

CBM/PET NEWS



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

Lieber Pet - Freund,

Für viele unter uns sind die wohlverdienten Sommerferien bereits zu Ende. Jetzt, wo langsam die Abende wieder länger werden, kommt wieder die Zeit, sich vermehrt mit seinem Computer zu beschäftigen. Die CBM/PET NEWS wollen dazu mit Anregungen, Rat und Tat hilfreich zur Seite stehen.

Ab der nächsten Ausgabe beginnen wir mit einer neuen Serie, die sich vor allen Dingen mit dem Aufbau von kommerziellen Programmen befasst. Solche Programme werden in vielen Fällen von einem Personenkreis benutzt, der wenig oder keine Ahnung im Programmieren hat. Es ist daher eine unbedingte Voraussetzung und für den Programmierer sicher eine reizvolle Herausforderung, darauf zu achten, das bei der Eingabe von Daten und deren Verarbeitung keine Fehler auftreten können, das heisst, die Programme müssen weit stärker gegen Fehlbedienung abgesichert sein, als dies bei Programmen der Fall ist, die nur von sachkundigen Personen bedient werden.

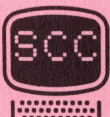
Gleichzeitig benötigen derartige Programme einen weitaus grösseren Aufwand an Hardware, denn ohne Floppy - Station und Drucker sind die Probleme der kommerziellen Datenverarbeitung in der Regel nicht zu bewältigen, da neben dem eigentlichen Programm meist noch grosse Mengen von Daten anfallen.

Für diese Serie haben wir bewusst noch kein endgültiges Konzept aufgestellt, damit Sie, liebe Leser, uns Ihre Wünsche und Ihre Anregungen von Heft zu Heft dazu mitteilen können. Selbstverständlich sind wir interessiert auch von Ihnen, wenn immer möglich, geeignete Beiträge zu diesem interessanten Themenkreis zu erhalten.

Zunächst haben wir uns vorgestellt, in einigen Fortsetzungen einmal eine Reihe von Standard - Routinen zu veröffentlichen, die bei der kommerziellen Datenverarbeitung auf Kleincomputersystemen üblich sind und bei nahezu allen Programmen immer wieder vorkommen. Anschliessend wollen wir dann komplette Programme wie z.B. eine einfachere Adressverwaltung oder Lagerbuchhaltung vorstellen. Also nochmals, es liegt nur die Idee und das grundsätzliche Vorgehen fest, für Ihre Wünsche und Vorstellungen haben wir jederzeit ein offenes Ohr, und wir freuen uns über Ihre hoffentlich zahlreichen Beiträge zu diesem Thema.

Hin und wieder erhalten wir Leserzuschriften, mit dem Hinweis, dieses oder jenes Programm laufe nicht oder es würden Fehler auftreten. Bei der Kontrolle stellen wir dann fest, dass es sich um einfache Typfehler seitens der Benutzer handelt. Gerade beim Eintippen etwas längerer Programme ist doppelte Vorsicht geboten. Sie können aber solche ärgerlichen Fehler elegant umgehen, indem Sie mehr Gebrauch von unserem Kassettenservice machen. Sie wissen ja, jedes "Programm des Monats" ist ausgetestet auch auf Kassette erhältlich.

Heinz Kastien



Tricks und Tips

LISTSCHUTZ

Von einigen Lesern ist schon verschiedentlich nach Programmen gefragt worden, die es ermöglichen, ein Programm vor dem Listen oder Kopieren zu schützen.

Grundsätzlich gibt es genau gleich viele Möglichkeiten, ein Programm vor unberechtigtem Listen oder Kopieren zu schützen als auch diesen Listschutz zu knacken.

Allerdings haben diejenigen Personen, die in der Lage sind, den Listschutz zu entschlüsseln, meist das zu erstellende Programm schneller selbst geschrieben.

Wir wollen Ihnen nun drei Möglichkeiten mit unterschiedlichem Komfort Ihr Programm zu schützen vorstellen, die alle von unseren Lesern gefertigt worden sind.

1. Das 5 ¶ - Verfahren

Geben Sie folgende Zeile ein:

```
10 ¶¶¶¶¶ PRINT"TEST"
```

Geben Sie nun POKE 1029,0 und RETURN ein, ohne diesen POKE-Befehl läuft das Programm nicht.

Starten Sie das Programm mit RUN, so wird der PRINT Befehl in Zeile 10 (der hier für das ganze Programm steht) ausgeführt. Geben Sie dagegen LIST ein, so erscheint von Zeile 10 nur noch die Zeilennummer.

Die Erklärung dieses Phänomens: Mit der Adresse 1024 beginnt der freie RAM-Bereich, diese enthält immer eine Null. Die Adressen 1025 und 1026 enthalten die Anfangsadressen der nächsten Programmzeile. In Adresse 1027 und 1028 steht die erste Zeilennummer und in Adresse 1029 der Inhalt der ersten Zeile (in unserem Fall das erste ¶, oder nach dem POKE eine Null für das erste ¶, danach kommen die vier restlichen ¶ in den Adressen 1030 - 1033).

Am Ende jeder Zeile steht eine Null, der Delimiter, gefolgt vom Hinweis auf die nächste Zeile.

Nach dem Listen findet das Betriebssystem direkt nach der Zeilennummer eine Null und fasst diese als Delimiter auf und springt zur nächsten Zeilennummer.

Diese Art von Listschutz ist denkbar einfach und benötigt pro geschützte Zeile zusätzlich 5 Byte. Es ist daher ratsam, den Listschutz nur auf einige wesentliche Zeilen zu beschränken, die das Know-how des gesamten Programms ausmachen.

Das folgende Programm wandelt automatisch das erste der 5 ¶ in eine Null um und es kann durch viermaliges Drücken der RETURN-Taste nach dem Erscheinen des Cursors gelöscht werden.

```
1 FOR N=1024 TO 8191:IF PEEK(N)⟨255 THEN 4
2 FOR M=1 TO 4:IF PEEK(N+M)-255 THEN M=4:N=N+5:GOTO 4
3 NEXT M:POKE N,0
4 NEXT N:PRINT"¶¶¶¶":FOR I=1 TO 4:PRINT I:NEXT I:PRINT"¶":END
10 REM *** HIER FOLGT IHR PROGRAMM ***
15 REM BEISPIEL
```


Programmbeschreibung

Zeile 1: Das RAM wird auf den ASC-Code des ¶ abgesehen. Ist der Inhalt der Speicherstelle ◊ 255, so wird die nächste Zeile abgesehen, ansonsten erfolgt Sprung zu Zeile 2

Zeile 2: War der Inhalt der Speicherstelle 255, dann wird geprüft ob auch die anderen vier Speicherstellen 255 enthalten. Wenn ja, wird in die erste Speicherstelle eine Null geschrieben (Zeile 3), wenn nein wird die Adresse um 5 erhöht und es erfolgt Sprung zu Zeile 1.

Zeile 4: Bildschirm löschen und 1 - 4 schreiben. ENDE.

Sie können natürlich anstelle der fünf ¶ auch jedes andere Zeichen verwenden, allerdings könnte dies zu Fehlern führen, aber welches Programm enthält schon fünf ¶ am Anfang einer Zeile, ebensogut könnten Sie auch das " @ " Zeichen verwenden, dann muss statt 255 für den ASC-Code die Zahl 64 stehen.

Wenn Sie ein Programm, das mit Listschutz versehen ist, abändern wollen, so müssen Sie den Listschutz rückgängig machen, also wieder die Null durch ein ¶ ersetzen. Das Programm ist also eine Umkehrung des zuerst genannten. Dieses Programm funktioniert unverändert auf allen Commodore Geräten.

```
0 REM FINDE 5 ¶
1 NL=255:A=8191:REM NL=CODE FÜR ¶ X=X+4
2 FOR X=1024 TO A:IF PEEK(X)=255 AND PEEK(X-1)=0THEN POKE X-1,NL:
3 NEXT X:PRINT"OK":FOR I=0 TO 3:PRINT"¶":END
READY.
```

Während es sich bei dem zuletzt genannten Programm um ein reines BASIC Programm handelt, ist das nun folgende ein Maschinensprachprogramm, das im zweiten Kassettenbuffer abgelegt ist und zusammen mit dem Hauptprogramm eingelesen wird. Am Anfang des Hauptprogramms muss dann SYS 800 stehen, mit dem die Listschutzroutine aufgerufen wird. Dieses Programm quittiert jeden LIST-Befehl mit "? LISTING NOT PRESENT". Das Programm ist für den CBM 3000 geschrieben, für den PET 2000 müssen die Werte in Klammern benutzt werden. Für den CBM 8000 ist dieses Programm nicht geeignet.

```
100 REM LISTSCHUTZPROGRAMM BY STEFAN CSOMOR 1981
110 FORA=800TO895:READB:POKEA,B:NEXT
120 DATA165,119,133,17,165,120,133,18,162,23
130 DATA189,249,224,149,112,202,208,248
140 DATA165,17,133,119,165,18,133,120
150 DATA169,76,133,203,169,71,133,204,169,3
160 DATA133,205,96,201,155,208,21,72,138,72
170 DATA186,189,4,1,201,198,208,7,189,3
180 DATA1,201,235,240,10,104,170,104,201,58
190 DATA176,226,76,207,0,162,8,189,119,3
200 DATA32,210,255,202,16,247,160,122,76,133
210 DATA245,71,78,73,84,83,73,76,63,17
READY.
```


SECURITY 8K : MASCHINENPROGRAMM-LISTING

```

.: 7376 $1CD0 A0 61 B9 9C 1D 91 2A 88 10 F8 A4 2B A6 2A D0 01
.: 7392 $1CE0 88 CA 8E F6 1F 8C F3 1F A5 2A A6 2B 18 69 1A 90
.: 7408 $1CF0 01 E8 A0 05 91 2A C8 C8 8A 91 2A A5 2A A6 2B 18
.: 7424 $1D00 69 46 90 01 E8 A0 0D 91 2A C8 C8 8A 91 2A A5 2A
.: 7440 $1D10 A6 2B 18 69 50 90 01 E8 A0 47 91 2A C8 C8 8A 91
.: 7456 $1D20 2A A5 2A 18 69 62 90 02 E6 2B 85 2A A4 2B 8D E7
.: 7472 $1D30 1F 8C E9 1F A9 93 A0 1D 20 1C CA 20 6F C4 E8 BD
.: 7488 $1D40 00 02 D0 FA 86 D1 A9 00 A0 02 85 DA 84 DB A9 01
.: 7504 $1D50 85 D4 20 D0 FD A9 01 A0 04 85 FB 84 FC A9 F2 A0
.: 7520 $1D60 05 85 C9 84 CA A9 01 20 DA F5 A9 0B A0 1E 85 FB
.: 7536 $1D70 84 FC A9 FB A0 1F 85 C9 84 CA 20 89 F8 A5 28 A4
.: 7552 $1D80 29 85 FB 84 FC A5 2A A4 2B 85 C9 84 CA 20 89 F8
.: 7568 $1D90 4C 89 C3 0D 46 49 4C 45 4E 41 4D 45 3D 00 A9 4C
.: 7584 $1DA0 85 70 A9 00 A0 00 85 71 84 72 A9 00 A0 00 85 92
.: 7600 $1DB0 84 93 20 42 C4 4C 72 C5 E6 77 D0 02 E6 78 20 76
.: 7616 $1DC0 00 08 C9 80 90 1C C9 8A F0 18 C9 99 F0 14 C9 9A
.: 7632 $1DD0 F0 10 C9 9C F0 0C C9 B8 F0 08 48 A5 78 C9 02 F0
.: 7648 $1DE0 03 68 28 60 A9 00 A0 00 20 1C CA 4C 89 C3 0D 0D
.: 7664 $1DF0 50 52 4F 54 45 43 54 45 44 20 46 49 4C 45 0D 00
.: 7680 $1E00 FF FF FF FF FF FF FF FF FF FF FF 32 04 0A 00 41
.: 7696 $1E10 24 B2 22 43 43 43 43 43 43 43 43 43 43 43 43
.: 7712 $1E20 43 43 43 43 43 43 43 43 43 43 43 43 43 43 43
.: 7728 $1E30 43 43 43 43 43 43 43 43 43 43 22 00 63 04 14 00
.: 7744 $1E40 42 24 B2 22 43 20 20 20 20 20 20 20 20 20 20
.: 7760 $1E50 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
.: 7776 $1E60 20 20 20 20 20 20 20 20 20 20 43 22 00 9F 04 1E
.: 7792 $1E70 00 99 22 93 11 22 3B 41 24 3A 99 42 24 3A 99 22
.: 7808 $1E80 43 20 44 49 45 53 45 53 20 50 52 4F 47 52 41 4D
.: 7824 $1E90 4D 20 49 53 54 20 47 45 47 45 4E 20 55 4E 42 45
.: 7840 $1EA0 46 55 47 2D 20 20 43 22 00 D2 04 28 00 99 42 24
.: 7856 $1EB0 3A 99 22 43 20 54 45 53 20 4B 4F 50 49 45 52 45
.: 7872 $1EC0 4E 20 55 4E 44 20 4C 49 53 54 45 4E 20 47 45 53
.: 7888 $1ED0 43 48 55 45 54 5A 54 2E 20 43 22 00 05 05 32 00
.: 7904 $1EE0 99 42 24 3A 99 22 43 20 44 45 52 20 5A 55 47 41
.: 7920 $1EF0 4E 47 20 5A 55 52 20 4D 41 53 43 48 49 4E 45 4E
.: 7936 $1F00 45 42 45 4E 45 20 49 53 54 20 00 20 43 22 00 38
.: 7952 $1F10 05 3C 00 99 42 24 3A 99 22 43 20 20 20 20 20
.: 7968 $1F20 20 20 20 20 47 20 45 20 53 20 50 20 45 20 52 20
.: 7984 $1F30 52 20 54 20 20 20 20 20 20 20 20 20 20 20 43
.: 8000 $1F40 22 00 6F 05 46 00 99 42 24 3A 99 42 24 3A 99 22
.: 8016 $1F50 43 20 43 4F 50 59 52 49 47 48 54 20 42 59 20 4B
.: 8032 $1F60 4C 41 55 53 20 47 41 4E 4F 20 20 4D 55 45 4E 43
.: 8048 $1F70 48 45 4E 20 20 20 43 22 00 B1 05 50 00 99 42 24
.: 8064 $1F80 3A 99 41 24 3A 99 22 11 11 57 41 52 54 45 4E 20
.: 8080 $1F90 53 49 45 2C 20 42 49 53 20 44 41 53 20 48 41 55
.: 8096 $1FA0 50 54 50 52 4F 47 52 41 4D 4D 22 3A 99 22 47 45
.: 8112 $1FB0 4C 41 44 45 4E 20 49 53 54 21 00 CE 05 5A 00 9E
.: 8128 $1FC0 31 34 38 38 3A 8F 22 14 14 14 14 14 14 14 14
.: 8144 $1FD0 14 14 14 14 14 14 14 00 00 00 78 EE 10 E8 A5 28
.: 8160 $1FE0 A4 29 85 FB 84 FC A9 00 A0 00 85 2A 84 2B 85 C9
.: 8176 $1FF0 84 CA A9 00 48 A9 00 48 4C 5E F8 AA AA AA AA

```

```

10 REM *****COPY-CHANGE*****
20 REM * VOR PROGRAMMSTART *
25 REM * RAM-ENDADRESSE AENDERN: *
30 REM * 8K=POKE52,206:POKE53,28 *
40 REM * 16K=POKE52,206:POKE53,60 *
50 REM * 32K=POKE52,206:POKE53,124 *
52 REM *
53 REM * ZEILE 110 AENDERN *
55 REM *****
60 PRINT"GEBEN SIE IHREN COPYRIGHT VERMERK EIN":PRINT
70 PRINT"(MAXIMAL 22 ZEICHEN)":PRINT
80 POKE14,1:INPUT"COPYRIGHT BY ":A$
90 POKE14,0:IFLEN(A$)>22THEN80
100 FORI=512TO512+LEN(A$):B=PEEK(I):IFB=0THENB=32:FORX=512+LEN(A$)TO534
110 POKE8031+P,B:P=P+1:REM (16K=16223, 32K=32607)
120 NEXT
READY.

```



Hardware

DIGITAL-ANALOG-WANDLER

In den CBM/PET NEWS haben wir bereits einen D/A-Wandler vorgestellt und seine Wirkungsweise erklärt. Auf theoretischen Grundlagen dieses Wandlerprinzips wollen wir daher in diesem Heft weitgehend verzichten.

D/A-Wandler werden benutzt, um digitale Informationen in ein Analogsignal umzuwandeln.

Die Schaltung in Heft 1/81 ist relativ kompliziert, bietet dafür aber eine hohe Genauigkeit. Sie eignet sich deshalb speziell für Messaufgaben aller Art.

Es soll nun eine zweite Schaltung beschrieben werden, die vor allem bei der Wiedergabe komplexer Töne erforderlich ist. Die Erzeugung einfacher Tonfolgen mit dem CBM-Rechner ist in einem früheren Heft beschrieben worden. Sollen nun Tonfolgen erzeugt werden, bei denen verschiedene Töne unterschiedlicher Höhe und Dauer gleichzeitig auftreten, wie dies bei Musikstücken der Fall ist, so müssen diese Töne, die am USER-PORT in einem 8 Bit Muster auftreten, in ein Analogsignal umgewandelt werden, das dann einem Verstärker zugeführt wird.

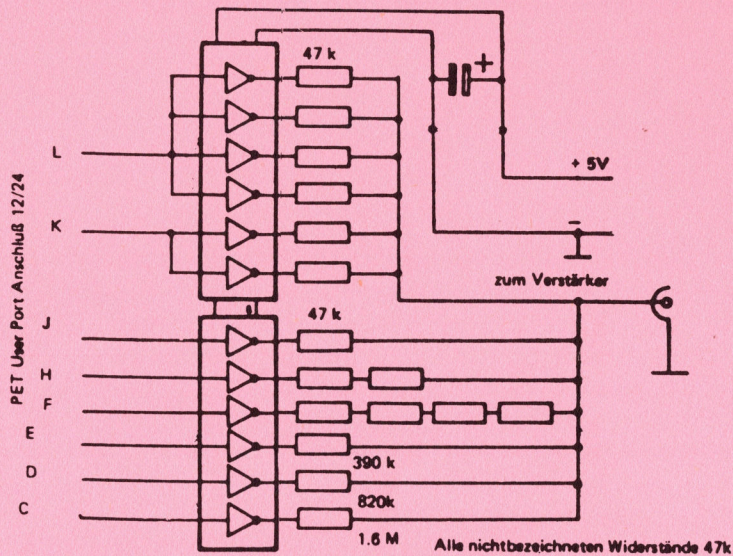
Selbstverständlich geht das auch mit dem D/A-Wandler aus Heft 1/81. Allerdings ist dieser Aufwand für die Erzeugung von Tönen nicht erforderlich. Man kann daher auf eine einfachere Schaltung zurückgreifen. Im Prinzip handelt es sich um einen Summierer, der aus den Widerständen R1 - R15 gebildet wird. Es entfällt aber in dieser Schaltung der Oamp und an seine Stelle treten billige, leicht zu beschaffende Inverter, die lediglich noch die Aufgabe haben, das Signal zu invertieren und mit einem definierten Pegel an den Ausgang zu leiten. Für diese Schaltung genügt auch nur eine einzige 5 V Spannungsversorgung.

Schaltung und Print sind so beschaffen, dass die Platine im dritten Stecker der USER-PORT Erweiterung (Siehe hierzu Heft 2/81 und 3/81 der CBM/PET NEWS) nur noch eingesteckt werden muss. Die Signalführung vom Rechner erfolgt dann über die Anschlüsse PA0-PA7 des USER-PORTS. Der Ausgang des D/A-Wandlers liegt auf dem Anschluss CB 2 und somit automatisch auf dem Eingang des Verstärkers P 2252 bzw. P 2253. Eine Änderung an der USER-PORT Erweiterung ist nicht erforderlich.

Der D/A-Wandler ist unter der Nummer P 9109 zu Fr. 50.-- als Bausatz oder unter der Nummer P 9110 zu Fr. 75.-- als Fertiggerät durch den SCC lieferbar.

Software zu diesem D/A Wandler ist unter der Nummer P 4912 zu Fr. 24.-- ebenfalls beim SCC erhältlich.

Für die Freunde der Selbstprogrammierung werden wir im nächsten Heft aber auch die ausführlichen Programmhilfen zur eigenen Erstellung solcher Musikprogramme bringen.



Stückliste

2 IC CD 4050
1 Elko 10 mF 16 V

USER-PORT ERWEITERUNG

Die Nachfrage nach User-Port-Erweiterung ist sehr gross, vor allem nach fertig montierten Geräten, daher ist es unvermeidlich, dass die Lieferfristen teilweise etwas länger sind, als normalerweise üblich.

Auch sind verschiedentlich Fragen zu diesem Gerät aufgetaucht zu denen wir hier Stellung nehmen wollen.

1. Es ist bemängelt worden, dass mit dem Bausatz keine elektrischen Teile wie Schalter, Stecker und Sicherungen mitgeliefert werden. In den verschiedensten Ländern sind die Vorschriften für Starkstrominstallationen unterschiedlich, und wir sind nur mit den schweizerischen Regeln vertraut.

2. Es wird beanstandet, dass im Gerät nur 4 Stecker für Printplatten enthalten sind. Die Stecker sind sehr teuer und wir liefern die Grundausführung mit nur 4 Steckern, damit das Gerät für alle erschwinglich ist. Diese 4 Stecker sind für die 4 zuerst erscheinenden Erweiterungsplatten Verstärker, D/A-Wandler und Joystick gedacht. Alle weiteren Bausätze enthalten dann einen Printstecker und auch die Schalter für den Device Select.

3. Deutsche Leser haben uns geschrieben, dass der Zoll für die Bausätze sehr hoch ist. Wir empfehlen Ihnen das Gerät ohne Trafo zu bestellen, da elektronisches Material nach Gewicht verzollt wird. Den Trafo können Sie in jedem Electronic Shop beziehen, es muss ein Transformator mit zwei 12 V Wicklungen und einer Belastbarkeit von mindestens 24 VA sein. Sie sparen auf diese Art viel Geld.

4. Verbindungskabel CBM-USER PORT Erweiterung sind ebenfalls durch den SCC erhältlich.

Programm des Monats BASIC-BASIC

NETZTEIL BERECHNUNG

Dieser Beitrag über die Berechnung und Dimensionierung von Netzteilen richtet sich in erster Linie an jene Teil unserer Leser, die auch einmal einen LötKolben in die Hand nehmen, um sich ein Hilfsgerät selbst zu bauen. Da Netzgeräte von nahezu jedem Elektronikbastler in den mannigfaltigsten Ausführungen benötigt werden, sollte dieser Beitrag aber auch einen breiten Leserkreis ansprechen.

Denn dieser Artikel bietet einige wertvolle Hinweise, die auch in vielen anderen Programmen benutzt werden können oder Denkanstöße zur Lösung anderer Programme geben.

Netzgeräte benötigen neben dem Transformator und dem Gleichrichter eine Stabilisierung, wenn von Ihnen eine konstante Spannung erwartet wird. Während man bis vor kurzem Transistoren und Zenerdioden eingesetzt hat, werden heute fast ausschliesslich integrierte Spannungsregler verwendet, die sich einerseits durch hohe Genauigkeit und Stabilität und andererseits durch ein Minimum an externen Bauteilen auszeichnen. Einer der bekanntesten Spannungsregler ist der LM 317, von Texas Instrument hergestellt wird. Er besticht vor allem durch seine Vielseitigkeit.

Kurzbeschreibung des Spannungsreglers LM 317

Spannungsregler mit nur drei Anschlüssen
Thermischer Ueberlastschutz und Kurzschlussstrombegrenzung
Ausgangsspannungsbereich 1.2- 37 V
Maximale Verlustleistung 20 W
Max. Spannungsdifferenz Eingang-Ausgang 40 V
Max. Sperrschichttemperatur 125 K
Minimaler Laststrom 10 mA
Maximaler Laststrom 1500 mA
Brummunterdrückung - 65 dB
Thermischer Widerstand Sperrschicht Gehäuse TO-3 3 K/W
TO-220 5 K/W

Der Spannungsregler ist sehr universell einsetzbar und kann durch externe Transistoren einen maximalen Laststrom bis zu 5 Ampere liefern, wobei dann allerdings 5 Widerstände erforderlich werden.

Das Programm ermöglicht folgende Berechnungen:

1. Berechnung des Netzteils für Ausgangsspannungen von 1.2 - 25 V und Ströme von 10 mA - 5 A.
2. Berechnung des Ladeelkos und Angabe der Trafosekundärspannung
3. Berechnung des Netzteils und der Wickeldaten des Transformators.
4. Ausgabe der Daten und des Schaltschemas auf dem Bildschirm

Um den Einstieg in die Dimensionierung etwas zu vereinfachen sollen die wichtigsten Berechnungen kurz erläutert werden.

Formel 1

Drahtdurchmesser

8 Der Drahtdurchmesser basiert auf einer Stromdichte von 2.55 Ampere/mm²

Formel 2

Spitzenspannung am Ladekondensator

$$CVD = 1.625 * (VA + 3)$$

Formel 3

Trafoprimärleistung

$$P = VA * IL * 1.25 + 1.4$$

Formel 4

Berechnung der min. Filtercapazität

$$C = \frac{0.2 * IL}{(VA * 3 * UBR) * 1000}$$

Programmbeschreibung

- 1040 Dimensionierung der Felder
1050 Setzen der Variablen
1060 - 1530 Bildschirmausdruck der Schaltschema
1540 - 1650 Programmerklärungen
1660 - 1820 Eingabe der Sollwerte
1. Ausgangsspannung VA
2. Ausgangsstrom IL
3. Brummspannung UBR
1740 - 1770 Errechnen der Mindesttrafospannung V1 wenn der Trafo nicht separat gerechnet werden soll.
1830 - 1850 Berechnung des Lastwiderstandes RL
1860 - 1880 Berechnung der min. Filtercapazität C (Formel 1)
1890 - 1900 Berechnung der Spannung am Ladekondensator CVD
1910 - 1920 Spannung am Regler D
1930 - 1940 Leistung am Regler PC
1980 - 2030 Einlesen der E 24 Normreihe der Widerstände
2040 - 2280 Iterative Berechnung der Widerstände in den Schaltungen. (Formel 2)
2290 - 2350 Berechnung der Erwärmung des Reglers TH=C/W bei Gehäuse TO-220 bei einem IL .5A oder 5 W Verlustleistung wird Gehäuse TO-3 gewählt.
2260 - 2410 Berechnung der Erwärmung des Reglers bei Gehäuse TO-3 über 1.5 A oder 12 W Verlustleistung werden Paralleltransistoren mit einbezogen.
2420 - 2460 Berechnung der Anzahl Transistoren. Für 12 W Verlustleistung wird ein Transistor gerechnet. Es wird dann generell der Regler mit Gehäuse TO-220 genommen.
2470 - 2640 Ausgabe der errechneten Daten.
2650 - 2700 Berechnung des Widerstandes R3.
2730 - 2760 Aufruf der Subroutinen zur Darstellung des Schaltschemas.
2770 - 3010 Subroutinen Fehlermeldungen und Warteschleifen.
3020 - 3190 Get - Subroutine zur Eingabe der Zahlenwerte, es sind nur numerische Zeichen und der "." möglich, (siehe Heft 3/81)


```

1000 REM DIMENSIONIERUNG VON NETZTEILEN
1010 REM PROGRAMMIERT H. KRÄSTEN JULI 1981
1020 DIMRE(39),DE(39),RS(2)
1030 P=0:L=1
1040 REM CMB 3032 9.25 KB
1050 GOTO 1550
1060 REM SCHALTSCHHEMA
1070 PRINT "SCHALTSCHHEMA ***"
1080 PRINT "EIN  LM317  AUS  O +"
1090 PRINT "      |      |      |
1100 PRINT "      |      |      |
1110 PRINT "      |      |      |
1120 PRINT "      |      |      |
1130 PRINT "      |      |      |
1140 PRINT "      |      |      |
1150 PRINT "      |      |      |
1160 IF V2=10 THEN 1180
1170 PRINT "      |      |      |
1180 PRINT "      |      |      |
1190 PRINT "      |      |      |
1200 PRINT "      |      |      |
1210 PRINT "      |      |      |
1220 PRINT "      |      |      |
1230 PRINT "      |      |      |
1240 IFS#="J"THENGSUB2980
1250 IFS#="N"THENEND
1260 GOTO3210
1270 PRINT "      |      |      |
1280 PRINT "      |      |      |
1290 PRINT "      |      |      |
1300 PRINT "      |      |      |
1310 PRINT "      |      |      |
1320 PRINT "      |      |      |
1330 PRINT "      |      |      |
1340 PRINT "      |      |      |
1350 PRINT "      |      |      |
1360 PRINT "      |      |      |
1370 PRINT "      |      |      |
1380 PRINT "      |      |      |
1390 PRINT "      |      |      |
1400 PRINT "      |      |      |
1410 PRINT "      |      |      |
1420 PRINT "      |      |      |
1430 IF R2=1000 THEN 1450
1440 PRINT "      |      |      |
1450 PRINT "      |      |      |
1460 PRINT "      |      |      |
1470 PRINT "      |      |      |
1480 PRINT "      |      |      |
1490 PRINT "      |      |      |
1500 PRINT "      |      |      |
1510 IFS#="J"THENGSUB2980
1520 IFS#="N"THENEND
1530 GOTO3210
1540 REM ERKLÄRUNGEN
1550 PRINT "DIMENSIONIERUNG EINES NETZTEILS ***"

```

- 3200 - 3230 Trafoberechnung, Kontrolle der Brummspannung
- 3240 Berechnung der Ausgangsspannung V bei idealer Glättung
- 3250 Uebertragungsleistung des Transformators P
- 3260 - 3330 Ausgabe der Trafoleistung. Es werden nur Typen bis 195 W berücksichtigt, sonst erfolgt Meldung in Zeile 3310.
- 3340 - 3480 Einlesen der 8 Trafodaten aus den DATA - Statements.
 1. Kerntyp K\$
 2. Max. Leistung P2
 3. Verlustfaktor F
 4. Primärwindungszahl für 220 V w1
 5. Drahtdurchmesser für die Primärwicklung D1
 6. Sekundärwindungszahl in Windungen/V W2
 7. Drahtdurchmesser für Sekundärwicklung D2
 Ausgabe der Kerntypen auf dem Bildschirm Zeile 3390
 Vergleich ob Verlustleitung gross genug Zeile 3370
- 3440 Eingabe des gewählten Kerntyps
- 3450 - 3520 Vergleich, ob gewählter Kern möglich, sonst Fehlermeldung in Zeile 3470 und erneute Frage nach dem Kerntyp in 3490 - 3520
- 3530 - 3600 Ausgabe der Trafoprimärspulendaten gemäss DATA - Statements
- 3610 - 3730 Iterative Berechnung der Nennspannung
- 3740 - 3770 Meldung, wenn genaue Berechnung nach der 20. Iteration nicht möglich war.
- 3780 Berechnung des Nennstroms I2
- 3790 Berechnung der Sekundärwindungszahl W
- 3800 Berechnung des Sekundärspulendrahtdurchmessers D
- 3810 Berechnung der Leerlaufspannung U0
- 3820 Berechnung des max. Stroms im Gleichrichter I3
- 3830 - 3950 Ausgabe der errechneten Sekundärdaten
- 3960 - 4000 Fehlermeldungen
- 4010 - 4090 DATA - Statements der Trafotypen




```

1560 PRINT "DIESE PROGRAMM DIMENSIONIERT"
1570 PRINT "DIE EIN REGELBARES NETZTEIL MIT "
1580 PRINT "DIE EINEM SPANNUNGSREGLER LM 317."
1590 PRINT "DIE GROSSTE AUSGANGSSPANNUNG : 25 V"
1600 PRINT "DIE KLEINSTE AUSGANGSSPANNUNG : 1.2 V"
1610 PRINT "DIE GROSSESTER AUSGANGSSTROM : 5.0 A"
1620 PRINT "DIE STROMGRENZUNG DURCH DEN LM 317 "
1630 PRINT "DIE ZUR FORTSETZUNG DES PROGRAMMS "
1640 PRINT "DIE BITTE SPACE - TASTE DRUECKEN !"
1650 GET #0:IF #0=" " THEN 1650
1660 REM EINGABE SOLLWERTE
1670 PRINT "DIE SOLLWERTEINGABE"
1680 PRINT "DIE GEMUNSCHTE AUSGANGSSPANNUNG :";GOSUB3030:VA=VAL(X1#):VB=VA
1690 IF VA>25 THEN 2830
1700 IF VA<1.2 THEN 2830
1710 PRINT "DIE GEMUNSCHTER AUSGANGSSTROM :";GOSUB3030:IL=VAL(X1#)
1720 IF IL>5 THEN 2860
1730 IF IL<.02 THEN 2850
1740 REM MIN. TRAFOSPANNUNG
1750 V3=VA+3
1760 V2=V3*1.3
1770 V1=1+INT(V2/1.4)
1780 PRINT "DIE BRUMMSPANNUNG AM AUSGANG :";GOSUB3030:UBR=VAL(X1#)
1790 PRINT "DIE GEMUNSCHTEN SIE ZUSATZLICH DIE"
1800 PRINT "DIE BERECHNUNG DES TRANSFORMATORS ?"
1810 GET #0:IF #0="N" THEN 1840
1820 IF #0="J" THEN 1810
1830 REM LASTWIDERSTAND
1840 VDC=(V2+V3)/2
1850 RL=VDC/IL
1860 REM FILTERCAPAZITRET
1870 C=.2*IL/(VA*UBR)*1E3
1880 C=INT(1.2*C)
1890 REM SPANNUNG AM KONDENSATOR
1900 CVD=INT(1.625*(VA+3))
1910 REM SPANNUNG AM REGLER
1920 I=VDC-VA
1930 REM LEISTUNG AM REGLER
1940 PC=IL*I
1950 REM WIDERSTRENDE
1960 R1=100
1970 DE(0)=4000
1980 REM DATEN DER WIDERSTRENDE
1990 DATA 0,10,15,22,33,47,56,68,82,91,100,120,150,180,220,270
2000 DATA 330,390,470,560,680,750,820,910
2010 FOR L=1 TO 24
2020 READ RE(L)
2030 NEXT L
2040 REM BERECHNUNG DER WIDERSTRENDE
2050 FOR I=1 TO 14
2060 RE(24+I)=10*RE(10+I)
2070 NEXT I
2080 REM
2090 FOR J=1 TO 2
2100 RX=(VA/1.2-1)*R1
2110 FOR I=1 TO 38
2120 DE(I)=ABS(RX-RE(I))
2130 IF DE(I)>DE(I-1) THEN 2150
2140 RS(I)=RE(I)
2150 NEXT I
2160 R1=120
2170 NEXT J
2180 RA=VA/1.2-1
2190 R210=RS(1)/100

```



```

2840 PRINT "KLEINER ALS 1.2 VOLT !"
2850 GOTO 2800
2860 PRINT "ODER GEMEINSCHTE AUSGANGSSTROM"
2870 PRINT "LIEBERSCHREIBT 5 AMPERE !" : GOTO 2800
2880 GOTO 1710
2890 PRINT "ODER LASTSTROM IST ZU KLEIN UM DIE "
2900 PRINT "KAPAZITRET DES LM 317 AUSZUNUTZEN !"
2910 GOTO 2800
2920 IFS#=1 THEN 3210
2930 IFS#="N" THEN END
2940 PRINT "AUSDRUCK DES SCHALTSCHEMA "
2950 PRINT "DIE 3SPACE-TASTE DRUECKEN !"
2960 GETZ$: IF Z$="" THEN 2960
2970 RETURN
2980 PRINT "ZUR BERECHNUNG DES TRANSFORMATORS"
2990 PRINT "DIE 3SPACE - TASTE DRUECKEN !"
3000 GETZ$: IF Z$="" THEN 3000
3010 RETURN
3020 REM EINGABE SUBROUTINE
3030 X1#="": ANZ=0: PRINT "
3040 GETZ$: IF Z$="" THEN 3040
3050 Z$=ASC(Z$)
3060 IF Z$=20 THEN 3120
3070 IF Z$=13 THEN 3180
3080 IF Z$=40 OR Z$=57 OR Z$=47 THEN 3040
3090 X1#=#X1#Z$ : ANZ=ANZ+1
3100 PRINT Z$ : "
3110 GOTO 3040
3120 IF ANZ=1 THEN X1#="": ANZ=0: GOTO 3160
3130 IF ANZ<1 THEN 3040
3140 ANZ=ANZ-1
3150 X1#=#LEFT$(X1#,ANZ)
3160 PRINT " "
3170 GOTO 3040
3180 PRINT " "
3190 RETURN
3200 REM TRAFOBERECHNUNG
3210 IF UBR<VA/2 THEN 3240
3220 GOSUB 3970
3230 RESTORE: GOTO 1670
3240 V=VA+.666*UBR
3250 P=1.25*IL*V+1.4
3260 REM AUSWAHL DES TRAFOS
3270 PRINT "DIMENSIONIERUNG DES TRAFOS"
3280 PRINT "ZU UEBERTRAGENDE GESAMTLEISTUNG DES"
3290 PRINT "TRANSFORMATORS : "P" W"
3300 IF P<=195 THEN 3330
3310 PRINT "FUER DIESE LEISTUNG IST KEIN KERN"
3320 PRINT "BEKANNT !" : FOR I=1 TO 3000 : GOTO 1550
3330 PRINT "WELCHER KERNE SOLL VERWENDET WERDEN : "
3340 K1=0: RESTORE: FOR I=1 TO 24: READ: NEXT
3350 FOR L=1 TO 8
3360 READ F,P2,F,M1,D1,M2,I2
3370 IF P<P THEN 3410
3380 L1#="STR(L): IFL<9 THEN L1#="L1$
3390 PRINT L1$ " TYP : "F$", MAX. LEISTUNG: "P2" W"
3400 GOTO 3420
3410 K1=L
3420 NEXT L
3430 K1=K1+1
3440 PRINT "WELCHE POSITION WAELHEN SIE : " : GOSUB 3930: P2=VAL(C1#)
3450 IF K2<K1 THEN 3470
3460 IF K2<9 THEN 3490
3470 GOSUB 3990

```



Maschinensprache

ASSEMBLER PROGRAMMIERUNG

PRINTOUT

Nachdem in der letzten Ausgabe mehrheitlich vom Aufbau und der Benützung von Assemblern die Rede war, wird heute ein komplettes Beispiel behandelt. Es ist ein recht nützliches Programm. Vor allem werden dabei Systemroutinen vom CBM/PET kennengelernt und mitverwendet. Dazu dienen die Unterlagen "ROM-Routinen" von H.J.Koch (siehe hierzu Neuheiten). Es ist die Aufgabe gestellt, beliebige Speicherbereiche des CBM auszudrucken. Der Printer wird am IEC-Bus angeschlossen. Das Ausgabeblatt soll links einen genügend grossen Rand aufweisen (Lochrand zum Einheften). Pro Zeile werden 32 Bytes ausgedruckt, wobei je 8 Byte zu einem Datenblock zusammengefasst und durch ein Blank getrennt sind. Auf jeder Zeile soll zu Beginn noch die Memoryadresse vom ersten Byte angegeben werden. Alle Bytes und Adressen werden in hexadezimaler Form dargestellt (siehe das beigefügte Listing).

Damit ist das Problem formuliert. Eine mögliche Lösung enthält etwa die folgende Funktionsabschnitte:

- Den gewünschten Memorybereich vom Gerätebediener verlangen.
- Die eingegebenen Memoryadressen auf Plausibilität prüfen, evtl. Fehlermeldungen ausgeben.
- Den Outputkanal zum Printer öffnen, sofern dies noch nicht erfolgt ist.
- Den Memorybereich ausdrucken.
- Die Frage ausgeben, ob weitere Bereiche ausgedruckt werden sollen. Wenn ja, wieder am Anfang weitermachen. Wenn nein, den Printerkanal schliessen, Programmabbruch und zurück ins BASIC.

Nun wollen wir die eingegebenen Funktionen etwas genauer ansehen. Mit der Systemroutine 'WRT' können einzelne Zeichen vom Accu in den Bildschirm geschrieben werden. Dabei werden die Werte in den Registern A,X und Y nicht zerstört und der Cursor wird automatisch mitgeführt. Bringt man in einem Loop nacheinander die auszugebenden Zeichen in den Accu und ruft die WRT-Routine auf, so können auf diese Art Meldungen geschrieben werden. Hier ein Beispiel, stellvertretend für alle Textausgaben im Programm PRINTOUT (s. Listing).

```
LDX #0 ; Den Zeichenzähler 0 setzen
Loop 1 LDA MSG1,X; Ein Zeichen vom Text lesen
      ISR WRT ; und zum Bildschirm geben
      INX ; Den Zeichenzähler erhöhen
      CPX #17 ; ganzer Text ausgeben?
      BNE LOOP 1; Springe wenn nein
MSG1 .ASCII 'STARTADR (HEX):' ; 17 Zeichen
```

Jetzt muss auch noch der Memorybereich eingelesen werden, respektive die Start - und Endadresse. Die geschieht mit der Systemroutine 'RDOA'. Diese Routine verarbeitet eine Hex-Adresse (4 Zeichen) vom Bildschirm und legt nach der RETURN-Eingabe das Resultat in der Zeropage auf den Adressen 251/252 ab. Die genaue Benützung wird gemacht durch:


```

ISR RDOA      ; Hex-Adresse einlesen
LDA INADR     ; Low-Byte der Eingabe in
STA SADR     ; der Startadresse ablegen
LDA INADR+1  ; High-Byte der Eingabe in
STA SADR+1   ; der Startadresse ablegen

```

Das gleiche gilt auch für die Eingabe der Endadresse.

Als nächstes wird geprüft, ob die Endadresse grösser ist, als die Startadresse. Wenn nicht, wird die Meldung 'ADDRESS ERROR!' ausgegeben.

Die folgende Funktion ist die Oeffnung des Printerkanals. In BASIC geschieht die allgemeine Oeffnung eines Datenkanals mit: 'OPEN LF, DN, SA, FNAM'. Für den Printer benützt man normalerweise OPEN 4,4. In Assembler muss man schreiben:

```

LDA # 4
STA LF      ; Logische Filenummer = 4
STA DN      ; Device Nummer = 4
LDA # 1     ; Sekundäradresse = 1 für Schreiben
STA SA      ;
LDA FNAMP
STA FNAR    ; Die Adresse vom Filenamen
LDA FNAMP+1 ; 'PRINTOUT' in FNADR
STA FNAMP+1 ; ablegen
LDA # 14
STA FNLEN   ; Die Länge des Filenamen
ISR OPEN

```

Diese Parameter (LF, DN, SA, FNADR, FNLEN) müssen angegeben werden, wenn man direkt einen Kanal öffnen will, ohne über die Filenamentabelle zu gehen. Dies ist noch komplizierter und wird bei mehreren gleichzeitig geöffneten Kanälen benötigt.

Nach dem Oeffnen des Datenkanals zum angeschlossenen Gerät, muss diesen noch mitgeteilt werden, dass es als "Listener" zu arbeiten hat. Darauf muss noch die Sekundäradresse über den Bus gesendet werden. Dies wird gemacht mit:

```

ISR LISTN   ; aktiviert Device als Listener
LDA SA      ; Sekundäradresse
ISR SECND   ; zum Bus senden

```

Ab jetzt können mit der Systemroutine 'CIOUT' ASCII-Daten ausgedruckt werden, genauso wie mit 'WRT' zum Bildschirm geschrieben wird..

Will man den Kanal wieder schliessen, so schreibt man:

```

ISR UNLSN   ; Device inaktiv machen
ISR CLOSE   ; LF muss noch 4 sein
              wie beim Oeffnen

```

Noch ein Wort zur Memoryausgabe. Ausgedruckt wird immer ab der laufenden Adresse, gespeichert in der Zeropage in SADR. Die Ausgabe erfolgt immer in der Einheit 8 Bytes mit der Routine 'BTOUT'. Ein Memorybyte muss zuerst in 2 ASCII-Zeichen umgeformt werden. Dies geschieht mit der Routine 'CONVT'. Das Zeichen wird mit 'CIOUT' ausgegeben.

Zu Beginn jeder Ausgabenzeile wird zur Startadresse SADR 32 addiert, in SADRP gespeichert und geprüft, ob die Endadresse EADR grösser als SADR ist. Wenn ja, kann ein ganzer Zeilenblock ausgegeben werden. Dies erfolgt mit 'OUT 1'. Wenn nein, wird bei 'DONE 1' weitergemacht. Hier wird zu

SADR 8 addiert und geprüft, ob die Endadresse grösser als SADR_P ist. Wenn ja kann in 'OUT2' ein 8 Byte Block ausgegeben werden. Wenn nein wird bei 'DONE2' weitergemacht. Hier ist das Ausdrucken zu Ende. Es wird dann gefragt, ob noch weitere Ausdrücke gewünscht werden. Die Keyboardeingabe wird mit der Systemroutine 'GET' gemacht. Diese liefert, sobald eine Taste gedrückt wird, im Accu den ASCII-Code der Taste. Dieser kann sofort ausgewertet werden. Solange keine Taste gedrückt ist, bleibt das Zero-Flag gesetzt. Damit kann eine Warteschleife gebildet werden. Alles Weitere kann aus dem Listing entnommen werden.

Wie wird nun das Programm erstellt ?

Nehmen Sie den Editor vom CBM-Rechner und geben Sie die Sourcezeilen ein, beginnend mit '*=31000'. Geben Sie den ganzen mnemonischen Code ein, inklusive der Kommentare bis zur Zeile 235 mit '.END'. Speichern Sie das ganze auf Diskette.

Nehmen Sie nun den Assembler und übersetzen den Sourcefile. Wenn alles fehlerfrei ist, so sollten Sie die Meldung '57 Symbols 0 Errors 501 Bytes' erhalten. Auf der Diskette wird auch ein Listing und ein Hex File erzeugt.

Möchten Sie den Maschinencode, welcher vom Assembler erzeugt wird, direkt in den Speicher 'poken' lassen, so ist vor dem Programm RUN der Speicherbereich mit POKE 52,0 POKE 53,121 abzublocken. Dies gilt insbesondere für das Arbeiten mit dem Programm EDASS 65.

Wie wird das Programm installiert und benützt ?

Das Maschinenprogramm 'PRINTOUT' liegt im Speicher absolut auf Adresse 31000, mit der Grösse von 501 Bytes. Damit dieser Speicherbereich durch ein BASIC-Programm nicht überschrieben wird, muss dem BASIC-Interpreter etwas ins Handwerk gepfuscht werden. Wir verkürzen einfach den freien Speicherbereich indem wir den BASIC-Pointer 'Top of Memory' heruntersetzen. Normalerweise zeigt er auf 128*256 = 32768. Wir setzen ihn auf 121 * 256 = 30976.

Beim CBM 3032 BASIC Version 3 und 4 wird dies gemacht mit

POKE 52,0 POKE 53,121

Beim CBM 2032 BASIC Version 1 mit

POKE 134,0 POKE 135,121

Jetzt können Sie mit dem Hexloader das Maschinenprogramm ab Diskette einlesen und in den reservierten Speicher einpoken lassen.

Das Maschinenprogramm kann jetzt mit SYS (31000) aufgerufen werden.

Als Beispiel hat sich das Programm 'PRINTOUT' selbst ausgedruckt.

PRINTOUT:

```

7918 A99320D2FFA90085 4D20937A20937AA2 00BDB17A20D2FFE8 E011D0F520A7E7A5
7938 FB8546A5FC854720 937AA200BDC27A20 D2FFE8E011D0F520 A7E7A5FB8546A5FC
7958 854938A546E548A5 47E549901320937A A200BDEF7A20D2FF E8E00ED0F54C00C7A
7978 A54DD02BA90485D2 85D4A90185D3854D AD087B85DAAD0C7B 85DBA90E85D12024
7998 F520BAF0A5D32028 F120897A208E7A18 A5468320854BA547 6900854C18A54BE5
79B8 48A54CE549101D20 787A20567AA90485 4E20847A20427A20 A37AC64ED0F32089
79D8 7A4CA77920787A20 567A20847A18A546 6908854BA5476900 854C18A54BE548A5
79F8 4CE549100920427A 29A37A4CE2792042 7A20897A20937AA2 00BDD37A20D2FFE8
7A18 E01CD0F520E4FFF0 FBC94EF00AC959D0 F320D2FF4C217920 D2FF20897A208E7A
7A38 2083F120ACF22093 7A60A900854FA44F B146205D7AE64FA5 4FC908D0F160A547
7A58 205D7AA5464829F0 186A6A6A6A206B7A 68290F186930C93A 90026906206FF160
7A78 A205864F20847AC6 4FD0F960A9204C74 7AA90D20747AA90A 4C747A20D0FD20D0
7A98 FD20CAFDD20CAFDD D2FF6018A5466908 8546A54769008547 6053544152544144
7AB8 5220284845582920 3A20454E44414452 2020202848455829 203A20434F4E5449
7AD8 4E55455320505249 4E544F5554202028 592F4E29203F2041 4444524553532045
7AF8 52524F5221202020 20205052494E544F 55543AFD7A

```





79DC	118	:	DOHE1	JSR	BLNKS	5 BLANKS AUSGEBEN	7A5B	A546	179	LDA	SADR	AUSZUGERENDES ZEICHEN RETTEN		
79DF	119	:	JSR	ADOUT	:MEMORYADRESSE AUSGEBEN	7A5D	48	179	CONVT	PHA	#%11110000	:LOWER DIGIT WEGSCHNEIDEN		
79E2	120	:	OUT2	JSR	BKOUT	:1 BLANK AUSGEBEN	7A5E	29F0	180	CLC				
79E5	121	:	CLC	LDA	SADR		7A60	18	181	ROR	A			
79E6	122	:	ADC	#8		7A61	6A	182	182	ROR	A			
79E8	124	:	STA	SADRP		7A62	6A	183	183	ROR	A	:DAS HIGHER DIGIT ZUM		
79EA	125	:	LDA	SADR+1		7A63	6A	184	184	ROR	A	: LOWER DIGIT VERSCHIEBEN		
79EE	127	:	ADC	#2		7A64	6A	185	185	ROR	A	: UND AUSGEBEN		
79F0	128	:	STA	SADRP+1	:SADRP=SADR+8	7A65	206E7A	186	186	PLA		: ZEICHEN ZURUECK HOLEN		
79F3	129	:	LDA	SADRP		7A68	68	187	187	AND	#2000031111	:HIGHER DIGIT WEGSCHNEIDEN		
79F5	130	:	SBC	EDDR		7A6B	18	188	188	CLC				
79F7	131	:	LDA	SADRP+1		7A6C	6930	190	190	ADC	#0	:EIN ASCII-ZEICHEN MACHEN		
79F9	132	:	SBC	EDDR+1		7A6E	C93A	191	191	CMP	#9+1			
79FB	134	:	RPL	DOHE2	:IST EADR <= SADR ?	7A70	9002	192	192	BCC	CONX			
79FD	134	:	JSR	BTOUT	:SPRINGE WENN JA	7A72	6906	193	193	ADC	#6	:ASCII = A...F		
7A00	136	:	JSR	ADADR	:1 BLOCK AUSGEBEN (8 BYTES)	7A74	206FF1	194	194	CONX	CIOUT	:ZEICHEN ZUM PRINTER GEBEN		
7A03	137	:	JMP	OUT2	:SADR=SADR+8 :WEITERMACHEN?	7A77	60	195	195	RTS				
7A06	139	:	DOHE2	JSR	BTOUT	7A78	A205	196	196	LDX	#5	:ALS LINKER RAND 5 BLANKS		
7A09	140	:	MCOUNT	JSR	PCRLF	7A7A	864F	198	198	STX	CNT8	: ZUM DRUCKER SENDEN		
7A0F	142	:	L9	JSR	CRORS	7A7C	20847A	199	199	JSR	BKOUT			
7A11	143	:	LDX	#0	:DIE LETZTEN 8 BYTES AUSGEBEN	7A7F	C64F	200	200	DEC	CNT8			
7A14	144	:	JSR	WRT	:ZEILENVORSCHUB BEIM PRINTER	7A81	DAF9	201	201	BNE	BLKS			
7A17	145	:	INX	#28	:TEXT AUSGEBEN	7A83	60	202	202	RTS				
7A18	146	:	BNE	L9	:FUER WEITERMACHEN?	7A84	A920	204	204	BKOUT	LDA	#32	:EIN BLANK AUSGEBEN	
7A1C	147	:	INLOP	JSR	GET	7A86	4C747A	205	205	JMP	CONX			
7A1F	148	:	BEO	INLOP	:ANTWORT EINLESEN	7A89	A900	206	206	PCRLF	LDA	#13	:RETURN AUSGEBEN	
7A23	149	:	CMP	#7H	:SPRINGE WENN KEINE TASTE	7A8B	20747A	208	208	JSR	CONX			
7A25	150	:	BEO	PGEND	:SPRINGE WENN N GEDRUECKT	7A8E	A90A	209	209	PLF	LDA	#10	:LINE FEED AUSGEBEN	
7A27	151	:	CMP	#4		7A90	4C747A	210	210					
7A29	152	:	BNE	INLOP	:SPRINGE WENN NICHT Y GEDRUECKT	7A93	2008FD	211	211	CRORS	JSR	CRLF		
7A2C	154	:	JSR	WRT	:Y ZUM BILDSCHIRM	7A96	2008FD	213	213	JSR	CRLF		:2 RETURN ZUM BILDSCHIRM	
7A2E	155	:	JMP	CONT	:WEITERMACHEN!	7A99	20C9FD	214	214	JSR	SPAC2		:5 BLANKS ZUM BILDSCHIRM	
7A3F	156	:	PGEND	JSR	WRT	7A9C	20D2FF	216	216	JSR	WRT			
7A3E	157	:	JSR	PCRLF	:N ZUM BILDSCHIRM	7AA2	60	217	217	RTS				
7A3E	158	:	JSR	PLF	:ZEILENVORSCHUB BEIM PRINTER	7AA3	18	218	218	ADADR	CLC		:PRINTOUTADRESSE ERHOEHEN:	
7A3E	159	:	JSR	UNLGH	:PRINTER INAKTIV MACHEN	7AA3	A546	219	219	LDA	SADR		:SADR=SADR+8	
7A3E	160	:	JSR	CLOSE	:DATENKANAL SCHLIESSEN	7AA4	6908	220	220	ADC	#8			
7A3E	161	:	JSR	CRORS	:ZURUECK INS BASIC	7AA6	8546	221	221	STA	SADR			
7A41	162	:	RTS			7AA8	A547	223	223	LDA	SADR+1			
7A42	163	:	BTOUT	LDA	#0	7AAA	6900	224	224	ADC	#9			
7A42	164	:	LDY	CNT8	:8 MEMORYBYTES	7AAE	8547	225	225	STA	SADR+1			
7A45	166	:	BYT8	LDA	(SADR),Y	7AB0	60	226	226	RTS				
7A48	168	:	JSR	CONVT	: ZUM DRUCKER AUSGEBEN	7AB1	227	227	227	MSG1	.ASCII	'STARTADR	(HEX)	///
7A4D	170	:	INC	CNT8		7AB2	454E44	228	228	MSG2	.ASCII	'ENDADR	(HEX)	///
7A4F	171	:	LDA	CNT8		7AB3	434FE4E	229	229	MSG3	.ASCII	'CONTINUES	PRINTOUT	(Y/N) ?
7A4F	172	:	CMP	#8		7ABF	414444	231	231	ERMSG	.ASCII	'ADDRESS	ERROR!	///
7A51	173	:	BNE	BYT8		7AFD	202020	232	232	FRAME	.ASCII	'PRINTOUT	///	
7A53	173	:	RTS			7BAB	FD7A	233	233	FNAMP	.WORD	FRAME		
7A55	174	:	LDX	SADR+1	:DIE MEMORYADRESSE	7B00		234	234					
7A56	175	:	JSR	CONVT	: ZUM DRUCKER AUSGEBEN			235	235					
7A58	177	:						236	236					

ADADR	7A83	ADOUT	7A56	BKOUT	7A84	BLK5	7A7C
BLNK5	7A78	BTOUT	7A42	BYTX	7A46	CIOUT	F16F
CLOSE	F2AC	CHSG	7AD3	CHOUT	7A6B	CNT4	004E
CNT8	004F	CONT	7921	COHMT	7A5D	CONK	7A74
CRCRS	7A93	CRLF	FDD0	DH	00D4	DONE	7978
DONE1	79DC	DONE2	7A06	DONE1	79A7	EADR	0048
ERLOP	796A	ERMSG	7AEF	FNADR	00DA	FNAME	7AFD
FNAMP	7B08	FNLEN	00D1	GET	FFE4	INADR	00FB
INLOP	7A1C	L1	7929	L2	7944	L9	7A11
LF	00D2	LISTN	F8BA	NCONT	7A0C	MSG1	7AB1
MSG2	7AC2	OPEN	F524	OUT1	79C9	OUT2	79E2
PCHAN	004D	PCRLF	7A89	PGEND	7A2F	PLF	7A8E
RDR	E7A7	SA	00D3	SADR	0046	SADRF	004B
SECHD	F128	SFAC2	FDCA	START	7918	UNL5H	F183
URT	FFD2						

Für diejenige, die noch nicht so vertraut mit der Assemblerprogrammierung sind, aber dennoch das PRINTOUT - Programm benutzen möchten, haben wir das Programm in BASIC mit DATA - Statements umgeschrieben. Das Programm läuft nur auf CBM 3000 er Geräten.

```

100 REM -----
110 REM #   PRINTOUT      22. JULI 1981   E.BREU   #
120 REM -----
130 :
200 POKE 52,0: POKE 53,121: REM CBM BASIC V3.0
220 :
300 FOR M=31000 TO 31500: READ A: POKE M,A: NEXT M
310 PRINT:PRINT"DATA'S GELADEN."
320 :
1000 DATA 169,147,32,210,255,169,0,133,77,32,147,122,32,147,122,162
1010 DATA 0,189,177,122,32,210,255,232,224,17,208,245,32,167,231,165
1020 DATA 251,133,70,165,252,133,71,32,147,122,162,0,189,194,122,32
1030 DATA 210,255,232,224,17,208,245,32,167,231,165,251,133,72,165,252
1040 DATA 133,73,56,165,70,229,72,165,71,229,73,144,19,32,147,122
1050 DATA 162,0,189,239,122,32,210,255,232,224,14,208,245,76,12,122
1060 DATA 165,77,208,43,169,4,133,210,133,212,169,1,133,211,133,77
1070 DATA 173,11,123,133,218,173,12,123,133,219,169,14,133,209,32,36
1080 DATA 245,32,186,240,165,211,32,40,241,32,137,122,32,142,122,24
1090 DATA 165,70,105,32,133,75,165,71,105,0,133,76,24,165,75,229
1100 DATA 72,165,76,229,73,16,29,32,120,122,32,86,122,169,4,133
1110 DATA 78,32,132,122,32,66,122,32,163,122,198,79,208,243,32,137
1120 DATA 122,76,167,121,32,120,122,32,86,122,32,132,122,24,165,70
1130 DATA 105,8,133,75,165,71,105,0,133,76,24,165,75,229,72,165
1140 DATA 76,229,73,16,9,32,66,122,32,163,122,76,226,121,32,66
1150 DATA 122,32,137,122,32,147,122,162,0,189,211,122,32,210,255,232
1160 DATA 224,28,208,245,32,228,255,240,251,201,78,240,10,201,89,208
1170 DATA 243,32,210,255,76,33,121,32,210,255,32,137,122,32,142,122
1180 DATA 32,131,241,32,172,242,32,147,122,96,169,0,133,79,164,79
1190 DATA 177,70,32,93,122,230,79,165,79,201,8,208,241,96,165,71
1200 DATA 32,93,122,165,70,72,41,240,24,105,106,106,106,32,107,122
1210 DATA 104,41,15,24,105,48,201,58,144,2,105,6,32,111,241,96
1220 DATA 162,5,134,79,32,132,122,198,79,208,249,96,169,32,76,116
1230 DATA 122,169,13,32,116,122,169,10,76,116,122,32,208,253,32,208
1240 DATA 253,32,202,253,32,202,253,32,210,255,96,24,165,70,105,8
1250 DATA 133,70,165,71,105,0,133,71,96,83,84,65,82,84,65,68
1260 DATA 82,32,40,72,69,88,41,32,58,32,69,78,68,65,68,82
1270 DATA 32,32,32,40,72,69,88,41,32,58,32,67,79,78,84,73
1280 DATA 78,85,69,83,32,80,82,73,78,84,79,85,84,32,32,40
1290 DATA 89,47,78,41,32,63,32,65,68,68,82,69,83,83,32,69
1300 DATA 82,82,79,82,33,32,32,32,32,80,82,73,78,84,79
1310 DATA 85,84,58,253,122
1320 END
READY.

```



Neuheiten

ROM-ROUTINEN CODE-LISTEN SPEICHERBELEGUNG

Während wir in den letzten Heften immer wieder Neuheiten vorgestellt haben, die vor allem aus dem Bereich Programmierhilfen und Toolkit gekommen sind und somit einen grossen Anwenderkreis ansprechen, wollen wir heute vor allem den Liebhabern der Maschinen- und Assemblerprogrammierung eine wirkliche Neuheit vorstellen, die ausserdem noch für jeden erschwinglich ist.

Es handelt sich um komplette Listen der Speicherbelegung beim Commodore Rechner 2001,3001 und 8001, sowie um die ausführlich Beschreibung zum Aufruf wichtiger Routinen aus dem Festwertspeicher der oben genannten Rechner.

Diese Listen sind vor allem daher so hervorragend, da sie endlich eine vollständige Sammlung der Adressen und Routinen bringen, die in dieser Form bisher in keinem Handbuch oder andersartigen Werk erschienen sind.

Liste der Speicherbelegung

Bei der Erstellung von Programmen in Maschinensprache ist die Kenntnis gewisser Speicherplätze von grösster Wichtigkeit. Der Assembler Programmierer kann sich damit, die in den ROM's vorhandenen Routinen zunutze machen und auf diese Weise eine Menge Zeit und Aufwand sparen. Es werden die Speicherplätze für die Seiten 0 - 3 (Adresse 0 - 1023) beschrieben, die den Schreib-/Lesespeicher für das Betriebssystem und BASIC darstellen. Es folgt die Beschreibung der Adressen der ROM - Bereiche mit Interpreter, Betriebssystem und Monitor sowie dem I/O - Bereich. Alle Adressen sind, sofern vorhanden, für die drei heute bekannten Rechnertypen in hexadezimaler Darstellung angegeben und genau beschrieben.

Die Beschreibung der Speicherbelegung hat einen Umfang von 18 A 4 Seiten mit über 630 Adressangaben.

ROM - Routinen

Die Computer der PET und CBM - Reihe enthält in den ROM's eine Reihe von Programmen, die für den Assembler Programmierer von unschätzbarem Wert sind. Mit diesen Routinen ist dem Assemblerprogrammierer z. B. auch der Cassettenbetrieb und die reelle Arithmetik auf einfache Art möglich. Bei einigen der beschriebenen Routinen müssen vor dem Aufruf Speicherplätze und Register gesetzt werden.

Diese 43 Seiten umfassende Beschreibung der ROM - Routinen soll nicht als Einstieg in die Assembler Programmierung verstanden sein. Hier wollen wir auf die Fortsetzungsreihe "Assemblerprogrammierung" in diesem Heft verweisen, sie sollen nur eine Hilfe darstellen. Wir werden daher in diesem Heft unter der Rubrik "Assemblerprogrammierung" auf die beschriebenen Tabellen Bezug nehmen.

Code Listen

Schliesslich ist eine dritte Liste lieferbar, die alle Code der Commodore Computer enthält. Neben den Tasten und Bildschirmcode sind auch alle Charaktercode aufgeführt. Diese Listen sind nicht nur für Freunde

der Maschinenprogrammierung interessant, sondern auch für jeden BASIC - Programmierer, der mit POKE und CHR\$-Befehlen arbeitet.

Die Tabellen sind durch den Ersteller zu folgenden Preisen lieferbar:

Speicherbelegung	DM 20.00
ROM - Routinen	DM 25.00
Code - Listen	DM 12.00
Speicherbelegung und ROM - Routinen	DM 35.00
Speicherbelegung, ROM-Routinen und Codelisten	DM 44.00

Die Listen werden gegen Verrechnungsscheck oder Vorauszahlung auf Postscheckkonto 269365-304 Hannover durch

Hans Joachim Koch
 Liegnitzer Str. 8
 3008 Garbsen 8

 Tel. 05131 53510

geliefert

Aus diesen Tabellen drucken wir zur besseren Uebersicht 3 Probeseiten in verkleinerter Darstellung ab. Um das Copyright zu wahren, sind die Tabellen unvollständig. Sicher haben Sie für diese Massnahme Verständnis, wenn Sie sich einmal überlegen, wieviel Arbeit hinter der Erstellung dieser Listen steckt.

 Rückkehr vom Interrupt
 Uhr weiterschalten, Tastatur abfragen (1/60 Sec)
 Piepton erzeugen
 Kodiertabelle für Tastatur
 Ausgabe Hex-Adresse aus \$FB,\$FC zum Schirm
 Ausgabe Hex-Byte aus A-Register zum Schirm
 Ausgabe ASCII aus X- und A-Register zum Schirm
 Umwandlung Hex-Byte in ASCII (A-Register)
 Vertauschung der Adressen in \$FB,\$FC mit \$FD,\$FE
 Eingabe Hex-Adresse vom Schirm in \$FB,\$FC
 Unterprogramm zur Ausgabe auf dem Schirm
 Eingabe Hex-Byte vom Schirm in A-Register
 gibt feste Monitor-Meldungen aus Bereich \$F000 aus
 Umwandlung ASCII in Hex-Byte (A-Register)
 Eingabe Zeichen vom Schirm in A-Register
 Ausgabe eines Fragezeichens als Fehlermeldung
 I/O Ports & Expansion I/O
 6528 PIA, Adressen \$E810-\$E813
 Position 68 (2001), C7 (3001)
 PIAL, I/O Port A
 Bit 0-3 (Output): Tastaturansteuerung
 Bit 4 (Input): Schalter Recorder #1
 Bit 5 (Input): Schalter Recorder #2
 Bit 6 (Input): EOI Signal vom IEEE-Bus
 EOI = End Or Identify
 Bit 7 (Input): Abfrage auf Testpin (User Port)
 WAIT 59488,16,16 wartet, bis Tasten an Recorder #1
 gedrückt werden. WAIT 59488,16 wartet, bis alle
 Tasten an Recorder #1 ausgerastet werden
 (32 für Recorder #2).
 PIAL, Kontroll-Register A
 Bit 3 (Output): Schirm dunkel steuern (nur 2001)
 POKE 59489,52 : Schirm dunkel, 68 : hell
 PIAK, I/O Port B
 Bit 0-7 (Input): Eingabe von Tastaturabfrage
 enthält normalerweise 255. RVS:254, (:253,
 SPACE:251, (:247. WAIT 59410,4,4 wartet, bis die
 Leertaste gedrückt wird.
 PIAS, Kontroll-Register B
 Bit 3 (Output): Motorkontrolle Recorder #1
 POKE 59411,53: Motor ein, 61 : aus
 6528 PIA, Adressen \$E820-\$E823
 Position 68 (2001), C6 (3001)
 IEEL, I/O Port A
 Bit 0-7 (Input): DI1-DIB Input vom IEEE-Bus
 Einlesen vom IEEE-Bus: PEEK(59424)
 IEEEIS, Kontroll-Register A
 Bit 3 (Output): NDAC Signal zum IEEE-Bus (CA2)
 NDAC = Not Data Accepted
 NDAC low: POKE 59425,52



Leserbriefe

MUEHLE ERGAENZUNG

Nun hat sich auch bei uns der Tippfehlerteufel eingeschlichen. Im Heft 3/81 haben wir eine Ergänzung des Mühlespiels abgedruckt und Herr Kast als aufmerksamer Leser und Ersteller des Programms hat uns auf zwei Tippfehler hingewiesen.

Wir möchten uns dafür entschuldigen und drucken der besseren Uebersicht halber, die gesamte Programmerweiterung noch einmal ab.

```
3 DIMP%(52):GOTO10:REM VERSION #3, 1. TEIL
600 M=0:ONUPGOTO630,650,670,690,730,810,850,910
830 N=1:IZ=RND(-TI)
840 UP=7:IZ=0
850 IZ=IZ+1:I=INT(8*RND(1))+2:J=INT(3*RND(1))*20:P=I+J
860 IFP%(P)=0THENM=8:RETURN
870 IFIZ<20THEN850
900 UP=8:I=1:J=0
910 I=I+1:IFI>9THENI=2:J=J+20:IFJ>40THEN940
920 P=I+J:IFP%(P)=0THENM=8:RETURN
930 GOTO910
940 IFN<3THENM=N+1:GOTO840
950 RETURN
3750 B=P%(I+F1)+P%(I+F1+20)+P%(I+F1+40)
3755 IFN=1ANDB=27ORN=1ANDB=11THENRETURN
3760 IFN=2ANDB=11THENRETURN
3860 B=P%(A-1)+P%(A+1)
3870 IFN=1ANDB=18ORN=1ANDB=2THENRETURN
3880 IFN=2ANDB=2THENRETURN
3890 PN=P:PA=A:GOTO2600
3900 B=P%(P+2*F1)+P%(P+3*F1)
3910 IFN=1ANDB=18ORN=1ANDB=2THEN3810
3920 IFN=2ANDB=2THEN3810
3930 GOTO3770
READY.
```

ZEILEN 880 UND 890 LOESCHEN

Sicher haben sich einige unserer Leser schon darüber geärgert, das Sie auf Ihren Brief oder Ihre Anfrage noch keine Antwort erhalten haben. Wir möchten uns hierfür entschuldigen, aber gleichzeitig bemerken, das unsere Redaktion mit einem Minimum an Personal besetzt ist und daher einige Zeit vergehen kann, bis wir die Zeit finden, um Ihr spezielles Problem zu behandeln.

Wir haben diverse Zuschriften von Lesern erhalten, die Probleme mit Druckroutinen haben. Welcher unserer Leser hätte die notwendige Zeit, um diese Probleme einmal näher anzusehen?

Herbstaktion

CBM 4032

Herbstaktion

zum aktuellen Hitpreis von

Fr. 2750.-



- 4000-Serie
- s/w-Bildschirm
- BASIC/Maschinensprache
- hohe Auflösung
- 25 Zeilen à 40 Zeichen
- und, und, und...

* * * * *

PET 2001

(Demomodelle)

Zum Sensationspreis von

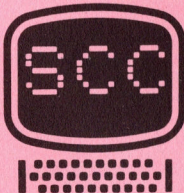
Fr. 1500.-

Nur solange Vorrat!

Alle Geräte mit **SCC-Vollgarantie**

Bestellungen bitte direkt an SCC-Shop, Seeburgstrasse 18,
6002 Luzern, Telefon 041- 31 45 45

23



Korrespondenz und
Manuskripte bitte an

CBM/PET NEWS
Verlag SCC AG
Seeburgstrasse 12
CH-6006 Luzern

Die Beiträge stammen grösstenteils von Clubmitgliedern oder sind gekürzte Übersetzungen. Für die Veröffentlichung wird keine Gewähr oder Garantie übernommen, auch nicht dafür, dass die verwendeten Schaltungen, Firmennamen und Warenbezeichnungen frei von Schutzrechten Dritter sind. Die Verwendung der Informationen erfolgt auf eigenes Risiko.

Copyright by SCC Lucerne, aber Speicherung in Datenverarbeitungsanlagen für den eigenen Gebrauch erlaubt.

Das Jahresabonnement (6 Ausgaben) kostet für Mitglieder des Schweizer Computer Club Fr. 18.-/DM 21.-, für Abonnenten von **Mikro- und Kleincomputer** Fr. 21.-/DM 24.- und für Nichtabonnenten Fr. 48.-/DM 55.-. Bereits erschienene Ausgaben des Jahrganges werden nachgeliefert. Verlag SCC AG, CH-6002 Luzern, Postkonto Luzern 60-27181; Stuttgart 3786-709 (BLZ 600 100 70) oder Euroscheck.

Commodore: Wegbereiter des Jedermann-Computer



C commodore
Commodore AG · Dufourstraße 9 · 4010 Basel · Tel. 061/23 78 00 · Telex 64 961

Autorisierte Commodore-Wiederverkäufer mit technischem Kundendienst

Aarau Dahms Computersysteme · Tel. (064) 22 77 66. **Basel** BD-Electronic · Tel. (061) 35 36 37. Geiger-Microcomputer · Tel. (061) 44 13 13. **Leobag** Computer AG · Tel. (061) 35 31 14. **SYSAG** Systems & Services AG · Tel. (061) 38 21 20. **Thürlemann** Discount · Tel. (061) 22 41 66. **Bern** Computerland AG · Tel. (031) 24 25 54. **Radio TV** Steiner AG · Tel. (031) 55 45 81. **Biel** EIM Computer AG · Tel. (032) 23 15 88. **Brugg** Megos AG · Tel. (056) 41 34 17. **Fontainemelon** Urs Meyer Electronic · Tel. (038) 53 43 43. **Freiburg** Sovitrel SA · Tel. (037) 22 78 37. **Genf** Centre Informatique Gesmarco · Tel. (022) 21 11 75. **Egg-Telsa** SA ·

Tel. (022) 20 06 00. **Gesmarco** SA (Thônex) · Tel. (022) 49 88 44. **Irco** Electronic · Tel. (022) 20 33 06. **Radio TV** Steiner AG · Tel. (022) 28 52 22. **Gossau** Pius Schäfler · Tel. (071) 85 13 87. **Huttwil** Compu-Life, Rüfenacht AG · Tel. (063) 72 11 13. **Interlaken** Datatechnik · Tel. (036) 22 10 21. **Lausanne** Mafioly SA · Tel. (021) 22 00 44. **Schaer** Informatique · Tel. (021) 23 55 55. **Lugano** Luigi Chiodoni · Tel. (091) 23 23 09. **Luzern** Dialog Computer Treuhand AG · Tel. (041) 31 45 45. **Helfenstein + Bucher** AG · Tel. (041) 22 13 43. **Schweizer** Computer Club · Tel. (041) 31 45 45. **Magliaso** Marah SA · Tel. (091) 71 14 28. **Mellingen** Instant-Soft AG · Tel. (056) 91 20 21. **Niederrohrdorf** Nöthiger Electronic · Tel. (056)

96 28 96. **Rüti/ZH** Logon AG · Tel. (055) 31 72 30. **Schaffhausen** PIM Systems · Tel. (053) 45 45 0. **Syntron** Electronic · Tel. (053) 5 33 77. **Sion** Sphère Corporation · Tel. (027) 22 68 14. **St.Gallen** LASYS Tel. (071) 28 39 05. **Thun** HMB electronic · Tel. (033) 22 66 88. **Wettingen** Elbatex AG · Tel. (056) 26 98 27. **Winterthur** Nowak AG · Tel. (052) 22 08 03. **Wohlen/AG** SYSAG Systems & Services AG · Tel. (057) 6 36 50. **Zürich** Furrer Büro-Computer · Tel. (01) 202 49 92. **Hannes Keller** AG · Tel. (01) 69 36 33. **Logon** AG · Tel. (01) 62 59 22. **Microspot** AG · Tel. (01) 241 20 30. **Erhard** Wipf AG · Tel. (01) 221 21 00.