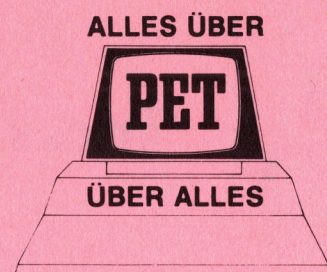


6
80

24. Nov.

SCC PET NEWS

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



Lieber PET-Freund

Mit dieser Ausgabe ist der erste Jahrgang der SCC PET NEWS komplett. Wir hoffen, Ihre Erwartungen, die Sie an diese Zeitschrift gestellt haben, sind erfüllt worden. Aus Ihren Reaktionen und Beiträgen haben wir in diesem ersten Jahr bereits eine ganze Fülle von wertvollen Fingerzeigen und Anregungen erhalten. Dadurch sollte es uns möglich sein, noch mehr auf Ihre speziellen Bedürfnisse einzugehen.

Damit wir aber ganz sicher gehen, machen wir mit dieser Ausgabe gleichzeitig eine Meinungsumfrage. Machen Sie sich bitte die Mühe, benützen Sie die beiliegende Karte und teilen Sie uns Ihre Wünsche, Anregungen und Kritik nicht nur zum gesamten Heft, sondern auch zu den einzelnen Rubriken mit. Mit Hilfe dieser spezifischen Daten wollen wir dann die weiteren Hefte im Jahr 1981 gestalten. Es liegt bei Ihnen, wieviel Sie von uns profitieren können. Bitte senden Sie uns die ausgefüllte Karte möglichst umgehend retour. Für Ihr aktives Mitmachen danken wir Ihnen. Die Auswertung der Umfrage und die Konsequenzen, die wir daraus ziehen, werden wir in einem der ersten Hefte 1981 veröffentlichen.

Für Ihre bisherige Treue möchten wir Ihnen ein ganz herzliches Danke schön sagen. Wir freuen uns, wenn wir Sie auch im zweiten Jahrgang weiterhin zu unseren Lesern zählen dürfen. Aus diesem Grund liegt dieser Ausgabe ein Einzahlungsschein/Zahlkarte bei. Die Abonnementsgebühren für weitere sechs Nummern betragen wie bisher Fr. 18.-- für Mitglieder des Schweizer Computer Club, Fr. 21.-- für Abonnenten von m&k computer und Fr. 48.-- für Nichtmitglieder. Sichern Sie sich durch Einzahlung des Betrages die weiteren Hefte dieser News Letters für Ihren Computer.

Ab der ersten Ausgabe im neuen Jahr trägt Ihre Zeitschrift den neuen Titel CBM/PET NEWS. Verschiedene Gründe haben uns zu dieser geringfügigen Titeländerung veranlasst. Commodore bezeichnet schon seit geraumer Zeit seine Geräte nur noch als CBM. Das liegt zum Teil daran, dass in Deutschland der Name PET bereits von einer anderen Firma für ein anderes Produkt geschützt ist. Wir finden es aber unerheblich ob CBM oder PET, schliesslich verbirgt sich hinter diesen drei Buchstaben einer der populärsten Kleincomputer unserer Tage. Wir kommen damit auch unseren vielen Computerfreunden in Deutschland und Oesterreich entgegen. Gerade von dort werden wir immer wieder angefragt, was denn nun der Unterschied zwischen CBM und PET sei. Es gibt keinen.

Wir wünschen Ihnen zur bevorstehenden Jahreswende alles Gute, viel Erfolg und Vergnügen mit Ihrem CBM/PET.

Die Redaktion



Tricks und Tips

BASIC - TOOLKIT IN SOFTWARE

Der Toolkit - ein ROM, das zusätzlich zum bisherigen Befehlssatz des PET 10 weitere Befehle enthält - wurde in Heft 1 und 2 der PET NEWS eingehend besprochen. Diese 10 zusätzlichen Befehle bringen dem Programmierer wesentliche Erleichterungen vor allem dann, wenn sich Fehler in das Programm eingeschlichen haben und behoben werden sollen. Nun ist aber die Anschaffung eines solchen ROM eine Preisfrage. Ausserdem ist der Einsatz des Toolkit mit gewissen Schwierigkeiten verbunden, da er auf einigen PET-Serien nur mit gewissen Einschränkungen möglich ist. Weiterhin verlangsamte sich nach Initialisieren dieses ROM's die Programmausführung wesentlich.

Viele unserer Leser und natürlich auch wir haben uns daher Gedanken gemacht, ob es nicht möglich sei, einen solchen Toolkit oder zumindest gewisse Befehle softwaremässig zu erzeugen. Es geht.

Einen solchen Software Toolkit wollen wir Ihnen hier vorstellen. Zu beachten ist, dass die Programme, die in Maschinensprache geschrieben sind, gegenüber den BASIC-Programmen einen wesentlichen Vorteil haben, da die Maschinenprogramme im 2. Kassettenbuffer abgelegt sind und daher durch das Ueberschreiben mit BASIC und vor dem Löschen durch den NEW-Befehl geschützt sind, ausserdem arbeiten sie schneller als entsprechende BASIC-Programme. Die Programme beginnen alle mit der Zeilennummer 60000, um Sie an bestehende Programme anzuhängen. Sollen mehrere solcher Toolkitprogramme angehängt werden, sind sie erst umzunummerieren und mit END-Befehlen zu versehen, damit sie mit RUN (Zeilennummer) einzeln aufgerufen werden können.

APPEND

Mit diesem Befehl können Programme ab Kassette an Programme, die sich im Speicher des Computer befinden, angehängt werden. Dies ist dann wichtig, wenn sie die Toolkit-Befehle an bestehende Programme anhängen wollen. Mit einem kleinen Trick ist dies möglich ohne ein zusätzliches Programm, indem man den Bildschirmspeicher als Zwischenspeicher benutzt.

Voraussetzung ist, dass sich die Zeilennummern nicht überschneiden und das Programm das auf dem Bildschirm gespeichert werden muss, nicht länger als 20 Zeilen ist. Letzteres ist meist dadurch möglich, indem man möglichst viele Befehle in eine Zeile schreibt also die Zeile voll ausnutzt. Gehen Sie nun wie folgt vor:

1. Laden Sie ab Kassette zuerst das Programm, das angehängt, also auf dem Bildschirm gespeichert werden soll.
2. Listen Sie das Programm und schieben Sie es durch Drücken der RETURN-Taste an den oberen Bildschirmrand.
3. Geben Sie NEW ein, löschen Sie aber auf gar keinen Fall den

Bildschirm

4. Mittels Cursor und Space nehmen Sie bis auf das gelistete Programm alle überflüssigen Texte wie NEW und READY weg.
5. Fahren Sie nun mit dem Cursor direkt unter das gelistete Programm.
6. Geben Sie LOAD ein und laden Sie das längere Programm.
7. Drücken Sie die HOME Taste (nicht CLEAR) und drücken Sie sooft die RETURN-Taste, wie das auf dem Bildschirm befindliche Programm Zeilen hat.

Das auf dem Bildschirm befindliche Programm und das zuletzt geladene Programm hängen jetzt zusammen.

DELETE

Mit Delete können Sie nicht nur einzelne Zeilennummern, sondern ganze Programmteile löschen. Das dazugehörige Programm haben wir bereits in Heft 4/80 gebracht, da es aber nur auf dem Neuen Betriebssystem läuft, bringen wir Ihnen in Ergänzung auch die Version für den alten PET.

```
60000 REM DELETEPROGRAMM
60001 REM PROGRAMMIERT DURCH U. KISSLING
60002 INPUT "VON ADRESSE ? BIS ADRESSE";A,E:Y=A:D=E-A:IFD>5THEND=0
60004 A#="GOTO60003":B#="60003 Y=":C#=":E=":IFD=0THEND=6
60005 Y=Y+6:IFY>ETHENY=E:A#="60003"
60006 PRINT "###":PRINTB#;Y;C#;E:FORI=Y-DTOY:PRINTI:NEXT
60007 POKE 158,9:PRINTA#"#":FORI=1TO9:POKE622+I,13:NEXT:END
```

VERSION ALTER PET: STATT 158 IST 525 UND
STATT 622 IST 527 EINZUSETZEN. REM - WEGLASSEN !

Sie brauchen lediglich die erste und letzte Zeilennummer einzugeben, die Sie löschen wollen und nach Drücken der RETURN-Taste werden die eingegebenen Programmteile gelöscht.

FIND

Mit dem Befehl FIND können Sie Befehle oder Variablen im gesamten Programm suchen. Geben Sie z. B. ein FIND GOSUB, so werden nach Drücken der RETURN-Taste alle Zeilen auf dem Bildschirm aufgelistet, in denen der Befehl GOSUB vorkommt. Es handelt sich hier um einen sehr nützlichen Befehl, der vor allem dann eine Rolle spielt, wenn kein Drucker zur Verfügung steht.

```
60000 REM FIND
60010 REM H.KASTIEN LUZERN 11.80
60020 A = 1025
60030 B = PEEK(1029)
60040 FOR I = 1 TO 1000
60050 FOR C = A + 4 TO A + 83
60060 D = PEEK(C)
60070 IF C = B THEN GOSUB 60120
60080 IF D <> 0 THEN NEXT C
60090 A = 256 * PEEK(A+1) + PEEK(A)
60100 IF A > 0 THEN NEXT I
60110 STOP
60120 FOR E = 1 TO 80
60130 S = PEEK(1029 + E)
60140 IF S = 0 THEN PRINT 256*PEEK(A+3) + PEEK(A+2);
60150 RETURN
```

TRACE

Mit dem Trace Befehl kann ein Programm verlangsamt ausgeführt werden, sodass die einzelnen Programmschritte verfolgt werden können. Bei dem aufgeführten Trace-Programm handelt es sich um ein Maschinenprogramm. Nach Laden des Programms ist zuerst RUN einzugeben, um das Programm in den zweiten Kassettenspeicher zu laden, danach kann mit SYS (893) das Programm zu jeder Zeit aufgerufen werden.

```

1 REM *****
2 REM *TRACE PROGRAM FOR BASIC 3.00 ROM
3 REM *MACHINES. DISPLAYS EACH LINE OF A
4 REM *PROGRAM ON THE SCREEN AS IT IS
5 REM *EXECUTED. USE SYS COMMANDS SHOWN
6 REM *WHEN RUNNING THE PROGRAM. NOTE
7 REM *THESE VALUES EACH TIME SINCE THEY
8 REM *WILL VARY.
9 REM *****S.SIMDES*****
10 E=52
15 D=2
20 DATA-342,162,5,189,249,224,149,112,202,16,248,169,239,133,128,96
30 DATA173,-342,133,52,173,-341,133,53,169,255,133,42,160,0,162,3
40 DATA134,43,162,3,32,-271,208,249,202,208,248,32,-271,32,-271,76
50 DATA121,197,162,5,189,-6,149,112,202,16,248,169,242,133,128,96
60 DATA230,42,208,2,230,43,177,42,96,230,119,208,2,230,120,96
70 DATA32,115,0,8,72,133,195,138,72,152,72,166,55,165,54,197
80 DATA253,208,4,228,254,240,106,133,253,133,35,134,254,134,36,165
90 DATA152,208,14,169,3,133,107,202,208,253,136,208,250,198,107,208
100 DATA246,32,-54,169,160,160,80,153,255,127,136,208,250,132,182
110 DATA37,132,38,132,39,120,248,160,15,6,35,38,36,162,253,181,132
120 DATA40,117,40,149,40,232,48,247,136,16,238,216,88,162,2,169
130 DATA48,133,103,134,102,181,37,72,74,74,74,32,-44,104,41
140 DATA15,32,-44,166,102,202,16,233,32,-38,32,-38,165,184,197,119
150 DATA240,55,165,195,208,4,133,253,240,47,16,42,201,255,208,8
160 DATA169,105,32,-30,24,114,33,41,127,170,160,0,185,145,192,48
170 DATA3,200,208,248,200,202,16,244,185,145,192,48,6,32,-32,200
180 DATA208,245,41,127,32,-32,165,119,133,184,104,168,104,170,104
190 DATA96,168,173,64,232,41,32,208,249,152,96,9,48,197,103,208,40
200 DATA4,169,32,208,2,198,103,41,63,9,128,132,106,32,-54,164,182
210 DATA153,0,128,192,195,208,2,160,7,200,132,182,164,106,96,76
220 DATA-255,32,-262
300 S2=PEEK(E)+PEEK(E+1)*256:S1=S2+D-344
310 FORJ=S1TOS2-1
320 READX:IFX>0ORX=0THENGOTO350
330 Y=X+S2:X=INT(Y/256):Z=Y-X*256
340 POKEJ,Z:J=J+1
350 POKEJ,X
360 NEXTJ
400 PRINT"*****TRACE"
410 PRINT"*****"
500 PRINT"INITIALISE WITH SYS(;"S1+17")"
510 PRINT"ENABLE WITH SYS(;"S1+56")"
520 PRINT"DISABLE WITH SYS(;"S1+2")"
530 PRINT"CHANGE SPEED WITH POKE;"S1+125-D",X"
540 END

```

REPEAT

Die REPEAT oder Wiederholfunktion ist ein Befehl, der im Befehlsatz des TOOLKIT nicht vorhanden ist.

Mit REPEAT soll bewirkt werden, dass eine Taste so oft ausgeführt wird, solange die Taste gedrückt ist. Der Befehl ist vor allem beim Zeichnen langer Striche oder ähnlichem von Nutzen. Wir bringen eine BASIC- und eine Maschinensprachversion. Auf die Vorteile der Maschinenversion haben wir schon eingangs hingewiesen.

```
60000 REM MINIINTERUPT, REPEAT
60010 REM CH. BUESS REINACH
60020 REM REPEAT EINSCHALTEN UND AUSSCHALTEN MIT SYS 893
60030 REM MININTERRUPT EIN MIT SYS 910
60040 REM MININTERRUPT AUS MIT SYS 945
60050 FOR I=826 TO 955:READ A:POKE I,A : NEXT
60060 DATA 165,151,201,255,240,8,205,140,3,240,11,141,140,3,169,0
60070 DATA 141,141,3,76,46,230,173,141,3,201,42,240,6,238,141,3
60080 DATA 76,46,230,233,3,141,141,3,174,140,3,189,247,230,166,152
60090 DATA 240,3,24,105,128,141,111,2,169,2,133,168,169,1,133,158,76
60100 DATA 46,230,120,169,20,69,144,133,144,169,229,69,145,133,145,88
60110 DATA 96,27,0,165,144,141,188,3,165,145,141,189,3,169,161,133
60120 DATA 144,169,3,133,145,96,32,234,255,201,255,240,3,32,177,3
60130 DATA 104,168,104,170,104,64,173,188,3,133,144,173,189,3,133,145,
```

96

```
60000 REM REPEAT FUNKTION
60010 REM H.KASTIEN LUZERN 11.80
60020 GET A$
60030 IF A$ = " " THEN 60020
60040 Q = PEEK(151)
60050 A = 6
60060 PRINT A$;
60070 T = TI + A
60080 IF PEEK(151) = 255 THEN 60020
60090 IF TI < T THEN 60080
60100 A = 6
60110 GOTO 60060
```

Gleichzeitig behebt das Programm einen weiteren Nachteil, nämlich der lange Interrupt des PET, denn auch während eines Programms wird die ganze Tastatur abgefragt, die interne Uhr weitergezählt und die Kassettentastatur überprüft. Es ist möglich, den Interrupt ganz zu unterdrücken (SEI), was einen Zeitgewinn von knapp 10 % ergibt, aber auch den grossen Vorteil hat, dass die Stop-Taste nicht mehr aktiv ist.

Mit Hilfe des folgenden Maschinenprogrammes werden beide Nachteile behoben. Drei Befehle sind möglich:

```
SYS893 Repeat aus/ein
SYS910 Mini-Interrupt ein
SYS945 Mini-Interrupt aus
```

Anmerkungen:

1. Die Repeat-Fkt. funktioniert mit sämtlichen Tasten, also auch speziell für die Cursor-Bewegungen.
2. Während des Repetierens eines Zeichens wird das Blinken des Cursors unterbunden, man sieht auf diese Weise besser, wo er sich gerade befindet.
3. Wiederholrate bzw. Zeit bis zum Einsetzen der Repeat-Fkt. kann verändert werden ($\emptyset 354$ bzw. $\emptyset 35E$).
4. Beide Programme funktionieren auch mit Toolkit
5. Der "Mini-Interrupt" zählt nur die Uhr weiter und testet, ob

5

- die Stop-Taste gedrückt ist.
6. SYS 945 schaltet den "Mini-Interrupt" aus und stellt den vorherigen IRQ-Vektor wieder her. War also z. B. vor dem Aufruf (SYS910) die Repeat-Fkt. eingeschaltet, so ist sie es nach dem Ausschalten auch wieder.
 7. Wird das Programm durch eine Fehlermeldung unterbrochen, so muss nach dieser noch die Stop-Taste gedrückt werden, damit der Mini-Interrupt sich abschaltet.
 8. Vor längeren Rechnungen, d.h. nach dem letzten Input, sollte der Mini-Interrupt eingeschaltet werden. Er bewirkt einen Zeitgewinn von etwa 8.5 % gegenüber normal, und mit Toolkit ist der PET etwa gleichschnell wie vorher ohne.
 9. Zum Laden bzw. Abspeichern von Programmen muss die Repeat-Fkt. abgeschaltet werden.

ROM LISTING (FORTSETZUNG)

```

F63C-F655 Got start & end program addresses from tape header
F656-F66B Set cassette buffer address according to
           device number
F66C-F683 Set tape start & end addresses from buffer address
F684-F68C Perform CMD
F68D-F69D Set tape start & end addresses from Basic pointers
F69E-F728 Perform SAVE
F729-F76C Update TI and TI%, and copy STOP key to work area
F76D-F76F TI constant: limit of clock (24 hours)
F770-F78B Set input device
F78C-F805 Set output device
F806-F811 Advance tape buffer pointer (for INPUT , GET , and
           PRINT )
F812-F834 Wait: PRESS PLAY ON TAPE
F835-F846 Test if cassette button(s) pressed
F847-F854 Wait: PRESS PLAY & RECORD ON TAPE
F855      Initiate tape read
F886-F8E5 Initiate tape write
F8E6-F8EF Test for I/O interrupt completion
F8F0-F8FF Test stop key
F900-F930 Set expected timing for next input bit from tape
F931      Interrupt entry: Read tape bits
FA57-FB75 Store received tape characters
FB76-FD7E Set tape read/write address back to starting point
FB7F-FB83 Flag I/O error into ST
FB84-FB92 Reset 8-count and flags for a new byte
FB93-FBAE Write a transition to cassette tape
FBAF-FC40 Write interrupt 2: write data to tape
FC41-FC7A Write interrupt 1: write tape shorts (leader)
FC7B-FC95 Terminate tape: restore normal interrupt vector
FC96-FCA5 Set interrupt vector from table
FCA6-FCB3 Turn off cassette motors
FCB4-FCC5 Perform running checksum calculation
FCC6-FCDD Advance read/write pointer
FCD1-FCFD Power-on reset entry point
FCFE-FD00 NMI interrupt entry point
FD01-FD10 Table of interrupt vectors
FD11-FFB0 Machine Language Monitor (MLN) - see Commodore
           documentation
FFB1-FFBF CBM copyright statement
*****Jump Table*****
FFC0 OPEN
FFC3 CLOSE
FFC6 Set input device
FFC9 Set output device
FFCC Restore default I/O devices
FFCF Input character
FFD2 Output character
FFD5 LOAD
FFD8 SAVE
FFDB VERIFY
FFDE SYS
FFE1 Test stop key
FFE4 Out character
FFE7 Abort all I/O activity
FFEA Clock update
FFF0-FFF9 unused
FFFA-FFFF Hardware vectors: NMI, Reset, Interrupt

```

Hardware

ANALOG - DIGITAL - WANDLER

PET COMPATIBLE MESSGERÄTE

Nahezu alle Signale, die uns erreichen, haben analogen Charakter, die von einem Computer nicht direkt verarbeitet werden können. Es ist daher unbedingt erforderlich, diese Signale in digitale Informationen umzuwandeln. Die Wandlung erfolgt in den meisten Fällen in 8 bit Signale, die dem Computer über den USER PORT oder den IEC-Bus zugeführt werden. Mit einem entsprechenden Interface ist es dann möglich, analoge Werte zu messen, zu registrieren und periphere Geräte zu steuern. Einige Beispiele sollen dies veranschaulichen.

Die direkte Messung von Spannungen und Stromstärken, Registrierung und Verarbeitung, sowie Anzeige auf dem Bildschirm

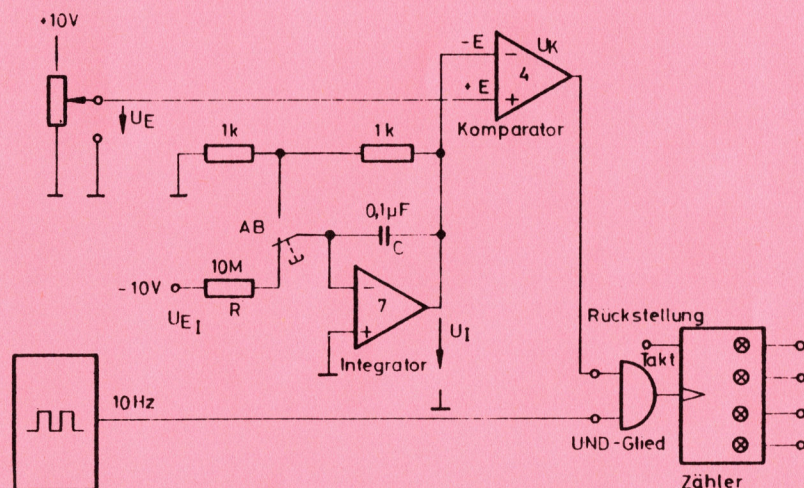
- Erfassung der Temperatur und Steuerung von Heizung oder Lüftung
- Erfassung der Raumhelligkeit und Steuerung der Beleuchtung
- Erfassung von Messdaten, Erstellen eines Protokolls und Ausdruck einer Kurve

Die Einsatzmöglichkeiten werden durch diesen Zusatz wesentlich erweitert und der PET kann durch diesen Ausbau für mannigfaltige Regel und Steueraufgaben im Labor, der Industrie und dem Haushalt eingesetzt werden.

Funktionsweise von A/D-Wandlern

Bei der A/D-Wandlung nach dem Sägezahnverfahren wird ein Vergleich zwischen einer linear ansteigenden Spannung und der Eingangsspannung vorgenommen. Es wird nun die Zeit, die bis zur Erreichung der Spannungsgleichheit verstreicht durch einen konstanten Taktgenerator ausgezählt und das Zählergebnis bis zum Beginn des nächsten Zählvorganges in einem binären Zähler festgehalten. Je höher die Eingangsspannung, desto länger dauert es, bis die Vergleichsspannung mit der Eingangsspannung identisch ist, folglich läuft der Taktgenerator länger und das Zählergebnis ist höher.

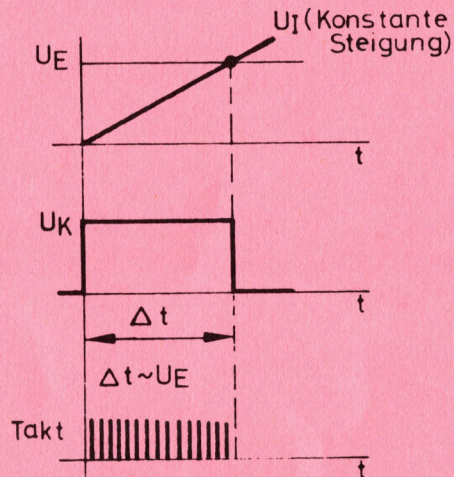
Im nachfolgenden Bild ist ein A/D-Wandler nach dem Sägezahnverfahren gezeigt.



Im Handel befinden sich eine ganze Reihe von Geräten, die zur Umwandlung von analogen Signalen herangezogen werden können, wobei die Auswahl eines geeigneten Modells nicht zuletzt eine Frage des Preises ist, da sich der Preis zwischen 180.-- und 8000.-- Franken bewegt. Wir wollen daher versuchen, einen Ueberblick über Geräte zu geben, die auch für den Normalverbraucher absolut erschwinglich sind, wenn nicht sogar der Weg des Selbstbaus gewählt wird.

Der Taktgenerator erzeugt die konstante Frequenz, die aber über das UND-Glied nur dann auf den Zähler gelangen kann, wenn gleichzeitig der Ausgangs des Komparators auf High-Level liegt, also das UND durchsteuert. Der Integrator erzeugt eine langsam ansteigende Vergleichsspannung, die im Komparator mit der Eingangsspannung verglichen wird. Mit dem Schalter AB wird der Beginn des Vergleichsvorgangs gestartet.

Der Komparator ist ein Vergleicher, dessen Ausgang solange auf High-Level liegt, solange die beiden Eingänge nicht die gleiche Spannung haben. Das Diagramm zeigt den Verlauf der Spannungen am Eingang, am Ausgang des Komparators und die Taktimpulse.



Einkanal-Analog/Digital-Wandler zum Anschluss an den USER PORT

Die einfachste und vor allem billigste Lösung der Umsetzung analoger Signale besteht in der Umwandlung in ein 8-bit Parallelsignal, das dem USER PORT zugeführt wird, dort kann es mittels PEEK (59457) abgefragt und verarbeitet werden. Diese kleinen Zusatzplatinen werden direkt auf den USER PORT aufgesteckt und mit dem zweiten Kassettenrecorderanschluss mit der erforderlichen Speisepannung von 5 V versorgt. Die maximale Eingangsspannung beträgt 1.999 V kann aber durch entsprechende Eingangsspannungsteiler wesentlich erweitert werden. Die Genauigkeit der Geräte beträgt 0.1070, die Platinen verfügen über automatische Nullpunkteinstellung und Overflowanzeige sowie positive und negative Vorzeichen. Einfachen Messaufgaben sind diese Geräte also absolut gewachsen. Dieser A/D-Wandler ist sowohl als Bausatz unter der Nummer P 9946 zu Fr. 110.- als auch als Fertiggerät P 9947 zu Fr. 180.- inkl. Software beim SCC erhältlich. Durch Einsatz eines solchen A/D-Wandlers am USER PORT wird dieser natürlich blockiert und kann nicht mehr zu anderen Steueraufgaben, z. B. zum Schalten von Peripheriegeräten benutzt werden.

Multimeter mit BCD-Ausgang und Interface mit adressierbarem IEC-Bus

Durch den Preiseinbruch auf dem Messgerätesektor sind auch handelsübliche Digitalmultimeter heute in einem annehmbaren preislichen Rahmen. So wird z. B. von der Fa. Neukom in ZH der Digitalmultimeter 3209 mit BCD-Ausgang zu einem Preis von nur Fr. 485.-- angeboten. Dieses Instrument kann also zu normalen Messaufgaben herangezogen werden, verfügt es doch über folgende Messbereiche:

DCV 100	/ μ V - 1000 V	100 MOhm	0.1 %
ACV 100	/ μ V - 1000 V	2 MOhm	0.5 %
DCA 0.1	/A - 2 A		0.4 %
ACA 0.1	/A - 2 A		0.7 %
Cap. 1	pF - 20 / μ F		1.5 %

Mit Shunt kann das Instrument auf 10 A und mit Hochspannungssonde auf 30 kV ausgebaut werden, ausserdem ist ein Temperaturadapter lieferbar. Das wichtigste bei diesem Instrument ist der gemultiplexte BCD-Ausgang, mit dem man einerseits direkt auf den USER PORT gehen kann. Andererseits kann durch Zwischenschalten eines Interface das BCD-Signal auf den IEC-Bus eingespielen werden.

Erst durch dieses Interface wird das Messgerät voll einsatzfähig. Es ist nun möglich, das Interface mit einer Gerätenummer zwischen 4 und 15 zu versehen und neben anderen Geräten an jeder beliebigen Stelle des Programms aufzurufen und das Messergebnis mit Input einzulesen und weiter zu verarbeiten. Selbstverständlich können nun auch mehrere Geräte nebeneinander betrieben werden und auf diese Art beliebig viele Daten aufgenommen und verarbeitet werden. Das Interface, es passt übrigens zu jedem Computer mit IEC-Bus, kostet incl. Software nur Fr. 490.-- und ist durch das Ing. Büro Steinlin, Riedenerstr. 17, CH-8305 Dietlikon zu beziehen.

Ein völlig neuartiges Interface ist inzwischen auch beim SCC erhältlich, das vor allem dann interessant ist, wenn mehrere analoge Signale verarbeitet werden sollen. Es handelt sich um einen A/D-Wandler mit 16 Eingängen, die ebenfalls adressierbar sind. Das Interface wird im PET auf den Stecker zur Speicher-Erweiterung eingesteckt. Die Adressierung der Eingänge erfolgt ab Adresse 8F80. Die Analogwerte an den einzelnen Eingängen können also durch Print PEEK (37636) - Print PEEK (37651) eingelesen werden. Die maximale Eingangsspannung beträgt pro Kanal 10 V. Das Interface ist beim SCC zum Preis von Fr. 850.-- erhältlich.

Unter den sogenannten Systemmultimetern, hierunter versteht man Digitalmultimeter, die bereits mit einem IEC-Interface ausgerüstet sind, sticht vor allem das Keithley Instrument 179 ins Auge, das zu einem Preis von Fr. 1490.-- excl. Wust durch die Fa. Seyffert ZH geliefert wird. Das Instrument mit folgenden Messbereichen:

200 mV - 1200 V DC	0.04 %
202 mV - 1000 V AC	0.06 %
200/ μ A - 2000 mA DC	0.2 %
200/ μ A - 2000 mA AC	0.2 %

200 μ A - 2000 mA AC	0.2 %
24 - 20M	0.1 %

kann also ebenfalls direkt mit einem IEC-IEC-Kabel an den Bus des PET angeschlossen werden und wie bereits oben erwähnt, die Daten eingelesen werden. Im Gegensatz zu der oben erwähnten preisgünstigeren Variante, ist dies Instrument zusätzlich mit einer Triggermöglichkeit versehen, die über den Computer gesteuert wird. Das nachfolgende Programm zeigt, wie einfach die Daten in den Computer übernommen werden können.

```

10 OPEN 2,24
20 INPUT#2,D$
30 IF ST = 2 THEN 20
40 PRINT"XXXXXXXXXXXX";D$
50 GOTO 20
60 CLOSE 2
70 END

```

Selbstverständlich sind auch Instrumente im Handel, bei denen die Bereichsumschaltung und die Auflösung des Gerätes durch den Computer gesteuert werden, jedoch liegen diese Geräte in einem Preisniveau über Fr. 3000.-- und sind daher für den Amateur und Kleincomputerliebhaber nicht erschwinglich, daher soll hier auf die Besprechung dieser Geräte verzichtet werden.

Selbstverständlich können aber weitere Auskünfte zu den Geräten über den Leserdienst eingeholt werden.

WAS BRINGEN DIE PET NEWS 1981

Hardware

Digital-Analogwandler
7-teilige Fortsetzungsreihe über eine Eingangs- und Ausgangserweiterung des USER PORTS mit
Light PEN
Joystick
Verstärker
Anzeige usw.

Tricks und Tips

Benutzung von Printer und Plotter
Programmgestaltung zur sinnvollen Benutzung dieser Geräte
Arten des Ausdrucks
Arbeiten mit File
Hardcopyprogramme

Basic-Basic

Musikprogrammierung für Fortgeschrittene
Wetterprognose
Basic-Schach
Elektronik-Schaltungsentwurf
Kaufmännische Programme
Spielprogramme:
Der Schweizer Nationalsport
Mühlespiel

Maschinensprache

Fortsetzungsserie: Programmierung in Maschinensprache mit Assembler Monitoren

Programm des Monats BASIC-BASIC

G R A P H I K

Grafik und die graphische Gestaltung, beispielsweise von Spielprogrammen, ist etwas vom Kurzweiligsten, was man sich bei der Programmierung vorstellen kann, nicht zuletzt deshalb, weil man das Ergebnis des Programmes vor sich sieht. Aber auch bei professionellen Programmen ist die graphische Gestaltung ein wichtiger Faktor. Ein Programm sollte nicht nur seinen Zweck erfüllen, sondern auch graphisch so gestaltet und gegliedert sein, dass Zusammenhänge und Ergebnisse übersichtlich und leicht erkennbar sind.

Bei der Auswertung von mathematischen Funktionen und deren Ausdruck ist das graphische Moment eine absolute Notwendigkeit. Das nachfolgende Programm ist nun ganz der Graphik gewidmet. Es zeigt an einfachen Beispielen, wie Bilder und Bewegungen auf dem Bildschirm erzeugt werden können. Es ist weniger ein selbständiges Programm, sondern mehr eine Anleitung zur eigenen Kreativität. Leider beschränkt sich die durch Basic erreichbare Auflösung nur auf 25 Zeilen zu je 40 Punkten, also relativ wenig. Mit einigen Tricks lässt sich die Auflösung vervierfachen. Davon soll hier aber nicht die Rede sein, zumal dies mit Maschinensprache viel einfacher ist. Die Feinauflösung des Bildschirminhaltes mittels Maschinensprache wurde im Heft 5/80 besprochen und wird 1981 fortgesetzt. Weiter lässt sich mit Hardwarezusätzen die Auflösung des Bildschirms ebenfalls wesentlich erhöhen. Dazu sind dann allerdings Investitionen von ca. SFr. 1200.-- zu tätigen. Ein entsprechender Umbau des PET oder CBM auf hochauflösende Graphik bietet der SCC an.

Wir wollen Ihnen drei Methoden der Graphik vorstellen, nämlich mittels:

1. PRINT - Befehlen
2. POKE - Befehlen
3. Programmierung von Bewegungen

Es ist wichtig, dass Sie Heft 2/80 der SCC PET NEWS als Arbeitsunterlage für die nachfolgende Besprechung des Programmes zu Rate ziehen, da an dieser Stelle die theoretischen Grundlagen der Bildschirmprogrammierung aufgeführt sind.

Das Programm gliedert sich in 6 Blöcke

Zeile 100 - 360	und 640 - 730	eigenes Zeichnen von Bildern
Zeile 400 - 635		Zeichnen einer Spirale und Kreisen
Zeile 1000 - 1630		Schreiben des Wortes Graphik
Zeile 1700 - 1980		Zeichnen eines Schreibmaschinenbildes
Zeile 2000 - 2540		Zeichnen eines PET mit POKE
Zeile 3000 - 4040		Bewegte Schrift, pfeifender und fliegender Vogel

Teil 1

- Zeile 100 - 230 Es wird mittels INPUT bzw. GET-Befehlen nach der Zeile, (115) der Spalte, (120) dem Zeichen, (130) der Anzahl der Zeichen (140) und der Richtung (160 - 170) in der gezeichnet werden soll, gefragt.
In den Zeilen 110 und 190 werden jeweils die ersten beiden Zeilen des Bildschirms gelöscht, damit beim Ueberschreiben kein "Synthax Error" entsteht.
Je nach Richtung (Zeile 160, 170, 180 und 220) wird mit den Subroutinen in 250 (abwärts), 270 (aufwärts), 290 (rechts), 310 (links), 330 (schräg aufwärts), 350 (schräg abwärts) die Änderung der Variablen P definiert, die in Zeile 220 das erste Zeichen festlegt. Das Zeichen erfolgt mit POKE-BEFEHLEN (210)
- Zeile 640 - 730 Mittels Printbefehlen wird lediglich eine Erklärung zum Programm gegeben und mittels GET (Zeile 720) das eigentliche Zeichenprogramm in Zeile 100 aufgerufen.

Teil 2

- Zeile 400 - 630 Es werden Kreise und Spiralen gezeichnet
410 Festlegen des Kreismittelpunktes
420 - 450 Schlaufe für die obere Kreishälfte
430 - 440 Berechnen und Zeichnen des Kreises
460 - 490 Schlaufe für die untere Kreishälfte
500 Definieren eines kleineren Kreises
530 - 610 Schlaufen für obere (530 - 565) und untere (570 - 610) Kreishälfte

Teil 3

- Zeile 1000 - 1630 Es wird in grossen Schriftzeichen das Wort "Graphik" auf den Bildschirm geschrieben.
Zeile 1000 - 1045. Löschen des Bildschirms (1000), Dimensionierung der Felder (1010), Einlesen der Daten (1020 - 1040) und Definieren der Zahl 33286 = P als Bezugspunkt für die folgenden POKE-Befehle
Zeile 1050 - 1070 Schreiben des ersten geraden Balkens bis Mitte Buchstaben R
Zeile 1200 - 1260 Schreiben des unteren Schrägbalkens von R und K
Zeile 1090 - 1110 Schreiben der Buchstaben "APHI" und des senkrechten Balkens des K.
Zeile 1300 - 1350 Schreiben des oberen Schrägbalkens von K
Zeile 1160 Warteschlaufe
Zeilen 1500 - 1630 sind die DATA Statements zu den oberen POKE-Befehlen
Zeilen 1700 - 1980 Zeichnen eines Schreibmaschinenbildes wobei das eigentliche Zeichnen nur über inverse PRINT-Befehle erfolgt

Teil 4

In den Zeilen 2000 - 2540 wird auf den Bildschirm das Bild eines PET gezeichnet, wobei der eigentliche Zeichenvorgang wiederum mittels POKE-Befehlen erfolgt, wie bereits in Teil 2. Auch hier sind die Adressen der Zeichen in DATA-Befehle abgelegt. Da hier aber die Zeichen und somit deren Bildschirmcode ständig wechselt, sind diese ebenfalls als DATA-Statements festgesetzt. Jeweils das erste Statement ist die Differenz der Adresse zu 32000 und die zweite Zahl der Bildschirmcode des Zeichens.

Teil 5

Zeile 3000 - 4040 Bewegte Bilder. Auch hier können drei Untergruppen gemacht werden:

Zeile 3000 - 3080
 Zeile 3090 - 3230
 Zeile 3240 - 3580

Zeile 3000- 3080 Auf dem Bildschirm erscheint der Text "Auch bewegte Bilder sind möglich" von rechts nach links, anschliessend verschwindet der Text nach oben.

In Zeile 3010 wird der Cursor auf die Mitte des linken Bildrandes gesetzt und mit Zeile 3020 als Variable C% mit der Schleife Zeile 3000 je um 1 Spalte nach links geschoben. Mit Zeile 3030 werden die Buchstaben eingelesen, in Zeile 3040 wird der neue Buchstabe zum alten String dazugezählt und das Ergebnis neu ausgedruckt (Zeile 3050). Mit Leerzeilen wird die geschriebene Zahl nach oben geschoben.

Zeile 3090 - 3230 Nacheinander werden mittels einer Schlaufe 10 mal zwei verschiedene Bilder eines pfeifenden Vogels mit PRINT-Befehlen auf den Bildschirm auf der gleichen Stelle geschrieben, es entsteht dadurch der Eindruck einer Bewegung.

Zeile 3240 - 3580 Zeichnen einer Landschaft mit PRINT (Zeilen 3240 -3280).Einlesen der Zeichen mit Zeile 3290 - 3320 (oberer Flügelschlag)
 3330 - 3360 Gleicher Vorgang für Flügelschlag unten
 3510 Vorwärtsbewegung durch Addition eines Space
 3520 Festlegung der Flughöhe
 3530 Vorwärtsbewegung und Flügelschlag oben ausdrucken
 3550 - 3570 Vorwärtsbewegung und Flügelschlag unten ausdrucken

G R A P H I K

```
10 REM DIESES DEMONSTRATIONSPROGRAMM
20 REM ZEIGT DIE PROGRAMMIERUNG VON
30 REM GRAPHIK UND BEWEGUNGEN AN EIN-
40 REM FACHEN BEISPIELEN.
50 REM PROGRAMMIERT R. KONZ
60 REM LUZERN JULI 1980
70 GOTO 1000
90 REM ZEICHNEN EIGENER BILDER
100 PRINT"□":PRINT:PRINT:FOR I=1 TO 22:PRINT I:NEXT
110 FOR J=0 TO 79: POKE 32768+J,32: NEXT J
115 INPUT"□BEGINN: ZEILE ";Z
120 PRINT"□"TAB(22);:INPUT"SPALTE ";S
130 INPUT"□ZEICHEN: WELCHES ";Z$
140 PRINT"□"TAB(22);:INPUT"WIEVIELE ";M
150 Q=PEEK(32787):P=32808+Z*40+S
160 PRINT"□RICHTUNG: TIEF=1, HOCH=2, RECHTS=3, ";
170 PRINT"LINKS=4, SCHRAEG HOCH=5, SCHRAEG TIEF=6"
180 GET R :IF R=0 OR R>6 GOTO 180
190 FOR J=0 TO 79: POKE 32768+J,32: NEXT J
200 FOR I=0 TO M-1
210 POKE P,Q
220 ON R GOSUB 250,270,290,310,330,350
230 NEXT
240 GOTO 110
250 P=P+40
260 RETURN
270 P=P-40
280 RETURN
290 P=P+1
300 RETURN
310 P=P-1
320 RETURN
330 P=P-39
340 RETURN
350 P=P+41
360 RETURN
370 REM ZEICHNEN EINES KREISES
400 PRINT"□SIE KOENNEN KREISE MALEN"
410 R=10 :Z=33265
420 FOR X=-R TO R
430 Y=INT(SQR(R*R-X*X))
440 POKE Z+X-Y*40,81
450 NEXT
460 FOR X=R TO -R STEP -1
470 Y=INT(SQR(R*R-X*X))
480 POKE Z+X+Y*40,81
490 NEXT
500 IF R<>5 THEN R=5 :GOTO 420
510 FOR J=0 TO 2000 :NEXT J
515 REM ZEICHNEN EINER SPIRALE
520 PRINT"□ODER SPIRALEN"
530 FOR R=1 TO 11
540 Z=33265
545 FOR X=-R TO R
550 Y=INT(SQR(R*R-X*X))
560 POKE Z+X-Y*40,81
```

```
565 NEXT X
570 R=R+1
580 FOR X=R TO -R STEP -1
590 Y=INT(SQR(R*R-X*X))
600 POKE Z+X+Y*40,81
610 NEXT X
620 NEXT R
630 FOR J=0 TO 1000 :NEXT
635 GOTO 1700
640 PRINT"SELBSTVERSTAEENDLICH KOENNEN SIE "
650 PRINT
660 PRINT"AUCH EIGENE BILDER ZEICHNEN."
670 PRINT:PRINT
680 PRINT"GEBEN SIE EINFACH DIE ANTWORTEN AUF DIE "
690 PRINT"GESTELLTEN FRAGEN."
700 PRINT:PRINT:PRINT
710 PRINT"DRUECKEN SIE NUN IRGEND EINE TASTE"
720 GET V$: IF V#="" GOTO 720
730 GOTO 100
990 REM ZEICHNEN DES GRAPHIKTITELS
1000 PRINT"J"
1010 DIM T(160)
1020 FOR I=1 TO 160
1030 READ T(I)
1040 NEXT
1045 P=33268
1050 FOR I=1 TO 45
1060 POKE P+T(I),160
1070 NEXT
1080 GOSUB 1200: GOSUB 1200
1090 FOR I=52 TO 147
1100 POKE P+T(I),160
1110 NEXT
1120 GOSUB 1300: GOSUB 1300
1130 GOSUB 1200: GOSUB 1200
1140 I=160
1150 POKE P+T(I),160
1160 FOR J=0 TO 4000:NEXT
1170 GOTO 400
1200 I=I+1
1210 POKE P+T(I)-1,225
1220 I=I+1
1230 POKE P+T(I)-40,97
1240 I=I+1
1250 POKE P+T(I)-40,160
1260 RETURN
1300 I=I+1
1310 POKE P+T(I)+1,97
1320 I=I+1
1330 POKE P+T(I)-40,225
1340 I=I+1
1350 POKE P+T(I)-40,160
1360 RETURN
1500 DATA -216,-256,-257,-258,-259,-219,-179,-139,-99,-59,-19
1510 DATA 21,61,101,141,142,143,144,104,64,24,-16,-56,-57
1520 DATA -253,-213,-173,-133,-93,-53,-13,27,67,107,147,-252
1530 DATA -251,-250,-210,-170,-130,-90,-50,-51,-52,-12,-11
1540 DATA 29,69,70,110,150,-247,-207,-167,-127,-87,-47,-7,33
1550 DATA 73,113,153,-246,-245,-244,-204,-164,-124,-84,-44,-4
1560 DATA 36,76,116,156,-46,-45,-241,-201,-161,-121,-81,-41
1570 DATA -1,39,79,119,159,-240,-239,-238,-198,-158,-118,-78
1580 DATA -38,-39,-40,-235,-195,-155,-115,-75,-35,5,45,85,125
1590 DATA 165,-232,-192,-152,-112,-72,-32,8,48,88,128,168,-34
```

```

1600 DATA -33,-229,-189,-149,-109,-69,-29,11,51,91,131,171
1610 DATA -226,-186,-146,-106,-66,-26,14,54,94,134,174,-223
1620 DATA -183,-143,-144,-104,-64,-65,-25,15,16,56,96,97,137
1630 DATA 177
1690 REM ZEICHNEN EINES SCHREIBMASCHINENBILDES
1700 PRINT "AUCH SCHREIBMASCHINENBILDER "
1750 PRINT "M"
1760 PRINT "M"
1770 PRINT "M"
1780 PRINT "M"
1790 PRINT "M"
1800 PRINT "M"
1810 PRINT "I M"
1820 PRINT "I M"
1830 PRINT "I M B B"
1840 PRINT "M M M M B B B B C"
1850 PRINT "M M M O P B O M"
1860 PRINT "M M P M"
1870 PRINT "M M T T P M"
1880 PRINT "M M T P M"
1890 PRINT "M M T P B B M"
1900 PRINT "M M M"
1910 PRINT "M M M M M B B B F M"
1920 PRINT "M M M M M V B B V F F M"
1930 PRINT "M M M M M F F M"
1940 PRINT "M M M M M M E E E E E F F M"
1950 PRINT "M M M M M M F F F F M"
1960 PRINT "M M M M M M M M"
1970 PRINT "M M M M M M M M"
1980 FOR I=1 TO 3000 :NEXT
1990 REM ZEICHNEN DES PET
2000 PRINT "ODER GEWOEHNLICH ZEICHNEN"
2010 DIM A(300),B(300)
2020 P=32000
2030 FOR I=1 TO 275
2040 READ A(I),B(I)
2050 POKE P+A(I),B(I)
2060 NEXT
2070 FOR J=0 TO 3000:NEXT
2080 GOTO 3000
2100 DATA 861,79,862,99,863,99,864,99,865,99,866,99,867,99
2110 DATA 868,99,869,99,870,99,871,99,872,99,873,99,874,80
2120 DATA 900,103,915,101,940,89,955,84,980,72,995,71,1020,93
2130 DATA 1035,66,1060,66,1075,93,1100,71,1115,72,1140,84
2140 DATA 1155,89,1180,76,1181,100,1182,100,1183,100,1184,100
2150 DATA 1185,100,1186,100,1187,100,1188,100,1189,100,1190
2160 DATA 100,1191,100,1192,100,1193,100,1194,100,1195,122
2170 DATA 1222,76,1223,100,1224,100,1225,100,1226,100,1227
2180 DATA 100,1228,100,1229,100,1230,100,1231,100,1232,100
2190 DATA 1233,122,1297,79,1298,99,1299,99,1300,99,1301,99
2200 DATA 1302,99,1303,99,1304,99,1305,99,1305,99,1306,99
2210 DATA 1307,99,1308,99,1309,99,1310,99,1311,99,1312,99
2220 DATA 1313,99,1314,99,1315,99,1316,99,1317,99,1318,80
2230 DATA 1358,103,1398,122,1397,100,1396,100,1395,100,1394
2240 DATA 100,1393,100,1392,100,1391,100,1390,100,1389,100
2250 DATA 1388,100,1387,100,1386,100,1385,100,1384,100,1383
2260 DATA 100,1382,100,1381,100,1380,100,1379,100,1378,100
2270 DATA 1377,76,1337,101,1179,78,1218,78,1257,78,1196,77
2275 DATA 1237,77,1278,77,1417,101,1456,89,1496,84,1535,72
2276 DATA 1575,71,1614,89
2280 DATA 1654,76,1694,76,1695,100,1696,100,1697,100,1698
2290 DATA 100,1699,100,1700,100,1701,100,1702,100,1703,100

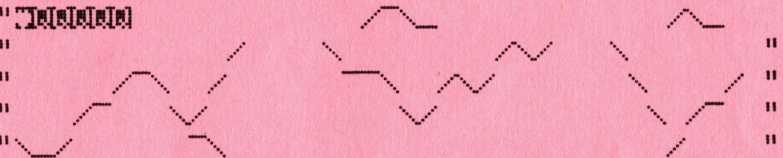
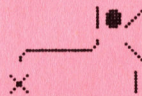
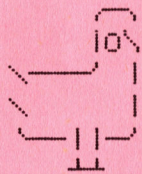
```



```

2300 DATA 1704,100,1705,100,1706,100,1707,100,1708,100,1709
2310 DATA 100,1710,100,1711,100,1712,100,1713,100,1714,100
2320 DATA 1715,100,1716,100,1717,100,1718,100,1719,100,1720
2330 DATA 100,1721,100,1722,122,1682,122,1681,100,1680,100
2340 DATA 1679,100,1678,100,1677,100,1676,100,1675,100,1674
2350 DATA 100,1673,100,1672,100,1671,100,1670,100,1669,100
2360 DATA 1668,100,1667,100,1666,100,1665,100,1664,100,1663
2370 DATA 100,1662,100,1661,100,1660,100,1659,100,1658,100
2380 DATA 1657,100,1656,100,1655,100,1438,103,1479,84,1519,89
2390 DATA 1560,71,1601,84,1642,84,1458,79,1459,99,1460,99,1461
2400 DATA 99,1462,99,1463,99,1464,80,1497,72,1537,71,1576,89
2410 DATA 1616,76,1617,100,1618,100,1619,100,1620,100,1621,100
2420 DATA 1622,122,1583,89,1544,71,1504,72,1506,79,1507,99
2430 DATA 1508,99,1509,99,1510,99,1511,99,1512,99,1513,80,1553
2440 DATA 103,1593,122,1592,100,1591,100,1590,100,1589,100
2450 DATA 1588,100,1587,100,1586,76,1546,101,1515,79,1516,99
2460 DATA 1517,80,1558,93,1598,122,1597,100,1596,76,1555,66
2470 DATA 1735,77,1736,100,1737,100,1738,100,1739,100,1740,100
2480 DATA 1741,100,1742,100,1743,100,1744,100,1745,100,1746
2490 DATA 100,1747,100,1748,100,1749,100,1750,100,1751,100
2500 DATA 1752,100,1753,100,1754,100,1755,100,1756,100,1757
2510 DATA 100,1758,100,1759,100,1760,100,1761,78,1345,16,1347
2520 DATA 5,1349,20,944,19,945,3,946,8,947,23,948,5,949,9,950
2530 DATA 26,951,46,1024,3,1025,15,1026,13,1027,16,1028,21
2540 DATA 1029,20,1030,5,1031,18,1106,3,1107,12,1108,21,1109,2
2990 REM BEWEGTER TITEL
3000 FOR I=1 TO 40
3010 PRINT"#####";
3020 C#=C#+"";PRINT C#;
3030 READ A#
3040 B#=B#+A#
3050 PRINT B#;
3060 FOR N=0 TO 100:NEXT N
3070 NEXT
3080 FOR I=1 TO 25:PRINT:FOR N=0 TO 50:NEXT N,I
3090 FOR I=1 TO 10
3095 REM PFEIFENDER VOGEL
3100 PRINT"#####";
3110 PRINT"
3120 PRINT"
3130 PRINT"
3140 PRINT"
3150 PRINT"
3160 PRINT"
3170 FOR N=0 TO 200 :NEXT N
3180 PRINT"#####";
3190 PRINT"
3200 PRINT"
3210 PRINT"
3220 FOR N=0 TO 200 :NEXT N
3230 NEXT I
3235 REM FLIEGENDER VOGEL
3240 PRINT"#####";
3250 PRINT"
3260 PRINT"
3270 PRINT"
3280 PRINT"#####";
3290 FOR I=1 TO 3
3300 READ Z#
3310 V#=V#+Z#+"#####";
3320 NEXT I
3330 FOR I=1 TO 3

```



```

3340 READ Y$
3350 X$=X$+Y$+"XXXXXXXXXX"
3360 NEXT I
3500 FOR I=1 TO 33
3510 J#=J#+ " "
3520 PRINT "XXXXXXXXXXXXXXXXXXXX"
3530 PRINT J#;:PRINTV$
3540 FOR N=0 TO 70:NEXT N
3550 PRINT "XXXXXXXXXXXXXXXXXXXX"
3560 PRINT J#;:PRINTX$
3570 FOR N=0 TO 70:NEXT N
3580 NEXT I
3900 GOTO 640
4000 DATA "A","U","C","H"," ","B","E","W","E","G","T","E"
4010 DATA "B","I","L","D","E","R"," ","S","I","N","D"
4020 DATA "M","O","E","G","L","I","C","H"," ","!"
4030 DATA " "," "," "," "," "," "
4040 DATA " \ /", "\ /", "|| \ /", "|| \ /", "|| \ /", "|| \ /", "|| \ /", "|| \ /", "|| \ /", "|| \ /"

```

**DAS INTERESSANTE PROGRAMM
IN LETZTER MINUTE**

SKIRENNEN

Die Skisaison rückt näher und damit auch die Zeit, in der man mit Spannung vor dem Bildschirm sitzt und den Abfahrtsrennen zusieht. Sicher kann es mancher kaum erwarten, bis dann die Resultate auf dem Bildschirm erscheinen und mit noch mehr Spannung wird auf die Rangliste gewartet.

Warum wollen Sie diesem Zustand nicht ein Ende bereiten, geben Sie während des Rennens die Resultate, Zwischenresultate, Namen und Nationalität in den PET ein und Sie erhalten sofort, noch vor der Rennleitung die Rangliste.

Wir erlauben uns daher, aus diesem aktuellen Anlass, die Spalte "Leichte Kost" für dieses Programm zu opfern.

```

10 PRINT"█          █":REM SCREEN CLR
20 PRINT"  █ISKI-RENNENI  "
30 PRINT"  █          █":PRINT:PRINT
40 REM
50 REM COPYRIGHT 1980 BY H.KAST,BALGACH
60 REM
70 PRINT"DIESES PROGRAMM ERSTELLT FUER MAX. 50":PRINT
75 PRINT"FAHRER EINE RANGLISTE, DIE ZEITEN":PRINT
80 PRINT"WERDEN EINGETIPPT IN FOLGENDER FORM ":":PRINT
82 PRINT"█ MIN.SEK.1/100SEK █. SCHEIDET EIN FAHRER ":":REM ZEIT RVS
85 PRINT"AUS, MUSS FUER DIE ZEIT EINE '0'":PRINT
90 PRINT"EINGEGEBEN WERDEN, FUER DIE NATION SIND":PRINT
92 PRINT"DREI BUCHSTABEN VORGESEHEN.":PRINT:PRINT
100 DIMN$(50),L$(50),Z$(50),Z(50)
110 I=1
120 PRINT"1.FAHRER ":":GOSUB1000
200 I=I+1:PRINT:PRINT"NAECHSTER FAHRER ":":GOSUB1000
210 J=I

```



```

220 IFZ(J)>=Z(J-1)THEN300
230 N#=N$(J-1):N$(J-1)=N$(J):N$(J)=N#
240 L#=L$(J-1):L$(J-1)=L$(J):L$(J)=L#
250 Z#=Z$(J-1):Z$(J-1)=Z$(J):Z$(J)=Z#
260 Z=Z(J-1):Z(J-1)=Z(J):Z(J)=Z
270 J=J-1:IFJ>1THEN220
300 J1=0:PRINT"DIE BESTEN:":PRINT:FORJ=1TO10:REM SCREEN CLR
302 IFN$(J)=""THENJ=10:GOTO306
304 GOSUB2000
306 NEXTJ
310 PRINT:PRINT:PRINT" 'N' = NAECHSTER FAHRER"
320 PRINT:PRINT" 'R' = RANGLISTE"
330 GETE$:IFE$=""THEN330
340 IFE$="N"THEN200
350 IFE$<>"R"THEN330
400 J1=0:GOSUB4000:FORJ=1TO15
402 IFN$(J)=""THENJ=15:GOTO406
404 GOSUB2000
406 NEXTJ:GOSUB3000
410 IFI<=15THEN500
420 GOSUB4000:FORJ=16TO30
422 IFN$(J)=""THENJ=30:GOTO426
424 GOSUB2000
426 NEXTJ:GOSUB3000
430 IFI<=30THEN500
440 GOSUB4000:FORJ=31TO45
442 IFN$(J)=""THENJ=45:GOTO446
444 GOSUB2000
446 NEXTJ:GOSUB3000
500 PRINT" 'N' = NAECHSTER FAHRER":REM SCREEN CLR
510 PRINT:PRINT" 'E' = ENDE"
520 GETE$:IFE$=""THEN520
530 IFE$="N"THEN200
540 IFE$<>"E"THEN520
550 PRINT"ENDE":END
1000 INPUT"NAME ";N$(I)
1010 N$(I)=N$(I)+" " :N$(I)=LEFT$(N$(I),15)
1020 INPUT"LAND ";L$(I)
1030 IFLEN(L$(I))>3THENPRINT"3 BUCHSTABEN":GOTO1020
1040 INPUT"ZEIT ";Z$(I)
1050 IFZ$(I)=""THENI=I-1:GOTO200
1060 IFLEN(Z$(I))=7THEN1080
1070 PRINT"ZEIT IN MIN . SEK . 1/100SEK ODER '0'":GOTO1040
1080 X#=LEFT$(Z$(I),1):M=VAL(X#)
1090 X#=MID$(Z$(I),3):S=VAL(X#)
1100 Z(I)=M*60+S:RETURN
2000 IFZ(J)=Z(J-1)THENJ1=J1+1:GOTO2010
2005 PRINTJ-J1;
2010 PRINTTAB(4)"■";N$(J);"■";TAB(20);L$(J);TAB(24);"■";Z$(J);"■";
2020 ZD=Z(1)-Z(J):ZD=INT(ZD*100+.5)/100
2030 PRINTTAB(32);ZD:RETURN
3000 PRINT:PRINT"FORTSETZUNG MIT █SPACE█":REM 'SPACE' RVS
3010 GETE$:IFE$=""THEN3010
3020 IFE$<>" "THEN3010
3030 RETURN
4000 PRINT"█ RANGLISTE":REM SCREEN CLR
4010 PRINT"█ ":PRINT:RETURN

```

READY.

Maschinensprache

Maschinensprache oder nicht Maschinensprache, die einen können mit der doch sehr abstrakten Materie recht wenig anfangen, andere wiederum finden die Maschinensprache als das Salz in der Suppe der Programmierer.

Heute bringen wir Ihnen den letzten Teil der Maschinensprachprogrammierung. Es sind dies die Fortsetzung der ROM-Listings und eine Liste der Maschinensprachbefehlen in deutscher Sprache.

In der Literatur findet man sie leider meistens nur auf Englisch. Detaillierterklärungen zu den Befehlen (Zeitablauf, Aenderung von Flags im Statusregister, Bedeutung, Anwendung, usw.) gibt es in ausgezeichnet geschriebenen deutschen Büchern. Es hat keinen Sinn, alle diese Einzelheiten in den PET-News zu beschreiben, da Sie mit dem Buch viel schneller vorwärts kommen und nicht lange auf Fortsetzungen warten müssen, wenn Sie konzentriert arbeiten wollen.

Maschinensprachprogramme speichert man indem man den entsprechenden Zahlencode in DATA-Befehlen ablegt und mittels POKE in die Speicherstellen einliest. Es ist allerdings unbefriedigend, die Befehle mittels Tabellen in Zahlen umzuwandeln und diese dann einzutippen. Diese Arbeit der Umsetzung in den Zahlencode kann ebensogut mittels Programm vorgenommen werden.

Ein solches Programm nennt man Assembler ("Zusammensetzer"). Mit diesem Assembler kann ein Programm bequem, schnell und erfolgreich realisiert werden.

Auf dem Markt gibt es allerdings verschiedene Arten von Assembler Programmen, die sich im Aufbau, in den Möglichkeiten und der Geschwindigkeit teilweise stark unterscheiden. Die Frage, welche Assembler sich für die verschiedenen Commodore-Computer eignen, werden wir im nächsten Heft in einer Uebersicht beantworten.

Vom Heft 1/81 an sind wieder vermehrt grössere Maschinenprogramme vorgesehen, z. B. ein Programm für Liniengrafik, das die im Heft 5/80 angeschnittene Idee weiterführt. (Auf einem Feld von 50x80 Punkten können Verbindungen von Punkt zu Punkt gezeichnet werden. Die Eingabe ist stark vereinfacht: Variablen werden erkannt und gerundet). Einige Programme befinden sich noch in der Entwicklungsphase. Um den Nutzen zu erhöhen, werden wir jeweils darauf achten, sowohl ein Maschinensprachelisting, wie auch ein Basic-Listing abzudrucken. Die Eingabe wird damit problemlos.

Ab Heft 1/1981 werden wir unter der Rubrik Maschinensprache daher das Arbeiten mit solchen Assemblern beschreiben.

Gleichzeitig werden wir dann vermehrt grössere Maschinensprachprogramme mit den erforderlichen Kommentaren abdrucken. Um den Nutzen der Programme zu erhöhen, schreiben wir zusätzlich zum Maschinensprachlisting ein BASIC-Listing. Die Eingabe der Programme wird somit problemlos und wir hoffen, dass auch diejenigen die bisher der Maschinensprache skeptisch gegenüber gestanden haben, ebenfalls ihre Freude an dieser interessanten Programmierart haben werden.

MASCHINENBEFEHLE

DEUTSCHE UEBERSETZUNG

- ADC Addiere Speicher und Akkumulator mit Uebertrag (carry), Resultat im Akkumulator
- AND Logische UND-Funktion mit Speicher und Akkumulator, Resultat im Akkumulator
- ASL Schiebe Akkumulator bzw. Speicher um ein Bit (eine Binärstelle) nach links (d. h. Multiplikation mal 2)
- BCC Verzweige, wenn das Uebertragsflag (carry flag) gelöscht (=0) ist (Test im Status-Register)
- BCS Verzweige, wenn das Uebertragsflag (carry flag) gesetzt (=1) ist (Test im Status-Register)
- BEQ Verzweige, wenn das Null-Flag (zero flag) gesetzt ist (d. h. wenn das Resultat gleich Null ist)
- BIT Logische UND-Funktion mit Speicher und Akkumulator, Resultat wird nicht gespeichert, aber Flags werden verändert (Bit-Test)
- BMI Verzweige, wenn das Negativ-Flag (negative flag) gesetzt ist, d. h. wenn das Resultat grösser als 127 ist
- BNE Verzweige, wenn das Null-Flag (zero flag) gelöscht ist (d. h. wenn das Resultat ungleich Null ist)
- BPL Verzweige, wenn das Negativ-Flag (negative flag) gelöscht ist, d. h. wenn das Resultat kleiner als 128 ist
- BRK Programmunterbruch (programmierter Interrupt)
- BVC Verzweige, wenn das Ueberlauf-Flag (overflow flag) gelöscht ist (Test im Status-Register)
- BVS Verzweige, wenn das Ueberlauf-Flag (overflow flag) gesetzt ist (Test im Status-Register)
- CLC Lösche das Uebertrags-Flag (carry flag) im Status-Register
- CLD Lösche das Dezimal-Flag (decimal flag) im Status-Register
- CLI Lösche das Unterbruchssperre-Flag (interrupt disable flag) im Status-Register, Unterbrüche werden nun beachtet
- CLV Lösche das Ueberlauf-Flag (overflow flag) im Status-Register
- CMP Vergleiche Speicher und Akkumulator, Flags werden verändert:

car. zer. ne.
- CPX Vergleiche Speicher und X-Register/

/	Register	<	Speicher	0	0	?
/	Register	=	Speicher	1	1	0
- CPY Vergleiche Speicher und Y-Register/

/	Register	>	Speicher	1	0	?
---	----------	---	----------	---	---	---
- DEC Speicherinhalt um 1 verkleinern (herunterzählen)
- DEX Inhalt des X-Registers um 1 verkleinern
- DEY Inhalt des Y-Registers um 1 verkleinern
- EOR Logische "Exklusiv-ODER"-Funktion mit Speicher und Akkumulator, Resultat im Akkumulator
- INC Speicherinhalt um 1 erhöhen (hinaufzählen)
- INX Inhalt des X-Registers um 1 erhöhen

- INY Inhalt des Y-Registers um 1 erhöhen
- JMP Sprungbefehl: Der Mikroprozessor fährt bei der angegebenen Adresse weiter
- JSR Unterprogrammprung: wie JMP, zusätzlich wird die Rücksprungadresse des Hauptprogrammes im Stack gespeichert SP=SP-2
- LDA Lade den Akkumulator mit einem Speicherinhalt, Speicher bleibt erhalten
- LDX Lade das X-Register mit einem Speicherinhalt, Speicher bleibt erhalten
- LDY Lade das Y-Register mit einem Speicherinhalt, Speicher bleibt erhalten
- LSR Schiebe Akkumulator bzw. Speicher um ein Bit (eine Binärstelle) nach rechts (d. h. Division durch 2)
- NOP Der Befehl NOP dient zum Füllen des Speichers, er bewirkt nichts
- ORA Logische ODER-Funktion mit Speicher und Akkumulator, Resultat im Akkumulator
- PHA Uebertrage den Akkumulator auf den Stack, Stack-Pointer (SP) wird um 1 erniedrigt (SP=SP-1), Akkumulator bleibt gleich
- PHP Uebertrage das Status-Register auf den Stack, Stack-Pointer (SP) wird um 1 erniedrigt (SP=SP-1) Status-Register bleibt gleich
- PLA Hole den Akkumulator vom Stack, Stack-Pointer wird um 1 erhöht (SP=SP+1)
- PLP Hole das Status-Register vom Stack, Stack-Pointer wird um 1 erhöht (SP=SP+1)
- ROL Rotiere den Akkumulator bzw. Speicher um ein Bit nach links, das Carry-Flag dient als Zwischenspeicher für das 1.+8. Bit
- ROR Rotiere den Akkumulator bzw. Speicher um ein Bit nach rechts, das Carry-Flag dient als Zwischenspeicher für das 1.+8. Bit
- RTI Rücksprung vom Programmunterbruch (Interrupt) zum normalen Programmablauf, Adressen und Status-Register vom Stack, SP=SP+3
- RTS Rücksprung vom Unterprogramm ins Hauptprogramm, Rücksprungadresse wird vom Stack geholt, SP=SP+2, siehe JSR
- SBC Subtrahiere den Speicher vom Akkumulator mit Uebertrag (carry), Resultat im Akkumulator
- SEC Setze das Uebertrags-Flag (carry flag) im Status-Register
- SED Setze das Dezimal-Flag (decimal flag) im Status-Register
- SEI Setze das Unterbruchssperre-Flag (interrupt disable flag) im Status-Register, Unterbrüche werden nicht beachtet
- STA Speichere den Akkumulator im Speicher, Akkumulator bleibt erhalten.
- STX Speichere das X-Register im Speicher, X-Register bleibt erh.
- STY Speichere das Y-Register im Speicher, Y-Register bleibt erh.
- TAX Uebertrage den Akkumulator ins X-Register, Akk. bleibt erh.
- TAY Uebertrage den Akkumulator ins Y-Register, Akk. bleibt erh.
- TSX Uebertrage den Stack-Pointer ins X-Register, SP bleibt erh.
- TXA Uebertrage das X-Register in den Akk., X-Regist.bleibt erh.
- TXS Uebertrage das X-Register in den SP, X-Register bleibt erh.
- TYA Uebertrage das Y-Register in den Akk., Y-Regist.bleibt erh.



Neuheiten

UTILITY - EPROM

Bereits im Heft 5/80 der SCC PET NEWS wurde das UTILITY EPROM in groben Zügen vorgestellt. Wir wollen Ihnen nun diese interessante Neuentwicklung ausführlich vorstellen.

Mit dem nachfolgend beschriebenen UTILITY EPROM können Sie in der Programmentwicklung und der Floppy Drive Benützung auf dem PET/CBM viel Zeit einsparen.

Das Utility Eprom wurde für den CBM 3032 Mikrocomputer mit Commodore Dual Floppy 2040 entwickelt. Es enthält Assembler Routinen, die einerseits das Zusammenfügen von BASIC-Programmen ab Floppy Drive ermöglichen und andererseits die Bedienung des Floppy Drive im Command-Modus wesentlich vereinfachen.

Zusammenfügen von Basic Programmen

In einem übersichtlich aufgebauten Programm werden Sie einzelne Programmteile (wie z. B. Sortierung, Ein- und Ausgabe auf den Floppy Drive etc.) als Subroutinen aufbauen.

Da diese Subroutinen, die Sie mittels GOSUB aufrufen, auch in anderen Programmen verwendet werden können, empfiehlt sich folgendes Vorgehen:

1. Verwenden Sie für diese Subroutinen Zeilennummern z. B. ab 50000. (GOSUB 50000, GOSUB 50100 etc.)
2. Beschreiben Sie die Funktion, die Input- und Output-Variablen jeder Subroutine.
3. Legen Sie diese Subroutinen auf einer separaten Programm-Diskette (Ihre Subroutinen Library) ab.
4. Bei der Entwicklung eines Hauptprogrammes rufen Sie die gewünschten Funktionen durch die entsprechenden GOSUB-Aufrufe ab. Verwenden Sie für das Hauptprogramm Zeilennummern unter 50000.

Mit dem UTILITY EPROM können Sie nun die einzelnen Programmteile bequem und zeitsparend (ohne erneutes Eintasten) wie folgt zusammenfügen:

Voraussetzungen:

- Die einzelnen Programmteile, die zu einem neuen Programm zusammengefügt werden sollen, sind auf dem Floppy Disk bereitgestellt
- Die Programmteile sind in der Reihenfolge in der sie zusammengefügt werden sollen aufsteigend nummeriert d. h. die Zeilennummern überschneiden sich nicht.

Utility Commands

Neben der Hauptfunktion "APPEND" (Zusammenfügen von BASIC Programmen) enthält das EPROM die Standard Utility Floppy Commands, die nachfolgend aufgeführt sind:

<u>Eingabe</u>	<u>Funktion</u>
>	Error Channel am Floppy auslesen und anzeigen
/d:name	Programm <u>name</u> von Drive d laden
d:name	Programm <u>name</u> von Drive d laden und starten
> command	Disk Command absetzen, z. Beispiel: IO Initialisieren Drive 0 CO:NEU=1:ALT Kopieren eines Files
>#0	Directory von Diskette in Drive 0 lesen und am Bildschirm anzeigen, ohne dass ein eventuell geladenes Programm überschrieben wird.

Hardcopy auf Printer

Durch Aufrufen einer entsprechenden Assembler Subroutine im EPROM erreichen Sie die schnelle Ausgabe einer Hardcopy des gesamten CBM-Bildschirmes auf den Printer.

Beispiel:

Programm MAIN sei auf Floppy Drive 1:

```
10 REM HAUPTPROGRAMM
20 GOSUB 100
30 IF A$="A" THEN GOSUB 1000:GOTO 50
40 IF A$="B" THEN GOSUB 2000
50 END
```

Die Subroutine GOSUB100 sei auf Floppy Drive 0:

```
100 GET A$:IF A$="" THEN 100
110 RETURN
```

Die Subroutine GOSUB1000 sei auf Floppy Drive 0:

```
1000 A=B*C+D
1010 PRINT A
1020 RETURN
```

Die Subroutine GOSUB2000 sei auf Floppy Drive 0:

```
2000 A=B+C*D
2010 PRINT A
2020 RETURN
```

Nach den Eingaben:

```
@ 1: MAIN
@@:GOSUB 100
@@:GOSUB 1000
@0:GOSUB 2000
```

ist folgendes Programm im PET/CBM Speicher:

```
10 REM HAUPTPROGRAMM
20 GOSUB 100
30 IF A$="A" THEN GOSUB 1000:GOTO 50
40 IF A$="B" THEN GOSUB 2000
50 END
100 GET A$:IF A$="" THEN 100
110 RETURN
1000 A=B*C+D
1010 PRINT A
1020 RETURN
2000 A=B+C*D
2010 PRINT A
2020 RETURN
```

Das Programm wird dem SAVE Command auf Disk gesichert
 Das UTILITY EPROM für den CBM 3032 und das CBM Dual Floppy 2040
 ist beim SCC für Fr. 158.-- erhältlich.

NEWTIM SV 2.1

NEWTIM SV 2.1 ist ein Monitor mit umfangreicher Floppy-Steuerung und Druckerausgabe für die CBM 3000-er Serie. Das Programm ist in einem EPROM untergebracht und belegt die Adressen \$9000 - \$9FFF, das in den letzten, leeren Sockel des PET eingesetzt wird.

Der Monitor wird mit SYS 36864 initialisiert und mit SYS 4 aufgerufen. Was kann man mit dem NEWTIM Programm erreichen?

1. Ausschalten des DOS-HELP (Monitor bleibt initialisiert)
2. Umschaltung Grossschrift-Graphik
3. Floppy-Kurzbefehle

Mit SYS 36870 wird der Monitor initialisiert und über SYS 4 aufgerufen. In dieser Betriebsart können folgende Operationen durchgeführt werden:

1. Mini-assembler (Synthax wie Disassembler)
2. Copieren von Speicherbefehlen
3. Copieren und Umrechnen von Speicherbefehlen
4. Disassembler
5. Verzögertes Abarbeiten des Programms unter Softwarekontrolle
6. Füllen des Speichers mit HEX-Zahlen
7. Byte-/Stringsuche
8. Byte und Stringaustausch
9. Suche unzulässiger OP-Code
10. Step-Modus
11. Copieren des PROM in das RAM
12. Programmieren des PROM
13. Setzen eines Druckerflag und Ausdruck der Zeile
14. Vergleich von Speicherbereichen
15. Arithmetische und logische Verknüpfungen
16. Umrechnung von Hexadezimal- in Dezimalzahlen
17. Umrechnung von Hexadezimal- in Dualzahlen
18. Umrechnung von Dezimalzahlen in Hexadezimalzahlen

Daneben enthält das Eprom weitere 14 Befehle auf die hier aber nicht eingegangen werden könnte.

Das Eprom ist mit deutscher Bedienungsanleitung beim SCC Luzern zum Preis von Fr. 475.-- lieferbar.

KIT 779

Der KIT 779 ist eine kleine Platine, mit der der Drucker Centronic 779 nun auch Kleinschrift und Sonderzeichen ausdruckt. Da gerade der Centronic 779 bei vielen PET-Liebhabern installiert ist, dürfte dieser KIT dem Wunsch nach universellem Einsatz des Druckers gerecht werden.

Zur Installation muss nach Entfernen der Rückwand lediglich ein IC aus seinem Sockel gezogen werden und an dessen Stelle eine kleine Platine eingesteckt werden; diese muss dann noch über zwei kurze Kabel mit den Anschlüssen eines anderen IC verbunden werden. Der Einbau ist absolut problemlos, allerdings empfiehlt es sich an Stelle der Klemmen an den Kabeln, diese abzuschneiden und anzulöten.

Der KIT 779 ist mit Einbau und Bedienungsanleitung beim SCC zum Preis von Fr. 226.-- erhältlich.

Leserbriefe

Wir haben diesmal bereits einen Teil der Leserbriefe in die Rubrik Tricks und Tips eingebaut und besprechen daher nur diejenigen, die ausserhalb dieses Themenkreises liegen.

Einer unserer Leser hat festgestellt, dass sich im Programm des Monats 1/80 ein kleiner Fehler eingeschlichen hat. In Zeile 800 muss es heissen REM CURSOR 6 mal nach unten. Dies hat allerdings keinen Einfluss auf das Programm selbst.

Herr Heinrich aus Weilheim hat den angekündigten WAIT-Befehl vermisst und senden uns daher gleich eine Reihe interessanter Anwendungen.

```
Wait 59411,8,8: warten auf Motor on tape #1 an
Wait 59411,8:  warten auf Motor on tape #1 aus
Wait 59408,16,16: warten auf gedrückte Taste on Tape #1
Wait 59408,16:  warten auf STOP-Taste on Tape #1
```

Zeitverzögerung

```
a) 1 Sekunde warten: POKE 143,Ø:Wait 143,2 6
b) 2 Sekunden warten:POKE 143,Ø:Wait 143,2 7
c) 4 " " :POKE 142,Ø:POKE 143,Ø:Wait 142,2 Ø
d) 8 " " :POKE 142,Ø:POKE 143,Ø:Wait 142,2 1
e) 16 " " :POKE 142,Ø:POKE 143,Ø:Wait 142,2 2
```

alles sind ca.-Werte;

Anmerkung für PET-Besitzer: andere Wait-Befehle bei:

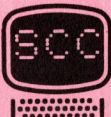
```
a): Wait 514
b): Wait 513
c): Wait 512
```

Warten auf beliebige Taste:

```
CBM: POKE 158,Ø: Wait 158,1: POKE 158,Ø
PET: POKE 525,Ø: Wait 525,1: POKE 625,Ø
```

In Heft 4/80 fragte ein Leser nach einem Programm, das den Wert eines INPUTS-Statements in eine DATA-Anweisung umwandelt. Nachfolgend finden Sie das dazugehörige Programm, die Werte müssen aber durch / getrennt sein und nicht wie bisher üblich durch Kommas.

```
2 PRINT"JALS LIMITER FUER DATA INPUT BITTE / BENUETZEN"
5 PRINT"UND NICHT - , - "
6 REM***** A. SIMOES *****
15 INPUT"ERSTE DATA ZEILENNUMMER ANGEBEN ";A
20 A=ABS(INT(A))
25 IFA<1000THEN15
30 INPUT"DATA EIGEBEN *|||";F$
35 IFF$="*"THEN90
40 A$=F$:F$=""
45 FORX=1TOLEN(A$):D$=MID$(A$,X,1)
50 IFD$="/"THEND$=","
55 F$=F$+D$:NEXTX
60 PRINT"|||"
65 PRINTA;"DATA";F$
70 PRINT"A=";A;" +10:GOTO30
75 IFPEEK(50500)=ØTHENPOKE525,2:POKE527,13
80 IFPEEK(50500)=ØTHENPOKE528,13:PRINT"|||":END
85 POKE158,2:POKE623,13:POKE624,13:PRINT"|||":END
90 REM***IHR PROGRAMM HIER EINSETZEN***
```



Empfohlener Verkaufspreis für den Detailhandel gültig ab 1. Oktober 1980 / Preis in Fr. (inkl. Wust)

MIKROCOMPUTER

PET 2001	Mikrocomputer (Standard) mit 8K Benützerspeicher, mit Rechnertastatur und eingebautem Kassettengerät	1'575.--
cbm 3008 N	Mikrocomputer mit 8K Benützerspeicher, Schreibmaschinentastatur für Gross- und Kleinschreibung und graphischen Zeichen, ohne Kassettengerät, Bildschirm 40 Zeichen/Zeile, grün, mit Basic-Interpreter 2.0	1'750.--
cbm 3016 N	wie 3008 N, jedoch mit 16K Benützerspeicher	2'250.--
cbm 3032 N	wie 3008 N, jedoch mit 32K Benützerspeicher	2'750.--
cbm 8032 B	Mikrocomputer mit 32K Benützerspeicher, Schreibmaschinentastatur für Gross- und Kleinschrift und graphischen Zeichen; ohne Kassettengerät, Bildschirm 80 Zeichen/Zeile, grün, mit Basic-Interpreter 4.0	3'650.--

SPEICHEREINHEITEN cbm Serie 3001 und 8001

cbm C2N	Kassettengerät (für alle Mikrocomputer)	225.--
cbm 3040	Floppy-Disk mit 2x176K Bytes Speicherkapazität, mit DOS 1.0 (für Serie 3001)	2'750.--
cbm 4040	Floppy-Disk wie 3040, jedoch mit DOS 2.0 (für Serie 8001)	2'750.--
cbm 8050	Floppy-Disk mit 2x512K Bytes Speicherkapazität, mit DOS 2.0 (für Serie 8001 wie auch 3001)	3'650.--

DRUCKER

cbm 3022	Matrixdrucker für Normalpapier, 7x6 Matrix, 80 Zeichen/Zeile, 150 Zeichen pro Sekunde; Endlosformularführung. Mit IEEE-488 Standard-Steckverbindung.	1'750.--
cbm 8024	Matrixdrucker für Normalpapier, Nadelkopf mit 7x7 Matrix, 132 Zeichen/Zeile, 160 Zeichen pro Sekunde (bidirektional), Endlosformularführung. Mit IEEE-488 Standard-Steckverbindung.	4'450.--
cbm 8024	wie cbm 8024, jedoch Nadelkopf mit 9x7 Matrix	4'900.--
Option 001		
cbm 8026	Kompaktdrucker (Typenrad), Standard-Schreibmaschinenwalze; 25 Zeichen pro Sekunde; ohne Tastatur. Mit IEEE-488 Steckverbindung.	3'300.--
cbm 8027	wie cbm 8026, jedoch mit Standard-Schreibmaschinentastatur	4'400.--

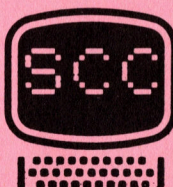
UMRÜSTSAETZE

cbm 1101	Umrüstsatz von Basic 1.0 (PET) auf Betriebssystem der cbm 3001 Basic 2.0	240.--
cbm 1102	Umrüstsatz von Basic 2.0 auf Basic 4.0 (Serie 8001)	240.--
cbm 1103	Umrüstsatz für Floppy cbm 3040 von DOS 1.0 auf DOS 2.0	240.--

SOFTWARE UND UTILITIES

cbm 1204	Wordcraft 80 (Textverarbeitung)	1'300.--
cbm 1211	Assembler I (Serie 3001)	320.--
cbm 1212	Monitor, Editor zu 1211 (S V2.0)	475.--
cbm 1231	Pascal TCL (Serie 3001)	350.--
cbm 1232	Visicalc	580.--

Aenderungen vorbehalten



Korrespondenz und Manuskripte bitte an

SCC PET NEWS
Seeburgstrasse 12
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.

Mitglieder des Schweizer Computer Club bezahlen Fr. 18.- pro Jahr, Nichtmitglieder Fr. 48.- pro Jahr (6 Ausgaben)

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. (0 64) 22 77 66.
Basel · BD-Electronic · Tel. (0 61) 35 36 37. Geiger Micro-computer · Tel. (0 61) 44 13 13. Leobag Computer AG · Tel. (0 61) 35 31 14. Radio TV Steiner AG (Filiale) · Tel. (0 61) 23 25 60. **Bern** · Computerland AG · Tel. (0 31) 24 25 54. Interelectronic · Tel. (0 31) 22 10 15. Thali AG (Filiale) · Tel. (0 31) 22 88 21. Radio TV Steiner AG · Tel. (0 31) 55 45 81. Radio TV Steiner AG (Filiale) · Tel. (0 31) 22 20 62. **Biel** · EIM Computer · Tel. (0 32) 23 15 88.
Brugg · Megos AG · Tel. (0 56) 41 34 17. **Buchs** · Büro Marxer · Tel. (0 85) 6 33 10. Obtron Elektronik · Tel. (0 85) 6 18 56. **Fontainemelon** · Urs Meyer Electronic ·

Tel. (0 38) 53 43 43. **Fribourg** · Sovitrel SA · Tel. (0 37) 22 78 37. **Genève** · Corylus · Tel. (0 22) 29 10 10. Gesmarco SA · Tel. (0 22) 36 51 36. Irco Electronic · Tel. (0 22) 20 33 06. Radio TV Steiner AG (Filiale) · Tel. (0 22) 28 52 22. **Hittnau** · Brunner Electronic · Tel. (0 1) 950 17 95.
Hitzkirch · Thali AG · Tel. (0 41) 85 28 28. **Interlaken** · H. U. Gurtner Datatechnik · Tel. (0 36) 22 10 21.
Langwiesen · Novotec R. Nagler · Tel. (0 53) 4 54 50.
Lausanne · Erhard Wipf SA (Filiale) · Tel. (0 21) 22 61 26.
Mafioy SA · Tel. (0 21) 22 00 44. **Luzern** · Dialog Computer Treuhand AG · Tel. (0 41) 31 45 45. Hunziker Elektronik · Tel. (0 41) 23 78 42. Schweizer Computer

Club · Tel. (0 41) 31 45 45. **Magliaso** · Marah SA · Tel. (0 91) 71 14 28. **Mellingen** · Instant-Soft AG · Tel. (0 56) 91 20 21. **Montreux** · Mafioy SA · Tel. (0 21) 62 12 12. **Niederrohrdorf** · Nöthiger Elektronik · Tel. (0 56) 96 28 96. **Schaffhausen** · Syntron Elektronik · Tel. (0 53) 5 33 77. **Sion** · Sphère Corporation · Tel. (0 27) 22 68 14. **St. Gallen** · Labor für Systemtechnik · Tel. (0 71) 28 39 05. Urs Meyer Electronic (Filiale) · Tel. (0 71) 23 41 33. Radio TV Steiner AG (Filiale) · Tel. (0 71) 23 10 33. **Thun** · HMB electronic · Tel. (0 33) 22 66 88. **Vevey** · Mafioy SA (Filiale) · Tel. (0 21) 52 99 52. **Volketswil** · Madelco Ltd. Zürich · Tel. (0 1) 945 04 10. **Wettingen** · Elbatex AG · Tel. (0 56) 26 56 41. **Winterthur** · Nowak AG · Tel. (0 52) 22 08 03. **Wohlen** · Tschachtli AG · Tel. (0 57) 6 68 66. **Zürich** · Logon AG · Tel. (0 1) 62 59 22. Hannes Keller AG · Tel. (0 1) 69 36 33. Erhard Wipf AG · Tel. (0 1) 221 21 00.