

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,

Wo sind die Computerenthusiasten geblieben, die sich mit der standardisierten Hardware für jedermann nicht begnügten, und durch selbst vorgenommene Hardwareänderungen aus Ihrem "Ackergaul-computer" eine "Rennpferd-Maschine" machten? Wo sind die begeisterten Elektroniker und Hobbybastler geblieben, die ihre gemachten Erfahrungen und ihre Ideen auch anderen Computerfreunden zugänglich gemacht haben? Und wo ist der Wissensdurst all derer, die mit ihrem Computer nicht nur Reversi spielen wollen? Sind wir alle schon wunschlos glücklich? Wir können es fast nicht glauben.

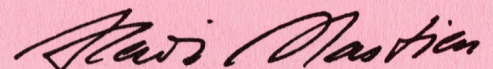
Im letzten Heft haben wir Sie gebeten uns zum Thema "Hardware" Ihre Meinung, Ihre Wünsche aber auch Ihre Anregungen mitzuteilen. Das Echo war nicht gerade überwältigend. Nur einige wenige Leser haben sich geäußert und dann auch nur um zu sagen, wir sollten dieses Thema vermehrt berücksichtigen; das Gleiche wurde übrigens auch zum Thema "Maschinensprache" gesagt.

Trotz der bescheidenen Resonanz haben wir uns über künftige Themen Gedanken gemacht. Ein grosses und für viele sicher auch interessantes Thema ist z.B. sicher die Steuerung und Ueberwachung von Modelleisenbahnen. Wenn Computersteuerungen bei der SBB bereits vorhanden sind, warum sollte dies dann bei der Modelleisenbahn nicht möglich sein. Gleisschaltbilder auf dem Bildschirm, Blockstellenbelegung mit dem Computer, Zusammenstellung von Zugskompositionen per Programm ... Warum eigentlich nicht?

Ein weiteres, nicht weniger interessantes Thema stellt z.B. auch der Amateurfunk dar. Sicher gibt es hier eine ganze Reihe von Problemen, die sich via Computer einfacher und schneller lösen lassen.

Zu beiden Themenvorschlägen sind wir auf Ihre tatkräftige Mithilfe angewiesen, um die wir Sie an dieser Stelle herzlich bitten. Wir sind keine Eisenbahn- oder Funkspezialisten, und wir wollen uns in den CBM/PET NEWS nicht in Gemeinplätzen ergehen, sondern interessante Anwendungen auf diesen Gebieten bringen. Wer schreibt uns?

Viel Freude mit Commodore wünscht Ihnen im Namen der Redaktion



Heinz Kastien

Inhaltsverzeichnis

3 Daten Append	21 Cursorsteuerung
7 Indexsequentielle Programme	22 HEX - BCD Umwandlung
11 Floppy Disk Erweiterung	25 Betriebssystemerweiterung
13 Berechnung von Siedekurven	26 Dezimaldatum

Das ausführliche CBM-Handbuch

für jeden Commodore-Benutzer, der auch in Maschinensprache arbeiten will.
Sämtliche "Spezialitäten", neue Adressen, Funktionen und Möglichkeiten der
CBM-Betriebssysteme 3000 und 4000/8000 werden eingehend erläutert.



Umfang ca. 220 Seiten im praktischen A5-Format, Paperback. Auslieferung Ende
November 1982. Subskriptionspreis für MIKRO+KLEINCOMPUTER- und CBM/PET NEWS-
Leser SFr./DM 40.--, gültig bis 15. November 1982, danach SFr./DM 49.--.
Bestellung mittels Einzahlungsschein auf Postkonto Luzern 60-27181, Stuttgart
3786-709 oder Eurocheck direkt beim Verlag MIKRO+KLEINCOMPUTER INFORMA VERLAG AG
Postfach 1401, CH-6000 Luzern 15

Tricks und Tips

INDEXSEQUENTIELLE ABSPEICHERUNG APPEND

In den vergangenen Heften haben wir die Adressverwaltungsprogramme und die Lagerbuchhaltung so vervollstaendigt, dass eine komplette Programmsammlung vorliegt, die allen Anforderungen gerecht werden sollte.

Beim Gebrauch der Programme hat sich aber auch sicher bei Ihnen bereits gezeigt, dass immer noch einige Schwachstellen in den Programmen auszumerken sind, denn einerseits ist es mit dem vorhandenen Eingabeprogramm nicht möglich, an die bestehenden Datenfile weitere Daten anzuhaengen und andererseits werden die Suchzeiten des Floppy in den Abrufprogrammen bei grösseren Datenmengen zu lang.

Heute sollen Wege aufgezeigt werden, die es ermöglichen, diese Maengel zu beheben.

Anhaengen von Daten an bestehende Datenfile

Zum Erweitern von Dateien bieten sich prinzipiell zwei Möglichkeiten an, naemlich der APPEND-Befehl, der aber leider nur im Befehlssatz des CBM 8000 enthalten ist und die Möglichkeit einer Programmerweiterung, die neue Daten an das Ende der bestehenden Datei anhaengt, ohne die vorhandenen Daten zu veraendern. Es wird hier am Beispiel der Lagerbuchhaltung die Anwendung des APPEND-Befehls gezeigt, waehrend bei der Adressverwaltung die gleiche Aufgabenstellung mit einer kleinen Programmaenderung bewerkstelligt wird, wobei diese Möglichkeit natürlich auf allen Programmen und Geraetetypen angewendet werden kann.

APPEND

Programmerweiterung

Wir sind bei der Dateneingabe der oben erwaehnten Programme von der Voraussetzung ausgegangen, dass nach jeweils 25 Datenstrings ein "eof" gesetzt wird und ein neues Datenfile eröffnet wird, wobei alle Datenfile fortlaufend nummeriert sind, diese Regel hat nur eine einzige Ausnahme, dies ist der letzte Datenstring, sobald die Frage "Wollen Sie weitere Daten eingeben?" mit NEIN beantwortet wird, wird hinter den letzten Datenstring ein "end" gesetzt. Weiterhin ist bekannt, dass alle Datenstring durch zwei Zahlen charakterisiert sind, naemlich durch die Filenummer f% und Stringnummer b%. Das vorliegende Programm bewirkt nun ein Absuchen aller Datenfile nach dem "end", sobald dieses gefunden ist, wird f% und b% zwischengespeichert und das Programm springt in die eigentliche Eingabe zurück, wobei der nun eingegebene Datenstring die Filenummer f% und die Stringnummer b% des "end" traegt, also dieses überschreibt. Das Programm wird nun wie bekannt fortgesetzt, wobei das "end" beim letzten Datenstring der neu hinzukommenden Daten wieder gesetzt wird und somit erneutes Anhaengen von Daten möglich ist. Das Absuchen der Datenstring, geht selbst bei grösseren Datenmengen relativ schnell, ausserdem werden wir Ihnen in der Fortsetzung dieser Abhandlung zeigen, wie Sie gerade die Floppyroutine noch weiter beschleunigen können.

Wenn Ihre Disk bereits Daten enthaelt, also auf der Disk auch bereits ein "end" gesetzt worden ist, können Sie das erweiterte Eingabeprogramm direkt benutzen. Sofern Sie aber eine neue Disk einsetzen, muss auf dieser Disk ein "end" gesetzt werden, da das Programm sonst nicht funktioniert. Hierzu kann ein kleines Programm "Eröffnen" benutzt werden.

Dieses Programm muss nur einmal, naemlich bei der ersten Anwendung einer neuen Disk benutzt werden. Dieses kleine Hilfsprogramm hat aber noch weitere Vorteile. Es nummeriert naemlich gleichzeitig die Disk und macht es möglich, bei einer sehr grossen Datenmenge nun auch mit einem Mehrdiskettensystem zu arbeiten, wobei die neue Disk automatisch verlangt wird, wenn der Rechner die Daten auf den bisherigen Disk nicht gefunden hat. Diese Erweiterung werden wir aber zu einem spaeteren Zeitpunkt in das Programm einbauen.

Artikeleingabe

```

900 rem vorbereitung
910 t$="***** Artikeleingabe *****"
920 print$ 1!"
930 printtab(14)"      Lesen Sie eine formatierte Disk in Drive #
940 printtab(14)"      Wenn Sie fertig sind, dann <RETURN> druecken !
950 setz$:ifz$<>chr$(13)then950
960 dimg$(25)
970 gosub10000:gosub9000
1000 rem artikeleingabe
1010 printtab(19)t$
1020 printtab(19);"      Artikelnummer      ";:a=6:gosub3000:a1$=x$
1030 printtab(19);"      Artikelname       ";:a=25:gosub3000:a2$=x$
1040 printtab(19);"      Lieferant        ";:a=20:gosub3000:a3$=x$
1050 printtab(19);"      Einstandspreis   ";:a=7:gosub3000:a4$=x$
1060 printtab(19);"      Verkaufspreis    ";:a=7:gosub3000:a5$=x$
1070 printtab(19);"      Lagerbestand     ";:a=6:gosub3000:a6$=x$
1080 printtab(19);"      Minimalbestand   ";:a=4:gosub3000:a7$=x$
1090 printtab(19);"      Stimmen die Daten ?"
1100 getz$
1110 if z$=""then1100
1120 ifz$="j"then 1150
1130 if z$="n"then1010
1140 goto1100
1150 printtab(19);"      Wollen Sie weitere Artikel ?"
1160 getz$
1170 if z$=""then1160
1180 ifz$="j"thengosub2000:goto1010
1190 if z$="n"thene$="end":gosub2000:gosub8000:goto4000
1200 goto1160
2000 g$(b%)=a1$+a2$+a3$+a4$+a5$+a6$+a7$
2010 b%=b%+1
2020 ifb%=>25thene$="eof":gosub8000:b%=0:f%=f%+1
2030 return
3000 x$="":an=0:print "  ";
3010 getzz$:ifzz$=""then3010
3020 zz=asc(zz$)
3030 ifzz=20then3090
3040 ifzz=13then3150
3050 x$=x$+zz$:an=an+1
3060 ifan>athenprint " ";:fori=1toan+1:print"  ";:next:goto3000
3070 printzz$;"  ";
3080 goto3010
3090 ifan=1thenx$="":an=0:goto3130
3100 ifan<1then3010
3110 an=an-1
3120 x$=left$(x$,an)
3130 print "  ";
3140 goto3010

```

```

3150 print" "
3160 iflen(x$)<athen:x$=x$+" ":goto3160
3170 return
4000 print"🔽🔽🔽load";chr$(34);"0:menu";chr$(34);",8"
4010 print"🔽🔽🔽🔽run🔽";poke158,2:poke623,13:poke624,13:end
8000 rem abspeicherroutine
8010 open1,8,2,"@0:"+str$(f%)+",s,w"
8020 fori=0to b%:print#1,g$(i):next
8030 print#1,e$:gosub9000
8040 dclose#1
8050 return
9000 rem fehlererkennung
9010 fs=ds
9020 ifds=0thenreturn
9030 print"🔽🔽🔽🔽🔽🔽🔽🔽🔽🔽";tab(24)ds$
9040 dclose#1:end
10000 rem ende suchen
10010 dopen#1,(str$(f%)),d0,r
10020 input#1,g$(b%)
10030 ifg$(b%)="eof"thenf%=f%+1:b%=0:dclose#1:goto10010
10040 ifg$(b%)="end"thendclose#1:return
10050 b%=b%+1:goto10020

```

Programmbeschreibung

10000	Remark
10010	Eröffnen eines Lesefiles
10020	Einlesen eines Strings des Lesefiles
10030	Prüfen ob die linken drei Zeichen des String = "eof" sind, wenn ja, dann wird der Filezaehler um 1 erhöht und der Stringzaehler b% auf Null gesetzt, einlesen des naechsten File.
10040	Prüfen ob die linken drei Zeichen des String = "end" sind, wenn ja, dann wird b%=b% und f%=f% gesetzt und es erfolgt Rücksprung in die Eingabe.
10050	Wenn weder 10030 oder 10040 zutrifft, wird b% um 1 erhöht und es erfolgt Rücksprung nach 10020.

Das Programm "Eröffnen" wird zweckmässigerweise nicht in das Menu integriert, da bei einer Fehlmanipulation ansonsten die gesamten Daten gelöscht werden. Es ist daher anzuraten, das Programm "Eröffnen" nur mittels dload bei Bedarf direkt zu laden.

Eröffnen

```

100 rem eroeffnen
110 print"🔽🔽🔽🔽";tab(20)"*** Eroeffnen einer neuen Disk ***"
120 print"🔽🔽";tab(20)"Name der Disk          :";:a=16:gosub1000:dn$=x$
130 print"🔽🔽";tab(20)"Identifikationsnummer :";:a=2:gosub1000:in$=x$
140 g$  ="end"
150 open15,8,15
160 print#15,"n1:"+dn$+", "+in$
170 open1,8,5,"1: 0,s,w":print#1,g$
180 close1:close15
190 print"🔽🔽🔽load";chr$(34);"0:menu";chr$(34);",8"
200 print"🔽🔽🔽🔽run🔽";poke158,2:poke623,13:poke624,13:end

```

```

1000 x$="":an=0:print " II";
1010 getzz$:ifzz$=""then1010
1020 zz=asc(zz$)
1030 ifzz=20then1090
1040 ifzz=13then1150
1050 x$=x$+zz$:an=an+1
1060 ifan>athenprint " ":fori=1toan+1:print"II II":next:goto1000
1070 printzz$:"II";
1080 goto1010
1090 ifan=1thenx$="":an=0:goto1130
1100 ifan<1then1010
1110 an=an-1
1120 x$=left$(x$,an)
1130 print " IIIII";
1140 goto1010
1150 print " "
1160 iflen(x$)<athenx$=x$+" ":goto1160
1170 return

```

Programmbeschreibung

100	Remark
110 - 130	Eingabe der Disknummer
140	Eröffnen eines Schreibfile
150	Es werden die Disknummer und ein "end" auf die Disk geschrieben und das File geschlossen.
160 - 170	Laden des Menu

Bedienungsanleitung des Programms "Eröffnen"

1. Es wird eine leere Disk in Drive 1 eingelegt.
2. Das Programm "Eröffnen", das sich auf der Programmdisk befindet, wird vom Drive 0 geladen und mit RUN gestartet.
3. Es wird der Name der Disk, die Identifikationsnummer und die Disknummer geladen, wobei die Identifikationsnummer immer zweistellig sein muss.

Append - Befehl

Der Append Befehl ist nur im BASIC - 4 enthalten und kann daher nur beim CBM 4000 und 8000 angewendet werden. Prinzipiell handelt es sich hierbei um einen speziellen Save-Befehl zum Abspeichern sequentieller Daten, wobei die Daten aber automatisch an das Ende der bestehenden File angehaengt werden. Für das Programm "Lagerbuchhaltung" würde dies wie folgt aussehen.

Im Programm Dateneingabe muss also lediglich die Abspeicherroutine in Zeile 8000 geaendert werden. Das Programm kann nun anstelle des normalen Eingabeprogramms benutzt werden. Da mit diesem Programm nur Daten an bereits bestehende Datenfile angehaengt werden, muss entweder die Disk erstmals mit dem Programm "Eroeffnen" eroeffnet werden oder aber im Menu müssen zwei Eingabeprogramme figurieren, naemlich einmal das Programm "Eingabe" für eine erstmalige Dateneingabe und das Programm Append für alle weiteren Eingaben. In jedem Fall ist es aber einfacher, mit dem Programm "Eroeffnen" zu arbeiten, da dies nur ein Dateneingabeprogramm erforderlich macht.

APPEND

```
8000 rem appendroutine
8010 dopen#1,(str$(f%))
8020 fori=0to9:print#1,g$(i):next
8030 print#1,e$:gosub9000
8040 dclose#1
8050 append#2,str$(f%)
8060 fori=0to9:print#2,g$(i):next
8070 print#1,e$
8080 dclose#2
ready.
```

INDEXSEQUENTIELLE ABSPEICHERUNG

Gerade bei grossen Datenmengen sind rein sequentielle Dateien nicht gerade das Ideale, da der Rechner unter ungünstigen Umständen sehr lange braucht, bis er den richtigen Datenstring gefunden hat. Der Grund für dieses Verhalten muss in der sequentiellen Abspeichermethodik gesucht werden. Es werden alle Daten hintereinander auf die Disk geschrieben, sobald ein File gefüllt ist, wird ein neues File eröffnet und mit Daten gefüllt um dann hinter dem vorhergehenden File auf der Disk abgespeichert zu werden. Wird nun ein bestimmter Datenstring gesucht, so liest der Rechner das erste File ein, sucht es ab und laedt, wenn er den richtigen String nicht gefunden hat, das naechste File ein, sucht es ab usw., befindet sich der gesuchte String z.B. auf dem 20 File, so muss diese Prozedur 20 mal wiederholt werden, was natürlich sehr lange dauert. Um diesem Uebelstand abzuhelpen gibt es folgende Möglichkeiten:

- a. indexsequentielle File
- b. relative File
- c. random File

Die random File, also den Direktzugriff sowie die relativen File wollen wir auf die naechsten Hefte verschieben und dann sehr ausführlich behandeln. Heute sollen die indexsequentiellen File besprochen werden, vor allem, da sich die bestehenden Programme sehr leicht in indexsequentielle umwandeln lassen.

Wie der Name bereits sagt, ist auch das indexsequentielle File ein normales sequentielles File, mit all seinen Nachteilen. Nach erfolgter Abspeicherung der Daten wird nun aber von allen File ein Register angelegt. Dieses Register beinhaltet, jeweils den ersten Wert eines File, z.B. bei einer Adressverwaltung den Namen oder bei der Lagerbuchhaltung die erste Artikelnummer. Dieses Register wird als Indexfile abgespeichert und beim Abrufen eines Artikels zuerst abgerufen. Der Rechner vergleicht nun, ob die gesuchte Artikelnummer grösser oder kleiner ist als der erste Wert des Indexregisters. Ist der Indexwert des ersten und zweiten File kleiner als der gesuchte Wert, so kann der gesuchte Wert nicht im ersten Datenfile sein. Es wird dann weiter gesucht bis das File ermittelt ist, indem sich der gesuchte Wert befinden muss. Es wird nun direkt dieses ermittelte Datenfile in den Rechner geladen und abgesucht. Waehrend vorher 20 Lade- und Suchvorgaenge erforderlich waren, um einen Wert aus dem 20. File zu finden, ist nun nur noch einer erforderlich, da das Indexfile nur einmal am Anfang des Abrufprogrammes geladen wird. Es laesst sich somit sehr viel Zeit einsparen. Ein Nachteil die-

ser Programmierart ist die Notwendigkeit der Neuindexierung nach der Eingabe weiterer Daten oder dem Löschen von Daten, da aber die Indexierung sehr schnell geht, ist dies nur von untergeordneter Bedeutung.

Artikelindex

```

1000 rem index
1010 print"#####";tab(20)"@Lesen Sie die Datendisk im Drive 1
1020 printtab(18)"###Wenn bereit druecken Sie < R E T U R N > !#"
1030 setz$:ifz$<>chr$(13)then1030
1040 n$=""
1050 dima$(256):aZ=0
1060 fZ=0
1070 open1,8,2,"0:"+str$(fZ)+",s,r"
1080 input#1,n$:n$=n$
1090 input#1,n$:ifasc(n$)=10thenn$=right$(n$,len(n$)-1)
1100 ifn$="eof"orn$="end"thena$(aZ)="":goto1150
1110 a$(aZ)=left$(n$,6)
1120 input#1,n$:ifasc(n$)=10thenn$=right$(n$,len(n$)-1)
1130 ifn$="eof"orn$="end"then1150
1140 n$=n$:goto1120
1150 aZ=aZ+1:close1:if"end"=n$then1170
1160 fZ=fZ+1:goto1070
1170 open1,8,2,"@:indexfile,s,w"
1180 fori=0toaZ-1:print#1,a$(i)
1190 nexti:print#1,"eof"
1200 close1
1210 print"###load";chr$(34);"menu";chr$(34);",8"
1220 print"###run";poke158,2:poke623,13:poke624,13:end

```

Weiterhin darf nicht unerwähnt bleiben, dass der Nutzen einer index-sequentiellen Abspeicherung nur dann voll wirksam ist, wenn die Artikelnummern sortiert sind. In dieser Ausgabe der CBM/PET NEWS wollen wir die Art der Indexierung und das Einlesen des Index beschreiben, in einer der nächsten Ausgaben der News gehen wir dann auf die Sortier-routinen ein.

Nach Abschluss der Dateneingabe, oder nach mehreren Mutationen wird vom Menu das Programm "Index" aufgerufen, das die Indexierung und die Ab-speicherung des Indexfile vornimmt.

Programmbeschreibung

- 1000 - 1030 Remark, Aufforderung zum Einlegen der Disk und Warteschlei-fe durch Druecken der RETURN-Taste.
- 1040 - 1060 Setzen von Variablen und Dimensionieren
- 1070 Das erste Datenfile wird eroeffnet
- 1080 - 1090 Es wird der erste Datenstring eingelesen, wenn das erste Zeichen ein Space ist, wird dieses eliminiert
- 1100 Wenn der Datenstring ein "eof" ist, erfolgt Sprung nach Zeile 1150
- 1110 a\$(0) wird gleich der Artikelnummer des ersten Artikels im File gesetzt
- 1120 - 1140 Es werden die übrigen Datenstring des betreffenden File gelesen, sobald ein "eof" oder "end" gefunden wird, erfolgt ein Sprung nach Zeile 1150
- 1150 aZ wird um 1 erhöht, das Datenfile geschlossen, wurde ein "end" gefunden, erfolgt ein Sprung nach 1170

1160 Wurde kein "end" gefunden, wird f% um 1 erhöht und es erfolgt Sprung nach 1070
 1170 Es wird das "Indexfile" eröffnet und alle a\$(a%), also alle Nummern der ersten Datenstring der File abgespeichert und zuletzt ein "eof" gesetzt
 1210 - 1220 Es wird automatisch das Menu geladen

Programmbeschreibung

Damit das indexsequentielle Programme voll zur Wirkung kommt, muss nun natürlich zuerst das Indexfile geladen werden, dies erfolgt mit der Einlesesubroutine in Zeile 40000.

Nach jedem Datenabruf nach der Artikelnummer, wird das Datenfile gelesen und durch Vergleich festgestellt, in welchem File sich der gesuchte String befindet `for i=0 to a%-1: if a$(i) < a$ then next i: i=i-1` nur dieses File wird geladen. Die beiden Subroutinen werden in den Programmen "Abruf" und "Mutation" wie folgt eingebaut.

Programm "Abruf" Zeile 2010 `dim g$(30), a$(50)`
 Zeile 2015 `gosub 40000`
 Zeile 10050 `gosub 50000: gosub 8000`

Programm "Mutation" Zeile 2010 `dim g$(30), a$(50)`
 Zeile 2015 `gosub 40000`
 Zeile 10050 `gosub 50000: gosub 8000`

In den Programmen "Listen" und "Inventar" erübrigt sich das Indexieren, da hier ohnehin alle Daten nacheinander abgerufen werden.

```
40000 rem index
40010 open 1,8,2,"0:indexfile,s,r":a%=0
40020 input#1,n$:if asc(n$)=10 then n$=right$(n$,len(n$)-1)
40030 if n$="eof" then 40050
40040 a$(a%)=n$:a%=a%+1:goto 40020
40050 close 1: return
50000 if len(a$)<6 then a$="0"+a$:goto 50000
50010 for i=0 to a%-1: if a$(i) < a$ then next i: i=i-1
50020 f%=i: return
```

40010 Eröffnen des Indexfile
 40020 - 40050 Einlesen des Indexfile

50000 Jede eingelesene Zahl wird auf die Stringlänge 6 gebracht, ansonsten mit Nullen aufgefüllt.
 50010 Durch Vergleich wird die Filenummer ermittelt
 50020 f% wird gleich i gesetzt, Rückkehr ins Programm

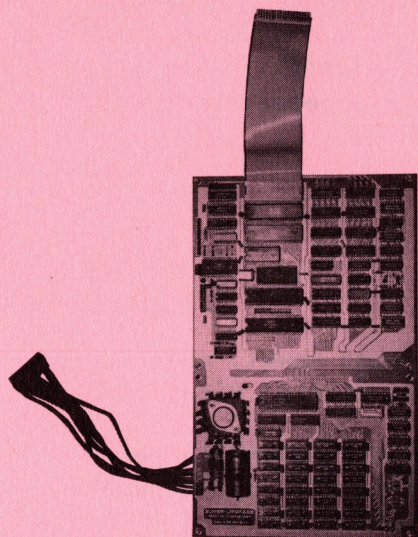
Im nächsten Heft wollen wir uns den Sortier Routinen zuwenden, um die Artikelnummern der Grösse nach zu ordnen, bzw Namen nach den Anfangsbuchstaben zu sortieren.

spina COMPUTER

SUPER- UPGRADE-SYSTEM

256 K

für alle **CBM** (**CBM 8096 kompatibel**)



Das SUPERUPGRADE ist ein zusätzliches Board, welches innerhalb des CBM 8032, ohne Lötarbeiten eingebaut wird, und das Ihrem CBM 8032 erweiterte RAM-Kapazität bringt.

In der Grundstufe wird das SUPERUPGRADE mit 64k RAM geliefert. Die erforderlichen Plätze für 128k, 192k oder 256k sind bereits auf dem Board als Stecksocket ausgebildet, so dass eine spätere Nachrüstung jederzeit vorgenommen werden kann. Es finden die neuen dynamischen RAMs mit 64kx1 bit Verwendung.

Mit dem SUPERUPGRADE können Sie den CBM 8032 als RAM-Rechner betreiben. Er kann sich somit, nachdem das entsprechende Betriebssystem geladen wurde, wie ein alter PET oder wie ein CBM 3032 (auch mit 80 Zeichen Bildschirm) verhalten!

Eigene Betriebssystemveränderungen sind hiermit auf einfache Weise möglich.

Das SUPER UPGRADE SYSTEM eignet sich auch für CBM 3032 + 4032

Die wichtigsten Vorteile:

-- 64k RAM zusätzlichen Speicherraum -- aufrüstbar bis 256k RAM -- Kompatibilität zu 64k Board (zum CBM 8096) -- nachrüstbar mit dem Arithmetik Processor 8231 von INTEL -- Option für REAL TIME Clock -- Verwendung des Processor 6809 von MOTOROLA mit interner 16 Bit Verarbeitung und der umfangreichen MOTOROLA-Software (Assembler, Basic, Fortran, Pascal, Forth, APL etc.) -- Parallele 20 Bit Schnittstelle (6522).

Art.Nr.

Preis

2200	SUPER UPGRADE 64k	DM 1'690.--
2210	SUPER UPGRADE 128k	DM 1'910.--
2220	SUPER UPGRADE 256k	DM 2'350.--

Nachrüstbare Optionen:

2230	REAL TIME Clock für Superupgrade	DM 380.--
8110	INTEL 8231 Arithmetik Processor (mit erforderlichem Zubehör)	DM 500.--
8094	MC 6522	DM 30.--
8120	MC 6551 ACIA (mit erforderlichem Zubehör)	auf Anfrage
2240	Adapterkabel zum Umsetzen Stiftleiste/RS 232 Stecker	DM 80.--
	MC 6809 Motorola (mit erforderlichem Zubehör)	DM 49.50

Die genannten Preise sind Netto + MWSt.

Im Lieferumfang des SUPER UPGRADE SYSTEMS ist ausser der Demodiskette und dem deutschen Handbuch, ebenfalls ein komfortables Dateiverwaltungs-Grundprogramm (INSAM) 8.0 in Maschinsprache enthalten.

Händler gesucht!

Turbinenstraße 4 · 6800 Mannheim 31 · West-Germany
Telefon 0621 / 72 15 15 · Telex 463708 spima d

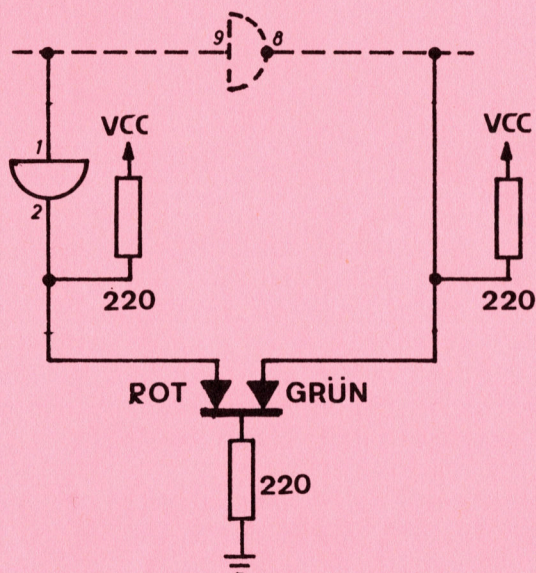
Hardware

FLOPPY DISK ERWEITERUNG

Welchem Besitzer eines Floppy 2040, 3040 oder 4040 ist es nicht schon einmal passiert, dass er vergessen hat, sein Gerat abzuschalten, da man bei diesen Typen den Einschaltzustand nicht kontrollieren kann, wie dies beispielsweise beim Modell 8050 der Fall ist. Beim letztgenannten Floppy Disk brennt eine grüne Leuchtdiode, wenn das Gerat eingeschaltet ist, und die Farbe wechselt nach Rot sobald ein Fehler vorliegt.

Es ist nun kein Problem, aeltere Gerate abzuendern, dass Sie den gleichen Komfort bieten, wie die Gerate der 8000 er Serie. Selbst CBM Besitzer, die nicht gerade über grosse handwerkliche Faehigkeiten verfügen, sind in der Lage, den Einbau dieser Erweiterung vorzunehmen, wenn die beschriebenen Anweisungen beachtet werden.

Die beschriebene Schaltung geht von der Voraussetzung aus, dass die grüne Leuchtdiode immer dann brennen muss, wenn das Gerat eingeschaltet ist, und kein Fehler vorliegt, also die rote Leuchtdiode nicht brennt. Die eingebaute rote Leuchtdiode wird über einen invertierenden Treiber mit offenem Kollektor SN 7406 angesteuert. Wenn man nun das Signal am Eingang dieses IC abnimmt, liegen genau die oben beschriebenen Verhaeltnisse vor. Das Signal zur Ansteuerung des grünen LED wird über einen Treiber SN 7407 erzeugt. Die Widerstaende von 220 Ohm dienen vornehmlich der Strombegrenzung der LED.

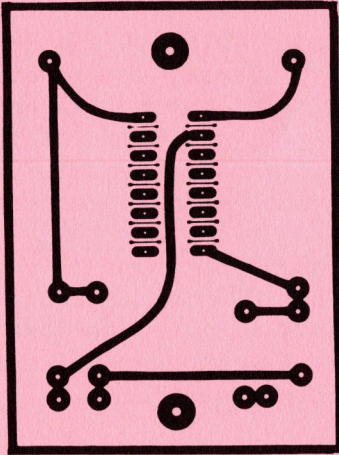


Schaltbild

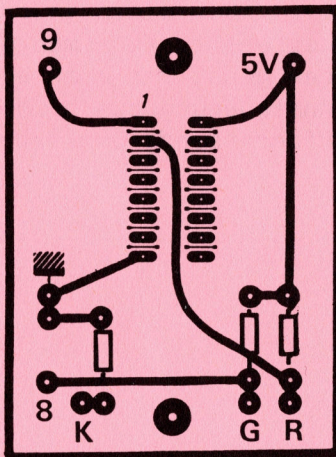
Zur Installation dieser Erweiterung sind folgende Arbeiten erforderlich

1. Bestückung der Hilfsplatine und deren Einbau an der Unterseite des Deckels.
2. Anschluss des roten und des schwarzen Drahtes am Kondensator C 27 zur Stromversorgung der Hilfsplatine (roter Draht + 5 V schwarzer Draht Masse).

3. Anschluss des gelben und des grünen Drahtes an IC 7406.
Gelber Draht am Anschluss 8 und grüner Draht am Anschluss 9.
4. Ausbau des vorhandenen roten LED und Ersatz durch ein zweifarbiges LED.



Print
Lötseite



Print
Bestückungsseite

Der Ausbau des roten ERROR LED ist sehr einfach, da es mittels eines zweipoligen Steckers mit der Platine des Floppy verbunden ist. Wichtig ist vor allem, dass alle Arbeiten am Floppy bei herausgezogenem Netzstecker und mit einem guten, geerdeten LötKolben erfolgen.

Die fertig bestückte Hilfsplatine mit allen Anschlüssen und dem zweifarbigem LED ist zum Preis von Fr. Fr. 35.-- durch den

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

Lieferbar. Wir bitten Sie, Ihre Bestellung bis spätestens 1. Dezember 1982 aufzugeben. Die Auslieferung der Platinen erfolgt dann ab Mitte Dezember. Sofern Sie die Bauteile selbst beschaffen, achten Sie unbedingt darauf, dass die Leuchtdiode eine gemeinsame Kathode besitzt.

Programm des Monats BASIC-BASIC

BERECHNUNG VON SIEDEKURVEN

Während in den vergangenen Ausgaben entweder Spielprogramme oder Programme aus dem Bereich der Elektronik zur Wiedergabe gekommen sind, soll heute einmal ein physikalisches Programm besprochen werden. Es handelt sich um die Berechnung des Siedepunktes eines Stoffes bei allen Luftdrücken.

Es ist sicher bekannt, dass ein Stoff desto niedriger siedet, je tiefer der Luftdruck ist. Beispielsweise siedet Wasser auf dem Jungfraujoch (3030 m Meereshöhe) bereits bei 80 Grad Celsius, da in dieser Höhe der Luftdruck nur noch 700 mm Hg beträgt.

Es lassen sich nun alle Siedepunkte eines Stoffes bei allen Drücken zwischen 760 mm Hg und 0 mm Hg berechnen, wenn der Siedepunkt des Stoffes bei Normaldruck und seine molare Verdampfungswärme bekannt sind. Unter der molaren Verdampfungswärme versteht man die Anzahl Kalorien, die erforderlich sind, ein Mol eines Stoffes zu verdampfen, also z.B. 18 g Wasser von 100 Grad Celsius in Dampf von 100 Grad Celsius zu überführen. Die Verdampfungswärme kann aus einschlägigen Tabellen entnommen werden, wir geben Ihnen aber die Werte einiger typischer Stoffe an:

Wasser	ca. 10000 cal/mol
Alkohol	9300 cal/mol
Aceton	7230 cal/mol
Chloroform	7020 cal/mol

Der Berechnung liegt die Formel von Clausius-Clapeyron zugrunde

$$T1 = \frac{L}{4.57 \lg \frac{p2}{p1} + \frac{L}{4.57 * T2}}$$

hierin bedeuten

L = Verdampfungswärme in cal/mol (mittlerer Wert in der Umgebung des Siedepunktes T2)

p1 = vorhandener Druck in mm Hg

p2 = Normaldruck (760 mm Hg)

T1 = gesuchte Siedetemperatur in Grad Kelvin

T2 = Siedetemperatur beim Druck p2 in Grad Kelvin

Das Programm ist vielleicht auf den ersten Blick für viele uninteressant, da diese Anwendung nur selten benötigt wird. Trotzdem können aus diesem Programm Hinweise für eine ganze Reihe Anwendungen entnommen werden, da das Programm sehr schön zeigt, wie aus einer Wertetabelle eine Kurve auf einem normalen Printer oder einem Plotter gemacht werden kann. Das Programm errechnet 36 (Printer) bzw. 72 Werte (Plotter) des gesamten Druckbereichs von 0 - 760 mm Hg und zeichnet die Kurve wahlweise auf einem Printer oder einem Plotter. Als Printer kann nahezu jeder Typ verwendet werden, da im Programm nur zwei printerspezifische Befehle verwendet worden sind, die aber eliminiert werden können, beim Plotter ist dagegen nur der WATANABE Plotter MX 222 möglich.

SIEDEKURVE VON WASSER

MOL. VERD. WAERME : 10000

KP760 MM : 100

DRUCK IN MM HG

760
740
720
700
680
660
640
620
600
580
560
540
520
500
480
460
440
420
400
380
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0



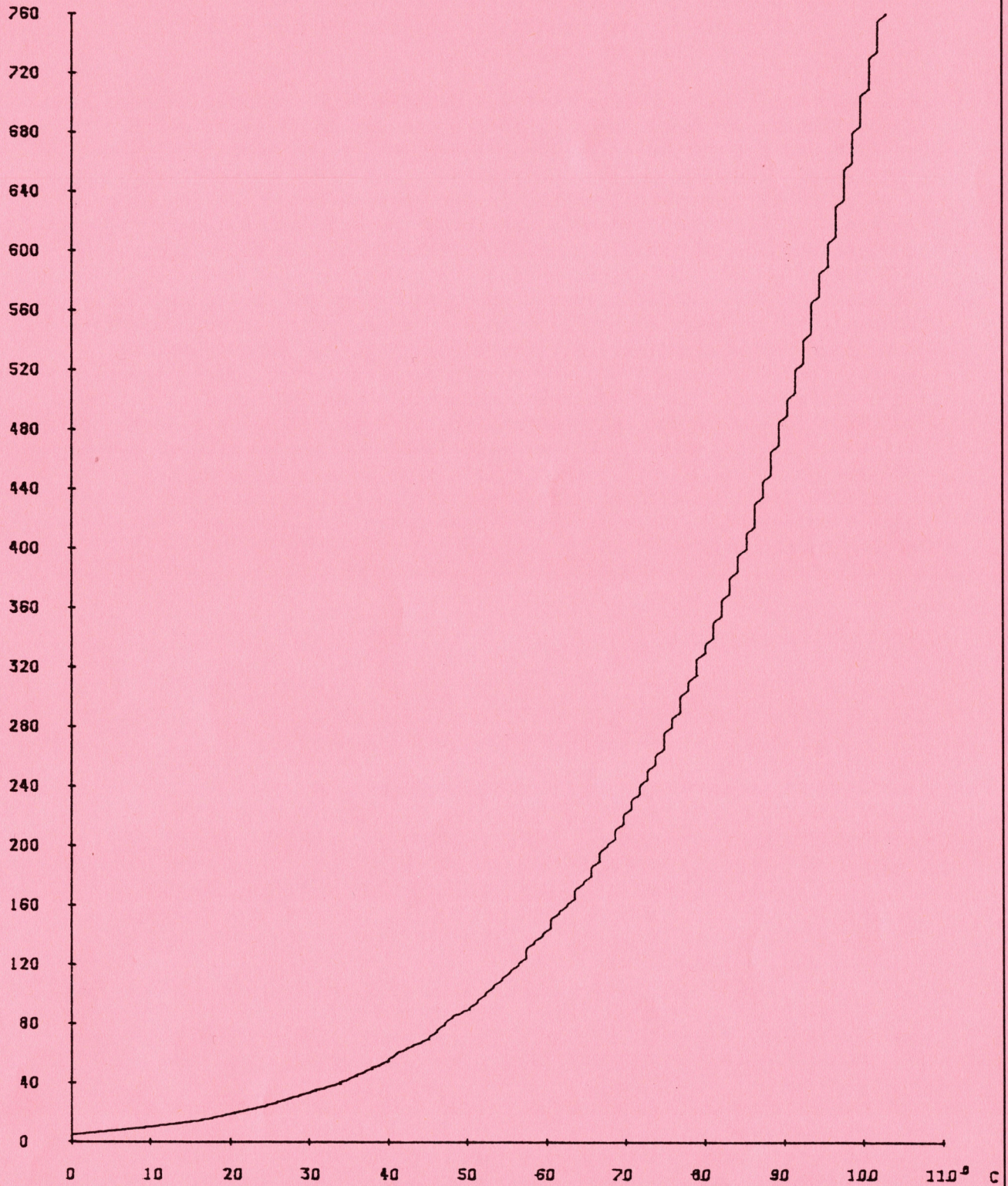
Ausdruck Printer

SIEDEKURVE VON WASSER

DRUCK IN MM HG

KP 760 MM = 100

MÖL. VERD. WÄERME = 10000



Ausdruck Plotter

Das Programm gliedert sich in 4 Teile

1. Einleitung und Berechnung für einen Punkt der Kurve
2. Ausgabe der gesamten Kurve auf einem Printer
3. Ausgabe der gesamten Kurve auf einem Plotter
4. Ausgabe der gesamten Kurve auf dem Bildschirm mit hochauflösender Graphik

Das Programm laeuft unveraendert auf dem CBM 3000 und 8000. Ebenso kann jeder Printer verwendet werden, sofern von der Möglichkeit der Breitschrift und des variablen Zeilenabstandes abgesehen wird. Auf diese speziellen Eigenheiten, die nur mit einem CBM - Printer möglich sind, wird in der Programmbeschreibung in den entsprechenden Zeilen hingewiesen. Die Plotterroutine ist allerdings nur für den Watanabe Plotter vorgesehen. Ebenso ist die Ausgabe mittels hochauflösender Graphik nur auf dem CBM 8000 möglich, allerdings kann das Programm leicht auf die HRG des CBM 3000 umgeschrieben werden. Die Programmerweiterung für hochauflösende Graphik erscheint in Heft 6/82, jedoch sind die erforderlichen Verzweigungen bereits in diesem Programm vorgesehen. Der wesentlichste Unterschied zwischen den einzelnen Programmen liegt in der Auflösung der Kurve, waehrend auf dem Printer nur eine Auflösung von $60 * 76 = 4560$ Punkten möglich ist, wird beim Plotter eine A4 Seite mit einer Genauigkeit von 0.1 mm also 30400 Punkten aufgelöst, bei der HRG auf dem Bildschirm ist theoretisch eine Genauigkeit von $256 * 512$ also 131072 Bildpunkten möglich, hier wird allerdings dann die Rechnung derart umfangreich, dass man von dieser hohen Auflösung wohl kaum Gebrauch machen wird.

SIEDEKURVE

```
100 REM ABHAENGIGKEIT DES SIEDEPUNKTES VOM DRUCK
110 REM PROGRAMMIERT H. KASTIEN 26.02.1981
120 POKE59468,14:IFPEEK(50003)=160THENX=20
130 DIM K(154)
140 PRINT"TITEL";TAB(X+13)"ABHAENGIGKEIT"
150 PRINTTAB(X+7)"DES SIEDEPUNKTES VOM DRUCK"
160 PRINTTAB(X+1)"FUR IERECHNUNG DES SIEDEPUNKTES BEI"
170 PRINTTAB(X+1)"GEGEBENEM DRUCK WIRD DIE GLEICHUNG VON"
180 PRINTTAB(X+1)"DE LAUSIUS--LAPEYRON HERANGEZOGEN. TS"
190 PRINTTAB(X+1)"MUSS HIERZU DIE MOLARE VERDAMPFUNG-"
200 PRINTTAB(X+1)"WAERME BEIM SIEDEPUNKT BEKANNT SEIN."
210 FOR I=1TO3000:NEXT
220 PRINT"TITEL";TAB(X+3)"GEBEN SIE DIE CHEMISCHE FORMEL ODER"
230 PRINTTAB(X+3)"DEN NAMEN DER SUBSTANZ AN : ":INPUTS$
240 PRINTTAB(X+3)"GEBEN SIE DIE MOLARE VERDAMPFUNG -"
250 PRINTTAB(X+3)"WAERME VON "S$" AN :";
260 INPUTM
270 PRINTTAB(X+3)"SIEDEPUNKT VON "S$:" BEI 760 MM IG"
280 PRINTTAB(X+3)"IN GRAD CELSIUS : ":INPUTKP:KP=KP+273
290 PRINT"TITEL";TAB(X+10)"WENN SIE WUENSCHEN SIE ***"
300 PRINTTAB(X+6)"1. DEN SIEDEPUNKT BEI EINEM DRUCK"
310 PRINTTAB(X+6)"2. KURVE AUF PRINTER"
320 PRINTTAB(X+6)"3. KURVE AUF PLOTTER"
330 PRINTTAB(X+6)"4. KURVE AUF DEM BILDSCHIRM"
340 GETA$:IFVAL(A$)=0ORVAL(A$)>4THEN340
350 ONVAL(A$)GOTO360,500,1000,2000
360 PRINT"TITEL";TAB(X+3)"BEI WELCHEM DRUCK IN MM IG WIRD ""
370 PRINTTAB(X+3)"DER SIEDEPUNKT GEWUENSCHT : ":INPUTP
```



```

380 I=1:P1=P/10:GOSUB10000:K1=K(1)+273
390 PRINTTAB(X+3)"MM";S$;" SIEDET BEI "P" MM IG"
400 PRINTTAB(X+3)"BEI ";K1;" IRAD °ELVIN,"
410 PRINTTAB(X+3)"ODDER ";K1-273;" IRAD -ELSIUS."
420 PRINTTAB(X)"MENSCHEN WIE EINE WEITERE RECHNUNG"
430 PRINTTAB(X+17)"(J/N)"
440 GETZ$:IFZ$=""THEN440
450 IFZ$="J"THEN 220
460 IFZ$="N"THEN 480
470 GOTO440
500 REM KURVE AUF PRINTER
510 PRINT"*****";TAB(X+8)"PLAUSDRUCK DER KURVE"
520 S2$="SIEDEKURVE VON "
530 S1$=S2$+S$
540 FORI=1TO76
550 P1=I:GOSUB10000:NEXT:V=K(76)/60
560 OPEN1,4
570 C#=CHR$(1)
580 PRINT#1,CHR$(1)S1$:PRINT#1
590 PRINT#1
600 PRINT#1,C#"MOL. VERD.WAERME ";M;" KP"CHR$(129);"760 MM";
610 PRINT#1:PRINT#1
620 PRINT#1," DRUCK IN MM HG" C#" ";KP-273
630 PRINT#1
640 OPEN6,4,6:PRINT#6,CHR$(18)
650 FORI=76TO0STEP-1
660 X=6:I1=I*10
670 IFI/2<>INT(I/2)THENI$=" "
680 IFI/2=INT(I/2)THENI$=STR$(I1):X=3
690 IFVAL(I$)<100THENX=4
700 IFI=0THENX=5
710 PRINT#1,TAB(X)I$;" I";
720 FOR R=0TOINT(K(I)/V)
730 PRINT#1," ";
740 NEXT
750 PRINT#1,"*"
760 NEXT
770 PRINT#1," ";
780 FORI=0TO67:PRINT#1,"-":NEXT
790 PRINT#1," ";
800 PRINT#1,""
810 PRINT#1," ";
820 FORI=0TO66STEP6
830 PRINT#1," I";
840 NEXT
850 PRINT#1," "
860 PRINT#1
870 PRINT#1," 0 ";
880 FORI=1TO66STEP6
890 K$=STR$(INT((KP-273)/10*(I/6)))
900 IFLEN(K$)=4THENK$=" "+K$:GOTO900
910 PRINT#1," ";
920 PRINT#1,K$;
930 NEXT
940 PRINT#1,""
950 PRINT#1
960 PRINT#1,TAB(52)"SIEDEPUNKT GRAD CELSIUS"
970 PRINT#6,CHR$(24):CLOSE6:PRINT#1:CLOSE1
980 END
1000 REM PLOTAUSDRUCK
1010 REM WATANABE
1011 H=M

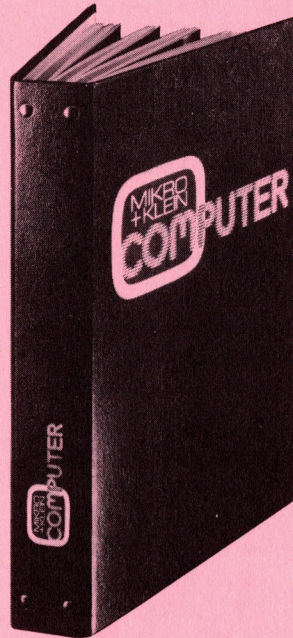
```

```

1012 GOSUB 3650
1014 Y1#:=STR$(KP-273):Y2#:=STR$(M)
1015 Y#="KP 760 MM = "+Y1#+ "      MOL. VERD. WAERME = "+Y2#
1020 X=150:Y=80
1030 GOSUB3580
1040 X=1:Y=160:R=11
1050 GOSUB3550
1060 X=150:Y=80
1070 GOSUB3580
1080 X=0:Y=120:R=19
1090 GOSUB3550
1100 X=5:Y=2430
1110 GOSUB3580
1120 S=3:X#="DRUCK IN MM HG"
1130 X9#="S":Y=S:GOSUB 3530
1135 GOSUB 3630
1136 X=500:Y=2430
1137 GOSUB3580
1138 X#=Y#
1139 X9#="S":Y=S:GOSUB3530
1140 GOSUB3630
1141 X=5:Y=2530
1150 GOSUB3580
1160 S=7:X#=S1#
1170 X9#="S":Y=S:GOSUB3530
1180 GOSUB3630
1190 X=5:Y=2350
1200 GOSUB 3580
1210 S=2:X#=" 760"
1211 X=5
1215 X9#="S":Y=S:GOSUB3530
1216 GOSUB3630
1220 FORI=0TO18
1240 Y=2230-(I*120)
1250 GOSUB3580
1260 M=VAL(X#)-40:X=5
1265 IFM<130THENX=25
1266 IFM<40THENX=65
1270 X#:=STR$(M)
1280 X9#="S":Y=S:GOSUB 3530
1290 GOSUB3630
1300 NEXT
1310 X=120:Y=5
1320 GOSUB 3580
1330 X#=" 0"
1331 X9#="S":Y=S:GOSUB3530
1332 GOSUB3630
1340 FORI=1TO11
1350 Y=5:X=120+(I*158)
1360 GOSUB 3580
1370 N=VAL(X#)+INT((KP-273)/10)
1380 X#:=STR$(N)
1390 X9#="S":Y=S:GOSUB3530
1400 GOSUB3630
1410 NEXT
1420 X=1950:Y=20
1430 GOSUB 3580
1440 S=1:X#="0"
1450 X9#="S":Y=S:GOSUB 3530
1460 GOSUB 3630
1470 S=2:X#=" C"
1480 X9#="S":Y=S:GOSUB 3530
1490 GOSUB 3630

```

**Jetzt
wieder
erhältlich!**



der praktische Sammelordner

mit einem strapazierfähigen Kunststoffüberzug in ansprechender blauer Farbe und bequemer Stabmechanik für jeweils sechs Nummern **Mikro + Kleincomputer** (also ein ganzer Jahrgang), damit jedes einzelne Heft unbeschädigt bleibt.

Diesen praktischen Sammelordner können Sie für nur Fr. 14.50 inkl. Versandkosten bestellen. Übrigens, zwei Exemplare kosten bei gleichzeitiger Bestellung nur noch Fr. 27.-. Zahlen Sie bitte den entsprechenden Betrag auf unser **Postkonto Luzern 60-27181** ein und vermerken Sie auf der Rückseite Ihres Einzahlungsscheins «Sammelordner».

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

```

1495 X=150:Y=80
1496 GOSUB 3580
1500 FORI=1TO152
1510 P1=I:GOSUB10000
1515 NEXT
1520 FORI=1TO152
1525 IFL(I)<=0THENL(I)=0
1530 X=L(I)*1640/L(152)+150:Y=I*15+80
1540 GOSUB 3570
1550 NEXT
3510 END
3520 REM ZEICHEN SUBROUTINEN
3530 X8#=X9#+STR$(INT(Y))+CHR$(10)
3540 GOT03690
3550 X8#="X"+STR$(INT(X))+", "+STR$(INT(Y))+", "+STR$(INT(R))
3560 GOT03690 +CHR$(10)
3570 X8#="D":GOT03610
3580 X8#="M":GOT03610
3590 X8#="I":GOT03610
3600 X8#="R"
3610 X8#=X8#+STR$(INT(X))+", "+STR$(INT(Y))+CHR$(10)
3620 GOT03690
3630 X8#="P"+X#+CHR$(10)
3640 GOT03690
3650 POKE59459,255
3660 POKE59457,0
3670 POKE59457,128
3680 X8#="H"+CHR$(10)
3690 FORX8= 1TOLEN(X8#)
3700 IFPEEK(59469)AND2THEN3720
3710 GOT03700
3720 POKE59457,0
3730 POKE59457,ASC(MID$(X8#,X8,1))+128
3740 NEXTX8
3750 RETURN
3760 RETURN
3770 END
10000 REM RECHNUNG
10010 K(I)=INT(M/4.57/((LOG(760/P1/10)/LOG(10))+M/4.57/KP)-273)
10020 RETURN

```

Programmbeschreibung

- 100 - 110 Remark
- 120 - 130 Dimensionierung der Variablen, Definition von Variablen und Abfrage nach dem Rechnertyp (PEEK 50003)
- 140 - 200 erklärender Text
- 210 Warteschleife
- 220 - 280 Eingabe der Variablen
- 290 - 350 Menu und berechneter Sprungverteiler
- 360 - 470 Abfrage des Drucks, Berechnung des Siedepunktes und Ausgabe auf dem Bildschirm, Rückkehr ins Menu
- 500 Remark
- 510 - 530 Definieren von Variablen
- 540 - 550 Berechnen von 76 Werten des Kochpunktes zwischen 0 und 760 mm Hg. als indizierte Variable k(i).
- 550 Die Schreibbreite des Printers wird mit 60 Zeichen festgelegt und aus dem höchsten errechneten Siedepunkt k(76) durch Division mit 60 der Faktor v errechnet, der angibt, wieviel Grad Celsius einer Printerstelle entsprechen.

- 560 - 570 Eröffnen des Druckfile und Umschalten in den Command Modus, c\$ wird gleich CHR\$(1) gesetzt. CHR\$(1) schaltet einen Commodore Drucker auf Breitschrift um, CHR\$(129) schaltet wieder zurück in Normalschrift. Bei anderen Druckertypen müssen diese Charakter geändert werden oder können völlig entfallen.
- 580 - 630 Ausdruck des Diagrammkopfes
- 640 Der Drucker wird auf einen Zeilenabstand von 18 Zeilen pro Zoll (normal 24 Zeilen pro Zoll) umgeschaltet. Dieser Befehl muss bei anderen Druckern ebenfalls geändert werden oder kann weggelassen werden.
- 650 - 760 Es erfolgt der Ausdruck der eigentlichen Kurve und gleichzeitig der y-Achse mit der Beschriftung durch eine FOR... NEXT Schleife, die mit dem höchsten Wert beginnt.
- 660 Setzen von Variablen (x=Abstand der Y-Achse und Beschriftung vom linken Rand. i1=Umrechnungsfaktor.
- 670 Wenn der Index der Variablen k nicht durch 2 teilbar ist, ist i\$ gleich drei Space.
- 680 Wenn der Index der Variablen K durch 2 teilbar ist, ist i\$ gleich dem zehnfachen Wert des Index, das entspricht dem Druck, es wird mit diesen beiden Zeilen erreicht, dass nur jeder 2. Luftdruckwert auf dem Printer ausgedruckt wird, damit wird die Graphik übersichtlicher.
- 690 - 700 Je nach Grösse des Luftdruckwertes wird die Anzahl der Leerstellen (x) gewählt.
- 710 Es erfolgt der Ausdruck des Luftdruckwertes (bzw. Space) und entsprechender Leerstellen sowie eines senkrechten Strichs.
- 720 - 740 Der Printerkopf wird um so viele Stellen nach rechts verschoben, wie die Variable k(i) dividiert durch v angibt.
- 750 - 760 Es wird ein "*" hinter die Leerstellen gesetzt und der nächste Wert von k(i) aufgerufen.
- 770 - 800 Es wird die X - Achse gezeichnet, wobei Sie um 7 Stellen nach rechts eingerückt wird.
- 810 - 850 Die X-Achse wird mit 11 senkrechten Strichen markiert.
- 860 - 950 Aus den errechneten Siedepunktswerten wird die Beschriftung der x-Achse vorgenommen. Die Berechnung der Zahlen erfolgt in Zeile 890.
- 960 Ausdruck eines abschliessenden Textes
- 970 Zurückschalten des Printers auf 24 Zeilen pro Zoll, Schliessen aller File und Rücksprung ins Menu

Das Programm zum Ausdruck auf dem Plotter ist prinzipiell identisch mit dem Ausdruck auf dem Printer, jedoch sind hier die Plottbefehle als Subroutinen aufgebaut. Diese Subroutinen sind dem Watanabe Manual entnommen und befinden sich in den Zeilen 3500 - 3750. Um das Programm möglichst leicht verständlich zu machen, soll diese Subroutine zuerst besprochen werden. Die Ansteuerung des Plotters erfolgt über den USER - PORT, daher muss dieser in jedem Fall als Ausgang geschaltet sein.

- 3530 - 3540 Definition der Schriftgrösse eines auszudruckenden Textes durch die Variable s=1 bis s=15
- 3550 - 3560 Befehl zum Zeichnen einer X-Achse vom Punkt x,y mit der Einteilung R
- 3570 Zeichnen einer Linie vom Punkt des Schreibers zur Position x,y
- 3580 Bewegung des Zeichenkopfes mit abgehobenem Zeichenstift von der momentanen Position zur Position x,y
- 3590 Relative Bewegung des Zeichenkopfes mit den Parametern x,y
- 3600 Relatives Zeichnen einer Linie mit den Parametern x,y
- 3630 Befehl zum Schreiben eines Textes, der in der Stringvariablen x\$ gespeichert ist.

3740 - 3750 RESET des Plotters und Verarbeitung des Befehls sowie Rückmeldung an den Rechner, dass der Befehl ausgeführt ist.

Nach der Besprechung der Plottersubroutinen kann nun die eigentliche Programmbeschreibung für den Plotterausdruck folgen :

1000 - 1010 Remarks
1011 - 1015 Definition von Variablen und RESET des Plotters
1020 - 1030 Plotter wird in Ausgangsstellung gefahren
1040 - 1050 Zeichnen der X-Achse
1060 - 1070 Plotter wird in Ausgangsstellung gefahren
1080 - 1090 Zeichnen der Y-Achse
1100 - 1180 Titel schreiben
1190 - 1300 Beschriftung der Y-Achse
1310 - 1490 Beschriftung der x-Achse
1495 - 1496 Plotter wird in Ausgangsstellung gefahren
1500 - 1550 Berechnen von 152 Werten und Zeichnen der Werte auf dem Plotter

Variable im Plotterprogramm

x und y Koordinaten des Plotters in 1/10 mm bezogen auf den Startpunkt
s Zeichengrösse
x\$, x9\$ Hilfsvariable

Wussten Sie schon...

CURSORSTEUERUNG

dass es mittels eines kleinen Maschinenprogramms sehr einfach ist, den Cursor an jede beliebige Stelle des Bildschirms zu bringen. Natürlich ist dies auch mittels Cursorbewegungen oder POKE möglich, das nachfolgende Maschinenprogramm ist jedoch wesentlich schneller und eleganter. Um den Cursor zu programmieren muss lediglich SYS 826, Zeile, Spalte eingegeben werden. Dieses kleine Programm kann also von jeder beliebigen Stelle eines BASIC-Programms aufgerufen werden.

```
1000 REM *** CURSORSTEUERUNG ***
1010 REM HUBERT MIHLAN BURGHAUSEN
1020 REM CBM 3000
1030 A=826
1040 READ P:IF P<0 THEN END
1050 POKE A,P:A=A+1:GOTO 1040
1060 :
1070 DATA 32,248,205,32,120,214,224,1,48,29
1080 DATA 224,26,16,25,134,216,169,145,32,210
1090 DATA 255,32,248,205,32,120,214,224,1,48
1100 DATA 8,224,41,16,4,202,134,198,96,169
1110 DATA 13,32,210,255,162,53,76,106,195,170,-1
READY.
```

Maschinensprache

HEX - BCD UMWANDLUNG

Vielfach möchte man gerne irgend einen Zaehlerstand oder den Inhalt eines Timers in dezimaler Form darstellen. Dies ist nur durch eine Umrechnung von Binaer in Dezimal, also von HEX in BCD zu bewerkstelligen.

Das hier gezeigte Verfahren kann für allgemeine Verfahren verwendet werden, respektive angepasst werden. Es arbeitet schnell und ist komfortabel.

Das im Listing gezeigte Programm wandelt eine maximal 8 Digit HEX-Zahl in die entsprechende Dezimalzahl um. Die Stellenzahl laesst sich einfach reduzieren oder erweitern. Will man das Programm für andere Stellenzahlen verwenden, so ist folgendes zu unternehmen:

Die beiden Datenfelder BCD und HEX sind neu zu definieren. Zum Beispiel: BCD wird nur 3 Bytes gross, HEX nur 2 Bytes

$$\text{HEX} = \text{BCD} + 3 \quad \text{EXH} = \text{HEX} + 2$$

Dies hat dann automatisch Einfluss auf die beiden Zaehler in den Zeilen 13 und 14. Die Zeilen 11 und 12 entfallen, und die "ROTATE LEFT" Befehle in den Zeilen 15 und 23 müssen entsprechend angepasst werden (neue einsetzen oder streichen).

Da die Stellenzahl der HEX-Zahlen meist kleiner ist als die der Dezimalzahlen, muss für diese genügend Speicherplatz reserviert werden.

Die grösste Dezimalzahl kann errechnet werden mit: $16^n - 1$, wobei n die maximale Digitzahl der HEX-Zahl ist. Im Programmbeispiel ist das HEX-Feld 4 Bytes gross, n wird somit 8 und die grösste Dezimalzahl $4.2949 \cdot 10^9$ oder (4.2949E9).

Die Anzahl der nötigen BCD-Bytes kann wie folgt ermittelt werden:

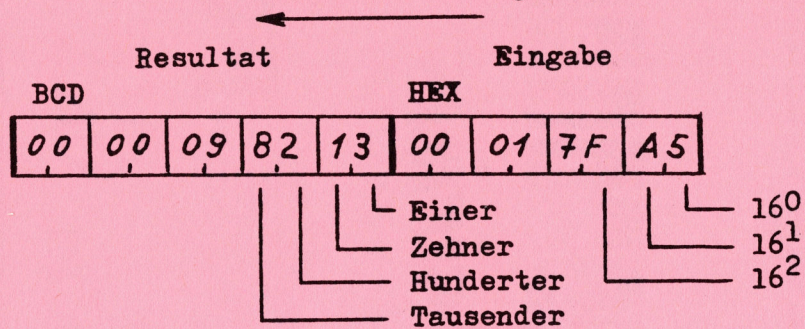
$$\text{Anzahl BCD-Bytes} = \text{INT}(0.5 + 1.024 * \text{Anzahl HEX-Bytes})$$

Im Programm wird das Indexregister Y mit diesem Wert - 1 gesetzt und dient als Zeiger auf das jeweils zu prüfende Byte des Ergebnisfeldes BCD, siehe Zeile 14.

Das Programm arbeitet nach einem festen Algorithmus. Die HEX-Zahl wird, beginnend mit dem höchstwertigen Bit, von rechts nach links in das direkt nebenanliegende BCD-Feld geschoben. Ueberschreitet eine BCD Stelle (Digit) den Wert 4 (0100), so muss vor dem naechsten Schiebevorgang eine 3 an dieser Stelle addiert werden. Schiebevorgang und, falls nötig, Korrektur wechseln sich ab, bis das letzte Bit der HEX-Zahl in das BCD-Feld geschoben ist. Hier steht dann das Ergebnis in der richtigen Reihenfolge.

Beispiel: Die Vorgabe von 17FA5 HEX ergibt
das Resultat 98213 Dezimal.

Bit-Schieberichtung



 * CBM * ASSEMBLER 6502 * HEXTOBCD.LST PAGE 1

```

0380          1  *=896
0380          2  !
0340          3  BCD = 832
0345          4  HEX = BCD+5
0349          5  EXH = HEX+4
0380          6  !
0380  A900          7  START LDA #0          ;DAS RESULTAT-FELD IST
0382  8D4003          8          STA BCD          ; MIT 0 ZU LOESCHEN
0385  8D4103          9          STA BCD+1
0388  8D4203         10          STA BCD+2
038B  8D4303         11          STA BCD+3
038E  8D4403         12          STA BCD+4
0391  A21F          13          LDX #EXH-HEX*8-1      ;BIT-ZAEHLER   HEX-FELD
0393  A004          14  L10    LDY #HEX-BCD-1      ;BYTE-ZAEHLER  BCD-FELD
0395  0E4803         15          ASL HEX+3
0398  2E4703         16          ROL HEX+2
039B  2E4603         17          ROL HEX+1      ;VERSCHIEBE DAS GANZE HEX-
039E  2E4503         18          ROL HEX          ; UND BCD-FELD UM 1 BIT
03A1  2E4403         19          ROL BCD+4      ; NACH LINKS
03A4  2E4303         20          ROL BCD+3
03A7  2E4203         21          ROL BCD+2
03AA  2E4103         22          ROL BCD+1
03AD  2E4003         23          ROL BCD
03B0  E000          24          CPX #0          ;SCHIEBEVORGANG BEENDET?
03B2  F024          25          BEQ L50          ;SPRINGE WENN JA
03B4  B94003         26  L20    LDA BCD,Y      ;PRUEFEN OB EIN UEBERTRAG INNER-
03B7  48            27          PHA          ; HALB VOM BCD-BYTE VORLIEGT;
03B8  290F          28          AND #15         ;MASKIERE DIE OBEREN 4 BIT WEG
03BA  C905          29          CMP #5         ;IST (A)<=4?
03BC  9002          30          BCC L30         ;SPRINGE WENN JA
03BE  6902          31          ADC #3-1       ;ADDIERE 2+CARRY
03C0  994003         32  L30    STA BCD,Y      ;WIEDER ZURUECK LEGEN
03C3  68            33          PLA          ;RESTORE ALTER BCD-WERT
03C4  29F0          34          AND ##F0      ;MASKIERE DIE UNTEREN 4 BIT WEG
03C6  C950          35          CMP ##50      ;IST (A)<=#40?
03C8  9002          36          BCC L40         ;SPRINGE WENN JA
03CA  692F          37          ADC ##30-1     ;ADDIERE #2F+CARRY
03CC  194003         38  L40    ORA BCD,Y      ;SETZE DIE BEIDEN BCD-ZAHLEN
03CF  994003         39          STA BCD,Y      ; ZUSAMMEN UND SPEICHERE ZURUECK
03D2  88            40          DEY          ;ALLE BCD BYTES BEARBEITET?
03D3  10DF          41          BPL L20        ;SPRINGE WENN NEIN
03D5  CA            42          DEX          ;ALLE HEX-BITS DURCHGESCHOBEN?
03D6  10BB          43          BPL L10        ;SPRINGE WENN NEIN
03D8  60            44  L50    RTS
03D9          45  !
03D9          46  .END
  
```

9 SYMBOLS 0 ERRORS 89 BYTES

BCD	0340	EXH	0349	HEX	0345	L10	0393
L20	03B4	L30	03C0	L40	03CC	L50	03D8
START	0380						

DCT

Kleincomputer

...und alles, was dazugehört

Commodore
COMPUTER

DCT-SUPERBRAIN

 **MONROE**
Systems For Business

 **apple computer**

 **HEWLETT
PACKARD**

Zubehör

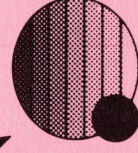
- Disketten
- Farbbänder
- Bücher

Peripherie

- Drucker/Plotter
- Floppies
- Hard-Disk



digital



Sirius
COMPUTER



apple computer



**HEWLETT
PACKARD**



**Mikrocomputer
Schulungs-Center**

IBM-PC



Fordern Sie den ausführlichen Katalog an beim

Computer Shop Luzern

DIALOG COMPUTER
TREUHAND AG
Seeburgstrasse 18
6002 Luzern

Telefon 041 - 31 45 45

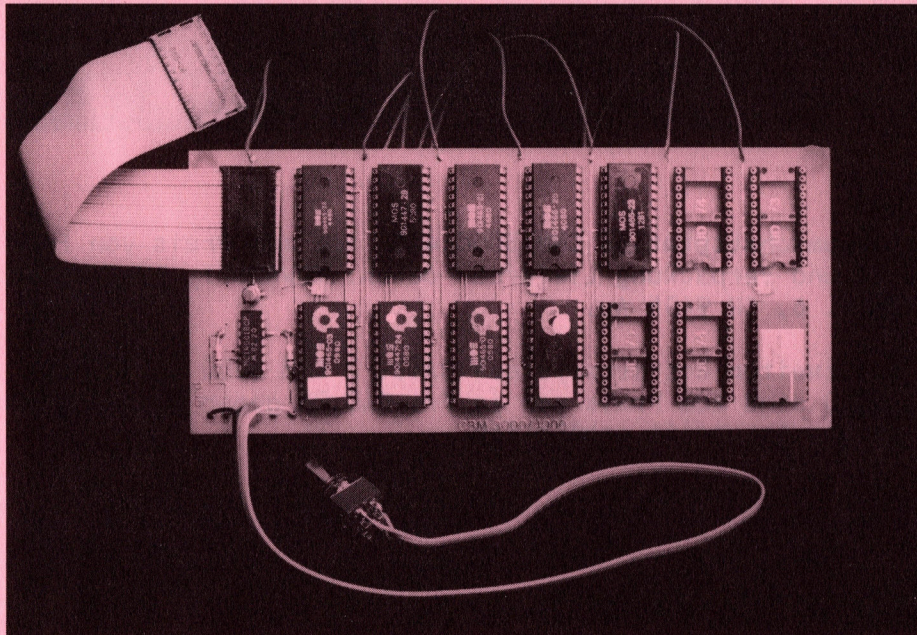
Neuheiten

BETRIEBSSYSTEMERWEITERUNG CBM 3000/4000

Seit dem Erscheinen der CBM 4000 und 8000 er Rechner hat sich schon mancher Besitzer eines CBM 3000 Rechners gewünscht, seine Maschine auf das neue Betriebssystem aufzustocken, vor allem, um mit dem erweiterten DOS arbeiten zu können. Die nachfolgende beschriebene Platine erlaubt

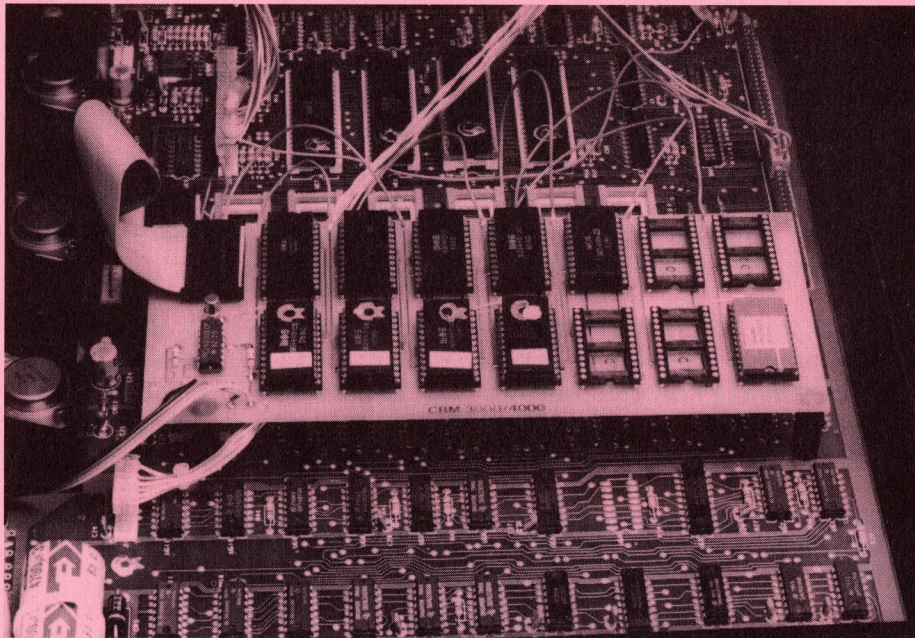
- den Betrieb des 3000 und 4000 er Systems im gleichen Rechner.
- dadurch werden die ROM um 100 % erweitert.
- es müssen keine ROM oder TOOLKIT umgesteckt werden.
- durch eine Schalter erfolgt ein Systemreset und es kann sofort mit dem anderen Betriebssystem weiter gearbeitet werden.
- die freien Plaetze für Ihre ROM und TOOLKIT bleiben erhalten.
- der Einbau der Erweiterung ist sehr einfach.

Die Systemplatine 3000/4000 übernimmt die ganze Reihe der Steckplaetze vom Originalrechner CBM 3000 (UD3 bis UD9 7 Steckplaetze). Saemtliche Signal- und Adressleitungen werden von den Originalsteckplaetzen durch eine 24 poliges Anschlusskabel übertragen. Die auf Computerplatine vorhandenen Systemerweiterungstecker J4 und J9 werden nicht belegt.



Der Einbau der Platine in den Rechner ist denkbar einfach. Bei ausgeschaltetem Rechner werden die ROM und TOOLKIT aus dem Rechner entfernt und in der gleichen Reihenfolge auf der Erweiterungsplatine eingesteckt das gleiche erfolgt mit dem zweiten Betriebssystem. Letzters muss auf der hinteren Seite der Platine neben den Anschlusskabeln liegen. Die Platine wird so in den Rechner eingesetzt, dass sie über den beiden IC Reihen vor den Steckplaetzen zu liegen kommt. Der 24 polige Stecker wird in den linken Platz der Computerplatine eingesteckt. Es müssen nun noch die 6 kurzen Draehnte jeweils in den PIN 20 der nunmehr freien Sokkel UD 3 bis UD 9 eingesteckt werden und der Umschalter auf der Frontseite des Rechners herausgeführt werden.

Wird die Umschaltung des Rechners nur im ausgeschalteten Zustand durchgeführt, so ist mit diesen Arbeiten der Einbau bereits beendet. Soll aber die Umschaltung im eingeschalteten Zustand durchgeführt werden, so muss noch ein RESET ausgelöst werden. Wenn dieser noch nicht eingebaut ist, wir haben 1980 bereits diese Möglichkeiten in den CBM/PET NEWS beschrieben, muss diese RESET Leitung noch mit dem PIN 2 des NE 555 (bzw mit dem Verbindungspunkt des R16 1 MOhm und C68 .1mF) verbunden werden. Damit sind alle Arbeiten ausgeführt und der Rechner kann wieder in Betrieb genommen werden.



Die Betriebsumschaltplatine CBM 3000/4000 ist zum Preis von Fr. 148.00 inkl. Porto und Verpackung lieferbar durch: EDWASOFT, DV-Systemlösungen, Industriestr. 6, CH 9435 Au.

Die ROM des zweiten Betriebssystems BASIC 2 CBM 3000 sowie BASIC 4 CBM 4000 und 8000 sind zum Preis von je Fr. 240.00 gegen Rechnung ebenfalls bei der obigen Anschrift erhältlich.

Wussten Sie schon...

DEZIMALDATUM

dass es machmal erforderlich ist, ein Datum in eine Dezimalgrösse umzurechnen. Zum Beispiel um aus zeitabhaengigen Grössen eine graphische Darstellung zu errechnen. Das Dezimaldatum wird in einem Wort abgespeichert. Das Programm, welches als Subroutine benutzt werden sollte, enthaelt folgende Variable.

IT=Tag IM = Monat IJ = Jahr IS = Stunde IN = Minute IC = Sek.

ID ist das Dezimaldatum, das Dezimaldatum entspricht der Zeit in 1/10 Sekunden, die seit dem 1.1.1900 00.00.00 Uhr vergangen ist. So entspricht der 27.1.1982 20.50.00 Uhr dem Dezimaldatum 2.5900014E+10. Wenn die Tage des Jahres fortlaufend nummeriert werden, muss das Programm zweimal aufgerufen werden. Im ersten Durchlauf wird das Dezimaldatum errechnet, wobei als Jahr dasjenige Jahr eingegeben wird, in dem mit der Tagesnummerierung begonnen worden ist. Als Monat muss IM = 1 eingegeben werden.

DEZIMALDATUM

```

100 REM ERNST GRABNER NEUENHOF
110 DIM I5(12)
150 I0=315360000.: I1=864000.: I2=36000.: I3=600.: I4=10.
160 I5(0)=0.: I5(1)=26784000.: I5(2)=50976000.: I5(3)=77760000.
170 I5(4)=103680000.: I5(5)=130464000.: I5(6)=156384000.
180 I5(7)=183168000.: I5(8)=209952000.: I5(9)=235872000.
190 I5(10)=262656000.: I5(11)=288576000.
200 IF(IJ<0) THEN 280
210 IF(IM<=0 OR IM>12) THEN PRINT "FALSCHER MONAT": STOP
220 IZ=ABS(IJ-1)/4: IZ=INT(IZ)
230 IK=INT(IZ-4*IZ)
240 IF(IK=4 AND DIM>=3) THEN IZ=IZ+1
250 IE=IJ*I0+I5(IM-1)+(IZ+IT-1)*I1+IS*I2+IN*I3+IC*I4
260 ID=INT(IE)+.5
270 RETURN
280 IE=ABS(ID)+.5
290 IJ=INT(IE/I0)
300 IZ=INT(ABS(IJ-1)/4)
310 I6=IE-IJ*I0-IZ*I1
320 IF(I6<0) THEN 350
330 IF(I6=0) THEN 370
340 IF(I6>0) THEN 370
350 IJ=INT(IJ-1)
360 GOTO 300
370 IK=INT(IJ-4*IZ)
380 IF(IK<>4) THEN 400
390 FOR I=2 TO 11: I5(I)=I5(I)+I1: NEXT
400 FOR I=1 TO 11: IF(I6>I5(I)) THEN 420
410 IM=I: I=11: GOTO 440
420 NEXT I
430 IM=12
440 IE=I6-I5(IM-1)
450 IT=INT(IE/I1): IE=IE-IT*I1: IT=INT(IT+1): IS=INT(IE/I2): IE=IE-IS*I2
460 IN=INT(IE/I3): IE=IE-IN*I3: IC=INT(IE/I4)
470 RETURN
READY.

```

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

