

# SVI

# JOURNAL

---

Die Zeitschrift des Spectra-Video-Club Austria

---

Datenfernübertragung

Akustikkoppler

Telebox



**HOE**

HOBBY-ELEKTRONIK-COMPUTERSHOW

21. BIS 24. NOVEMBER 1985

**Heft 11/85**

**MSX**

**S 15.-**

# Lieber SVI-Journal-Leser!

Kein SVI-Journal ohne Neuerung! Nachdem wir in der vorigen Ausgabe den Seitenspiegel kräftig verändert haben, wurden nun die internen Nachrichten und sonstige Rubriken etwas erneuert. Damit man sich besser im Heft zurecht finden kann, haben wir das Inhaltsverzeichnis auf eine Seite ausgedehnt. Dadurch rutschen alle Texte um eine Seite zurück. Als Ausgleich haben wir die Kleinanzeigen und Clubnachrichten noch vorne gezogen und bilden so eine "bunte" Informationsseite über die verschiedensten Themen.

Etwas überraschend ist für uns die Teilnahme an der diesjährigen "hobby electronic"-Messe gekommen. Da wir darauf angewiesen sind, daß uns ein gutgesinnter Aussteller "mitnimmt", entscheidet es sich oft erst im letzten Augenblick, ob der SVCA auf einer Messe vertreten ist oder nicht. Wir wollen nämlich das Clubbudget nicht für solche Dinge strapazieren. Zum Glück stellt jedoch die Firma Wehner wieder aus und wir bekommen einen Teil des Standes.

Da dies jedoch erst vor ein, zwei Wochen bekannt wurde, stehen wir (wie fast immer) recht unvorbereitet da. Es fehlt an Ideen, Personal und origineller Software. Wenn Sie daher Lust, Zeit oder geeignete Software haben, dann besteht für Sie noch die Chance, unserem Club während der Messe helfen und unterstützen zu dürfen! Näheres steht im Inneren des Heftes.

Nun ist es übrigens auch wieder an der Zeit, einen etwas "alten Hut" aufzuwärmen. Unsere Aktion mit dem Clubemblem hat keinen sehr durchschlagenden Erfolg gehabt. Zwei Vorschläge sind eingelangt. Für die bedanken wir uns. Eigentlich haben wir uns aber gedacht, daß etwas mehr in die Redaktion flattert. Sicher, es stimmt schon, daß wir letztlich nur ein Emblem verwenden können, aber ist das ein Grund, daß von haus aus nicht mehr geschickt werden? Sollten Sie also ein Zeichengenie sein oder auch nur Lust haben, ein Emblem zu entwerfen, dann tun Sie dies. Wir sind über jeden Vorschlag glücklich.

Apropos Mitarbeit beim Club! Da auch die Teamarbeit-Aktion auf etwas zu wenig Anklang

\*\*\*\*\*  
\* Programmiererecke: \*  
\* Programmierer suchen andere Program- \*  
\* mierer für Teamwork! \*  
\* Bin an Mühle- und Schachalgorithmen \*  
\* interessiert. Habe ebenso Interesse \*  
\* für allerlei andere Spielealgorithmen, \*  
\* auch Kartenspiele. Bevorzugte Program- \*  
\* miersprache: BASIC. \*  
\* Interessenten für Teamarbeiten werden \*  
\* ersucht, sich bei der Redaktion zu \*  
\* melden. Wir vermitteln Sie gerne wei- \*  
\* ter. \*  
\*\*\*\*\*

gestoßen ist, werden wir nun unsere Vorstel-  
lungen konkretisieren.

Ein Clubmitglied hat sich auf unseren Aufruf hin gemeldet. Er möchte mit anderen Mitglie-  
dern Mühle- und Schachalgorithmen program-  
mieren. Bevorzugte Programmiersprache:  
BASIC. Diese und andere Meldungen werden wir  
auf unserer Clubinfoseite veröffentlichen,  
damit vielleicht doch noch gemeinsame Arbei-  
ten zustandekommen. Oder ist unser Club so  
unfähig, daß er nicht einmal ein Mühlepro-  
gramm oder ein Autorennen programmieren  
kann?

Gesucht werden außerdem noch MSX-Mitarbeiter  
für das SVI-Journal. Da die Zahl der MSX-  
Leser in unserem Leserkreis ständig wächst,  
wollen wir mehr und bessere Information über  
MSX geben.

Ihr SVI-Journal-Chefredakteur Gerhard Fally!

---

## KLEINANZEIGEN:

---

Verkaufe Steuerungscomputer, 4 Bit, Maschin-  
codeähnlicher Befehlssatz, sehr leistungs-  
fähig, Top-Zustand, BUSH 2090

VB 1500 Schilling

Verkaufe Drucker, Seikosha GP 500,  
1 Monat alt, kaum gebraucht, um ÖS 3000,-  
Tel 34 29 945 (bitte öfters probieren)

Spectravideo SVI-328 MK II mit Datasette und  
Joystick günstig abzugeben. Tel.: 25 26 94

*Wieder-besoldete es Platz + Großfeld bedarf 2. Mo. -  
Dank Richard - 68 / Stefan TICSCH*

\*\*\*\*\*  
\* Die nächsten Clubtermine: \*  
\* Do, 21. 11. Hobby-electronic Anfang \*  
\* So, 24. 11. Hobby-electronic Ende \*  
\* Sa, 30. November 1985, ab 19 Uhr \*  
\* Mi, 4. Dezember 1985, ab 19 Uhr \*  
\* Di, 10. Dezember 1985, ab 19 Uhr \*  
\* Mi, 18. Dezember 1985, ab 19 Uhr \*  
\* \*  
\* Clubabende wie immer im Clublokal im \*  
\* Computer-Studio, \*  
\* 1040 Wien, Paniglgasse 18-20 \*  
\* Nichtmitglieder sind willkommen \*  
\* Ende jeweils ca. 22 Uhr! \*  
\* \*  
\* Aktivitäten an den Clubabenden: \*  
\* Arbeiten an Spectravideo-Systemen, \*  
\* Informationsaustausch zwischen Club- \*  
\* mitgliedern, Gelegenheit zum Aus- \*  
\* drucken von Programm listings. \*  
\* \*  
\* Telefonische Auskünfte über den Club \*  
\* und seine Aktivitäten erhalten Sie \*  
\* unter der Telefonnummer 65 88 93. \*  
\* \*  
\*\*\*\*\*

<b>Programmiersprachen &amp; Serien:</b>			
Spectravideo-BASIC	Seite 4	Telebox	Seite 11
Die Erklärung der BASIC-Befehle des MSX- und SVI-BASIC, insbesondere der Befehle, die im Handbuch unvollständig oder falsch beschrieben sind, diesmal ein Zusatz zum BLOAD.		Die von "Radio-Austria" neugegründete Telebox bietet viele Anwendungsmöglichkeiten. Wir berichten exklusiv darüber.	
Turbo-PASCAL, von Anfang an	Seite 7	Programmecke	Seite 19
Der vollständige Lehrgang für eine der weitverbreitetsten und besten Sprachen für Mikrocomputer.		Programme von Clubmitgliedern und "normal" käufliche Programme werden beschrieben und einige Tips gegeben. Diesmal ist es das Assemblermodul von A. Holy und G.Wolfbauer.	
Das CP/M-Betriebssystem am SVI-328	Seite 5	Die Gesetze von Murphy	Seite 18
Eine umfassende Einführung und detaillierte Beschreibung des CP/M 80 am Spectravideo mit allen Eigenheiten und Pluspunkten.		Eine ironische Darstellung bestimmter Sachverhalte, die in der Physik und Elektrotechnik häufig auftreten, für die man aber normalerweise keine Erklärung finden kann.	
<b>Programmlistings:</b>		Leitartikel	Seite 2
Schneller Sinus/Kosinus	Seite 13	Impressum	Seite 20
Für Graphikanwendungen oder sonstige schnelle trigonometrische Rechnungen ist der Microsoft-Sinus zu langsam. Was liegt daher näher, als einen schnellen, aber nur auf drei Stellen runden Sinusgenerator zu entwerfen?		Kleinanzeigen	Seite 2
<b>Hintergrundinformation, Rubriken:</b>		*****	
Probleme bei Datenübertragung	Seite 9	* ACHTUNG !!!	
Jeder, der Daten über die RS-232 C-Schnittstelle übertragen möchte, steht vor einigen Schwierigkeiten. Dieser Bericht gibt eine leicht verständliche Einführung in die Datenkommunikation zwischen Computern.		* NEUE MITGLIEDER!	
Akkustikkoppler SONIC-300	Seite 12	* Wenn Sie jetzt dem SVCA beitreten, gilt der Mitgliedsbeitrag bis Ende 1986.	
Ein besonders günstiger und sehr leistungsfähiger und robuster Akkustikkoppler wurde von uns "in die Mangel genommen".		*****	

\*\*\*\*\*  
 \* **Vorschau auf die "he 85"** \*  
 \* \*  
 \*\*\*\*\*

In diesem Jahr findet die Hobby Electronic-Messe zum ersten Mal im Messegelände statt. Die Messe okkupiert diesmal hauptsächlich die Hallen 12, 14, 15 und 16. Wir stellen in Halle 14, Stand 245 aus.

Unser Stand in Halle 14 ist zwar nicht sehr groß, aber das heißt noch lange nicht, daß wir dort deshalb wenig machen. Gesucht sind wieder Personal, Programme und andere originelle Ideen. Sollten Sie also einen singenden Computer zuhause stehen haben, dann bringen Sie einmal diesen Hardwarezusatz mit. Gleiches gilt für andere nette Programme oder Hardware.

Wir beabsichtigen, wieder zwei oder drei Spectravideo-Computer aufzustellen, wahrscheinlich einen SVI-728, einen 738er und

einen SVI-328. Dazu wird sich das Angebot der Firma Wehsner ergänzen, die vor allem den Spectravideo als Personalcomputer vorstellen wird. Wichtig sind wie immer schriftliche Unterlagen über Club und Journal. Diese müssen natürlich auch werbewirksam an den Mann gebracht werden. Deshalb bitten wir alle, die Lust haben, einen oder zwei Nachmittage zu opfern, sich bei uns zu melden. Jedoch sind für uns auch Leute, die am Vormittag ihren "Dienst" versehen können, besonders wichtig.

Wir hoffen natürlich, daß uns möglichst viele Clubmitglieder auf der Messe besuchen. Wenn Sie daher auf der "he" sind, dann schauen Sie sich unseren Stand an und tun Sie dann Ihre Meinung kund. Wir können aus Kritiken (Lob eingeschlossen!) nur lernen.

Die Messe beginnt am Donnerstag, dem 21. November, und endet am Sonntag, dem 24. November. Sie ist von 9 bis 18 Uhr durchgehend geöffnet. Wir werden in unserer nächsten Ausgabe ausführlich über die HE berichten.

## Spectravideo-BASIC

Bevor wir mit unserem Dateiprogramm weitermachen, das wir schon in der vorigen Folge begonnen haben, wollen wir einen kurzen Einschub unterbringen. Wir haben dem Computer nämlich beim BLOAD-Befehl weniger zuge-  
traut, als er wirklich kann.

Konkret: Wir hatten damals bemängelt, daß beim Laden von Maschincode-Programmen (oder ähnlichen Speicherausztügen) eine Option nicht funktioniert. Im Handbuch steht nämlich, daß man eine absolute Adresse angeben kann, ab der dann das File in den Speicher geladen wird. Inzwischen haben wir uns weitergebildet (sprich das ROM-Listing studiert), und sind zu dem weisen Schluß gekommen, daß die Adresse nicht absolut sondern als Offset zu gebrauchen sei. Und wirklich, wenn man das Ganze als Offset ansieht, zeigt sich der Computer willig und verschiebt das Programm ganz richtig. Dazu einige Beispiele:

Voraussetzung für unsere Beispiele ist, daß wir ein Programm von C000H bis C200H abgespeichert haben. Wenn wir das Programm nun ganz normal wieder in den gleichen Speicherbereich laden wollen, müssen wir wie gehabt BLOAD "TEST" eingeben. Wenn wir das Programm gleichzeitig starten wollen, können wir BLOAD "TEST",R eingeben. Soweit ist alles bekannt. Doch nun kommt die große Neuerung:

Um das besagte Programm, aus welchen Gründen auch immer, nicht nach C000H sondern nach B000H zu bringen, tippen wir die Anweisung BLOAD "TEST",-&H1000 und haben nach der Ausführung das Programm um 4096 Bytes nach unten verschoben.

Wie verhält sich nun der Offset mit dem Zusatz ",R"?

Antwort: Durchaus verträglich, wenn man das ",R" nur richtig positioniert. Durch das Kommando BLOAD "TEST",R,-&H1000 kann man das Programm zuerst verschieben und dann starten.

Soweit zum BLOAD-Befehl. Wozu kann es nun gut sein, das mit BSAVE abgespeicherte FILE im Speicher zu verschieben?

Dafür gibt es sogar einige Gründe. Zum Ersten existieren Assembler-Versionen, die den Befehl CODE zulassen. In Verbindung mit dem obligaten ORG kann man Programme auf einem bestimmten Speicherbereich assemblieren und für einen ganz anderen Speicherbereich vorsehen. Diese Files haben dann alle Sprungadressen auf den Bereich, für den sie eingestellt sind, gerichtet. Um nun das Programm in den richtigen Bereich zu lotsen, verschiebt man es gleich beim Laden mit dem Offset.

Zum Zweiten kann man mit dem BLOAD-Befehl nicht nur MC-Programme verschieben, sondern auch Datenfelder, die mit BSAVE abgespeichert wurden. Und Daten verschiebt man bekanntlich oft und gerne. Um nun nicht immer ein Programm schreiben zu müssen, das den Speicher "umdreht", kann man BLOAD mit Offset verwenden.

In der nächsten Ausgabe werden wir wieder auf Aktuelles eingehen und unser Programm weiterschreiben.

## Die Grundsätze des ÖJRK

Das Österreichische Jugendrotkreuz (ÖJRK) ist ein Teil des Österreichischen Roten Kreuzes; es steht als eine Erziehungsgemeinschaft von Lehrern, Eltern und Schülern in einem Nahverhältnis zur Schule. Ebenso wie das Rote Kreuz ist es eine unpolitische und überparteiliche Vereinigung, deren Aufgabe es ist, junge Menschen zu humanitärer Gesinnung und zu mitmenschlichem Verhalten hinzuführen, sowie konkrete Hilfe zu leisten. Gleichzeitig erfüllt das Jugendrotkreuz die Aufgabe, das Gedankengut des Roten Kreuzes sowie Geist und Inhalt der Genfer Rotkreuz-Abkommen und der Zusatzprotokolle zu verbreiten.

### Die pädagogischen Ziele des ÖJRK

Junge Menschen sollen bereits in der Schulzeit unter Mithilfe ihrer Lehrer, die vielfach auch Funktionen im ÖJRK haben, lernen, selbst initiativ zu werden, wenn es gilt, Not zu lindern und Hilfe zu leisten. Damit wird ein nicht zu unterschätzender Beitrag für die Erziehung zur Eigenverantwortlichkeit der Schüler geleistet, wie sie auch der Erlaß "Politische Bildung in der Schule" Zl. 33.464/6-19a/1978 fordert. Schüler sollen bewußt lernen, nicht nur zu fragen, was ihre

Mitmenschen für sie zu tun bereit sind, sondern auch, was sie für die Mitmenschen tun können.

Das Österreichische Jugendrotkreuz erschließt damit Lernfelder, die von der Schule allein aufgrund ihrer strukturellen Situation nicht gänzlich abgedeckt werden können.

Die vom Jugendrotkreuz entwickelten, Schülern und Lehrern zur Verfügung gestellten Unterrichtsbehelfe zur Gesundheitserziehung, zur Umwelterziehung, zur Leseerziehung und zur Verkehrserziehung sind, entsprechend den Bildungs- und Erziehungsaufgaben sowie den didaktischen Grundsätzen nicht nur ein wichtiger Beitrag für die schulische Erziehungsarbeit, sondern sie stellen eine nicht zu unterschätzende Leistung des Jugendrotkreuzes für die österreichische Schule dar.

Schulbehörden, Lehrer und Eltern haben dies frühzeitig erkannt. Sie fördern daher das ÖJRK im Sinne des Erziehungsauftrages der österreichischen Schule nach besten Kräften.

Aus: Das ÖSTERREICHISCHE JUGENDROTKREUZ - seine Mitwirkung an den Bildungs- und Erziehungsaufgaben der Schule

# Das CP/M-Betriebssystem am SVI-328

Aus Platzgründen konnte der letzte Beitrag leider nicht vollständig abgedruckt werden, weshalb wir hier einfach ohne viel Einleitung fortsetzen. Um den Anschluss etwas zu erleichtern ist ein nochmaliges Überfliegen der letzten Textblöcke vor der "Abbruchstelle" im letzten Beitrag empfehlenswert.

Solange man also den vom Linker erzeugten, jedoch fehlerhaften "COM-File" nicht startet, bleibt dieser Umstand folgenlos.

Eine fehlerhafte Compilierung kann daran erkannt werden, daß nach Fertigstellung des "COM"-Files die Fehlermeldungen des "M80" noch auf dem Bildschirm stehen.

Der "M80" fügt zum Beispiel bei nicht definierten Labels von selbst immer 0 ein, was in fast allen Fällen einen Warmstart auslöst.

Schlimmer wäre es jedoch, würden für Hardware vorgesehene, fälschlicherweise aber nicht definierte Steuerungs-Bits im Maschinencode auf diese Art zu Fall gebracht...

Der automatische Start des Linkers trotz möglicher Fehler in der Quelldatei ist also ein durchaus kalkulierbares Risiko. Optische Kontrolle ist hier natürlich vorausgesetzt.

Weitaus gefährlicher wird es aber, veranlaßt man den Linker unbedacht durch den Switch "/G" in der "Batch"-Datei, das jeweilige, jedoch zufällig fehlerhafte Programm sofort im Speicher zu starten.

In den meisten Fällen verabschiedet sich der Computer nur still und leise oder führt einen "geordneten" Warmstart durch.

Bei Hardware-Erweiterungen, Steuerungen oder Ähnliches wäre ein Absturz etwas ungünstiger. Es genügt aber auch schon, wenn ein Programm aufgrund irgendwelcher Fehler unkontrolliert auf die Diskette schreibt.

Bei der Wahl solcher "bedingungsloser" Befehle in automatisch ablaufenden Prozessen wie "SUBMIT", wo einzelne Module fehlerhaftes produzieren können und, was noch viel schlimmer ist, auch akzeptieren, sollte man die Tragweite der möglichen Folgen genau abgrenzen.

Akzeptieren deshalb, da es ja auch theoretisch möglich wäre, daß der "M80" die Produktion der "REL"-Datei bei fehlerhaftem Quellcode einstellt und diese löscht.

Bei der Erstellung eines Stapelauftrages ist somit ein genaues Konzept unter Prüfung und Berücksichtigung aller Eigenschaften der jeweiligen Programme und möglichen unerwünschten Ereignissen unbedingt empfehlenswert.

Die zuletzt beschriebene Anwendung soll neben seiner Beispielwirkung auch aufzeigen, worauf bei Vorbereitung eines Stapelauftrages geachtet werden muß.

Vor der Untersuchung von "XSUB" werden wir uns aber noch einige weitere, auch besondere Eigenschaften von "SUBMIT" anschauen und einen bedeutenden Fehler korrigieren.

## Besondere Eigenschaften von "SUBMIT"

Die Umwandlung einer Stapelverarbeitungsdatei durch "SUBMIT" wird abgebrochen, wenn eine Leerzeile in der "SUB"-Datei vorgefunden wurde. Die Folge ist eine unvollständige "\$\$\$SUB"-Datei, die aber trotzdem, wenn auch etwas kleiner, abgearbeitet wird.

Das Abbrechen der Stapelverarbeitung selbst mit gleichzeitigem Löschen der "\$\$\$SUB"-Datei erfolgt, wenn der CCP einen Befehl ein Umbenennen oder Löschen kann oder ein aufzurufendes Programm sich nicht auf der Diskette befindet.

Das trifft aber nur auf die CCP-Anweisungen und das Vorhandensein von aufzurufenden Programmen zu. Konnte beispielsweise der CCP ein Umbenennen oder Löschen aufgrund der fehlenden Originaldatei nicht korrekt ausführen, wird ungeachtet dessen ganz normal weitergearbeitet.

An dieser Stelle sei auch vermerkt, daß bei fehlerhaften Anweisungen in "\$\$\$SUB" an das durch Stapelverarbeitung aufgeführte Programm "PIP" eine Fortsetzung des "Batch"-Auftrages nicht mehr möglich ist.

Das betrifft aber nur die "Befehlssprache" von "PIP", nicht aber die Folgen von unauflösbaren Dateien auf der Diskette. Im Fehlerfall, wenn "PIP" zum Beispiel die Meldung "INVALID FORMAT" auf den Bildschirm bringt, wird noch vor dem Verlassen von "PIP" die "\$\$\$SUB"-Datei gelöscht.

"STAT" und fast alle anderen CP/M-Programme hingegen erlauben auch fehlerhafte Anweisungen ohne im Nachhinein als "Strafe" "\$\$\$SUB" zu löschen und ermöglichen somit die "ordnungsgemäße" Fortsetzung der Stapelverarbeitung.

Wie schon früher erwähnt verursacht auch der Druck auf eine Taste, bevorzugt wird "DELETE", ebenfalls eine vorzeitige Unterbrechung des Stapelverarbeitungsprozesses inklusive Löschen von "\$\$\$SUB".

Angeschnitten sei hier auch noch die Möglichkeit des Löschens von "\$\$\$SUB" durch das jeweils laufende Anwenderprogramm.

Erfordert eine Befehlszeile das eigentlich für Variable reservierte Zeichen "\$", muß das Zeichen verdoppelt werden. Das schaut dann zum Beispiel so aus:

Ursprüngliche Anweisung an den CCP:

```
ERA *.$$$ und STAT *.COM $R/O
```

Anweisung in der "SUB"-Datei:

```
ERA *.$$$$$$ und STAT *.COM $$R/O
```

Das Ausgeben von Kommentaren oder Aufforderungen am Bildschirm für den Benutzer während der Abarbeitung eines Stapelauftrages war schon im Gespräch.

Als Kommentar werden die Zeilen in der "Batch"-Datei verstanden, die keine Reaktion des CCP-Befehlsinterpreters erwarten.

Die Sonderzeichen ". : ; < = > \_" einfach vor die auf den Bildschirm auszugebende Kommentar-Textzeile gesetzt, erlauben Meldungen auf dem Bildschirm wie folgt.

:BITTE KEINE TASTEN BETÄTIGEN

Da "SUBMIT" selbst keine Möglichkeit bietet, auf die Erledigung einer bestimmten Sache durch den Benutzer zu warten, bedient man sich eines kleinen "Kunstgriffs".

In Kombination dieses Tricks mit Kommentarzeilen kann man den Benutzer zu bestimmten Aktionen auffordern, und nach Abschluß auf eine Bestätigung warten.

Zu diesem Zweck wird einfach "irgendein" Programm aufgerufen, welches nach Betätigung einer bestimmten Taste sofort einen geordneten Warmstart ausführt.

Dieser Umstand erlaubt dem CCP das Abarbeiten der nächsten Befehlszeilen, mit dem hoffentlich richtig erfüllten Wunsch der "Batch"-Datei durch den Benutzer. Er kann sich ja nun damit beliebig viel Zeit lassen und in Ruhe die bestimmten Aufforderungen erledigen.

Für diesen Vorgang bietet sich das auf der Systemdiskette befindliche Kopierprogramm "PIP.COM" an.

In der Praxis könnte das so aussehen.

```
:BITTE DISKETTE "ABC" IN LAUFWERK B EINLEGEN
:WENN FERTIG, EINMAL "RETURN"-TASTE DRÜCKEN
PIP
```

Erscheint nach Aufruf von "PIP" der Bereitschaftsstern, wird erst dann mit der weiteren Ausführung des Stapelauftrages fortgesetzt, bis die "RETURN"-Taste gedrückt wurde und der CCP zu seinem hiefür notwendigen Warmstart kommt.

Steuerzeichen in "SUB"-Dateien

Einige Programme benötigen in Ihrer Befehlssprache Steuerzeichen wie "^L", "^X" oder "^Z" und ähnliches mehr.

Wird mit "SUBMIT" alleine, also ohne "XSUB" gearbeitet, ergibt sich der Bedarf dieser Steuerzeichen kaum. Schließlich können mit "SUBMIT" ja nur Programme mit eventuellen Parametern von der Diskette aufgerufen werden, wobei sich die Notwendigkeit von Steuerzeichen in der Regel erübrigt.

In Kombination mit "SUBMIT" und "XSUB" ist es nun aber möglich, mit dem gerade aufgerufenen Programm in dessen "Sprache" zu sprechen. Und diese Sprachen erfordern häufig solche Steuerzeichen oder Control-Character.

Diese Zeichen, als Text in die Befehlszeile eingegeben, werden, so ist es zumindest vorgesehen, durch "SUBMIT" bei der Erstellung der "\$\$\$SUB"-Datei auf das entsprechende Ein-Byte-Steuerzeichen komprimiert.

Und genau dieser Auftrag wird durch die "SUBMIT"-Fehlermeldung "INVALID CONTROL CHARACTER" beantwortet und die Erstellung von "\$\$\$SUB" somit verhindert.

Glücklicherweise handelt es sich hierbei nur um einen Subtraktionsfehler in "SUBMIT" über

dessen Vorhandensein man sich wie folgt rasch Klarheit verschaffen kann.

```
A>DDT SUBMIT.COM oder A>ZSID SUBMIT.COM
DDT VERS 2.2          ZSID VERS 1.4
NEXT PC              NEXT PC END
0600 0100            0600 0100 B5FF
-L441                #L441
 03FE CALL 0309      03FE CALL 0309
0441 SUI 61          0441 SUB 61
0443 STA 0E7D        0443 LD (0E7D),A
0446 MOV C,A         0446 LD C,A
0447 MVI A,19        0447 LD A,19
0449 CMP C           0449 CP C
044A JNC 0456        044A JP NC,0456
044D LXI B,019D      044D LD BC,019D
0450 CALL 02A7       0450 CALL 02A7
. . .                . . .
```

In der Zeile 03FE holt sich "SUBMIT" unser Steuerzeichen in Textform aus dem Disk-Buffer. Das Zeichen "^" selbst hat ja bereits seine Funktion erfüllt. Es führte den Programmablauf zur jetzigen Routine.

In der nachfolgenden Zeile sollte durch Abzug des ASCII-Werts von "A" (41H) der eigentliche Ein-Byte-Wert -1 des jeweiligen Steuerzeichens errechnet werden.

Haben Sie nun eine fehlerhafte Version von "SUBMIT" werden Sie feststellen, daß in Zeile 0441 der Wert 61H, das entspricht dem ASCII-Zeichen "a", abgezogen wird.

Durch diesen Fehler wird der im Akku stehende, schon früher zu einem Großbuchstaben gewandelte ASCII-Wert nicht zu einem Wert zwischen 0 und 19H. Das entspricht jenem Wert +1, als wäre das Steuerzeichen direkt über die Tastatur eingegeben worden.

Zu guter Letzt wird noch die Zulässigkeit des Steuerzeichens überprüft. Liegt ein verbotener Wert vor, wird in Zeile 044D die Ausgabe der oben beschriebenen Fehlermeldung vorbereitet.

Abhilfe kann diesem Fehler wie folgt geschaffen werden.

```
Mit dem "DDT":          Mit dem "ZSID":
-S442                    #S442
0442 61 41                0442 61 41
0443 32 .                  0443 32 .
-^C                       #^C
```

Nach erfolgreichem Warmstart wird mit dem Befehl

```
A>SAVE 5 SUBMIT.COM
```

die richtiggestellte Version von "SUBMIT" auf der Diskette verewigt.

Ab nun ist es somit möglich vor allem unter "XSUB" Steuerzeichen von ^A bis ^Z an das jeweils aufgerufene Programm zu übergeben.

Und wer jetzt glaubt, er kann in der "SUB"-Datei den Drucker mit "^P" ein- bzw. ausschalten, irrt. Das Übergeben von Steuerzeichen an den CCP-Monitor selbst ist über "SUBMIT" nicht möglich. Das funktioniert leider nur im "Direktmodus".

Der nächste Beitrag schneidet noch die Verkettung von "SUB"-Dateien an und führt uns anschließend zu "XSUB".

# Turbo-PASCAL, von Anfang an Folge 9

In dieser Folge werde ich auf die sogenannten strukturierten Daten in PASCAL eingehen: TYPE und RECORD. Ein bißchen Praxis gibt's natürlich auch wieder.

Bis jetzt haben wir sogenannte "einfache" Datentypen wie REAL, INTEGER, CHAR und BOOLEAN kennengelernt. Daneben haben wir schon vom Typ STRING gehört, den man sich als Array von CHAR-Variablen denken kann, und von ganz normalen ARRAYS. Diese zwei Datentypen sind bereits die zwei einfachsten "zusammengesetzten" oder "strukturierten" Datentypen.

PASCAL bietet aber noch weitere Möglichkeiten, Daten zu strukturieren. Man kann mit dem Befehl TYPE "neue" Variablentypen definieren, die aus den schon bekannten bestehen. Ein Beispiel:

In einem Programm soll nach einem Datum gefragt werden. Es soll in den Variablen "jahr", "monat" und "tag" abgelegt werden. Bis jetzt hätten wir diese Variablen als INTEGER definiert, mit TYPE ergeben sich aber ganz neue Möglichkeiten. Sehen wir uns folgende Befehlsgruppe an:

```
TYPE year = 1970..2100;
      month = 1..12;
      day = 1..31;
```

```
VAR jahr:year;
     monat:month;
     tag:day;
```

Mit dem TYPE-Befehl, der zu Beginn eines Programmes oder einer Prozedur steht, werden drei neue Variablentypen definiert. Zum ersten einmal der Variablentyp "year", der ganze Zahlen von 1970 bis 2100 beinhalten kann. Dann der Typ "month", der von 1 bis 12 geht, und schließlich der Typ "day", in dem man Zahlen von 1 bis 31 ablegen kann. Nachdem der Variablentyp mit der Anweisung TYPE vereinbart wurde, kann man ihn ganz normal wie INTEGER oder REAL in einem VAR-Statement verwenden. In unserem Beispiel werden so die Variablen "jahr", "monat" und "tag" vereinbart. Will man z.B. jetzt der Variablen "monat" den Wert 100 zuweisen, wird TURBO einen Fehler melden, da der Wert 100 nicht in der Anweisung enthalten ist.

Der TYPE-Befehl ist aber noch viel leistungsfähiger. Denkbar ist auch folgende Definition des Variablentyps "month":

```
TYPE month=(jan,feb,mar,apr,jun,jul,aug,
            sep,oct,nov,dec);
```

Nach diesem Statement können Variablen der Type "month" die Werte in der Klammer annehmen, aber keine anderen. Man kann sich die Standardtypen BYTE und BOOLEAN auch folgendermaßen definiert denken:

```
TYPE BYTE=0..255;
      BOOLEAN=(FALSE,TRUE);
```

Die großen Buchstaben kann man z.B. so als Variablentyp "upper" definieren:

```
TYPE upper='A'..'Z';
```

Ein TYPE-Statement erzeugt zugleich eine

Ordnung innerhalb der möglichen Werte des neuen Variablentyps. Es gibt drei Standardfunktionen in PASCAL, um diese Ordnung abzufragen. Um das zu verdeutlichen, definieren wir einen Variablentyp, in dem die vier verschiedenen Farben einer Spielkarte enthalten sind:

```
TYPE karte=(herz,karo,pik,treff);
```

Diese vier Werte kann man jetzt mit normalen Vergleichsoperatoren vergleichen. Es gilt:

```
herz < karo < pik < treff
```

Zusätzlich kann man den Nachfolger eines Elements durch die Anweisung "nachfolger:=SUCC(wert);" erhalten.

SUCC(karo) ergibt demnach "pik" und SUCC(pik) ist "treff". Mit der Anweisung:

```
vorgaenger:=PRED(wert);
```

erhält man den Vorgänger eines Werts. PRED(treff) ist also "pik" und PRED(karo) ist "herz". Um die Rangnummer eines Elements zu erhalten, verwendet man folgendes Statement: nummer:=ORD(wert);.

ORD(karo) ist also 2 und ORD(treff) ist 4. Diese drei Anweisungen sind auch auf die Standardtypen INTEGER, CHAR und BYTE anwendbar, nicht jedoch auf REAL. SUCC(1000) ist demnach 1001 und PRED('b') ist "a". Es ist übrigens auch nicht möglich, aus REAL-Zahlen neue Variablentypen zu konstruieren. Die einzige Ausnahme ist folgende:

```
TYPE feld: ARRAY (1..10.) OF REAL;
```

Nur in Arrays dürfen REAL-Zahlen in einer TYPE-Anweisung vorhanden sein. Die TYPE-Anweisung kann uns also viel Tipparbeit ersparen. Benötigt man zum Beispiel in einem PASCAL-Programm mehrere Arrays mit der gleichen Struktur, dann ist es sinnvoll, dieses Array mit TYPE einmal zu definieren, und dann diesen neuen Variablentyp zu verwenden, um die einzelnen Variablen zu definieren.

Neben den Funktionen ORD, PRED und SUCC gibt es in PASCAL auch noch eine Möglichkeit, verschiedene Typen ineinander umzuwandeln. Ausgenommen von dieser Option ist jedoch der Typ REAL.

Die Funktion ORD benutzen wir, um die "Rangnummer" eines Elementes zu bekommen, doch wie kann man von der Nummer aus das Element selbst erhalten?. Nehmen wir z.B. folgende TYPE-Vereinbarung:

```
TYPE tag=(mo,di,mi,do,fr,sa,so);
      lower='a'..'z';
```

Wenn wir wissen wollen, wie der dritte Tag der Woche heißt, tippen wir ganz einfach:

```
WRITELN(tag(3));
```

und erhalten "mi". Genauso erfahren wir mit dem Statement buchstabe:=lower(14), daß der 14. Kleinbuchstabe "o" ist. Diese "Umkehrung" der Funktion ORD gibt es aber nur in TURBO, nicht jedoch im Standard-PASCAL.

Nun wissen wir, wie man neue Variablentypen definieren kann. Natürlich gibt es noch eine Menge Feinheiten, und der Phantasie des Programmierers sind keine Grenzen gesetzt, immer neue Variablentypen zu kreieren, aber das würde den Rahmen dieses Kurses sprengen.

So weit, so gut. Aber was ist, wenn ich nun in einem Programm viele verschiedene Variablen definiert habe, und ich will sie alle in einem File abspeichern? Ein Beispiel: Ein Programm soll die Funktion eines kleinen Notizblocks übernehmen. Dazu definieren wir folgende Variablen:

```
TYPE year:1980..2000;
    month:1..12;
    day:1..31;
    inhalt:STRING(.30.);

VAR jahr:ARRAY(.1..20.) OF year;
    monat:ARRAY(.1..20.) OF month;
    tag:ARRAY(.1..20.) OF day;
    notiz:ARRAY(.1..20.) OF inhalt;
```

Mit diesen Variablen kann man also 20 Daten mit den dazugehörigen Texten erfassen, aber wie soll man sie abspeichern? Man könnte natürlich vier getrennte Files eröffnen, eines für die Jahre, eines für die Monate usw. Oder man könnte alle Zahlen in Strings verwandeln und diese dann zusammenfassen und gemeinsam abspeichern. Beide Möglichkeiten erfreuen aber nicht das Herz eines geübten PASCAL-Programmierers. Er wird zu sogenannten RECORDs greifen.

Was ist aber nun ein RECORD? Um das zu klären, sehen wir uns das Programm "notizblock" an, das genau den Notizblock bildet, den wir oben beschrieben haben. Das Programm ist aber unvollständig, es fehlen die Prozeduren "input\_termin" und "suche\_termin", die in der nächsten Folge abgedruckt werden.

Ein RECORD ist auf Deutsch ein "Datenverbund", d.h. mehrere Variablen werden unter einem Dach zusammengefaßt und bekommen einen gemeinsamen Namen. Bildlich gesehen schnüren wir ein Paket Variablen zusammen, geben ihm einen Namen, und unter diesem Namen kann es dann bearbeitet werden. In unserem Fall ist es das Paket "notiz", das aus drei Ziffern (tag,monat,jahr) und einem String besteht (inhalt). Die Anweisung zum Schnüren eines Paketes mit der Bezeichnung "name" lautet folgendermaßen:

```
name=RECORD
    ...;
    einzelne Variablen;
    ...;
END;
```

Um anzudeuten, daß hier eine "Datenstruktur" geschaffen wurde, werden die einzelnen Variablen eingerückt geschrieben, wie die Anweisungen eines Programmes. Tatsächlich beginnen hier bereits die Grenzen zwischen Programm und Daten zu verschwimmen, da ein RECORD sehr komplex, sogar mit mehreren Möglichkeiten, ähnlich wie ein Programm aufgebaut werden kann.

Die Vorteile eines RECORDs liegen darin, daß man sich z.B. bei Diskoperationen nicht darum kümmern muß, aus welchen Variablen der Verbund besteht, sondern es wird in einem Rutsch der ganze RECORD auf die Disk geschrieben.

In der nächsten Folge werden sich die Nebel um die RECORDs noch etwas lichten, und auch das Programm "notizblock" werden wir genauer betrachten; so wird z.B. ein Algorithmus verwendet, der erkennt, ob ein File schon existiert oder nicht, und auch von den "Compilerswitches" werden wir hören.

RAFAEL RAZIM

```
program notizblock;

TYPE notiz=RECORD
    tag: 1..31;
    monat: 1..12;
    jahr: 1980..2000;
    inhalt: STRING(.50.);
end;
notizblock=ARRAY(.1..20.) of notiz;
```

```
VAR block: notizblock;
    datafile: file of notizblock;
    name:STRING(.15.);
    written:BOOLEAN;
    ein:CHAR;
```

```
PROCEDURE openfile;
VAR error:INTEGER;
BEGIN
    (*$I-*);
    REPEAT
        CLRSCR;
        WRITELN('');
        WRITE('Name des Notizblocks: ');
        READLN(name);
        ASSIGN(datafile,name);
        RESET(datafile);
        error:=IORESULT;
        IF error=144 THEN
            CLOSE(datafile);
        IF error=1 THEN
            BEGIN
                REWRITE(datafile);
                error:=0;
            END;
        UNTIL error=0;
    (*$I+*);
END;
```

```
PROCEDURE input_termin;
BEGIN
END;
```

```
PROCEDURE suche_termin;
BEGIN
END;
```

```
BEGIN
    openfile;
    written:=FALSE;
    IF FILESIZE(datafile)=1
    THEN READ(datafile,block);
    REPEAT
        CLRSCR;
        WRITELN('1.Termin eingeben');
        WRITELN('2.Termin abfragen');
        WRITELN('3.Programmende');
        ein:=CHR(BIOS(2));
        CASE ein OF
            '1': input_termin;
            '2': suche_termin;
        END;
    UNTIL ein='3';
    IF written THEN
        BEGIN
            RESET(datafile);
            WRITE(datafile,block);
        END;
    CLOSE(datafile);
END.
```



```
*****
*
*   Probleme bei Datenübertragungen
*
*****
```

Im Rahmen des Schwerpunktes dieser Ausgabe - nämlich "Datenübertragung und Kommunikation", - soll der vorliegende Artikel einige der Probleme aufzeigen, mit denen ein Schnittstellenneuling konfrontiert wird.

Im Bereich der Home- und Personalcomputer finden im wesentlichen zwei Schnittstellen, die eine Verbindung mit der Außenwelt ermöglichen, Verwendung. Fast jeder kennt ihre Namen. Während die eine, Centronics-Schnittstelle genannt, einen guten Ruf genießt, wird die andere, die RS-232 C-Schnittstelle, so weit wie möglich gemieden, weil man mit ihr entweder schon selber schlechte Erfahrungen gemacht hat oder weil man aus Erzählungen verzweifelter Anwender wahre Schreckensgeschichten hört.

#### Warum Centronics so beliebt ...

Im Inneren eines Computers werden die Daten, im allgemeinen immer Byte für Byte übertragen, das heißt, es werden alle acht Bits über acht Leitungen gleichzeitig gesendet. Diese Methode, die parallele Übertragung genannt wird, ist äußerst schnell, aber sehr störanfällig. Sie ist jedoch vertretbar, sofern die Leitungen nicht zu lange sind (im Computer nur Milli- oder Zentimeter von Bauteil zu Bauteil) und die Übertragung in einer berechenbaren (im wahrsten Sinne des Wortes) Umgebung (in bezug auf Temperatur, Kapazität und Induktivität) stattfindet. Diese Bedingungen sind außerhalb des Computers nicht mehr gegeben, daher funktioniert die Übertragung zuverlässig auch nur auf eine Entfernung von einigen Metern. Sie ist also nur zur Verbindung mit Peripheriegeräten geeignet, die sich in unmittelbarer Nähe des Computers befinden, zum Beispiel Drucker.

Eine zweite Rolle spielen dabei auch die Kosten. Ein paralleles Kabel braucht außer den acht Datenleitungen auch noch Steuerleitungen, die bestimmte Signale des Computers an den Drucker leiten und umgekehrt. Ein kleines Beispiel dazu: Sie wollen einen Text zum Drucker schicken, dieser ist jedoch nicht bereit (nicht eingeschaltet oder nicht im ONLINE-Modus). Gäbe es nur die Datenleitungen, würde der Computer die Daten hinaus-schicken, ohne sich um den Drucker zu kümmern. So ist es jedoch nicht; der Computer nimmt solange keine anderen Befehle an, bis der Drucker aktiviert wird und sein "Bereit"-Signal sendet.

Eine zweite Tatsache, die die Centronics-Schnittstelle so beliebt gemacht hat, ist der Umstand, daß man nur das Verbindungskabel an Rechner und Drucker anschließen muß (sofern die Stecker passen) und sofort drucken kann.

#### ... und RS-232 C so unbeliebt ist

Um Daten auf größere Entfernung übertragen zu können, mußte man jedoch eine andere Methode wählen. Die Alternative ist natürlich klar: statt alle acht Bits gleichzeitig zu übertragen, übermittelt man sie hintereinander.

Das bedeutet zwar eine Verringerung der Übertragungsgeschwindigkeit, ist jedoch wesentlich sicherer und vor allem billiger zu verwirklichen, da man im Prinzip mit nur zwei Leitungen auskommt.

Im Laufe der Zeit entwickelten verschiedene Konstrukteure eigene Schnittstellen. Als dann noch die Möglichkeit hinzukam, Daten über Telefonleitungen mittels eines Modems zu übertragen, wurde die Situation vollends unübersichtlich, sodaß einigen Herstellern verboten wurde, ihre Schnittstellen und Übertragungsgeräte im öffentlichen Telefonnetz zu verwenden. Die Hersteller und die Telefongesellschaften verlangten nach einer Vereinheitlichung der Schnittstelle. So wurde ein Standard geschaffen, der die Bezeichnung RS-232 C erhielt (RS steht für Recommended Standard, das heißt empfohlener Standard).

Die Schnittstelle war ursprünglich nur für die Verbindung von Computern und Modems gedacht. Da die Hersteller diese Schnittstelle jedoch auch zum Anschluß anderer Geräte mißbrauchten, bekam sie zu Unrecht einen schlechten Ruf (Die Hersteller haben auch dadurch, daß sie sich nicht auf einen gemeinsamen Stecker einigen konnten, zur Unbeliebtheit der Schnittstelle beigetragen. Ein Steckertyp, der sich jedoch ziemlich durchgesetzt hat, ist der D-förmige DB-25-Stecker, der über zwei Stiftreihen mit oben 13 und unten 12 Stiften verfügt, die vom Kabel her gesehen von links oben durchnummeriert sind.).

Es gibt nämlich in der RS-232-Kommunikation zwei verschiedene logische Geräte: eine Datenendeinrichtung (engl. Data Terminal Equipment, DTE), im allgemeinen ein Computer, und eine Datenkommunikationseinrichtung (englisch Data Communications Equipment, DCE), das Modem. Diese beiden sind so konstruiert, daß Daten über Stift 2 des Steckers der DTE gesendet und über Stift 2 des Steckers der DCE empfangen werden. In die umgekehrte Richtung wird Stift 3 verwendet. Hier ergeben sich bereits die ersten (allerdings leicht lösbaren) Schwierigkeiten. Wenn Sie nämlich versuchen, zwei Computer, die beide als DTE konfiguriert sind, miteinander zu verbinden, wird sich nichts tun, weil jeweils die Sende- und die Empfangsstifte miteinander verbunden sind, anstatt der Sendestift des einen mit dem Empfangsstift des anderen. Sie können jedoch durch Umlöten der Kabel das Problem leicht beheben.

Das war nur eine der vielen Schwierigkeiten, die sich schon auf der Hardwareseite ergeben haben. Auf der Softwareseite ist vor allem darauf zu achten, daß die Übertragungsprotokolle der Geräte übereinstimmen (Übertragungsgeschwindigkeit, Parität, Wortlänge, etc.). Falls Sie sich trotz der vorangegangenen Zeilen dennoch mit der seriellen Kommunikation näher befassen wollen, können Sie zum Beispiel in

Campbell, J.  
V24/RS-232 Kommunikation  
Sybex

näheres dazu lesen. Zu Ihrer Information sei gesagt: die serielle Schnittstelle des SVI-328 ist als DTE konfiguriert und es wird ein DB-25-Stecker verwendet.

### Eine Fallstudie

Nachdem der erste Teil des Artikels ziemlich allgemein gehalten war, möchte ich jetzt auf ein konkretes Beispiel eingehen.

Angenommen, Sie wollen zwischen ihrem SVI-328 und einem anderen Rechner, zum Beispiel dem ACORN BBC B, Daten und Programme austauschen. Da die Rechner vom Diskettenformat her nicht kompatibel sind, bleibt nur der Weg über die serielle Schnittstelle. Hier ergibt sich das erste Problem: laut Handbuch verfügt der ACORN über keine RS-232 C, sondern über eine RS-432 C-Schnittstelle. Bei näherer Betrachtung erweist sich dieses Problem als geringfügig, da die Schnittstellen kompatibel sind und der Anschluß im ACORN-Handbuch ausführlich beschrieben ist. Da der ACORN über ein Verbindungskabel mit einem für seine Schnittstelle passenden Stecker verfügt, entstehen lediglich Kosten für einen DB-25-Stecker (ca. 90 Schilling).

Nachdem die Verbindung zwischen Stecker und Kabel hergestellt (im ACORN-Handbuch ausführlich beschrieben), die Verbindung zwischen den Rechnern hergestellt und die Schnittstellen entsprechend initialisiert sind, steht der Datenübertragung nichts mehr im Wege. Oder doch? Mit welchem Programm sollen die Daten übertragen werden? Die Verwendung des Dienstprogrammes PIP funktioniert nur für ASCII-Dateien in beide Richtungen, da das Ende von Nicht-ASCII-Dateien nicht erkannt wird (in der PIP-Beschreibung im CP/M-Handbuch findet sich darüber kein Hinweis.). Es blieb also nichts anderes übrig, als für das Empfangen von Nicht-ASCII-Dateien eigene Programme zu schreiben (das Senden von Daten funktionierte zumindest beim SVI mit PIP). Das SVI-Programm

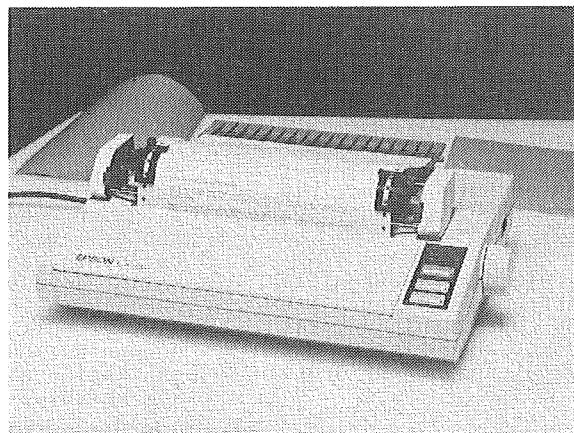
wurde in TURBO-Pascal geschrieben, das des ACORN in BBC-BASIC.

Hier ergab sich die nächste Schwierigkeit, denn sobald der Buffer des ACORN voll war und er die Daten auf die Diskette schrieb, schickte der SVI weitere Daten, die jedoch verloren gingen, das heißt, der SVI "wartete" nicht auf den ACORN. Eine Nachfrage bei unserem CP/M-Spezialisten ergab, daß es im BIOS des Rechners eine Speicherstelle gibt, die bestimmt, ob ein ankommendes "Bereit"-Signal ignoriert wird oder nicht. Nachdem diese auf den richtigen Wert geändert wurde, sandte der SVI erst dann Daten, wenn der ACORN dazu bereit war. Die Datenübertragung funktionierte also bereits in eine Richtung. Die Übertragung von Nicht-ASCII-Dateien zum SVI brachte jedoch das nächste Problem: es stellte sich nämlich heraus, daß das höchstwertige Bit jedes ankommenden Bytes vom SVI gelöscht wurde. So wurde für den ACORN ein eigenes Sendeprogramm geschrieben, das jedes Byte in zwei Hälften zerlegt und erst dann hinaus-schreibt. Das Empfangsprogramm am SVI wurde so modifiziert, daß es die ankommenden Informationen wieder in die tatsächlichen Bytewerte zurückwandelte. Zu guter Letzt funktionierte die Übertragung doch problemlos, aber angesichts der hier aufgetretenen Schwierigkeiten scheint es nicht mehr verwunderlich, daß Anwender durch serielle Schnittstellen zu Wahnsinnstaten getrieben werden.

In der nächsten Ausgabe beabsichtige ich, ein Kommunikationsprogramm zu veröffentlichen, das es ermöglichen soll, Daten zwischen zwei beliebigen Computern zu übertragen, ohne daß die Benutzer dem Wahnsinn anheimfallen.

## LX-80 und LX-90. Gut und günstig.

*Kompakt, leistungsstark und äußerst anpassungsfähig, das ist der LX-90 von EPSON. Für Einsteiger wie für Profis ein guter Drucker. Über die entsprechenden Interface-Module versteht er sich mit jedem Computer.*



*Fürs reibungslose Drucken ist der LX-80 bestens gerüstet. Hier mit einer Traktorführung, die es wie den Einzelblatt-Einzug als Option gibt. Der NLO-Mode für schönes Schreiben ist eingebaut.*

## EPSON hat noch mehr Drucker.

# Telebox

Seit 1. November ist unser Club auch über die Telebox von Radio-Austria erreichbar. Wie das funktioniert, welche Vorteile es bietet, dabei zu sein, und auch über die Kosten informiert der folgende Artikel.

## Was ist Telebox?

Telebox ist ein öffentlicher Textspeicherdienst - ein Gemeinschaftsprojekt der Post und der Radio-Austria AG. In einer zentralen elektronischen Einheit werden schriftliche Nachrichten von und für Kunden gespeichert. Berechtigte Benützer können diese Nachrichten jederzeit mittels Kennwort abrufen.

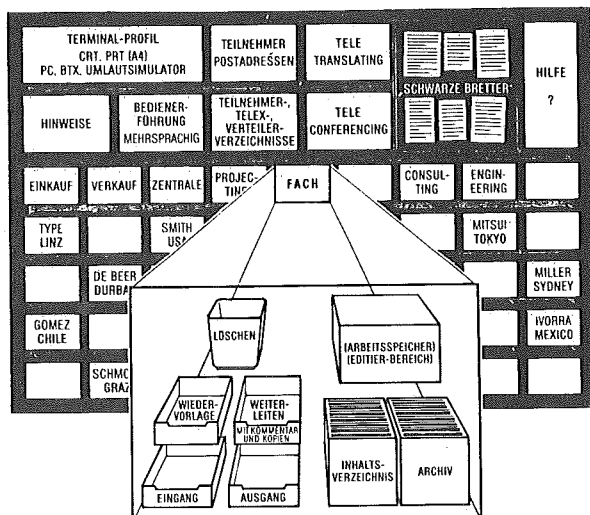
Telebox kann Text und Daten speichern. Man wird Telebox dann einsetzen, wenn zwischen bestimmten Korrespondenzpartnern ein regelmäßiger Nachrichtenaustausch stattfindet, z.B. zwischen der Zentrale eines Betriebes und Filialen im In- und Ausland, für Mitarbeiter im Außendienst, die Aufträge an die Zentrale übermitteln, für den Informationsfluß zu Lieferanten und Kunden. Aber auch im Privatleben läßt sich Telebox sinnvoll einsetzen.

Telebox stellt auch "Schwarze Bretter" zur Verfügung. In diese kann jeder Teilnehmer öffentlich zugängliche Eintragungen vornehmen (also ohne Kennwort erreichbar). Firmen können dort ihr Angebot anschlagen, Private können Gebrauchtgeräte zum Tausch oder Verkauf anbieten, der Anwendung sind keine Grenzen gesetzt.

Dank der Vielseitigkeit der Telebox ist auch der Zugang zu allen Telex-Teilnehmern möglich. Jeder Fernschreiber der Welt ist über die Telebox erreichbar, jeder Telebox-Teilnehmer kann von allen Fernschreibern aus erreicht werden.

Die Grafik liefert eine Übersicht über die Möglichkeiten der Telebox.

DIE TELEBOX DER RADIO-AUSTRIA.



## Was kostet die Telebox-Teilnahme?

Die einmaligen Herstellungskosten und die Eintragungsgebühr betragen je S 300,-. Die monatliche Grundgebühr für ein Fach (mit Fassungsvermögen von 100 Nachrichten) beträgt S 300,-. Bis Ende dieses Jahres entfällt diese Gebühr als Einführungsangebot. Die Benutzungszeit kostet pro Minute 2,- Schilling, abgerechnet wird in Sekunden. Pro abgesandter Nachricht zahlt man 50 Groschen. Die Abspeicherung von 20 Nachrichten pro Tag ist gebührenfrei, für jede zusätzliche Nachricht werden 15 Groschen berechnet.

Wie man sieht, ist die Angelegenheit äußerst preiswert. So sind beispielsweise die monatlichen Grundgebühren und vorgeschriebenen Wartungsentgelte bei Fernschreibanschlüssen wesentlich höher.

## Wie wird man Telebox-Teilnehmer?

Man bestellt bei Radio-Austria-AG die Anmeldeunterlagen. Man erhält postwendend ausführliche Unterlagen über die Telebox, ein Handbuch, einen Vertrag und auch gleich die entsprechenden Kennwörter zugeteilt.

Zwei bis drei Tage nach Rücksendung des unterzeichneten Vertrages erhält man von Radio-Austria die telefonische Verständigung, daß man die Telebox-Einrichtung benutzen kann. Nun steht nichts mehr im Wege, die Anlage in Betrieb zu nehmen und mit dem Datenaustausch zu beginnen.

## Welche Hardware benötigt man für Telebox?

Prinzipiell kann man die Telebox-Einrichtungen mit jedem Computer benutzen, der über eine serielle Schnittstelle verfügt. Wir haben unsere Anlage mit einem SVI-738 X'press ausgestattet, der bereits eine serielle Schnittstelle eingebaut hat. Ein Terminalprogramm befindet sich bereits im ROM und kann vom BASIC aus mit CALL COMTERM aufgerufen werden. Vorher muß man allerdings die Schnittstelle initialisieren. Für die Telebox gilt dabei folgender Aufruf:

CALL COMINI ("0:7E1XNNNN",300,300,0)

Dann ist die Schnittstelle richtig eingestellt. Wer genauer wissen will, was die einzelnen Parameter bedeuten, kann vorher mit CALL COMHELP nähere Informationen erhalten. Die Telebox arbeitet mit 7 Bit-Übertragung, even parity, einem Stopp-Bit und einer Übertragungsrates von 300 Baud.

Neben unserem Computer benötigen wir noch einen Akustikkoppler und das entsprechende Anschlußkabel. Wir verwenden den Sonic 300, einen Testbericht über diesen preiswerten Koppler finden Sie in diesem Heft.

Als Sichtgerät genügt übrigens ein Fernseher, man kann aber auch die komfortablere 80 Zeichen-Darstellung mit Monitor wählen.

Um die Telebox als Fernschreiber zu verwenden

Fortsetzung auf Seite 13

## SONIC-300

Das wichtigste Werkzeug für die Datenfernübertragung (DFÜ) ist der Akustikkoppler. Er eröffnet den Zugang zum Telefonnetz und über die Vermittlung durch Radio Austria auch zum Fernschreibnetz. Den Zugang zu diesen Netzen behindern keine nationalen Grenzen. Weltweit stehen Ihnen diese Möglichkeiten zur Verfügung und Sie können Mailboxen und Datenbanken erreichen. Ein preiswerter Akustikkoppler ist der SONIC 300.

Die wichtigste Frage beim Kauf eines Akustikkopplers sollte die Frage nach der Postzulassung sein. Denn nur Akustikkoppler mit dem Prüfzeichen der österreichischen Post dürfen in Österreich betrieben werden. Der von uns beschriebene Akustikkoppler verfügt über dieses "Prüfpickerl". Die Aufkleber sind auf der Unterseite des Akustikkopplers angebracht und tragen neben dem Bundeswappen auch die Prüfnummer (PTV-ONr.) der Post.

Wie funktioniert nun der Datenaustausch über das öffentliche Telefonnetz? Neben dem Akustikkoppler benötigen wir natürlich einen Computer mit einer geeigneten Schnittstelle. Bei Spectravideo bieten sich dabei folgende Möglichkeiten an:

Der neue SVI-738 X'press verfügt bereits über eine eingebaute serielle Schnittstelle und über ein geeignetes Programm (im ROM verfügbar) zur Initialisierung der Schnittstelle und zur Datenübertragung. Über unsere Erfahrungen mit dieser Konfiguration berichtet der Artikel über die Kommunikation mit der Telebox in diesem Heft.

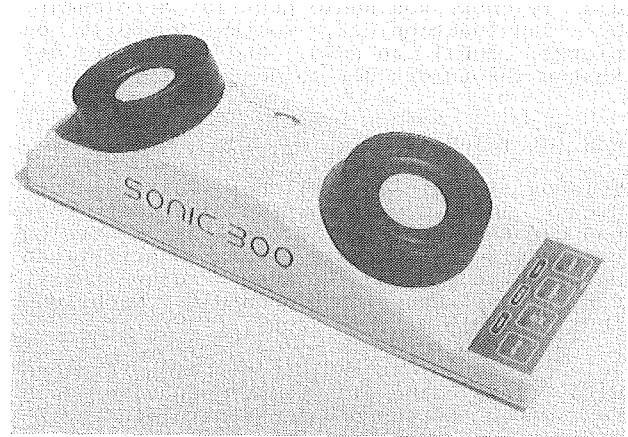
Der MSX-Computer SVI-728 bietet die selben Möglichkeiten, allerdings ist die RS232-Schnittstelle nicht eingebaut, sondern als separate Steckkarte erhältlich und wird in den Modulschacht des Computers gesteckt.

Ähnlich ist es beim SVI-328, wo allerdings auch das Datenübertragungsprogramm erstellt werden muß. Wir werden in einem der kommenden Hefte ein derartiges Programm (verwendbar für alle SVI-Computer und mit mehr Komfort als die bei MSX eingebauten Übertragungsprogramme) bringen.

Nun brauchen wir noch ein geeignetes Kabel für die serielle Datenübertragung zwischen Computer und Akustikkoppler. Ein solches Kabel kann man selbst herstellen. Man kann es auch fertig gemeinsam mit dem Akustikkoppler kaufen. Ein solches Kabel kostet je nach Länge zwischen 450 und 550 Schilling.

Und nun zum Akustikkoppler SONIC 300. Er arbeitet mit dem Übertragungsstandard CCITT-Norm V 21 (von der Post vorgeschrieben) mit einer fix eingestellten Übertragungsgeschwindigkeit von 300 Baud (300 Bit pro Sekunde) im Halb- und Vollduplex-Betrieb.

Im Gegensatz zu den meisten angebotenen Akustikkopplern kann man den SONIC 300 direkt an das 220 V-Netz anschließen. Man kann ihn jedoch auch mit Batterie oder Akku betreiben (2 x 9 Volt), der Platz dafür ist hinter einem abnehmbaren Deckel an der Rückseite des Kopplers vorgesehen. Wenn Akkus im Batteriefach eingelegt sind, werden diese bei Netzbetrieb automatisch aufgeladen.



Man kann den Akustikkoppler also sowohl stationär als auch mobil einsetzen (in Telefonzellen stehen leider keine 220 Volt-Anschlüsse zur Verfügung).

Der SONIC 300 bietet uns folgende Funktionen:

- originate für den Dialog mit Telebox und Mailboxen
- answer wenn man selbst eine automatische Mailbox einrichtet
- self-test zur Prüfung der Funktionsfähigkeit der Geräteanordnung

Der SONIC 300 besitzt eine Folientastatur mit sieben LED-Anzeigen für originate, answer, Ein-Aus, ready, self-test, stand-by und level.

Die Ankoppelung ans Telefonnetz erfolgt akustisch über den Hörer des Telefonapparates. Die Konstruktion der Gummimuscheln ermöglicht auch die Verwendung von Hörern, die keine runde Form der Hör- oder Sprechmuschel aufweisen. Die Gummimuscheln sind auf dem Gehäuse des Akustikkopplers beweglich angebracht und erleichtern durch Verschieben die Anpassung an die Hörerform. Durch Federdruck werden die Schallwandler an den Telefonhörer angepreßt. Eine Druckausgleichsöffnung verhindert, daß die Sprechkapseln durch Druckunterschiede beschädigt werden.

Der Sonic 300 kann mit dem Ein-/Aus-Schalter nur dann eingeschaltet werden, wenn ein Rechner oder Datenterminal angeschlossen ist. Wird der Computer ausgeschaltet, schaltet sich auch der Akustikkoppler automatisch ab. Bei Batteriebetrieb wird dadurch die Entladung der Batterie verhindert, wenn man vergessen hat, das Gerät auszuschalten.

Abschließend noch zum Äußeren des Gerätes. Der Sonic 300 ist 28 cm lang, knapp 9 cm breit und 5 cm hoch. Das Gehäuse ist formschön, die Farbe ist elfenbein, Tastatur und Aufdrucke sind silber. Die Auflegerichtung des Telefonhörers ist durch die Abbildung eines Hörers mit Kabel signalisiert.

Der Akustikkoppler SONIC 300 ist im Computer-Studio und in guten Fachgeschäften zum Preis von S 3.590,- erhältlich.

```
*****
*                               *
*           Schneller Sinus/Kosinus           *
*                               *
*****
```

Bei Graphikanwendungen stört oft die langsame Geschwindigkeit des Microsoft-Sinus. Da man im BASIC darauf Rücksicht nehmen muß, daß man auch einmal einen 14stelligen Sinus braucht, gibt es keine Möglichkeit, für die Graphik ungenaue aber schnelle Sinusberechnungen zu bekommen. Daher muß man ein eigenes Maschincodeprogramm schreiben.

### Die Konzeption

Geplant war ein Programm, das so schnell wie möglich aber doch noch halbwegs komfortabel Sinus und Kosinus berechnet. Dabei wurde eine Genauigkeit von 3 Nachkommastellen gewählt, die für die meisten Graphikanwendungen ausreicht. Außerdem wurde darauf Rücksicht genommen, daß kleinere Genauigkeiten schneller ausgeführt werden als größere. Zu guter Letzt wird auch der Kosinus, da er als 90-Sinus definiert ist, etwas langsamer bearbeitet als der Sinus. Die Parameter werden in Altgrad angegeben, Bogenmaß sind zwar Standard, aber in Österreich sind Grad noch immer anschaulicher.

Das Programm wurde für BASIC geschrieben, es wird daher mit USR aufgerufen und bearbeitet alle numerischen Variablentypen. Integervariablen werden schneller bearbeitet als SNG- oder DBL-Variablen. Der Bereich wird auf -1000 bis +1000 Grad eingeschränkt. Die Schrittrate beträgt 1 Grad, Zwischenschritte

in der Schrittweite von einem Zehntelgrad werden bei Bedarf linear erzeugt. Dadurch erreicht man auch bei großen Kreisradien sehr genaue Bilder. Natürlich kann man nur bei SNG- oder DBL-Variablen Zehntelschritte verwenden, bei ganzzahligen Variablen funktioniert das natürlich nicht.

Definiert werden die Aufrufe am besten mit

```
DEFUSR=&HC000:DEFUSR1=&HC001,
```

aufgerufen wird folgendermaßen:

```
Sinus : USR(Argument)
Kosinus: USR1(Argument)
```

Natürlich kann man das Programm auch in Maschincodeprogramme selber einbauen. Hier bleibt es dann jedem überlassen, in welche Genauigkeitsstufe (und damit Schnelligkeitsstufe) man einsteigt. Das Programm jedenfalls ist so konzipiert, daß man ohne größere Schwierigkeiten Teile davon als Subroutinen adaptieren kann.

Über die Anwendungsmöglichkeiten von Sinus und Kosinus wurde schon sehr viel geschrieben, wir wollen hier einmal die praktische Seite voraussetzen. Deshalb können wir uns gleich auf das Listing stürzen und es genauer betrachten.

### Das Programm selber:

Wenn wir uns das Assemblerlisting genauer ansehen, dann erkennen wir, daß es am Anfang durch die Verschiebung des Aufrufes um ein Byte zu der Unterscheidung zwischen Sinus

### Telebox

Fortsetzung von Seite 11

den, ist natürlich der Anschluß eines Druckers empfehlenswert. Wir verwenden einen EPSON RX-80. Wir schalten ihn nur dann dazu, wenn Nachrichten ausgedruckt werden sollen. Beim Kommunikationsprogramm, das im SVI-738 X'press verfügbar ist, geht das Zuschalten durch Drücken der Funktionstaste F8. Durch neuerliches Betätigen dieser Taste wird der Drucker wieder weggeschaltet.

### Das Arbeiten mit Telebox

Mit den Anmeldeformularen erhält man eine 44-seitige Bedienungsanleitung, mit deren Hilfe der Zugriff auf die Dienste der Telebox mühelos möglich ist. Das Inhaltsverzeichnis ist logisch und alphabetisch gegliedert und ermöglicht das schnelle Auffinden der Funktionen. Die aktuellste Ausgabe der Bedienungsanleitung kann man übrigens jederzeit auch vom Computer abrufen und ausdrucken.

Nach eingehendem Studium dieser Gebrauchsanweisung wollen wir nun die Kommunikation mit dem Computer von Radio Austria aufbauen. Wir beginnen wie oben beschrieben mit der Initialisierung der Schnittstelle (CALL COMINI...) und rufen das Terminalprogramm auf (CALL COMTERM). Nun legen wir den Telefonhörer auf den Akustikkoppler (fest anpressen bis der Hörer gut sitzt) und wählen

die Telefonnummer der Telebox. Nach zweimaligem Signal des Telefons meldet sich der Computer von Radio Austria mit einem Pfeifton, der nach kurzer Zeit seine Frequenz ändert. Nun wird die originate-Taste am Koppler gedrückt und die Verbindung ist hergestellt. Am Bildschirm erscheint die Aufforderung "Enter speed detection character =".

Wir befolgen diese Anweisung und Radio Austria meldet sich am Bildschirm. Nun wird unsere Kennung NUI (Network User Identification) verlangt. Wir geben ein:

```
"N (unsere Kennung) ,." und ENTER
```

Jetzt meldet sich bereits die Telebox und verlangt unser Kennwort (Password). Wenn wir auch dieses eingegeben haben, begrüßt uns Telebox je nach Tageszeit mit einem freundlichen Guten Morgen, Guten Tag oder Guten Abend.

Fehler bei der Kennworteingabe werden nur zweimal toleriert, nach dem dritten Fehler schaltet die Telebox ab.

Die Telebox kann übrigens von ganz Österreich aus zum Ortstarif angerufen werden.

Den Spectra Video Club Austria können Sie über die Telebox-Anschrift "Computer-Studio" erreichen. In die Betreff-Spalte schreiben Sie "SVI-Club".

## Programme

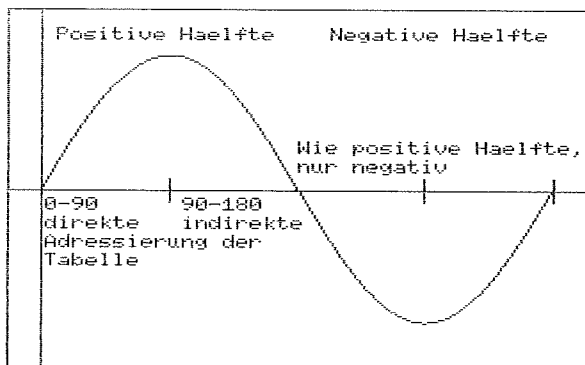
und Kosinus. Dieser Trick wird übrigens in der Computertechnik oft verwendet. Für den Sinus wird 'LD A,AFH' (Bytefolge 3E AF) ausgeführt, 'XOR A' (Bytefolge AF) wird beim Kosinus verarbeitet. Durch diese Formel kann man A in gelöscht oder nicht gelöscht unterscheiden und ein Flag damit setzen.

Nach dieser Unterscheidung teilt sich das Programm in zwei Teile. Der Integerteil tut im wesentlichen das Gleiche wie der andere, mit dem Unterschied, daß alles etwas einfacher getan wird. Sehen wir uns nun an, wie die SNG- und DBL-Variablen verarbeitet werden.

Zuerst muß man des Wertes habhaft werden, den wir verarbeiten. Wir werden das Argument in der Form '###.#' berechnen, daher werden wir die Mantisse in zwei Bytes so verschieben, daß die oben abgebildete Form gebildet wird. Die ersten Operationen testen daher auf Overflow und verschieben anhand des Exponenten die Mantisse. Haben wir einmal die Form, wird sie ins Registerpaar HL geladen und dort weiterverarbeitet.

Da wir einen Bereich von -1000 bis +1000 angegeben haben, müssen wir erst einmal auf den Bereich -360 bis +360 abspecken. Zugleich wird auch 90-Wert ausgerechnet, sollte ein Kosinus gewünscht sein. Alles wird übrigens in BCD-Arithmetik berechnet. Haben wir den relativ engen Bereich einmal gesichert, bringen wir das "-", falls eines da ist, weg. Wir addieren einfach mit 360 und haben den equivalenten positiven Wert.

Nach diesen Operationen müssen wir die 360 Grad in vier Teile teilen, denn jeder Quadrant hat andere Regeln zur Herstellung des Sinus. Im Bild kann man deutlich die verschiedenen Regeln sehen. Wir müssen nun den Computer dazu bringen, daß er anhand der Tabelle von 0 bis 90 Grad alle vier Quadranten mit Werten versorgt. Daher gehen wir schrittweise vor und suchen einmal alle die Quadranten, die vor der Zahl ein negatives Vorzeichen haben werden. Alle Zahlen über 180 Grad reagieren so. Wenn wir 180 subtrahieren und nur das "-" vormerken, haben wir diesen Teil erledigt.



Etwas schwieriger wird die Teilung innerhalb der 180 Grad. Hier ist zwar das Vorzeichen gleich, doch muß bei 90° die Richtung der Zählweise geändert werden. Ist das Argument also jetzt noch größer als 90 (oder gleich), dann verwenden wir 180-Argument, um auf die richtige Adressierung zu kommen. Nach diesem

Trick haben wir alle vier Quadranten bearbeitet und können uns nun um die Ansteuerung auf die Tabelle bemühen.

Jedes Grad hat einen Wert mit zwei Bytes in der Tabelle. Daher müssen wir die Gradzahl in normalen Hexcode umwandeln und mit zwei multiplizieren. Noch haben wir ja BCD-Zahlen verwendet. Gleichzeitig retten wir die Nachkommastelle, um später noch mit ihr weiterzuarbeiten. Ist sie Null, wird nur die Gradzahl geladen. Hat sie jedoch einen Wert zwischen Eins und Neun, werden die zwei Zahlen (Gradwert und Gradwert + 1) geladen und deren Differenz durch 10 dividiert. Wenn man nun noch die Zahl, die sich ergeben hat mit der Nachkommastelle multipliziert und den niederen Wert der Tabelle addiert, erman eine lineare Näherung.

Nach dieser Berechnung werden nur mehr organisatorische Probleme erledigt, der BASIC-Akkumulator wird als Single-Variable eingestellt und das Argument abgelegt.

Der Weg eines Integer-Arguments geht nicht viel anders vor sich. Es werden allerdings keine BCD-Zahlen verarbeitet, sondern ganz normale Werte. Erst in einer der letzten Phasen treffen beide Routinen zusammen.

Zum Schluß noch eine Tabelle, die Aufschluß darüber geben soll, warum dieser Sinus entwickelt worden ist. Dokumentiert braucht sie nicht mehr zu werden:

	Microsoft	FASTSIN
1000 Sinus:	80 sec	.5 sec
Kreis zeichnen (1 Grad Step):	66 sec	8.5 sec
Kreis zeichnen (.5 Grad Step):	122 sec	16 sec

```

org c000h
; error ist Einsprung für Fehler-
;
; erzeugung, Fehlernummer in E.
; vartyp enthält Variablentyp des
; BASIC-Akkus.
;
error equ 96h
vartyp equ f793h
;
; Sinus oder Kosinus? Bestimmen
; des Variablentyps.
;
sinus defb 3eh
cosin  xor a
      ld (cos),a
      xor a
      ld (negtem),a
      ld (reserv),hl
      ld a,(vartyp)
      cp 2
      jp z,integ
      cp 4
      jr z,just
      cp 8
      jp nz,illeg

```

```

;
; SNG- und DBL-Routine
; auf Bereich prüfen und in Matrix
; ###.# pressen.
;
just   ld a,(hl)
       push af
       and 7fh
       cp 44h
       jp nc,ovflow
       cp 40h
       jp c,zero
       inc hl
       push af
       xor a
       ex af,af'
corr   pop af
       cp 43h
       jr z,right
       ex af,af'
       rrd
       inc hl
       rrd
       dec hl
       xor a
       ex af,af'
       inc a
       jr corr
;
; Lade DE mit Zahl, 90-Zahl, wenn
; Kosinus verlangt. Danach Negativ-
; ausgleich und unter 360 bringen.
;
right  ld d,(hl)
       inc hl
       ld e,(hl)
       ex de,hl
       ld a,(cos)
       or a
       call z,cosdbl
       ld de,3600h
bigger rst 20h
       jr c,negnon
       ld a,h
       sub d
       daa
       ld h,a
       jr bigger
negnon pop af
       rla
       ld a,(negtem)
       rra
       or a
       jr z,noneg
       cp d7h
       jr z,noneg
       call subhd
;
; Prüfen des Wertes auf Quadranten
; Setzen des "-" in Abhängigkeit davon
;
noneg  ld a,17h
       cp h
       jr nc,less
       ld a,h
       sub 18h
       daa
       ld h,a
       jr nz,notest
       or l
       jr z,less
notest defb 3eh
       less xor a
       ld (neg),a
       ld a,8
       cp h

```

```

       jr nc,less1
       ld de,1800h
       call subhd
;
; auf 0-90 gebracht, nun Zeiger auf
; Tabelle richten, Wert(e) auslesen
; und berechnen.
;
less1  ld a,h
       rla
       rla
       ld b,a
       rla
       rla
       add a,b
       push hl
       ld c,0
       ld b,4
loop   rl l
       rl c
       djnz loop
       rl c
       add a,c
       pop hl
       ld c,l
all    ld hl,tabel
       add a,l
       jr nc,noinc
       inc h
noinc  ld l,a
       ld a,c
       and 15
       ld (rest),a
       ld e,(hl)
       inc hl
       ld d,(hl)
       jr z,only
;
; Berechnung der Zwischenwerte.
;
       inc hl
       ld c,(hl)
       inc hl
       ld h,(hl)
       ld l,c
       ld a,l
       sub e
       daa
       ld l,a
       ld h,0
       ld b,4
loop1  rl l
       rl h
       djnz loop1
       ld a,(rest)
       ld b,a
       push de
       ld d,h
       ld e,l
       ld hl,0
loop2  call addhd
       djnz loop2
       pop de
       ld l,h
       ld h,0
       call addhd
;
; Ausgabe des Wertes, Richten der
; Variablenzeiger.
;
both   call negate
       jp write
only   ex de,hl
       jr both
;
; Routine, wenn Wert = 0.

```

```

zero   pop hl
        ld hl,0
        jr both
;
;
;   Addiere hl mit de
;
addhd  ld a,e
        add a,l
        daa
        ld l,a
        ld a,d
        adc a,h
        daa
        ld h,a
        ret
;
;   Fehlerbehandlungen.
;
ovflow pop af
        ld e,6
        defb 01
illeg  ld e,5
        jp error
;
;   Maskiert Ergebnis auf negativ.
;
negate ld a,(neg)
        or a
        ret z
        ld a,h
        or c0h
        ld h,a
        ret
;
;   speichert Ergebnis als SNG-Typ.
;
write  ex de,hl
        ld hl,(reserv)
        ld a,41h
        cp d
        jr nc,nomin
        set 7,a
        push af
        ld a,d
        and 1fh
        ld d,a
        pop af
nomin  ld (hl),a
        inc hl
        ld (hl),d
        inc hl
        ld (hl),e
        inc hl
        ld (hl),0
        inc hl
        ld (hl),0
        ld a,4
        ld (vartyp),a
        ei
        ret
;
;   Subtrahiere de-hl, e=0
;
subhd  xor a
        sub l
        daa
        ld e,a
        ld a,d
        sbc a,h
        daa
        ld d,a
        ex de,hl
        ret
;
;   Integerroutine

```

```

;   Beginn:
;
integ  inc hl
        inc hl
        ld c,(hl)
        inc hl
        ld h,(hl)
        ld l,c
        ld a,(cos)
        or a
        call z,cosin
        ld de,1000
;
;   Overflowtest
;
        rst 20h
        jr c,noov
        ld de,-1000
        rst 20h
        jr c,ovflow
        defb 3eh
;
;   kein Overflow, auf 0-360 bringen.
;
noov   xor a
        ld (negtem),a
        bit 7,h
        jr z,noneg2
        ld de,0
        ex de,hl
        sbc hl,de
noneg2 xor a
        ld c,a
        ld de,360
comp   rst 20h
        jp c,furter
        sbc hl,de
        jr comp
furter ld a,(negtem)
        or a
        jr z,furt1
        ex de,hl
        sbc hl,de
;
;   Quadranten bestimmen
;
furt1  ld de,181
        rst 20h
        jr c,furt2
        sbc hl,de
        inc hl
        defb 3eh
furt2  xor a
        ld (neg),a
        ld a,90
        cp l
        jr nc,noturn
        add a,a
        sub l
        defb 6
noturn ld a,l
        add a,a
;
;   Gradwert eliminiert, Sprung zu
;   gemeinsamem Zeigerjustieren.
;
        jp all
;
;   90-Wert INT, für Kosinus.
;
cosin  ex de,hl
        ld hl,90
        sbc hl,de
        ret
cosdbl ld de,900h
;

```



```

; 90-Wert SNG, für Kosinus.
;
    call subhd
    jr nc,noinv
    ld de,0
    call subhd
    defb 3eh
noinv xor a
    ld (negtem),a
    ret
reserv defw 0
rest defb 0
neg defb 0
cos defb 0
negtem defb 0
mant2 defw 0
;
; Sinuswertetabelle
;
tabel defw 0000H,0017H;00 01
defw 0035H,0052H;02 03
defw 0070H,0087H;04 05
defw 0105H,0122H;06 07
defw 0139H,0156H;08 09
defw 0174H,0191H;10 11
defw 0208H,0225H;12 13
defw 0242H,0259H;14 15
defw 0276H,0292H;16 17
defw 0309H,0326H;18 19
defw 0342H,0358H;20 21
defw 0375H,0391H;22 23
defw 0407H,0423H;24 25
defw 0438H,0454H;26 27
defw 0469H,0485H;28 29
defw 0500H,0515H;30 31
defw 0530H,0545H;32 33
defw 0559H,0574H;34 35
defw 0588H,0602H;36 37
defw 0616H,0629H;38 39
defw 0643H,0656H;40 41
defw 0669H,0682H;42 43
defw 0695H,0707H;44 45
defw 0719H,0731H;46 47
defw 0743H,0755H;48 49
defw 0766H,0777H;50 51
defw 0788H,0799H;52 53
defw 0809H,0819H;54 55
defw 0829H,0839H;56 57
defw 0848H,0857H;58 59
defw 0866H,0875H;60 61
defw 0883H,0891H;62 63
defw 0899H,0906H;64 65
defw 0914H,0921H;66 67
defw 0927H,0934H;68 69
defw 0940H,0946H;70 71
defw 0951H,0956H;72 73
defw 0961H,0966H;74 75
defw 0970H,0974H;76 77
defw 0978H,0982H;78 79
defw 0985H,0988H;80 81
defw 0990H,0993H;82 83
defw 0995H,0996H;84 85
defw 0998H,0999H;86 87
defw 0999H,1000H;88 89
end defw 1000H ;90
end

```

```

B000 : 3E AF 32 A9 C1 AF 32 AA ³.2...2.
B008 : C1 22 A5 C1 3A 93 F7 FE ."...:
B010 : 02 CA 31 C1 FE 04 28 05 ..1...().
B018 : FE 08 C2 F5 C0 7E F5 E6 .....ß..
B020 : 7F FE 44 D2 F1 C0 FE 40 ..D...$
B028 : DA E2 C0 23 F5 AF 08 F1 ...#....
B030 : FE 43 28 0C 08 ED 67 23 .C(...g#
B038 : ED 67 2B AF 08 3C 18 F0 .g+...²..
B040 : 56 23 5E EB 3A A9 C1 B7 V#°:....
B048 : CC 91 C1 11 00 36 E7 38 .....6.8
B050 : 06 7C 92 27 67 18 F7 F1 ..ö.'g...
B058 : 17 3A AA C1 1F B7 28 07 .....().
B060 : FE D7 28 03 CD 27 C1 3E ..(.'.³
B068 : 17 BC 30 0B 7C D6 18 27 ..0.ö..'¹
B070 : 67 20 03 B5 28 01 3E AF g...(.³
B078 : 32 A8 C1 3E 08 BC 30 06 2..³..0.
B080 : 11 00 18 CD 27 C1 7C 17 ....'ö.
B088 : 17 47 17 17 80 E5 0E 00 .G.....
B090 : 06 04 CB 15 CB 11 10 FA .....
B098 : CB 11 81 E1 4D 21 AD C1 ....M!..
BOA0 : 85 30 01 24 6F 79 E6 0F .0.$oy..
BOA8 : 32 A7 C1 5E 23 56 28 2F 2..°#V(/
BOB0 : 23 4E 23 66 69 7D 93 27 #N#fiü.'
BOB8 : 6F 26 00 06 04 CB 15 CB o&.....
BOC0 : 14 10 FA 3A A7 C1 47 D5 .....G.
BOC8 : 54 5D 21 00 00 CD E8 C0 TÜ!.....
BOD0 : 10 FB D1 6C 26 00 CD E8 ...l&...
BOD8 : C0 CD FA C0 C3 04 C1 EB .....
BOE0 : 18 F7 E1 21 00 00 18 F1 ...!....
BOE8 : 7B 85 27 6F 7A 8C 27 67 ä.'oz.'g
BOF0 : C9 F1 1E 06 01 1E 05 C3 .....
BOF8 : 96 00 3A A8 C1 B7 C8 7C .....ö
B100 : F6 C0 67 C9 EB 2A A5 C1 ..g..*..
B108 : 3E 41 BA 30 08 CB FF F5 ³A.0...
B110 : 7A E6 1F 57 F1 77 23 72 z..W.w#r
B118 : 23 73 23 36 00 23 36 00 #s#6.#6.
B120 : 3E 04 32 93 F7 FB C9 AF ³.2.....
B128 : 95 27 5F 7A 9C 27 57 EB .'z.'W.
B130 : C9 23 23 4E 23 66 69 3A .#N#fi:
B138 : A9 C1 B7 CC 8A C1 11 E8 .....
B140 : 03 E7 38 07 11 18 FC E7 ..8.....
B148 : 38 A7 3E AF 32 AA C1 CB 8.³.2...
B150 : 7C 28 06 11 00 00 EB ED ö(...
B158 : 52 AF 4F 11 68 01 E7 DA R.O.h...
B160 : 66 C1 ED 52 18 F8 3A AA f..R...:
B168 : C1 B7 28 03 EB ED 52 11 ..(...R.
B170 : B5 00 E7 38 04 ED 52 23 ...8..R#
B178 : 3E AF 32 A8 C1 3E 5A BD ³.2..³Z.
B180 : 30 03 87 95 06 7D 87 C3 0...ü..
B188 : 9D C0 EB 21 5A 00 ED 52 ...!Z..R
B190 : C9 11 00 09 CD 27 C1 30 .....¹.0
B198 : 07 11 00 00 CD 27 C1 3E .....¹.³
B1A0 : AF 32 AA C1 C9 00 00 00 .2.....
B1A8 : 00 00 00 00 00 00 00 17 .....
B1B0 : 00 35 00 52 00 70 00 87 .5.R.p..
B1B8 : 00 05 01 22 01 39 01 56 ...".9.V
B1C0 : 01 74 01 91 01 08 02 25 .t.....%
B1C8 : 02 42 02 59 02 76 02 92 .B.Y.v..
B1D0 : 02 09 03 26 03 42 03 58 ...&.B.X
B1D8 : 03 75 03 91 03 07 04 23 .u.....#
B1E0 : 04 38 04 54 04 69 04 85 .8.T.i..
B1E8 : 04 00 05 15 05 30 05 45 .....0.E
B1F0 : 05 59 05 74 05 88 05 02 .Y.t....
B1F8 : 06 16 06 29 06 43 06 56 ...)C.V
B200 : 06 69 06 82 06 95 06 07 .i.....
B208 : 07 19 07 31 07 43 07 55 ...¹.C.U
B210 : 07 66 07 77 07 88 07 99 .f.w....
B218 : 07 09 08 19 08 29 08 39 .....).9
B220 : 08 48 08 57 08 66 08 75 .H.W.f.u
B228 : 08 83 08 91 08 99 08 06 .....
B230 : 09 14 09 21 09 27 09 34 ...!.'¹.4
B238 : 09 40 09 46 09 51 09 56 .$.F.Q.V
B240 : 09 61 09 66 09 70 09 74 .a.f.p.t
B248 : 09 78 09 82 09 85 09 88 .x.....
B250 : 09 90 09 93 09 95 09 96 .....
B258 : 09 98 09 99 09 99 09 00 .....

```

Auf der folgenden Spalte folgt das Hexdump für das Programm. Ab der Speicherzelle B260H müssen noch folgende drei Zahlen angehängt werden:

B260 : 10 00 10

Die Gesetze von Easel Murphy und D.L. Klipstein  
ueber das Verhalten lebloser Gegenstaende

1. Toleranzen summieren sich stets nach der unguenstigen Seite.
2. Gleiche Teile, unter gleichen Vorraussetzungen geprueft, verhalten sich im Einsatz anders.
3. Die Sicherheit von Lieferterminen ist umgekehrt proportional zu den Beteuerungen ueber die Einhaltung der Termine.
4. Alle Konstanten sind variabel.
5. Die Ausfallwahrscheinlichkeit eines Bauteils ist umgekehrt proportional zu seiner Zugaenglichkeit.
6. Kommas stehen immer eine Stelle zu weit links.
7. Wenn  $n-1$  von  $n$  Schrauben geloest sind, stellt sich heraus, dass man das falsche Geraet geoeffnet hat.
8. Masseinheiten werden immer in den ungebraeuchlichsten Dimensionen angegeben, zum Beispiel: Geschwindigkeit in Angstroem pro Woche
9. In allen Ueberlegungen ist diejenige Groesse die haeufigste Fehlerquelle, die vorher ueber jeden Zweifel erhaben war.
10. Die Erhaeltlichkeit eines Teiles ist umgekehrt proportional zu seiner Wichtigkeit.
11. Zugesechnittene Draehnte sind immer zu kurz.
12. Ein aus der Hand fallendes Werkzeug faellt immer so, dass es den groesstmoeeglichen Schaden anrichtet. (sog. Gesetz der selektiven Schwerkraft)
13. Wird ein Geraet durch die Summe seiner moeglichen Plus- oder Minus-Fehler beschrieben, so ist der Gesamtfehler gleich der Summe aller Einzelfehler, addiert in die gleiche Richtung.

```
*****
*                               *
*           Assembler           *
*                               *
*****
```

Nachdem wir schon seit einigen Monaten nur Programme beschrieben haben, die von größeren Softwarehäusern erzeugt wurden, kommt wieder ein "Qualitätsprodukt" aus unserem Club an die Reihe. Zwei begabte Assembler-spezialisten haben nämlich einen Assembler in Modulform erzeugt.

Der Assembler wird wie ein normales Modul in den Modulschacht gesteckt und meldet sich nach dem Einschalten mit einem Menü. Beim Einstecken des Moduls sollte man darauf achten, daß die Platine richtig paßt und daß sie nicht verkehrt ist. Entsprechende Hinweise sind mitgeliefert. Anhand des Menüs werden wir die einzelnen Features des Moduls besprechen:

Assembler/Editor  
(C) 1985 by A. Holy & G. Wolfbauer

MENU

```
S save a program
L load a program
A save source file
B load source file
G start a program
C create source file
P print source file
H hexdump + editor
? MENU
```

Das Wichtigste ist der eigentliche Assembler. Man startet ihn mit C. Danach verfügt man über einen komfortablen Bildschirmediator, mit dem man den Sourcetext schreiben kann. Am unteren Bildschirmrand werden die jeweilige Zeichen- und Zeilenposition und der noch verfügbare freie Speicherplatz angezeigt. Die meisten Sondertasten funktionieren, CTRL-N fügt eine Zeile ein, CTRL-Y löscht eine Zeile. Die Funktionstasten sind auch belegt:

F1	F2	F3	F4	F5	
PUSH	POP	LD	JP	CALL	ohne SHIFT
JR	CP	DEC	INC	ASSEMBLE	mit SHIFT

Dies ermöglicht ein sehr komfortables Arbeiten. Mit F10 kann man den Assembler starten, doch davon später.

Der Assembler hält sich in der Syntax sehr stark an die Normmnemonics, wer das Buch "Programmieren in Assembler" von Rodney Zaks hat, kennt so gut wie alle Befehle des Prozessors. Zusätzlich hat das Modul noch einige andere Features:

```
ORG adr : Programmbeginn, normalerweise auch die Position des MC-Codes im Speicher.
CODE adr : ändert die Position des Programms im Speicher, obwohl der Programmbeginn gleich bleibt.
EQU wert : weist einem Label einen expliziten Wert zu.
D(EF)B : speichert explizite Werte in die jeweiligen Speicherzellen,
```

```
die Werte stehen als Argumentliste hinter dem D(EF)B.
D(EF)W : speichert Wörter (etwa Adressen oder sonstige 16-Bitdaten) in die jeweiligen Speicherzellen, Übergabe der Parameter wie D(EF)B.
D(EF)S wert: reserviert ein Feld von 'wert' Nullen im Speicher.
DEFM string: speichert einen ASCII-String im Speicher.
```

Wir wollen hier nicht näher auf die einzelnen Befehle eingehen. Makros sind nicht möglich.

Das Assemblieren geht ebenfalls sehr bequem vor sich. Durch Drücken von F10 wird assembliert, tritt ein Fehler auf, wird eine Fehlermeldung ausgegeben (nicht nur eine Zahl!). Durch Betätigen von ESC wird der Cursor in besagte Zeile gebracht. Das Assemblieren wird sehr schnell erledigt, MC-Codeprogramme unter einem KByte sind ohne nennenswerte Wartezeiten im Speicher. Am Ende des Compilervorganges wird die Endadresse des Programmes angegeben.

Durch Betätigen der ESC-Taste, wenn kein Fehler aufgetreten ist, kann man jederzeit in das Hauptmenü zurückgelangen.

Die zweite Hauptfunktion ist das Editieren per Hexdump. Durch Drücken von "H" befindet man sich in einem - ebenfalls durch einen komfortablen Bildschirmediator unterstützten - Hexdump mit ASCII-Anzeige. Man kann nun die Hexcodes ändern. Durch ESC steigt man wieder aus. Durch Bildschirmscrolling kann man beim Hexdump beliebige Speicherbereiche ansehen.

Dritter wichtiger Punkt beim Bearbeiten von MC-Codeprogrammen ist das Laden und Speichern. Bei diesem Modul ist man in der Lage, sowohl das eigentliche Programm als auch das Sourcefile zu laden und zu speichern. Allerdings funktioniert das nur auf Kassette. Disketten werden nicht anerkannt.

Da man manchmal auch Listings von Programmen haben möchte, kann man mit P das Sourcefile ausdrucken. Leider ist es nicht möglich, ein Hexdump vom Modul aus auf ein Druckerpapier zu bekommen, doch das ist ein vernachlässigbarer Mangel.

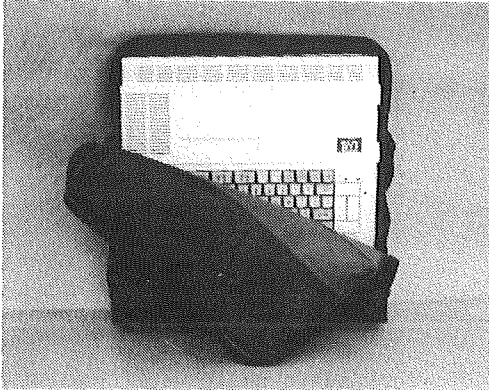
Zuletzt kann man das entwickelte Programm noch durch die Taste "G" laufen lassen. Mit "?" läßt sich das Menü aufrufen.

Der Assembler kann für jeden, der sich ernsthaft mit Maschinsprache beschäftigt, eine sehr große Hilfe sein. Der einzige Nachteil, den dieses Modul besitzt, ist sein Preis. 999 Schilling sind für ein Programm doch recht viel, obwohl dieses sehr viel leistet. Wie der Autor dieses Berichts hörte, kursieren Gerüchte über eine Version, die von Kassette geladen wird. Das Modul würde wegfallen, aber man hätte auch weniger Speicherplatz zur Verfügung. Vielleicht sollte man doch diese billigere Variante forcieren, dann wäre diesem Modul ein sehr hoher Absatz gewiß.

Bekommen kann man das Modul über Georg Wolfbauer in unserem Clublokal. Es ist auf dem SVI-318 und SVI-328 lauffähig.

**Spectravideo SVI-328**

S 5.990,-

**MSX-Computer SVI-728**

S 5.990,-

**SVI-738 X'press**

mit eingebauter 80 Zeichen-Karte  
mit eingebautem 3 1/2"-Diskettenlaufwerk  
mit Centronics-Schnittstelle  
mit RS-232 C-Schnittstelle  
mit Anschluß für zweite Diskettenstation  
mit MSX-BASIC und MSX-DOS  
mit CP/M-Betriebssystem

S 12.990,-

An den Einkaufssamstagen von 9 bis 18h geöffnet

# Computer-Studio

PANIGLGASSE 18 · A-1040 WIEN · TEL.(0222) 65 88 93

**IMPRESSUM:**

Chefredakteur: Gerhard Fally

Ständige freie Mitarbeiter: Constantin Gagnas, Stefan Kovalovsky, Philipp Ott, Rafael Razim, Wolfgang Rotschek, Christoph Sator, Heinz Schmid, Stephan Traxler, Georg Wolfbauer

Medieninhaber (Verleger): Spectra Video Club Austria, p.A. Computer-Studio, A-1040 Wien, Paniglgasse 18-20, Tel (0222) 65 88 93

Hersteller: HTU-Wirtschaftsbetriebe Ges. m. b. H., 1040 Wien

Herausgeber: Spectra Video Club Austria, p.A. Computer-Studio, A-1040 Wien, Paniglgasse 18-20, Tel. 65 88 93

Erscheinungsweise: monatlich, jeweils zur Monatsmitte, Einzelheft S 15,-

Abonnementpreise:  
jährlich S 150,-  
halbjährlich S 80,-

Erscheinungsort Wien  
Verlagspostamt 1040 Wien

Alle im SVI-Journal abgedruckten Beiträge sind urheberrechtlich geschützt. Nachdruck, Vervielfältigung und Übersetzung sind nur mit schriftlicher Erlaubnis der Redaktion gestattet. Für die Richtigkeit der Beiträge wird keine Haftung übernommen, die Redaktion übernimmt keinerlei Verantwortung für die Verletzung von Patentrechten.

**Spectra Video Club  
Austria**

Was bietet Ihnen der Spectra Video Club Austria?

- regelmäßige Clubabende mit Gelegenheit zum Informationsaustausch
- Möglichkeit zum kostenlosen Arbeiten an SVI-Computern während der Clubtreffen und zum Ausdrucken von Programmlistings
- außerordentliche Clubabende mit Vorträgen über Themen rund um Hard- und Software der SVI-Computer und über MSX
- kostenloser Bezug der monatlich erscheinenden Clubzeitschrift SVI-JOURNAL
- verbilligter Einkauf von Spectravideo-, MSX- und Bondwell-Produkten
- Besitzer von MSX-Computern sind ebenfalls willkommen.

Mitgliedsbeitrag: Jahresbeitrag S 500,-  
für Schüler, Studenten, Lehrlinge S 250,-

Für neue Mitglieder gilt der Mitgliedsbeitrag ab sofort bereits für 1986.

Nähere Informationen beim

Spectra Video Club Austria  
c/o Computer-Studio  
1040 Wien, Paniglgasse 18-20  
Telefon (0222) 65 88 93