

Informa Verlag AG
Postfach 1401
CH-6000 Luzern 15

3. Jahrgang
Erscheint 6 mal jährlich
ISSN 0251-2017

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

Lieber Commodore-Freund

Es ist etwas ruhiger geworden. Die Entwicklung der Kleincomputer scheint zur Zeit langsamer vorwaerts zu gehen als noch vor ein bis zwei Jahren. Untrügliches Zeichen dafür sind immer noch Computer-Ausstellungen und Messen, an denen lediglich die bekannten Typen dominieren, die allerdings in den verschiedensten Ausführungen angeboten werden. Aber wie gesagt, umwaelzende Neuerungen werden nicht gezeigt. Das Gleiche gilt auch für den Peripheriesektor.

Comodore hat zwar in den USA drei neue Kleincomputer vorgestellt, die sich aber nur als Weiterentwicklung bekannter Modelle entpuppen. Dabei handelt es sich einmal um einen Farbcomputer mit 64 kByte Speicherkapazität, der als Commodore 64 auf den Markt kommen wird und über einen 6510 Prozessor (als Weiterentwicklung des 6502), 64 k RAM und eine hochauflösende Farbgraphik verfügt, zum anderen wird der VC 20 in Baelde mit einer Erweiterung auf 16 kByte Speicherkapazität in der Grundversion erhaeltlich sein. Und zu guter Letzt schickt sich Commodore an mit Video-Spiel-Computern völliges Neuland zu betreten.

In den CBM/PET News wollen wir Ihnen in diesem und dem naechsten Heften wiederum einige echte Hardware-Neuheiten bieten. Die Barcode Lese- und Druckprogramme, die zur Zeit in einfacher Form vorliegen, werden im naechsten Heft in wesentlich verbesserter Form mit völlig neuem Barcode erscheinen wobei gleichzeitig die Programme als EPROM erhaeltlich sein werden. Ebenfalls im naechsten Heft stellen wir eine Hardwareuhr vor, die im Rechner eingebaut und einmal gestellt auch beim Abschalten des Rechners weiterläuft und jederzeit via Programm abgerufen werden kann und schliesslich erscheinen in einem der naechsten Hefte Programme zur Aenderung des Zeichensatzes des Rechners und damit der freien Zeichenwahl !

Dies sind einige Rosinen, die sicherlich auf reges Interesse stossen werden, da wir, wie immer, zur Hardware auch die erforderliche Software mitliefern und auch für den nötigen Service bei der Programmierung der EPROMs bieten.

Viel Freude mit Commodore wünscht Ihnen im Namen der Redaktion

Heinz Kastien

Heinz Kastien

Inhaltsverzeichnis

3	Ausgabeprogramm Lagerbuchhaltung	15	Aktive RC-Filter
8	Ausgabeprogramm Adressverwaltung	20	Barcodeausdruck
9	Wussten Sie schon ...	23	Leserbriefe
11	Barcodeleser	24	Squash für VC 20
12	Billigdrucker für VC 20	27	Umrüstsatz für EPSON MX 80

BASIC Programmier-Kurse

	Kurs Nr.	Datum	Kursbezeichnung
MAI	402	3./4. 5. 82	BASIC-Abend-Schnupperkurs
JUNI	322	3. – 4. 6. 82	BASIC-Fortsetzungskurs
	132	5. 6. 82	BASIC-Schnupperkurs
SEPTEMBER	133	4. 9. 82	BASIC-Schnupperkurs
	227	16. – 17. 9. 82	BASIC-Grundkurs
OKTOBER	134	23. 10. 82	BASIC-Schnupperkurs
	228	29. – 30. 10. 82	BASIC-Grundkurs
NOVEMBER	135	27. 11. 82	BASIC-Schnupperkurs
DEZEMBER	323	3. – 4. 12. 82	BASIC-Fortsetzungskurs

Kurskosten

BASIC-Abend-Schnupperkurs	Fr. 90.– (für SCC-Mitglieder Fr. 70.–)
BASIC-Schnupperkurs	Fr. 90.– (für SCC-Mitglieder Fr. 70.–)
BASIC-Grundkurs	Fr. 330.– (für SCC-Mitglieder Fr. 300.–)
BASIC-Fortsetzungskurs	Fr. 390.– (für SCC-Mitglieder Fr. 340.–)

Kurszeiten

Schnupperkurs:	jeweils 09.00 – 11.45 und 13.30 – ca.16.00 Uhr
Grund- und Fortsetzungskurs:	jeweils 09.00 – 12.00 und 13.30 – 17.00 Uhr
Abend-Schnupperkurse:	jeweils 19.00 – 21.00 Uhr

In den Kurskosten sind Kursmaterial und Dokumentationen als Nachschlagwerk inbegriffen. Alle Kurse werden im Schulungslokal des Schweizer Computer Club an der Seeburgstrasse 18 in Luzern (Nähe Verkehrshaus der Schweiz) durchgeführt.

Anmeldung bitte bei: Frl. Heidi Schüpfer, Telefon 041 - 31 45 45

Tricks und Tips

LAGERBUCHHALTUNG ADRESSVERWALTUNG

Programme, die auf Disk abgespeicherte Daten abrufen und nach bestimmten Kriterien auf dem Bildschirm darstellen oder auf einem Drucker ausgeben, sind prinzipiell alle gleich aufgebaut. Daher können auch hier für die Adressverwaltung und die Lagerkartei die gleichen Routinen angewendet werden.

Darüberhinaus besteht aber nicht nur eine Ähnlichkeit zwischen dem Abruf eines Artikels oder einer Adresse, sondern auch Programme zur Erstellung von

Inventaren
Lagerkarteien
Lagerlisten
Adresskarteien
Adressaufklebern
Adresslisten
usw.

sind in Ihrem Aufbau nahezu identisch und unterscheiden sich lediglich in der Art der Aufarbeitung der abgerufenen Daten und Ihrer Darstellung auf dem Drucker. Es können daher derartige Programme zusammengefasst werden.

Wenn wir hier für den einfachen Adress- und Artikelabruf und für das Inventar oder die Adressliste separate Programme machen, so nur, um die Uebersichtlichkeit innerhalb der Programme zu gewährleisten.

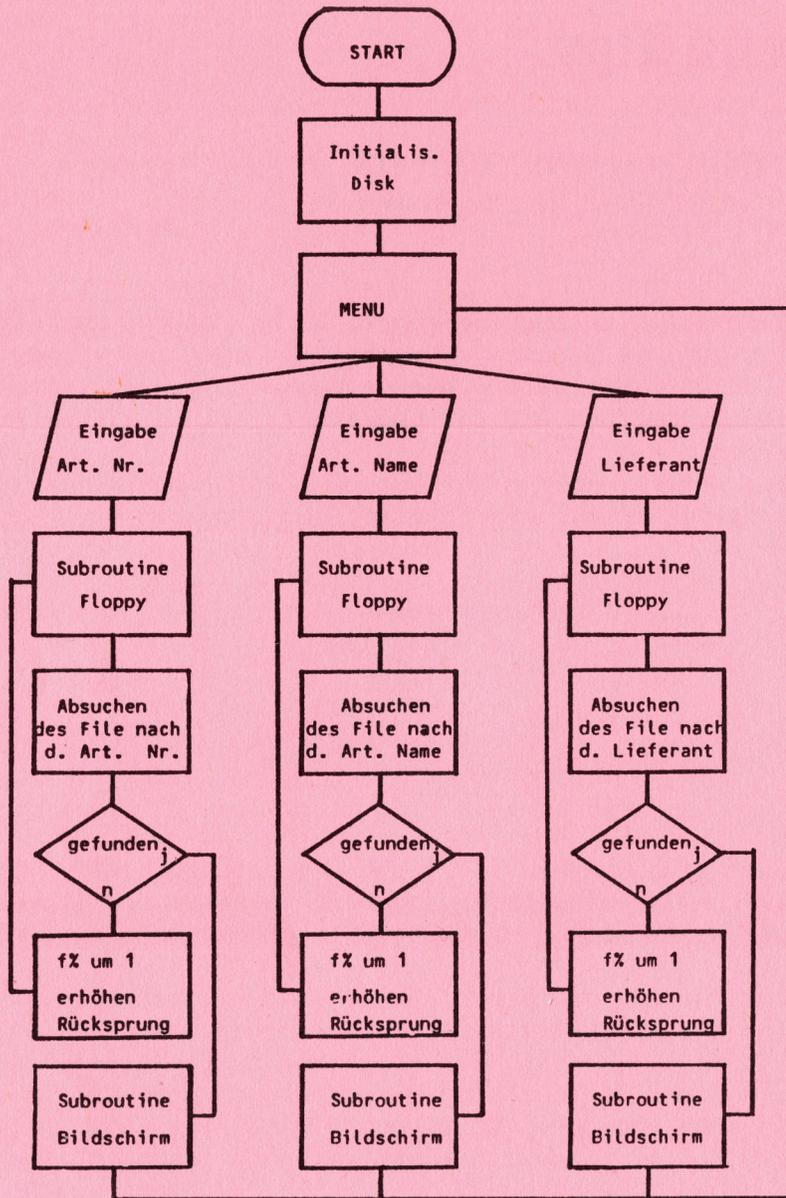
Sobald wir in einer der nächsten Fortsetzungen über die index-sequentiellen Programme sprechen, so werden wir mehrere Einzelprogramme, wie Inventar, Artikelliste usw. in einem Programm zusammenfassen.

Wie bereits gesagt, bestehen die genannten Programme aus Routinen, die immer wieder vorkommen. Man kann daher die Programme in folgende Blöcke unterteilen.

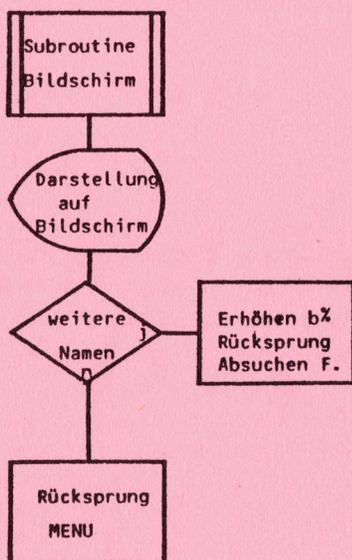
1. Auswahl des Kriteriums nach dem gesucht werden soll.
In einer Adressverwaltung kann beispielsweise nach dem Namen, dem Ort, der PLZ usw. gesucht werden.
2. Laden der Daten ab Floppy.
3. Absuchen der Daten und Laden des nächsten File, wenn der gesuchte Wert nicht im Datenfile enthalten war.
4. Darstellung der Daten auf dem Bildschirm
5. Rücksprung zu 1.

Die Teilprogramme 1, 2, 3 und 5 kommen immer wieder vor und die Unterschiede liegen lediglich in der Aufarbeitung der Daten, also Punkt 4. Um die Gleichartigkeit noch besser zu zeigen, werden wir die verschiedenen Routinen auch in allen noch folgenden Programmen immer mit den gleichen Variablen und den gleichen Zeilennummern versehen.

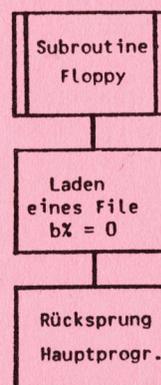
In verfeinerter Form können diese einzelnen Programmschritte auch in einem Flow-Chart dargestellt werden, wie dies hier für die Lagerbuchhaltung erfolgen soll, für die Adressverwaltung gilt sinngemäss das Gleiche.



Flow-Chart des Hauptprogramms Lagerbuchhaltung



Subroutine Bildschirmausgabe



Subroutine File ab Floppy laden

Programmbesprechung Lagerbuchhaltung

- 1000 - 1060 (1080) Aufruf zum Einlegen einer Datendisk.
2010 Dimensionierung des Feldes g\$
2020 - 2090 Ausgabemenu und Verzweigung des Programms mit ON GOTO
3000 - 3010 Laden des Hauptmenu ab Floppy.
8000 - 8080 Abruf eines Datenfiles ab Floppy.
Die Daten werden ab Drive 1 gelesen, wobei f% der Filezähler und b% der Datenzähler ist. Vor allem beim Floppy 2040 und 3040 kommt es immer wieder vor, dass vor das erste Zeichen im String ein Line Feed (ASCII 10) eingeschoben wird, die Daten können dann nicht mehr gelesen werden. Mit Zeile 8030 wird dies verhindert.
Ist der letzte String ein EOF oder END, so wird das File geschlossen und es erfolgt Rücksprung (Zeile 8040 und 8050). Aufruf der Fehlererkennungsroutine in Zeile 8070.
- 9000 - 9040 Fehlererkennungsroutine (siehe hierzu Heft 6/81)
10010 Definition der Variablen f% und b%
10020 - 10040 Titel und Eingabe der zu suchenden Artikelnummer. Die Eingabe der Artikelnummer erfolgt mit der gleichen Subroutine in Zeile 20000 wie bei der Eingabe und mit der gleichen Stellenzahl (Variable a).
10050 Sprung in die Einlesesubroutine (8000)
10200 Rücksetzen der Variablen b% auf 0.
10210 Wenn die linken ersten drei Zeichen des String ein END sind, wurde der gesuchte Artikel in keinem File gefunden und es erfolgt ein Sprung in die Zeile 18000.
10220 Wird die gesuchte Artikelnummer im File gefunden, $\text{left}(g$(b\%),6)=n\$$, erfolgt Sprung in die Druckroutine in Zeile 17000.
10230 Wenn die linken drei ersten Zeichen des String ein EOF sind wurde die Artikelnummer nicht in diesem File gefunden, f% wird dann um 1 erhöht und b% auf 0 zurückgesetzt und nach Laden des nächsten File erfolgt erneutes Absuchen des File.
10240 Erhöhen des Datenzeigers um 1 und Rücksprung.
12000 - 12240 Gleiche Suchroutine wie Zeile 10000 - 10240 jedoch nach dem Kriterium Artikelnamen $\text{mid}(g$(b\%),7,25)=n\$$
14000 - 14240 Gleiche Suchroutine wie Zeile 10000 - 10240 jedoch nach dem Kriterium Lieferant $\text{mid}(g$(b\%),32,20)=n\$$
17000 Ausgabe der Daten auf dem Bildschirm, das Flag dr wird gesetzt.
17010 - 17080 Der Datenstring g\$(b%) wird in seine Bestandteile zerlegt und auf dem Bildschirm dargestellt.
17100 - 17150 GET-Routine ob weitere Artikel aufgerufen werden sollen ansonsten erfolgt Rücksprung in das Ausgabemenu.
18000 - 18030 Diese Subroutine wird aufgerufen wenn ein Artikel in der Datei nicht vorhanden ist, wird die RETURN-Taste gedrückt, erfolgt Rücksprung in das Ausgabemenu.
Mit dem Flag dr wird erreicht, dass die Meldung "Artikel nicht bekannt !" nicht erfolgt, wenn zB. mehrere Artikel gleichen Namens bereits aufgerufen worden sind und in der Datei kein weiterer mehr vorhanden ist. Es erfolgt dann Rücksprung in das Ausgabemenu ohne Meldung.
20000 - 20170 GET - Subroutine (Heft 3/81)


```

14050 gosub8000
14200 b%=0
14210 ifleft$(g$(b%),3)="end"then18000
14220 ifmid$(g$(b%),32,20)=n$thengosub17000
14230 ifleft$(g$(b%),3)="eof"thenf%=f%+1:b%=0:goto14050
14240 ifleft$(g$(b%),3)="eof"thenf%=f%+1:b%=0:goto14050
14250 b%=b%+1:goto14210
17000 dr=1:rem ausgabe auf dem bildschirm
17010 printa$
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 print"#####Wollen Sie einen weiteren Artikel ?"
17110 print"#####R(j/n)!"
17120 getc$:ifc$=""then17120
17130 ifc$="j"thenreturn
17140 ifc$="n"then2020
17150 goto17120
18000 ifdr=1then18030
18010 print"#####Nicht 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+1: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" ||_";
20140 goto20010
20150 print" "
20160 iflen(x%)<athenx%=x%+" ":goto20160
20170 return

```

Das Programm der Adressverwaltung sieht prinzipiell genau gleich aus, wie das der Lagerbuchhaltung. Vor allem werden auch hier unter den gleichen Zeilennummern die gleichen Routinen ausgeführt.

Ein Unterschied liegt in der Initialisierung der Datendiskette (Zeilen 1000 - 1100), die im Gegensatz zum CBM 3000 beim CBM 8000 nicht erforderlich ist. Weiterhin ist der Aufruf der Fehlererkennung (Zeilen 9000-9100) beim CBM 8000 wesentlich einfacher als beim CBM 3000, im ersten Fall muss nur die Variable ds oder ds\$ aufgerufen, beim CBM 3000 dagegen mit INPUT der Fehlerstatus übernommen und dann ausgedruckt werden. Für dieses andersartige Verhalten ist das unterschiedliche DOS verantwortlich.

Durch den 40 Zeichen Bildschirm des CBM 3000 ergeben sich weitere kleinere Änderungen in der Darstellung der Zeilen 2000 - 3000 und 17000-18030.

ADRESSABRUF

```
1000 rem vorbereitung
1010 poke 59468,14
1020 t$="***** Adressausgabe ***"
1030 printt$
1040 print"***** Legen Sie die Datendisk in Drive # 1 !"
1050 print"**** Wenn fertig, dann <RETURN> druecken ! "
1060 setz$:ifz$<>chr$(13)then1060
1070 open15,8,15:print#15,"i1"
1080 gosub9000
2000 dima$(255):rem adressabruf
2010 a$="***** Adressverwaltung ***"
2020 printa$
2030 print"*****1. Ausgabe nach Namen "
2040 print"*****2. Ausgabe nach Ort "
2050 print"*****3. Ausgabe nach PLZ"
2060 print"*****4. zurueck ins Menu"
2070 setzx$:ifval(zx$)=0orval(zx$)>4then2070
2080 onval(zx$)goto10000,12000,14000,3000
3000 close15:print"****load****";chr$(34);"0:menu";chr$(34);",8
3010 print"****run****";poke158,2;poke623,13;poke624,13:end
8000 rem einleseroutine
8010 open1,8,2,"1:"+str$(f%)+",s,r":gosub9000
8020 input#1,g$(b%)
8030 ifleft$(g$(b%),1)=chr$(10)theng$(b%)=right$(g$(b%),len(g$(b%)))
8040 ifleft$(g$(b%),4)="end"thenclose1:return
8050 ifleft$(g$(b%),4)="eof"thenclose1:return
8060 b%=b%+1
8070 goto8020
8080 return
9000 input#15,en$,em$,et$,es$
9010 ifval(en$)=0thenreturn
9020 print"*****";en$;" " ;em$;" " ;et$;" " ;es$
9030 close1:close15:end
10000 rem aussgabe nach namen
10010 f%=0:b%=0
10020 printt$
10030 print"*****Welcher Name : ";
10040 a=15:gosub20000:n$=x$
10050 gosub8000
10200 b%=0
10210 ifleft$(g$(b%),4)="end"then18000
10220 ifmid$(g$(b%),16,15)=n$thengosub17000
10230 ifleft$(g$(b%),4)="eof"thenf%=f%+1:b%=0:goto10050
10240 b%=b%+1:goto10210
12000 rem aussgabe nach Ort
12010 f%=0:b%=0
12020 printt$
12030 print"*****Welcher Ort : ";
12040 a=20:gosub20000:n$=x$
12050 gosub8000
12200 b%=0
12210 ifleft$(g$(b%),3)="end"then18000
12220 ifmid$(g$(b%),37,20)=n$thengosub17000
12230 ifleft$(g$(b%),3)="eof"thenf%=f%+1:b%=0:goto12050
12240 b%=b%+1:goto12210
14000 rem aussgabe nach plz
14010 f%=0:b%=0
14020 printt$
14030 print"*****Welche PLZ : ";
14040 a=6:gosub20000:n$=x$
```

```

14050 gosub8000
14200 b%=0
14210 ifleft$(g$(b%),3)="end"then18000
14220 ifmid$(g$(b%),31,6)=n$thengosub17000
14230 ifleft$(g$(b%),3)="eof"thenf%=f%+1:b%=0:goto14050
14240 b%=b%+1:goto14210
17000 dr=1:rem druckroutine
17010 printt$
17020 print"#####Anrede          : ";left$(g$(b%),5)
17030 print"#####Vorname        : ";mid$(g$(b%),6,10)
17040 print"#####Name           : ";mid$(g$(b%),16,15)
17050 print"#####PLZ             : ";mid$(g$(b%),31,6)
17060 print"#####Ort             : ";mid$(g$(b%),37,20)
17070 print"#####Strasse & Nr.   : ";mid$(g$(b%),57,20)
17100 print"#####Wollen Sie einen weiteren Namen ?"
17110 print"#####(j/n)!"
17120 getc$:ifc$=""then17120
17130 ifc$="j"thenreturn
17140 ifc$="n"then2020
17150 goto17120
18000 ifdr=1then18030
18010 print"#####Nicht 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
ready.

```

Bei der Programmierung der Adress- und Artikelausgabe hat es sich herausgestellt, dass in den Eingabeprogrammen einige kleine Aenderungen gemacht werden sollten, um das Programm noch ein wenig komfortabler zu gestalten.

Weiterhin sind durch den Drucker CBM 8024 beim Listing einige Graphikzeichen nicht ausgedruckt worden, die ebenfalls noch eingefügt werden müssen.

Nachstehend sind die geänderten Zeilen der beiden Programme aufgelistet.

Programm Artikeleingabe

```
1080 printtab(19);"Minimalbestand  ";a=4:gosub3000:a7#=x$
```

Programm Adresseingabe

```
1000 REM ADRESSEINGABE  
1055 A=6:GOSUB3000:A4#=X$:GOSUB5000:A5#=B$:GOTO1070  
2020 IFB%>=25THENE$="EOF":GOSUB8000:B%=0:F%=F%+1
```

In beiden Programmen sollten die Subroutinen 3000-3170 wie folgt aussehen.

```
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" ||||";  
20140 GOTO20010  
20150 PRINT" "  
20160 IFLEN(X%)<ATHENX%=X%+" ":GOTO20160  
20170 RETURN  
READY.
```

WUSSTEN SIE SCHON ...

dass der Befehl MERGE, der im Toolkit des CBM 8000 enthalten ist und mit dem ein Programm ab Floppy an ein anderes im Speicher des Rechners angehängt wird, mit dem hier beschriebenen kleinen Programm auch auf dem CBM 3000 mit Floppy 2040 oder 3040 verwendet werden kann.

DISK-APPEND

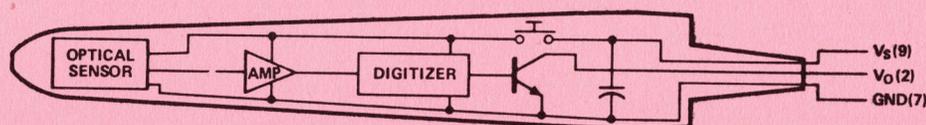
```
10 FORI=826TO947:READA:POKEI,A:NEXT  
20 DATA169,156,160,3,32,28,202,32,111,196,232,189,0,2,208,250,134,209  
30 DATA169,0,160,2,133,150,133,218,132,219,136,132,210,169,8,133,212  
40 DATA169,96,133,211,32,10,244,32,102,244,32,182,240,165,211,32,40,241  
50 DATA32,140,241,32,140,241,165,43,133,252,56,165,42,233,2,133,251  
60 DATA176,2,198,252,169,169,160,3,32,28,202,32,85,243,164,201,166,202  
70 DATA132,42,134,43,32,66,196,76,137,195,13,70,73,76,69,78,65,77,69  
80 DATA32,61,32,0,13,65,80,80,69,78,68,73,78,71,0  
READY.
```

Hardware

BARCODELESER

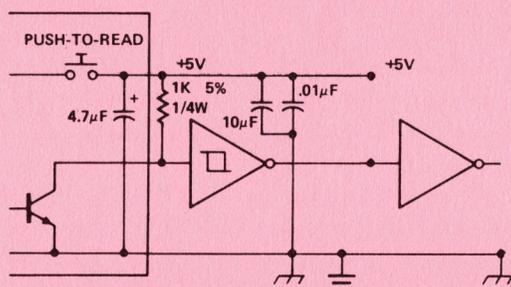
Schon im Heft 1/82 haben wir die Barcodeleser in ihrem grundsätzlichen Aufbau beschrieben. Heute soll der Barcode Lesestift von Hewlett-Packard näher beschrieben und vor allem ein Interface vorgestellt werden, das den Anschluss dieses Lesestiftes an die USER-PORT-Erweiterung bzw. direkt an den USER-PORT zulässt.

Der Lesestift HEDS 3000 von Hewlett Packard zeichnet sich vor allem durch seine handliche Form aus. Darüberhinaus ist die Lichtquelle kleiner und besser zentriert als der bereits im letzten Heft beschriebene Lesestift, dadurch tastet der Stift auch einen kleineren Bereich des Barcodes ab und führt weniger zu Lesefehlern als dies bei dem anderen Stift der Fall ist. Ausserdem verfügt der Lesestift HEDS 3000 über einen Schalter, somit kann der Stift nur dann eingeschaltet werden, wenn er wirklich gebraucht wird.



Barcodelesestift

Im Lesestift ist bereits ein Verstärker eingebaut, es handelt sich um einen Operationsverstärker, dem ein Transistor mit offenem Kollektor nachgeschaltet ist. Der Lesestift kann aber so nicht direkt an den CBM-Rechner angeschlossen werden. Hierzu ist ein spezielles Interface erforderlich. Es besteht aus einem Schmitt-Trigger und einem Inverter.



Schaltung des Einleseinterface

Am Kollektorausgang des Lesestiftes liegt ein 1 Kiloohm Lastwiderstand, sowie ein Schmitt-Trigger SN74LS14, der die Flankensteilheit des Signals erhöht. Da der Schmitt-Trigger das Signal invertiert, muss noch ein Inverter SN 7404 nachgeschaltet werden. Die beiden Kondensatoren C1 und C2 dienen der Entkopplung und Siebung.

Das Interface ist in zwei Versionen erhältlich:

1. Zum Einbau in die USER-PORT Erweiterung.
2. Zum direkten Anschluss an den USER-PORT und Stromversorgung aus dem Kassettenrecorderanschluss.

Der Einbau des Interface in die USER-PORT Erweiterung ist denkbar einfach. Es müssen hierzu die drei Anschlüsse 1,2 und 3 der Basisplatine mit einem mitgelieferten speziellen Stecker, der auf der Rückseite der USER-Port Erweiterung montiert wird, verbunden werden.

Bei der Version zum direkten Anschluss an den USER-PORT wird das Interface auf den Kassettenrecorderanschluss und den USER-PORT aufgesteckt und der Lesestift mit dem Interface über den Stecker verbunden.

Zum Einlesen des Barcodes wird das Programm aus Heft 1/82 geladen und mit dem Lesestift nach Betätigen des Schalters die Barcodezeile abgetastet.

Interface und Lesestift können unter folgenden Bestellnummern beim Computer Shop Luzern bezogen werden.

Einbauinterface für USER-PORT-Erweiterung	P 2270	Fr. 55.00
Interface für direkten Anschluss an USER-PORT	P 2271	Fr. 70.00
Lesestift HEDS 3000	P 2272	Fr. 280.00

BILLIGDRUCKER FUER VC-20

Zu den Besitzern des VC 20 zählen sicher auch viele Programmierer der ersten Stunde, also jene die einmal mit dem Onken Fernkurs (in der BRD Christiani) "Mikroprozessor-Labor" angefangen haben. Alle auf die das oben gesagte zutrifft, sind damit im Besitz eines Metallpapierdruckers. Dieser Drucker hat eine Schreibbreite von 22 Zeichen und druckt auf Aluminium beschichtetem Papier.

Was liegt also näher, als diesen einfachen und vor allem billigen Drucker zum Ausdruck von Programmen und Daten des VC 20 zu benutzen. Der Verwendung des Druckers zusammen mit den Rechnern der CBM Reihe dürften auf Grund der geringen Schreibbreite Grenzen gesetzt sein.

Der Drucker ist zwar nur in der Lage den normalen ASCII-Zeichensatz zu drucken, für die Graphikzeichen werden die Buchstaben der entsprechenden Tasten ausgegeben, aber was braucht man im Hobbybereich mehr.

Der Anschluss des Druckers an den VC 20 ist sehr einfach. Ein Stecker, der auf den USER-PORT passt, wird mittels eines 9 adrigen Kabels mit einer 16 poligen IC-Fassung verbunden, in die der Stecker des Druckers eingesteckt wird. Folgende Punkte müssen miteinander verbunden werden. Zusätzlich muss zwischen Punkt 8 und 9 der DIL-Fassung ein 10 Kilo-ohm-Widerstand eingelötet werden.

USER-PORT-Bez. Punkt der Steckerleiste PIN des 16 pol. Steckers

PB0	C	16	A8
PB1	D	15	A7
PB2	E	14	A6
PB3	F	13	A5
PB4	H	12	A4
PB5	J	11	A3
PB6	K	7	CALL
PB7	L	9	PRINT
GND	A	8	GND

Da der Drucker über eine eigene Spannungsversorgung verfügt, ist nur eine Masseverbindung mit dem Rechner erforderlich.

Das Druckprogramm ist in Maschinensprache geschrieben und vor dem Ueberschreiben mit BASIC geschützt.

```

12 REM (C)2.03.1982
16 REM SPERRSTR. 91
18 REM 4057 BASEL
19 REM TEL. 061/32 86
97
20 POKE56,29
30 FORI=7424TO7514
40 READA:POKEI,A
50 NEXT
60 PRINTCHR$(147)** C
  BM-VC20/DRUCKER **
70 PRINTCHR$(145)"====
  ====="
80 POKE1,254:POKE2,29:
  POKE3,3:SYS7437
90 END
100 DATA234,169,23,133
  ,3,169,254,133
110 DATA1,169,29,133,2
  ,234,169,191
120 DATA141,18,145,160
  ,23,177,1,9
130 DATA128,170,173,16
  ,145,41,64,240
140 DATA249,142,16,145
  ,173,16,145,41
150 DATA64,208,249,136
  ,208,231,24,165
160 DATA1,105,22,133,1
  ,165,2,105
170 DATA0,133,2,198,3,
  208,0,169
180 DATA0,141,16,145,9
  6,234,234,234
190 DATA234,169,0,141,
  16,145,162,255
200 DATA160,255,136,20
  8,253,202,200,248
210 DATA240,185,0

```

Programmlisting auf Metallpapierdrucker

Das Programm wird wie ein BASIC-Programm geladen und mit RUN gestartet. Nach dem Einlesen der DATA-Statement erscheint auf dem Bildschirm

```

** CBM-VC20/DRUCKER **
=====

```

Sie können nun das Druckprogramm mit NEW löschen, da es sich im Maschinenspeicher befindet und das zu listende Programm laden.

Zum Ausdruck gibt es nun verschiedene Möglichkeiten.

1. Mit SYS(7424) wird ein Hardcopy des Bildschirminhaltes erstellt.

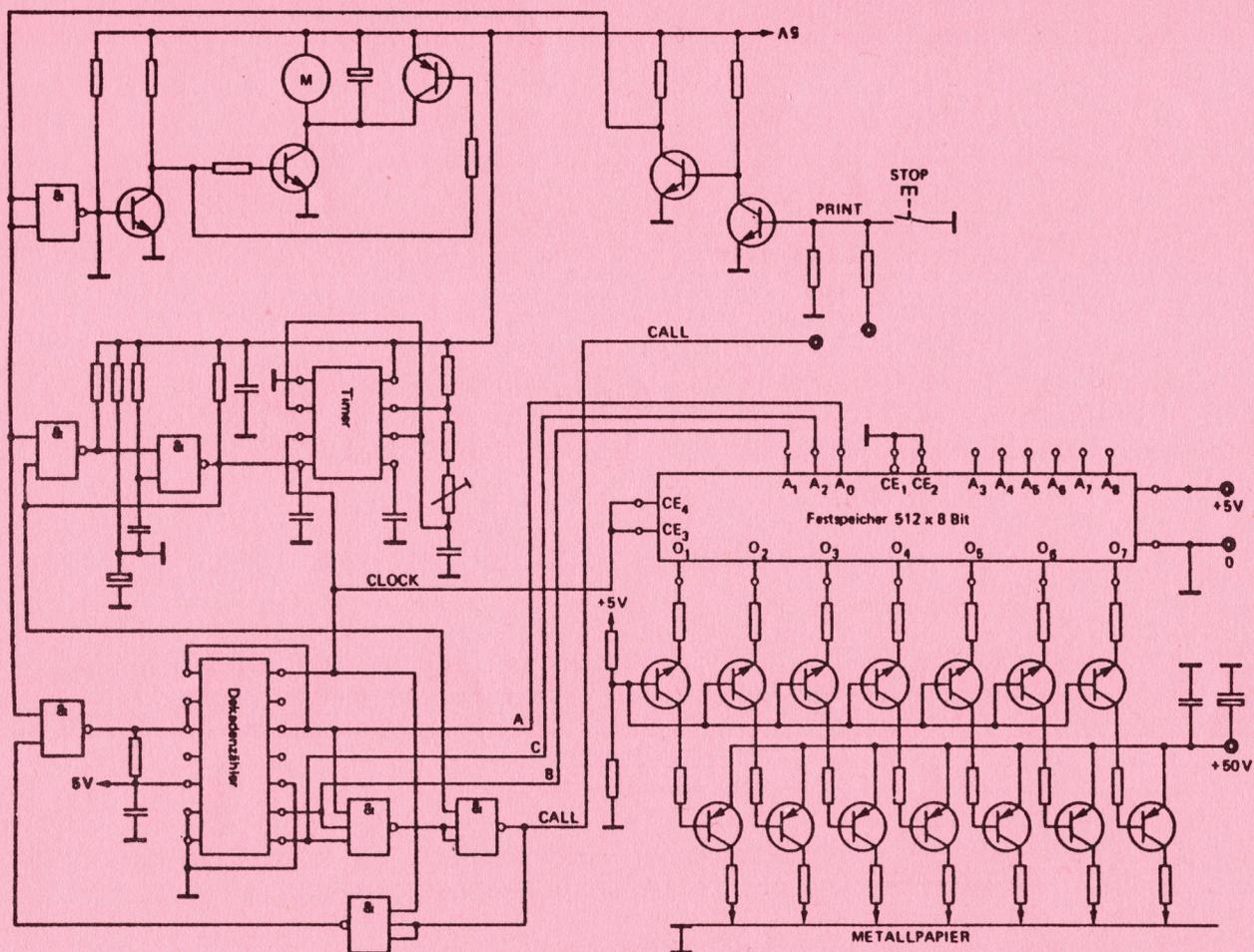
2. Mit POKE 3,Z:SYS(7429) werden alle Zeilen vom oberen Bildschirmrand beginnend bis zur Zeile Z ausgedruckt.

3. Mit POKE1,L:POKE2,H:POKE3,Z:SYS(7443) werden Z Zeilen a 22 Zeichen ab der Speicheradresse (L+(H*256)) ausgedruckt.

Programmbeschreibung

- 10 - 19 Textremarks
- 20 Reservierung des Speicherplatzes für das Maschinenprogramm
- 30 - 50 FOR-NEXT Schleife zum Einlesen der DATA-Statements in den Maschinenspeicher
- 60 - 70 Textausgabe auf dem Bildschirm
- 80 Ausgabe von drei Bildschirmzeilen auf den Drucker
- 90 Ende des BASIC-Teils
- 100 - 210 DATA-Statements

Die hier beschriebene Art des Anschlusses eines Druckers an den USER-PORT ist prinzipiell auch mit anderen Druckern, die gleich oder ähnlich konzipiert sind möglich, um eine Anpassung vorzunehmen drucken wir den Schaltplan eines ähnlichen Druckers ab.



Prinzipschaltung des Druckers

Programm des Monats BASIC-BASIC

AKTIVE RC - FILTER

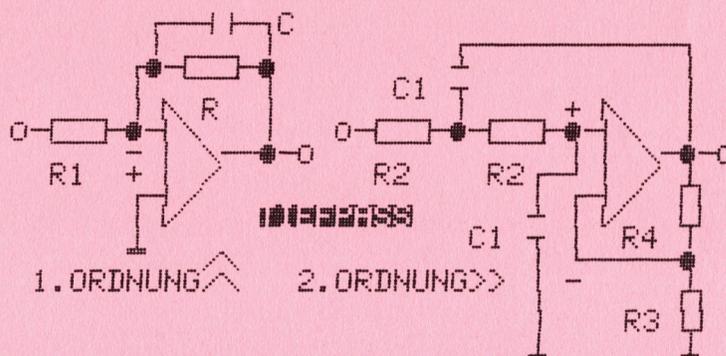
Heinrich Emmerl Kehlheim

Dieses Programm soll wieder einmal ein spezielles Thema aus der Elektronik behandeln, das auf Grund seiner komplexen Berechnungsgrundlagen zur Programmierung direkt prädestiniert ist, vor allem, da es gleichzeitig den Ausdruck des Schaltbildes und der erforderlichen Bauteile erlaubt.

Seit der allgemeinen Verbreitung der Operationsverstärkertechnik ist der Aufbau selektiver Filter auch für den Amateur problemlos. Unhandliche Spulen entfallen und an ihre Stelle treten leicht zu beschaffende Kondensatoren und Widerstände. Die Berechnung der Filter übernimmt ein BASIC-Programm.

Auf tiefschürfende Erklärungen der Filter soll verzichtet werden, eine kurze Beschreibung der mit dem Programm erfassbaren Filter soll genügen. Für weitere Informationen ist auf die Literatur verwiesen.

Das Berechnungsprogramm dimensioniert Hoch- und Tiefpassfilter 1. - 6. Ordnung, aufgebaut aus RC-Kombinationen und Operationsverstärkern. Diese Filter haben den Vorteil, dass durch Aenderung weniger passiver Bauelemente, Filtereigenschaften wie Grenzfrequenz, Steilheit und Uebertragungsverstärkung in weiten Grenzen vorgegeben werden können.



1. FILTER 1. ORDNUNG

1. FILTER 1. ORDNUNG

R= 6.48 KOHM
R1= 8.37 KOHM
C= 4.7 NF

2. FILTER 2. ORDNUNG

R2= 2.91 KOHM
R3= 10 KOHM
R4= 2.2 KOHM
C1= 10 NF

3. FILTER 2. ORDNUNG

R2= 9.53 KOHM
R3= 10 KOHM
R4= 9.04 KOHM
C1= 2.7 NF

*** FG= 3500 HZ
*** UE= 3 FACH

Es stehen drei Filtertypen zur Auswahl:

Filter in kritischer Kopplung besitzen eine geringe Flankensteilheit, dafür aber einen glatten Frequenzgang im Durchlassbereich und ein sehr gutes Impulsverhalten.

Besselfilter stellen einen Kompromiss zwischen Flankensteilheit und Impulstreue dar.

Tschebyschev-Filter haben eine hohe Steilheit, dafür aber eine grosse Welligkeit im Durchlassbereich und ein schlechteres Verhalten gegenüber Rechteckimpulsen.

Demzufolge wird man erstere z.B. in der HiFi-Technik anwenden, letztere etwa in der Sprechfunktechnik zur Unterdrückung unerwünschter Störsignale, wo es nicht so sehr auf extreme Uebertragungsqualität ankommt. Diese Unterschiede gelten nicht für Filter erster Ordnung. Hier sind Dimensionierung und Eigenschaften gleich. Erst ab der 2. Ordnung zeigt sich das abweichende Verhalten der verschiedenen Typen. Die Formeln zur Berechnung lehnen sich eng an die in (1) zu findende praxisgerechte Aufbereitung der Grundlagen (2) an.

Für Tiefpassfilter 1. Ordnung

$$F_g = \frac{k}{2 \cdot 3.14 \cdot R \cdot C} \quad U_e = \frac{R}{R_1}$$

Für Tiefpassfilter 2. Ordnung

$$F_g = \frac{k}{2 \cdot 3.14 \cdot R^2 \cdot C^2}$$

Für Hochpassfilter 1. Ordnung

$$F_g = \frac{1}{2 \cdot 3.14 \cdot R \cdot C \cdot k} \quad U_e = \frac{R_1}{R}$$

Für Hochpassfilter 2. Ordnung

$$F_g = \frac{1}{2 \cdot 3.14 \cdot R^2 \cdot C^2 \cdot k}$$

(F_g = Grenzfrequenz; U_e = Uebertragungsverstärkung; k = eine Filterkonstante, die je nach Typ und Ordnung variiert.)

Formelsammlung aktive RC-Filter

Die übrigen Bezeichnungen beziehen sich auf das von CBM und Drucker ausgegebene Schaltbild. Bei Filtern zweiter Ordnung muss die Uebertragungsverstärkung fest eingestellt werden. Filter höherer Ordnung erhält man durch Hintereinanderschalten von Filtern niederer Ordnung, ein Filter 5. Ordnung etwa aus einem Filter 1. Ordnung und zwei Filtern 2. Ordnung. Da sich dabei die Filter gegenseitig beeinflussen, muss die Dimensionierung jeder Filterstufe separat gerechnet werden. Der Rechner erledigt dies bis zu Filtern 6. Ordnung und gibt an, welche Filterordnung bei jeder Stufe notwendig ist. Sollte der Wert eines Widerstandes mit exakt 0 Ohm ausgedruckt werden, so ist hier eine Drahtbrücke einzusetzen, dies gilt vor allem bei Filtern in kritischer Kopplung.

AKTIVE RC - FILTER

```

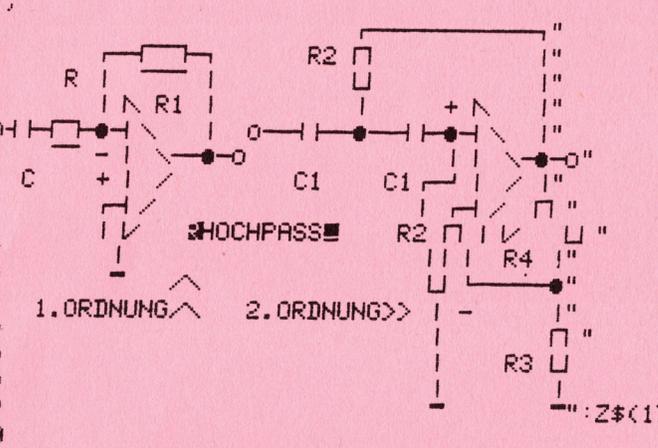
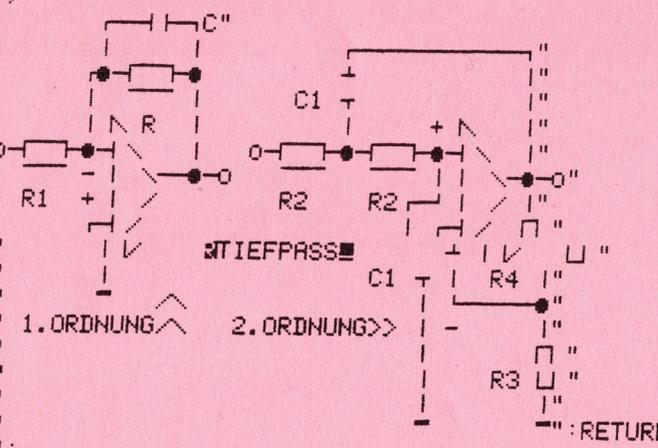
5 SQ$="XXXXXXXXXXXXXXXXXXXX":BA=33448:BE=33645:SP=32:HT$(2)="BESSEL":DIMZ$(17)
10 Z=12:DIMAK(Z),BK(Z),AB(Z),BB(Z),AT(Z),BT(Z):HT$(1)="KRITISCH":DIMA$(27)
15 FORI=1TOZ:READAK(I),BK(I),AB(I),BB(I),AT(I),BT(I):NEXT:DIMC(3),R2(3),R4(3)
20 HT$(3)="TSCHEBYSCHEV"
25 PRINT"TAB(10)"BERECHNUNG AKTIVER":PRINTTAB(7)"HOCH- UND TIEFPASSFILTER"
30 INPUT"HOCHPASS (1) ODER TIEFPASS (2)";HT:IFHT<1ORHT>2THEN25
35 IFHT=1THENHT$="HOCHPASS":GOTO45
40 HT$="TIEFPASS"
45 PRINT"KRITISCHE DAEMPfung (1)":PRINT" Bessel-FILTER (2)"
50 PRINT" TSCHEBYSCHEV 3DB (3)":INPUT";FT:IFFT<1ORFT>3THEN25
55 PRINT" WELCHER ORDNUNG SOLL DER "HT$:DZ=0:DL=0
60 PRINT" SEIN (1) BIS (6)";INPUT";FO:IFFO<1ORFO>6THEN25
65 ONHTGOTO70,75
70 GOSUB635:GOSUB80:ONFTGOTO120,125,130
75 GOSUB545:GOSUB80:ONFTGOTO120,125,130
80 FORI=1TO17:PRINTZ$(I):NEXT
85 PRINTLEFT$(SQ$,14):PRINT"HT$(FT)","FO"IL ORDNUNG":RETURN
90 GOSUB740
95 PRINTSQ$TAB(23)"NEUBEGINN MIT 'N'":PRINTTAB(23)"AUSDRUCK MIT 'P'"
100 GETN$:IFN$=""THEN100
105 IFN$="N"THEN25
110 IFN$="P"THEN860
115 PRINT"END"
120 ONFOGOTO135,150,165,190,210,245
125 ONFOGOTO135,275,290,315,340,375
130 ONFOGOTO135,410,425,450,475,510
135 PRINTSQ$:ANZ=1:ODZ=1:IX=1:AA=AK(1):GOSUB755:GOSUB760:GOSUB765
140 GOSUB770:GOSUB780:GOSUB815:GOSUB825:GOSUB835
145 GOTO90
150 PRINTSQ$:ANZ=1:ODZ=2:IX=1:AA=AK(2):BB=BK(2):GOSUB755:GOSUB765
155 GOSUB795:GOSUB815:GOSUB840:UE=1:GOSUB835
160 GOTO90
165 PRINTSQ$:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755
170 GOSUB760:GOSUB765:AA=AK(3):IX=1:GOSUB770:GOSUB780:IX=2:BB=BK(4)
175 GOSUB795:ANZ=1:ODZ=1:IX=1:GOSUB815:GOSUB825:ANZ=2:ODZ=2:IX=2
180 GOSUB720:GOSUB815:GOSUB840:GOSUB835
185 GOTO90
190 PRINTSQ$:ANZ=1:ODZ=2:IX=1:GOSUB755:ANZ=2:IX=2:GOSUB755:GOSUB765
195 IX=1:BB=BK(5):GOSUB795:IX=2:GOSUB795:ANZ=1:GOSUB815:IX=1
200 GOSUB840:ANZ=2:IX=2:GOSUB720:GOSUB815:GOSUB840:UE=1:GOSUB835
205 GOTO90
210 PRINTSQ$:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755
215 ANZ=3:IX=3:GOSUB755:GOSUB760
220 GOSUB765:AA=AK(7):IX=1:GOSUB770:GOSUB780:IX=2:BB=BK(8):GOSUB795
225 IX=3:GOSUB795:ANZ=1:ODZ=1:GOSUB815:GOSUB825
230 GOSUB720:ANZ=2:ODZ=2:GOSUB815:IX=2:GOSUB840:GOSUB720
235 ANZ=3:GOSUB815:IX=3:GOSUB840:GOSUB835
240 GOTO90
245 PRINTSQ$:ANZ=1:ODZ=2:IX=1:GOSUB755:ANZ=2:IX=2:GOSUB755:ANZ=3:IX=3
250 GOSUB755:GOSUB765:IX=1:BB=BK(10):GOSUB795:IX=2:GOSUB795:IX=3:GOSUB795
255 ANZ=1:GOSUB815:IX=1:GOSUB840:GOSUB720:ANZ=2:IX=2
260 GOSUB815:GOSUB840:GOSUB720:ANZ=3:IX=3:GOSUB815:GOSUB840
265 UE=1:GOSUB835
270 GOTO90
275 PRINTSQ$:ANZ=1:ODZ=2:IX=1:GOSUB755:GOSUB765:BB=BB(2)
280 GOSUB795:AA=AB(2):GOSUB810:GOSUB815:GOSUB850:GOSUB835
285 GOTO90
290 PRINTSQ$:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755
295 GOSUB760:GOSUB765:IX=1:AA=AB(3):GOSUB770:IX=2:BB=BB(4):GOSUB795
300 AA=AB(4):UA=UE:GOSUB810:GOSUB790:ANZ=1:ODZ=1:GOSUB815:IX=1
305 GOSUB825:GOSUB720:ANZ=2:ODZ=2:GOSUB815:IX=2:GOSUB850:GOSUB835
310 GOTO90
315 PRINTSQ$:ANZ=1:ODZ=2:IX=1:GOSUB755:ANZ=2:IX=2:GOSUB755:GOSUB765
320 IX=1:AA=AB(5):BB=BB(5):GOSUB795:GOSUB810:UC=UE:IX=2:AA=AB(6):BB=BB(6)
325 GOSUB795:GOSUB810:UD=UE:ANZ=1:ODZ=2:GOSUB815:IX=1:GOSUB850
330 GOSUB720:ANZ=2:GOSUB815:IX=2:GOSUB850:UE=UD*UC:GOSUB835
335 GOTO90
340 PRINTSQ$:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755
345 ANZ=3:IX=3:GOSUB755:GOSUB760:GOSUB765:IX=2:BB=BB(8):AA=AB(8)
350 GOSUB795:UA=UE:GOSUB810:UC=UE:IX=3:AA=AB(9):BB=BB(9):GOSUB795:GOSUB810
355 UD=UE:IX=1:AA=AB(7):GOSUB770:UE=UD*UC:GOSUB790:ANZ=1:ODZ=1
360 GOSUB815:GOSUB825:GOSUB720:ANZ=2:ODZ=2:GOSUB815:IX=2:GOSUB850
365 GOSUB720:ANZ=3:GOSUB815:IX=3:GOSUB850:GOSUB835
370 GOTO90

```

```

375 PRINTSQ#:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755:ANZ=3:IX=3
380 GOSUB755:GOSUB765:IX=1:BB=BB(10):AA=AB(10):GOSUB795:GOSUB810:UC=UE
385 IX=2:BB=BB(11):AA=AB(11):GOSUB795:GOSUB810:UD=UE:IX=3:BB=BB(12)
390 AA=AB(12):GOSUB795:GOSUB810:UE=UE*UD*UC:ANZ=1:GOSUB815:IX=1
395 GOSUB850:GOSUB720:ANZ=2:GOSUB815:IX=2:GOSUB850:GOSUB720
400 ANZ=3:GOSUB815:IX=3:GOSUB850:GOSUB835
405 GOTO90
410 PRINTSQ#:ANZ=1:ODZ=2:IX=1:GOSUB755:GOSUB765:BB=BT(2)
415 GOSUB795:AA=AT(2):GOSUB810:GOSUB815:GOSUB850:GOSUB835
420 GOTO90
425 PRINTSQ#:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755
430 GOSUB760:GOSUB765:IX=1:AA=AT(3):GOSUB770:IX=2:BB=BT(4):GOSUB795
435 AA=AT(4):UA=UE:GOSUB810:GOSUB790:ANZ=1:ODZ=1:GOSUB815:IX=1
440 GOSUB825:GOSUB720:ANZ=2:ODZ=2:GOSUB815:IX=2:GOSUB850:GOSUB835
445 GOTO90
450 PRINTSQ#:ANZ=1:ODZ=2:IX=1:GOSUB755:ANZ=2:IX=2:GOSUB755:GOSUB765
455 IX=1:AA=AT(5):BB=BT(5):GOSUB795:GOSUB810:UC=UE:IX=2:AA=AT(6):BB=BT(6)
460 GOSUB795:GOSUB810:UD=UE:ANZ=1:ODZ=2:GOSUB815:IX=1:GOSUB850
465 GOSUB720:ANZ=2:GOSUB815:IX=2:GOSUB850:UE=UD*UC:GOSUB835
470 GOTO90
475 PRINTSQ#:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755
480 ANZ=3:IX=3:GOSUB755:GOSUB760:GOSUB765:IX=2:BB=BT(8):AA=AT(8)
485 GOSUB795:UA=UE:GOSUB810:UC=UE:IX=3:AA=AT(9):BB=BT(9):GOSUB795:GOSUB810
490 UD=UE:IX=1:AA=AT(7):GOSUB770:UE=UD*UC:GOSUB790:ANZ=1:ODZ=1
495 GOSUB815:GOSUB825:GOSUB720:ANZ=2:ODZ=2:GOSUB815:IX=2:GOSUB850
500 GOSUB720:ANZ=3:GOSUB815:IX=3:GOSUB850:GOSUB835
505 GOTO90
510 PRINTSQ#:ANZ=1:ODZ=1:IX=1:GOSUB755:ANZ=2:ODZ=2:IX=2:GOSUB755:ANZ=3:IX=3
515 GOSUB755:GOSUB765:IX=1:BB=BT(10):AA=AT(10):GOSUB795:GOSUB810:UC=UE
520 IX=2:BB=BT(11):AA=AT(11):GOSUB795:GOSUB810:UD=UE:IX=3:BB=BT(12)
525 AA=AT(12):GOSUB795:GOSUB810:UE=UE*UD*UC:ANZ=1:GOSUB815:IX=1
530 GOSUB850:GOSUB720:ANZ=2:GOSUB815:IX=2:GOSUB850:GOSUB720
535 ANZ=3:GOSUB815:IX=3:GOSUB850:GOSUB835
540 GOTO90
545 PRINT "T";
550 Z$(1)="
555 Z$(2)="
560 Z$(3)="
565 Z$(4)="
570 Z$(5)="
575 Z$(6)="O
580 Z$(7)="
585 Z$(8)="
590 Z$(9)="
595 Z$(10)="
600 Z$(11)="
605 Z$(12)="
610 Z$(13)=" 1. ORDNUNG ^
615 Z$(14)="
620 Z$(15)="
625 Z$(16)="
630 Z$(17)="
635 PRINT "T";
640 Z$(1)="
645 Z$(2)="
650 Z$(3)="
655 Z$(4)="
660 Z$(5)="O
665 Z$(6)="
670 Z$(7)=" C
675 Z$(8)="
680 Z$(9)="
685 Z$(10)="
690 Z$(11)="
695 Z$(12)=" 1. ORDNUNG ^
700 Z$(13)="
705 Z$(14)="
710 Z$(15)="
715 Z$(16)="
720 GOSUB740
725 PRINTSQ#:PRINTTAB(23)"WEITER MIT DER":PRINTTAB(23)"SPACE-TASTE"
730 GETW#:IFW#<" THEN730
735 RETURN
740 PRINTSQ#"T":OPEN1,3
745 FORI=DZ+1TODZ+9:PRINT:INPUT#1,A$(I):A$(I)=LEFT$(A$(I),25):NEXT
750 DZ=DZ+9:DL=DL+1:CLOSE1:RETURN
755 PRINTANZ"FILTER"ODZ"ORDNUNG, C-WERT (NF)":INPUTC(IX):RETURN
760 PRINT"UEBERTRAGUNGSFAKTOR (?-FACH)":INPUTUE:RETURN

```



```

765 PRINT" GRENZFREQUENZ (HERTZ)          " : INPUTFG: RETURN
770 R=1/(2*PI*FG*AA*C(I%)*1E-6) : IFHT=2 THEN R=AA/(2*PI*FG*C(I%)*1E-6)
775 RETURN
780 R1=R*UE : IFHT=2 THEN R1=R/UE
785 R1=INT(R1*100+.5)/100: RETURN
790 UB=UA/UE : R1=R*UB : R1=INT(R1*100+.5)/100: UE=UA: RETURN
795 R2(I%)=1/(2*PI*FG*C(I%)*1E-6*SQR(BB))
800 IFHT=2 THEN R2(I%)=SQR(BB)/(2*PI*FG*C(I%)*1E-6)
805 RETURN
810 UE=3-AA/SQR(BB) : R4(I%)=(UE-1)*10: RETURN
815 FORI=BAT0BE:POKEI,SP:NEXT:PRINTSQ$ANZ"### FILTER"OD%"### ORDNUNG"
820 PRINT" " : FORI=1TO18:PRINT"" : NEXT:PRINT: RETURN
825 PRINT" R="INT(R*100+.5)/100"KOHM":PRINT" R1="R1"KOHM"
830 PRINT" C="C(I%)"NF":RETURN
835 PRINT"0 *** FG="FG"HZ":PRINT" *** UE="INT(UE*10+.5)/10"FACH": RETURN
840 PRINT" R2="INT(R2(I%)*100+.5)/100"KOHM":PRINT" R3 ENTFAELLT"
845 PRINT" R4= 0 KOHM":PRINT" C1="C(I%)"NF":RETURN
850 PRINT" R2="INT(R2(I%)*100+.5)/100"KOHM":PRINT" R3= 10 KOHM"
855 PRINT" R4="INT(R4(I%)*100+.5)/100"KOHM":PRINT" C1="C(I%)"NF":RETURN
860 OPEN4,4:OPEN6,4,6:PRINT#6,CHR$(18)
865 PRINT#4:FORI=1TO17:PRINT#4,Z$(I)
870 NEXT:PRINT#4
875 PRINT#6,CHR$(24):PRINT#4,HT$(FT),"LEFT$(STR$(FO),2)".ORDNUNG":PRINT#4
880 FORI=1TODL*9:PRINT#4,A$(I)
885 NEXT:PRINT#4
890 CLOSE4:CLOSE6:GOTO25
895 DATA1,0,1,0,1,0,1,29,41,1,36,62,1,07,1,93,51,0,76,0,3,35,0,1,26
900 DATA1,48,36,1,19,87,19,1,34,49,2,19,5,53,87,19,77,39,2,1,2,39,0
905 DATA,67,0,5,63,0,77,15,1,14,41,76,2,65,77,15,62,32,12,1,07,7,12
910 DATA1,22,39,3,27,11,7,7,12,97,35,41,1,99,7,12,51,28,08,1,09

```

Programmbeschreibung

- 5 - 20 Festlegen von Bildschirmadressen, Dimensionierung der Filterkonstanten.
- 25 - 60 Programmkopf und Auswahl des Filtertyps und der Filterordn.
- 65 - 75 Verzweigung Hochpass - Tiefpass, Einlesen der Schaltbildelemente.
- 80 - 85 Ausgabe des Schaltbildes auf dem Bildschirm.
- 90 Aufruf der Einleseroutine für Bildschirmzeilen zwecks späterer Ausgabe auf dem Drucker.
- 95 - 115 Programmfortsetzung nach beendeter Berechnung.
- 120 Sprungverteiler für Filter kritischer Kopplung.
- 125 Sprungverteiler für Besselfilter.
- 130 Sprungverteiler für Tschebyschevfilter.
- 135 Filter erster Ordnung
- 150 Filter zweiter Ordnung
- 275 Filter zweiter Ordnung
- 410 Filter zweiter Ordnung
- 135 - 540 Lineares Abarbeiten der in Unterprogrammen abgelegten Formeln mit Parameterübergabe je nach Filtertyp.
- 545 - 715 Festlegen der Zeichenketten für Bildschirm und Drucker.
- 725 - 735 Programmfortsetzung nach Teilberechnung.
- 740 - 750 Uebernahme der Rechenergebnisse vom Bildschirm zur späteren Ausgabe auf dem Drucker.
- 755 - 855 Formeln für Widerstände, Verstärkung, Unterscheidung zwischen Hoch- und Tiefpass, Ausgaberroutine auf dem Bildschirm.
- 860 - 890 Ausgaberroutine für den Drucker.
- 895 - 910 Filterkonstanten in DATA-Statements.

Literatur:

- (1) Pietsch, Amateur-Funkfern-schreibtechnik RTTY, München
- (2) Tietze-Schenk, Halbleiter-Schaltungstechnik, Berlin

Maschinensprache

BARCODEAUSDRUCK

Im letzten Heft wurde das Wesentlichste des Barcodes erklärt, dazu wurde das entsprechende Einleseprogramm vorgestellt. Der heutige Teil befasst sich mit dem Ausdruck des vorgestellten und verwendeten Barcodes. Das nachfolgend in BASIC geschriebene Programm druckt Strichcodezeilen aus. Vorab sind hierzu noch einige Bemerkungen erforderlich:

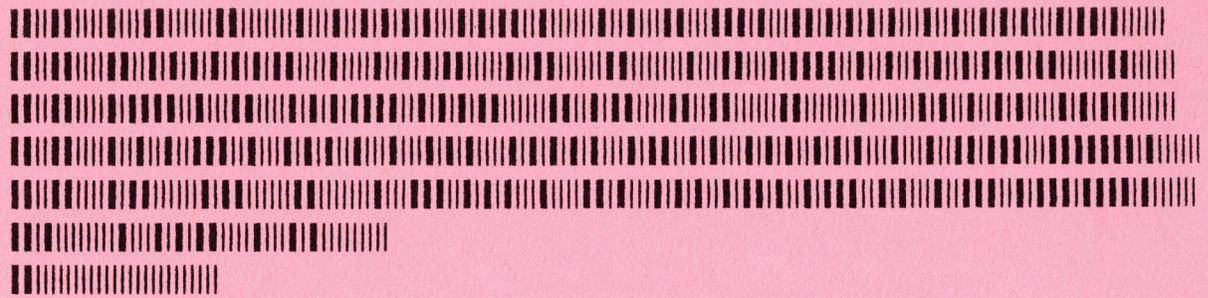
- Das Programm läuft auf allen CBM Rechnern ohne Anpassung, aber nur auf dem CBM-Drucker 4022.
- Mit diesem Drucker können zwar Graphikzeichen frei definiert werden, aber nur eines pro Druckzeile. Um trotzdem eine ganze Strichcodezeile zu bekommen, bedient man sich eines Tricks. Sobald das Sonderzeichen ausgegeben ist, wird eine Wagenrücklauf ohne Zeilenvorschub programmiert: `PRINT #4,CHR$(141);`. Erst danach kann ein neues Zeichen definiert und ausgegeben werden. Natürlich muss die alte Druckerposition wieder mit Blanks angefahren werden. Dadurch entsteht ein ständiges Hin und Her, das sich bei diesem Druckertyp nicht umgehen lässt.
- Dem Autor stand vorläufig nur dieser Drucker zur Verfügung, dafür war der Lerneffekt bei der Programmierung und der Handhabung des Druckers von grossem Nutzen. Dies könnte auch für manchen Anwender zutreffen, wenn man mehr als nur BASIC - Listings aus dem Drucker herausholen möchte. Weitere Druckertypen werden untersucht, ebenso werden auch die Printerprogramme verbessert, damit der Ausdruck des Barcodes schneller und komfortabler erfolgt. Entsprechende Programme werden Ihnen in der nächsten Ausgabe vorgestellt.
- Das Programm kann auch als Vorlage für andere Graphikausdrucke dienen. Im Detail kann es aber nur verstanden werden, wenn ausreichende Kenntnisse der Druckerprogrammierung vorhanden sind.

Programmbedienung

Nach dem Starten mit RUN erscheint TEXT : . Es wird nun das Programm, welches im Barcode ausgedruckt werden soll, auf den Bildschirm geschrieben und jede Zeile mit RETURN abgeschlossen. Die Zeilen werden entsprechend gespeichert. Die Eingabe wird abgebrochen, wenn END ohne Zeilennummer eingegeben wird. Es wird nun automatisch der eingetippte Text ausgedruckt. Anschliessend erfolgt die Uebersetzung des Textes in den Barcode und der zeilenweise Ausdruck. Auf dem Bildschirm wird dies entsprechend dokumentiert.

```
10 REM 1.VERSUCH
20 FORZ=1TO50
30 PRINT"*TEST*";
40 NEXTZ:END
```

Testprogramm im Klartext



Testprogramm im Barcode

```
1000 REM -----
1005 REM *   PRINT * BARCODE   *
1010 REM *
1015 REM *   PRINTER   CBM 4022 *
1020 REM -----
1030 DIM P$(90),T$(250),B$(90),A(7)
1035 C#=CHR$(254);D#=CHR$(141)
1040 C0#=CHR$(0);C1#=CHR$(127)
1045 N1#=CHR$(4);N2#=CHR$(36)
1050 LX=0;LE=12;F1=0;CR#=CHR$(13)
1055 FORI=1TO90:B$(I)=B$(I-1)+" ";NEXTI
1060 M=0;CR=13;CE=20;PRINT"□"
1061 Z#="" ;N=0;M=M+1;PRINT;PRINT"TEXT: ";
1062 GETA#;IFA#=""THEN1062
1063 A=ASC(A#);IFA=CRTHEN1067
1064 IFA<>CETHENZ#=Z#+A#;N=N+1;PRINTA#;GOTO1062
1065 IFN=0THEN1062
1066 N=N-1;PRINT"■ ■";Z#=MID$(Z#,1,N);GOTO1062
1067 IFN=0THEN1062
1070 T$(M)=Z#;IFZ#<>"END"THEN1061
1072 PRINT;PRINT
1075 OPEN4,4;FORJ=1TOM-1
1080 PRINT#4," ";T$(J);NEXTJ
1085 PRINT#4;CLOSE4
1090 M=1;ZX#=""
1100 Z#=T$(M);M=M+1;IFZ#<>"END"THEN1140
1120 IFLX=0THEN1260
1130 F1=1;LE=LX;GOTO1160
1140 LZ=LEN(Z#);LX=LX+LZ+1;ZX#=ZX#+Z#+CR#
1150 IF(LX<LE)THEN1100
1160 Z#=MID$(ZX#,1,LE);LX=LX-LE
1170 IFLX=0THENZX#="" ;GOTO1190
1180 ZX#=MID$(ZX#,LE+1)
1190 PRINTZ#
1200 A=LE;GOSUB5030
1210 FORJ=1TOLE;A=ASC(MID$(Z#,J,1))
1220 PS=PS+A;GOSUB6030;NEXTJ
1240 GOSUB3030;IFF1=0THEN1150
1260 A=0;GOSUB5030;GOSUB3030
1270 PRINT;PRINT"FERTIG!"
1280 PRINT;END
3000 REM -----
3010 REM * PRINTOUT *
3020 REM -----
3030 PH=INT(PS/256);PL=PS-PH*256
```

```

3040 A=PL;GOSUB6030:A=PH;GOSUB6030
3050 IFNN<6THENFORI=NNTO6:P$(I)=P$(I)+C0$:NEXTI
3060 OPEN4,4:OPEN6,4,6:PRINT#6,N1$
3080 OPEN5,4,5:FORK=1TO6:FORI=1TON
3100 PRINT#5,P$(I):PRINT#4,B$(I);C$;D$;:NEXTI
3140 PRINT#4:NEXTK
3150 CLOSE5:PRINT#6,N2$:CLOSE6
3200 PRINT#4:CLOSE4:RETURN
5000 REM -----
5010 REM * 1ST BYTE FROM LINE *
5020 REM -----
5030 FORI=1TO90:P$(I)="" :NEXTI
5040 P$(1)=C1$+C1$+C1$+C0$+C0$+C1$
5050 P$(2)=C1$+C1$+C0$+C0$:PS=0:N=2:NN=4
6000 REM -----
6010 REM * KONVERT BYTE *
6020 REM -----
6030 Z=128:FORI=0TO7:A(I)=0
6050 IFINT(A/Z)>0THENA(I)=1:A=A-Z
6070 Z=Z/2:NEXTI
6080 FORI=7TO0STEP-1:IN=1:IFA(I)=1THENIN=3
6100 FORII=1TOIN:IFNN=6THENNN=0:N=N+1
6120 NN=NN+1:P$(N)=P$(N)+C1$:NEXTII
6150 FORII=1TO2:IFNN=6THENNN=0:N=N+1
6170 NN=NN+1:P$(N)=P$(N)+C0$:NEXTII
6200 NEXTI:RETURN
READY.

```

Nun die Erklärung zum Programm selbst :

Um die Uebersicht zu behalten, was im Programm geschieht, ist es in logische Blöcke unterteilt.
Zu Beginn werden Programm- und Drucker spezifische Variable definiert.

Zeile 1030-1060	C\$ =CHR\$(254)	Ausdrucken der Sonderzeichen
	D\$ =CHR\$(141)	Wagenrücklauf ohne Zeilenvorschub
	C0\$=CHR\$(0)	Strichabstand
	C1\$=CHR\$(127)	alle Drucknadeln eines dünnen Strichs
	N1\$=CHR\$(4)	reduzierter Zeilenvorschub;
		36 Zeilen pro Zoll
	N2\$=CHR\$(36)	normaler Zeilenvorschub;
		4 Zeilen pro Zoll
	LE =12	begrenzt die Strichcodezeile auf
		12 ASCII Zeichen.

Zeile 1061-1072 Der zu druckende Text wird eingegeben. Wird die Anforderung mit END beantwortet, wird die Eingabe abgebrochen.

Zeile 1075-1085 der eingegebene Text wird zur Kontrolle im Klartext ausgedruckt.

Zeile 1090-1280 Der Text wird zeichenweise zur Strichcodezeile aufbereitet und ausgedruckt, dabei helfen die Subroutinen ab Zeile 3000, 5000 und 6000.

Zeile 3000-3200 Gibt jeweils eine Strichcodezeile aus. Dabei wird die gleiche Zeile mit kleinem Zeilenvorschub 6 mal ausgedruckt. Dies ist nötig, um eine gute Druckqualität zu erhalten.

Zeile 3080 Der Zähler k=1to6 kann bei gutem Farbband kleiner gewählt werden (3), dafür dann aber ein etwas grösserer Zeilenvorschub gesetzt werden, zB. N1\$=CHR\$(8). Durch Experimentieren Optimum herausfinden !

Zeile 3150 Drucker geht wieder in Normalzustand.
 Zeile 6000-6200 wandelt einen ASCII-Charakter in Strichcode um. Der Inputparameter wird in der Variablen A übergeben, siehe Zeile 1200-1210
 Zeile 5000-5050 Hier wird der Start einer Strichcodezeile gebildet, dazu wird die Subroutine 6000 mibenutzt.

Solchermassen ausgedruckte Zeilen können mit dem Einleseprogramm aus Heft 1/82 wieder eingelesen und kontrolliert werden. Sowohl die Art des Strichcodes als auch der Druckertyp ist für grössere Programme nicht geeignet, jedoch suchen wir ständig nach alternativen Lösungen. Gute Lösungen mit teureren Matrix- und Typenraddruckern sind uns bekannt, da sich diese aber in einer Preisklasse zwischen Fr. 3500 und 9000.-- bewegen, kommen diese für den Hobbyprogrammierer nicht in Frage, allerdings können wir Interessenten Lösungshinweise geben.

Leserbriefe

Herr Gano aus München sendet uns zum Programm Printout der Serie Assembler-Programmierung vom vergangenen Jahr eine verbesserte Ausführung.

PRINTOUT

```

90 PRINTCHR$(147)
100 PRINT" PRINTOUT BASIC-VERSION":PRINT
110 PRINT"AUSGABE BILDSCHIRM ODER DRUCKER:3/4":PRINT
120 GETA$: IFA$="" THEN120
130 IFA$="3" THENH=8:GOTO170
140 IFA$<"4" THEN120
150 INPUT"ANZAHL HEX-ZEICHEN PRO ZEILE":H:PRINT
160 PRINT"UEBERSCHRIFT : (0 = OHNE)":INPUTU$:PRINT:IFU$="" THENU$=""
170 INPUT"VON, BIS(DEZ)":X,Y:OPEN4,VAL(A$)
180 PRINT#4," "U$
190 IFX=999 THENPRINT#4:PRINT#4," "X".:":GOTO220
200 IFX=999 THENPRINT#4:PRINT#4," "X".:":GOTO220
210 PRINT#4:PRINT#4,X".:":
220 Q=X:A=X:G=1
230 GOSUB270:G=0:X=Q:T=0
240 FORZ=X TOY:A=PEEK(Z):T=T+1
250 IFT>H THENX=X+H:IFX=Y THEN190
260 GOSUB270:NEXTZ:PRINT#4:CLOSE4:END
270 N=0:IF A<16 THENPRINT#4,"0":
280 N=N+1:B=INT(A/16):C=A-(B*16):B(N)=INT(C+.5)
290 IF A<16 AND G=1 AND N<4 THENPRINT#4,"0":
300 A=B:IF B>0 THEN280
310 GOTO330
320 N=N-1:IF N=0 THEN360
330 IF B(N)=>10 THENA1=B(N)+55:GOTO350
340 A1=B(N)+48
350 PRINT#4,CHR$(A1):GOTO320
360 PRINT#4," ":RETURN
READY.
```

Ebenso erreichte uns ein Hilferuf eines Lesers.

Wer verfügt bereits über die Speicherbelegung des VC 20 und könnte diese dem Leser und uns zur Veröffentlichung zur Verfügung stellen.

LEICHTE KOST

SQUASH FUER VC 20

Wir stellen wieder einmal ein Programm für den VC 20 vor, das ein aktuelles Spiel zum Gegenstand hat. Squash hat in den letzten Jahren eine sehr starke Verbreitung gefunden, so dass unser Spiel eine gewisse Berechtigung hat.

Das Spiel kann sowohl mit einem, als auch mit zwei Spielern gespielt werden und ist dadurch sehr universell. Es sollen aber in diesem Programm vor allem die Möglichkeiten der Programmgestaltung mit einem Joystick gezeigt werden. Ein Joystick ist eine Art Steuerknüppel, der sich in alle Richtungen bewegen lässt und mit dem der Cursor auf dem Bildschirm beliebig verschoben werden kann. Der Joystick für den VC 20 ist seit einigen Wochen auf dem Markt. Der VC 20 hat gegenüber den andern Commodore Rechner einen direkten Anschluss für den Joystick. Für den CBM 3000 und 8000 werden wir im nächsten Heft ein entsprechendes Interface für diesen Joystick zusammen mit den anderen Rechnern anbieten. Hierdurch ergeben sich natürlich sehr grosse Möglichkeiten bei der Gestaltung von Spielen. Es wurde auch die Untermalung der einzelnen Spiele mit Tönen einbezogen. Das Spiel soll Sie also einerseits erfreuen und Ihnen auf der anderen Seite zeigen, wie Sie selbst Spiele zusammen mit dem Joystick auf dem VC 20 programmieren können.

```
1 REM   SQUASH VC 20:::2513 BYTES
2 REM   PROGRAMMIERT R. KONZ, MAERZ 1982
3 REM   NACH EINER IDEE VON H. KAST, BALGACH
10 POKE36879,25:PRINT"■■■"
15 PRINT"■ ■ S Q U A S H   MIT "
20 PRINT" ■ JOYSTICK UND TON "
25 PRINT"■■■"
30 PRINT"DAS IST EIN SPIEL,BEI"
32 PRINT"DEM EIN BALL MIT EINEM";
34 PRINT"SQUASHRAKET GEGEN DIE"
36 PRINT"WAND GESPIELT WIRD"
38 PRINT"UND WIEDER GETROFFEN"
40 PRINT"WERDEN SOLL. DER"
42 PRINT"BALLEINWURF ERFOLGT"
44 PRINT"DURCH DRUECKEN DER"
46 PRINT"TASTE ■■ 'ACTION' ■■ AM"
48 PRINT"JOYSTICK,DIE
50 PRINT"SCHLAEGERBEWEGUNG "
52 PRINT"DURCH BETAETIGEN DES"
54 PRINT"STEUERKNUEPPELS NACH"
56 PRINT"LINKS ODER RECHTS."
58 PRINT" ■■(WEITER MIT 'SPACE')■■"
60 GETW$:IFW$<" "THEN60
65 POKE36879,107:PRINT"■■■■■ BIS ZU 2 SPIELER"
70 PRINT" KOENNEN ABWECHSELND"
75 PRINT"GEGENEINANDER SPIELEN."
80 PRINT" DIE PUNKTE WERDEN"
85 PRINT" LAUFEND ANGEZEIGT."
```

```

90 PRINT"■■■■■■■■ WIEVIELE SPIELER ?"
100 GETE$:IFE$=""THEN100
110 AS=ASC(E$)-48:IFAS<10RAS>2THEN100
120 POKE36879,216:PRINT"■"
130 C1=32:C2=120:C3=81:SA=8130:LI=8121:RE=8140:AP=7725
140 L=0:R=19:O=0:U=17:FH=1:FV=1
150 I=RND(-TI):P1=36874:P2=36875:P3=36876:P0=36878
200 PRINT"■■■■";
210 PRINT"_____ "
220 FORI=1TO19
230 PRINT" |           | ";
235 NEXT
240 I=0:GOSUB900:POKESA,C2
250 GOSUB9000
260 IFFR<>1THEN250
300 H=INT(RND(1)*21):V=INT(RND(1)*10):TX=240:GOSUB1000
310 POKESA,C1:SA=8130:SN=SA:POKESA,C2
399 REM --- BALLBEWEGUNG ---
400 TX=200:IFH>=RTHENFH=-1:GOSUB1000:GOTO420
410 IFH<=LTHENFH=1:GOSUB1000
420 H=H+FH:IFV=0THENFV=1:GOSUB1000
430 V=V+FV:GOSUB2000
435 BN=AP+V*22+H:POKEBN,C3:POKEBA,C1:BA=BN
436 FORX=21TO23:IFPEEK(BN+X)=C2THEN440
438 NEXT:IFV<0THEN600
440 FV=-1:IFSN-BN=22THEN480
450 IFSN-BN=23THENFH=-1:GOTO480
460 IFSN-BN<>21THEN800
470 FH=1
480 T(I)=T(I)+1:TX=255:GOSUB1000:REM --- TREFFER ---
599 REM --- SCHLAEGERBEWEGUNG ---
600 FORJ=1TO2:GOSUB9000
610 IFJ0=1THENSN=SA+1:IFPEEK(SN+1)=101THENSN=SN-1
620 IFJ1=1THENSN=SA+22:IFSN>RETHENSN=SN-22
630 IFJ2=1THENSN=SA-1:IFPEEK(SN-1)=103THENSN=SN+1
640 IFJ3=1THENSN=SA-22:IFPEEK(SN-22)=100THENSN=SN+22
650 POKESA,C1:POKESN,C2:SA=SN:NEXT:GOTO400
799 REM --- FEHLSCHLAG ---
800 POKEBA,C1:GOSUB1100:GOSUB900:GOTO250
899 REM +++ SBR TREFFERANZEIGE +++
900 PRINT"■";:FORJ=1TOAS:PRINTTAB((J-1)*11+1);J;"-";T(J);:NEXTJ
910 I=I+1:IFI>ASTHENI=1
920 PRINT"■";TAB((I-1)*11+1);"■";I:RETURN
999 REM +++ SBR AKUSTIK +++
1000 POKEP0,15:POKEP1,200:POKEP2,200:POKEP3,TX
1010 FORTZ=1TO20:NEXT:POKEP1,0:POKEP0,0
1020 RETURN
1100 POKEP0,15:POKEP1,200:POKEP2,200
1110 FORTZ=128TO255:POKEP3,TZ:NEXT:POKEP1,0:POKEP0,0
1120 RETURN
2000 REM +++ FARBGEBUNG +++
2010 FG=38445:FA=BF
2020 BF=FG+V*22+H
2030 POKEBF,2:POKEFA,0
2040 RETURN
9000 REM --- EINLESEN VOM JOYSTICK ---
9020 DD=37154
9030 Q1=37151
9040 Q2=37152
9050 POKEDD,127
9060 P=PEEK(Q2)AND128

```

```

9070 J0--(P=0)
9080 POKEDD,255
9090 P=PEEK(Q1)
9100 J1--((PAND8)=0)
9110 J2--((PAND16)=0)
9120 J3--((PAND4)=0)
9130 FR--((PAND32)=0)
9140 RETURN
READY.

```

Programmbeschreibung

- 10 - 60 Titel und Spielerklärung
- 65 - 110 Frage nach der Anzahl der Spieler
- 120 Lässt nur die Zahlen 1 und 2 zu
- 130 - 160 Setzen der POKE-Variablen
 - Bildschirmpoke (130)
 - Ballcode (130)
 - Randbegrenzungen (140)
 - Töne (150)
 - Joystickpeek (160)
- 200 - 240 Spielfeld auf den Bildschirm zeichnen
- 250 - 260 Sprung zur Joystick Subroutine und warten auf Balleinwurf
- 300 - 310 Balleinwurf mittels Zufallszahlengenerator
- 400 - 430 Ballbewegung
- 440 - 480 Frage nach Treffer
- 600 - 620 Schlägerbewegungen
- 800 Fehlschlag mit Spezialakustik
- 900 - 920 Trefferanzeige
- 1000 - 1010 Tonuntermalung für Ballbewegung
- 1100 - 1110 Akustik für Fehlschlag
- 2000 - 2040 Unterprogramm für roten Ball
- 9000 - 9140 Leseroutine für Joystick PEEKs



- Grösste Auswahl an Geräten, Programmen, Büchern
 - Kompetente Fachberatung
 - Schnelle Serviceleistung, eigener technischer Dienst
 - Grösstes Computer-Versandhaus
 - Basic-Programmierkurse
- Seeburgstrasse 18, 6002 Luzern, Tel. 041 31 45 45

Nowak AG

Eidg. konz. Fachgeschäft

Technikumstrasse 46
8400 Winterthur
Telefon 052/22 08 03



CBM/PET NEWS

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

Manuskripte
Mit der Zustellung von Manuskripten anerkennt der Autor die Copyrightbestimmungen des Verlages. Mit der Annahme von Manuskripten durch die Redaktion und der Autor-Honorierung durch den Verlag hat dieser das Recht zur Veröffentlichung der entsprechenden Beiträge in anderen verlagseigenen Publikationen und 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.
© 1982 by 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.

Fertige Lösungen für Ihren Klein- oder Mittelbetrieb

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

LOGON AG

Baslerstrasse 145
8048 Zürich
Telefon 01 62 59 22

Zugerstrasse 69
8048 Zürich
Telefon 042 31 44 80



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

Hard-+ Software
aus einer Hand

Neuheiten

EPROM UMRÜSTSATZ FÜR EPSON MX-80

Obwohl der COMMODORE Tractor Printer 3022 und 4022 mit dem EPSON-Drukker nahezu identisch ist, verhält er sich hinsichtlich des Ausdrucks völlig anders als der entsprechende COMMODORE Typ, da der Original EPSON - Printer MX 80 nicht über den Zeichensatz des COMMODORE Printers verfügt. Dies macht sich in erster Linie bei den Graphikzeichen und in der Art der Ansteuerung bemerkbar.

Da der EPSON-Printer MX 80 jedoch sehr stark verbreitet ist, hat die Firma:

Unitronic GmbH
Münsterstr. 338

D 4000 Düsseldorf

einen entsprechenden Umbausatz auf den Markt gebracht, der es erlaubt, mit dem EPSON MX 80 nahezu gleich zu arbeiten, wie mit den COMMODORE Printern.

Der Umbausatz besteht aus einem Anschlussstecker und einem steckbaren ROM-Modul und ist denkbar einfach zu montieren.

Nach dem Einbau des Moduls erlaubt der EPSON-Printer folgende Druckmöglichkeiten.

Darstellung der CBM-Graphik einschliesslich Cursorsymbolen.
Gross - Kleinschreibung mit OPEN1,4
Gross-Graphikzeichen mit OPEN1,4,4
Anschluss an 24 poligen Stecker mit Erweiterung an Floppy
Alle Datenmodes des EPSON wie Breitschrift, 40-,80-,132 Zeichen/Zeile sind möglich, ebenso TAB-Funktion und Doppeldruck.

Alle Befehle zur Steuerung der Druckausführung, Papiertransport und Tabulatorfunktionen, Schriftbreite und Stärke sowie die Sonderbefehle sind über CHR\$ zugänglich. Eine Steuerung über die Sekundäradresse wie beim CBM-Printer ist nicht möglich, jedoch auch nicht erforderlich, da die Charakterstringfunktionen einen gleichwertigen Ersatz bieten. Dagegen fehlt die Möglichkeit des formatierten Ausdrucks in der gewohnten Form völlig, kann aber ebenfalls durch eine horizontale und vertikale TAB-Funktion ersetzt werden.

Der gesamte Umbausatz U-8010 L ist zum Preis von DM 260.-- bei der oben genannten Firma erhältlich.

Ebenso kann ein Anschlusskabel CBM-EPSON Printer unter der Nr. U-8010 K zu DM 90.-- und ein Anschlusskabel mit Floppy Adapter zu DM 130.-- geliefert werden.

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

Also einen Commodore.

Besuchen Sie uns
an der
COMPUTER 82
25.-28. Mai
Palais de Beaulieu
Lausanne
Halle 2, Stand 201



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:

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

Absender: _____

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

C/PN

commodore
COMPUTER

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

2-82