

Mikro+Kleincomputer
Informa Verlag AG
Postfach 1401, CH-6000 Luzern 15

3. Jahrgang
Erscheint 6 mal jährlich
ISSN 0251-2017

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

Lieber Commodore-Freund,

die Rubrik "Hardware" ist bestimmt derjenige Teil des Heftes, der am meisten umstritten ist, da er nur bei einem ganz bestimmten Teil der Leserschaft auf Interesse stösst. Sicher ist diese Rubrik vor allem für denjenigen Leser interessant, welcher selbst einmal gerne einen Lötkolben in die Hand nimmt und sich das Erfolgserlebnis des "Selbstgemacht" nicht entgehen lassen will. Andererseits zeigt uns aber die ständige Nachfrage nach "Fertiggeraeten", die teilweise in grösseren Stückzahlen bezogen werden, dass auch der technisch weniger versierte Leser hier manche nützliche Anregung findet.

Da mit dem Jahrgang 1982 die fortsetzungsreiche USER-PORT Erweiterung Ihren Abschluss findet, möchten wir gerade für diese Rubrik ein Thema wählen, das in Zukunft einen möglichst breiten Leserkreis anspricht. Wir bitten Sie daher lieber Leser, um möglichst zahlreiche Zuschriften, welche "Hardware"-Wünsche wir erfüllen sollen.

Bedenken Sie aber bitte, dass das Kapitel Hardware auch für uns nicht ohne Probleme ist, da eine Neuentwicklung oder eine Erweiterung von der Idee bis zum fertigen Geraet einen enormen Aufwand an Zeit und Material erfordert, wenn gewährleistet sein soll, dass das Geraet auch fehlerfrei laeuft. Wir waeren daher sehr froh, wenn Sie uns nicht nur Anregungen zu neuen Geraeten geben würden, sondern wenn sich auch Leser finden würden, die - selbstverstaendlich mit unserer Hilfe - bereit waeren, Geraete oder Erweiterungen selbst zu entwerfen oder zumindest zur Produktionsreife zu bringen. Bitte setzen Sie sich doch einmal mit uns in Verbindung.

Viel Freund mit Commodore wünscht Ihnen im Namen der Redaktion



Heinz Kastien

Inhaltsverzeichnis

2 Artikelliste, Inventar	16 Barcodeausdruck
6 Adressliste, Adressetiketten	19 Leserbriefe
10 Barcodepack	20 Zeichen ROM für CBM 8032
11 Vergleicher	21 Umrechnungen auf VC 20
14 Wussten Sie schon	23 Redaktionelle Aenderungen

© 1982 by Mikro+Kleincomputer Informa Verlag AG, Luzern, aber Speicherung in Datenverarbeitungsanlagen für den eigenen Gebrauch erlaubt. Nachdruck, auch auszugsweise, sowie Vervielfältigungen, jedwelcher Art, nur mit schriftlicher Genehmigung des Verlages und unter voller Quellenangabe.

zur Übersetzung in andere Sprachen erworben. Für die Veröffentlichung wird keine Gewähr oder Garantie übernommen, auch nicht dafür, dass die verwendeten Schaltungen, Firmennamen und Warenbezeichnungen usw. frei von Schutzrechten Dritter sind. Die Verwendung der Informationen erfolgt auf eigenes Risiko.

Manuskripte
Mit der Zustellung von Manuskripten anerkennt der Autor die Copyrightbestimmungen des Verlages. Mit der Annahme von Manuskripten durch die Redaktion und der Autor-Honorierung durch den Verlag hat dieser das Recht zur Veröffentlichung der entsprechenden Beiträge in anderen verlags eigenen Publikationen und

Tricks und Tips

LAGERBUCHHALTUNG

ADRESSVERWALTUNG

Der wohl wichtigste Teil einer Lagerbuchhaltung ist am Ende des Jahres das Inventar, das in schriftlicher Form den Lagerbestand und den Lagerwert angibt. Ebenso grosse Bedeutung hat eine Artikelliste, die jederzeit Auskunft über die am Lager befindlichen Artikel, deren Preise und Lieferanten gibt. Das gleiche gilt für die Adressliste einer Adressverwaltung.

Das Programm zur Erstellung der oben genannten Listen ist prinzipiell ein Abrufprogramm, das aber nicht einen einzelnen Datenstring, sondern alle Datenstring nacheinander abrufen. Die Floppyroutine und der gesamte Abruf ist daher denkbar einfach. Die programmtechnischen Schwerpunkte sind daher bei diesem Programm nicht auf Abrufen und Floppyroutinen gelegt, sondern auf den Ausdruck der Daten auf dem Printer, wobei es von grösster Wichtigkeit ist, dass die Daten tabellarisch ausgedruckt werden, was ja bekanntlich bei den normalen Druckern einige Schwierigkeiten bietet, da die Drucker, genau wie der Bildschirm normalerweise linksbündig drucken und bei allen Zahlen die Vor- und Nachkommastellen unterdrückt werden.

Sowohl bei den Commodoredruckern, als auch bei einigen anderen Druckertypen besteht die Möglichkeit des formatierten Ausdrucks, das heisst, in einem Formatfile wird festgelegt, wie die auszudruckende Zeile aussehen soll, also Anzahl Zeichen eines Textes, Anzahl Stellen einer Zahl, Vor- und Nachnullen usw. Leider ist dieser formatierte Ausdruck in jedem Fall Drucker spezifisch und funktioniert bei Wechsel des Druckers nicht mehr. Wir zeigen Ihnen daher den formatierten Ausdruck nicht, er ist ausserdem im Handbuch gut beschrieben. Beim Inventar und Artikellistenprogramm beschreiten wir einen anderen Weg, der es uns gestattet, jeden beliebigen Drucker zu verwenden, ohne dass Programmänderungen vorgenommen werden müssen.

Man erreicht diese Universalität, indem man alle auszudruckenden Texte, wie z.B. den Artikelnamen auf die gleiche Länge bringt, dies ist sehr leicht durch Auffüllen mit Space möglich. Das Gleiche gilt auch für Zahlen. Bei Beträgen wird auf zwei Nachkommastellen gerundet und gleichzeitig mit einer "0" oder mit ".00" aufgefüllt, wenn die Zahl nur eine oder keine Nachkommastellen hat, natürlich muss auch hier mit Space aufgefüllt werden, jedoch müssen diesmal die Space vor der Zahl stehen.

Werden diese Regeln eingehalten können Tabellen ohne Formatierung oder Print Using ausgedruckt werden. Alle Texte erscheinen nun linksbündig und alle Zahlen rechtsbündig, wie wir dies von Tabellen gewohnt sind. Ein Nachteil dieses Verfahrens darf allerdings nicht unerwähnt bleiben, jeder String muss aufgearbeitet werden und jede Zahl muss vor Ihrer Aufarbeitung in einen String umgearbeitet werden. Dies erfolgt meist in einer oder mehreren Subroutinen, die aber Zeit brauchen, das heisst, diese Programme laufen in der Regel langsamer als entsprechende Programme mit Formatausdruck.

Sofern Sie einen CBM 8000 mit Toolkit besitzen, so können Sie den PRINT USING Befehl benutzen, der ebenfalls einen Formatausdruck erlaubt, allerdings ist auch hier zu berücksichtigen, dass durch den Toolkit die Programme bis zu 8 % langsamer laufen als ohne Initialisierung desselben.


```

14230 print#4,"Art. Nr. Artikelname Lieferant ";
14240 print#4," E. Preis U. Preis"
14250 fori=1to79:print#4,"-";next:print#4
14260 zz=zz+5:goto10000
14500 i$=str$(val(mid$(g$(b%),52,7))):gosub15900:ep$=i$
14510 i$=str$(val(mid$(g$(b%),59,7))):gosub15900:vp$=i$
14520 print#4,left$(g$(b%),6);" ";mid$(g$(b%),7,25);mid$(g$(b%),32,20);
14530 print#4,ep$;" Fr. ";vp$;" Fr. ":jj=jj+1:zz=zz+1
14540 ifzz<69thenreturn
14570 fori=1to3:print#4;next:zz=0:s=s+1:goto14200
15000 rem inventar
15010 a1$="*** I n v e n t a r ***"
15020 jj=0:zz=3:s=1:fori=1to3:print#4;next
15030 print#4,tab(15)chr$(1)a1$
15040 print#4,tab(15)chr$(1)"-----"
15050 print#4
15060 zz=zz+3
15070 gosub15210:goto10000
15200 fori=1to3:print#4;next:zz=zz+3
15210 print#4," Datum : ";m$; Seite : ";s
15220 print#4
15230 print#4,"Art. Nr. Artikelname E. Preis ";
15240 print#4," Lagerbestand Lagerwert"
15250 fori=1to79:print#4,"-";next:print#4
15260 zz=zz+5:return
15500 i$=str$(val(mid$(g$(b%),52,7))):gosub15900:ep$=i$
15505 lb$=mid$(g$(b%),66,6)
15510 lw$=str$(val(mid$(g$(b%),52,7))*val(mid$(g$(b%),66,6))):lw=lw+val(lw$)
15520 lb=lb+val(mid$(g$(b%),66,6)):sw=sw+val(lw$):iv=iv+val(lw$)
15540 print#4,left$(g$(b%),6);" ";mid$(g$(b%),7,25);ep$;
15545 i$=lw$:gosub15900:lw$=i$
15550 print#4," Fr. ";lb$;" ";lw$;" Fr. ":jj=jj+1:zz=zz+1
15560 ifzz<66thenreturn
15570 fori=1to3:print#4;next:zz=0:s=s+1:gosub15700:goto15200
15700 i$=str$(sw):gosub15900:print#4:print#4,"Seitentotal : ";
15710 sw=i$:print#4,tab(32)"Menge : ";lb;" ";sw$;" Fr.":sw=0:return
15900 ifval(i$)=int(val(i$))theni$=i$+".00":goto15920
15910 ifmid$(i$,len(i$)-1,1)=". "theni$=i$+"0"
15920 iflen(i$)<7theni$=" "+i$:goto15920
15930 return
16000 rem abschluss
16010 ifxx=2then16500
16020 print#4:print#4,jj;" Artikel":zz=zz+1
16030 zz=72-zz
16040 fori=1tozz:print#4;next
16050 close4
16060 return
16500 gosub15700:i$=str$(iv):gosub15900:print#4:print#4:print#4,jj;" Artikel";
16510 print#4,tab(37)"Inventarsumme : ";i$;" Fr.":zz=72-zz
16520 fori=1tozz:print#4;next
16530 close4
16540 return
17000 rem aussage auf dem bildschirm
17010 printa$
17020 print"#####Nummer : ";left$(g$(b%),6)
17030 print"#####Name : ";mid$(g$(b%),7,25)
17040 print"#####Lieferant : ";mid$(g$(b%),32,20)
17050 print"#####Einstandspreis : ";mid$(g$(b%),52,7)
17060 print"#####Verkaufspreis : ";mid$(g$(b%),59,7)
17070 print"#####Lagerbestand : ";mid$(g$(b%),66,6)
17080 print"#####Minimalbestand : ";mid$(g$(b%),72,4)
17100 return

```

*** I n v e n t a r ***

Datum : 19.08.1982

Seite : 1

Art. Nr.	Artikelname	E. Preis	Lagerbestand	Lagerwert
821100	Kondensator .0.047	.30 Fr.	100	30.00 Fr.
821101	Kondensator 0.068	.30 Fr.	100	30.00 Fr.
720931	Widerstand 6.8	1.65 Fr.	100	165.00 Fr.
Seitentotal :			Menge : 8200	4170.00 Fr.

82 Artikel Inventarsumme : 10375.00 Fr.

Programmbeschreibung Artikelliste

Auch in diesem Programm sollen die Programmteile, die mit dem Abrufprogramm identisch sind, unerwähnt bleiben.

- 1000 - 1060 Diskvorbereitung (siehe Abrufprogramm)
- 2000 - 2090 Menu und Sprungverteiler (siehe Abrufprogramm)
- 3000 - 3010 Laderoutine für das Menu (siehe Abrufprogramm)
- 4000 - 4230 Eingabe des Datums
- 8000 - 8080 Einleseroutine (siehe Abrufprogramm)
- 9000 - 9040 Fehleroutine (siehe Abrufprogramm)
- 10000 - 10070 Filebehandlung und Sprungverteiler (siehe Abrufprogramm)
- 14000 - 14570 Ausdruck der Artikelliste
- 15000 - 15570 Ausdruck des Inventars
- 15700 - 15710 Seitentotal des Inventars
- 15900 - 15930 Rundungsroutine für die Beträge
- 16000 - 16060 Abschluss Artikelliste
- 16500 - 16540 Abschluss Inventar
- 17000 - 17100 Ausgabe der Daten auf dem Bildschirm (siehe Abrufprogramm)

Bevor wir auf die Details der Ausdruckroutine eingehen, soll noch einmal kurz der Aufbau eines Datenstring, in dem alle Informationen über einen Artikel enthalten sind, dargestellt werden.

Nr.	Name	Lieferant	E.Pr.	V.Pr.	L.B.	MB.	Res
6		31	51	58	65	71	75
14000	Remark						
14010	Definition des Titels						
14020	Setzen der Variablen jj = Anzahl Artikel, s = Seitenzahl, zz = Zeilenzahl (eine A4 Seite hat 72 Zeilen), es werden drei Leerzeilen geschrieben.						
14030 - 14060	Es wird der Titel in Breitschrift (CHR\$(1)) geschrieben und der Zeilenzaehler um 3 erhöht.						
14070	Subroutinenaufruf						
14200 - 14260	Subroutine zum Ausdruck des Datums, der Seitenzahl sowie einer Kopfzeile auf jeder neuen Druckseite, zz wird um 5 erhöht.						
14500	Der Einkaufspreis ep\$ wird ermittelt und gerundet.						
14510	Der Verkaufspreis vp\$ wird ermittelt und gerundet.						
14520 - 14530	Die Artikelnummer, der Artikelname, der Lieferant und der Einkaufs- und Verkaufspreis werden ausgedruckt, Zeilenzaehler und Artikelzaehler werden mit jeder Zeile um 1 erhöht.						
14540 - 14570	Wenn der Zeilenzaehler den Wert 69 erreicht hat, werden drei Leerzeilen gedruckt, die Variablen zz zurückgesetzt und ss um 1 erhöht. Rücksprung nach 14200.						
15000 - 15020	Gleiche Routine wie in 14000-14020, jedoch diesmal für das Inventar.						
15030 - 15060	wie 14070						
15200 - 15260	wie 14200-14260						
15500	Definition des Einkaufspreises						
15510	Definition des Lagerbestandes						
15520	Multiplikation des Einkaufspreises mit dem Lagerbestand zur Ermittlung des Lagerwertes und fortlaufende Addition der Lagerwerte zum Inventartotal						

- 15530 Mit lb werden die Lagerbestaende laufend addiert und mit sw und iv die Lagerwerte
- 15540 - 15560 Die Artikelnummer, der Artikelname, der Einkaufspreis, der Lagerbestand und der Lagerwert werden ausgedruckt und der Artikelzaehler jj um 1 erhoelt.
- 15570 Solange der Zeilenzaehler kleiner 66 ist, erfolgt Aufruf des naechsten Artikels.
- 15700 - 15710 Beim Inventar wird nach jeder Seite ein Seitentotal des Wertes und der Menge erstellt, sw wird auf 0 gesetzt
- 15900 - 15930 Rundungssubroutine für Betraege, gleichzeitig wird der String auf die vorgegebene Laenge gebracht
- 16000 - 16060 Wenn in 10030 ein "end" gefunden wird und xx% = 1 ist, wird die Anzahl der Artikel ausgedruckt und die Seite mit Leerzeilen aufgefüllt
- 16500 - 16540 Wenn in 10030 ein "end" gefunden wird und xx%=2 ist, wird das Inventartotal ausgedruckt und die Seite ebenfalls mit Leerzeilen aufgefüllt

Im Lagerbuchhaltungsprogramm dürfte vor allem der Programmteil erwaehenswert sein, der die Eingabe des Datums und dessen Prüfung beinhaltet.

- 4000 - 4030 Eingabe des Datums im vorgegebenen Format
- 4040 Prüfung, ob der String 10 Zeichen lang ist
- 4050 - 4060 Zerlegen des String in Einzelzeichen
- 4070 - 4100 Prüfen, ob es sich bei diesen Zeichen um Ziffern handelt
- 4110 Prüfen ob Jahreszahl kleiner 1000 ist
- 4120 Prüfen, ob der Monat kleiner 1 oder grösser 12 ist
- 4130 - 4150 Umwandeln des Monats und des Tages in numerische Variable
- 4160 - 4170 Einlesen der DATA-Statements als indizierte Variable
- 4180 Korrektur der Variablen m(2) auf 29, wenn es sich um ein Schaltjahr handelt
- 4190 Prüfen, ob die eingebene Tageszahl nicht grösser ist, als für den betreffenden Monat überhaupt möglich
- 4200 Schliessen des File und Rücksprung
- 4210 - 4220 Löschen des geschriebenen Datums auf dem Bildschirm, wenn im Datum ein Fehler vorliegt
- 4230 DATA-Statements, grösst mögliche Anzahl Tage eines Monats

Dieser Programmteil ist vor allem für den Beginner interessant, da er Wege aufzeigt, eine Eingabe so zu prüfen, sodass praktisch keine Eingabefehler mehr möglich sind.

Das Programm der "Adressliste" zeigt gegenüber dem oben gezeigten Programm "Artikelliste" keinen Unterschied, es ist sogar noch ein wenig einfacher, da in Zeile 14100 der gesamte String ausgedruckt wird, ohne ihn vorher zu zerlegen, wie dies bei der Artikelliste erfolgt ist. Aus diesem Grund soll nur das Programm zum Ausdruck von Adressetiketten an dieser Stelle ausführlich behandelt werden.

Bevor man solche Etiketten drucken will, die im Endlosformat im Handel sind, muss man zuerst einmal feststellen, wie gross der Zeilenvorschub sein muss, damit eine Etikette gerade durchläuft, in unserem Fall benötigt die Etikette 13 Line Feed (zv=13). Das Etikettenprogramm wird über das Menu aufgerufen, und zuerst wird mittels der Suchroutine in Zeile 12000-12240 die richtige Adresse gesucht. Erst wenn dies der Fall ist, erfolgt der Sprung in die Druckroutine Zeile 30000-30390.

ADRESSLISTE ADRESSETIKETTEN

```

1000 REM VORBEREITUNG
1010 OPEN4,4:POKE 59468,14
1020 T$="***** DRESSAUSGABE ***"
1030 PRINTT$:A8$=" "
1040 PRINT"LEGEN SIE DIE DATENDISK IN DRIVE # 1 !"
1050 PRINT"ODENN FERTIG, DANN ↵ DRUECKEN !"
1060 GETZ$:IFZ$<>CHR$(13)THEN1060
1070 OPEN15,8,15:PRINT#15,"I0"
1080 GOSUB9000
2000 DIMG$(255):A8$=A8$+"":REM ADRESSABRUF
2010 A$="***** DRESSVERWALTUNG ***"
2020 PRINTA$
2030 PRINT"1. ADRESSENLISTE"
2040 PRINT"2. ADRESSAUFKLEBER"
2050 PRINT"3. ZURUECK INS MENU"
2060 GETZX$:IFVAL(ZX$)=0ORVAL(ZX$)>3THEN2060
2070 ONVAL(ZX$)GOTO2080,12000,3000
2080 GOSUB14000:GOTO10000
3000 CLOSE4:CLOSE15:PRINT"LOAD";CHR$(34);"0:MENU";CHR$(34);",8
3010 PRINT"RUNS";POKE158,2:POKE623,13:POKE624,13:END
8000 REM EINLESEROUTINE
8010 OPEN1,8,2,"0:"+STR$(FX$)+",S,R":GOSUB9000
8020 INPUT#1,G$(B%)
8030 IFLEFT$(G$(B%),1)=CHR$(10)THENG$(B%)=RIGHT$(G$(B%),LEN(G$(B%))-1)
8040 IFLEFT$(G$(B%),4)="END"THENCLOSE1:RETURN
8050 IFLEFT$(G$(B%),4)="EOF"THENCLOSE1:RETURN
8060 B%=B%+1
8070 GOT08020
8080 RETURN
9000 INPUT#15,EN$,EM$,ET$,ES$
9010 IFVAL(EN$)=0THENRETURN
9020 PRINT" ";EN$;" ";EM$;" ";ET$;" ";ES$
9030 CLOSE1:CLOSE15:END
10000 REM AUSGABE
10010 GOSUB8000
10020 B%=0
10030 IFLEFT$(G$(B%),4)="END"THENGOSUB16000:GOTO2010
10040 IFLEFT$(G$(B%),4)="EOF"THENFX=FX+1:B%=0:GOTO10010
10050 KL=0:GOSUB17000:GOSUB14100
10060 B%=B%+1
10070 GOT010030
12000 REM AUSGABE NACH NAMEN
12010 FX=0:B%=0
12020 PRINTT$
12030 PRINT"WOELCHER NAME : ";
12040 A=15:GOSUB20000:N$=X$
12050 GOSUB8000
12200 B%=0
12210 IFLEFT$(G$(B%),3)="END"THEN18000
12220 IFMID$(G$(B%),16,15)=N$THENKL=1:GOSUB17000
12230 IFLEFT$(G$(B%),3)="EOF"THENFX=FX+1:B%=0:GOTO12050
12240 B%=B%+1:GOTO12210
14000 REM ADRESSLISTENAUSDRUCK
14010 JJ=0:S=1:FORI=1TO3:PRINT#4:NEXT
14020 PRINT#4,TAB(7)CHR$(1)"*** DRESSLISTE ***"
14030 PRINT#4,TAB(7)CHR$(1)"-----"
14040 PRINT#4:ZZ=3
14050 PRINT#4,"EITE : ";S
14060 PRINT#4
14070 PRINT#4,"NR. XDRNAME NAME L RT ";
14080 PRINT#4,"TRASSE UND AUSNUMMER"
14090 FORI=1TO77:PRINT#4,"-";NEXT:PRINT#4,"-":PRINT#4:ZZ=ZZ+5:RETURN
14100 PRINT#4,G$(B%):ZZ=ZZ+1:JJ=JJ+1
14110 IFZZ<=65THENRETURN
14120 FORI=1TO6:PRINT#4:NEXT:ZZ=3
14130 S=S+1:GOSUB14050
14140 RETURN
16000 PRINT#4:PRINT#4,JJ;" ADRESSEN AUSGEGEBEN ";ZZ=ZZ+2
16010 FORI=1TO72-ZZ:PRINT#4:NEXT:RETURN
17000 DR=1:REM DRUCKROUTINE
17010 PRINTT$
17020 PRINT"NRREDE : ";LEFT$(G$(B%),5)
17030 PRINT"DRNAME : ";MID$(G$(B%),6,10)
17040 PRINT"NAME : ";MID$(G$(B%),16,15)
17050 PRINT" L : ";MID$(G$(B%),31,6)
17060 PRINT"RT : ";MID$(G$(B%),37,20)
17070 PRINT"TRASSE & R. : ";MID$(G$(B%),57,20)

```

```

17000 IFKL=0THENRETURN
17100 PRINT"#####,ST DIES DER GESUCHTE /AMEN ?"
17110 PRINT"#####<J/N>#"
17120 GETC$:IFC$=""THEN17120
17130 IFC$="J"THENGOSUB30000:GOTO2020
17140 IFC$="N"THENRETURN
17150 GOTO17120
17999 END
18000 IFDR=1THEN18030
18010 PRINT"#####/ICHT VORHANDEN"
18020 GETZ$:IFZ$<CHR$(13)THEN18020
18030 DR=0:GOTO2020
20000 X$="":AN=0:PRINT" _||";
20010 GETZZ$:IFZZ$=""THEN20010
20020 ZZ=ASC(ZZ$)
20030 IFZZ=20THEN20090
20040 IFZZ=13THEN20150
20050 X$=X$+ZZ$:AN=AN+1
20060 IFAN>ATHENPRINT" ";FORI=1TOAN:PRINT"|| ||";:NEXT:GOTO20000
20070 PRINTZZ$;" _||";
20080 GOTO20010
20090 IFAN=1THENX1$="":AN=0:GOTO20130
20100 IFAN<1THEN20010
20110 AN=AN-1
20120 X$=LEFT$(X$,AN)
20130 PRINT" ||L||";
20140 GOTO20010
20150 PRINT" "
20160 IFLEN(X$)<ATHENX$=X$+" ";GOTO20160
20170 RETURN
30000 REM DRUCKROUTINE
30010 A1=0:A2=0:A3=0:A4=0:A5=0:A6=0:A7$="-----"
30020 PRINT"#####IEVIELE TIKETTEN WOLLEN *IE : ";
30030 A=3:GOSUB20000:EZ=VAL(X$):ZZ=0
30040 A1$=LEFT$(G$(B%),5)
30050 IFA1$<>" "THENZZ=ZZ+1:A1=1
30060 A2$=MID$(G$(B%),6,10)
30070 IFA2$<>" "THENA2=1
30080 FORK=2TOLEN(A2$)
30090 B1$=MID$(A2$,K-1,1):B2$=MID$(A2$,K,1)
30100 IFASC(B1$)>ANDASC(B2$)=32THENA2$=LEFT$(A2$,K-1):GOTO30120
30110 NEXTK
30120 A3$=MID$(G$(B%),16,15)
30130 IFA3$<>" "THENZZ=ZZ+1:A3=1
30140 A4$=MID$(G$(B%),57,20)
30150 IFA4$<>" "THENZZ=ZZ+1:A4=1
30160 A5$=MID$(G$(B%),31,6)
30170 IFA5$<>" "THENA5=1
30180 A6$=MID$(G$(B%),37,20)
30190 IFA6$<>" "THENZZ=ZZ+1:A6=1
30200 AA=0:I=2
30210 IFMID$(A6$,I-1,1)=" "ANDMID$(A6$,I,1)=" "THENA7$=LEFT$(A7$,I-2):AA=1
30220 IFAA=1THENZZ=ZZ+1:GOTO30240
30230 IFI<21THENI=I+1:GOTO30210
30240 ZV=13-ZZ:Z1=INT(ZV/2):Z2=Z1:PRINTA8$;"#####,CH DRUCKE DIE ";
30250 IFZ1+Z2+ZZ<13THENZ1=Z1+1
30260 FORII=1TOEZ:PRINTTAB(19)"J";II;". TIKETTE"
30270 FORJ=1TOZ1-2:PRINT#4:NEXTJ
30280 IFA1=0THEN30300
30290 PRINT#4,A1$
30300 IFA2=0THEN30320
30310 PRINT#4,A2$+" "A3$
30320 IFA4=0THEN30340
30330 PRINT#4,A4$
30340 PRINT#4
30350 PRINT#4,A5$+" "+A6$
30360 PRINT#4,TAB(7)>A7$
30370 FORJ=1TOZ2:PRINT#4:NEXTJ
30380 NEXTII
30390 RETURN

```

- 30000 - 30010 Setzen von Variablen
- 30020 - 30030 Frage nach Anzahl auszudruckender Etiketten (ez)
- 30040 - 30050 Zerlegung des String, definieren der Anrede (a1\$), wenn die Anrede nur aus Space besteht, also nicht vorhanden ist, erfolgt Sprung zum Namen

- 30060 - 30070 Zerlegen des String und definieren des Namens.
- 30080 - 30110 Es wird festgestellt, wie lang der Namen ist, wobei ein einzelnes Space zwischen einem Doppelnamen als solches erkannt wird
- 30120 - 30130 Wie 30040, jedoch für den Vornamen
- 30140 - 30150 Wie 30040, jedoch für die Strasse und Hausnummer
- 30160 - 30170 Wie 30040, jedoch für die Postleitzahl
- 30180 - 30190 Wie 30040, jedoch für den Ort
- 30200 - 30230 Es wird festgestellt, wie lang der Namen des Ortes ist, wobei auch hier Doppelnamen, die über Space getrennt sind erkannt werden, es wird a7\$ definiert, dies ist die Linie zum Unterstreichen des Ortsnamens.
- 30240 - 30260 Definieren von Variablen und Angabe der Anzahl noch zu druckender Etiketten, z1 und z2 sind die Anzahl der Leerzeilen vor und nach dem Etikettentext.
- 30260 FOR .. NEXT - Schlaufe für die Etikettenzahl
- 30270 FOR .. NEXT - Schlaufe für die Leerzeilen vor dem Etikettentext
- 30280 - 30360 Ausdruck des Etikettentextes, wenn eine Textzeile (z.B. rede) fehlt, wird die Zeile überschlagen
- 30370 FOR .. NEXT - Schlaufe für die Leerzeilen nach dem Etikettentext
- 30380 - 30390 NEXT für Anzahl Etiketten und Rücksprung

Hr.
 Felix Muster
 Schildbuerdenstr. 15

9999 Holzhausen

Musteretikette

*** A d r e s s l i s t e ***

Seite : 1

Ann.	Vorname	Name	PLZ	Ort	Strasse und Hausnummer
Hr.	Felix	Muster	9999	Holzhausen	Schildbuerdenstr. 15
Fr.	Annelise	Meier	D 1111	Musterstadt	Eulenspiegelallee 7
Firma	Gottfried	Stutz	A 7000	Eisenstadt	Maschinenfabrik

3 Adressen ausgegeben

Probeausdruck Adressliste

Hardware

BARCODE — PACK

Nachdem nun in diesem Heft die Folge über den Barcode abgeschlossen ist, können wir Ihnen nicht nur die erforderliche Software, sondern auch die entsprechende Hardware anbieten, die Ihnen einen komfortablen Umgang mit diesem interessanten Datenkommunikationssystem gewährleistet.

Nachdem wir festgestellt haben, dass die Nachfrage nach einzelnen Interface oder Lesestiften sehr klein war, haben wir uns entschlossen, die Hardware und Software nur als komplette Pack zu verkaufen.

Dieses Pack enthaelt :

1. Lesestift zum Einlesen jedes beliebigen Barcodes
2. EPROM mit dem Maschinenprogramm zum Einlesen des CBM-Barcodes
3. Disk oder Kassette mit folgenden Programmen
 - a. Einleseprogramm für mc-Barcode
 - b. Ausgabeprogramm für mc-Barcode
 - c. Ausgabeprogramm für CBM-Barcode

Bei der Bestellung müssen folgende Detail angegeben werden :

1. Rechnertyp
2. Disk oder Kassette
3. Floppy Disk Typ

Trotz der guten Nachfrage nach unserer Hardware ist es nur schwer möglich, die vergleichbar geringen Stückzahlen zu einem niedrigen Preis zu produzieren, vor allem wenn wir auf Zulieferanten angewiesen sind. Wir möchten Sie daher bitten für unseren neuen Subskriptionsbestellmodus Verstaendnis zu zeigen. Bestellen Sie die unten genannte Hardware bis spaetestens 2.10.1982 ausschliesslich beim Mikro + Kleincomputer Informa Verlag AG Postfach 1401 6000 Luzern.

A c h t u n g ! ! ! Bestellungen an den Schweizer Computer Club können nicht mehr berücksichtigt werden. Die Auslieferung der gesamten Pack erfolgt ca. 15.10.1982. Nur so ist es uns möglich, grössere Stückzahlen auf einmal zu produzieren und den Preis in einem annehmbaren Rahmen zu halten.

Sie erhalten das Barcodepack zu Fr. 345.00 incl. Lesestift. Eine Lieferung der Software allein, also nur die Programmdisk oder Kassette plus EPROM ist zum Preis von Fr. 145.00 lieferbar.

Dem Pack liegt als Sonderdruck die Bedienungsanleitung und die Einbauanleitung für das EPROM bei.

Programm des Monats BASIC-BASIC

PROGRAMM — VERGLEICHER

Ist es Ihnen auch schon einmal passiert, dass Sie ein Programm veraendert haben und Sie nach einiger Zeit nicht mehr wussten, welche Aendernngen Sie gegenueber dem Original vorgenommen haben. Nun hier schafft das Programm "Vergleicher" Abhilfe. Es ist in BASIC 4.0 fuer den CBM 8000 und das Floppy 8050 geschrieben und basiert auf dem amerikanischen "Compare". Es vergleicht zwei beliebige Programme, die auf Disk abgespeichert sind, blockweise miteinander, dadurch koennen auch Programme, die sehr gross sind, verglichen werden. Das Programm selbst belegt 6.1 kByte. Die Ausgabe der unterschiedlichen Zeilen erfolgt wahlweise auf dem Bildschirm oder einem Drucker. Die Ausgabe kann auf jedem Drucker vorgenommen werden, da keine druckerspezifischen Befehle verwendet worden sind.

Diejenigen Leser, denen das Abschreiben des Programms zu viel Arbeit macht, koennen das Programm gegen Einsendung einer leeren Kasette oder Disk und Fr. 15.-- sowie einem frankierten Rueckantwortumschlag auch beim Ersteller fertig beziehen.

PROGRAMM — VERGLEICHER

```
1 clr:pg$="vergleich":nm$="22":goto62000 : rem by bruesser
3 print"J","*** J. Bruesser Elektronik  vergleicher  ***"
100 pn=0:df=4:pd=4:dimeof(3):nu$=chr$(0):rem df,pd druckerfile,printerdevice
110 v$=chr$(1)+chr$(4):bl$=" "
120 hy$="-----":hy$=hy$+hy$
130 ol=-1
140 dimtk$(127)
150 fori=0to127:readtk$(i):nexti
155 gett$:ift$=""then155
157 print"J";
160 print"Haben Sie einen Drucker (j/n) ? ";
165 gosub60000:pr=(left$(in$,1)="j")
170 open15,8,15
180 print" Welches Programm  vergleichen.... ";:gosub60000
190 ifin$=""then660
195 of$=in$
200 print"in welchem Drive (0/1) gespeichert ? ";:gosub60000
205 a$=in$:open1,8,2,+a$+": "+of$+",p"
210 gosub1070:iferthenclose1:goto180
220 n=1:gosub920:t$=nm$
230 ift$<>v$thenprint" ";of$;" ist kein Basic-File.":close1:goto180
240 print" Mit welchem anderen Programm ? ";:gosub60000
245 nf$=in$
250 print"in welchem Drive (0/1) gespeichert ? ";:gosub60000
255 b$=in$:open2,8,3,+b$+": "+nf$+",p"
260 gosub1070:iferthenclose2:goto240
270 n=2:gosub920:u$=nm$
280 ifu$<>v$thenprint" ";nf$;" ist kein Basic-File.":close2:goto240
290 iferthenopendf,pd:goto310
300 opendf,3:pd=3
310 n=1:gosub860:l1=ln:s1$=in$
320 n=2:gosub860:l2=ln:s2$=in$
330 printl1;"|l1| ";l2;"|l2|":IFPRTHEN380
340 sett$:ift$=""then380
350 printtab(len(bl$));" @druecke eine Taste@"
360 sett$:ift$=""then360
370 printtab(len(bl$));bl$;"@
380 on2+sgn(l1-l2)goto400,450,570
400 tt$="-"
410 ln=l1:s$=s1$:gosub740
420 n=1:gosub860:ifeofthen620
```

```

430 l1=ln:s1$=in$:goto330
450 ifs1$=s2$then500
460 tt$="-"
470 ln=l1:s$=s1$:gosub740
480 tt$="+"
490 ln=l2:s$=s2$:gosub740
500 d=0
510 n=1:gosub860:ifeofthend=d+1
520 l1=ln:s1$=in$
530 n=2:gosub860:ifeofthend=d+2
540 l2=ln:s2$=in$
550 ond+1goto330,620,610,660
570 tt$="+"
580 ln=l2:s$=s2$:gosub740
590 n=2:gosub860:ifeofthen610
600 l2=ln:s2$=in$:goto330
610 n=1:in$=s1$:ln=l1:goto630
620 n=2:in$=s2$:ln=l2:goto630
630 tt$=mid$("-+",n,1)
640 s$=in$:gosub740
650 gosub860:ifeof=0then640
660 print
670 ifol<0thenprinttab(25);"Beide Programme genau gleich"
680 ifol>=0thenprint#df,hy$
685 ifol>=0thenprint#df,"- ";of$;" in Drive ";a$;bl$;" + ";nf$;" in Drive ";b$
690 closedf
700 close2
710 close1
720 close15
730 goto62500
740 ifol<0thengosub60210
760 ifol>0andol<>lnthenprint#df,bl$:lz=lz+1:gosub60200
770 ol=ln:qt=0:print#df,tt$:ln;:fori=1tolen(s$)
780 t=asc(mid$(s$,i,1))
790 ift=34thenqt=1-qt
800 ift<128orqtthenprint#df,chr$(t);:goto830
810 ift>127thenprint#df,tk$(t-128);
820 ift=131ort=143thenprint#df,mid$(s$,i+1);:goto840
830 nexti
840 ifrr=0thenifpos(0)<len(bl$)thenprint#df,bl$;
850 print#df:lz=lz+1:gosub60200:return
860 gosub960:t$=c$:gosub960:lk=asc(t$)+asc(c$)*256
870 iflk=0theneof=1:eof(n)=eof:return
880 gosub960:t$=c$:gosub960
890 in$=t$+c$:ln=asc(t$)+256*asc(c$):in$=""
900 gosub960:ifc$<>nu$thenin$=in$+c$:goto900
910 return
920 gosub960:ifeofthennm=-1:nm$="":return
930 nm$=c$:gosub960
940 nm$=asc(nm$)+asc(c$)*256:nm$=nm$+c$
950 return
960 eof=eof(n):ifeof=0thenset#n,c$:ifsttheneof=1:eof(n)=eof
970 ifeoforc$=""thenc$=nu$
980 return
990 dataend,for,next,data,input#,input,dim,read
1000 datalet,goto,run,if,restore,gosub,return,rem
1010 datastop,on,wait,load,save,verify,def,poke
1020 dataprint#,print,cont,list,clr,cmd,sys,open
1030 dataclose,set,new,tab(,to,fn,spc(,then
1040 datanot,step,+,-,*,/,f,and,or,>,<,sgn,int,abs,usr
1045 datapos,fre,sqr,rnd,log,exp,cos,sin
1050 datatan,atan,peek,len,str$,val,asc,chr$,left$,right$,mid$,go
1055 dataconcat,dopen,dclose,record,header,collect,backup,copy,append,dsave
1060 datadload,catalog,rename,scratch,directory,,c220,c221,c222,c223
1065 datac224,c225,c226,c227,c228,c229,c230,c231,c232,c233
1066 datac234,c235,c236,c237,c238,c239,c240,c241,c242
1067 datac243,c244,c245,c246,c247,c248,c249,c250,c251
1068 datac252,c253,c254,phi
1070 input#15,er,er$,tk,sc
1080 ifer<=1thenreturn
1090 ifer=62thener$="kein solches File auf der Diskette"
1100 print"@";er$
1110 return
60000 in$=" ";zt=ti:zc=2:zd$=chr$(20)
60010 getz$:ifz$<>""then60070
60020 ifzt<=tithenprintmid$(" ",zc,1);"||";:zc=3-zc:zt=ti+15
60030 goto60010
60070 z=asc(z$):zl=len(in$):if(zand127)<32thenprint" ||";:goto60110

```

```

60080 iffland(zand127)>64and(zand127)<91thenz$=chr$((z+128)and255)
60090 ifzl>254then60010
60100 in$=in$+z$:printz$;zd$;z$;
60110 ifz=13thenin$=mid$(in$,2):printcr$;:return
60120 ifz=20andzl>1thenin$=left$(in$,zl-1):print"||";:goto60010
60130 ifz=141thenz$=chr$(-20*(zl>1)):forz=2tozl:printz$;:nextz:goto60000
60140 goto60010
60200 iflz<60thenreturn
60203 ifpd<4thenreturn
60205 fora=lzto72:print#df:nexta
60210 ifpd<4then60230
60212 pn=pn+1:print#df,"Programmvergleich Glen Fisher & Josef Bruesser ";
60215 print#df,"Elektronik (c) 1982 Seite";:pn
60220 fora=1to20:print#df,"****";:nexta:print#df
60230 print#df,bl$
60240 print#df,"- ";of$;" in Drive ";a$;bl$;" + ";nf$;" in Drive ";b$
60250 print#df,hy$:poke224,10:lz=6
60270 return
60300 clr:gosub60400:goto100
60400 qk=525:q1=245:qm=134:qp=515:qq=234:qs=516:qv=1:cr$=chr$(13)
60410 ifpeek(50000)=0thenreturn
60420 qk=158:q1=216:qm=52:qp=151:qq=205:qs=152:qv=2
60430 return
60500 fora=1to20:print"****";:nexta:return
62000 print"###";:gosub60500:gosub60500
62003 print"***";tab(31-len(pg$));"@ ";pg$;"      Version CBM - Basic 4.0 ";
62005 printtab(78);"***";
62006 print"***";tab(20);"                                von Glen Fisher";tab(78);"***";
62007 print"***";tab(20);"      Basic 4.0 und Deutsch: Josef Bruesser";tab(78);"***";
62008 print"***";tab(20);"      Muehlehof CH6038 Gisikon 041/91 18 67";tab(78);"***";
62010 print"***";tab(20);"                                Copyright (c) 1982";tab(78);"***";
62020 gosub60500:gosub60500:closedf
62030 printtab(20);"###Auffinden von Unterschieden in Programmen"
62035 printtab(15);"### Sie koennen zwei Programme in Drive 1,0 oder 1 & 0"
62040 printtab(15);"Zeile um Zeile vergleichen und die unterschiedlichen"
62050 printtab(15);"      Zeilen auf Drucker oder Bildschirm ausgeben."
62135 printtab(20);"### Diskette(n) mit zu vergleichenden"
62140 printtab(20);"      Programmen einlesen."
62150 printtab(20);"###      Druecke eine Taste zum Besinnen"
62170 goto60300
62500 ifpd>3thenopendf,pd:fora=lzto70:print#df:nexta:closedf
62505 poke224,0:print"###      weiterer Vergleich      j/n ? ";
62510 gosub60000:ifin$="j"thanrun
62515 ifin$<>"n"then62510

```

ready.

Programmbeschreibung

Zeile	1 -	165	Definition von Variablen und Dimensionierung
Zeile	170		Fehlerkanal eroeffnen
Zeile	180 -	210	Programmname und Drive einlesen und positionieren
Zeile	220 -	230	Kontrolle ob es sich um ein BASIC-File handelt
Zeile	240 -	260	Vergleichprogrammname und Drive einlesen
Zeile	270 -	280	Kontrolle ob es sich um ein BASIC-File handelt
Zeile	290 -	300	Output auf Drucker oder Bildschirm
Zeile	310 -	320	Einlesen aus Programm 1 und 2
Zeile	330		Zeilennummer von Prg. 1 und 2 auf dem Bildschirm
Zeile	340 -	380	GET-Routine Programmzeilennummern, Sprungverteiler
Zeile	400 -	450	Zeile des 1. Programms drucken, neue Zeile einlesen und naechste Zeile einlesen, beide Zeilen vergleichen
Zeile	460 -	550	Zeile Programm 1 und 2 drucken, Verteiler fuer EOF und Verteiler fuer Ende des Programms
Zeile	570 -	620	Gleiche Routine wie 400 - 450 fuer Zeile des 2. Prg.
Zeile	630		Wahl von tt\$
Zeile	640		Zeile drucken
Zeile	650		Zeile einlesen zurueck zum Vergleich
Zeile	660		Ende des Vergleichs
Zeile	670 -	720	Abschluss drucken

Zeile	730	Papiervorschub
Zeile	740	Titel drucken
Zeile	760	Seitenlaenge kontrollieren
Zeile	770	Zeile drucken
Zeile	780	Interpretercode
Zeile	790	Behandlung von "
Zeile	800 - 820	ASCII-C., Interpretercode u. Editorzeichen drucken
Zeile	840 - 850	Zeilennummer löschen und Papiervorschubkontrolle
Zeile	860 - 910	Einlesen aus Floppy-Buffer
Zeile	920 - 950	Einlesen und Kontrolle des BASIC-Files
Zeile	960 - 980	Ein Zeichen ab Floppy-Buffer einlesen
Zeile	990 - 1068	DATA-Statements des Interpretercodes
Zeile	1070 - 1100	Auswertung der Fehlermeldung
Zeile	60000 - 60140	GET-Routine
Zeile	60200 - 60210	Seitenlaenge kontrollieren, Kontrolle Drucker- ausgabe, Seitenvorschub, Seitenzaehler
Zeile	60210 - 60300	Programmkopf drucken, unterstreichen, Programmbe- ginn
Zeile	60400 - 60430	Betriebssystemwahl (nicht benutzt)
Zeile	60500 - 62170	Programmerklaerung auf dem Bildschirm
Zeile	62500 - 62515	Restart und Ende

WUSSTEN SIE SCHON

dass Scrolling des Bildschirms auch beim VC-20 möglich ist. Die Speicherstelle \$ 9000 (36864) enthaelt die Adresse des horizontalen Bildschirmausschnittes, die Speicherstelle \$ 9001 (36865) den vertikalen Ausschnitt. Das nachfolgende kleine Programm bewegt das Textfenster von der Mitte nach links (Zeile 10) und danach von links nach rechts (Zeile 40) und zuletzt von rechts wieder in die Mitte (Zeile 70). In den Zeilen 100, 130 und 160 erfolgt die Verschiebung aus der Mitte nach oben, von oben nach unten und wieder zurück in die Mitte. Wird versucht, das Textfenster ganz aus dem Bildschirm zu schieben, erlebt man eine grosse Enttaeuschung, da es zu erheblichen Verzerrungen kommt.

```

10 for a=12 to 8 step -1: poke 36864, a
20 x=150: gosub 200
30 next
40 for a=8 to 16: poke 36864, a
50 x=150: gosub 200
60 next
70 for a=16 to 12 step -1: poke 36864, a
80 x=150: gosub 200
90 next
100 for a=38 to 15 step -1: poke 36865, a
110 x=50: gosub 200
120 next
130 for a=15 to 61: poke 36865, a
140 x=50: gosub 200
150 next
160 for a=61 to 38 step -1: poke 36865, a
170 gosub 200
180 x=50: gosub 200
200 for b=1 to x: next: return

```

Maschinensprache

BARCODE AUSDRUCK

Im Heft 2/82 stellten wir Ihnen ein Programm vor, mit dem Sie die BASIC Programme als Barcode im mc-Format ausdrucken konnten.

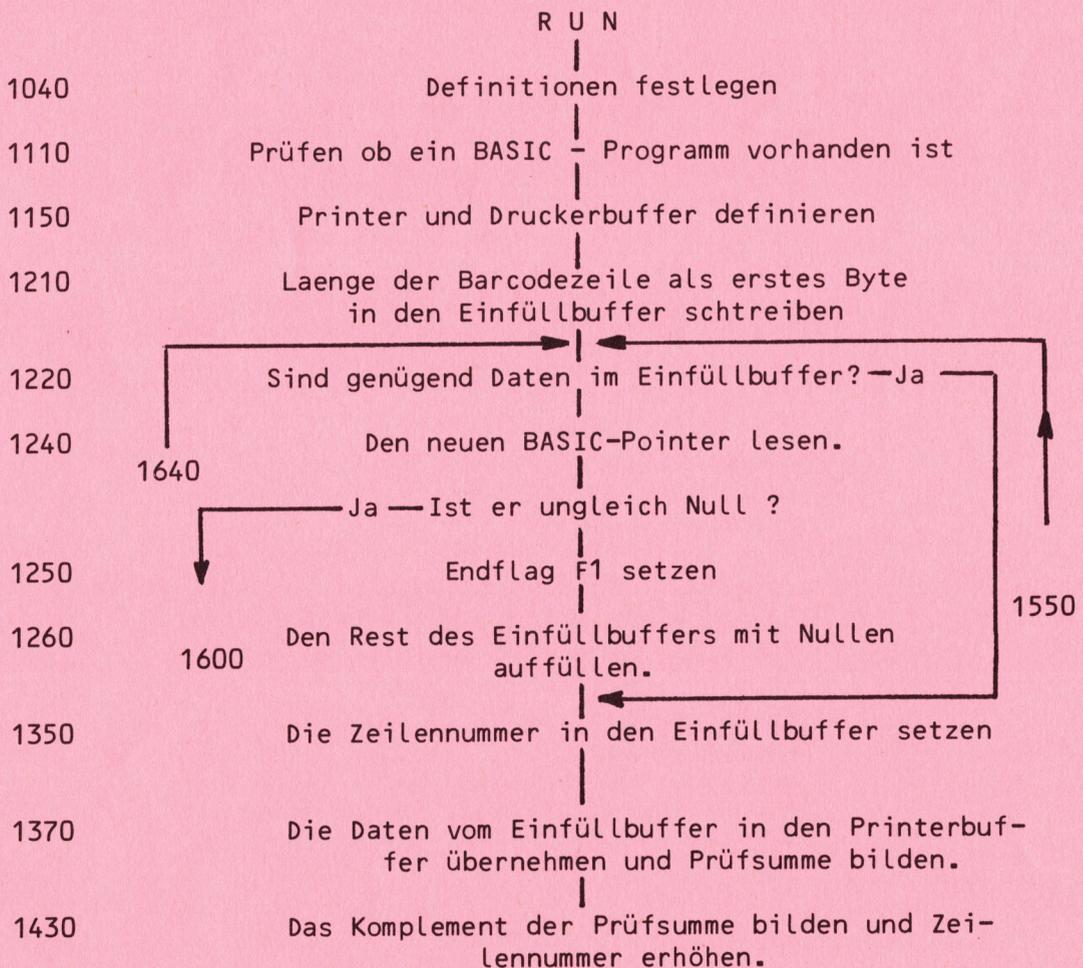
Heute zeigen wir Ihnen einen allgemeinen Weg, wie Barcodelistings direkt aus dem Speicher des CBM-Computer ausgedruckt werden und zwar nach dem neuen CBM-Format (siehe hierzu Heft 3/82).

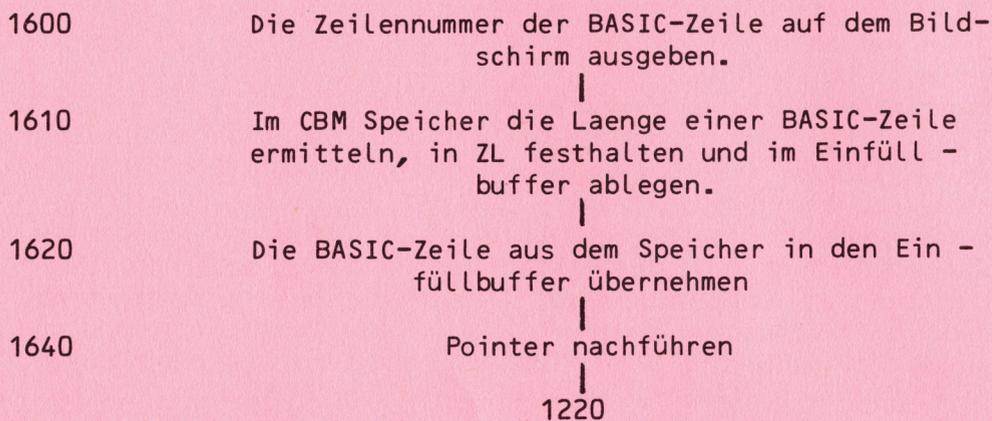
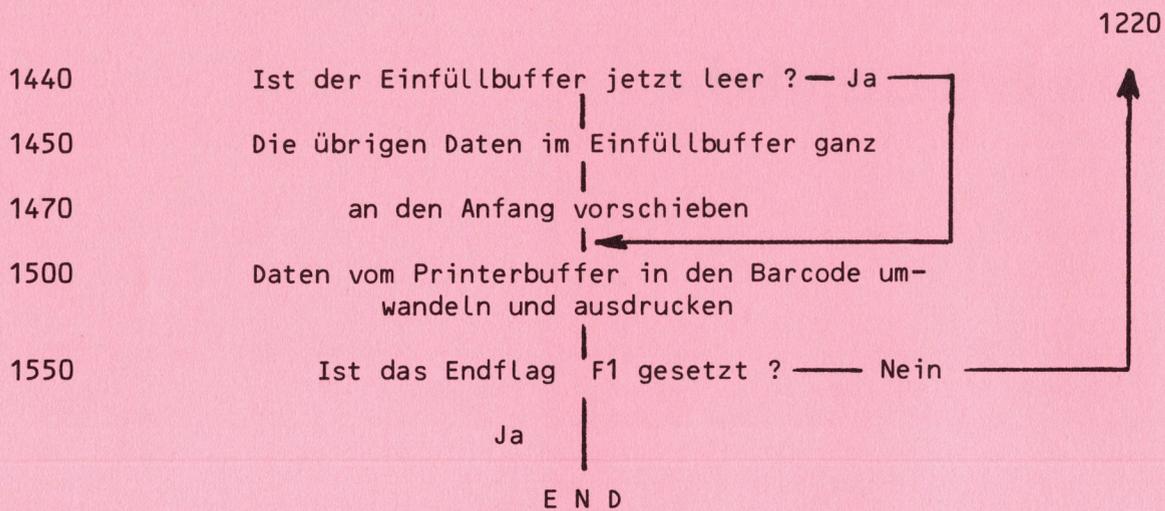
Der Autor benutzte bei der Entwicklung einen EPSON MX 80 II Drucker, an einem CBM 4032. Es stand kurzzeitig auch ein SEIKOSHA GP-80 Drucker zur Verfügung. Als Demonstrationsmaterial haben wir die entsprechenden Listings beigelegt.

Einige Bemerkungen zum neuen Druckerprogramm:

Es ist so konzipiert, dass einmal geladene BASIC-Programme direkt aus dem CBM RAM ausgedruckt werden, das Druckerprogramm kann sich also auch selber ausdrucken. Dies bedeutet aber, dass das Druckerprogramm, als auch das auszudruckende Programm gleichzeitig im Speicher liegen müssen. Methode und Anwendung werden erklärt. Das Druckerprogramm ist in BASIC geschrieben und in seinem Ablauf so einfach, dass eine Änderung des Programms zur Ansteuerung anderer Drucker ohne Schwierigkeiten möglich ist. Jeder hat dadurch die Möglichkeit selbst zu experimentieren.

Flowchart zum neuen Printerprogramm





Variablenliste

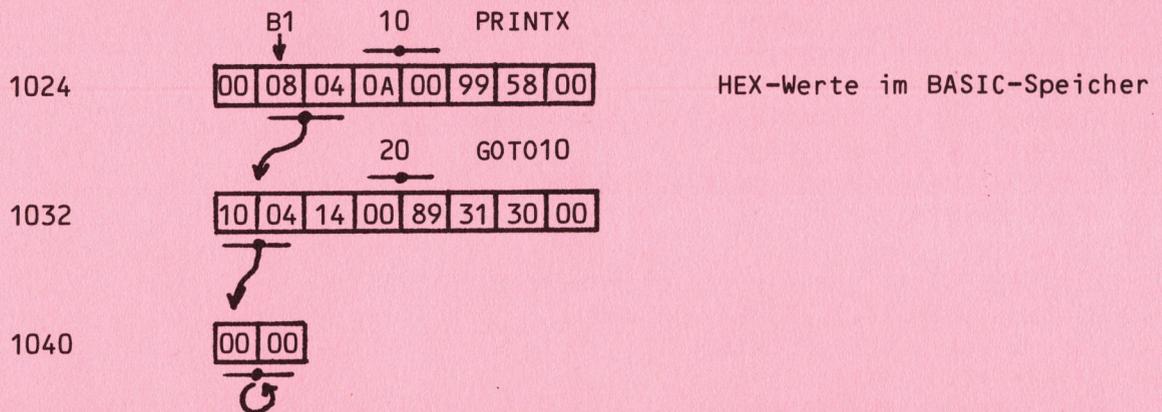
ES\$, LF\$, FF\$, DM\$	Druckerspezifische Steuerwerte
SP\$	einfaches Leerzeichen in der Barcodezeile
SB\$	einfacher Strich in der Barcodezeile
LE=18	gesamte Bytes pro Barcodezeile
B1 = 1025	zeigt auf den Anfang eines gespeicherten BASIC Programms
P\$(..)	alle Printernadeln aus SP\$ und SB\$
PO	Anzahl Bytes ohne Zeichenzähler u. Prüfsumme
E1 = 0	Zeichenzähler im Einfüllbuffer
CN = 1	Barcodezeilennummer
PB = 826	Printbuffer Startadresse
EB	Einfüllbuffer Startadresse
PT	temporaerer Zeiger von PB
ET	temporaerer Zeiger von EB
F1 = 0	Flag für letzte BASIC-Zeilenerkennung
B2	zeigt auf den Beginn einer neuen BASIC-Zeile
ZL	Laenge einer gespeicherten BASIC-Zeile
CS	Prüfsumme einer Barcodezeile
C	Zwischenspeicher eines Bytes
TT	Auslesezeiger vom Einfüllbuffer
I, J, II	Laufvariablen
N	Laufindex der belegten Nadelreihen
A	In Barcode zu verwandelnden Bytewert
IN	Zähler zum Regulieren der Balkendicke

Datenfluss und Organisation im Printerprogramm

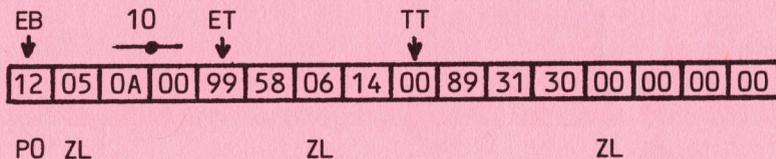
Die genaue Ablage eines BASIC-Programms im Speicher kann aus dem Commodore Bedienungshandbuch entnommen werden.

Als Beispiel sei das folgende kleine Programm dienen :

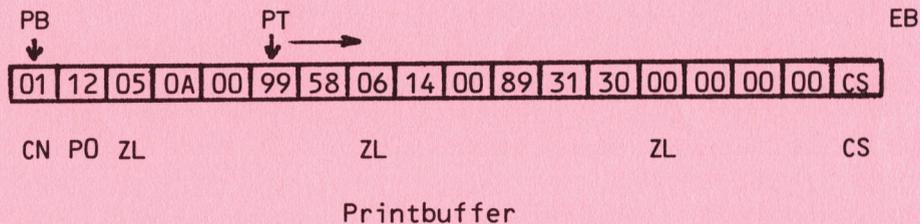
```
10 PRINTX
20 GOTO10
```



Diese Daten werden konvertiert und in den Einfüllbuffer überführt :



Danach wird von den Daten im Einfüllbuffer die Prüfsumme gebildet. Im Printerbuffer wird die Zeilennummer gesetzt, die Daten aus dem Einfüllbuffer übernommen und hinten angehängt. Danach wird der komplette Printbuffer byteweise in den Barcode übersetzt und ausgedruckt. Danach wird wieder BASIC gelesen, konvertiert usw.



```
1000 REM -----
1005 REM * * BARCODE COMPACT * *
1010 REM * PRINTER EPSON MX82 *
1020 REM * PROGRAMMING BY EDWIN BREU *
1025 REM -----
1030 PRINTCHR$(147):REM 'CLEAR SCREEN'
1040 ES$=CHR$(27):DM$="L"
1050 SP$=CHR$(0):SB$=CHR$(255)
1100 LE=18:B1=1+4*256:DIM P$(58*(LE+1))
1110 IF (PEEK(B1)ORPEEK(B1+1))<>0THEN1150
1120 PRINT:PRINT"NO PROGRAM FOR PRINTOUT!":GOTO1580
1150 PO=LE-2:E1=0:CN=1:PB=826:EB=PB+LE:PT=PB:ET=EB:F1=0
1160 KE=2:ZV=2:ZZ=11:VE=ZZ*72:NA=6*(ZV*(KE-1)+12):NV=NA+ZV
1200 OPEN4,4:PRINT#4,ES$;CHR$(65);CHR$(ZV)
```

```

1210 POKEET,LE:ET=ET+1:E1=E1+1
1220 IFE1>=POTHEN1350
1240 B2=PEEK(B1)+256*PEEK(B1+1):IFB2<>0THEN1600
1250 F1=1:FORI=1TO(P0-E1)
1260 POKEET,0:ET=ET+1:NEXTI:E1=P0
1350 POKEPB,CN:CS=CN:PT=PB+1:TT=EB
1370 FORI=1TOP0:C=PEEK(TT):POKEPT,C:CS=CS+C
1390 TT=TT+1:PT=PT+1:NEXTI
1400 IFCS>255THENCN=CS-256:GOTO1400
1420 IFCS=0THENCN=256
1430 POKEPT,(256-CS):CN=CN+1:IFCN=256THENCN=0
1440 ET=EB:E1=E1-P0:IFE1=0THEN1500
1450 FORI=1TOE1:POKEET,PEEK(TT)
1470 ET=ET+1:TT=TT+1:NEXTI
1500 GOSUB5030
1520 FORJ=0TO(LE-1):A=PEEK(PB+J):GOSUB6030:NEXTJ
1550 GOSUB3030:IFF1=0THEN1220
1555 PRINT#4,ES$:CHR$(65);CHR$(12):CLOSE4
1560 PRINT:PRINT"FERTIG!"
1580 PRINT:END
1600 B1=B1+2:PRINT"  B# =";PEEK(B1)+256*PEEK(B1+1)
1610 ZL=B2-B1:POKEET,ZL:ET=ET+1:E1=E1+1
1620 FORI=2TOZL:POKEET,PEEK(B1)
1640 ET=ET+1:B1=B1+1:E1=E1+1
1660 NEXTI:B1=B1+1:GOTO1220
3000 REM *** PRINTOUT
3030 N=N-1:HB=INT(N/256):LB=N-256*HB
3040 PRINT"          N =";N:FORK=1TOKE
3050 PRINT#4,ES$:DM$:CHR$(LB);CHR$(HB);
3080 FORI=1TON:PRINT#4,P$(I);:NEXTI
3085 PRINT#4:NEXTK
3090 FORK=2TO(12/ZV):PRINT#4:NEXTK
3100 NV=N+ZV*(KE-1)+12:IFNV<VETHENRETURN
3120 NV=NA+VE-NV-48
3130 PRINT#4,ES$:CHR$(65);CHR$(48)
3140 PRINT#4,ES$:CHR$(65);CHR$(NV)
3150 PRINT#4,ES$:CHR$(65);CHR$(ZV):NV=NA+ZV:RETURN
5000 REM *** START CHARACTER
5030 P$(1)=SB$:P$(2)=SB$:P$(3)=SB$:P$(4)=SP$:P$(5)=SP$
5040 P$(6)=SP$:P$(7)=SP$:P$(8)=SP$:P$(9)=SB$:N=10:RETURN
6000 REM *** KONVERT BYTE
6030 Z=128:FORI=0TO7:IFINT(A/Z)=0THENIN=1:GOTO6070
6060 IN=4:A=A-Z
6070 Z=Z/2:P$(N)=SP$:P$(N+1)=SP$:P$(N+2)=SP$
6075 P$(N+3)=SP$:P$(N+4)=SP$:N=N+5
6080 FORII=1TOIN:P$(N)=SB$:N=N+1:NEXtii
6090 NEXTI:RETURN
READY.

```

Wie wird das Druckprogramm geladen und benutzt ?

Damit ein normales BASIC-Programm aus dem CBM-Speicher ausgedruckt werden kann, muss das Druckerprogramm zwangsläufig viel weiter oben im Speicher liegen. Teilen wir den Speicher auf : 28 K für BASIC, 4 K für das Druckerprogramm. Sie verschieben also den Betriebssystempointer "Begin of BASIC" nach Tabelle 1 und tippen dann das Printerprogramm ein. Nach Abschluss wird es normal auf Diskette oder Kassette abgespeichert. Später beim Wiederladen wird es automatisch wieder an diesen Platz gesetzt, wobei vor dem Laden die Schritte der Tabelle 1 auszuführen sind.

CBM Rechner

Schritte	3008 4008	3016 4016	3032 4032 8032
1.	POKE 40,1	POKE 40,1	POKE 40,1
2.	POKE 41,16	POKE 41,48	POKE 41,112
3.	POKE 4096,0	POKE 12288,0	POKE 28672,0
4.	NEW	NEW	NEW

Tabelle 1

Sobald das Printerprogramm korrekt eingegeben und abgespeichert worden ist, muss wieder auf den Originalplatz zurückgeschaltet werden. Dieser Schritt ist für alle Rechner gleich :

1. POKE 40,1
2. POKE 41,4
3. POKE 1024,0
4. NEW

Jetzt kann das auszudruckende BASIC-Programm eingetippt oder ab Massenspeicher geladen werden. Danach wird wieder nach Tabelle 1 umgeschaltet und das Printerprogramm ab Massenspeicher geladen. Dieses kann nun mit RUN gestartet werden. Dieses dauernde Umschalten gilt für jedes auszudruckende BASIC-Programm.

Damit ist unser Beitrag über den Barcode beendet, in den weiteren Hefen werden wir die Programme auch im Barcodelisting abdrucken. Zur Hardware nehmen wir unter der Rubrik "Hardware" Stellung. Sollten Sie in der Anwendung Probleme haben, oder sonstige Fragen zum Barcode, so schreiben Sie an die Redaktion der CBM/PET News. Wir helfen Ihnen dann gerne weiter.

Leserbriefe

Verschiedene unserer Leser haben Probleme mit den Modulen des VC-20, vor allem mit dem Super Expansion VC 1211A (Hires Graphic) und dem Programmers Aid VC 1212. Die Fehler treten vor allem beim gleichzeitigen Betrieb der Module mit einer 8 oder 16 k Speichererweiterung auf. Wer hat hier Erfahrungen ? Gerne beschreiben wir dann die auftretenden Fehler etwas genauer.

Ein anderer Leser würde es begrüßen, wenn wir in einer der nächsten Ausgaben folgende Themenkreise behandeln würden.

1. Programmierung von Sonderzeichen auf dem CBM ohne Aenderung der Hardware, also kein neues ROM.
2. Programmierung eigener Befehle beim CBM, wie dies beim VC-20 möglich ist.

Sicher hat jemand Erfahrungen mit diesen Themen und kann Auskunft darüber erteilen, in wie weit diese Wünsche realisierbar sind.

ZEICHEN - ROM ZUM CBM 8000

In den vergangenen 2 Monaten haben uns recht viele Leserbriefe mit der Frage erreicht, ob es nicht möglich sei, das Programm zur Gestaltung eines neuen Zeichen ROM auch für den CBM 8000 zu machen. Nun hier ist das umgeschriebene Programm.

Es funktioniert genau gleich, wie dies beim CBM 3000 der Fall war, darüberhinaus können wir Ihnen mitteilen, dass der Inhalt des Charaktergenerators beim CBM 8000 und CBM 3000 genau gleich ist, Sie können also auch den Inhalt der Adressen \$2000 - 2800 übernehmen.

Drei Dinge müssen Sie allerdings beim Gebrauch des Programms auf dem CBM 8000 beachten.

1. Der Charaktergenerator im CBM 8000 sitzt an einem anderen Ort, nämlich in der Fassung UA 3 und ist bei den meisten neueren Geräeten eingelötet. Wenn Sie nicht ganz sicher sind, gehen Sie in Ihre Vertragswerkstatt und lassen Sie sich das ROM in eine Fassung setzen.
2. Die Zeichen sind anders codiert als bei CBM 8000. Sie finden diesen Code aber sehr leicht heraus, wenn Sie die Zeichen einfach einmal alle nacheinander aufrufen.
3. Wenn Sie die Zeichen nicht auf Kassette abspeichern wollen, sondern auf Disk, benutzen Sie statt der Geräteadresse 01, die Adresse 08.

Für den CBM 4000 können Sie das gleiche Programm benutzen, wie beim CBM 3000, jedoch gelten auch hier die Punkte 1 - 3 der obigen Ausführungen.

ZEICHEN - ROM ZUM CBM 8000

```
100 rem neues zeichenrom fuer cbm 8000
110 poke59468,12
120 print"000000";tab(30)chr$(176);:fori=1to8:printchr$(177);:next
130 printchr$(174):fori=1to8:printtab(26)i" chr$(179)"      ";
140 printchr$(171):next
150 printtab(30)chr$(173);:fori=1to8:printchr$(178);:next
160 printchr$(189)
170 printtab(20)"00'a' am ende der zeichen-nr.: aenderung
180 printtab(20)"zeichnen mit 'Q' und dann 'return'"
190 printtab(20)"0'space' nach 'return': vorwaertszaehlen"
200 printtab(20)"0zeichen im ram: ab 8192 ($2000)";
210 t=8192;z=32:p=81
220 ifpeek(151)<6thenpoke158,0:goto240
230 r=r+1:goto250
240 print"0"tab(25);:input"zeichen-nr.":r$:print"0"tab(25)"      ":r=val(r$)
250 print"0000"tab(32)"      ":print"0000"tab(33)r:ifright$(r$,1)="a"then300
260 s=t+r*8:b=33286:ford=0to7:a=peek(s+d)
270 fori=1to8:gosub280:next:b=b+88:next:h=0:goto220
280 a=a/2:ifa=int(a)thenpokeb-h,z:h=h+1:return
290 a=int(a):pokeb-h,p:h=h+1:return
300 print"000000"
310 fork=0to7:f$="":print:printtab(31);:open1,0:input#1,z$:close1
320 iflen(z$)<9thenfori=1to9-len(z$):f$=f$+" ":next:z$=f$+z$
330 g=1:j=0:fori=8to1step-1:ifmid$(z$,i,1)="Q"thenj=j+g
340 g=g+g:next:poke1+r*8+k,j:next:goto220
ready.
```

LEICHTE KOST

Der Zweitwagen ist für viele Wohlstandsbürger eine Selbstverständlichkeit, warum also kein Zweitcomputer? Sicher haben viele unserer Leser neben Ihrem grossen CBM 3032 oder 8032 noch den kleinen VC 20. Gerade für diesen Leserkreis ist diesmal unser Programm gedacht. Es soll naemlich dem Programmierer eine echte Hilfe bieten, der auf dem grossen CBM in Maschinensprache programmiert und dazu auf dem kleinen VC 20 seine Dezimalzahlen in Hexadezimalzahlen oder Binaerzahlen umrechnen muss oder umkehrt.

Das Programm rechnet Dezimalzahlen in Hexidezimalzahlen und Binaerzahlen um. Ebenso ist jede Umkehrung möglich. Darüberhinaus können Hexadezimalzahlen addiert und subtrahiert werden. Allerdings wird Ihnen das Studium des Programms zeigen, dass die Kost diesmal nicht so leicht ist.

Das Programm rechnet Dezimalzahlen in Hexadezimalzahlen und Binaerzahlen um. Ebenso ist jede Umkehrung möglich. Darüberhinaus können Hexadezimalzahlen addiert und subtrahiert werden. Allerdings wird Ihnen das Studium des Programms zeigen, dass die Kost diesmal nicht so leicht ist.

HEXADEZIMAL UMRECHNUNG

```
5 POKE36879,25
10 REM ***CR ING F.TILGNER-WIEN-1982***
20 PRINT"HEX-      DEZ-      BIN-"
21 PRINT:Q$="!"
22 PRINT"+HEX+HEX  DEHEX+DEZ"
23 PRINT"-HEX-HEX  DEHEX-DEZ"
24 PRINT"-----"
25 PRINT"RECHNER"
30 GETA$:IFA$=""THEN30
31 IFA$="R"THEN1900
32 IFA$="D"THENINPUT"DEZ      ";D:GOTO1310
34 IFA$="H"THENINPUT"HEX-    ";H$:GOTO1600
38 IFA$="+"ORR$="-"THEN90
40 IFA$="@"THENINPUT"HEX(+DEZ)";H$:GOTO1600
41 IFA$="*"THENINPUT"HEX(-DEZ)";H$:GOTO1600
42 IFA$="B"THENINPUT"BINAE-  ";B$
44 IFLEN(B$)>16THENPRINT"***OVERFLOW***":GOTO1850
48 GOTO1620
80 GOTO30
90 FORK=1TO10
100 IFA$="+"THENINPUT"HEX+. (↑) ";H$(K)
101 IFA$="-"THENINPUT"HEX-. (↑) ";H$(K)
102 PRINT"POSTEN:"K
110 IFH$(K)="↑"THEN1135
1000 DATA"0000","0","0001","1","0010","2","0011","3"
1010 DATA"0100","4","0101","5","0110","6","0111","7"
1020 DATA"1000","8","1001","9","1010","A","1011","B"
1030 DATA"1100","C","1101","D","1110","E","1111","F"
1050 H$(K)="0000"+H$(K):H$(K)=RIGHT$(H$(K),4)
1070 B$=""
1080 FORJ=1TO4
1090 HJ$=MID$(H$(K),J,1)
1100 GOSUB1150
```

```

1110 B#=B#+BJ#:NEXTJ
1120 GOSUB1210
1130 IFA#="+ "THEND(K)=D(K-1)+D(K)
1131 IFA#="- "THEND(K)=D(K)-D(K-1):D(K)=ABS(D(K))
1135 IFH#(K) <> "↑" THENNEXTK
1136 PRINT
1140 GOTO1300
1150 RESTORE
1160 FORI=1TO16
1170 READBJ#,F#
1180 IFHJ#=F# THENRETURN
1190 NEXT:BJ#="****":RETURN
1200 END
1210 D(K)=0:B=32768
1220 FORI=1TO16
1230 IFMID$(B#,I,1)="0" THEN1260
1240 IFMID$(B#,I,1) <> "1" THEND=-1E5:GOTO1260
1250 IFA#="B" ORA#="D" ORA#="H" ORA#="@" ORA#="*" THEND=D+B
1252 IFA#="+ " ORA#="- " THEND(K)=D(K)+B
1260 B=INT(B/2+.2):NEXTI:RETURN
1270 END
1300 IFA#="+ " ORA#="- " THEND=D(K-1)
1302 IFA#="H" THEN1310
1310 GOSUB1328:GOSUB1410
1312 IFD>65535 THENPRINT "OVERFLOWHEXSUM ";H#:PRINT:GOTO1850
1313 IFA#="+ " ORA#="- " THENPRINT "OVERFLOWHEXSUM ";H#:PRINT:POSTEN:";K-1
1316 IFA#="H" ORA#="D" ORA#="B" ORA#="@" ORA#="*" THENPRINT "OVERFLOWHEX ";H#
1317 IFA#="+ " ORA#="- " THENPRINT "OVERFLOWBINAERSUM ";B#:GOSUB1892
1318 IFA#="H" ORA#="D" ORA#="B" ORA#="@" ORA#="*" THENGOSUB1891
1319 IFA#="H" ORA#="D" ORA#="B" ORA#="@" ORA#="*" THENPRINT "OVERFLOWDEZIMAL ";D
1320 IFA#="+ " ORA#="- " THENPRINT "OVERFLOWDEZ.SUM ";D(K-1)
1327 GOTO1850
1328 A=D:B=65536
1330 IFA>(B-1) THENB#="****OVERFLOW****:RETURN
1340 B#=""
1350 FORI=1TO16
1360 B=INT(B/2+.2)
1370 A=A-B
1380 IFAC0 THENA=A+B:B#=B#+ "0":GOTO1400
1390 B#=B#+ "1"
1400 NEXT
1410 H#=""
1420 FORJ=1TO13STEP4
1430 BJ#=MID$(B#,J,4)
1440 GOSUB1500
1450 H#=H#+HJ#:NEXTJ
1460 RETURN
1500 RESTORE
1510 FORI=1TO16
1520 READE#,HJ#
1530 IFBJ#=E# THENRETURN
1540 NEXT:HJ#="*":RETURN
1600 IFLEN(H#)>4 THEN1850
1601 H#="0000"+H#:H#=RIGHT$(H#,4):B#=""
1602 FORJ=1TO4
1604 HJ#=MID$(H#,J,1):GOSUB1150
1606 B#=B#+BJ#:NEXTJ
1608 D=0:B=32768
1610 GOSUB1220:GOTO1612

```

```

1612 IFA$="H"ORA$="D"ORR$="B"THEN1312
1614 IFA$="@ "THENINPUT"加+DEZ.Z. " ;DZ:GOTO1625
1615 IFA$="*"THENINPUT"加-DEZ.Z. " ;DZ:GOTO1625
1620 B$="00000000000000000000"+B$:B$=RIGHT$(B$,16)
1623 D=0:B=32768:GOSUB1220:GOSUB1410:GOTO1312
1625 IFA$="@ "THEND=D+DZ:GOTO1310
1626 IFA$="*"THEND=D-DZ:GOTO1310
1850 PRINTTAB(15)"加加RETURN"
1860 GETX$:IFX$=""THEN1860
1870 IFASC(X$)=13THENIFQ$="*"THEN1900
1875 IFASC(X$)=13THENIFQ$="!"THEN20
1880 GOTO1860
1890 END
1891 PRINT"加BINAER 加 " ;B$
1892 PRINTTAB(14)"加加加加":RETURN
1894 END
1900 PRINT"加加RECHNER(加 = HEXDEZRIN)"
1905 PRINT:Q$="*"
1910 INPUT"加FAKTOR X加- " ;X$:X=VAL(X$)
1912 IFX$="*"THEN20
1920 INPUT"加FAKTOR Y加- " ;Y
1930 PRINT
1940 PRINT"加 加 加 加/加 加 加加IRZEL
1950 GETF$:IFF$=""THEN1950
1955 PRINT:PRINT
1960 IFF$="+"THENS=X+Y:PRINT"加1.+2.加=" ;S
1962 IFF$="-"THENS=Y-X:PRINT"加1.-2.加=" ;S
1964 IFF$="*"THENS=X*Y:PRINT"加1.*2.加=" ;S
1966 IFF$="/"THENS=X/Y:PRINT"加1./2.加=" ;S
1968 IFF$="^"THENS=X^Y:PRINT"加1.^2.加=" ;S
1970 IFF$="W"THENS=X^(1/Y):PRINT"加Y.W X加=" ;S
2000 GOTO1850

```

READY.

ANMERKUNG DER REDAKTION

Lieber CBM/PET News Leser,

von den vielen Zuschriften, die uns erreichen, sind die meisten an den Schweizer Computer Club Luzern gerichtet. Selbstverstaendlich erreichen uns auch diese Briefe, jedoch meist mit einiger Verspaetzung. Dürfen wir Sie daher bitten, im Interesse einer möglichst raschen Abwicklung, Ihre Zuschriften, Anfragen und Hardwarebestellungen nur an die

Mikro + Kleincomputer
 Informa Verlag AG
 z. Hd. Hr. Kastien
 Postfach 1401
 CH 6000 L U Z E R N

zu senden. Sie erhalten auf Ihre Anfragen innert nützlicher Frist Antwort, wenn nicht ohnehin im naechsten Heft die Beantwortung Ihrer Fragen oder die Veröffentlichung Ihres Beitrages vorgesehen ist. Haben Sie aber bitte auch Verstaendnis dafür, dass Ihr Beitrag nicht immer sofort erscheinen kann, da wir ein gewisses Konzept einhalten möchten und sich teilweise mehrere Beitrage überschneiden.

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

Also einen Commodore.



Commodore Volkscomputer VC20

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



Commodore CBM 4032

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



Commodore CBM 8032

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



Commodore MMF

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

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

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

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

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

Informations-Gutschein:

C/PN

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

Absender: _____

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

commodore
COMPUTER

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