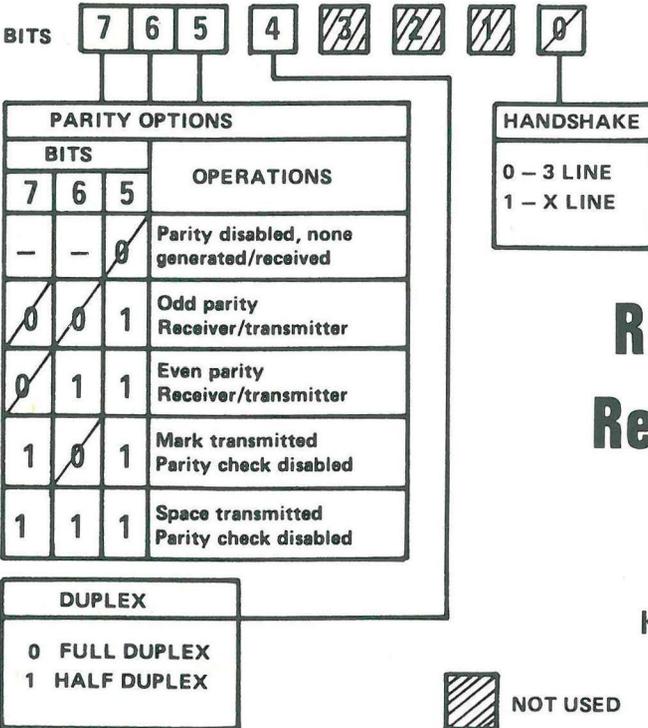
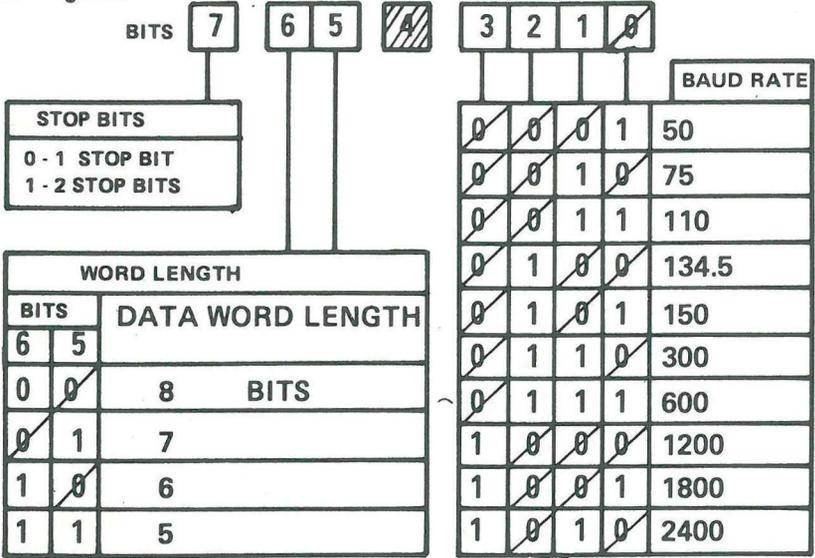
Note	Wert	Note	Wert	Note	Wert	Note	Wert
C	135	A#	187	G#	217	F#	233
C#	143	B	191	A	219	G	235
D	147	C	195	A#	221	G#	236
D#	151	C#	199	B	223	A	237
E	159	D	201	C	225	A#	238
F	163	D#	203	C#	227	B	239
F#	167	E	207	D	228	C	240
G	175	F	209	D#	229	C#	241
G#	179	F#	212	E	231		
A	183	G	215	F	232		

Lautsprecher-Kommando	wobei X folgendes sein kann:	Funktion
POKE 36878,X	0 to 15	Lautstärke
POKE 36874,X	128 to 255	Ton
POKE 36875,X	128 to 255	Ton
POKE 36876,X	128 to 255	Ton
POKE 36877,X	128 to 255	Geräusch

# BASIC TOKENS

128	END	167	THEN
129	FOR	168	NOT
130	NEXT	169	STEP
131	DATA	170	+
132	INPUT #	171	-
133	INPUT	172	.
134	DIM	173	/
135	READ	174	↑
136	LET	175	AND
137	GOTO	176	OR
138	RUN	177	)
139	IF	178	=
140	RESTORE	179	<
141	GOSUB	180	SGN
142	RETURN	181	INT
143	REM	182	ABS
144	STOP	183	USR
145	ON	184	FRE
146	WAIT	185	POS
147	LOAD	186	SQR
148	SAVE	187	RND
149	VERIFY	188	LOG
150	DEF	189	EXP
151	POKE	190	COS
152	PRINT #	191	SIN
153	PRINT	192	TAN
154	CONT	193	ATN
156	LIST	194	PEEK
157	CLR	195	LEN
158	CMD	196	STR\$
159	SYS	197	VAL
160	OPEN	198	ASC
161	CLOSE	199	CHR\$
162	GET	200	LEFT\$
163	NEW	201	RIGHT\$
164	TAB <	202	MID\$
165	TO	203	- 254 unused
166	SPC <	255	π

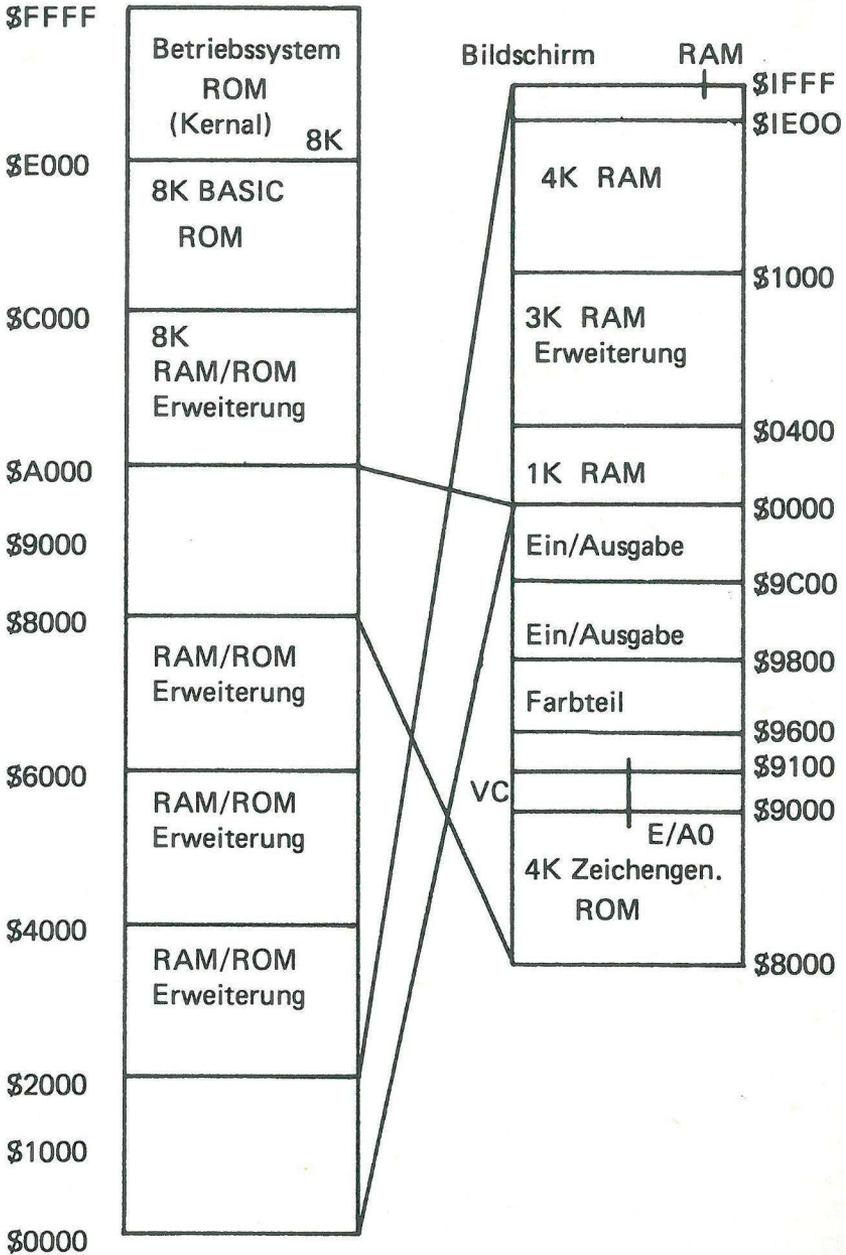
# Kontroll Register



# RS-232 Register

Kommando Register

# Speicherbelegung des VC-20



# VC-20 MEMORY MAP

## KERNAL und Microsoft BASIC

### VC-20 MEMORY MAP KERNAL und Microsoft BASIC

Hex	Dezimal	Beschreibung
0001–0002	1–2	USR-Sprungadresse
0007	7	Suche Zeichen
0008	8	Scan-quotes flag
0009	9	TAB Spalten abspeichern
000A	10	0 = LOAD                      1 = VERIFY
000B	11	Bestimmter Puffer # 1 für Eingabe
000C	12	Grundzustand des DIM-Flag
000D	13	Type: FF = Zeichenkette; 00 = numerisch
000E	14	Type: 80 = Integer; 00 = Fließkomma
000F	15	Flagge für DATA/LIST/Speicher
0010	16	Subscript/FNx-Flag
0011	17	0 = Input; \$40 = GET; \$98 = Read
0013	19	Momentane I/O-Flagge
0014–0015	20–21	Ganzzahliger (Integer) Wert
0016	22	Zeiger: Vorübergehender Stack für Zeichenkette
0017–0018	23–24	Letzter vorübergehender Vektor für Zeichenkette
0019–0021	25–33	Stack für vorübergehende Strings
0022–0025	34–37	Feld für Zeiger zum Hilfsprogramm
0026–002A	38–42	Feld für das Produkt einr Multiplikation
002B–002C	43–44	Zeiger: Beginn von BASIC
002D–002E	45–46	Zeiger: Beginn der Variablen
002F–0030	47–48	Zeiger: Beginn der Felder (Arrays)
0031–0032	49–50	Zeiger: Ende der Felder
0033–0034	51–52	Zeiger: Speicherung der Zeichenkette (nach unten)

0037-0038	55-56	Zeiger: Grenze des Speicherplatzes
0039-003A	57-58	Momentane Zeilennummer in BASIC
003B-003C	59-60	Vorhergende Zeilennummer in BASIC
003D-003E	61-62	Zeiger: BASIC-Befehl für CONT
003F-0040	63-64	Momentane Zeilennummer für DATA
0041-0042	65-66	Momentane Adresse für DATA
0043-0044	67-68	Eingabe-Vektor
0045-0046	69-70	Momentaner Name der Variablen
0047-0048	71-72	Momentane Adresse der Variablen
0049-004A	73-74	Zeiger für Variable bei FOR/NEXT
004B-004C	75-76	Y-Save; op-save; save Zeiger für BASIC
004D	77	Vergleich: Symbol Akku
004E-0053	78-83	Feld für diverse Berechnungen (Pointer etc.)
0054-0056	84-86	Sprung-Vektor für Funktionen
0057-0060	87-96	Feld für diverse numerische Berechnungen
0061	97	Akku # 1: Exponent
0062-0065	98-101	Akku # 1: Mantisse
0066	102	Akku # 1: Vorzeichen
0068	104	Akku # 1: Überlauf
0069-006E	105-110	Akku # 2: Exponenten, etc.
006F	111	Vorzeichen-Vergleich: Akku # 1 mit Akku # 2
0070	112	Akku # 1: Rundung
0071-0072	113-114	Zeiger: Cassetten Puffer Länge
0073-008A	115-138	CHRGET Unterroutine (hole Zeichen in BASIC)
007A-007B	122-123	BASIC-Zeiger (innerhalb d. Unterroutine)
008B-008F	139-143	RND-Wert
0090	144	Statuswort ST
0091	145	Tastatur: STOP und RVS-Flagge
0092	146	Zeitkonstante für den Cassettenrecorder
0093	147	LOAD = 0; VERIFY = 1
0094	148	Serielle Ausgabe: Flagge für zurückgestelltes Zeichen
0095	149	Zurückgestelltes Zeichen (seriell)
0096	150	Tape: EOT erhalten
0097	151	Register speichern
0098	152	Wieviele offene Files ?
0099	153	Eingabegerät (normalerweise 0)
009A	154	Ausgabegerät (CMD) (normalerweise 3)
009B	155	Parität der Cassettenaufzeichnungen
009C	156	BYTE erhalten
009D	157	Ausgabecontrolle: Direkt = \$80; RUN = 0
009E	158	Cassettenfehler im 1. Durchgang log/char buffer
009F	159	Cassettenfehler im 2. Durchgang log corrected
00A0-00A2	160-162	Jiffy Clock (HML)

00A3	163	Flagge, serieller Bit Zähler/EOI
00A4	164	Zyklus Zähler
00A5	165	Herunterzählen: Cassette schreiben/Bits zählen
00A6	166	Zeiger: Buffer für Cassette
00A7	167	Cassette schreiben
00A8	168	Cassette schreiben
00A9	169	Schreibe Startbit/Lesebit Fehler/stbit
00AA	170	Cassetten Abfrage
00AB	171	Schreibe Leader auf Cassette
00AC-00AD	172-173	Zeiger: Puffer für Cassette; Scrolling
00AE-00AF	174-175	Adressen: Ende des Tonbandes, Ende des Programms
00B0-00B1	176-177	Cassette: Zeitkonstante
00B2-00B3	178-179	Zeiger: Beginn des Cassettenbuffers
00B4	180	Timer für Tonband (1 = enable) Bitgezählt
00B5	181	Tonband EOT/RS232; nächstes zu sendene Bit
00B6	182	Fehler wegen Lesen eines Zeichens
00B7	183	# Zeichen im Filename
00B8	184	Momentanes logisches File
00B9	185	Momentane sekundäre Adresse
00BA	186	Momentanes Gerät (Device)
00BB-00BC	187-188	Zeiger für File-Namen
00BD	189	Shift-Taste
00BE	190	Anzahl an Blöcken die zum Schreiben/Lesen übrig sind
00BF	191	Puffer für serielles Wort
00C0	192	Cassettenmotor schalten
00C1-00C2	193-194	Startadresse für I/O
00C3-00C4	195-196	Betriebssystem Aufbau-Zeiger
00C5	197	Taste die momentan gedrückt wird
00C6	198	Anzahl der Zeichen im Tastaturpuffer
00C7	199	Bildschirm negative Anzeige (Flag)
00C8	200	Zeiger: Ende der Zeile bei Eingabe
00C9-00CA	201-202	Eingabecursor (Reihe, Spalte)
00CB	203	Welche Taste; 64 = keine Taste gedrückt
00CC	204	Cursor einschalten; (0 = Blinkcursor)
00CD	205	Timer für Cursor
00CE	206	Zeichen unterm Cursor
00CF	207	Cursor in Blinkphase
00D0	208	Eingabe vom Bildschirm/von Tastatur
00D1-00D2	209-210	Zeiger zur Bildschirmzeile
00D3	211	Position des Cursors in obengenen. Zeile
00D4	212	0 = direkter Cursor; ansonsten programmiert
00D5	213	Länge der momentanen Zeile auf dem Bildschirm
00D6	214	Reihe wo der Cursor ist

00D7	215	Letzter inkey/checksum/puffer
00D8	216	# an ausstehenden Einfügungen (Insert)
00D9-00F0	217-240	Bildschirmzeilen Link-Tabelle
00F1	241	Dummy Bildschirm Link
00F2	242	Anzeiger für Bildschirmreihe
00F3-00F4	243-244	Zeiger: Bildschirmfarbe
00F5-00F6	245-246	Zeiger: Tastatur
00F7-00F8	247-248	Zeiger: RS232 Rcv, Empfangen
00F9-00FA	249-250	Zeiger: RS232 Tx, senden
00FB-00FE	251-254	Betriebssystem; freier Zero page-Raum
00FF	255	BASIC Speicher
0100-010A	256-266	Floating to ASCII Arbeitsbereich
0100-103E	256-318	Band-Fehler
0100-01FF	256-511	Gebiet des Prozessor-Stacks
0200-0258	512-600	Eingabepuffer für BASIC
0259-0262	601-610	Tabelle für logische Files
0263-026C	611-620	Device # Tabelle
026D-0276	621-630	Sekundäre Adress-Tabelle
0277-0280	631-640	Puffer für Tastatur
0281-0282	641-642	Beginn des Speichers für Betriebssystem
0283-0284	643-644	Ende des Speichers für Betriebssystem
0285	645	Serieller Bus timeout Flag
0286	646	Code für momentane Farbe
0287	647	Farbe unter dem Cursor
0288	648	Bildschirm-Page
0289	649	Maximale gröÙe des Tastaturpuffers
028A	650	Tastenwiederholung (128 = wiederhole alle Tasten)
028B	651	Repeat speed Zähler
028C	652	Repeat Verzögerungs-Zähler
028D	653	Tastatur Shift-Tasten/Kontroll-Flagge
028F-0290	655-656	Zeiger: Liste für Tastaturentschlüsselung
0291	657	Shift: 0 = frei; 128 = verriegelt (Shift-Taste)
0292	658	Automatisches scroll down Flag; 0 = an, < > = aus
0293	659	RS232: Controllregister
0294	660	RS232: Befehlsregister
0295-0296	661-662	Nicht-Standard (Bit time/2-100)
0297	663	RS232: Statusregister
0298	664	Anzahl der Bits, die gesendet werden müssen
0299-029A	665-666	Baud-Rate
029B	667	RS232: Zeiger Aufnahme
029C	668	RS232: Zeiger Eingabe
029D	669	RS232: Zeiger Senden
029E	670	RS232: Zeiger Ausgabe

029F-02A0	671-672	Hält IRQ während der Cassettenrecorder läuft	
02A1-02FF	673-767	Programm indirects	
0300-0301	768-769	Fehlermeldung-Link	
0302-0303	770-771	Warmstart-Link	
0304-0305	772-773	Berechnung der BASIC-Tokens	
0306-0307	774-775	Ausdruck Tokens Link	
0308-0309	776-777	Start neuen Basic Code-Link	
030A-030B	778-779	GET arithmetisches Element-Link	
030C	780	Speicher für 6502 .A-Register	
030D	781	Speicher für 6502 .X-Register	
030E	782	Speicher für 6502 .Y-Register	
030F	783	Speicher für 6502 .P-Register	
0314-0315	788-789	Hardware interrupt vector	(EABF)
0316-0317	790-791	Break interrupt vector	(FED2)
0318-0319	792-793	NMI interrupt vector	(FEAD)
031A-031B	794-795	OPEN vector	(F40A)
031C-031D	796-797	CLOSE vector	(F34A)
031E-031F	798-799	Set - input vector	(F2C7)
0320-0321	800-801	Set - output vector	(F309)
0322-0323	802-803	Restore I/O vector	(F3F3)
0324-0325	804-805	INPUT vector	(F20E)
0326-0327	806-807	Output vector	(F27A)
0328-0329	808-809	Text-STOP vector	(F770)
032A-032B	810-811	GET vector	(F1F5)
032C-032D	812-813	Abort I/O vector	(F3EF)
032E-032F	814-815	User vector	(FED2)
0330-0331	816-817	Link to load RAM	(F549)
0332-0333	818-819	Link to save RAM	(F685)
033C-03FB	828-1019	Cassettenpuffer	
0400-0FFF	1024-4095	3K Expansion RAM-Erweiterung	
1000-1DFF	4096-7679	Nutzbarer Platz für Basic	
1E00-1FFF	7680-8191	Bildschirmspeicher	
2000-3FFF	8192-16383	Block 1 für 8K RAM/ROM-Erweiterung	
4000-5FFF	16384-24575	Block 2 für 8K RAM/ROM-Erweiterung	
6000-7FFF	24576-32767	Block 3 für 8K RAM/ROM-Erweiterung	

---

Merke:

Wenn der Speicher (Block 1, 2, 3) erweitert wird, so wird folgendes für Basic ersetzt:

1000-11FF	4096-4607	Bildschirmspeicher
1200-?	4608-?	Nutzbarer Platz für BASIC
9400-95FF	37888-38399	RAM für Farbe

---

8000—8FFF	32768-36863	4K ROM für Zeichengenerator			
8000—83FF	32768-33791	Großbuchstaben und Graphikzeichen			
8400—87FF	33792-33815	Reverse — Großbuchstaben u. Grafik			
8800—8BFF	33816-35839	Groß- und Kleinbuchstaben			
8C00—8FFF	35840-36863	Umgekehrte Groß- und Kleinbuchstaben			
9000—900F	36864-36879	Adressen der VC-20 Chip-Register			
9000	36864	Bit 0 — 6 horizontale Zentrierung			
9001	36865	Vertikale Zentrierung			
9002	36866	Bit 0 — 6 setzt Anzahl der Spalten Bit 7 ist Teil der Videomatrix-Adresse			
9003	36867	Bit 1 — 6 setzt Anzahl der Reihen Bit 0 setzt 8 x 8 oder 16 x 8 Zeichen			
9004	36868	TV Raster für Strahl			
9005	36869	Bit 0 — 3: Beginn des Zeichenspeichers Bit 4 — 7: Rest d. Bildschirmadresse Bit 3, 2, 1, 0 CM Startadresse			
			HEX	DEC	
	0000	ROM	8000	32768	
	0001		8400	33792	
	0010		8800	34816	
	0011		8C00	35840	
	0100		9000	36864	nicht verfügbar
	0101		9400	37888	nicht verfügbar
	0110		9800	38912	nicht verfügbar
	0111		9C00	39936	nicht verfügbar
	1000	RAM	0000	0000	
	1001		xxxx		
	1010		xxxx		nicht verfügbar
	1011		xxxx		
	1100		1000	4096	
	1101		1400	5020	
	1110		1800	6144	
	1111		1C00	7168	
9006	36870	Horizontale Position des Lichtgriffels			
9007	36871	Vertikale Positon des Lichtgriffels			
9008	36872	X-Wert vom Paddle (digital)			
9009	36873	Y-Wert vom Paddle (digital)			
900A	36874	Frequenz für Oszillator 1 (niedrig) (An 128 — 255)			
900B	36875	Frequenz für Oszillator 2 (mittel)			
900C	36876	Frequenz für Oszillator 3 (hoch) (An 128 — 255)			

900D	36877	Frequenz für Geräuschgenerator		
900E	36878	Bit 0 – 3 setzt Lautstärke für alle Geräusche Bit 4 – 7 sind zusätzliche Farbinformationen		
900F	36879	Farbregister für Bildschirm und den Bildrand Bit 4 – 7: Hintergrundfarbe Bit 0 – 2: Randfarbe Bit 3: wählt aus, ob umgekehrter oder ob normaler Modus		
9110–93FF	37136	I/O-Block 0		
9110–91FF	37136-37151	6522 PIA # 1		
9110	37136	Ausgaberegister von Ton B		
		PIN 6522 DESCRIPTION EIA ABV		
		ID ID		
		-----	-----	-----
		C PB0 Received data (BB) Sin		
		D PB1 Request to Send (CA) RTS		
		E PB2 Data terminal ready (CD) DTR		
		F PB3 Ring indicator (CE) RI		
		H PB4 Received line signal (CF) DCD		
		J PB5 Unassigned ( ) XXX		
		K PB6 Clear to send (CB) CTS		
		L PB7 Data set ready (CC) DSR		
		B CB1 Interrupt for Sin (BB) Sin		
		M CB2 Transmitted data (BA) Sout		
		A GND Protective ground (AA) GND		
		N GND Signal ground (AB) GND		
9111	37137	Ausgaberegister von Tor A (PA2) Bit 2 = Joystick 3 (PA3) Bit 3 = Joystick 1 (PA4) Bit 4 = Joystick 2 (PA5) Bit 5 = Lichtgriffel/Feuerknopf (PA6) Bit 6 = Cassetten Schalter Abfrage (PA7) Bit 7 = ATN seriell Ausgabe		
9112	37138	Richtungsregister für TOR B		
9113	37139	Richtungsregister für TOR A		
9114	37140	Timer 1 low Byte		
9115	37141	Timer 1 high Byte & Zähler		
9116	37142	Timer 1 low Byte		
9117	37143	Timer 1 high Byte		
9118	37144	Timer 2 low Byte		
9119	37145	Timer 2 high Byte		
911A	37146	Schieberegister		
911B	37147	Hilfskontrollregister (Auxiliary C. R.)		

911C	37148	Peripherer Kontrollregister (CA1, CA2, CB1, CB2) CA1 = Restore Taste CA2 = Kontroller für Cassettenmotor CB1 = Unterbrechungssignal für erhaltene RS232-Daten CB2 = Übermittelte RS232-Daten
911D	37149	Unterbrechungs-Flagge
911E	37150	Unterbrechungs-Register
911F	37151	Tor A (Abfrage Cassetten-Schalter)
9120-912F	37152-37167	6522 PIA # 2
9120	37152	Ausgaberegister für Tor B Tastatur Abfrage der Spalten (PB3) Bit 3 = Cassetten Schreibleitung (PB7) Bit 7 = Joystickanschluß 0
9121	37153	Ausgaberegister für Tor A
9122	37154	Datenrichtungsregister Tor B
9123	37155	Datenrichtungsregister Tor A
9124	37156	Timer 1, low, byte latch
9125	37157	Timer 1, high byte latch
9126	37158	Timer 1, low byte counter
9127	37159	Timer 1, high byte counter Timer 1, used for the 60 time/second interrupt
9128	37160	Timer 2, low byte latch
9129	37161	Timer 2, high byte latch
912A	37162	Schieberegister
912B	37163	Hilfskontrollregister
912C	37164	Peripheres Kontrollregister CA1 Cassette lesen CA2 Serieller Takt Ausgang CB1 Serieller Daten Ausgang CB2 Serieller SRQ Ausgang
912D	37165	Unterbrechungs-Flag-Register
912E	37166	Unterbrechung möglich-Register
912F	37167	Ausgaberegister Port A
9200	37376	I/O Block 2
9300	37632	I/O Block 3
9400-95FF	37888-38399	Farb-RAM mit zusätzlichem RAM am Block 1
9600-97FF	38400-38911	Normalerweise Platz für Farb-RAM
A000-BFFF	40960-49152	8 K Block für ROM-Erweiterung
C000-DFFF	49152-57343	8 K Basic ROM
E000-FFFF	57344-65535	8 K Betriebssystem ROM (Kernal)

## Unterroutinen des Betriebssystems, die vom Anwender aufgerufen werden können

Name	Adresse		Funktion
	Hex	Dez	
ACPTR	\$FFA5	65445	Eingabe eines Zeichens vom IEEE-Bus her (seriell)
CHKIN	\$FFC6	65478	Eröffnen eines logischen Files in R(X) für Eingabe
CHKOUT	\$FFC9	65481	Eröffnen eines logischen Files in R(X) für Ausgabe
CHRIN	\$FFCF	65487	Eingabe eines Zeichen vom Kanal her
CHROUT	\$FFD2	65490	Ausgabe eines Bytes vom Accu nach Kanal
CIOUT	\$FFA8	65448	Ausgabe eines Zeichens an den seriellen Port
CLALL	\$FFE7	65511	Schließen aller Kanäle und Files
CLOSE	\$FFC3	65475	Schließen eines bestimmten Files
CLRCHN	\$FFCC	65484	Schließen der Ein- und Ausgabe Kanäle
GETIN	\$FFE8	65512	Hole Zeichen von der Tastatur (Tastaturpuffer)
IOBASE	\$FFF3	65523	Liefert die Basisadresse der Ein-/Ausg. Vorrichtung
LISTEN	\$FFB1	65457	Kommando-Vorrichtung auf dem seriellen Bus für "LISTEN"
LOAD	\$FFD5	65493	Lade RAM von einem Device (Vorrichtung)
MEMBOT	\$FF9C	65436	Lese und lege die untere Speichergrenze fest
MEMTOP	\$FF99	65433	Lese und lege die obere Speichergrenze fest
OPEN	\$FFC0	65472	Eröffne ein logisches File
PLOT	\$FFF0	65520	Lese und setze X,Y Cursor Position
RDTIM	\$FFDE	65502	Lese die Echtzeituhr
READST	\$FFB7	65463	Lese Ein-/Ausgabe Status-Wort
RESTOR	\$FF87	65415	Restore default Ein-/Ausgabe Vektoren (Standard)
SAVE	\$FFD8	65496	Speichere RAM nach Device
SCNKEY	\$FF9F	65439	Tastaturabfrage
SCREEN	\$FFED	65517	Lese X,Y Organisation des Bildschirms
SECOND	\$FF93	65427	Sende die "secondary Adresse" nach LISTEN
SETLFS	\$FFBA	65466	Setze logische, erste und zweite Adresse
SETMSG	\$FF90	65424	Kontrolliere Kernel-Meldungen
SETNAM	\$FFBD	65469	Setze Filename
SETTIM	\$FFDB	65499	Setze Echtzeituhr
SETMO	\$FFA2	65442	Setze "timeout" auf dem seriellen Bus
STOP	\$FFE1	65505	Frage die Stop-Taste ab
TALK	\$FFB4	65460	Kommando an seriellen Bus für TALK
TKSA	\$FF96	65430	Sende die secondary Adresse nach TALK
UDTIM	\$FFEA	65514	Erhöhe die Uhr um eins
UNLSN	\$FFAE	65454	Kommando für serial bus für UNLISTEN
UNTLK	\$FFAB	65451	Kommando für serial bus für UNTALK
VECTOR	\$FF84	65412	Read/Set vektorierten I/O

# ANHANG

## Interessante und praktische Produkte für unsere Leser

### ANHANG

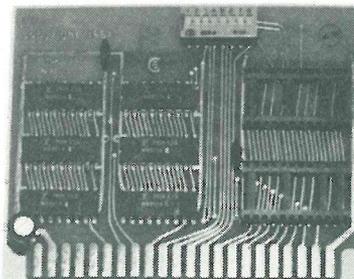
#### Interessante und praktische Produkte für unsere Leser

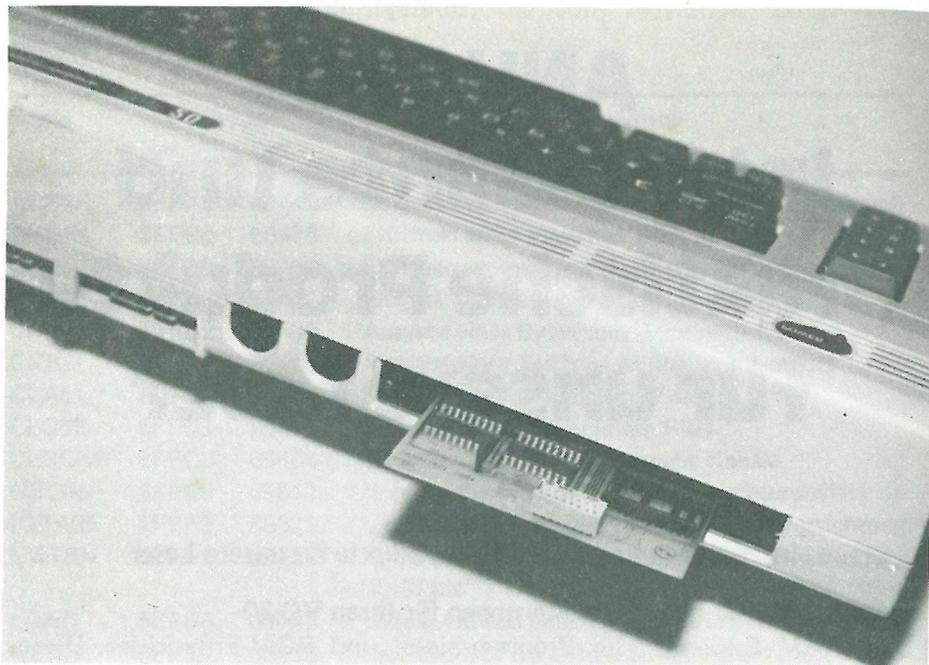
##### Interessante Hardware-Erweiterungen für Ihren VC-20

3 K RAM Expander für Programmierer und ROM-Entwickler. Diese Platine sollte sich jeder VC-20 Besitzer als erstes zulegen. Sie bringt Ihnen eine 3 K RAM Erweiterung im Bereich 0400 – FFF Hex, sowie zwei 24 pin EPROM/ROM Sockel, in die Sie 2 K bis 16 K ROM oder EPROM einstecken können.

Mit Hilfe eines 8 poligen DIP-Schalters können Sie die Anfangsadresse und Sockel Ein/Aus in einem bestimmten Bereich festlegen. Maschinensprachen-Monitore, BASIC-Erweiterungs-Bausteine, hochauflösende Grafik-Bausteine, Spiele in EPROM usw. lassen sich in diese beiden freien Sockel einstecken.

##### 3 K RAM Expander-Platine





### **Die 3 K RAM Expander-Platine, eingesteckt im VC-20**

Die Platine wird über die +5V Spannungsquelle des VC-20 versorgt. Es ist keine zusätzliche Stromversorgung erforderlich. Das BASIC Programmers Utility ROM von ELCOMP ist ein wertvolles Werkzeug in Verbindung mit dieser Karte.

Ganz besonders interessant wird die Karte dadurch, daß Sie dann auch Programme, die für die 8 K PET und CBM-Serie entwickelt wurden, in Ihren VC-20 einlesen und entsprechend anpassen können.

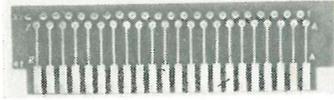
Die Platine wird fertig geprüft geliefert und kann sofort in den Expansionsport eingesteckt werden.

**Best.-Nr. 4860**

**269,- DM**

### Stecker für Erweiterungsport

Ein recht interessantes und praktisches Zubehöreelement ist dieser Stecker für den Erweiterungsport.



Er kann für Messungen oder Erweiterungen aller Art am Erweiterungsport verwendet werden. Er dient auch dazu, um Platinen mit unterschiedlicher S-44 Steckerbelegung an den VC-20 anzuschließen.

**Best.-Nr. 4847**

**19,80 DM**

### Stecker für den User Port

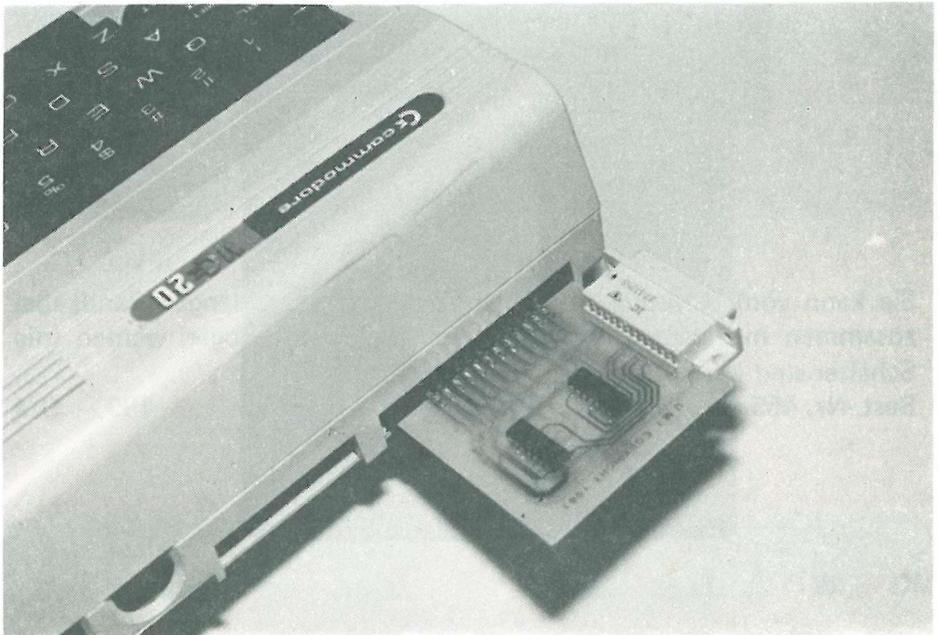
Hier verwendet man die 24 poligen Stecker von TRW CINCH 251-12-50-170/50-24 SN-9 8124.

**Best.-Nr. 4847**

**19,80 DM**

### RS 232 Communications-Interface

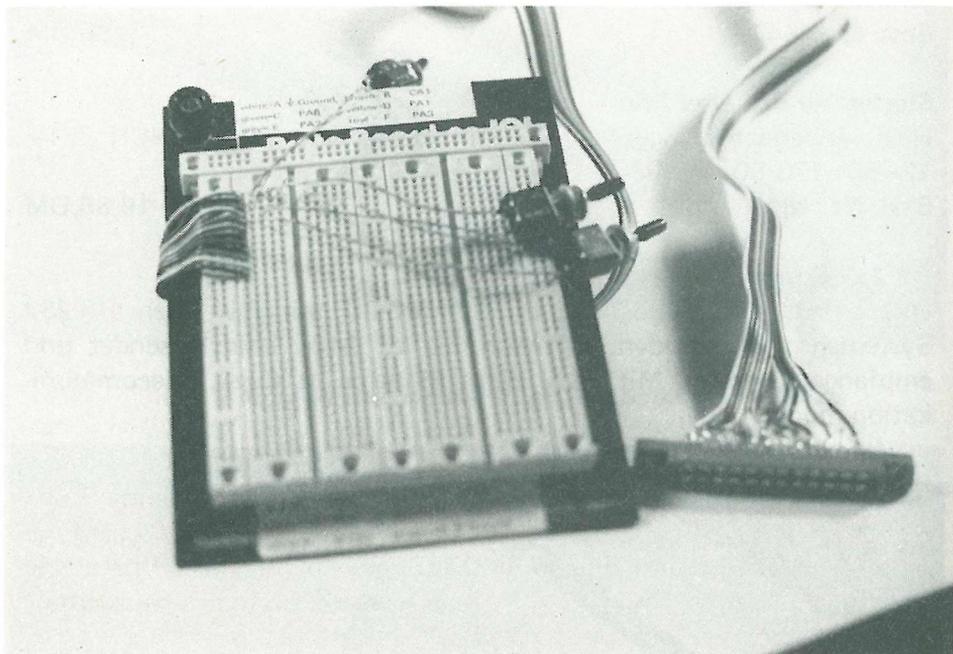
Dieses Interface ermöglicht es den VC-20 mit fast allen RS-232 Systemen zu verbinden. Es kann bis zu 2400 Baud gesendet und empfangen werden. Mit Hilfe eines Modems ist sogar Telekommunikation möglich.



Es sind keine speziellen Maschinensprachentreiberroutinen erforderlich.  
Best.-Nr. 4816 299,— DM

### Externe Experimentierplatine

Wer am User-Port gerne eigene Experimente machen möchte und externe Steuerschaltungen mit Hilfe des Computers betreiben will, dem sei diese Experimentierplatine empfohlen.



Sie kann vom Hofacker Verlag mit einem ca. 1 m langen Bandkabel zusammen mit dem Stecker für den User Port bezogen werden (die Schalter sind in der Lieferung nicht enthalten).

Best.-Nr. 455

199,— DM

## Sehr interessante Programme für Ihren VC-20

Hier eine Auswahl unserer Programme, die wir selbst für ausgezeichnet halten. Alle arbeiten mit der VC-20 Grundversion 3,5 K RAM. Alle arbeiten mit Ton und nützen die phantastischen Grafikeigenschaften des VC-20.

### Alien Blitz

Dieses Spiel gehört zur Spitzenklasse! Ähnlich Alien Invaders, phantastische Farben und Bewegung.

Best.-Nr. 4865

99,- DM

### Amokläufer

Sie befinden sich in einem durch Wände unterteilten Raum zusammen mit böartigen Robotern. Die Roboter in verschiedenen Farben schießen nach Ihnen und Sie können das Feuer natürlich erwidern. Das Spiel kann über die Tastatur oder einen Joystick gespielt werden.



Best.-Nr. 4866

99,- DM

# NOTIZEN

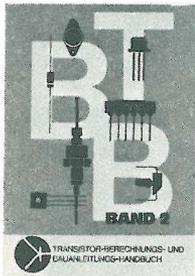
# Hofacker-Bücher

# Deutsch



**TBB-Transistor-, Berechnungs- und Bauleitungs-HB, Band 1, Hofacker**  
 Völlig neu überarbeitete Auflage. Das Buch soll bei der täglichen Arbeit im Labor, in der Werkstatt oder am Elektronik-Hobbytisch Ihnen ein guter Begleiter sein. Berechnungsgrundlagen, Berechnungsbeispiele, Tabellen, Vergleichslisten, Digitaltechnik, Netzgeräte, BASIC-Programme zur Berechnung spezifischer Schaltungen usw. sind in übersichtlicher Form dargestellt. Ca. 300 Seiten.

Best.-Nr. 1 29,80 DM



**TBB-Handbuch, Band 2, Hofacker**  
 Dieses Buch ist die Fortsetzung des erfolgreichen Handbuches, Band 1. Ein Buch, das sich in der Hand des Praktikers bestens bewährt hat. Weitere neueste Schaltbeispiele und Berechnungsgrundl., Experimentier- und Versuchsbeschreibungen. Integrierte Spannungsregler, Wärmeableitung, Operationsverstärker Einführung, RC-Zeitglieder, Transistortester u. v. a.

Best.-Nr. 2 19,80 DM



**Electronic im Auto, H. Gebauer**

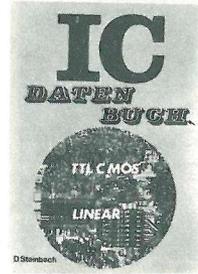
Mit Handbuch für Polizeiradar. Ein Buch für jeden technisch interessierten Autofahrer. Es zeigt Ihnen die vielen Möglichkeiten zur Verbesserung von Sicherheit, Leistung und Fahrkomfort in Ihrem Auto. Thyristorzündung, Beschleunigungsmesser, Drehzahlmesser, Batterie-ladegerät, Alarmanlagen u. v. a.

Best.-Nr. 3 9,80 DM



**IC-Handbuch, C. Lorenz**  
 Ein Handbuch für digitale und lineare integrierte Schaltungen. Daten- und Auswahl-, Vergleichslisten, Gehäuseformen, Grundlagen, viele Schaltbeispiele, Printvorlagen, u. v. a. Alles über TTL-Technik, C MOS, MOS-Schaltungen, integrierte NF-Verstärker u. v. a.

Best.-Nr. 4 19,80 DM



**IC-Datenbuch, D. Steinbach**  
 Daten- und Auswahllisten der gebräuchlichsten integrierten Schaltkreise. Digital und analog. Gerade bei ICs ist es wichtig die Anschlußfolgen genau zu kennen. Die wichtigsten TTL-Schaltkreise, NF-Verstärker, C-Mos Serie, lineare Schaltungen wie Operationsverstärker, Komparatoren, Spannungsregler, Trigger-Schaltungen, u. v. a. Das IC-Datenbuch wird auch Ihnen ein unentbehrlicher Begleiter bei allen Arbeiten mit integrierten Schaltungen sein.

Best.-Nr. 5 9,80 DM



**IC-Schaltungen, D. Steinbach**  
 Hier finden Sie eine gelungene Zusammenstellung der wichtigsten Anwendungsbeispiele aus dem Bereich der integrierten Schaltungen. TTL, C MOS, Linear. Alle Schaltungen sind übersichtlich und klar dargestellt und mit einer kurzen, jedoch sehr genauen Beschreibung versehen. Tastenentprellung, Zähler, Impulsgeber, Codierer, Dekodierer, Datenübertragung, Serien-Parallel-Wandler, Digitalvoltmeter u. v. a.

Best.-Nr. 6 19,80 DM



**Elektronik Schaltungen, Hofacker**  
Die ideale Schaltungssammlung zum Basteln u. Experimentieren, Schaltungen mit Operationsverstärkern, Spannungsreglern, TTL, C-MOS-Schaltkreisen. MOS Uhr mit Wecker-elektronischer Würfel u. v. a.  
Best.-Nr. 7 9,80 DM



**IC Bauanleitungen Handbuch -IC-KIT, C. Lorenz**  
Ein Bauanleitungsbuch mit vielen hochinteressanten Bauanleitungen aus dem Bereich der LSI Schaltungstechnik. Schaltbeispiele mit Printvorlagen zum Selbsterstellen der Leiterplatten mit genauesten Beschreibungen. Hochaktuell und brandneu: Funktionsgenerator XR 2206, MOS-Uhr mit Wecker, programmierbarem Wecktongenerator, Schlummerautomatik, IC-Netzteil, Experimentieranleitung und Grundkurs über Flip Flops, u. v. a. Zu allen Schaltungen finden Sie Platinenvorlagen oder Sie können die Experimentierschaltungen auf der Experimentierplatine WH-1 g durchführen. Über 100 Seiten.  
Best.-Nr. 8 19,80 DM



**Feldeffekttransistoren, C. Lorenz**  
Der Feldeffekttransistor (FET) gehört heute zu den interessantesten Bauteilen überhaupt. Wie man damit experimentiert, wie man seine Funktion versteht und wie man damit brauchbare u. hochinteressante Schaltungen aufbauen kann, zeigt Ihnen dieses Buch. Grundlagen, Kennlinienfelder, Tabellen, Rechenbeispiele, Anschlußbilder und eine Vergleichsliste für Feldeffekttransistoren bilden den Kern dieser umfangreichen Darstellung. Alles in allem finden Sie hier eine praxisnahe und komplette Arbeitsunterlage, mit der Sie im Beruf und auch im Hobby erfolgreich arbeiten können.  
Best.-Nr. 9 9,80 DM



**Elektronik und Radio, C. Lorenz**  
4. Auflage. Völlig neu bearbeitet und stark erweitert. Eine Schritt für Schritt Einführung in die Radiotechnik mit vielen Bildern. Vom einfachen Diodenempfänger (Detektor) bis zu interessanten Sender- und Empfängerschaltungen (Minispione). IC-Radio, IC-Sender, Antennen, Berechnungsgrundlagen, Tabellen u. v. a. Über 150 Seiten.  
Best.-Nr. 10 19,80 DM



**NF-Verstärker, C. Lorenz**  
Grundlagen der integrierten NF-Verstärker, Berechnung von kompletten IC-NF-Verstärkerstufen. Anwendungsbeispiele mit den interessantesten und gebräuchlichsten Standard IC-NF-Verstärkern wie TBA 800, TBA 830, usw. Printvorlagen, Auswahltabellen, Experimentieranleitungen und Anschlußbilder machen dieses Buch zu einem unentbehrlichen Begleiter für alle, die sich m. NF-Verstärkern beschäftigen wollen.  
Best.-Nr. 11 9,80 DM



**BIS, Beispiele integrierter Schaltungen, H. Bernstein**  
Auf über 130 Seiten Anwendungsbeispiele mit integrierten Schaltkreisen, Zeitgeber 555, Funktionsgenerator ICL 8038, Opto Elektronik, Operationsverstärker, Festwertspeicher (ROM), u. v. a.  
Best.-Nr. 12 19,80 DM



**HEH, Hobby Elektronik Handbuch**

C. Lorenz

Das Schaltungsbuch f. jeden Hobbyelektroniker, Schaltbeispiele und Bauanleitungen aus dem gesamten Hobbybereich. Lichtorgeln, Alarmanlagen, Eiswarngerät fürs Auto, PLL-Schaltungen u. v. a.

Best.-Nr. 13

9,80 DM



**IC-Vergleichsliste,**

C. Lorenz

Neben integrierten Schaltungen sind auch noch Transistoren, Dioden, optoelektronische Bauteile, Sicherungen und Röhren in dieses praktische HB aufgenommen worden. Das ideale Nachschlagewerk, wenn es dann gilt das richtige Bauelement für die gewünschte Anwendung zu finden. Dieses Buch wird Ihnen in der Praxis viele Dienste erweisen können. Viele Auswahl- und Vergleichslisten mit Datenangaben. Ca. 800 Seiten.

Best.-Nr. 14

29,80 DM



**Opto-Handbuch,**

C. Lorenz

Das Handbuch für die gesamte Optoelektronik. Eine Einführung und ein ideales Nachschlagewerk. Grundlagen, Definitionen aller Kenngrößen, Opto-Lexikon, Berechnungsgrundlagen, Lichtsender, Lichtempfänger, Anzeigen, Infrarot Detektoren, Optokoppler, Opto-Vergleichsliste, u. v. a. 106 Seiten.

Best.-Nr. 15

19,80 DM



**C MOS Einführung, Entwurf, Schaltbeispiele, Teil 1**

H. Bernstein

Vom C MOS Gatterbaustein über Schieberegister und Zähler bis hin zum C MOS Schreib- Lesespeicher. Insgesamt werden neunzehn interessante und bekannte C MOS Schaltkreise beschrieben. Zu jedem Bauelement sind genaue Daten, Schaltbild und Anwendungsbeispiele angegeben. Im großen Applikationsteil finden Sie: C MOS-Kippstufen, Addierwerke u. Rechenschaltungen, Digital Analog Wandler, Schieberegister für analoge Spannungen, Multiplexsysteme f. analoge Signale u. v. a. Eine komplette Einführung u. gut geeignet für das Selbststudium der C MOS Technik. 140 Seiten.

Best.-Nr. 16

19,80 DM



**C MOS Entwurf u. Schaltbeispiele, Teil 2,**

H. Bernstein

Fortsetzung von Best-Nr. 16. Anwendungsbeispiele mit genauen Schaltungsbeschreibungen und Bauelementunterlagen. Daten, Anschlußbelegungen weiterer wichtiger hochintegrierter C MOS Elemente. Ein komplettes Arbeits- u. Experimentierbuch. C-MOS Uhrenschaltungen, Schieberegisterschaltungen, Parallel-Serien Umsetzung, statische u. dynamische Speicherschaltungen, Zählschaltungen, Digital Analog Wandler, Analog Digital Wandler. Digital Voltmeter, I/O Registerschaltungen. RAM und ROM Anwendungen. Über 140 Seiten.

Best.-Nr. 17

19,80 DM



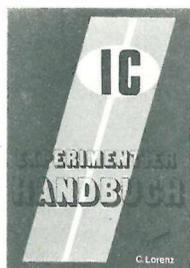
**C MOS Entwurf u. Schaltbeispiele, Teil 3,**

H. Bernstein

Fortsetzung von Best.-Nr. 17. Eine sehr umfangreiche Applikationsammlung mit hochintegrierten C MOS Elementen. Speicher- und Steuerschaltungen, Multiplex- und Datenbussysteme, Liquid Cristal Anzeigen, Uhrenschaltungen, PLL-Schaltungen, Optoelektronik in Verbindung mit C MOS. Aufbau und Wirkungsweise der Prozeß-rechentechnik, Arithmetische Logische Einheiten (ALU) u. andere wichtige Funktionen aus der Prozeß-rechentechnik. RAMs, ROMs, und FIFO-Speicherschaltungen.

Best.-Nr. 18

19,80 DM



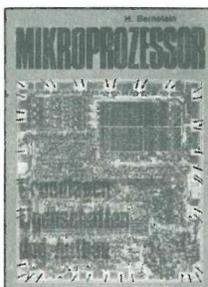
**IC Experimentier Handbuch -IC, -EX,** C. Lorenz  
 Eine sehr umfangreiche Schaltungs- und Bauanleitungssammlung mit neuen, jedoch meist beim Fachhandel erhältliche Standard ICs. Rechnerschaltungen, Mikroprozessoren, I/O-Schaltungen, Stoppuhren, druckende und anzeigende Rechner, Digitalvoltmeter, Hilfschaltungen für den Elektronik Experimentierer, A/D-Wandler, Frequenzzähler u. v. a. Viele Schaltungen können auf der IC KIT Experimentierplatine WH-1g aufgebaut werden.  
 Best.-Nr. 19 19,80 DM



**Operationsverstärker,** C. Lorenz  
 Dieses Buch umfaßt das gesamte Gebiet der linearen Schaltungstechnik und stellt ein in dieser Preislage bisher noch nie dagewesenes Nachschlagwerk und Einführungshandbuch dar. Bestens geeignet für das Selbststudium. Nach einer pädagogisch geschickt gemachten Einführung folgen theoretische Arbeitsunterlagen und die zugehörigen Schaltbeispiele mit Daten und Gehäuseanschlüssen. Dieses wertvolle Buch dürfte seinen Platz auch bei Ihren Arbeitsunterlagen finden, und wird dann immer von Nutzen sein, wenn es um die Lösung von nicht routinemäßigen Aufgaben geht. Über 150 Seiten.  
 Best.-Nr. 20 19,80 DM



**Digitaltechnik Grundkurs,** C. Lorenz  
 Ein Einführungskurs in die Digitaltechnik für Anfänger und Fortgeschrittene. Ein Fachbuch für den programmierten Selbstunterricht. Der ideale Kurzlehrgang für das Selbststudium. Der Kurs vermittelt Ihnen alle wichtigen Grundkenntnisse vom TTL-Gatter bis zum Mikroprozessor und Lösung von Schaltungsaufgaben durch Software. Viele Versuchsaufbauten u. Experimente aus diesem Kurs können auf der IC-KIT Platine WH-1g durchgeführt werden. Grundlagen, Gatter, Zähler, programmierbare Zähler, IC-Tester, Schieberegister, Speicher, Mikroprozessoren u. v. a.  
 Best.-Nr. 21 19,80 DM



**Mikroprozessoren, Eigenschaften u. Aufbau, Teil 1,** H. Bernstein  
 Grundlagen, Eigenschaften u. Aufbau von Mikroprozessoren. Organisation von Recheneinheiten und Mikroprogr. Programmierung und Klassifizierung v. Mikroprozessoren. Ablaufdiagramm, Flußdiagramm. Ein Cip-Technik und Multi Chip-Technik, Transfer- und Sprungfunktionen. Speichertechnik: RAMs ROMs, FIFO, FILO. Programmierbare logische Arrays (PLA). Anwendungsbeispiele u. Anwendungsbereiche. Über 120 Seiten.  
 Best.-Nr. 22 19,80 DM



**Elektronik Grundkurs,** C. Lorenz  
 Eine leichtverständliche und pädagogisch geschickt gemachte Einführung in die Technik der elektronischen Schaltungen. Ein Kurzlehrgang und Schnellkurs zugleich. Aber auch ein recht brauchbares Nachschlagwerk für den fortgeschrittenen Elektroniker. Mit wenig Mühe können Sie sich hier die Grundkenntnisse der elektronischen Schaltungspraxis aneignen. Das Buch schafft die Voraussetzungen für ein erfolgreiches und sicheres Arbeiten mit interessanten Schaltkreisen modernster Technologien. Unentbehrlich f. das Experimentieren mit den heutigen modernen hochintegrierten Schaltkreisen.  
 Best.-Nr. 23 9,80 DM



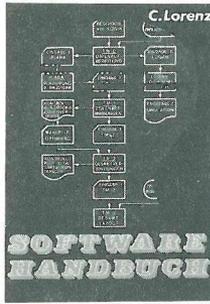
**Mikrocomputer Technik,** Hans Peter Blomeyer-Bartenstein  
 Völlig neue Auflage: Herbst 1979. In diesem Buche finden Sie eine umfassende, einführende u. weiterführende Hilfe zum Einstieg in die Mikrocomputertechnik mit vielen Schalt- und Programmierbeispielen. Als praktische Betrachtungsgrundlage dient das supermoderne Microcomputerkonzept Z80A von ZILOG. Das Buch geht auf alle wichtigen Zusammenhänge ein und erklärt diese dem Leser so ausführlich, daß kaum noch Fragen offen bleiben. über 240 Seiten.  
 Best.-Nr. 24 29,80 DM



**Hobby Computer Handbuch,**  
C. Lorenz

Eine leicht verständliche Einführung in die Mikrocomputertechnik. Diese sehr umfangreiche Einführung in die Microcomputertechnik dürfte zu diesem Preis einmalig sein. Auf über 450 Seiten finden Sie – Grundlagen der Computer- und Microcomputer-Technik, was ist ein Microcomputer? Microcomputer KITS, Einplatinencomputer, CAT, OSI, POLY 88 u. v. a. Das ideale allumfassende Buch für den Micro-computertechniker. Für Industrie-anwendung ebenso geeignet wie für den Hobby-Computer-Fan.

Best.-Nr. 25 29,80 DM



**Mikroprozessor Software Handbuch**  
C. Lorenz

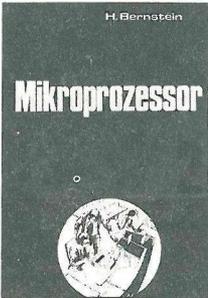
Grundlagen und Einführung in die Mikrocomputerprogrammierung und Programmiersprachen (BASIC, FORTRAN, Assembler-Sprachen). Zusammenstellung der wichtigsten Befehlslisten: 8080, Z80, M6800, 6502 etc. Ein Software Handbuch für jeden, der mit Mikroprozessoren oder Mikrocomputern zu tun hat.

Best.-Nr. 27 29,80 DM

ohne Abbildung  
Erscheint ca. Ende 1981

**Mikrocomputer Datenbuch, Lorenz**  
Eine übersichtliche u. sehr informative Zusammenstellung der wichtigsten Mikroprozessorbausteine auf dem Markt. 8080A, 8085, 8048, Z80, Z8, 6500, 6800, 2650, 1802, F8, 3870, 6809, PACE u. v. a. Daten, Anschlußbilder, wichtige technische und elektrische Daten, Architektur, grundlegende Eigenschaften. Zu jedem Mikroprozessor werden dann auch noch die peripheren Bausteine sowie RAM und ROM Elemente behandelt. Das ideale Handbuch für jeden modernen Elektroniker. Ideal f. den Computer-Service. (Englisch)

Best.-Nr. 29 49,80 DM



**Mikroprozessor, Teil 2, H. Bernstein**  
Die Fortsetzung von Best.-Nr. 22. Technologie von Mikroprozessor- und Speicherbausteinen. Festwertspeicher, PROM, EPROM, FIFO, Schieberegister, MPR-, ARL- und SAR-Register. Aufbau eines Mikroprozessorsystems mit 8080, RAM- und ROM-Schnittstellen. Befehls-satz 8080. Über 120 Seiten.

Best.-Nr. 26 19,80 DM



**Microcomputer Lexikon u. Wörterbuch von A – Z,**  
C. Lorenz

Englisch/Deutsch – Der Fachausdruck wird übersetzt, ausführlich erklärt und erläutert. Deutsch/Englisch – Übersetzung des Fachausdrucks für jeden, der sich heute mit der modernsten Elektronik beschäftigt. Viele engl. Ausdrücke werden heute in der Elektronik, Computer- und Mikroprozessortechnik verwendet und oft fehlt uns eine genaueste und präzise Erläuterung. Ein Lexikon und Wörterbuch in einem einzigen Buch vereint.

Best.-Nr. 28 29,80 DM



**Aktiv Training Microcomputer 8080 8085,**  
S. Lehrer

Dieses Werk mit über 360 Seiten beschäftigt sich ausschließlich mit den Microcomputerbausteinen und Peripherieelementen der 8080A und 8085 Mikroprozessoren. Grundlagen Einführung 8080 Microcomputer, Programmieren in Assembler, PL/M Compiler, Speicherbausteine u. v. a. Ideal für jeden, der ein System mit 8080, 8085 CPU besitzt. Auch der Z-80 Systembesitzer kann von diesem Buch viel profitieren. Am Schluß des Buches finden Sie noch ca. 150 Programm listings in Maschinensprache (Nützliche Utilities und Spielprogramme).

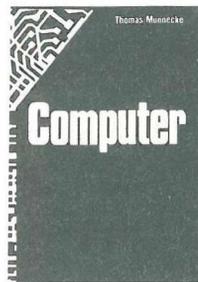
Best.-Nr. 30 49,80 DM



**57 Programme in BASIC, Lorenz**  
 Ein Buch mit techn.-wissenschaftlichen Programmen u. einer großen Anzahl von Spielprogrammen in BASIC (Games). Ein Buch für jeden, der sich mit dem faszinierenden Hobby der Mikrocomputertechnik befassen will. Alle Listings sind in BASIC und können auf den meisten Personal Computer Systemen gefahren werden.  
 Best.-Nr. 31 39,00 DM



**Mikrocomputer Programmierbeispiele für 2650, Dr. J. Hatzenbichler**  
 Eine Einführung in die Programmierung von Mikrocomputern anhand des Prozessors 2650 von Signetics. Viele Programmierbeispiele in Maschinensprache, die Sie auf einem preiswerten Mikroprozessorsystem MIKIT 2650-P2 ausführen können. Zeitschleifenprogr., Blinkschaltung, Lauflicht, Stufenzähler, Stopuhr, Reaktionszeittester, u. a. Zu diesem Buch ist auch ein komplett aufgebautes und getestetes Mikrocomputersystem erhältlich, auf dem Sie alle beschriebenen Programme selbst ausführen können. Über 120 Seiten. (Nur solange Vorrat reicht, wird nicht mehr nachgedruckt.)  
 Best.-Nr. 33 19,80 DM



**Der freundliche Computer – Was können Sie mit einem Personal Computer anfangen? T. Munnecke**  
 Das Buch soll Ihnen auf die im Titel gestellte Frage eine ausführliche Antwort geben. Es eignet sich für alle, die bisher viel über Mikros gehört haben und gerne ausführlicher Bescheid wissen möchten. Viele interessante Fakten – Welche Computersprache? Welche Anwendungen? Welches Gerät soll ich mir kaufen? 153 Seiten.  
 Best.-Nr. 35 29,80 DM



**ATARI BASIC Handbuch**  
 Das komplette Einführungs- und Programmierhandbuch für die neuen ATARI-Computer ATARI 400/800. Anhand von vielen Beispielen wird die Leistungsfähigkeit des ATARI-Computer-Systems gezeigt. Auch für den Anfänger ideal als Einstieg gedacht. Das Buch kann auch als Kurs (Lehrgang) für die allg. BASIC-Sprache verwendet werden.  
 Best.-Nr. 32 29,80 DM



**TINY BASIC Handbuch, Hermann**  
 Das erste deutschsprachige Handbuch über Tom Pittman's TINY BASIC. Eine Einführung in die TINY BASIC-Programmiersprache. Wie kann ich meinen Computer (KIM-1) erweitern und BASIC programmieren. Systemvorschläge. Viele Programmierbeispiele, Tricks und Kniffe.  
 Best.-Nr. 34 19,80 DM

ohne Abbildung  
**Microcomputer und Roboter**  
 Ein Buch für denjenigen, der sich externe Schaltungen für Microcomputer bauen möchte, die roboterartige Funktionen ausführen können. Spracherkennung, Analog-/Digital-Wandler, Digital-/Analog-Wandler, Ultraschallsensoren, Lichtschranken, Tonerzeugung u. v. a. Erscheint Anfang 1982.  
 Best.-Nr. 36 29,80 DM

ohne Abbildung  
**Oszillographen Handbuch**  
 Ein Buch für jeden, der seinen Oszillographen optimal nutzen will. Der Anfänger, der noch nie einen Oszillographen benutzt hat, wird auf einfache Weise mit der Technik und Handhabung vertraut gemacht. Auch die Anwendung in Zusammenhang mit den modernen Mikrocomputersystemen wird beschrieben. Oszillograph als alphanumeres Darstellgerät u. v. mehr. Erscheint ca. Anfang 1982.  
 Best.-Nr. 103 19,80 DM

ohne Abbildung

### 1000 Elektronik Schaltungen

Ein ideales Handbuch für jeden, der öfter eine Schaltung zur Lösung eines bestimmten Problems sucht. Tausend Schaltungen aus fast allen Bereichen der Elektronik. Industrielle Steuerschaltungen, Microcomputer, Peripherie, Hobby-Elektronik-Schaltungen u. v. a. mehr. Ein Buch, daß bei keinem Elektroniker fehlen sollte. Erscheint ca. Mitte 1982.

Best.-Nr. 104 49,00 DM

ohne Abbildung

### Praktische Antennentechnik

Ein Buch für jeden Funkamateure oder Hobbyfunker. Grundlagen, Einführung, praktische Beispiele, Berechnungsgrundlagen u. v. a. Erscheint ca. Anfang 1982.

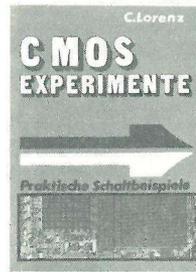
Best.-Nr. 107 19,80 DM



### TTL-Experimentierbuch

Eine kleine Einführung in die Digitaltechnik. Grundlagen der Digitaltechnik kurz erklärt. Bauanleitung für einen praktischen Logik-Tester. Viele Experimente u. Anwendungsschaltungen mit dem TTL Gatterbaustein 7400. Das ideale Einführungsbuch in die Digitaltechnik.

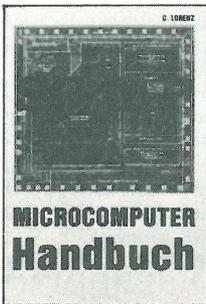
Best.-Nr. 105 1 5,00 DM



### CMOS-Experimentierbuch

Eine kleine Einführung in die CMOS Schaltungstechnik. CMOS Grundlagen, Behandlungshinweise für CMOS Bausteine, Zusammenst. der wichtigsten CMOS Bausteine und deren Anschlußbilder. Viele praktische CMOS Schaltbeispiele. Ideal für jeden Elektroniker.

Best.-Nr. 106 1 5,00 DM



### Handbuch für SC/MP

Ein echtes Handbuch für SC/MP (INS8060) Microcomputerbesitzer und solche die es werden wollen. Komplette Einführung mit vielen Schaltbeispielen, Schaltbildern und Programmlistings in Maschinensprache und TINY BASIC. Komplettes ELBUG-Listing, CPU-Karte, RAM-Karte, ROM-Karte, ROM-Programmierer, Cassetten-Interface, Fernsehinterface, Netzteil, Hex Ein-/Ausgabe, SC/MP Einkartenmicrocomputer u. v. a. 330 Seiten.

Best.-Nr. 108 29,80 DM



### 6502 Microcomputer Programmierung, P. Heuer

Eine deutschsprachige Einführung in die Maschinensprachenprogrammierung anhand des 6502 Microcomputers. Ein echtes Anleitungsbuch zum Einstieg in die Microcomputertechnik mit Hilfe des KIM-1. Viele Programmierbeispiele, die von einem Pädagogen speziell für Anfänger entwickelt wurden. Auch PET, AIM, SYM und ATARI-Besitzer brauchen dieses Buch.

Best.-Nr. 109 29,80 DM



### Programmierhandbuch für PET

Ein für den PET produziertes Buch. Es beginnt da, wo Ihr mitgeliefertes Handbuch aufhört. Einführung Maschinensprachenprogrammierung, Assembler, Ein-/Ausgabeprogrammierung, Programmiertricks, Analog / Digital-Wandler, Graphik, Spracherkennung u. v. a. mehr. Viele Listings, die Sie selbst eintippen können. Viele Informationen eignen sich auch für VC-20-Besitzer. 324 Seiten.

Best.-Nr. 110 29,80 DM



**Programmieren mit TRS-80, Stübs**  
 Das erste in einem deutschen Verlag produzierte Buch über den erfolgreichen Personal Computer von TANDY. Ein Buch für jeden, der einen TRS-80 bereits besitzt oder vor der Entscheidung steht, welchen Computer er sich anschaffen soll. Einführung, Programmiertricks, Erweiterungen, Maschinenprogrammierung und viele Programme (Listing mit Beschreibung). 202 Seiten.  
 Best.-Nr. 111 29,80 DM



**BASIC Programmierhandbuch**  
 Einführung und Nachschlagewerk. Speziell für die BASIC-Versionen der modernen Microcomputersysteme. Jeder Befehl wird ausführlich beschrieben und ein Beispielprogramm gezeigt. Sehr übersichtlich und praktisch. Am Schluß finden Sie ein komplettes BASIC-Programm, das Ihnen über einen Computer BASIC lehrt.  
 Best.-Nr. 113 19,80 DM



**Der Microcomputer im Kleinbetrieb**  
 Das Buch für jeden Geschäftsmann. Auf über 170 Seiten erfahren Sie, was Sie als Gewerbetreibender oder freiberuflich Tätige über Microcomputer und die Anwendung wissen sollten. Geschichtliche Hintergrund Geräteauswahl, Beispiele aus der Praxis, Programmbeispiele wie z. B. Textverarbeitung, Reisebüro, Ladenkasse, Adressverwaltung, u. v. a. Betriebswirtschaftliche Auswertung, Finanzbuchhaltung, Erfolgsanalyse mit dem Microcomputer, Liquiditätsrechnung, kurzfristige Erfolgsrechnung, Microcomputer für Freiberufler, Grundlagen der Finanzbuchhaltung für Microcomputeranwender. Dieses Buch kann Ihnen als Geschäftsmann für die Zukunft tausende einsparen.  
 Best.-Nr. 114 39,80 DM



**PASCAL Handbuch, E. Flögel**  
 Von BASIC zu PASCAL. Ein Einführungs- Lehr und Arbeitsbuch für jeden der sich mit PASCAL beschäftigen will oder muß. Viele Programmbeispiele, viele Tricks wie PEEK und POKE, Einbinden von Maschinenprogrammen u. v. a.  
 Best.-Nr. 112 29,80 DM



**16 Bit Microcomputer, J. Koller**  
 Einführung, Daten, Eigenschaften, Anwendungen. Dieses Werk ist eine echte Sensation! Alle 16 Bit Prozessoren werden beschrieben und erläutert. Applikationsbeispiele, Programmierhinweise. TMS 9900, 8086, Z8000, MC 68000, NS 16000, IAPX 486, IAPX 432. Über 370 Seiten.  
 Best.-Nr. 116 29,80 DM

ohne Abbildung

**FORTRAN für Heimcomputer**  
 Einführung in die FORTRAN-Programmiersprache mit vielen Beispielen. Grundsätzliches über die verschiedenen Microcomputersysteme, die bereits mit FORTRAN-Compiler lieferbar sind. Allgemeine Übersicht, Tips und Hinweise. Erscheint ca. Ende 1982.  
 Best.-Nr. 117 19,80 DM



**Programmieren in Maschinensprache mit 6502, E. Flögel, W. Hofacker**  
 Das deutschsprachige Werk über 6502 Maschinenprogrammierung. Einführung, Grundlagen, Eigenschaften, Adressierungsarten, Befehlsarten. Wie entwickelt man ein 6502 Maschinenprogramm? Handassemblierung, viele Programmbeispiele mit genauen Angaben direkt zum Eingeben in den Apple II mit Adressangabe (keine blutleeren Beispiele ohne Adresse), Verwendung von Assemblern. AIM-Assembler, Disassembler, Relocator, 6522 VIA, 6520, Interrupt, Fehlersuche in Maschinenprogrammen, Maschinensprache, Programmiertricks. Spezielle Abschnitte für Maschinensprachenprogrammierung über PET, CBM 3000, CBM 4000 u. VC-20, ATARI 400/800, Apple II, AIM sowie Ohio Scientific Challenger. Dieses Buch sollte jeder 6502 Systemanwender besitzen. Ca. 240 Seiten.

Best.-Nr. 118 49,00 DM

**Anwenderprogramme für TRS-80 von Martin Stübs**

Ein Buch, voll mit interessanten Anwenderprogrammen für TRS-80 Level II 16K und Video Genie (teilweise Diskette u./od. Cassette). Hauptsächlich Programme für den Manager, Geschäftsmann, Klein- und Mittelbetrieb. Auch einige interessante Spiele sind enthalten. Terminkalender, Reservierungsprogramm für Omnibusunternehmen und Hotels, Textverarbeitung, usw.

Best.-Nr. 120 29,80 DM



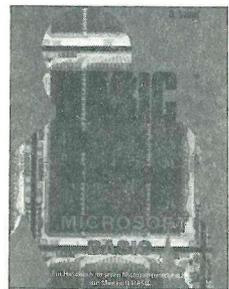
**BASIC für Fortgeschrittene**  
 Endlich ein BASIC-Buch für den fortgeschrittenen Programmierer. Alle wichtigen Befehle aus der Stringmanipulation, Disk-Befehle, WAIT, INSTR, WHILE WHEND usw. werden an Beispielen besprochen. Alle Befehle sind übersichtlich wie in einem Nachschlagewerk mit großen Überschriften angeordnet. Dann folgt ein umfangreicher BASIC Kurs für Fortgeschrittene mit vielen Beispielen (Inventur, Rechnungen schreiben, Adressverwaltung usw.). Am Schluß finden Sie dann noch einen Vergleich der wichtigsten Sortiermethoden sowie ein Programm zur Vorhersage von Ereignissen.

Best.-Nr. 122 39,00 DM



**Programmieren in Maschinensprache (Z80), C. Lorenz**  
 Eine sehr ausführliche Einführung in die Z80 Maschinensprache mit vielen Beispielen. Die Beispiele können mit Hilfe des TRS-80 Level II sowie dem T-BUG von TANDY und den T-BUG-Erweiterungen (IN LOCO, T-STEP, T-LEGS) ausgeführt werden. Ein unentbehrliches Buch für jeden, dem die BASIC-Programmiersprache von der Geschwindigkeit her zur Lösung seiner Aufgaben nicht mehr ausreicht.

Best.-Nr. 119 49,00 DM



**Microsoft BASIC-Handbuch**  
 Die deutsche Übersetzung des erfolgreichen Microsoft BASIC-Handbooks. Leicht verständliche Einführung mit vielen interessanten Programmbeispielen. Das kompetente Werk von Microsoft selbst. Ideal als Zusatzliteratur zu jedem BASIC-Buch.

Best.-Nr. 121 29,80 DM



**IEC Bus-Handbuch, M. P. Gottlob**  
 Ein Handbuch und Nachschlagewerk für alle Besitzer von Computern mit IEC (IEEE 488 Bus). Dazu gehört auch der PET sowie alle CBM-Computer. Grundlagen, das BUS-System, Meßdatenübertragung, Adressierung eines Instruments, kl. IEC-BUS-Lexikon u.v.a.  
**Best.-Nr. 123 19,80 DM**



**Programmieren in Maschinensprache mit CBM**

An Hand eines praktischen Beispiels (Sortierroutine) wird der Unterschied zwischen BASIC und Maschinensprogrammen gezeigt. Das Maschinenprogramm kann mit dem leistungsfähigen MONJANA/1 Monitor in ROM erstellt werden. Am Schluß finden Sie weitere wichtige Informationen wie Dez/Hex-Umrechnungstabelle, Befehlslisten, ASCII-Tabelle sowie eine ROM-Vergleichsliste zwischen 8k PET und den neuen CBM-Maschinen.  
**Best.-Nr. 124 19,80 DM**  
**MONJANA Monitor im ROM**  
**Best.-Nr. 1241 79,00 DM**

ohne Abbildung

**ELCOMP, Fachzeitschrift für Microcomputertechnik**  
 Die kompetente Fachzeitschrift für das moderne Gebiet der Microcomputertechnik. Erscheint 10 x pro Jahr. Jahrespreis \* **59,00 DM** incl. MwSt., Porto und Verpackung. Wer die neuesten Informationen aus diesem Gebiet für sich nutzen möchte, muß ELCOMP lesen. Software, Technische Tips, Programmiertricks, Bauanleitungen, Systembeschreibungen, u. v. a. Jeden Monat brandneu. ISSN-Nr. 0171-0958  
**Best.-Nr. 125 je Heft \* 4,50 DM**

**ELCOMP-Doppelheft**

Von den 10 Heften erscheint im Juli/August und November/Dezember ein Doppelheft.  
**Best.-Nr. 126 \* 9,00 DM**

\* ab 1.1.1982 gelten die neuen Preise:  
 Abbonement 69,00 DM, Einzelheft 5,00 DM  
 Doppelheft 10,00 DM



**Einführung in die Microcomputer Programmierung mit 6800**

Eine sehr gute Einführung in die Microcomputertechnik mit Hilfe des Mikroprozessors 6800. Ausführliche Erklärungen mit vielen Beispielen und Anleitungen. Theoretische Grundlagen. CPU-Architektur, Befehlssatz, Systemaufbau, Hilfsmittel der Programmierung, Trainingsprogramme, Systemkomponenten, FIRMWARE. Ein komplettes Monitorprogramm (Betriebssystem) ist als Listing enthalten. Über 250 Seiten.  
**Best.-Nr. 127 49,00 DM**

ohne Abbildung

**Programmieren mit dem CBM**

Ein Hand- und Programmierbuch für alle CBM-Besitzer der 3000, 4000 sowie der 8000er Serie. Viele Tricks und Programmierbeispiele, Anleitungen. Erscheint Ende 1981.  
**Best.-Nr. 128 29,80 DM**

ohne Abbildung

**ELCOMP Leser Programmierhandb.**  
 Hier fassen wir die besten Programme unserer ELCOMP-Leser zusammen. Programme für PET, CBM, TRS-80, AIM, Superboard, C4P, Exidy, Sharp, MZ80K, Apple II, Nascom I und II und TI 99/4 werden als Listing mit kurzer Beschreibung allen Lesern zugänglich gemacht. Erscheint Anfang 1982.  
**Best.-Nr. 129 69,00 DM**



**Programmierbeispiele für CBM**

Ein Buch mit vielen BASIC-Programmen für CBM und PET. Spiele, Geschäftsbereich, Erziehung und Wissenschaft, Utilities, Hilfen für Maschinensprachenprogrammierung, trickreiche Programme. Viele Programme für wenig Geld.  
**Best.-Nr. 130 19,80 DM**

ohne Abbildung

### Cobol für Anfänger

Eine Einführung in die Cobol-Programmierung für den Micro-computer-Besitzer. Erscheint ca. Mitte 1982.

Best.-Nr. 131 19,80 DM

ohne Abbildung

### CP/M Handbuch

Grundlagen, Einführung, Hilfs- und Handbuch für jeden der mit dem "Software-Bus" arbeiten möchte. Ideal auch für Anfänger. Praktisches Handbuch für den Profi. Erscheint ca. Ende 1981.

Best.-Nr. 132 19,80 DM

ohne Abbildung

### Welches Betriebssystem brauche ich?

Ein Leitfaden und Handbuch für den Anwender der gehobenen Micro-computerklasse. UNIX, CP/M, OASIS, NORTHSTAR DOS, USCD, Apple DOS, ATARI DOS, CP/A, usw. Erscheint Mitte 1982.

Best.-Nr. 133 19,80 DM

ohne Abbildung

### Microcomputer im Unterrichtsfeld M. Penzkofer

Wie werden Microcomputer im Schulbetrieb eingesetzt? Grundlagen Wissenswertes, Tips, Erfahrungen und viele Beispiele. Für alle die die moderne Microelektronik für Schule Ausbildung und Erziehung nutzen wollen. Erscheint ca. Anfang 1982.

Best.-Nr. 136 29,80 DM

ohne Abbildung

### FORTH Handbuch und Einführung von E. Flögel, W. Hofacker

FORTH ist nicht nur eine sehr leistungsfähige Programmiersprache — es ist schon fast eine "Religion". FORTH eignet sich bestens für industrielle Steuerungen, Grafik etc. Grundlagen und viele Programmbeispiele (Apple II, Ohio, ATARI usw.). Erscheint Mitte 1982.

Best.-Nr. 137 39,00 DM



### Redysoft News

Das Magazin für den Microcomputer-einsatz im kommerziellen Bereich. Redysoft-News erscheint in zwangsloser Folge. Bisher sind drei Hefte erschienen. Aus dem Inhalt:

**Heft 1:** Microcomputer für freiberuflich Tätige. Die Anwendung von Microcomputern im Geschäftsbereich. Microcomputer-Markt. Ein Microcomputer zum Arbeiten. Die Microschule. MICROS werden voll geschäftsfähig. Challenger Superboard, der ideale Einstieg.

**Heft 2:** ACCEL - ein Compiler f. TRS-80 Level II BASIC. Maschinenprogramme von Southern Software für TRS-80 Level II

Laden von Maschinenprogrammen ACCEL, DLOAD, RENUM, HDUMP, FGRAF, XREF, ZBUG, SDUMP, TSAVE, USRN, SRCH, LIFE1 + LIFE2. Auch Siemens auf VLSI-Kurs. Dateiverwaltung. Struktur von FIBU-80.

**Heft 3:** APPLE Utility Disk. Bilanzbuchhaltung u. Finanzbuchhaltung f. den Microcomputer TRS-80 Model II. ACCEL u. ACCEL2 ? - BASIC-Compiler für den TRS-80 Model I. Eine Spielcassette für den TRS-80. Adressenverwaltung. Disassembler. TAPECHECK. SAVE. Terminkalender. Lagerverwaltung. ADEKART. Computerspiele für höchste Ansprüche.

Best.-Nr. 135 je Heft 2,00 DM

ohne Abbildung

### ADA - Handbuch, E. Flögel

Vorankündigung für Mitte 1982. Mit einer praktischen und leicht verständlichen Einführung soll hier dem Leser diese neue Programmiersprache nähergebracht werden. APA wurde, wie Cobol vom US-Verteidigungsministerium entwickelt und soll die steigenden Kosten bei der Softwareentwicklung eindämmen. ADA ähnelt in gewisser Weise d. Programmiersprache PASCAL und dürfte bald zum neuen Standard werden. Zur Zeit (Mitte 1981) gibt es jedoch noch keinen Microcomputer auf denen ADA implementiert ist.

Best.-Nr. 138 29,80 DM

ohne Abbildung

### BASIC für blutige Laien von W. Hofacker

Endlich ein Buch für den Anfänger und Laien. Ziel des Buches ist es, dem "blutigen Laien" die Grundlage der Programmiersprache BASIC zu vermitteln. Lernen wird nun zum echten Vergnügen und Freizeitspaß. Auch der Preis macht Spaß. Erscheint ca. Ende 1981.

Best.-Nr. 139 nur 19,80 DM

# ELCOMP-Bücher

# Englisch



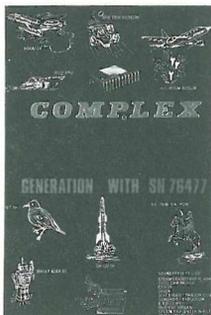
## Care and Feeding of the Commodore PET

Das ideale Buch für den Hardware-Bastler. Viele Tricks, Schaltbilder, Hinweise und Erläuterungen für den, der gerne selbst Erweiterungen bauen möchte. Memory Map für 8k PET und CBM, Bauanleitung für eine serielle Schnittstelle u. v. a.  
Best.-Nr. 150 19,80 DM



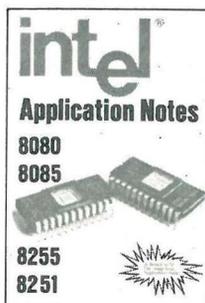
## Microsoft 8k BASIC Reference Manual

Eine sehr gute BASIC-Einführung. Auch als Handbuch zum Nachschlagen bestens geeignet. Ideal für jeden PET, CBM, TRS-80, KIM-BASIC, SYM-BASIC, AIM- und APPLE-Besitzer. 73 Seiten DIN A4 mit vielen Beispielen.  
Best.-Nr. 151 19,80 DM



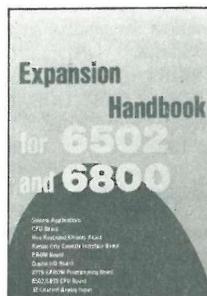
## Complex Sound Generation with SN 76477

Ein Applikationsheft für einen der interessantesten integrierten Bausteine unserer Zeit. Ein LSI-Baustein zur Tonerzeugung. Je nach äußerer Beschaltung können Sie mit diesem Baustein die verrücktesten Töne erzeugen. Dampfisenbahngeräusch mit Dampfpfeife, Vogelgezwitscher, Hundegebell, elektronische Orgel, Schuß mit Explosion u. v. a. mehr.  
Best.-Nr. 154 19,80 DM



## Expansion Handbook for 6502 and 6800

Das ideale Handbuch für alle KIM, SYM, AIM, PET und Challenger Computer-Freunde. Das Buch beschäftigt sich ausschließlich mit dem S-44-Bus. Dies ist exakt der Bus von SYM, AIM und KIM. Sehr viele Schaltbilder: CPU-Platine, Hex-Tastatur Eingabe, Kansas City Interface, RAM u. ROM-Karte, Analog-Eingabe Board u. v. a. Das Buch ist für jeden 6502 Systembesitzer unentbehrlich. Ca. 150 Seiten.  
Best.-Nr. 152 19,80 DM

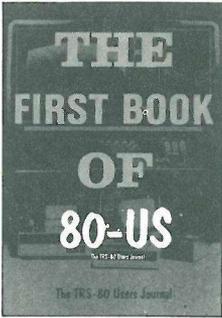


## Intel Application Notes (8080, 8085, 8255, 8251)

Dieses Buch braucht jeder, der mit 8080, 8085 oder Z-80 Mikroprozessoren arbeitet.

Wir haben die interessantesten Applikationsberichte in diesem Buch zusammengefasst. Aus dem Inhalt: Designing with Intel's Static RAM's 2102, Memory Design with the Intel 2107B. 8255 Programmable Peripheral Interface Applications, Using the 8202 Dynamic RAM Controller u. v. a.

Best.-Nr. 153 29,80 DM

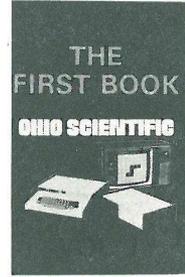


**The First Book of 80-US**

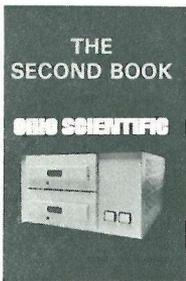
Für den TRS-80 Freund eine echte Preissensation. Die ersten fünf Hefte aus 80-US Journals in einem Sammelband zusammengefaßt. Voll mit vielen sehr interessanten Hard- und Softwareideen, Tricks. Viele komplette Programmbeispiele (Listings) in BASIC u. Z-80 Maschinensprache. Über 250 Seiten DIN A4. Farbiger Umschlag. Dieses Buch sollte jeder TRS-80 Besitzer oder der es werden will im Schrank haben.  
**Best.-Nr. 155 29,80 DM**



**Small Business Programs, S. Roberts**  
 Ein Buch für denjenigen, der die modernen Microcomputer (speziell TRS-80, Apple, PET, North Star, Challenger) zur Rationalisierung in seinem Klein- oder Mittelbetrieb einsetzen möchte. Viele nützliche Tips, Hinweise und Programmierbeispiele. Dieses Buch sollte jeder Geschäftsmann u. Microcomputerfreund besitzen.  
**Best.-Nr. 156 29,80 DM**



**The first Book of Ohio Scientific**  
 Das erste weltweit produzierte Buch für die erfolgreiche Ohio Scientific Challenger Computerserie. Grundlagen, viele Programmiertricks, Hardwaretips, Umbauanleitungen, Programmierbeispiele u. v. a. Glanzumschlag, 186 Seiten.  
**Best.-Nr. 157 19,80 DM**



**The second Book of Ohio Scientific**  
 Eingehende Beschreibungen über praktische und geschäftsorientierte Software. Speicher Test Programm, Tricks und Tips für Disketten-Anwender. Mini-Floppy-Expansion u. v. a. 159 Seiten.  
**Best.-Nr. 158 19,80 DM**

ohne Abbildung  
 Erscheint ca. Ende 1981

**The third book of Ohio**  
 Wie erweitere ich mein Challenger System ? Universelle I/O-Karte, EPROM-Burner für 2716, EPROM, RAM-Karte, 6522 VIA-Karte. Wo notwendig mit kompletter Software. Dieses Buch braucht jeder Ohio-Benutzer. Komplette Schaltbilder und Aufbauhinweise.  
**Best.-Nr. 159 19,80 DM**



**The fourth Book of Ohio Scientific**  
 Ein Buch voll mit Programmen für das Superboard, C4P, C4PMF und C28P. Die Softwarequelle für jeden Challenger-Fan. Alle Programme sind getestet und auch auf Cassette verfügbar. 170 Seiten Listings und Beschreibungen.  
**Best.-Nr. 160 29,80 DM**  
**Best.-Nr. 8324 Cassette 29,80 DM**

ohne Abbildung

The fifth Book of Ohio Scientific Fortsetzung der beliebten Reihe über die Ohio Scientific Computer. Hier liegt der Schwerpunkt wieder bei der Software. Viel interessante Software. Eine komplette Adressenverwaltung mit Label und Listenausgabe. Mehrere Fakturierungsprogramme, viele interessante Spiele wie Autorennen, Tankwar, Tonerzeugung, Joystickprogrammierung usw. Erscheint ca. Anfang 1982. Best.-Nr. 161 19,80 DM

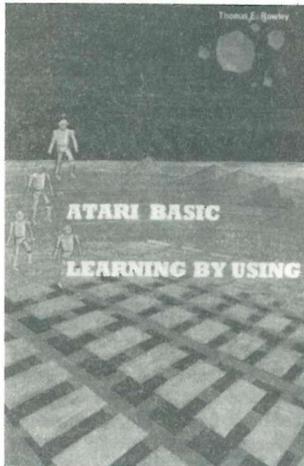
ohne Abbildung

25 Programs for the ATARI 25 sehr interessante Programme, Beschreibungen, Tips und Tricks für die ATARI 400 und 800 Computer. Wesentliche Unterscheidungsmerkmale zwischen ATARI - BASIC, MICROSOFT - BASIC und BASIC unter CP/A werden erläutert. Verwendung des ATARI - Editor / Assemblers zur Erstellung einfacher Maschinenprogramme. Erscheint ca. Anfang 1982. Best.-Nr. 162 19,80 DM

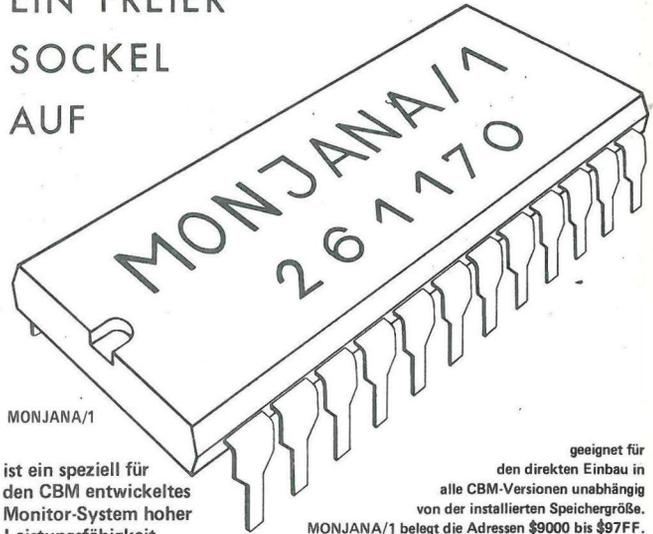
ohne Abbildung

The peripheral Handbook Ein Handbuch über Drucker, Floppys, Hard-Disks, und alles was Sie um den Microcomputer herum benötigen. Triks und Kniffe. Technische Beschreibung. Erscheint ca. Mitte 1982. Best.-Nr. 163 29,80 DM  
ohne Abbildung  
6502 Primer  
Erscheint ca. Anfang 1982. Best.-Nr. 165 29,80 DM

ATARI-BASIC - Learning by Using von Tom Rowley  
Ein sehr interessantes Lehr- und Programmierbuch für jeden ATARI-Besitzer. Kurze Programme und Lernbeispiele machen das Programmieren zum Kinderspiel. Viele Programme in diesem Buch verwenden weiterführende raffinierte Programmiertechniken andere wiederum sind einfach und ideal für den Erstkontakt mit dem ATARI-Mikrocomputer. Tonerzeugung, Grafik, Joystickprogrammierung, Player Missile Graphic, Video-Kunst u. v. a. Ca. 75 Seiten. Best.-Nr. 164 19,80 DM



# IN IHREM CBM WARTET EIN FREIER SOCKEL AUF



MONJANA/1

ist ein speziell für den CBM entwickeltes Monitor-System hoher Leistungsfähigkeit

geeignet für den direkten Einbau in alle CBM-Versionen unabhängig von der installierten Speichergröße. MONJANA/1 belegt die Adressen \$9000 bis \$97FF.

Ing. W. Hofacker GmbH, Tegernseerstr. 18, 8150 Holzkirchen



## BESTELLUNG

.... Stück Monitor-System MONJANA/1 (2K-Byte-EPROM, incl. ausführlicher Systembeschreibung)  
zum Stückpreis von DM 79,- + Porto  
Lieferung per Nachnahme

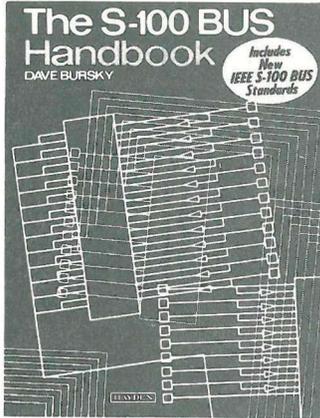
Monjana-1  
Der Maschinensprachen-Monitor, der auch Ihnen gute Dienste leisten wird. Ideal für jeden, der die Leistungsfähigkeit seines CBMs voll ausschöpfen will.

Heute noch bestellen !

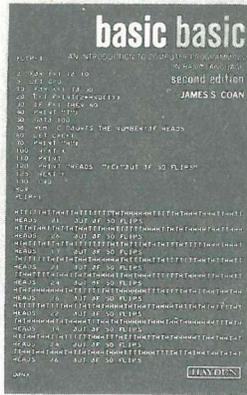
Name/Vorname \_\_\_\_\_  
Straße / Hs-Nr. \_\_\_\_\_  
PLZ \_\_\_\_\_ Ort \_\_\_\_\_  
Datum..... Unterschrift.....



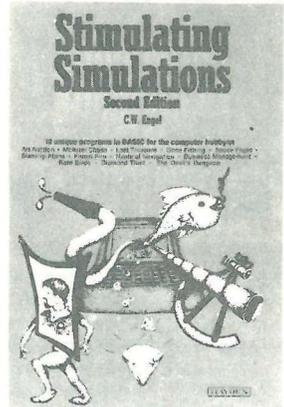
# Hayden-Bücher



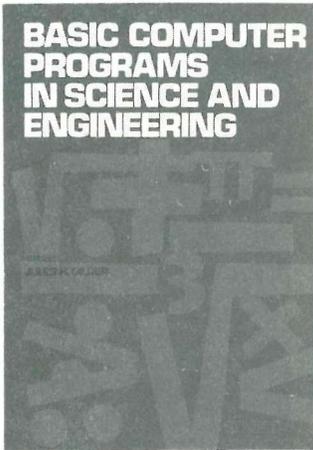
**The S-100 Bus Handbook**  
(enthält den Neuen S-100 Standard). Alles was Sie über den S-100 Bus wissen sollten. Einführung, Schaltungen, Schaltbilder der vier verwendeten S-100 Platinen. Service u. Fehlersuche in S-100-Systemen, Interface-beispiele u. v. a. 256 Seiten Großformat. Best.-Nr.: 254 49,00 DM



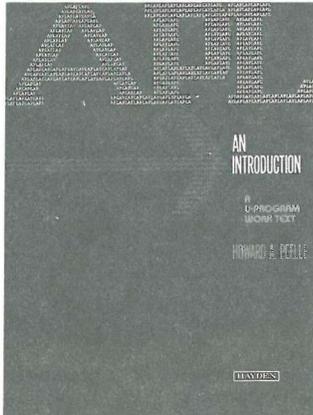
**BASIC BASIC**  
Eine Einf. in die BASIC-Programmierung v. Microcomp. Eines der besten BASIC-Bücher weltweit. Viele Beispiele aus dem professionellen u. wissenschaftl. Bereich. Best.-Nr.: 255 39,00 DM



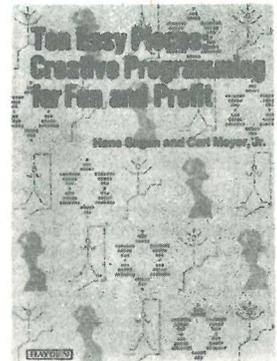
**Simulating Simulations**, v. C. W. Engel  
12 einzigartige Programme in BASIC f. d. Comp.-Hobbyisten. (Kunstauktion, Monster Chase, Lost Treasure, Gone Fishing, Space Flight, Starship, Forest Fire, Navigation, Business Management, usw. Best.-Nr.: 256 19,80 DM



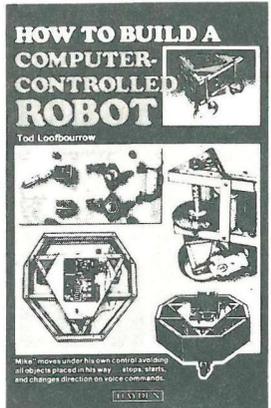
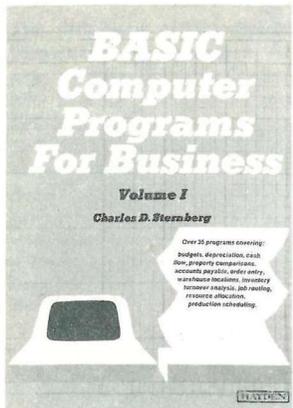
**BASIC Computer Programs in Science and Engineering.** (250 Seiten Großformat) Programme aus den verschiedensten Bereichen m. genauen Beschreibungen. Komplexe Zahlen, Engineering, Mathematik, Matrizen, Analyse von Daten, Elektrotechnik, Electronic, Filterberechnung usw. Best.-Nr.: 257 39,00 DM



**APL - An Introduction**  
Ein Programmier- und Arbeitsbuch. Ein Einführungsbuch zum Selbststudium von APL. Viele Beisp., Übungsbeisp. m. Lösungen. 245 Seiten Großformat. Best.-Nr. 258 39,00 DM



**Ten Easy Pucos:**  
Creative Programming for Firm and Profi  
10 Beispielhaft beschriebene BASIC-Programme. Schritt f. Schritt wird jeder Befehl u. Zusammenhang erläutert. Schwerpunkt: Simulationsprogramme. Sehr gut geeignet f. Ausbildung, Selbststudium und als Nachschlagewerk. Best.-Nr. 259 29,80 DM

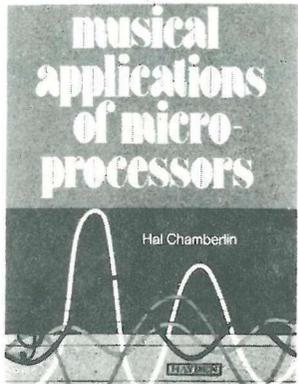
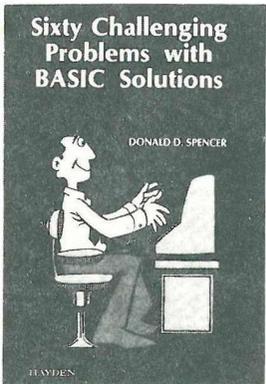


**BASIC Computer Programs für Business, Vol. I** Charles D. Sternberg  
 Über 35 Programme aus dem Gebiet : Budget, Abschreibung, Cash-Flow, Grundstückswertvergleich, Accounts Payable, Auftragsingang, Inventur-Umschlagsanalyse, Produkt-Terminplanung u. v. a. Alle Programme sind ausführlich beschrieben. Ein sehr wertvolles Buch. 250 Seiten, Großformat  
 Best.-Nr. 260 39,-

**How to build a Computer controlled Robot, v. Tod Loofbourrow**  
 Komplettre Bauanleitung für einen mit KIM-1 gesteuerten Roboter. Viele Schaltungen A/D- D/A-Wandler mit Software. (Ultraschall-Detektion, Spracherkennung u.v.a)  
 Best.-Nr.: 253 35,00 DM

**Home Computers Can Make Your Rich**  
 Ein Experte zeigt Ihnen wie man mit Personal-Computer Geld verdienen kann. Was ist zu beachten? Viele Ideen.  
 Best.-Nr. 262 19,80 DM

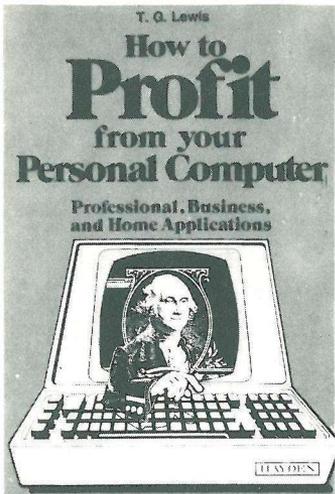
ohne Abbildung  
**BASIC Computer Programs For Business**  
 Best.-Nr. 261 39,00 DM  
**The complete 1802 Cookbook**  
 Best.-Nr. 264 19,80 DM



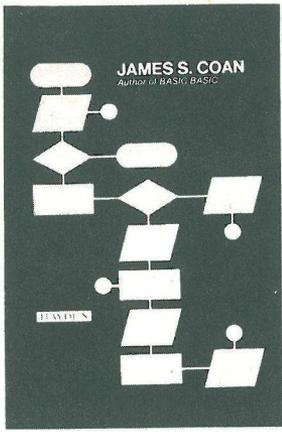
**Sixty Challenging Problems with BASIC Solutions.** 60 Spiele, Puzzles, mathematische und wissenschaftliche Problembeschreibungen, die der Programmierer in BASIC lösen soll. Am Ende des Buches sind zur Kontrolle alle Lösungen (60 BASIC Programme) aufgeführt.  
 Best.-Nr.: 263 19,80 DM

**Musical Applications of Microprocessors** von Hal Chamberlin.  
 Hal Chamberlin kann als Papst für Computermusik bezeichnet werden. In diesem 660seitigen Buch hat er sein Wissen und seine Erfahrungen der Öffentlichkeit zugänglich gemacht. Grundlagen, Einführung, Schaltungsbeisp., Programmbeisp., digitale Tonerzeugung, digitale Filter u. v. a.  
 Best.-Nr.: 265 79,00 DM

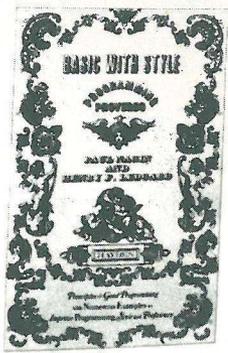
**Advanced BASIC, Applications a. Problems**  
 Kann als Fortsetzung von Nr. 255 angesehen werden. Tiefgehende Informationen über Files, Strings, Geometrie, Reihen u. Folgen, Matrizen, Statistik u. Simulationen.  
 Best.-Nr. 266 39,00 DM



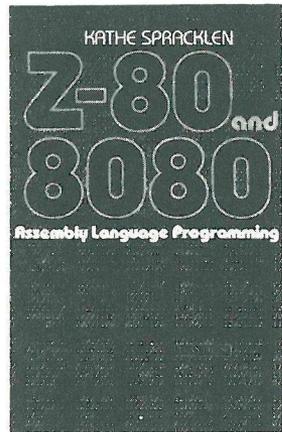
**How to Profit from your Personal Computer**  
 Dieses Buch ist für den Geschäftsmann und Computer-Hobbyisten geschrieben. Es zeigt Ihnen wie Sie den Computer für sich arbeiten lassen können und dabei noch Geld verdienen. (Viele Tips und Programmbeispiele in BASIC.)  
 Best.-Nr. 267 39,00 DM



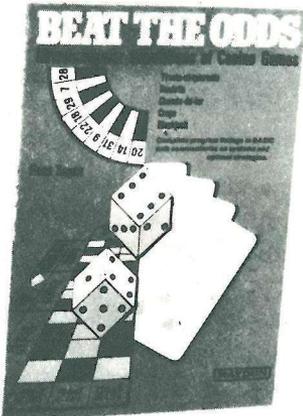
**BASIC FORTRAN**  
 Vom Autor des Buches "BASIC BASIC". Eine ausgezeichnete Einführung in die Programmiersprache FORTRAN. Schritt für Schritt mit vielen Beispielen (über 80) lernen Sie diese leistungsfähige Sprache.  
 Best.-Nr. 271 (engl.) 45,00 DM



**BASIC with Style**  
**COBOL with Style**  
**PASCAL with Style**  
 Diese Buchserie ist eine Art "Hohe-Schule" für den ernsthaften Programmierer. Viele Tips Beispiele, nützliche Unter- und Hilfsprogramme für die Programmerstellung am Arbeitsplatz oder in der Freizeit. Top-Down Programing, Program Standard Odds and Ends sind die Hauptthemen in diesen drei sehr wertvollen Büchern. Jedes Buch ca. 145 - 200 Seiten.  
**PASCAL with Style**  
 Best.-Nr. 268 39,00 DM



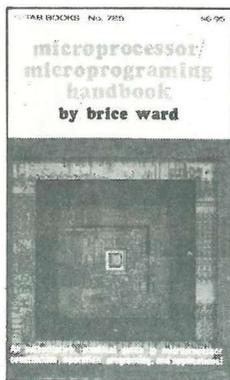
**Z-80 and 8080 Assembly Language Programming**  
 Ein Einführungsbuch in die Maschinensprache für 8080 und Z-80 Systeme. Es bringt Ihnen fast alles, was Sie zur optimalen Nutzung Ihres Computers brauchen. Ideal für Anfänger.  
 Best.-Nr. 272 (engl.) 39,00 DM



**Beat the ODDS**  
 Microcomputer Simulationen von Casino-Spielen, Trente-et-quarante, Roulette, Blackjack, Craps, Chemin-de-fer. Eine extrem nützliche Programmierhilfe für realistische Simulationen der wichtigsten Casino Spiele.  
 Best. Nr. 273 39,00 DM

**COBOL with Style**  
 Best.-Nr. 269 39,00 DM  
**BASIC with Style**  
 Best.-Nr. 270 39,00 DM  
**Endlich ein Buch über den 1802 Microprozessor**  
 Den leistungsfähigen 1802 Microcomputer in CMOS Technologie, der heute vorwiegend in der Industrie und KFZ-Technik eingesetzt wird und in riesigen Stückzahlen produziert wird, wurde bis jetzt wenig allgemeines Interesse entgegengebracht. Ganz zu unrecht! Den dieser Prozessor ist für das oben genannte Einsatzgebiet hervorragend geeignet. Das Buch eignet sich für den Einsteiger genau so wie für den Experten. Einführung, Grundl. der Assembler Programmierung, Befehlssatz mit Erläuterungen. Ein kompletter 1802 Assembler als Listing, Übungsbeispiele m. Antworten.  
 Best.-Nr. 264 19,80 DM

# TAB-Books



## Microprocessor/Microprogramming Handbook, Brice Ward

Ein praktisches Handbuch für jeden, der sich mit Mikroprozessoren beschäftigen möchte. Auf über 290 Seiten finden Sie Grundlagen, alles über Programmierung und Anwendungsbeispiele der interessantesten integrierten Schaltungen. Inhalt: Einführung in die Mikroprozessortechnik. Der Mikroprozessor von innen. MCS4, MCS40, MCS80, Speichersysteme, Mikroprogrammierung in Maschinen- und Assemblersprache.

Best.-Nr. 785

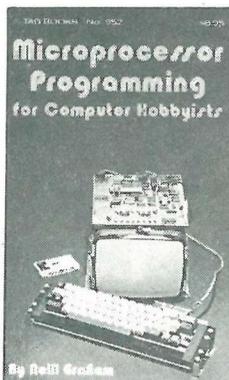
DM 35,-

## Microprocessor Programming for the Computer Hobbyist

Dieses Buch ist speziell für den Computer-Hobbyisten geschrieben, der sich bereits mit weiterführenden Programmier-techniken und Datenstrukturen beschäftigen möchte. Inhalt: Höhere Programmiersprachen (PL/M, PL/1), arithmetische Funktionen, Suchprogramme werden im Zusammenhang mit dem Schachspielproblem behandelt. Über 380 Seiten in englischer Sprache

Best.-Nr. 952

DM 39,-



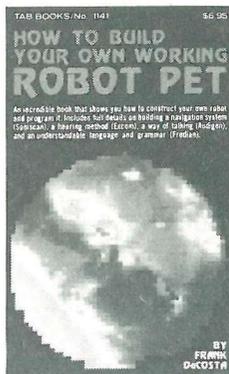
## How to build your own working ROBOT PET

An incredible book that shows you how to construct your own robot and program it. Includes full details on building a navigation system (Soniscan), a hearing method (Excom), a way of talking (Audigen) and an understandable language and grammar (Fredian).

Eine wertvolle Hilfe für alle, die sich einen Roboter bauen wollen. (8085) CPU. Viele wertvolle Schaltungen, die Sie im Zusammenhang mit TRS-80 oder 8085-Computern verwenden können. 238 Seiten.

Best.-Nr. 1141

DM 29,80

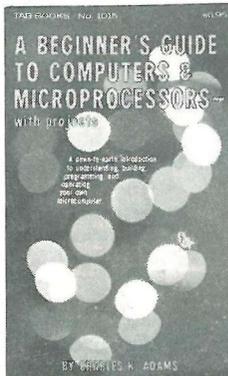


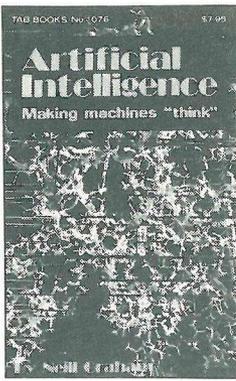
## Beginner's Guide to Computer Programming, Brice Ward

Eine ideale Einführung in die Programmierung von Computern (Mikroprozessoren). Das Buch beginnt mit der Entwicklung einer einfachen Programmiersprache für den eigenen Bereich und zum Selbststudium, sodann wird auf andere Sprachen übergegangen. Inhalt: Grundlegende Programmkonzepte, I/O-Schaltungen, Flußdiagramme, Programmtest, Schleifen, Indexregister, versch. Programmiersprachen, Compiler, Cobol u. v. a., 480 Seiten, 364 Bilder, in englischer Sprache.

Best.-Nr. 1015

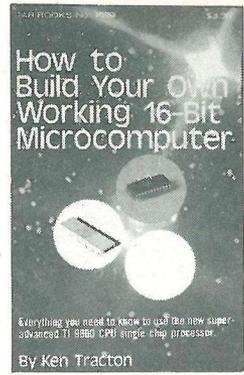
DM 29,80



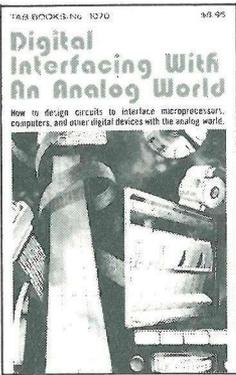


**Artificial Intelligence, Neil Graham**  
 Hier ist endlich ein Buch speziell über künstliche Intelligenz. Der Autor benutzt Computerspiele und Robotertechnik für die illustrative Behandlung dieses brandheißen Themas. Was ist der Unterschied zwischen einer Lösung einer komplexen Programmieraufgabe und der Programmerstellung für intelligente Computerentscheidungen.  
**Best.-Nr. 1076 DM 29,80**

**Microprocessor Cookbook, von Michael Hordeski**  
 A chip-by-chip-comparison of today's modern microprocessors.  
 Die wichtigsten Prozessortypen werden genau beschrieben - (Schaltung, Blockdiagramm, Befehlslisten, Programmbeispiele) und verglichen: 8080, 6800, F8, Z80, TMS 9900, SC/MP, Bit Slices, R6500, 264 S.  
**Best.-Nr. 1053 DM 24,80**

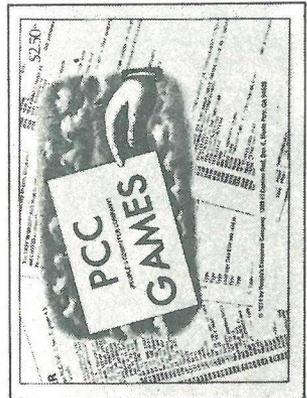


**How to Build your Own Working 16-Bit-Microcomputer von Ken Tracton**  
 Alles, was Sie über den neuen Supermicroprozessor TI9900 wissen müssen. Beschreibung des Schaltkreises, Peripherie-Bausteine für TMS9900, Tricks, Systemaufbau etc. Sehr interessant, da der TMS9900 das Herz des neuen Texas Instruments Personal Computers ist.  
**Best.-Nr. 1099 DM 14,80**

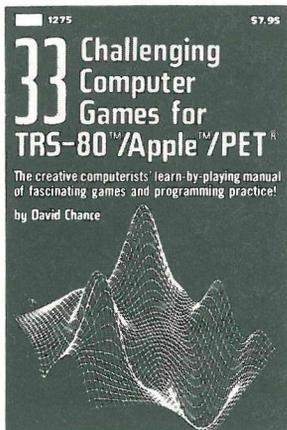


**Digital Interfacing with an Analog World, Joseph J. Carr**  
 Wie entwirft und baut man Interfaceschaltungen für Microcomputer? Der Schwerpunkt liegt bei der Verbindung mit der analogen Welt. Viele Tabellen, Schaltungen, Berechnungen und Hinweise. Über 400 Seiten.  
**Best.-Nr. 1070 DM 39,-**

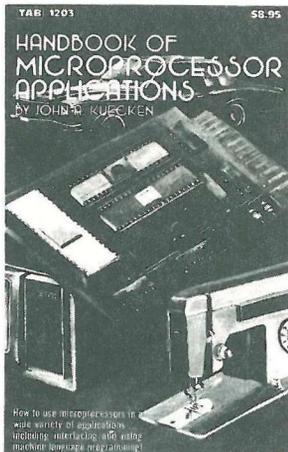
**The BASIC Cookbook, von Ken Tracton**  
 Ein komplettes Dictionary mit BASIC-Befehlen. Alle Befehle sind in alphabetischer Reihenfolge geordnet und jeder Befehl genau beschrieben. Zu jedem Befehl ist ein Demonstrationsbeispiel beigelegt.  
 Dieses Buch braucht jeder BASIC-Programmierer!  
**Best.-Nr. 1055 DM 24,80**



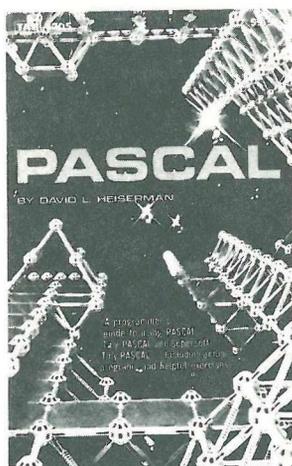
**25 Games in BASIC (Listings). Die Programme laufen mit kleinen Änderungen auf den meisten Microcomputern. Inhalt: Number, Letter, Stars, Trap, Bagels, Mugwump, Hurkle, Snark, Reverse, Button Chomp, Taxman u. v. a.**  
**Best.-Nr. 8057 DM 9,80**



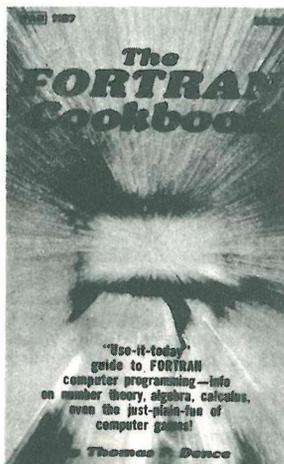
**33 Challenging Computer Games for TRS-80, APPLE, PET**  
Das Handbuch für den Computerist, der durch Spielen lernen will. Über faszinierende Spiele zur Programmierpraxis. Ein Buch f. jeden Personal Computer-Besitzer. Best.-Nr.: 1275 **29,80 DM**



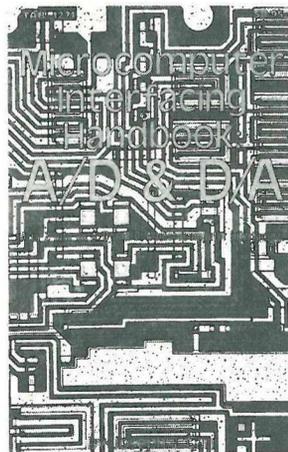
**Handbook of Microprocessor Applications (6800, 6802)**. Wie werden Microcomputer in der Praxis eingesetzt. Interfacetchniken u. Anwendung v. Maschinsprache, Schaltungsbeispiele mit Software. Beschäftigt sich in erster Linie mit 6800, 6802 und den zugehörigen Peripheriebausteinen. Best.-Nr.: 1203 **29,80 DM**



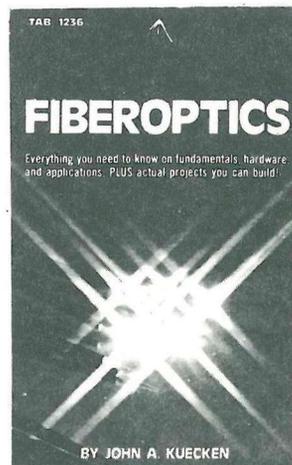
**PASCAL**  
Eine Programmieranleitung f. d. PASCAL-Freund. Wie verwendet man TINY PASCAL und Supersoft-PASCAL. Passt ideal zu diesen Paketen und TRS-80. Viele Programmierbeispiele. (350 Seiten) Best.-Nr. 1205 **35,00 DM**



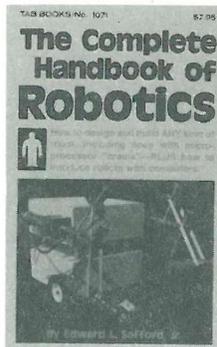
**The FORTRAN Cookbook**  
Heute noch verwenden! eine Einführung in die FORTRAN-Programmierspr., Grundlagen, Beispiele, Übungen und Beispiele aus dem Microcomputerbereich. (Mathem. Probleme bei Computerspielen u. v. a.) Best.-Nr.: 1187 **29,80 DM**



**Microcomputer Interfacing Handbook: A/D- und D/A-Wandler**  
Ein Hand- u. Nachschlagebuch f. jeden der Analog/Digital od. Digital/Analog-Wandler an seinen Microcomputer anschließen will. Unentbehrlich f. die Anwendung des Microcomputers im industriellen Einsatz. Best.-Nr.: 1271 **35,00 DM**



**FIBEROPTICS**  
Alles was Sie über d. Grundlagen, Hardware und Applikationen dieser faszinierenden neuen Technik wissen sollten. Mit Projekten, die Sie selbst bauen können. (Übertragungsraten bis zu 1.354 x 10<sup>9</sup> Bildpunkte pro Sekunde, das bringt die Fiberoptic!) Best.-Nr.: 1236 **29,80 DM**



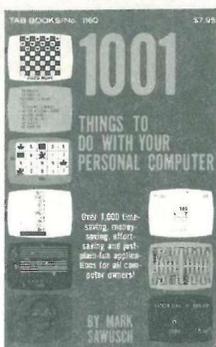
**The Complete Handbook of Robotics**

E. Safford, jr.

Wie entwickle und baue ich einen Roboter. Es werden auch microcomputer-gesteuerte Roboter beschrieben, und wie man Roboter mit Computern zusammenschaltet. Terminologie, Entwicklungsstand, Sensoren, Grundlagen, das Roboter "Gehirn", Servomechanismus, Tips, Interfacing u. v. a.

Best.-Nr. 1071

DM 29,80

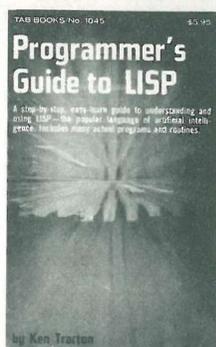


**1001 Things to do with your Personal Computer**, Mark Sawusch

Computeranwendungen für jedermann, Geschäft und Finanzprogramme, mathematische Applikationen, technisch/wissenschaftliche Anwendung von Personal Computern, Anwendungen in Erziehung und Ausbildung, Hobby Computer, Spiele, Control- und periphere Applikationen, künstliche Intelligenz. Viele Programme in BASIC, viele Tips. 300 Seiten

Best.-Nr. 1160

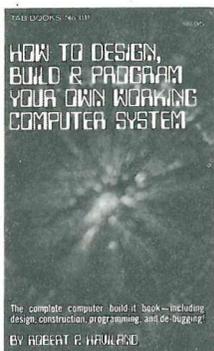
DM 29,80



**Programmer's Guide to LISP**, K. Tracton  
Eine Einführung in die Computersprache LISP. LISP ist eine spezielle Interpretation, die sich gut für die Programmierung im Bereich künstliche Intelligenz eignet. Das Buch zeigt Ihnen Wege und Lösungen. Viele Programmbeispiele. 210 Seiten.

Best.-Nr. 1045

DM 24,80



**How to Design, Build and Program your own Working Computer System**

R. Haviland

Dieses Buch zeigt Ihnen Schritt für Schritt, wie Sie einen eigenen Computer aufbauen können. Es geht sich dabei an den erfolgreichen SC/MP von National an. Ein Buch, das jeder SC/MP-Besitzer haben muß.

Best.-Nr. 1111

DM 29,80



**24 Tested Ready to RUN Game Programs in BASIC**, Ken Tracton

Spaß- und Spielprogramme in BASIC. Viele Programme enthalten spezielle Anpassungshinweise an die Homecomputertypen TRS-80 und PET. Viele Graphik- und Zeichenprogramme. Ideal für jeden BASIC-Computer-Besitzer.

Best.-Nr. 1085

DM 24,80



**34 MORE Tested Ready-To-Run Game Programs in BASIC**

34 Spielprogramme für Video Genie und TRS-80, aber auch für andere BASIC Personalcomputer. Ein Buch, welches Ihnen viel Freude mit Ihrem Computer bereiten wird.

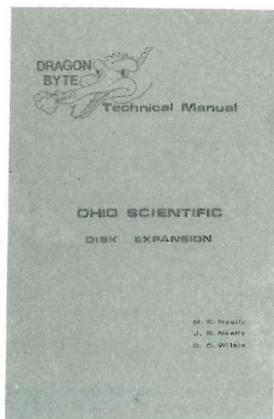
Best.-Nr. 1228

35,00 DM



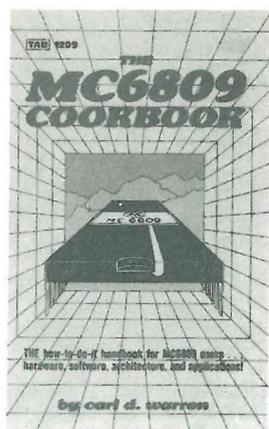
### 67 Ready to Run Programs in BASIC

Ein Buch mit vielen Programmbeispielen für alle die einen BASIC-Computer haben (ab 4K RAM). Viele Tips, Anregungen und Hilfen bringen auch dem Erstanwender großen Nutzen.  
**Best.-Nr. 1195** **29,80 DM**



### Dragon Byte Disk Expansion Book

Ein Büchlein für den, der seinen Ohio Scientific Computer zum Diskettensystem selbst ausbauen will. Sehr gute technische Informationen, die Sie sonst nur schwer finden.  
**Best.-Nr. 311** **29,80 DM**



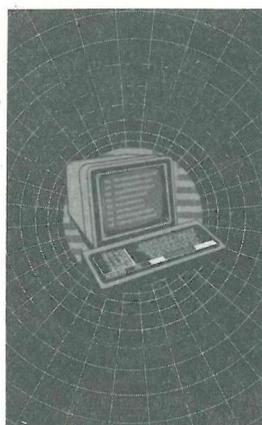
### The MC 6809 Cookbook

Endlich ein ausführliches Handbuch für den extrem Leistungsfähigen 6809 Prozessor, Hardware, Software, Architektur. Wie macht mans? Von Carl Warren.  
**Best.-Nr. 1209** **29,80 DM**



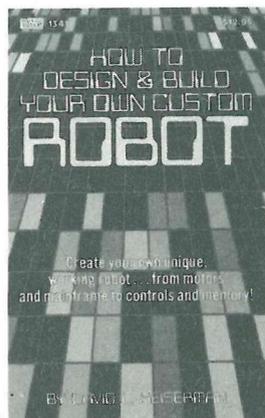
### Computer Graphics with 29 ready-to-run Programs

Eine Einführung in die Microcomputergrafik für Anfänger. Eine wahre Fundgrube für denjenigen der gute Programmbeispiele mit Grafik sucht. (Ideal für Video Genie und TRS-80 Level II BASIC.)  
**Best.-Nr. 1276** **39,00 DM**



### How to build your own working Microcomputer

Dieses Buch zeigt Ihnen wie Sie sich selbst einen einfachen Microcomputer bauen können. Programmispiele, Funktionsdiagramme, Schaltbilder. Auch für Anfänger geeignet.  
**Best.-Nr. 1200** **49,00 DM**



### How to Design and Build your own Custom Robot

Bauen Sie Ihren eigenen Roboter, der Ihre Arbeiten erledigen kann. Einführung, elektrische und mechanische Grundlagen, Microcomputer-Hardware (8085 und 780) u. v. a.  
**Best.-Nr. 1341** **59,00 DM**

**PROGRAMM BIBLIOTHEK**

**BASIC**



**Application Software!**

# INHALTSVERZEICHNIS DER BASIC PROGRAMMBIBLIOTHEK VOL I - VOL V

## VOLUME ONE

### Part 1 - Business + Personal Bookkeeping Programs

Name	Description
Bond	Computes price and interest for bond purchases.
Building	Analyzes the cost of building design proposals.
Compound	Computes effective compound interest rates.
Cyclic	Determines seasonal coefficients for two cycles.
Decision 1	Makes a Lease/Buy decision for you.
Decision 2	Makes a decision on whether to buy a component or make it.
Depreciation	Calculates depreciation by 4 different methods.
Efficient	Cal. the most efficient assignment of resources and / or personnel.
Flow	Predicts your yearly cash flow.
Installment	Performs monthly installment accounting.
Interest	Computes interest accruals, monthly.
Investments	Computes annual rates of return on investments.
Mortgage	Makes a comparison of mortgage terms.
Optimize	Optimizes the layout for a plant, shop, office, etc.
Order	Determines your economic order quantity for inventory items.
Pert Tree	Performs an analysis of a pert network.
Rate	Computes true annual interest rates.
Return 1	Computes lessor's rate of return for uncertain assets.
Return 2	Computes a lessor's rate of return after taxes.
Schedule 1	Schedules N jobs in a shop with M machines.

### Part 2 - Games + Pictures

Name	Description
Animals Four	Teach the computer all about animals.
Astronaut	Land your spaceship on another planet.
Bagel	Advanced number game, numbers may be algebraic, few clues.
Bio Cycle	Calculate your Bio-Life Cycle and plan your days.
Cannons	An advanced war game with big guns.
Checkers	Plays a regulation game of checkers.
Craps	A dice game with hard way odds.
Dogfight	Air fight w/missiles; between a phantom and a mig.
Golf	Plays any number of holes; inc. obstacle course.
Judy	Have a rap session with Judy via your computer.
Line Up	Simple number game, all you have to do is unscramble them.
Pony	Authentic horse race, any number of players.
Roulette	Gamblers delight, plays Las Vegas rules.
Sky Diver	Sky dive on another planet
Tank	A war game between two tanks.
Teach Me	Teach the computer to learn new things.
A Newman	He's absolutely MAD! MAD! MAD!
J. F. K.	Our 35th. president.
Linus	Loveable "Peanuts" character, w/blanket.
Ms. Santa	A modern miss to put a twinkle in your eye.
Nixon	Former "United States" president.
Noel Noel	Christmas or anytime this is a beautiful creation.
Nude	A true work of art for anyone's gallery.
Peace	A message for all seasons.

Name	Description
Policeman	True and blue, he's the law.
Santa's Sleigh	In banner form, perfect for decorating the mantle.
Snoopy	That paragon of Dogdom even plays football.
Virgin	A picture you can read as well as see.

## VOLUME TWO

### Part 3 - Math + Engineering Programs

Name	Description
Beam	Evaluates and selects steel beam sizes.
Conv.	Calculates convolutions.
Filter	Calculates low pass filter components.
Fit	Performs interpolations by spline fits.
Integration 1	Uses Gaussian Quadrature to do integration.
Integration 2	Integrates a function by spline fits.
Intensity	Calc. and plots RF or Acoustic intensities.
Lola	Calc. Long. and Lat. from interstellar fix or distance.
Macro	Simulates a language compiler.
Max. Min.	Calc. the max. + min. values of funct. over a special interval.
Navaid	Calc. position from altitude and azimuth of celestial bodies.
Optical	Calculates Blackbody energies, w/filter look-up tables.
Planet	Calculates Sun and Moon positions, hourly.
PSD	Calculates Power Spectral Densities and FFT's.
Rand 1	Generates random numbers between 0 and 1.
Rand 2	Generates random integers between (X) and (Y).
Solve	Solves polynominals by "Bairstows Methods".
Sphere Trian	Solves any spherical triangle.
Stars	Locates 50 stars (celestial).
Track	Calc. course and distance and incremental vectors.
Triangle	Solves for all parts of any triangle.
Variable	Finds all variables in BASIC programs.
Vector	Calc. final position; given start and motion vectors

### Part 4 - Plotting + Statistics Programs

Binomial	Calculates binomial probability distributions.
Chi-Sq.	Applies the Chi-Square test to samples.
Coeff	Calc. coefficients of fourier series to apprx. a function.
Confidence 1	Calculates confidence limits on linear regressions.
Confidence 2	Calculates confidence limits for a sample mean.
Correlations	Performs auto and cross correlations with plots.
Curve	Fits 6 different curves by the least squares method.
Differences	Calculates difference of means in non-equal variances.
Dual Plot	Plots two functions on the same sheet.
Exp-Distri	Calculates exponential distributions for a sample.
Least Squares	Performs least squares fit by linear, exp., or power function.
Paired	Compares 2 groups of data using the rank test.
Plot	Plots 6 equations on the same sheet.
Plotpts	Plots data points on standard teletypes.
Polynomial Fit	Performs least squares polynomial fit.
Regression	Performs multiple linear fit with or without transformations.
Stat 1	Finds the mean, variance and standard deviations.

Name	Description
Stat 2	Computes various stat. measures for a variable.
T-Distribution	Calculates normal and T-distributions.
Unpaired	Compares 2 groups of unpaired data.
Variance 1	Performs one way analysis of variances.
Variance 2	Analyzes a variance table of one way random design.
XY	Plots functions of X and Y.

## APPENDIX A - BASIC STATEMENT DEFINITIONS

### VOLUME THREE

#### Part 5 - Advanced Business Programs

Billing	Performs posting and billing of accounts.
Inventory	Maintains data for inventory records.
Payroll	Computes payrolls with full set of deductions.
Risk	Performs a risk analysis on capital investments.
Schedule 2	Performs the most effi. scheduling of men or resources to loca.
Shipping	Solves the problem of scheduling and assignments.
Stocks	Computes the value of stocks.
Switch	Calculates the effects of a bond switch.

### VOLUME FOUR

#### General Purpose Programs

Bingo	An age old favorite. "B9, C23, D4, E13, F21, BINGO!
Bonds	Computes the yields for a bond for different periods.
Bull	If you ever dreamed of being a Matador, here's your chance.
Enterprise	Take charge of the Enterprise while Capt. Kirk is on leave.
Football	Authentic NFL version of this well known sport.
Funds 1	Calculates long-term predictions of funds.
Funds 2	Plots the results of Funds 1.
Go-Moku	Ancient Chinese game of chance.
Jack	Plays Blackjack, Las Vegas style.
Life	Life is truly a battle for survival, a real challenger!
Loans	Calculates annuities, loans and mortgages.
Mazes	Generates unique maze puzzles for you to solve.
Poker	Five card draw - for up to 5 players.
Popul	Performs population projections for defined areas.
Profits	Determines the profitability of a firms various depts.
Qubic	3-Dimensional Tic-Tac-Toe.
Rates	Calc. the effective annual interest rate for stated interest.
Retire	Calculates your Civil Service Retirement benefits.
Savings	Computes savings plan profiles.
SBA	Calculates repayment schedules for SBA loans.
Tic-Tac-Toe	An all time favorite for young and old alike.

### VOLUME FIVE

#### Experimenter's Programs

Andy Cap	Draws this famous cartoon character.
Baseball	Plays a full 9 innings of baseball.
Compare	Compares two groups of data.
Confid 10	Determines the confidence limits for a normal population.
Descrip	Provides a description of uni-variant data.
Differ	Computes the diff. of the means for data of equal variance.
Engine	Calculates the otto cycle of engines.
Fourier	This program evaluates fourier series.
Horse	Draws a picture of a horse.
Integers	Computes integers as the sum of other integers.

Name	Description
Logic	Determines conclusions from logic statements.
Playboy	Draws the playboy symbol.
Primes	Factors numbers into their primes.
Probal	Calc. Chi-Sq. and probabilities from 2X2 data sets.
Quadratc	Solves quadratic equations
Red Baron	Draws a picture of the infamous Red Baron.
Regression 2	Calculates linear regressions.
Road Runner	"Beep! Beep!" Draws a picture of the Road Runner.
Roulette	Computerized "Wheel of Fortune", plays roulette.
Santa	Old Saint Nick appears as jolly as ever.
Stat 10	Calculates quantities for two groups of paired data.
Stat 11	Computes sample statistics.
Steel	Calculates steel beam capacities.
Top	Computes cost for surfacing a road or driveway, etc.
Vary	Performs an analysis of a vari. table; one-way random design.
Xmas	Generates a "SINGING" Christmas card.

#### APPENDIX B - STATEMENT CONVERSION ALGORITHMS

##### VOLUME SIX

##### A Complete Business System

Ledger	Maintains ALL Company accounts and generates ALL financial reports. Includes routines for: Pyrl, Inv. Depr. A/R, A/P, Balance Sheets and Profit + Loss statements, etc.
--------	--

ACBS rev: 80 Users Manual - A Proprietary Package

##### VOLUME SEVEN

##### Professional Programs

Chess	Designed to challenge the average player, fairly comprehensive. Great fun for all, offers a unique opportunity for beginners in need of an opponent.
Medbil	For Doctors and Dentists alike, a complete patient billing system which also permits the maintaining of a patient history record.
Wdproc	Wordprocessing for lawyers, publishers, writers etc. Write, store and change from rough draft to final copy in a variety of formats.

##### VOLUME SEVEN

##### Professional Programs

##### Utility

##### Licensing Agreement

#### BESTELLSCHHEIN

Heute noch an Ihren Buch- oder Fachhändler absenden! Bitte senden Sie mir folgende Bücher:

..... Stck.	Vol. I , Best.Nr. 80/50, DM 99,--
..... Stck.	Vol. II , Best.Nr. 80/51, DM 99,--
..... Stck.	Vol. III, Best.Nr. 80/52, DM 149,--
..... Stck.	Vol. IV, Best.Nr. 80/53, DM 39,--
..... Stck.	Vol. V , Best.Nr. 80/54, DM 39,--
..... Stck.	Vol. VI, Best.Nr. 80/48, DM 199,--
..... Stck.	Vol. VII, Best.Nr. 80/49, DM 159,--
..... Stck.	Programm Bibliothek BASIC, (Volume I bis Vol. V) Best.Nr. 80/21, DM 425,--

**Hofacker Verlag**

## GRUNDAUSSTATTUNG

Die Grundausrüstung der BASIC Programmbibliothek besteht aus fünf Büchern mit insgesamt 1100 Seiten Programm Listings und genauen Beschreibungen. (Format DIN A 4)

Sie finden in diesem Werk 149 verschiedene BASIC Programme aus folgenden Bereichen:

- Buchhaltung
- Computerspiele
- Bildprogramme
- Mathematik und Ingenieurwissenschaften
- Statistik, technische Mechanik etc.
- Commerzielle Programme, Rechnungswesen, Fakturierung
- Inventurprogramme, Lohn- und Gehaltsbuchhaltung
- Experimentierprogramme u.v.a.

Jedes Programm ist genau beschrieben.

Die fünf Bände bilden eine komplette "Do It Yourself" Programmbibliothek, die sehr einfach zu handhaben ist. Es werden jedoch Grundkenntnisse in der Programmierung vorausgesetzt.

## PREISENSATION

Bei 149 Programmen in dieser umfangreichen Bibliothek bezahlen Sie nur  
DM 2,85 pro BASIC Programm - Das ist einmalig!

## BESCHREIBUNG DER PROGRAMMBIBLIOTHEK

Diese Bibliothek ist die umfangreichste und universalste ihrer Art auf dem Weltmarkt. Die Besonderheit dieser fünf Werke liegt darin, daß dem Anwender die Möglichkeit gegeben wird, in erster Linie sinnvolle und produktive Aufgaben wie Buchhaltung, Statistik, mathematische Probleme etc., zu lösen. Computerspiele sind jedoch auch enthalten.

Alle Programme wurden mehrmals auf verschiedenen Systemen ausgeführt und getestet. Jedes Programm ist mit einer genauen Beschreibung und einer Aufstellung der infrage kommenden Anwender versehen. Befehle und mögliche Grenzen der Programme sind auch aufgezeigt, wenn die Programme auf unterschiedlichen Systemen ausgeführt werden sollen. Auch der benötigte Speicherbereich ist für jedes Programm genau angegeben.

Jedes Programm ist in seiner vollständigen Größe aufgeführt und nicht reduziert. Es ist in sich selbst vollständig und kann im Ablauf genau verfolgt werden. Es gibt Aufschluß über alle wichtigen Daten. Bei den meisten Programmen folgt unmittelbar nach dem Programm Listing ein Probelauf. Die meisten Programme sind in kompatibelem BASIC geschrieben, welche auf den meisten 4K BASIC-Versionen ausgeführt werden können.

Best.Nr. 80/21, 5 Bücher mit 1100 Seiten DIN A 4 (149 Programme) nur DM 425,--

Die Bände sind auch einzeln erhältlich:

### BASIC Volume I

Programme über Buchhaltung, Computerspiele und Bilder (Computerbilder per Programm)

Best.Nr. 80/50 DM 99,--

### BASIC Volume II

Programme über mathematische Probleme, technische Programme, Zeichnen, Plotting und Statistik. Grundlegende Statements.

Best.Nr. 80/51 DM 99,--

### BASIC Volume III

Erweiterte Geschäftsprogramme für den kommerziellen Bereich. Fakturierung, Inventuren, Gehaltsbuchhaltung.

Best.Nr. 80/52 DM 149,--

BASIC Volume IV

BASIC Programme aus allen Bereichen. Universelle Anwendung.

Best.Nr. 80/53

DM 39,--

BASIC Volume V

Experimentierprogramme.

Best.Nr. 80/54

DM 39,--



BASIC Volume VI

Ein komplettes, interaktives Programmpaket für Floppy Disk - Einsatz im kommerziellen Bereich. Gehaltsabrechnung, Inventur, Gewinn und Verlustrechnung u. v. a.

Best.Nr. 80/48

DM 199,--

BASIC Volume VII

Das Schachprogramm, auf das Sie schon lange gewartet haben. Spielt zwei Farben. Leichter bis mittlerer Schwierigkeitsgrad. Kann einfach erweitert werden. Zugzeit ab 3 Minuten.

Best.Nr.

Medical Billing package (Patientengeschichte und Kostenübersicht für Ärzte)

Disk interaktives Textverarbeitungssystem.

Best.Nr. 80/49

DM 159,--

Genauere Beschreibung des BASIC Volume VII

Schachprogramm (Chess)

Diese Version eines Schachprogrammes in BASIC ist für die meisten 12 K BASIC-Versionen (mit oder ohne kleinen Änderungen) geeignet. Weiterhin sollten noch ca. 12 K im RAM-Bereich zur Verfügung stehen. Das Programm spielt Schach im Schwierigkeitsgrad für Anfänger, kann aber durch entsprechende Änderungen auch schwieriger gestaltet werden. Züge werden im Bereich von 3 Minuten oder mehr durchgeführt. Der erste Zug benötigt ca. 3 Minuten. Die Zugzeit steigt dann entsprechend dem Spielfortschritt entsprechend an. Das Programm kann wahlweise schwarz oder weiß spielen. Das Spielfeld wird auf dem Bildschirm angezeigt oder kann auf einem Drucker ausgegeben werden.

BASIC-Programm für die Arztpraxis (Medbil)

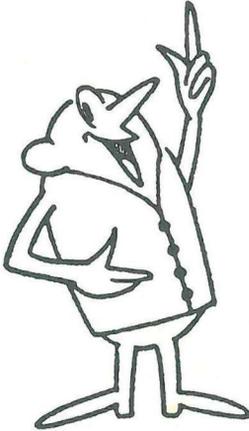
Dieses Programm wurde zur Erleichterung der so teuren und mühsamen Alltagsarbeit der Ärzte entwickelt. Dieses Programm ermöglicht die Speicherung und Durchsicht jeder Patientengeschichte auf dem Bildschirm, die vorher in einer Datenbank gespeichert wurde. Weiterhin besteht eine Möglichkeit der schnellen Überprüfung von aufgelaufenen Behandlungskosten pro Patient. Auf einer 250 K Byte Diskette können etwa 110 Patienten mit all ihren Daten aufgenommen werden. Mit einer Minifloppy (80 K Byte) reduziert sich die mögliche Patientenzahl auf ca. 25 pro Diskette.

Textverarbeitungsprogramm (Wrdpro)

Dieses Programm ermöglicht dem Besitzer von Microcomputern Texte zusammenzustellen und in beliebigen Formaten wieder auszudrucken. Das Programm läuft auf den meisten extended BASIC-Versionen mit mindestens 15 K freiem RAM-Bereich.

**W. Hofacker  
Verlag**

Auch Sie brauchen ELCOMP!



Jahresabonnement **69,00 DM**  
incl. Mwst. und Versand.

Zurückliegende Hefte zu  
Originalpreisen noch verfü-  
gbar.

# ELCOMP



## HOFACKER-VERLAG

Ing. W. Hofacker GmbH  
Tegernseer Straße 18

**D-8150 Holzkirchen/Obb.**

Die Fachzeitschrift für MICROCOMPUTER  
Eine unentbehrliche Informationsquelle für alle Elektroniker

Microcomputer-Anwendungsbeispiele  
Künstliche Intelligenz  
Block-Strukturierte Programme  
Datenverarbeitung im Kleinbetrieb  
Club-Neuheiten  
Computer und Kunst  
Musik mit dem Computer  
Monitore für 8080, 6800, 6502, Z 80,  
SC/MP, 2650, 1802  
Eigenbau-Computersysteme  
Interface-Techniken  
Microcomputer KITS

Neue Produkte  
Betriebssysteme für Floppys  
Programmiertechniken  
Software-Quellen  
Programmierbeispiele  
Soziale Aspekte der Microcomputer-  
technik  
Technologische Neuheiten  
Anwendungen in der Meß- und Regel-  
technik  
Anwendungen bei Funk-Amateuren



# ELCOMP

Absender  
Bitte deutlich ausfüllen

\_\_\_\_\_  
Vorname/Name

\_\_\_\_\_  
Beruf

\_\_\_\_\_  
Straße/Nr.

\_\_\_\_\_  
Plz      Ort

**ABSENDER:**

.....  
Name, Vorname

.....  
Straße

(    )  
.....  
PLZ      Ort

.....  
Telefon

Das Abonnement wird automatisch verlängert  
(Kündigung 8 Wochen z. Abonnement-Ablauf)

POSTKARTE

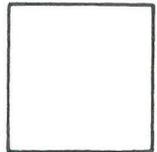


# ELCOMP

MIKROCOMPUTER BOOK STORE

Tegernseerstr. 18

D-8150 Holzkirchen /Obb.



# ELCOMP

Ing. W. Hofacker GmbH  
Tegernseer Straße 18

D-8150 Holzkirchen

M  
K  
B  
D  
C  
C  
M  
S  
E  
I  
R  
N

## Software und Hardware für

# VC-20

### **3K RAM Expander für Programmierer und ROM-Entwickler**

Diese Platine bringt Ihnen eine 3K RAM-Erweiterung im Bereich 0400-1000 hex, sowie zwei 24 pin ROM/EPROM-Sockel, in die Sie 2K bis 16K ROM oder EPROM einstecken können. Mit Hilfe eines 8poligen DIP-Schalters können Sie Anfangsadresse und Sockel Ein/Aus festlegen.

**Best.-Nr. 4860** (fertig)

**DM 269,00**

### **RS232 Kommunikationsinterface für VC-20**

Mit dieser Platine verwandeln Sie Ihren VC-20 in ein RS-232 Terminal mit Sende- und Empfangs-Baudraten bis 2400 Baud. Zusammen mit einem Modem können Sie Telekommunikation betreiben. Keine speziellen Treiberrouitinen erforderlich. Arbeitet direkt von BASIC her. Ideal für RS-232-Drucker.

(ohne Terminalprogramm (fertig) )

**Best.-Nr. 4861**

**DM 299,00**

### **Terminalprogramm für Kommunikationsinterface**

Dieses Programm erlaubt den Einsatz des VC-20 als Terminal oder als Anschlußgerät für Computernetzwerke, Großcomputer usw.

**Best.-Nr. 4862**

**DM 129,00**

### **8K RAM-/ROM-Platine**

Speziell für Programmentwickler. Mixen Sie RAMs, ROMs, EPROMs. Chipadressen können in 2K Abständen gewählt werden. (Ab 2000, 4000, 6000, 8000).

**Best.-Nr. 4863** (ohne Teile)

**DM 149,00**

### **BASIC Programmers UTILITY ROM**

Ein ROM für Ihre 4860-Platine oder RAM/ROM-Board. Startadresse an \$ A000. BASIC Auto-Line-Number, Renumber, Variable-Dump, Tape-Append, Block-Delete, Number-Conversion, Renew Pack/Unpack und einiges mehr.

**Best.-Nr. 4864**

**DM 199,00**

### **Alien Blitz**

Dieses Spiel gehört zur Spitzenklasse. In Farbe, mit Bewegung und Ton. Ein aufregendes Spiel, ähnlich Alien Invaders. Punkte werden gezählt. Wahlweise mit Joystick oder über die Tastatur.

**Best.-Nr. 4865**

**DM 99,00**

### **Amokläufer**

Ein aufregendes Bildschirmspiel mit Grafik, Bewegung und Ton. Ein kleines Männchen muß durch eine Anzahl von Robotern zu einem Ausgang eines Gartens Amok laufen. Die Roboter greifen an. Das Spiel wird durch Hindernisse erschwert. Sehr schöne Grafik. Läuft auf der Grundversion. Wahlweise mit Joystick oder über die Tastatur.

**Best.-Nr. 4866**

**DM 99,-**

### **Monster-Maze und Hurdler**

Zwei sehr nette Spiele für die VC-20 Grundversion. Monster-Maze ist eines unserer schönsten Spiele. Ein Männchen, gesteuert über Joystick oder Tastatur, muß durch einen Irrgarten zum Ausgang gelangen. Der Irrgarten baut sich erst während des Spieles auf. Auf dem Weg zum Ausgang wird das Männchen von Ungeheuern bedroht, die sich teilweise sogar versteckt halten. Grafik, Bewegung und Ton.

Hurdler ist ideal für Kinder. Ein Hürdenläufer rennt über eine bestimmte Distanz. Zwischenzeitlich müssen Aufgaben gelöst werden. Ist die Antwort richtig, überspringt der Läufer die Hürde. Bei falscher Antwort fällt die Hürde um, und der Läufer stürzt. Lustige Grafik mit Bewegung und Ton.

**Best.-Nr. 4842**

**DM 69,00**

### **Hangmath und Hangman**

Sehr nett aufgemachte Version des bekannten Galgenspiels. Sie werden begeistert sein. Zwei Spiele.

**Best.-Nr. 4841**

**DM 69,00**

### **Logik-Games – Code Breaker and Code-maker**

Diese Spiele eignen sich bestens zur Unterhaltung. Zusätzlich zeigen diese Spiele auf einfachste Weise, was ein Computer zu leisten vermag. Phantastisch, mit Sound und Farbgrafik.

**Best.-Nr. 4840**

**DM 79,00**

Lieferung gegen NN oder Vorkasse auf Postcheckkonto München Nr. 15 994-807

**Ing. W. Hofacker GmbH**  
**Tegernseerstr. 18**  
**D-8150 Holzkirchen**  
**Tel.: 08024/7331**

# Weitere interessante Bücher von Hofacker:

Best.-Nr.	Titel	Preis/DM	Best.-Nr.	Titel	Preis/DM
<b>Bücher in deutscher Sprache</b>					
1	Transistor Berechnungs- u. Bauanleitungsbuch -- 1	29,80	122	BASIC für Fortgeschrittene	39,00
2	Transistor Berechnungs- u. Bauanleitungsbuch -- 2	19,80	123	IEC-Bus Handbuch	19,80
3	Elektronik im Auto	9,80	124	Programmieren in Maschinensprache mit CBM	19,80
4	IC-Handbuch, TTL, CMOS, Linear	19,80	127	Einführung i. die Microcomputer- Progr. mit 6800	49,00
5	IC-Datenbuch, TTL, CMOS, Linear	9,80	128	Programmieren mit dem CBM	29,80
6	IC-Schaltungen, TTL, CMOS, Linear	19,80	129	ELCOMP-Leser Programmierhandbuch	69,00
7	Elektronik Schaltungen	9,80	130	Programmierbeispiele für CBM	19,80
8	IC-Bauanleitungsbuch	19,80	131	COBOL für Anfänger	19,80
9	Feldeffekttransistoren	9,80	132	CP/M-Handbuch	19,80
10	Elektronik und Radio	19,80	133	Welches Betriebssystem brauche ich ?	19,80
11	IC-NF Verstärker	9,80	139	BASIC für blutige Laien	19,80
12	Beispiele Integrierter Schaltungen (BIS)	19,80	140	ZX81 Programmierhandbuch	29,80
13	HEH, Hobby Elektronik Handbuch	9,80	141	Programmierhandbuch für VC-20	29,80
14	IC-Vergleichsliste	29,80	<b>Bücher in englischer Sprache</b>		
15	Optoelektronik Handbuch	19,80	150	Care and Feeding of the Commodore PET	19,80
16	CMOS Teil 1, Einführung, Entwurf, Schaltbeispiele	19,80	151	8k Microsoft BASIC Reference Manual	19,80
17	CMOS Teil 2, Entwurf und Schaltbeispiele	19,80	152	Expansion Handbook for 6502 and 6800	19,80
18	CMOS Teil 3, Entwurf und Schaltbeispiele	19,80	153	Microcomputer Application Notes	29,80
19	IC-Experimentier Handbuch	19,80	154	Complex Sound Generation using the SN76477	19,80
20	Operationsverstärker	19,80	155	The First Book of 80-US (TRS-80)	19,80
21	Digitaltechnik Grundkurs	19,80	156	Small Business Programs	29,80
22	Mikroprozessoren, Eigenschaften und Aufbau	19,80	157	The First Book of Ohio Scientific	19,80
23	Elektronik Grundkurs, Kurzlehrgang Elektronik	9,80	158	The Second Book of Ohio Scientific	19,80
24	Microcomputer-Technik	29,80	159	The Third Book of Ohio Scientific	19,80
25	Hobby Computer Handbuch	29,80	160	The Fourth Book of Ohio Scientific	29,80
26	Mikroprozessor, Teil 2	19,80	161	The Fifth Book of Ohio Scientific	19,80
27	Mikrocomputer Software Handbuch	29,80	162	ATARI Games in BASIC	19,80
28	Lexikon + Wörterb. f. Elektr. u. Mikroprozessor LEM	29,80	163	The Peripheral Handbook	29,80
29	Mikrocomputer Datenbuch	49,80	164	ATARI-BASIC Learning by Using	19,80
30	Aktivtraining Mikrocomputer	49,80	1050	The Most Popular Subroutines in BASIC	24,80
31	57 Programme in BASIC	39,00	1053	Microprocessor Cookbook	24,80
32	ATARI BASIC Handbuch	29,80	1055	The BASIC Cookbook	24,80
33	Microcomputer Programmierbeispiele	19,80	1062	The A to Z Book of Computer Games	29,80
34	TINY-BASIC Handbuch	19,80	1070	Digital Interfacing with an Analog World	39,00
35	Der freundliche Computer	29,80	1071	The Complete Handbook of Robotics	29,80
103	Oszillographen-Handbuch	19,80	1076	Artificial Intelligence	29,80
104	1000 Elektronisk Schaltungen	49,00	1085	24 Tested Ready to RUN Game Programs in BASIC	24,80
107	Praktische Antennentechnik	19,80	1088	Illustrated Dictionary of Microcomputer Terminology	35,00
108	SC/MP Mikrocomputer-Handbuch	29,80	1095	Programs in BASIC for Electronic Engineers	19,80
109	6502 Microcomputer Programmierung	29,80	1099	How to Build Your own Working 16-Bit Microc.	14,80
110	Programmierhandbuch für PET	29,80	1141	How to Build Your own Working ROBOT PET	29,80
111	Programmieren mit TRS-80	29,80	1160	1001 Things to do with Y.P.C.	29,80
112	PASCAL-Programmier-Handbuch	29,80	1169	The Giant Book of Computers	39,00
113	BASIC-Programmier-Handbuch	19,80	8029	Z-80 Assemblerhandbuch	29,80
114	Der Microcomputer im Kleinbetrieb	39,80	8042	6500 Software Manual	19,80
115	6809 Programmier Handbuch	49,00	8043	6500 Hardware Manual	19,80
116	Einführung 16-Bit Microcomputer	29,80	8048	BASIC Software Vol. VI	199,00
117	FORTRAN für Heimcomputer	19,80	8049	BASIC Software Vol. VII	159,00
118	Programmieren in Maschinensprache mit dem 6502	49,00	8050	BASIC Software Vol. I	99,00
119	Programmieren in Maschinensprache (Z80)	49,00	8051	BASIC Software Vol. II	99,00
120	Anwenderprogramme für TRS-80 u. Video Genie	29,80	8052	BASIC Software Vol. III	149,00
121	Microsoft BASIC-Handbuch	29,80	8053	BASIC Software Vol. IV	39,00
			8054	BASIC Software Vol. V	39,00

# HOFACKER

HOLZKIRCHEN

LOS ANGELES