

# SV!

# JOURNAL

Die Zeitschrift des Spectra-Video-Club Austria



**Heft 5/85**

**MSX**

**S 15.-**

**Lieber SVI-Journal-Leser!**

Manch treuer Leser unseres Journals wird sich vielleicht gewundert haben, daß auf dem Titelblatt dieser Ausgabe ein "fremder" Computer abgebildet ist. Werden wir vom SVI-Journal Spectravideo untreu?

Bestimmt nicht! Bondwell ist nämlich gar nicht so "fremd", wie man im ersten Moment annehmen könnte. Spectravideo und Bondwell sind zwei Schwesterfirmen, Bondwell sogar die größere. Wir glauben, daß Bondwell ebenso als Clubcomputer angesehen werden kann, wie ein Spectravideo. Aus diesem Grund haben wir ihn auch gleich einmal getestet. Natürlich freuen wir uns von nun an, wenn auch Bondwell-Besitzer dem SVCA beitreten und uns Berichte oder schöne Programme schicken.

Da wir schon bei transportablen Computern sind: Es gab schon lange das Gerücht, daß ein Spectravideo-Portable auch auf den Markt kommen soll. In Deutschland ist dieser geheimnisvolle Computer nun gesichtet worden, in der BRD gibt es ihn also schon. Erste Fotos kann man in deutschen Zeitschriften und Büchern auftreiben.

Neben diesen Neuigkeiten gibt es wieder einiges aus dem Clubgeschehen zu berichten. Zusammenfassend werden wir auch einiges über die "ifabo" berichten, die ja am 11. Mai zu Ende ging.

Um den Clubmitgliedern wieder mehr Möglichkeiten zu geben, sich untereinander zu verständigen, möchten wir eine Rubrik einrichten, die Fragen beantwortet. Ab nun können (ernstgemeinte) Fragen an uns eingesandt werden, und wir werden versuchen, sie zu beantworten. Sollte es uns nicht gelingen, so werden wir die Frage veröffentlicht und alle anderen Clubmitglieder bitten, das Problem zu lösen.

Am Montag, dem 20. Mai 1985, fand unsere erste Vereinssitzung statt. Trotz der 116 Mitglieder, die unser Club zählt, kamen nur 16 ins Cafe Landtmann. Dies war auf der einen Seite natürlich gut, da die Anwesenden im kleinen Kreis nicht über den ganzen gemieteten Saal rufen mußten, um etwas zu sagen. Auf der anderen Seite betrübt es natürlich den nun gewählten Vorstand, daß so wenig Interesse an vereintechnischen Vorgängen herrscht. Außerdem fanden sich sehr wenige Kandidaten für die einzelnen Posten. Es reichte gerade aus, um für jeden Posten

```

*****
*
* INHALT
* 2 Clubnachrichten
* 3 Spectravideo-BASIC
* 6 Z 80 - Programmieren in Assembler
* 9 SVI-Hardware
* 10 Turbo-Pascal, von Anfang an
* 12 Bondwell-Testbericht
* 14 Bericht von der Gründungsversammlung
* 15 Test: CP/M-Handbuch von Rodney Zaks
* 16 Nachlese: ifabo 1985
* 17 3D-Graphik-Nachlese Folge 2
* 19 Ein- und Ausgabe von
* Peripheriebausteinen
* 20 MSX-Seite
* MSX-Disk und 80-Zeichenkarte
* 22 Programmecke
* Graphik-Erweiterung
*
*****
    
```

```

*****
*
* Die nächsten Clubtermine:
*
* Sa, dem 1. Juni 1985, ab 17 Uhr
* Mi, dem 5. Juni 1985, ab 19 Uhr
* Di, dem 11. Juni 1985, ab 19 Uhr
* Sa, dem 22. Juni 1985, ab 17 Uhr
* Mi, dem 26. Juni 1985, ab 19 Uhr
* Di, dem 2. Juli 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 Programmlistings.
*
* Telefonische Auskünfte über den Club
* und seine Aktivitäten erhalten Sie
* unter der Telefonnummer 65 88 93.
*
*****
    
```

einen Kandidaten zu finden. Bei den Rechnungsprüfern konnten diese sogar erst bei der Vereinssitzung zur Aufstellung bewegt werden. Aber nun sind alle Positionen vergeben, der Club hat seinen Vorstand und in nächster Zeit werden einige Themen behandelt werden müssen.

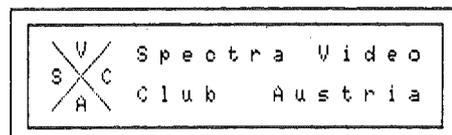
Näheres über diese Sitzung erfahren Sie im Inneren der Zeitschrift. Der neugewählte Clubobmann, Wolfgang Rotschek, hat einen detaillierten Bericht darüber geschrieben.

An besagtem Montag fiel auch die Rede auf das Clubemblem. In den ersten Clubtagen hatten wir ja kurz ein Vereinszeichen, das jedoch bald in Vergessenheit geriet. Nun möchten wir alle Clubmitglieder aufrufen, uns Vorschläge für Embleme zu schicken oder in der Redaktion vorbeizubringen. Wir wollen nämlich gemeinsam mit dem Vorstand und Ihnen, verehrter Leser, dem Club wieder ein ordentliches Zeichen verehren.

Quasi als Gedächtnisstütze zeigen wir Ihnen unser altes Clubemblem, das mehr an einen Ritterorden als an einen Computerclub erinnert. Sie brauchen sich jedoch nicht an diese Vorlage halten, im Gegenteil, eigenständige "Produktionen" sind uns lieber.

Im Übrigen bitten wir alle MSX-Freunde, bei uns aktiv mitzuwirken. Leider haben wir nämlich noch sehr wenig Kontakte zu Personen, die schon nette Programme geschrieben oder das Betriebssystem ihres Computers "umgekrempelt" haben. Natürlich freuen wir uns auch über Bediener anderer MSX-Computer, die bei uns mitarbeiten wollen.

Ihr SVI-Journal-Chefredakteur Gerhard Fally!



Diesmal werden die Ein-/Ausgabebefehle vervollständigt. Danach setzen wir mit den Graphikbefehlen fort. Vorerst veröffentlichen wir aber noch eine ganz neue Meldung über den PLAY-Befehl.

Mit dem PLAY-Befehl kann man nicht nur schön musizieren, es ist auch möglich, abzufragen, ob gerade musiziert wird. Es gibt nämlich PLAY nicht nur als Befehl, sondern auch als Funktion. Bei dieser Funktion kann man angeben, welcher Kanal auf seine momentane Tätigkeit abgefragt werden soll. "1" steht für Kanal A, "2" für Kanal B und "3" für Kanal C. Stellt man eine Null zwischen die Klammern hinter dem Befehlsword, werden alle drei Kanäle überprüft. Sobald ein Kanal "tätig" wird, ist das Ergebnis der Funktion "-1", also "wahr".

Beispiel:

```
10 PLAY "CDEFGFEDC" ' als Befehl
20 INPUT I:PRINT PLAY(I) ' als Funktion
```

Schon in der vorigen Folge erwähnt wurde der SOUND-Befehl. Wir brauchen ihn zur Erzeugung von Tönen oder Geräuschen. Er ist ja bekanntlich nicht so komfortabel wie der PLAY-Befehl, aber da er fast direkt in den Tongenerator schreibt, eignet er sich zum Experimentieren mit der Musik besser. Angegeben wird immer ein Adreßbyte und ein Datenbyte. Sehen wir uns nun die einzelnen Register und ihre Wirkung genauer an:

Die ersten sechs Register sind für die Tonhöhen der drei Kanäle verantwortlich. Je zwei Register "sorgen" sich um einen Kanal. Der Tonkanal A wird so vom Register 0 (niederwertiges Byte) und vom Register 1 (höherwertiges Byte) beeinflusst. Analog verhalten sich die Kanäle B und C zu den Registern 2,3,4 und 5.

Der Wert, der in die beiden Speicherzellen geschrieben wird, ist nicht direkt die Frequenz, sondern verhält sich indirekt proportional dazu. Je höher die Frequenz, desto niedriger der Wert. Die genaue Umrechnungsformel lautet:

$$\text{Wert} = 3579545 / (\text{Frequenz} * 16)$$

Wenn man nun für 'Frequenz' 440 Hz einsetzt, kommt der Wert 508,... heraus. Diesen Wert muß man jetzt folgendermaßen ein speichern, um den Kammerton a' zu hören:

```
SOUND 0, 508/256:SOUND 1, 508 AND 255
```

Voraussetzung dafür ist allerdings, daß vorher schon der Kanal 1 aktiviert worden ist.

Register 6 stellt den Modus für den Rauschgenerator ein. Es gibt verschiedene "Rauscharten". Je nach dem, wie groß das Frequenzband des Rauschens ist, wird es "härter" oder "weicher". Beim SVI wird dies durch eine Zahl zwischen 0 und 31 aktiviert.

Im Register 7 befinden sich die Informationen, welcher Tonkanal aktiv und welcher nicht aktiv ist. Ebenso wird zwischen "Rauschen" und "Ton" unterschieden. In diesem

Register muß man sich den Inhalt daher in einzelne Bytes zerlegt denken:

```
Bit 0: Tonkanal A mit Ton
Bit 1: Tonkanal B mit Ton
Bit 2: Tonkanal C mit Ton
Bit 3: Tonkanal A mit Rauschen
Bit 4: Tonkanal B mit Rauschen
Bit 5: Tonkanal C mit Rauschen
Bit 6: vom BASIC unbenutzt
Bit 7: vom BASIC unbenutzt
```

Eine Funktion ist dann aktiv, wenn das dazugehörige Bit auf Null gesetzt ist. So aktiviert SOUND 7,254 (&B11111110) zum Beispiel den Tongenerator A.

Die Lautstärken der drei Kanäle werden durch die drei Register 8,9 und 10 geregelt. Hier gelten die gleichen Bedingungen, wie für 'V' im Playbefehl.

Die Register 11 und 12 beinhalten den Frequenzwert für die Periode der Hüllkurve. Wieder ist der Wert nicht gleich der Frequenz. Berechnen kann man ihn aus der Formel für 'M' vom PLAY-Befehl.

Register 13 enthält die ausgewählte Hüllkurvenform. Zusammen mit den Registern 11 und 12 lassen sich so Klangmanipulationen ausführen. Es braucht allerdings viel Zeit und Geduld, bis zum Beispiel ein Klavierklang gut nachgebildet ist.

Mit dem "SOUND 'Adresse', 'Datenbyte'"-Befehl hat SOUND ON/OFF nichts zu tun. Mit diesem Kommando läßt sich die Dokumentationsspur einer Kassette auf den Fernsehlautsprecher schalten oder wieder abstellen. Dieser Befehl funktioniert natürlich nur beim Stereorekorder SVI-903.

Das BASIC der SVI-Computer hat auch Bankswitching vorgesehen. Wenn eine 64KRAM-Zusatzkarte verwendet wird, läßt sich mit SWITCH zwischen den beiden Banken 0 und 3 schalten. Für SWITCH gelten in der Bedienung die gleichen Regeln wie für CMD-SWITCH, die Befehlserweiterung, welche in einer der vorigen Ausgaben näher erklärt wurde. Einziger Unterschied ist, daß nicht zwischen Page 02 und 21 sondern zwischen Page 02 und 32 gewechselt wird.

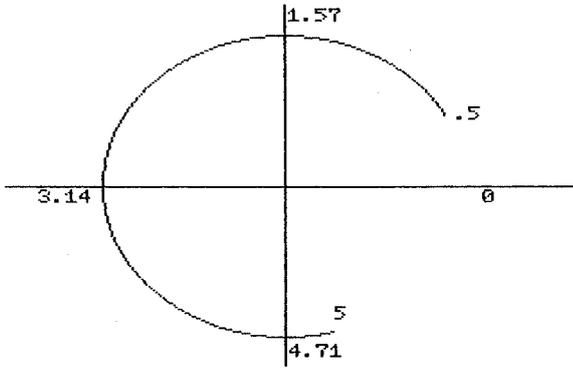
Im normalen Betrieb stellt der Computer 39 Zeichen pro Zeile am Bildschirm dar. Mit WIDTH 40 lassen sich ebenso 40 Zeichen darstellen. Ist eine 80-Zeichenkarte installiert, dürfen sogar 80 Zeichen eingestellt werden. Nach jedem WIDTH wird der Bildschirm gelöscht. Variablen sind hinter dem Befehlsword nicht erlaubt.

```
Beispiele: WIDTH 39
           WIDTH 40
           WIDTH 80
```

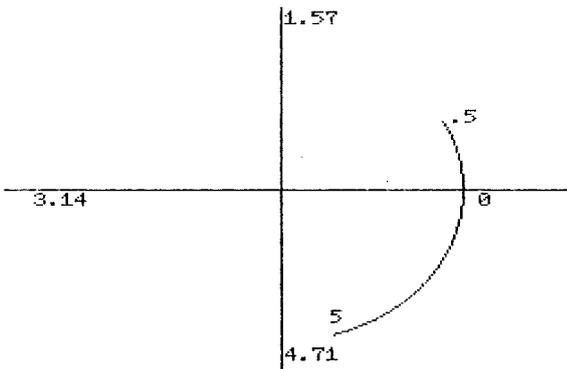
Nun wollen wir die I/O hinter uns lassen und widmen uns ganz einem neuen Kapitel, den Graphikanweisungen.

Die SVI-Computer besitzen einen Befehl, um den sie viele andere Systeme "beneiden", die CIRCLE-Anweisung. Mit diesem Kommando lassen sich wunderbar Kreise und Ellipsen erzeugen. Bis zu sieben Parameter dürfen angegeben werden. Als erstes kommen die zwei Koordinaten des Mittelpunktes in einer Klammer,

danach der Radius des Kreises. Diese drei Werte sind "Pflicht". Zusätzlich darf man natürlich auch die Farbe des Kreises bestimmen, tut man es nicht ausdrücklich, wird die eingestellte Vordergrundfarbe genommen. Will man keinen ganzen Kreis, sondern nur einen Bogen darstellen, so kann man mit den beiden nächsten Parametern den Anfangs- und den Endpunkt bestimmen. Dazu gibt man zwei Winkel in Bogenmaß an, die gegen den Uhrzeigersinn gerichtet sind. Die beiden Graphiken unterhalb zeigen die Arbeitsweise dieser Winkelmessung.



CIRCLE (128,96),80,,.5,5



CIRCLE (128,96),80,,5,.5

Wenn man für den Anfangspunkt einen negativen Winkel einsetzt, wird eine Verbindungslinie vom Mittelpunkt zu dem Ende des Kreisbogens gezeichnet. Der Anfangswert wird dann auf Null gesetzt. Beim Endpunkt entsteht dieser Effekt nicht. In diesem Fall ersetzt der Computer nur den Endwert durch Null, zieht aber keine Linie.

Zuletzt kann man auch noch ein Verzerrungsverhältnis angeben. Dabei wird zwischen Werten größer Eins und kleiner Eins unterschieden. Bei den größeren Werten bleibt der eingestellte Radius in der Höhe (y-Richtung) gleich, die x-Ausdehnung wird gemäß der Formel 1:Wert verkleinert. Je größer der Wert ist, desto kleiner wird daher die Waagrechte der Ellipse. Bei Werten kleiner Eins wird die x-Achse gleichgelassen und der y-Wert verkleinert. Von negativen Werten wird der Absolutbetrag genommen, "-2" ist daher im Verzerrungsverhältnis gleich mit "2". Dieses Verhalten stimmt übrigens auch geometrisch, ist also nicht vom Interpreter ge-

künstelt worden. Selbstverständlich kann man auch Ellipsenbögen erzeugen, indem man einen Anfangs- und einen Endpunkt angibt.

Wenn hinter das Befehlswort CIRCLE ein STEP eingefügt wird, interpretiert der Computer die angegebenen Koordinaten als relativ und addiert sie zu den letzten verwendeten Koordinaten dazu. Diese Koordinaten müssen allerdings nicht im Befehl vorkommen, bei einem DRAW wird das Ganze indirekt bestimmt. Ein LOCATE oder PRINT gilt in dem Fall übrigens nicht als Koordinate, nur Befehle wie DRAW, LINE, PSET.

Damit man seine Graphiken nicht nur in einem dezenten Schwarz-Weiß bewundern kann, gibt es eine komfortable Einstellung für die Grundfarben. Im Befehl COLOR dürfen drei Parameter angegeben werden, Vordergrund, Hintergrund und Rand.

Betrachten wir vorerst nur den Textmodus. Sobald man ein COLOR ausführen läßt, wird der Bildschirm auf die neuen Farben aktualisiert. Man kann zwar alle drei Parameter angeben, doch der Rand bleibt in der gleichen Farbe, wie für den Hintergrund eingestellt wurde. Erst im Graphikschirm wird der Rand mit einer eigenständigen Farbe erkennbar.

In der Graphik verhält sich der Interpreter etwas anders. Der Rand kann eigenständig definiert werden. Farbänderungen werden nur am Rand unmittelbar nach dem Befehl ausgeführt. Vordergrund und Hintergrund machen sich erst nach einem neuerlichen SCREEN-Kommando oder sonstigen "Bildschirm löschen"-Befehlen bemerkbar. Man kann daher keinen Farbenwechsel durchführen und gleichzeitig die Graphik am Schirm lassen. Es gibt allerdings eine Ausnahme. Wenn man am Anfang den Hintergrund als 0 (Transparent) definiert, nimmt der Hintergrund die Farbe des Randes an. Da der Rand nach einem Farbenwechsel unmittelbar seine Farbe ändert, wechselt auch der Hintergrund seine Farbe. Das Gleiche gilt für den Vordergrund.

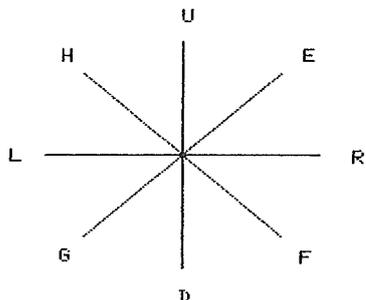
Zum Abschluß des Befehles wird noch die Farbtabelle abgedruckt, um den Farbcodes auch die richtigen Farben zuzuordnen zu können.

0	transparent	8	mittelrot
1	schwarz	9	hellrot
2	mittelgrün	10	dunkelgelb
3	hellgrün	11	hellgelb
4	dunkelblau	12	dunkelgrün
5	hellblau	13	magenta
6	dunkelrot	14	grau
7	cyan	15	weiß

In vielen Videospiele braucht man Landschaften im Hintergrund. Diese haben meistens "zerfranste" Berge oder Küstenabschnitte als Abschluß. Diese Landschaften mit LINE und CIRCLE nachzubilden, wäre heller Wahnsinn. Dafür gibt es einen eigenen Befehl, DRAW. DRAW enthält eine eigene kleine Makro-Sprache, die einer Turtle-Graphik ähnelt. Durch einzelne Buchstaben wird die Richtung des Cursors bestimmt, durch nachfolgende Zahlen die Länge seiner Bewegung festgelegt. Mit diesem System lassen sich ohne großen Aufwand schöne Kulissen herstellen. Man kann zwar nur gerade oder im Winkel von 45° Linien ziehen, aber meistens ist ohnehin nicht mehr verlangt. Sehen wir uns nun die einzelnen Buchstaben und ihre Bedeutung an:

Leicht zu merken sind die 4 Hauptachsenrichtungen. "U" (wie Up) für hinauf, "D" (wie Down) für herunter, "L" (wie Left) für links und "R" (wie Right) für rechts. Etwas schwerer wird es für die schrägen Linien.

Sie sind mit "E", "F", "G" und "H" gekennzeichnet. "E" zeigt nach rechts oben, alle weiteren Buchstaben stehen im Uhrzeigersinn für jeweils um 90° weitergedrehte Linien. So ergibt sich das allseits bekannte untenstehende Diagramm der 8 Richtungen.



Hinter jedem Buchstaben gibt eine Zahl über die Länge der Cursorbewegung Auskunft.

Einen Ausweg gibt es natürlich, um den 45° zu entweichen. Mit Mx,y wird ein Punkt angegeben, der als Endpunkt einer Linie fungiert. Den Anfangspunkt bildet das zuletzt angesprochene Pixel. Wenn vor den Koordinaten x und y Vorzeichen stehen, werden diese als relative Koordinaten angesehen und zum derzeitigen Positionsstand addiert (bei einem "-" natürlich subtrahiert).

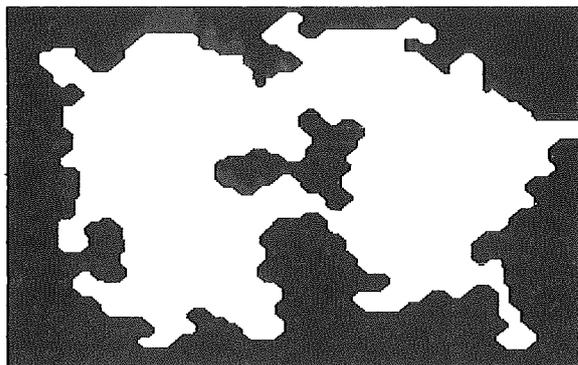
Weiters gibt es noch zwei sehr nützliche Kommandos, "B" und "N". Diese beiden Buchstaben werden vor die Zeichen gestellt, die sie beeinflussen sollen. B bewegt den Cursor weiter, zeichnet aber nicht. N bewegt den Cursor nach der Bewegung wieder zum Anfangspunkt der Bewegung zurück. Diese Zeichen gelten allerdings nur für jeweils ein Makrokommando.

Weniger gebraucht werden im Normalfall "A" und "S". A ist zwar am 328 vorgesehen, funktioniert aber nicht. Erst am 728 lassen sich Figuren drehen. Hinter A darf als n entweder 0,1,2 oder 3 stehen. Je nach dem Wert wird ein Winkel eingestellt, der das Gebilde um 90\*n Grad dreht. A bleibt auch nach CTRL-STOP und RUN weiter aktiv, erst durch ein DRAW "A0" wird wieder der normale Zustand erreicht. Mit S kann man Figuren vergrößern oder verkleinern. Dabei gibt man einen Wert an, der ganzzahlig sein muß und zwischen 0

und 255 differiert. Bei Null und Vier wird das Objekt in Originalgröße angegeben. Der Faktor hinter S wird nämlich durch 4 dividiert und dann erst mit den Längen des Objektes multipliziert.

Mit C ändert man die Farbe der gezeichneten Linien. Variablen dürfen natürlich auch verwendet werden. Durch "=Variable" lassen sich numerische Werte ersetzen, "XVariable;" ist für Strings gültig.

Zum Schluß zeigen wir noch, wie eine Graphik mit DRAW aussehen kann. Unter dem Bild kann man die für die Graphik notwendigen Befehle sehen. Außer zwei PAINT-Befehlen und einer Warteschleife ist alles vorhanden.



```
1 SCREEN1:PSET (255,60):DRAW"L20H2U3L2H5L3H7G
2L1H1U15H5L3G7D4L2H12L5U6R5F3R3E3U5H4L3H3L2G
6L30G5L4H3U4E2U2H3L4G12D2F4R3F4R2F5D1G4L6G3L
2G1D3G1L2H1U2H1U4H4U4H1L10H3L4H2U4H3L20G10D4
G7L6H8U4H1L8G2L3G2D5F6D3F4D1E3R2F4D2G5L3G1D1
0
2 DRAW"F4D8G5D3F2R4F2D13G2D8G2L3G2D13F3R7E4U
4H2U3E3R3E2R3F5D5F1D6G2D4F3D4G3L6H1L2H3U1H1L
3G5D4F5R4F4D3F4R14F5D3G2L3G2D1F3R10E7R4E2U3H
4U2E3R4F5R4F3R2F3D2F4R10E5U5H4U5E3U5H4L4H3U5
E4U5H3U4E6U1E2R9E3R5F4D4F3R2F3D5
3 DRAW"F5D3F5R7F2D3G4L9G3D2F6R14E5R4E6R5F4R3
E7R5F4D12F4D2G5D2F5R8E5U3H7U5H5U8H2U8H1U3H1L
4G3L2H3U8E6R6E4U3E5R4E2U4H4U3E4R5E4U2H3U6E5R
13
4 PSET (150,96):DRAW"U6E4U3H2U2E5U3E3U4H4L5G3
L2H5U1H3L2G4D5F6D3G3D4G3L5H4L3H3L5G4L7G2L2G2
D8F4R4F4R4E5R3E6R3F5D2F3R4F4D3F2R4E5U2H4
```

In der nächsten Folge werden weiter die Graphikanweisungen "aufs Korn genommen".

\*\*\*\*\*

Als vor einigen Jahren ein langjähriger Delegierter des Internationalen Roten Kreuzes gefragt wurde, welche drei Hilfsgüter für die Soforthilfe - im Katastrophenfall und bei bewaffneten Konflikten - am wichtigsten seien, gab er zur Antwort:

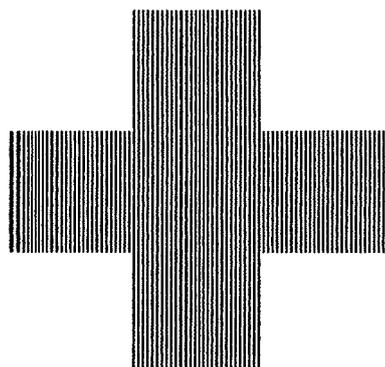
1. Decken, 2. Decken und 3. Decken.

Unterstützen auch Sie die Deckenaktion des ÖJRK!

Wir brauchen Decken, die folgendermaßen beschaffen sind: Länge 180 cm, Breite 130 cm, Gewicht 1.7 kg, Material: Schurwolle. Diese Decken können unter anderem auch beim Zentraleinkauf des ÖJRK, 1020, Wien, Nordportalstraße 248, erworben werden.

Weshalb gerade eine Decke mit einer so genauen Beschreibung?

1. Nur eine möglichst einheitliche Decke kann platzsparend und damit billig und schnell verschickt werden.
2. Die Erfahrung hat immer wieder gezeigt, daß unterschiedliche Decken immer wieder Anlaß zu Auseinandersetzungen bei der Verteilung geben.



Assemblerfortsetzungskurs Nr.:011

Langsam aber doch nähern wir uns dem Ende des Assemblerfortsetzungskurses. Es gibt nicht mehr allzuviele Befehle, die nicht erklärt wurden. Doch eine wichtige Befehlsgruppe - die Rotier- und Schiebepfehle - müssen noch erklärt werden. Zuerst wollen wir den Unterschied zwischen schieben und rotieren erläutern. Beim Schieben wird der Inhalt eines Registers um ein Bit nach links oder nach rechts verschoben, wobei das äußerste Bit hinausfällt und das Byte auf der anderen Seite mit Null aufgefüllt wird.

Da man das Schieben auch zum Dividieren verwenden kann, gibt es eine Ausnahme. Wenn man eine Zahl im Zweierkomplement hat, so ist das linke Bit das Vorzeichenbit. Nach der Division soll das Ergebnis aber wieder negativ sein. Beim Befehl SRA (Schiebe rechts arithmetisch) hat das Bit, das links hereinkommt, den Wert, den vorher das Vorzeichenbit hatte.

Doch nun zum Rotieren. Beim Rotieren wird entweder das hinausfallende Bit hineingeschoben, beziehungsweise wird das Übertragsbit hineingeschoben. Es gibt zwei Arten von Rotationen. Bei einer werden insgesamt acht Bit rotiert, bei der anderen neun Bit, wobei das Übertragsflag dabei das zusätzliche Bit darstellt.

Bei der Acht-Bit-Rotation wird das herausfallende Bit auf der anderen Seite nachgeschoben und gleichzeitig ins Übertragsflag kopiert.

Bei der Neun-Bit-Rotation wird das herausfallende Bit ins Carryflag rotiert und der alte Wert des Carryflags ins Register nachgeschoben.

Weiters existieren noch Rotierbefehle, die das Rotieren von BCD-Zahlen ermöglichen. RLD (Rotiere links dezimal) bewirkt, daß die vier unteren Bits der Speicherzelle, die durch HL adressiert wird, in die oberen vier Bits verschoben werden. Das obere Nibble der Speicherzelle kommt in das untere Nibble des Akkumulators. Das untere Nibble des Akkumulators wandert in das untere Nibble der Speicherzelle und das obere Nibble des Akkumulators bleibt unverändert.

RRD (Rotiere rechts dezimal) wirkt genau umgekehrt. Die vier oberen Bits der Speicherzelle, die durch den Inhalt von HL adressiert wird, werden in die unteren Bits des gleichen Bytes verschoben. Die vier unteren Bits kommen in die vier unteren Bits des Akkumulators. Die unteren Bits des Akkumulators kommen in die vier oberen Bits der festgelegten Speicherzelle. Alle diese Operationen werden gleichzeitig ausgeführt. Die Bits 4 bis 7 des Akkumulators bleiben unverändert.

Es gibt nun folgende Rotier- und Schiebepfehle. (s bedeutet ein beliebiges 8-Bit-Register oder (HL), (IX+d) oder (IY+d))

- RL s -Rotiere links durch das Carryflag.
- RLA -Rotiere Akku links durch das Carryflag.
- RLC s -Rotiere links ohne Carryflag.
- RLD -Rotiere links dezimal.
- RR s -Rotiere rechts durch das Carryflag.

RRA -Rotiere Akku rechts durch das Carryflag.

- SLA s -Schiebe links arithmetisch ins Carry.
- SRA s -Schiebe rechts arithmetisch ins Carry (Bit sieben bleibt gleich!).
- SRL s -Schiebe rechts logisch ins Carry.

Nun noch einige spezielle Befehle, die die Interrupts betreffen:

- HALT: Die CPU bleibt so lange stehen, bis ein Interrupt auftritt.
- DI: Sperrt alle Interrupts.
- EI: Gibt die Interrupts wieder frei.
- IM 0: Schalte in den Interrupt Modus Null.
- IM 1: -- " -- Eins.
- IM 2: -- " -- Zwei.

Doch was sind Interrupts (engl.: Unterbrechungen)? Die Interrupts werden hardwaremäßig erzeugt, das heißt, nicht die CPU erzeugt Interrupts, sondern externe Bausteine. Diese externen Bausteine senden der CPU ein spezielles Signal, das die CPU dazu veranlaßt, ihre Programmausführung zu unterbrechen und an eine bestimmte oder vom Benutzer definierte Adresse zu verzweigen.

Ein einfaches Beispiel für einen Interrupt: An einem Z80 hängen 10 Temperaturmesser in zehn verschiedenen Räumen, um die Temperatur zu speichern und um zu erkennen, ob Feuer ausgebrochen ist. Die CPU fragt nun der Reihe nach alle Temperaturmesser ab und registriert die Temperatur. Fängt es nun aber in einem Raum zu brennen an, so kann der Temperaturmesser nicht warten, bis er an die Reihe kommt, den bis dahin ist der Raum schon ausgebrannt. In so einem Fall klopft der Temperaturmesser der CPU freundlich auf den Rücken und gibt ihr zu erkennen, daß ihm langsam heiß wird. Nun klopft der externe Baustein der CPU nicht auf den Rücken, sondern er schickt einen Interrupt, worauf die CPU ihrerseits in eine Interruptroutine verzweigt, wo dann der Baustein, der den Interrupt gesendet hat, ermittelt werden kann.

Der Z80 kennt insgesamt drei Arten von Interruptmodi.

Interrupt Modus 0:

Dieser wurde vom 8080-Prozessor übernommen und wird nur am Ende eines Befehles erkannt. Dann wartet der Z80, daß das externe Gerät während des nächsten Zyklus einen Befehl auf den Datenbus legt. Normalerweise ist das ein RST oder ein CALL, damit der PC gerettet wird. Die Routine muß mit dem Befehl RETI (return from interrupt) enden, der noch den Zustand einiger Flip-Flops richtig stellt.

Interrupt Modus 1:

Tritt nun ein Interrupt auf, so wird automatisch ein CALL zur Adresse 38h ausgeführt. Beim Spectravideo treten nur in diesem Modus Interrupts auf, die vom Videochip gesendet werden (und zwar alle 20 Millisekunden). Bei der Adresse 38h steht dann die Interruptroutine, welche die Tastatur einliest, auf Spritkollision überprüft und noch einiges mehr macht.

Interrupt Modus 2:

Dieser Modus wird auch der Vektorinterrupt Modus genannt, da dabei eine Vektortabelle verwendet wird. Jedes externe Gerät liefert dabei eine ein Byte lange Information, wobei immer das nullte Bit auf Null gesetzt wird. Hier wird nun das Interruptvektorregister verwendet. Der Inhalt des Interruptvektorregisters wird mit der erhaltenen Information zu einer 16-Bit-Adresse zusammengesetzt, wobei der Inhalt des Interruptvektorregisters die Bits 8-15 und die erhaltene Information die Bits 0-7 bilden. Die zwei Bytes, die bei der soeben erhaltenen Adresse stehen, bilden die Adresse, zu der die CPU verzweigt.

Es gibt jedoch noch einige Arten von Interrupts. Als erstes der

Bus Request:

Hierbei wird nach Abarbeiten eines laufenden Maschinenzyklus der Datenbus und der Adreßbus auf hochohmigen Zustand gelegt, damit das externe Gerät auf den Speicher, den die Z-80 verwaltet, zugreifen kann. Dieser Vorgang wird übrigens DMA genannt, direct memory access, was übersetzt so viel wie direkter Speicherzugriff bedeutet.

Der Nichtmaskierbare Interrupt:

Dieser Interrupt kann nicht durch ein DI (disable interrupt) unterdrückt werden. Der NMI, so die Abkürzung des Interrupts, wird nach Abarbeitung eines Befehles erkannt und daraufhin zur Adresse 66h verzweigt.

Der normale Interrupt:

Dieser Interrupt kann in einem der drei oben erwähnten Interruptmodi arbeiten und mit DI gesperrt werden.

Doch jetzt zu dem in der letzten Ausgabe versprochenen Stringsuche-Programm.

```

10 REM          ORG C000H
11 ' ILLEG: Einsprung für die Fehlermeldung
"illegal function call"
12 ' DRUCK: String wird auf dem Bildschirm
ausgegeben, Anfangsadresse in HL.
13 ' BLOBEG: zeigt auf Anfang von Speicher,
der durchsucht wird.
14 ' BLOLEN: Länge des Speichers, welcher
durchsucht wird.
15 REM ILLEG EQU 0F9EH
16 REM DRUCK EQU 697DH
17 REM BLOBEG EQU 9000H
18 REM BLOLEN EQU FFH
19 REM          CP 3
20 REM          JP NZ,ILLEG
21 REM          INC HL
22 REM          INC HL
23 REM          LD E,(HL)
24 REM          INC HL
25 REM          LD D,(HL)
26 REM          EX DE,HL
27 REM          EX AF,AF'
28 REM          LD A,(HL)
29 REM          EX AF,AF'
30 REM          INC HL
31 REM          LD E,(HL)
32 REM          INC HL
33 REM          LD D,(HL)
34 REM          LD HL,BLOBEG
35 REM          LD BC,BLOLEN
36 REM WEITER LD A,(DE)
37 REM          CPIR
38 REM          JR NZ,ERROR0
39 REM          PUSH HL
40 REM          PUSH BC
41 REM          PUSH DE

```

```

42 REM LOOP    EX AF,AF'
43 REM        DEC A
44 REM        JR Z,END
45 REM        EX AF,AF'
46 REM        INC DE
47 REM        LD A,(DE)
48 REM        CPI
49 REM        JR Z,LOOP
50 REM        POP DE
51 REM        POP BC
52 REM        POP HL
53 REM        JR WEITER
54 REM ERROR0 LD HL,ERR1
55 REM        CALL DRUCK
56 REM        LD HL,ERR2
57 REM        CALL DRUCK
58 REM        RET
59 REM ERR1   DEFM "STRING NOT FOUND"
60 REM ERR2   DEFB 07,13,0
61 REM ADDRESS DEC DE
62 REM        LD HL,TEMP
63 REM        CALL ZEICH
64 REM        LD D,E
65 REM ZEICH LD (HL),D
66 REM        RLD
67 REM        CALL SIGN
68 REM        RLD
69 REM        CALL SIGN
70 REM        RET
71 REM SIGN   AND 15
72 REM        ADD A,030H
73 REM        CP 040H
74 REM        JR C,OUT
75 REM        ADD A,7
76 REM OUT    RST 18H
77 REM        RET
78 REM END    POP DE
79 REM        POP HL
80 REM        POP DE
81 REM        CALL ADDRESS
82 REM        LD B,3
83 REM SPACE LD A,020H
84 REM        RST 18H
85 REM        DJNZ SPACE
86 REM        RET
87 REM TEMP   DEFW 00
88 REM        END

```

```

C000 FE 03 C2 9E 0F 23 23 5E
C008 23 56 EB 08 7E 08 23 5E
C010 23 56 21 00 90 01 FF 00
C018 1A ED B1 20 13 E5 C5 D5
C020 08 3D 28 4C 08 13 1A ED
C028 A1 28 F5 D1 C1 E1 18 EB
C030 21 3D C0 CD 7D 69 21 4D
C038 C0 CD 7D 69 C9 53 54 52
C040 49 4E 47 20 4E 4F 54 20
C048 46 4F 55 4E 44 07 0D 00
C050 1B 21 7E C0 CD 58 C0 53
C058 72 ED 6F CD 64 C0 ED 6F
C060 CD 64 C0 C9 E6 0F C6 30
C068 FE 40 38 02 C6 07 DF C9
C070 D1 E1 D1 CD 50 C0 06 03
C078 3E 20 DF 10 FB C9 00 00

```

Das Programm wird mit PRINTUSR ('SUCHSTRING') aufgerufen. Danach gibt es entweder "illegal function call", "String not found" oder den gefundenen String samt Adresse aus. Die Parameter BLOBEG und BLOLEN können nach Belieben verändert werden, je nach dem, welchen Speicherbereich man durchsuchen möchte.

Am Anfang des Programms werden wieder einige Parameter festgelegt. ILLEG zeigt ins ROM und wird angesprungen, wenn die Fehlermeldung "illegal function call" ausgegeben werden soll. DRUCK ist ebenfalls eine ROM-Routine, die einen String ausgibt. Das Register muß vor dem Aufruf die Adresse des ersten Zeichens beinhalten. Das Ende des

Strings wird durch eine bestimmte Zeichenkombination (7,13,0) gekennzeichnet. Zwischen 7 und 13 kann man noch ein 10 (line-feed) einfügen, wenn man will. BLOBEG und BLOLEN sind zwei Zeiger, welche die Anfangsadresse des Speichers, der durchsucht wird, und seine Länge angeben.

Mit CP 3 beginnt das eigentliche Programm. Hier wird untersucht, ob der in USR übergebene Wert ein String ist. Andernfalls wird die Routine für die Meldung "Illegal function call" angesprungen. Danach wird der übergebene String im Speicher geortet. Das Programm bedient sich dabei der Informationen, die vom USR-Kommando mitgeliefert werden:

Im Register HL steht die BASIC-Akkumulator-Adresse.  
HL muß zweimal inkrementiert werden, um auf die Adresse zu zeigen, die ihrerseits wieder auf die Länge des Strings und seinen Anfang zeigt.  
DE wird mit der Adresse geladen, auf die HL momentan zeigt.  
HL wird aus DE geladen.  
Ein Register im Z80 wird mit der Länge des Strings geladen, danach HL inkrementiert.  
DE wird mit der Adresse geladen, auf die HL nun zeigt und enthält nun die Anfangsadresse des übergebenen Strings.

zum Beispiel:  
Nach USR ("2") steht in HL: d034h  
Im Speicher ist folgendes abgelegt:

d034h unwichtig  
d035h unwichtig  
d036h 28h  
d037h f5h

DE wird nun mit f528h geladen, im Speicher ab f528h steht:

f528h 6  
f529h 36h  
f52ah e4h

DE enthält nun die Anfangsadresse des Strings, ab e436h würde in unserem Fall der übergebene String stehen, der sechs Zeichen lang ist.

Nach dieser Prozedur, die bis Zeile 24 geht, werden HL und BC mit den Dimensionen des zu durchsuchenden Speichers geladen. HL enthält den Beginn, BC die Länge des Speichers.

Die Routine WEITER lädt das erste Zeichen des Suchstrings in das Register A. Danach wird der CPIR-Befehl verwendet. Sollte das erste Zeichen nicht gefunden werden, wird das ZERO-Flag auf Null gesetzt. Der Sprungbefehl nach dem CPIR prüft das Flag auf Null. Ist es Null, wird nach ERROR0 verzweigt. ERROR0 gibt die Meldung "String not found" aus. Sollte das erste Zeichen jedoch gefunden werden, prüft der Computer die nächsten Zeichen bis zum Ende des Suchstrings ab. Differiert das jeweilige Zeichen im Speicher mit dem im Suchstring, sucht der Computer das erste Zeichen weiter. Deshalb springt er wieder zur Routine WEITER. Wurde der ganze Suchstring im Speicher entdeckt, wird zur Routine END gesprungen, die die Adresse angibt, wo der String gefunden wurde und den String selber anzeigt. RST 18h ist ein Restart, der auf die Routine springt, die ein Zeichen ausdrückt.

Die Routine SPACE ist relativ uninteressant, sie schiebt nur mit Hilfe des RST 18h drei Leerzeichen in die Ausgabe ein.

Interessant ist die Möglichkeit, mit PRINT USR ein MC-Programm zu starten. Man kann

nämlich dadurch bequem gewisse Meldungen auf den Bildschirm bringen, die man gar nicht vom Programm aus anzeigt. Beim Zurückkehren ins BASIC drückt der Computer nämlich eine Zeichenkette aus, die man mitübergeben will. Bei uns ist das im Falle einer positiven Suche der String selber. Im Falle eines negativen Ergebnisses schicken wir unsere selbstgestrickte Fehlermeldung auf den Schirm, indem wir die DRUCK-Routine im ROM verwenden. Der Computer sieht aber diese Meldung beim Zurückkommen aus dem MC-Code als übergebenen Parameter an und drückt diese Meldung noch einmal aus. Deshalb muß man nach dieser Meldung noch eine ausdrucken lassen, nämlich den Leerstring. Dieser bewirkt überhaupt keine zusätzliche Anzeige, nicht einmal einen Zeilenvorschub, hat aber den Nutzen, daß er vom Interpreter als übergebener Parameter erkannt wird. Der Interpreter drückt ihn also noch einmal aus, da aber der Leerstring nichts am Bildschirm bewirkt, sieht man die ganze Aktion nicht. Der Effekt: Die Fehlermeldung erscheint nicht zweimal, sondern nur einmal. Jeder Leser kann dies testen, wenn er die Unterdrückung, die beiden Befehle "LD HL,ERR2" und "CALL DRUCK", wegnimmt und dann das MC-Programm laufen läßt.

Dieser Assemblerkurs sollte die regulären Befehle mit einem bestimmten Grundwissen vermitteln. Es liegt nun am Programmierer, sein Wissen durch viel Praxis auszubauen. Hier ein Tip: Tüftler, denen zuwenig ROM-Informationen im SVI-Journal zur Verfügung stehen, können anhand der abgedruckten Programme einiges lernen. Viele ROM-Routinen werden dort aufgerufen und gleichzeitig die Parameter gezeigt, die für die jeweiligen Routinen gebraucht werden.

\*\*\*\*\*

#### LITERATURHINWEIS

Als Ergänzung zu unserem nunmehr abgeschlossenen Assemblerkurs können wir Ihnen folgende Bücher empfehlen:

Rodnay Zaks

Programmierung des Z80

Die "Bibel" der Z80-Programmierer ist ein umfassendes Nachschlagwerk für diesen Mikroprozessor. Das Buch ist als Lehrbuch für die Programmierung des Z80 bestens geeignet.

James W. Coffron

Z80 Anwendungen

Dieses Buch bringt alle für den Anschluß von Peripheriebausteinen an den Z80 notwendigen Informationen.

\*\*\*\*\*

#### PROGRAMMIEREN IN ASSEMBLER

Praktische Vorführungen zur Z80-Programmierung in Assembler demnächst an einem außerordentlichen Clubabend.

Nähere Angaben zu diesem Informationsabend erbitten wir telefonisch im Clublokal zu erfragen.

\*\*\*\*\*

Eine wichtige Erweiterung unserer SVI-Computer zur Kommunikation mit der Außenwelt stellt die RS232-Schnittstelle dar. Wie sie funktioniert und was man mit ihr anfangen kann, soll diese Folge über die SVI-Hardware "I/O"-(*"Input/Output"* = Ein-/Ausgabe)-Befeh-

Die RS232-Schnittstelle erlaubt den Anschluß des SVI-Computers an Geräte, die ebenfalls mit einer solchen Schnittstelle ausgestattet sind. Beispielsweise kann man damit eine Verbindung zu einem zweiten Computer herstellen und einen Datenaustausch zwischen den beiden Rechnern durchführen. Man kann einen Akustikkoppler anschließen und so über das Telefonnetz mit Datenbanken kommunizieren. Auch Drucker kann man über diese Schnittstelle ansteuern.

Die RS232-Schnittstelle übermittelt Daten seriell, also Bit für Bit, die Centronics-Schnittstelle hingegen transportiert Daten parallel (also immer ein Byte auf einmal). So vorteilhaft die parallele Datenübermittlung hinsichtlich der Übertragungsgeschwindigkeit ist, so hat sie doch auch einen großen Nachteil. Sie funktioniert nur über kurze Distanzen und nur in einer "geordneten" Computerumgebung. Der Datentransfer innerhalb eines Computers wird beispielsweise immer parallel abgewickelt. Die Temperatur ist kontrollierbar, die elektrischen Eigenschaften können sorgfältig abgestimmt werden. Auch Drucker können in der Regel über kurze Distanzen (ca. 2 Meter) an die parallele Schnittstelle angeschlossen werden.

Für alle anderen Fälle der Datenübertragung muß man eine serielle Schnittstelle wählen. Bereits im Jahre 1969 wurde in den USA der RS-232-Standard geschaffen, der später mit kleinen Änderungen als RS-232-C-Standard herausgegeben wurde und bis heute Gültigkeit hat. Unter der Bezeichnung V.24 wurde später ein weiterer Standard publiziert, der jedoch lediglich die Liste der Schnittstellenleitungen enthielt. Die elektrischen Kennwerte wurden in der Empfehlung V.28 zusammengefaßt, somit entsprechen diese beiden Empfehlungen zusammen der RS-232-C-Schnittstelle. Dies zur Information, da in der Literatur häufig beide Bezeichnungen Verwendung finden.

Die RS-232-C-Schnittstelle SVI-805 wird unter CP/M initialisiert mit einer Übertragungsrate von 4800 Baud, einer Wortlänge von 8 Bits, 1 Stop-Bit und ohne Parity-Checking. Man kann aber mittels des auf der CP/M-Systemdiskette befindlichen Programms "IRS232.COM" diese Parameter ändern. Dem Programmierer stehen dabei Baudraten zwischen 50 und 19200 zur Verfügung.

Um nun den File FILENAME vom Spectravideo-Rechner über die serielle Schnittstelle zu einem anderen Rechner zu befördern, sind folgende Eingaben erforderlich:

beim sendenden Computer:

```
PIP PUN:=FILENAME
```

beim empfangenden Computer:

```
PIP FILENAME=RDR:
```

Voraussetzung ist natürlich, daß Sende- und

Empfangsparameter übereinstimmen und sich das Programm PIP.COM auf der eingelegten Diskette befindet.

Will man jedoch Textfiles zu einem angeschlossenen Drucker schicken, ist folgendermaßen vorzugehen.

Auf dem Drucker bzw. der Schreibmaschine stellt man mit den meist verfügbaren Schaltern die Baudrate ein bzw. initialisiert man mittels des erwähnten CP/M-Programms die SVI-Schnittstelle entsprechend den erforderlichen Werten des Druckers. Jetzt muß man nur noch die Druckerausgabe umleiten. Dies geschieht durch Änderung der Gerätezuordnung mittels des Programms STAT.COM wie folgt:

```
STAT LST:=TTY:
```

Ab jetzt werden alle Druckerausgaben an die serielle Schnittstelle ausgegeben, egal ob es sich um eine mit CTRL-P aktivierte CP/M-Ausgabe handelt oder zum Beispiel um WordStar-Texte.

Wenn Sie STAT DEV: aufrufen, zeigt Ihnen das der Computer auch an. Er wird dann unter anderem auch die Zeile

```
LST: is TTY:
```

anzeigen. Hingewiesen sei in diesem Zusammenhang darauf, daß Gerätebezeichnungen im CP/M immer mit einem Doppelpunkt abgeschlossen werden müssen.

Unter CP/M ist also die Ansteuerung eines Druckers über die RS-232-Schnittstelle relativ leicht zu bewerkstelligen. Im BASIC ist dies schon wesentlich schwieriger. Da alle die Druckerausgabe betreffenden Befehle (LLIST, LPRINT, LFILES) die parallele (Centronics-) Schnittstelle ansprechen und diese Befehle im ROM unveränderlich festgehalten sind, bietet sich hier nur der folgende Ausweg an: Man kopiert das BASIC-ROM in die Bank 2 und ändert dort die Druckerausgaberroutine entsprechend ab.

Die Schnittstelle ist mit einem 25-poligen D-Stecker ausgestattet. Die Steckerbelegung:

Pin	I/O		
1		GND	
2	O	SOUT	Serial data out
3	I	SIN	Serial data in
4	O	RTS	Request to send
5	I	CTS	Clear to send
6	I	DSR	Data set ready
7		GND	
8	I	RLSD	Receive line signal detect
20	O	DTR	Data terminal ready

Auf unserer Schnittstellenkarte ist das "Programmierbare Kommunikationsinterface" 8250 für die Abwicklung des Datentransfers verantwortlich. Das Format der asynchronen Datenübermittlung kann vom Programmierer im Line Control Register festgelegt werden. Der 8250 ermöglicht auch das Auslesen dieses Registers. Das Register wird über die I/O-Adresse 2BH angesprochen.

Wer mehr über die serielle Datenübertragung wissen möchte, sei auf das Buch "V24/RS-232 Kommunikation" von Joe Campbell, Sybex-Verlag, hingewiesen.

In der fünften Folge dieser Serie will ich die restlichen Stringfunktionen und Stringprozeduren erklären.

In der vorigen Folge wurden schon die Befehle DELETE und INSERT besprochen, doch PASCAL bietet noch einige Befehle mehr, Strings zu bearbeiten:

```
STR(wert,st)
```

Diese Prozedur wandelt die INTEGER- oder REAL-Zahl "wert" in den String "st" um. Zu beachten ist, daß eine Formatierungsanweisung der Variablen "wert" folgen kann. Der Befehl STR(x:10:2,st) erzeugt, wenn "x" den Wert 124.5 hat, den String "124.5". Die Format-Anweisung hat dieselbe Syntax wie in der WRITE-Prozedur:

```
x:Vorkommastellen:Nachkommastellen
```

```
VAL(st,zahl,code)
```

ist das Gegenteil zu STR und einspricht im Prinzip der BASIC-Anweisung VAL. Der String "st" wird in die INTEGER- oder REAL-Variable "zahl" umgewandelt. Im Gegensatz zu BASIC, das nur 0 erzeugt, wenn "st" keinen gültigen Wert enthält, liefert PASCAL in der INTEGER-Variablen "code" genauere Informationen, wo ein Fehler aufgetreten ist. Wenn kein Fehler aufgetreten ist, wird "code" null, ansonsten enthält "code" die Position des Buchstabens, an dem der Fehler aufgetreten ist. "zahl" nimmt dabei einen undefinierten Wert an. Wenn st "234" ist wird "zahl" 234 und "code" 0, falls "st" den Inhalt "153 stück" hat, enthält "zahl" einen undefinierten Wert, und "code" 4, da an der 4.Stelle ein Fehler aufgetreten ist. Führende und hintenangelassene Blanks werden ebenfalls als Fehler behandelt. VAL('123',zahl,code) liefert deswegen den Wert 1 für "code" und keinen Wert für "zahl".

Bei den Prozeduren STR und VAL ist noch zu beachten, daß sie nicht in einem WRITELN- oder WRITE-Statement verwendet werden dürfen. Das gilt allerdings nur für die 8-Bit-Version von TURBO. In anderen PASCAL-Dialekten ist die Verwendung von STR und VAL gestattet.

Neben diesen vier Standardprozeduren zur Stringverarbeitung bietet PASCAL noch vier weitere Funktionen an:

```
COPY(st,pos,num)
```

Diese Funktion liefert einen Substring von "st" der "num" Zeichen lang ist, und an der "pos"-ten Stelle von "st" beginnt. Die Befehlsfolge:

```
st:='TURBO auf SVI-328';
st1:=COPY(st,7,3);
```

weist "st1" die Kette "auf" zu. Ein Fehler tritt auf, wenn "pos"+"num" größer als 255 wird.

```
CONCAT(st1,st2,st3,.....,stn);
```

verkettet die Strings st1,st2,st3,.....,stn. Statt CONCAT kann man auch "+" verwenden.

Wenn st1 "TURBO" und st2 "-PASCAL" ist, weist das Statement

```
st3:=CONCAT(st1,st2);
```

"st3" den String "TURBO-PASCAL" zu und ist gleichbedeutend mit

```
st3:=st1+st2;
```

```
LENGTH(st)
```

ist dasselbe wie LEN in BASIC und liefert die Länge des Strings "st"

```
POS(obj,ziel)
```

Diese Funktion liefert die Position des ersten Auftretens des Strings "obj" im String "ziel". Wenn "obj" nicht in "ziel" enthalten ist, wird POS Null.

```
obj:='TURBO';
ziel:='Wir programmieren in TURBO';
a:=POS(obj,ziel);
```

weist der INTEGER-Variablen a den Wert 22 zu.

Damit sind alle STRING-Befehle von PASCAL erklärt. Das Programm "hangman" verdeutlicht den Gebrauch dieser Anweisungen. Daneben wird der Variablentyp CHAR verwendet, der im Prinzip dasselbe wie STRING(.1.) darstellt. Die Datentypen STRING und CHAR sind deshalb vollständig kompatibel, das heißt, sie lassen sich beliebig mischen. Außerdem wird die Standardprozedur GOTOXY eingeführt. Dieser Befehl plaziert den Cursor, die Syntax lautet:

```
GOTOXY(x-Koordinate,y-Koordinate);
```

Der Ursprung liegt in der linken oberen Ecke. Zu beachten ist noch, daß die Koordinaten bei Eins und nicht bei Null, wie in BASIC üblich, zu zählen beginnen. Noch eine weitere Funktion lernen wir in diesem Programm kennen:

```
CHR(wert);
```

Dabei ist "wert" ein INTEGER-Wert zwischen Null und 255. CHR entspricht dem CHR\$ im BASIC, das heißt, es liefert den Buchstaben mit dem ASCII-Code "wert".

Das Spiel "Hangman" dürfte ja wohl jeder kennen, ich will es trotzdem noch kurz erklären. Ein Mitspieler gibt ein Wort ein. Danach zeichnet der Computer so viele Punkte auf den Schirm, wie das Wort Buchstaben hat. Nun muß ein anderer Mitspieler versuchen, das Wort Buchstabe für Buchstabe zu erraten. Dabei zeigt der Computer, welche Buchstaben richtig geraten werden, und printet sie an die richtige Stelle. Das Spiel ist aus, wenn das ganze Wort vollständig am Bildschirm steht.

Nun aber zum Programmtechnischen! Zuerst werden wie immer die Variablen definiert. Der String "suchstring" soll das zu erratende Wort enthalten (höchstens 40 Buchstaben, 80-Zeichenkartenbesitzer können diese Zahl auf 80 vergrößern). Der getippte Buchstabe soll in der CHAR-Variablen "eingabe" abgelegt werden. "ort" ist der Platz, an dem der Buchstabe, falls überhaupt, auftritt, "rich-

tig" ist ein Zähler für die schon erratenen Buchstaben, "anzahl" für die Gesamtanzahl der Versuche. "f" ist ein Schleifenzähler und "tab" ist ein Zähler für einen GOTOXY-Befehl.

Das Hauptprogramm beginnt mit der Aufforderung, das zu erratende Wort einzugeben. Danach werden die Punkte gezeichnet. Dann geht es mit dem Rätselraten los. Die Auswertung des Tips übernimmt die Prozedur "ratebuchstabe". Mit POS wird überprüft, ob der geratene Buchstabe in dem Wort vorkommt. Wenn dies zutrifft, wird er angezeigt und im String "suchstring" durch CHR\$(1) ersetzt, damit er nicht noch einmal "gefunden" wird. CHR\$(1) wird deswegen verwendet, weil die eigentlichen Buchstaben im ASCII-Code erst bei CHR\$(32) (=Space) beginnen und CHR\$(1) somit nicht eingegeben werden kann. Und damit kann es auch nicht in "suchstring" enthalten sein. Sonst weist das Programm keine Besonderheiten auf. Die REPEAT-UNTIL Schleifen zeigen wieder sehr deutlich, wie man ein Problem in PASCAL elegant lösen kann, das im BASIC sehr "unelegant" mit IF und GOTO bewältigt werden müßte.

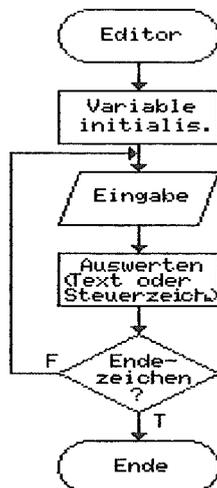
Doch nun komme ich zu einem Projekt, das uns die nächsten Folgen beschäftigen wird: Die Programmierung eines einfachen bildschirmorientierten Editors. Dabei will ich auch versuchen, ein Konzept zur Entwicklung von komplexen Programmen zu zeigen, da man schon in der Entwicklung daran denken muß, das Programm einfach und gut lesbar, eben strukturiert zu schreiben.

Die erste Frage, die man sich stellen muß, ist:

WAS MUSS DAS PROGRAMM ALLES KÖNNEN ?

In unserem Fall soll das Programm ein Editor sein. Zur Vereinfachung soll der gerade bearbeitete Text immer nur eine Seite lang sein, ein "Blättern" im Text ist also nicht notwendig. Der Editor soll alle wichtigen Funktionen des TURBO-Editors erfüllen, und die Texte müssen sich natürlich auf Diskette abspeichern lassen. Die Blockoperationen des TURBO-Editors sind bei Texten von der Länge einer Bildschirmseite nicht unbedingt notwendig und sollen deshalb auch nicht implementiert werden. INSERT, DELETE, Springen zum nächsten Wort und ähnliche Operationen sollen so wie im TURBO-Editor vorhanden sein.

Nachdem man diese Frage beantwortet hat, muß man zuerst ein grobes Flußdiagramm erstellen. Bei unserem Programm sieht das so aus:



Nach diesem groben Entwurf muß man die einzelnen Teile genauer planen, am besten auf einem großem Stück Papier. Bei unserem Beispiel ist besonders der Teil "Auswerten der Eingabe wichtig", da ja sehr viele Steuerzeichen verarbeitet werden müssen.

In der nächsten Folge werde ich genauer auf diesen Teil eingehen, dabei lernen wir eine neue Entscheidungsmöglichkeit in PASCAL kennen, die CASE-Anweisung. Natürlich gibt es auch wieder etwas Praxis, der Editor wird uns ebenso wieder beschäftigen.

RAFAEL RAZIM

```

PROGRAM hangman;

VAR suchstring:STRING(.40.);
    eingabe:CHAR;
    f,ort,richtig,tab,anzahl:integer;
    falsch:BOOLEAN;

LABEL 1;

PROCEDURE ratebuchstabe;
BEGIN
    GOTOXY(30,10);
    WRITE(' ');
    GOTOXY(1,10);
    WRITE('Bitte einen Buchstaben raten! ');
    READLN(eingabe);
    REPEAT
        ort:=POS(eingabe,suchstring);
        IF ort=0
            THEN
                BEGIN
                    falsch:=TRUE;
                    GOTOXY(tab,22);
                    WRITE(eingabe);
                    tab:=tab+1;
                END
            ELSE
                BEGIN
                    falsch:=FALSE;
                    richtig:=richtig+1;
                    GOTOXY(1+ort,3);
                    WRITE(eingabe);
                    DELETE(suchstring,ort,1);
                    INSERT(chr(1),suchstring,ort);
                END;
            UNTIL falsch;
    END;

BEGIN
    CLRSCR;
    richtig:=0;
    tab:=1;
    anzahl:=0;
    WRITE('Bitte das zu erratende Wort');
    WRITE(' eingeben: ');
    READLN(suchstring);
    CLRSCR;
    FOR f:= 1 TO LENGTH(suchstring) DO
        BEGIN
            GOTOXY(1+f,3);
            WRITE('.');
        END;
    REPEAT
        ratebuchstabe;
        anzahl:=anzahl+1;
    UNTIL richtig >= LENGTH(suchstring);
    GOTOXY(1,12);
    WRITE('Sie haben ',anzahl);
    WRITELN(' Versuche benötigt');
END.
  
```

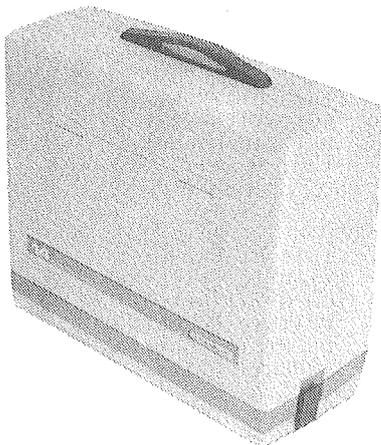
---

## Tragbarer Computer zum tragbaren Preis!

---

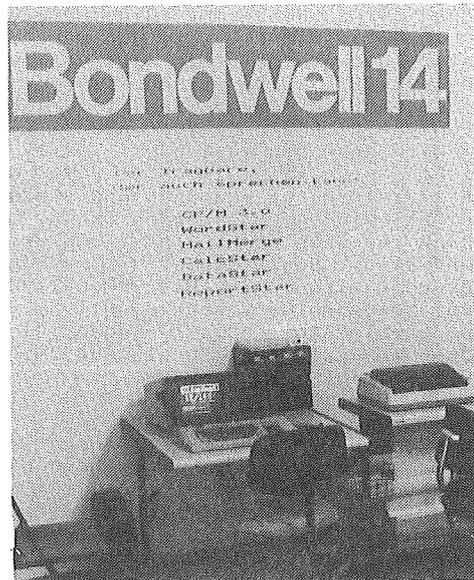
Im Fernen Osten gibt es eine Firma, von der bis jetzt wenig zu hören war. Lediglich auf dem CP/M-Manual der Spectravideo-Computer steht versteckt ein Name: Bondwell. Dabei ist diese Firma größer als Spectravideo und besitzt auch die Mehrheit der Firma, die unsere Computer erzeugt. Jetzt hat Bondwell einen tragbaren Computer in Österreich auf den Markt "geworfen". Als eine der ersten österreichischen Zeitschriften haben wir ihn schon getestet:

Der Bondwell 14 gehört in die Kategorie der transportablen Computer, ein echter Portable ist er nicht. Netzabhängigkeit und nicht zu verachtende Maße verhindern das Prädikat "wirklich tragbarer Computer". Trotzdem ist er kompakt, alle Einzelteile finden im Computer Platz, wenn er einmal auf Reisen mit soll. Ein großer bequemer Tragegriff sorgt für ein beschwerdefreies Tragen. Da er auf der Rückseite des Gerätes montiert ist, stört er bei der Arbeit nicht. Die Tastatur bildet beim Transport den "Deckel", während der Arbeit ist sie mit einem Spiralkabel mit dem Bildschirm und den Disketten verbunden.



Bleiben wir gleich beim Thema Tasten. Der Bondwell hat mit 91 Tasten eine reichliche Tastatur. Alle Tasten sind sehr leichtgängig, wirken jedoch nicht schwammig. Viele wird es freuen zu hören, daß der Computer auch eine deutsche Tastatur besitzt. Das "Z" ist neben dem "T" positioniert und man muß auch nicht mehr auf Umlaute-Suche gehen. Diese sind nämlich deutlich gekennzeichnet rechts vom Buchstabenfeld untergebracht. Zum Glück sind hier auch die kleinen Umlaute ohne SHIFT, die großen mit SHIFT zu erreichen. Dies bedeutet jedoch für Besitzer von Spectravideo-Computern anfangs eine unbequeme Umstellungsphase.

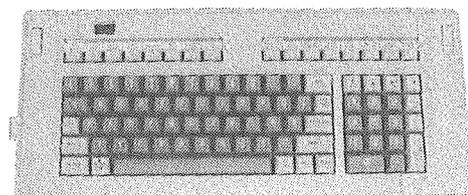
Ein separater Zehnerblock ist vorhanden, ebenso 16 (!) Funktionstasten. Diese Funktionstasten sind wie die des Spectravideo als Soft-Keys verwendbar, das heißt, man kann sie mit ganzen Befehlen belegen und diese aufrufen lassen. Allerdings ist es nicht möglich, pro Taste mehr als nur eine Befehlsgruppe zu belegen. Mit SHIFT wird keine zusätzliche Funktion erreichbar. Positiv ist auch noch die Beschriftung über den Funktionstasten. So kann man immer sehen, welche Tasten wie belegt sind, ohne daß wertvoller Bildschirmplatz verloren geht. Ein nettes Detail am Rande: Es werden Schab-



lonen mitgeliefert, die einige Funktionstastensätze der mitgelieferten Software anzeigen. Eine LED für CAPS-LOCK verhindert Ratespiele, ob Großschreibung an- oder ausgeschaltet ist.

Zwei Diskettenlaufwerke, die doppelseitig arbeiten, und ein 9"-Monitor bilden während des Transportes das Gegenstück zur Tastatur. Der Monitor ist monochrom, hat jedoch ein ausgezeichnetes Bild, da er bernsteinfarben ist. Die Helligkeit und der Kontrast können durch zwei große rote Rädchen verstellt werden. Die Diskettenlaufwerke arbeiten mit 5.25 Zoll-Disketten. Die Kapazität der Disks beläuft sich auf 360KByte pro Floppy.

Kommen wir zu den Anschlüssen. Der Bondwell hat den Großteil seiner Interfaces vornehm hinter einer Klappe versteckt. Neben dem unvermeidlichen Netzstecker gibt es noch einen Druckeranschluß und zwei RS-232-Schnittstellen. Außerdem läßt sich ein externer Monitor anschließen, sollten die eingebauten neun Zoll zu wenig sein.

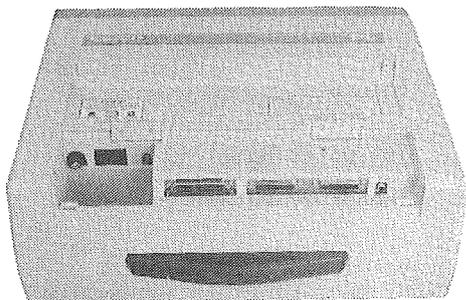


Der Bondwell ist übrigens auch mit einem RESET-Knopf ausgestattet. Dies erlaubt ein schnelles "rebooten", ohne aus- und wieder einzuschalten. Er ist deutlich sichtbar neben dem Ein-/Ausschalteknopf über den zwei Diskettenlaufwerken angebracht, ein besonders guter Platz. Einesseits braucht man keine Armverrenkungen, um ihn zu betätigen (manche Hersteller positionieren ihre RESET-Tasten auf der Hinterseite des Gerätes), andererseits läuft man keine Gefahr, die Taste versehentlich zu betätigen.

Apropos "booten"! Der Bondwell 14 wird mit dem Betriebssystem CP/M 3.00 geliefert. Dieses neue CP/M, es wird auch oft CP/M plus genannt, hat gegenüber dem alten 2.2er einige Vorteile. Darüber jedoch später. Gleich nach dem Einschalten führt der Computer einen Systemtest durch und fragt danach, ob man ein System laden möchte. Ganz durchschaubar ist diese Frage jedoch nicht, unserem Wissen nach bleibt dem Computer nämlich nichts anderes übrig, als das CP/M 3.00 zu laden. Hat man sich dann entschieden, den Computer sein System laden zu lassen, sucht er sich noch ein FILE auf der Diskette, das AUTORUN heißt. Dieses FILE wird dann gleich automatisch gestartet. Ähnlich dem IPL des Spectravideo im BASIC kann man so gewisse Grundeinstellungen definieren oder auf einer WordStar-Diskette gleich das WordStar laden lassen. Natürlich kann man ganz leicht selber jedes beliebige File durch Umbenennen als "AUTORUN" definieren. Existiert keine Datei mit diesem Namen auf der Diskette, gibt der Computer eine Fehlermeldung aus, die aber ganz normal ist und niemanden zu stören braucht.

Das CP/M 3.00 wird für den Bondwell auf zwei Disketten geliefert. Eine konnte nicht alle Systemfiles aufnehmen. Eine Datei nennt sich HELP und hilft dem Bediener bei einigen anderen Dateien, was diese können und wie man sie verwendet. Das neue CP/M unterstützt übrigens auch das Bankswitching. Der Bondwell besitzt ja mehrere Banken, die vom CP/M voll genutzt werden können. Außerdem braucht man nach einem Diskettenwechsel kein CTRL-C durchzuführen. Alle Dateien können mit einem Paßwort geschützt und mit dem Datum von Änderungen versehen werden.

Eine weitere sehr nützliche Einrichtung des CP/M 3.00 ist, daß der Editor verbessert wurde. Man kann in jeder Position mit den Tasten CTRL-A den Cursor ohne Löschen nach links und mit CTRL-F den Cursor nach rechts bewegen. Beim CP/M 2.2 konnte man im Betriebssystem nur mit gleichzeitigem Löschen



editieren. Dies sind die wichtigsten Neuerungen dieses Betriebssystems.

Interessant am Bondwell ist noch, daß verschiedene Diskettenformate gelesen werden können. Neben den Bondwell- und Spectravideo-Disketten sind auch Osborne I und Kaypro II lesbar. Unterschieden wird durch das Anwählen von verschiedenen logischen Laufwerken. Nur in Laufwerk B dürfen sich Fremdformate befinden. Die untenstehende Tabelle gibt über die einzelnen Formate Aufschluß:

"B:" Bondwell 14  
 "C:" Bondwell 12  
 "D:" Spectravideo  
 "E:" Kaypro II  
 "F:" Osborne I

Wir testeten natürlich die Kompatibilität zu SVI-Disketten. Einseitige waren ohne Einschränkung zu bearbeiten. Doppelseitige Floppies dürfen allerdings nicht verwendet werden.

Eine weitere nette Zusatzeinrichtung ist der Sprachsynthesizer, der eingebaut ist. Mit einem Programm, das auf der Systemdiskette mitgeliefert wird, kann man den Bondwell somit sprechen lassen. Es wird im Programm SPEECH zwischen "Englisch" und "Phoneme" unterschieden. Am Anfang muß man sich auf eine der beiden Arten festlegen. Mit "Englisch" wird die Eingabe schriftähnlicher, man kann jedoch nur englische Wörter halbwegs fehlerfrei erzeugen. Bei "Phoneme" wird die Eingabe von Wörtern gegenüber der normalen Schriftsprache total abgeändert, dafür läßt sich auch in Deutsch sprechen. Phoneme sind ja bekanntlich international. Einziger Nachteil: Der Bondwell besitzt einige Phoneme der deutschen Sprache nicht. Ein "ü" oder ein "ö" müssen mühsam umschrieben werden.

Wie bringt man den Bondwell zum Beispiel dazu, "Spectra Video Club Austria" auszusprechen?

in "English": spectra video cloob austria  
 in "Phoneme": spehktrahah vihihdehoh kluph ahahuhstrihahah

Die beiden Systemdisketten des Bondwell besitzen einige sehr interessante Dateien. SHOW gibt Auskunft über Spezifikationen und Zustand von Disketten (eine Art verbessertes STAT), LIB erzeugt Libraries, mit SETUP können Parameter leicht verstellt werden und mit DATE werden Files mit dem gewünschten Datum versehen.

Neben diesen neuen Files gibt es noch viele vom Spectravideo her bekannte. PIP, COPY,

Directory For Drive A: User 0

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes
AUTORUN	COM	2k	1 Dir RW	BACKUP	COM	2k	10 Dir RW
COPYSYS	COM	2k	14 Dir RW	CPM3	SYS	18k	140 Dir RW
DATE	COM	4k	22 Dir RW	DEVICE	COM	8k	58 Dir RW
DIR	COM	16k	114 Dir RW	DUMP	COM	2k	8 Dir RW
ED	COM	10k	73 Dir RW	ERASE	COM	4k	29 Dir RW
FORMAT	COM	2k	12 Dir RW	GET	COM	8k	51 Dir RW
HELP	COM	8k	56 Dir RW	HELP	HLP	64k	508 Dir RW
INITDIR	COM	32k	250 Dir RW	PATCH	COM	4k	19 Dir RW
PIP	COM	10k	68 Dir RW	PUT	COM	8k	55 Dir RW
RENAME	COM	4k	23 Dir RW	RS232R	COM	2k	13 Dir RW
RS232T	COM	2k	10 Dir RW	SAMPLE	SPH	6k	36 Dir RW
SET	COM	12k	81 Dir RW	SETDEF	COM	4k	32 Dir RW
SETUP	COM	14k	107 Dir RW	SHOW	COM	10k	66 Dir RW
SPEECH	COM	28k	224 Dir RW	SUBMIT	COM	6k	42 Dir RW
TYPE	COM	4k	24 Dir RW				

Total Bytes = 296k Total Records = 2146 Files Found = 29  
 Total 1k Blocks = 282 Used/Max Dir Entries For Drive A: 30/ 128

SUBMIT, ED, TYPE und andere können ebenso verwendet werden. Interessant ist noch das File DIR, das anfangs etwas unbeachtet geblieben war. Es stellte sich heraus, das der Bondwell wie jeder CP/M-Computer ein DIR als eingebauten Befehl hat, daß es aber auch ein DIR als Datei gibt. Der Befehl tut nichts anderes als schon vom Spectravideo her bekannt ist. Die Datei DIR verbindet das alte STAT mit dem Befehl DIR. Es werden alle Dateien einer Diskette in alphabetisch geordneter Reihenfolge ausgegeben, ihr Speicherverbrauch angegeben und andere Diskettendaten angezeigt. Natürlich besitzt die Datei DIR einige zusätzliche Options. Man kann unterscheiden, ob man alle Disketten aufgelistet haben will, welche Datenfiles gesucht werden sollen und so weiter.

Selbstverständlich ist am Bondwell auch WordStar lauffähig. Deshalb wurde dieses Programm auch gleich getestet. WordStar war nach dem Installieren ohne Einschränkungen zu bedienen. Ein kleiner Minuspunkt ist jedoch vorhanden. Das Scrolling am Bildschirm wird sehr langsam durchgeführt. Bei schnellem Schreiben oder beim Blättern durch einen Text kann sich das lästig auswirken.

Mitgeliefert werden übrigens auch einige CP/M-Programme von MicroPro. Neben dem guten

alten WordStar gibt es noch ReportStar, DataStar und CalcStar. Mit diesem Softwarepaket hat man den Grundstock für eine vernünftige professionelle Anwendung gelegt. WordStar unterstützt Textverarbeitung, CalcStar Tabellenkalkulation und DataStar ist als Datenbanksystem unentbehrlich. ReportStar wird für das Programm DataStar gebraucht. Es steuert die Ausgabe der Datenbanken auf den Drucker und kann die Files ebenso manipulieren.

Zusammenfassend kann man sagen, daß der Bondwell ein sehr guter Portable ist. Wir sagen das übrigens nicht, weil er einer unserer Clubcomputer ist, es ist die ehrliche Meinung des Testers, der von einem Spectravideo kommt. Der Bondwell hat eine gute Tastatur, wird mit dem verbesserten CP/M plus geliefert und kann auch für Spaß sorgen, wenn man seinen Sprachsynthesizer aktiviert. Schwachstellen sind vor allem das Handbuch, das in Englisch ist und einige wichtige Informationen vorenthält.

Interessant ist die Möglichkeit, Fremdformate zu lesen. Den Bondwell gibt es dabei schon um einen Preis von rund 40.000 Schilling. Welcher andere Portable kann solche Leistungen um so einen günstigen Preis bieten?

\*\*\*\*\*  
\*  
\* Bericht über die Gründungsversammlung \*  
\*  
\*\*\*\*\*

Als erster gewählter Vereinsobmann habe ich die Ehre, einen kurzen Bericht über die durchgeführte Gründungs- und erste Jahresversammlung zu liefern.

Als negativer Punkt ist hier die Anwesenheit von nur 16 Mitgliedern anzuführen. Die Versammlung konnte dadurch erst nach 30 Minuten Verzögerung beginnen, damit nach den Vereinsstatuten die Beschlußfähigkeit erreicht wurde.

Die Tagesordnung enthielt dann folgende Punkte:

- Wahl des Vorstandes
- Wahl der Funktionäre (Beiräte)
- Wahl der Rechnungsprüfer
- Rechnungsvoranschlag für 1985
- Festsetzung der Mitgliedsbeiträge
- Allfälliges

Die Wahl des Vorstandes, der Funktionäre und Rechnungsprüfer wurde, da jeweils nur ein Kandidat zur Verfügung stand, als offene Abstimmung durch Handzeichen mit einfacher Mehrheit durchgeführt. Dabei wurden alle Wahlvorschläge ohne Gegenstimme angenommen.

Das Ergebnis der Wahl im einzelnen:

Für den Vorstand

Obmann	Wolfgang Rotschek
Obmann Stellvertr.	Stefan Kovalovsky
Schriftführer	Ing. Helmut Jentzsch-Stricker
Kassier	Heinz Schmid

Als Funktionäre

Rechnungsprüfer 1	Richard Holzmann
Rechnungsprüfer 2	Dr. Otmar Mak
Korrespondent Inland	Andreas Stocker
Korrespondent Ausland	Rafael Razim
Chefredakteur des SVI-Journals	Gerhard Fally

Der Rechnungsvoranschlag für das Jahr 1985 sowie die Festsetzung der Mitgliedsbeiträge, welche ohne Änderung beibehalten werden, wurden dann ebenfalls ohne Gegenstimme angenommen.

Unter "Allfälliges" wurden einige interessante Punkte von den anwesenden Mitgliedern eingebracht.

So wurde zum Beispiel die Frage eines Vereinseblems aufgeworfen. Ein genauerer Beitrag darüber erscheint im folgenden SVI-Journal.

Weiters wurde über die Herstellung von Mitgliedsausweisen diskutiert. Dieser Vorschlag wurde dankend aufgenommen und die Ausgabe von Mitgliedsausweisen wird demnächst erfolgen.

Zum Abschluß wurde noch ein wichtiger Punkt erwähnt. Es wurde nämlich erörtert, unter welchen Bedingungen club-eigene Programme weitergegeben werden dürfen. Dazu ist zu sagen, daß ein Programmtausch mit club-eigenen Programmen solange gestattet ist, als dem SVI-Club die durch den Tausch erhaltenen Programme ebenfalls zugänglich gemacht werden. Davon ausgenommen sind selbstverständlich alle Programme, welche urheberrechtlich geschützt sind. Weiters ist es nicht erlaubt und gleichzeitig illegal, wenn jemand club-eigene Programme zum Verkauf oder Tausch anbietet um daraus persönliche Vorteile zu erzielen. Dies findet, so hoffe ich, bei allen Verständnis.

Zum Abschluß möchte ich mich noch für das bei der Wahl geschenkte Vertrauen bedanken und hoffe, es voll zu erfüllen.

Wolfgang Rotschek

---

KLEINANZEIGEN

---

SVI-318, 9 Monate alt, Fixpreis S 2.990,-, Auskunft im Computer-Studio, Tel. 65 88 93.

Derjenige CP/M-Benutzer, der der englischen Sprache nicht oder nur kaum mächtig ist, legt bald den "CP/M User's Guide" beiseite und macht sich frustriert auf die Suche nach einer Publikation, die ihn in Deutsch in die Geheimnisse von CP/M einweiht. Er stößt dabei mit ziemlicher Sicherheit auf das "CP/M-Handbuch" von Rodney Zaks.

Anhand des ersten Kapitels läßt sich bereits erkennen, für wen das Buch hauptsächlich gedacht ist. Es werden nämlich die Grundbegriffe des Arbeitens mit Computern erläutert. Weiters wird ein erster Einblick in das Arbeiten mit CP/M gewährt, sodaß der Anwender nach dem Durchlesen in der Lage sein sollte, grundlegende Operationen wie das Kopieren von Disketten und das Löschen oder Umbenennen von Dateien durchzuführen.

Im zweiten Kapitel werden die CP/M-Befehle für die Arbeit mit Dateien sowie die Wirkung der Steuerzeichen konkret beschrieben, man erfährt auch näheres über zusätzliche Befehle bei CP/M 3.0, CCP/M und MP/M.

Kapitel 3 widmet sich dem Dateitransferprogramm PIP in seinen sämtlichen Anwendungsmöglichkeiten. Zahlreiche Beispiele illustrieren die Mächtigkeit dieses nicht-residenten Befehles.

Ebenso ausführlich wie PIP wird der Editor ED in Kapitel 4 besprochen. Es wird jedoch darauf hingewiesen, daß ED nicht unbedingt das höchste der Gefühle in punkto Komfort ist, sondern daß bedeutend leistungsfähigere Textverarbeitungsprogramme unter CP/M zur Verfügung stehen.

Die Schwachstelle des Buches bildet nach der Meinung des Autors Kapitel 5. In diesem Abschnitt werden Struktur und Details aus dem Innenleben des Betriebssystems beschrieben. Diese sind jedoch ohne Zweifel für Anfänger, an die sich das Buch zweifellos richtet, zu kompliziert beschrieben. Dem Systemprogrammierer bringen sie jedoch kaum Neues. Interessant ist lediglich die Änderung in ein "Autostart-CP/M", das beim Einschalten des Computers automatisch ein bestimmtes Programm startet.

Im Kapitel 6 findet sich eine Zusammenfassung aller unter den sechs Betriebssystemen von Digital Research verwendeten Befehle und Programme. Außerdem wird festgehalten, für welche Betriebssysteme diese jeweils gelten, welches Format und welche Argumente sie haben. Eine Beschreibung des Befehls oder Programms und ein Beispiel sind ebenfalls dabei.

Kapitel 7 gibt dem Benutzer Tips zur Arbeit mit CP/M und zur Behebung von Fehlern, die von der Hardware, Software oder vom Bediener verursacht werden können.

Eine Beschreibung der Entwicklungsgeschichte von CP/M gibt Kapitel 8.

Im Anhang finden sich verschiedene Tabellen und Listen, zum Beispiel die CP/M-Fehlermeldungen oder die Liste der verwendbaren Steuerzeichen.

Das CP/M-Handbuch eignet sich für alle diejenigen, die das Betriebssystem nur als Mittel zum Zweck sehen, ihre eigenen, in einer höheren Sprache geschriebenen Programme zu fahren oder mit dem Computer zu arbeiten, ohne sich mit Details zu belasten, wie der Computer das genau tut. Die ausführlichen Beschreibungen der einzelnen Befehle sollten dieses Arbeiten möglich machen. Dem Systemprogrammierer kann das Buch jedoch kaum Interessantes bringen.

Stefan Kovalovsky

```
*****
*
*   ORGANISATION DER CLUBBIBLIOTHEK
*
*****
```

Um die bereits angekündigte Clubbibliothek in Schwung zu bringen, wird in diesem Artikel die geplante Form der Durchführung erläutert:

Die Bibliothek wird in zwei große Gruppen eingeteilt:

- 1) Programme aus der Clubzeitschrift
- 2) Bibliotheksprogramme

ad 1): Auf diese Programme haben alle jene Zugriff, welche die Clubzeitschrift in irgendeiner Form beziehen (Clubmitglieder, Abonnenten oder Einzelbezieher).

Bei Diskettenbenützern werden die Programme nur in gesammelter Form abgegeben, das heißt, eine Diskette umfaßt die Programme von mindestens einer kompletten Zeitschrift. Wir werden bemüht sein, in möglichst kurzer Zeit alle bereits veröffentlichten Programme zur Verfügung zu stellen. Wie weit diese Bemühungen bereits fortgeschritten sind, kann jederzeit im Clublokal erfragt werden.

ad 2): Bei der eigentlichen Bibliothek wird eine Liste der verfügbaren Programme in den Clubräumen aufliegen, zu der alle Clubmitglieder ständigen Zugriff haben. Darüber hinaus wird es ebenfalls eine Liste geben, in welche die Benutzer der Bibliothek jene Programme eintragen, die sie gerne besitzen möchten. Diese Eintragungen sollen auch die Art des Datenträgers beinhalten. Auf Grund der oben genannten Liste werden die gewünschten Programme dann auf den geforderten Datenträgern, wenn zeit- und aufwandsmäßig möglich, bis zum nächsten Clubabend bereitgestellt werden.

Um auch den Mitgliedern, die nicht an den Clubabenden teilnehmen können, den Zugriff auf dieses Service zu bieten, haben wir beschlossen, für diese Leute einen ebenfalls kostenlosen Postversand durchzuführen. Reservierungen von gewünschten Programmen werden aus diesem Grund auch telefonisch im Computer-Studio entgegengenommen.

Johann Kammerer

Die ifabo 1985 war eine der größten europäischen Messen ihrer Art. An fünf Tagen wurde allerlei Hard- und Software ausgestellt und angeboten. Auch der Spectra Video Club Austria hatte ein "Standl" okkupiert. Gemeinsam mit der Wehsner Ges.m.b.H. waren wir am Stand 1009 in der Halle 10 dabei. Hier ein Rückblick:

Die ifabo 85 wurde nach einem völlig neuen Konzept veranstaltet. Das Messegelände wurde in drei Teile gegliedert. Die Zentren SÜD, MITTE und WEST wurden nach ihren Schwerpunkten gekennzeichnet. Zu den drei Schwerpunkthemen wurden diesmal auch Vorträge gehalten. Auf der ifabo befanden sich heuer 461 Aussteller. 787 Firmen aus 25 Ländern wurden vertreten. Das Areal des "ifabo-Geländes" konnte gegenüber dem Vorjahr um 18.000 m<sup>2</sup> auf 69.000 m<sup>2</sup> gesteigert werden.

Neben den traditionellen Hardwareausstellern fand heuer zum zweiten Mal die Softwaremesse "programma" statt. 134 Softwarehäuser, Rechenzentren und andere Firmen konnten in der Galerie 15 dem fachkundigen Publikum ihre Waren zeigen.



Diese Daten sollen die Wichtigkeit der Messe verdeutlichen, auf der wir auch ausstellten. Kommen wir nun zu unserem Stand. Schon in den ersten Stunden des 7. Mai ging es recht turbulent zu. Alle Aktiven unseres Standes hatten sich - wie vereinbart - schon um 8.30 in der Halle 10 zusammengefunden, bis auf einen: unseren Hausherrn, dem Chef der Wehsner Ges.m.b.H. Da einige wichtige Utensilien noch im versperrten Raum des Standes waren, wir aber keinen Schlüssel hatten, mußten wir bis um halb zehn warten, um die Spectravideo-Computer in Betrieb zu nehmen. Doch schon um zehn Uhr hatte sich alles beruhigt, und wir konnten auf "Kundenfang" gehen. Glücklicherweise fanden sich in den ersten drei Tagen weniger Schaulustige sondern mehr wirklich Interessierte am Stand ein.

Wie erwartet konnten wir fünf Computer einsetzen. Zwei SVI-728 demonstrierten ohne Pause MSX-Spiele, ein SVI-318 wurde mit dem Modul "SectorAlpha" beschäftigt. Die zwei SVI-328 wurden für unterschiedliche Zwecke gebraucht. Am Anfang wurde CP/M bevorzugt, danach zeigten Clubmitglieder ihre Kunstwerke, unter ihnen Rafael Razim mit seinem Graphikmodul (Graphiken in fast jeder auf



dem SVI lauffähigen Sprache, Testbericht in diesem Heft). Am Wochenende mußten sie auch als Spielcomputer dienen.

Im Bild ist der Messestand sichtbar. Deutlich sind die Menschenmassen zu erkennen, die unseren Stand am Wochenende besuchten. Obwohl die offizielle Messezeit schon aus war (!), tummelten sich noch so viele zur Zeit des Fotografierens beim SVCA.

Neben dieser Ausrüstung hatte natürlich noch das Computer-Studio Computer aufgebaut. Dort konnte man die professionelle Seite der Computer sehen. CP/M in Reinkultur, WordStar und viele andere interessante Programme wurden präsentiert. Der neue Bondwell 14, der ebenfalls für diese Ausgabe getestet wurde, erlebte auf der ifabo sein Debut. Da Bondwell die Schwesterfirma von Spectravideo ist, haben wir übrigens auch Bondwell-Computer in unserem Club aufgestellt und begrüßen alle Bondwell-Besitzer, die unserem Club beitreten wollen.

Wie sieht nun das Fazit dieser Messe aus?

Obwohl die "ifabo" eine größere Messe als die Hobby-Elektronik ist, meldeten sich diesmal nur vier Personen während der Messezeit für den Club an. Es ist dies wahrscheinlich auf den leichten Besucherschwund der ersten Tage zurückzuführen. Außerdem waren am Wochenende dann oft zu viele "Nur-Spieler" am Stand. Vielleicht hat sich auch die fehlende Vorbereitung von Seiten des Clubs bemerkbar gemacht. Diesmal gab es keine Attraktionen zu bieten, wie etwa der Sprachsynthesizer im Vorjahr, die schriftlichen Unterlagen des Clubs für die Messe, sie waren als Experiment gedacht, ließen an Informationsgehalt zu wünschen übrig.

Trotzdem sind wir der Auffassung, daß es richtig und nützlich war, auf der "ifabo" zu erscheinen. Denn obwohl sich wenige adhoc entschließen konnten, unserem Club beizutreten, zollten viele dem SVCA ihr Interesse. Wir werden uns aber ohne Zweifel für die nächsten Messen gründlicher vorbereiten müssen.

Für besonders Fleißige: die nächste Messe, an der wir voraussichtlich teilnehmen werden, wird die nächstjährige ifabo sein. Ideen für Aktionen, Programme oder Hardware nehmen wir gerne entgegen.

```

*****
*
*      3D-Graphik-Nachlese   Teil 2
*
*****

```

Nachdem wir uns in der ersten Folge mit der Anzeigart des Programms vertraut gemacht haben, werden wir diesmal näheres über die kartesische Koordinatendarstellung hören.

Vor einigen hundert Jahren, als es noch unzählige Mathematik-Genies auf unserer Erde gab, machte sich auch ein gewisser Descartes (gesprochen "Dekart"), genannt Cartesius, Gedanken über die beste Darstellungsart von Funktionsgraphen und ähnlichem. Ihm verdanken wir die Tatsache, daß die einzelnen Achsen in den heute gebräuchlichen Diagrammen im Winkel von 90° zueinander stehen. Schon in der ersten Folge wurde ja darauf hingewiesen, daß auch bei unserem Programm so eine Regel verwendet wird. Daher kommt der Name "kartesisch".

In diesem System bekommt jede Achse ihren eigenen Namen, "x" (im Programm vom Mittelpunkt weg nach links zeigend), "y" (im Programm vom Mittelpunkt weg nach rechts zeigend) und "z" (im Programm vom Mittelpunkt weg nach oben zeigend). Wenn nun ein dreidimensionaler Graph gezeichnet wird, zerlegt der Computer die y-Achse in einzelne Streifen und zeichnet für jeden Streifen ein zweidimensionales Gebilde. Wie diese einzelnen zweidimensionalen Graphen erzeugt werden, dürfte jedem bekannt sein. Es wird für jeden x-Wert anhand der Formel, die man eingegeben hat, der z-Wert errechnet. Für y wird bei jedem Teilgraph der fixe Wert eingesetzt, der dem gerade benutzten Streifen analog ist. Man sieht diese Arbeitsweise am besten, wenn man dem Computer beim Erstellen eines Graphen zusieht. Nach dem ersten Durchgang vollführt das Programm dann noch einen Durchlauf, der etwas anders funktioniert und zum Karrieren des Ganzen dient.

Beim Erstellen einer Formel für kartesische Koordinatendarstellung hat man praktisch Narrenfreiheit. Jede Formel, die halbwegs mathematisch aussieht, kann man in der Regel verwenden (von  $z=x$ ,  $z=2*x$  über  $z=2*x+y$ ,  $z=\text{abs}(\text{sqr}(x*y))$  bis  $z=\sin(x)*.2+\cos(y*5)/5$ ). Aufpassen muß man nur, daß am Anfang "z=" steht und hinter dem Gleichheitszeichen zumindest einmal "x" oder "y" oder beides vorkommt. x und y sind die beiden Laufparameter für die Achsen. Gibt es sie nicht, entsteht eine horizontale flache Ebene, da kein einziger Wert während der Berechnung verändert werden kann.

Die kartesischen Koordinaten sind relativ leicht verständlich, aber es gibt trotzdem einiges zum Einstellen, bevor man einen Graphen zeichnen kann. Als erstes wird der Mittelpunkt am Schirm fixiert. Dazu gibt man die zwei Koordinaten in der beim Spectravideo üblichen Punktdressierung an. Danach folgen zwei Winkelangaben in Altgrad (nicht in Bogenmaß!). Sie geben den Winkel der x- und der y-Achse zu der Waagrechten an. Allerdings dürfen bestimmte Winkelkombinationen nicht verwendet werden. Der Computer gibt in diesen Fällen Fehlermeldungen aus.

Nach den ersten vier Parametern fragt der Computer nach dem Maßstab, mit dem die y-Ausdehnung der Figur verkleinert werden soll. Die Hintergrundfarbe darf man übrigens auch wählen. Diese sechs Werte sind für alle Eingabearten gleich. Die siebente Frage bezieht sich dann auf die Eingabeart selbst. Nun kommen für jede der drei verschiedenen Wahlmöglichkeiten verschiedene Frageblöcke, die auf die einzelnen Anzeigarten zugeschnitten sind. Bei der "Kartesische Koordi-

naten"-Darstellung kommen noch folgende Werte:

Darstellungsart voll/halb: Bei "halb" wird nur die positive y-Achse im Graph verwendet, "voll" zeichnet auch mit der negativen y-Achse.

Raster: man kann den Raster eingeben, in dem die Zeichnung gefertigt werden soll. Je größer die verwendete Zahl ist, desto größer wird die Zeichnung. Sie wird jedoch dadurch auch schneller fertig. Je kleiner der Raster gewählt wird, desto feiner wird der Graph. Der Computer braucht jedoch viel länger. Außerdem kann es einem passieren, daß man bei den engen Linien die Übersicht verliert und der Graph zwar mathematisch korrekt gezeichnet ist, aber kein Mensch eine Ebene sondern nur ein Liniengewirr oder eine schwarze Fläche sehen kann.

Nun werden wir uns mit den grundlegenden Graphikformeln beschäftigen und versuchen, für spätere Experimente gewisse Richtlinien zu finden:

Am Einfachsten sind Formeln, die nur mit den Grundrechenarten arbeiten. Sie erzeugen aber auch die langweiligsten Ebenen. Sehen wir uns zwei Ebenen an, die durch die vier Grundrechenarten erzeugt wurden:

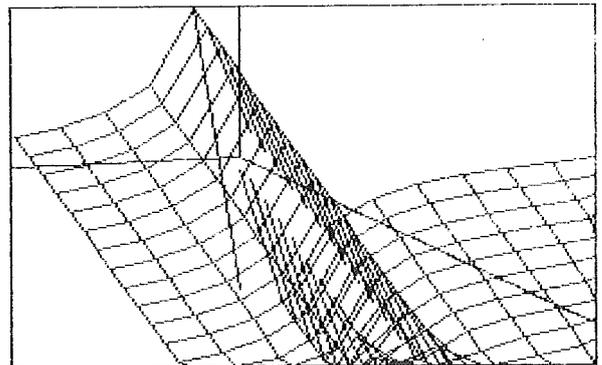


Bild 1

Bild wurde nach der Formel  $z=(1/x)-y/3$  erstellt. Eingabeparameter: 100,80,3,30,2,4, 1,2,.5

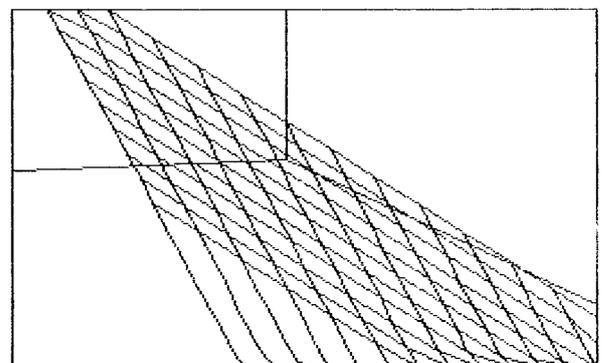


Bild 2

Bild wurde nach der Formel  $z=.8*x+(1-y)/2$  erstellt. Eingabeparameter: 120,80,3,30,2,4, 1,2,.5

Neben diesen Funktionen hält das BASIC noch einige zusätzliche Rechenarten und Anweisungen parat. Neben den Potenzen gibt es noch die Quadratwurzel und viele andere. Eine genaue Auflistung der Funktionen und ihres Arbeitsprinzips findet sich im BASIC-Manual. Man muß bei diesen Funktionen allerdings immer einige Dinge beachten:

Es kann eine unübersichtliche Ebene herauskommen, wenn die Werte der y-Achse steigend sind (also  $2*y$  oder  $y^4$ ). Ist unumgänglich, daß bestimmte Abschnitte der y-Achse ansteigend sind, zum Beispiel bei einer Sinusfunktion, so sollte man eine genügend steile Vogelperspektive wählen, das heißt, den Winkel der y-Achse so weit wie nötig nach unten klappen, um die Übersicht zu wahren.

Als Faustregel gilt hier: Je weniger oft ein Teil des Graphes von nachfolgenden Linien überdeckt oder gekreuzt wird, desto dankbarer sieht dieses Objekt zum Schluß aus.

Im Sinne der Übersichtlichkeit gilt auch die nächste Empfehlung. Normalerweise schickt es sich nicht, einen Graph um mindestens 20 Einheiten pro Punkt die z-Achse hinaufzujagen, etwa bei der Formel  $z=50*y^2$ . Erstens wird der Graph innerhalb kürzester Zeit nicht mehr erblickt, weil er sich schon weit über dem verwendeten Fernseher befindet (nur rechnerisch!). Ebenfalls schlecht wirkt es, wenn man eine trigonometrische Funktion mit einem hohen Multiplikator versieht und dann eine flache Perspektive wählt. Dann fährt der Graph nämlich bei seiner Entwicklung einige dutzend mal über den ganzen Bildschirm, die Linien überkreuzen sich und kommen "unmotiviert" von den Bildschirmrändern ins Bild herein, da die Kurven weit über einen Schirmumfang hinausreichen. In diesem chaotischen Liniengewirr kann man zum Schluß unter Garantie keine Ebene mehr erkennen. Ein abschreckendes Beispiel ist in Bild 3 zu sehen.

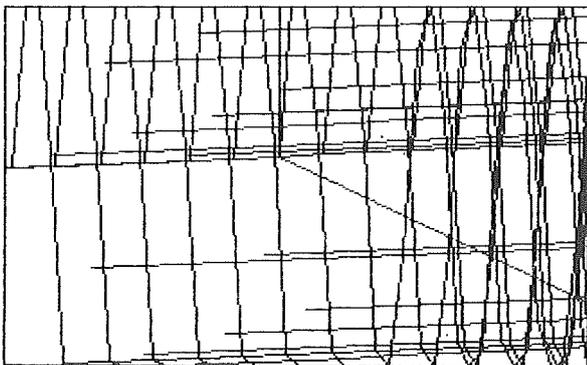


Bild 3

Bild wurde nach der Formel  $z=\sin(y^2)*3$  erstellt. Eingabeparameter: 120,80,3,30,2,4,1,2,.5

In dieser Serie können jedoch nur die nötigsten Grundlagen, einige Tips und Gedankenanstöße gegeben werden. Feste Formeln zum Erstellen von "schönen Ebenen am SVI-Computer" gibt es nicht. Es bleibt der Geduld und der Experimentierfreudigkeit jedes Einzelnen überlassen, ob letztendlich viele ansprechende Ebenen gesammelt werden können.

Hier noch einige Beispiele von Ebenen, die mit kartesischen Koordinaten erzeugt wurden:

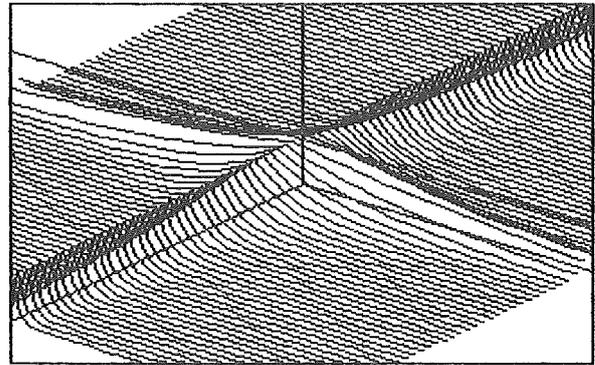


Bild 4

Bild wurde nach der Formel  $z=\exp(-\text{abs}(x^4))$  erstellt. Eingabeparameter: 128,96,30,20,4,4,1,2,.1

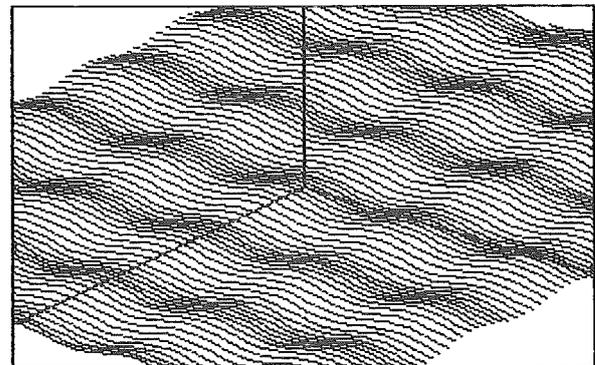


Bild 5

Bild wurde nach der Formel  $z=(\sin(x)+\cos(y))/4$  erstellt. Eingabeparameter: 128,96,30,20,10,4,1,2,.5

Zum Schluß dieser Folge noch ein Tip für alle, die keinen Ton am Datensichtgerät hören können. Wir haben eine kleine Erweiterung am Programm vorgenommen, daß nach Beendigung des Graphen ein Sprite in der rechten oberen Ecke zeichnet. Dieses Sprite wird auf keiner Hardcopy mitgedruckt und man kann es durch Betätigen der Taste ESC wieder zum Verschwinden bringen.

Zeile 1260 wird folgendermaßen abgeändert:

```
1260 GOTO 3300
```

Die Erweiterung:

```
3300 DATA 254,129,129,254,136,132,130,129
3310 A$="":RESTORE 3300:FOR I=1 TO8:READ
A:A$=A$+CHR$(A):NEXT
3320 IF C1=15 THEN C=1 ELSE C=15
3330 SPRITE$(0)=A$:PUTSPRITE0,(240,2),C,0
3340 A$=INKEY$:IF A$=""THEN3350ELSEGOTO3340
3350 A$=INKEY$:IF A$=""THEN3350
3360 IF ASC(A$)=27 THEN PUTSPRITE 0,(256,200),0,0 ELSE GOTO 3350
3370 GOTO 3370
```

Das Sprite stellt übrigens ein "R" wie "Ready" dar.

In der nächsten Folge zeigen wir noch einige Beispiele und gehen dann zur nächsten Eingabeart über.

```

*****
*                               *
*      EIN- und AUSGABE an      *
*      Peripheriebausteine      *
*                               *
*****

```

In dieser Ausgabe kann ich Ihnen leider kein neues Programm vorstellen, da ich kurz vor Redaktionsschluß einen nicht zu lösenden Fehler entdeckt habe. Das Programm, das sich mit dem Laden von Maschincodeprogrammen beschäftigen sollte, funktionierte zwar mit Kassette, nicht aber mit Diskette.

Somit folgt heute kein Programm, dafür aber einige Erklärungen im Umgang mit den peripheren programmierbaren Chips in unserem Computer.

Beginnen wir mit dem Videoprozessor. Der VDP belegt die I/O-Ports &H80, &H81, &H84 und &H85. Der Port &H80 dient zur Ausgabe eines Datenbytes in das VRAM, &H81 dient a) zum Setzen der Schreib/Leseadresse im VDP und b) zum Laden der VDP-Register mit Daten. Der Port &H84 dient zum Lesen eines Bytes aus dem VRAM und &H85 dient zum Lesen des Statusregisters des VDP.

Hier ein kleines Beispiel zur Ausgabe eines Datenblocks in das VRAM. Ab der Adresse, die im Registerpaar DE steht, wird eine bestimmte Anzahl von Bytes, die genaue Zahl steht in BC, in das VRAM geschrieben. HL enthält die Startadresse im VRAM.

```

DI
LD  A,L
OUT (81H),A
LD  A,H
OR  40H
OUT (81H),A
EX  (SP),HL
EX  (SP),HL
LOOP LD  A,(DE)
OUT (80H),A
INC DE
DEC BC
LD  A,B
OR  C
JR  NZ,LOOP
EI

```

Dieses kurze Programm lädt das VRAM ab HL mit BC Bytes aus dem normalen RAM des Computers. Die ersten Zeilen dienen zum Setzen des internen Zeigers des VDP. Nach jeder Schreib- oder Leseoperation wird er automatisch erhöht. Beim Schreiben muß das Bit 6 des Highbytes der Adresse gesetzt sein, um die Schreiboperation auch im VDP einzuleiten (es soll auch Modelle geben, bei denen das nicht mehr notwendig ist, Anm. der Red.). Die zwei EX (SP),HL dienen zur Zeitverzögerung. Mit LD A,(DE) holt der Computer das jeweilige Datenbyte aus dem RAM und mit OUT (80H),A wird es in das VRAM geschrieben.

Nun kommt die gleiche Routine zum Lesen von BC Bytes aus dem VRAM in das RAM ab DE. HL gibt wieder die VRAM-Adresse an.

```

DI
LD  A,L
OUT (81H),A
LD  A,H
AND 3FH
OUT (81H),A
EX  (SP),HL
EX  (SP),HL
LOOP IN  A,(84H)
LD  (DE),A
INC DE
DEC BC
LD  A,B
OR  C
JR  NZ,LOOP
EI

```

Sehen wir uns das Programm genauer an. Zuerst wird der interne Zeiger des VDP gesetzt, das Bit 6 ist dabei gelöscht. Bei LOOP beginnt das Einlesen aus dem VRAM, dann legt der Prozessor das Byte im RAM ab und erhöht DE. Das Ganze geschieht BC mal. Der interne Zeiger wird nach der Leseoperation IN A,(84H) um Eins erhöht.

Im folgenden möchte ich Ihnen zeigen, wie Sie Daten in die Register des VDP schreiben können.

```

DI
LD  A,B
OUT (81H),A
LD  A,C
OR  80H
OUT (81H),A
EI
RET

```

Dieses kleine Programm schreibt das Datenbyte im Register B in das durch C angegebene Register des VDP. Zuerst müssen Sie das Datenbyte ausgeben und dann durch Setzen des siebenten Bits dem VDP sagen, daß das Datenbyte in das Register, welches durch C adressiert wird, kommt. C kann einen Wert von 0 bis 7 annehmen, die Bedeutung der Bits der VDP-Register haben Sie im Heft 4/84 aufgelistet.

Die folgende Routine zeigt Ihnen mit dem Zeroflag des Z80, ob gerade zwei Sprites zusammengestoßen sind.

```

z.B.
...
CALL COLL
JP  Z,KILL_SHIP
...
...
COLL IN  A,(85H)
XOR 32
BIT 5,A
RET

```

In der Routine COLL wird das Statusregister des VDP gelesen. Das Bit 5 gibt an, ob zwei oder mehrere Sprites zusammengestoßen sind. Mit XOR 32 wird das Bit 5 invertiert, mit BIT 5,A wird das Zeroflag dem Zustand angepaßt. Sind also zwei zusammengestoßen, so ist das Zeroflag gesetzt. Ist Bit 5 nicht gesetzt, so ist das Zeroflag gleich 0.

Soweit der VDP. Nun kommt der Soundchip an die Reihe.

Der Soundchip belegt die I/O-Ports &H88, &H8C und &H90. Der Port &H88 dient alleine zur Adressierung der Register, &H8C dient zum Schreiben eines Datenbytes in das Register, und &H90 dient zum Lesen eines Registers.

```

DI
LD  A,C
OUT (88H),A
LD  A,B
OUT (8CH),A
EI
RET

```

Dieses kleine Programm schreibt das Datenbyte von B in das Register des PSG, welches durch C adressiert wird. Die Bedeutung der Register 0 bis 13 sind im Heft 6/84 erklärt. Die Register 14 und 15 werden des öfteren auch mit Port A und B angegeben. Die Bits des Registers 14 sind im Heft 5/84 erklärt, ebenso die des Registers 15 im Artikel 'Bank Switching' desselben Heftes.

Nun schnell weiter zum PPI, dem Ein-/Ausgabebaustein schlechthin.

Fortsetzung auf Seite 21

```
*****
*
*      Disk-Drive und 80-Zeichenkarte
*      für den Spectravideo SVI-728
*
*****
```

Für MSX-Rechner gibt es jetzt eine 80-Zeichenkarte von Spectravideo. Da jedoch eine Diskettenstation für den Betrieb der Karte gebraucht wird, haben wir uns gleich beides zusammen angesehen. Fangen wir mit dem Disk-Drive an:

An den SVI-728 kann man nur jeweils eine Diskettenstation anschließen. Diese verwendet doppelseitige 5 1/4 Zoll-Disketten mit einer Kapazität von 320 KByte (formatiert). Das Laufwerk wird direkt an die Hinterseite des Computers angeschlossen. Der SVI-728 hat ein eigenes Interface für die Disk. Die Stromversorgung wird leider nicht durch den Computer geregelt, man hat ein eigenes Netzteil zur Verfügung. An Software werden MSX-DOS und CP/M mitgeliefert. MSX-DOS ist für Spectravideo jedoch derzeit noch nicht erhältlich. CP/M stand uns zur Verfügung.

Interessant ist, daß das Laufwerk ein eigenes ROM eingebaut hat. Es befinden sich dort ein Ladeprogramm für CP/M und ein komplettes Disk-BASIC. Der Computer bootet also nicht von seinem MSX-ROM aus CP/M, sondern tut dies übers Diskettenlaufwerk-ROM. Ebenso ist das BASIC für Disketten verwendbar, wenn ein Laufwerk angeschlossen ist. Allerdings hat man vom diesen zusätzlichen Befehlen wenig, wenn keine Diskette eingelegt ist.

Wie schon oben erwähnt, ist eine Diskettenstation Voraussetzung für die Benutzung der 80-Zeichenkarte. Am Spectravideo wird diese Erweiterung in den Cartridge-Slot auf der Oberseite des Computers gesteckt. Da kein Expander verwendet werden muß (und auch nicht verwendet werden kann), hat man keine Möglichkeit, mehrere Erweiterungskarten auf einmal an den Computer anzuhängen. Man muß sich daher immer entscheiden, ob man eine RAM-Erweiterung verwenden will oder die 80-Zeichenkarte bevorzugt oder gar ein ROM-Modul zum Spielen braucht. Aber vielleicht wird sich auch das einmal ändern.

Angeschlossen werden kann die 80-Zeichenkarte nur an einen Monitor, ein Fernseher hat ein viel zu unscharfes Bild für die benötigte Auflösung. Im Betrieb mit CP/M werden die 80 Zeichen durch den Befehl STAT aktiviert. Mit "STAT CON:= UC1:" wird die Bildschirmausgabe auf den 80-Zeichenmonitor gelenkt, mit "STAT CON:= CRT:" kann man wieder 40 Zeichen verwenden. Der Befehl "STAT CON:= UC1:" gilt jedoch nur zum Umschalten während des Computerbetriebes. Das CP/M-Betriebssystem schaltet beim Bootvorgang automatisch die 80-Zeichenkarte, wenn vorhanden, ein.

Ähnlich dem SVI-328 kann übrigens auch der 728 auf beiden Schirmen operieren, da die Bildspeicher der 80-Zeichenkarte und des 40-Zeichenschirms unabhängig voneinander sind. Leider konnten wir noch keine Experimente mit dem MSX-DOS und der 80-Zeichenkarte starten, da keines vorhanden ist, aber ein ausführlicherer Bericht über die Diskettenstation folgt ebenso wie die weitere Verwendung der 80-Zeichenkarte.

```
*****
*
*      ROM-Adressen des SVI-728
*
*****
```

Soeben sind einige ROM-Adressen des MSX-ROMs bekannt geworden. Wir wollen sie Ihnen nicht vorenthalten.

Es gibt 8 Restarts beim Z80-Prozessor. Das MSX-BASIC hat diese natürlich auch genützt.

Der erste Restart ist bei Adresse 0 zu finden. Er führt einen kompletten RESET durch. Alle Register werden rückgesetzt und der Speicher wird kontrolliert und "gereinigt". Sämtliche Zeiger werden rückgesetzt. Der einzige Sinn für diese Routine ist ein Neustart des Computers.

Der zweite Restart fängt 8 Bytes später bei 08H an. Er wird zur Überprüfung einer BASIC-Zeile auf Syntax-Error gebraucht. Wird kein Fehler gefunden, arbeitet der Computer den RST 010H ab, andernfalls wird ein Syntax-Error generiert. Überprüft wird jeweils ein Zeichen, das Registerpaar HL zeigt auf dieses. Nach dem Test zeigt HL auf das nächste Zeichen, der Akkumulator enthält den soeben getesteten Code.

Auch RST 010H sollte vom Programmierer eher unbenutzt bleiben, denn auch dieser Restart wird vom BASIC-Interpreter gebraucht. Durch diesen Restart wird ein Zeichen einer BASIC-Zeile geholt.

RST 018H schreibt das Byte, das im Akkumulator steht, zum gerade angewählten Peripheriegerät hinaus.

RST 020H wird zum Vergleichen der beiden Registerpaare HL und DE gebraucht. Das C- und das Z-Flag des Prozessors ist wie nach einer CP (Compare)-Operation verändert.

RST 028H wird wieder vom BASIC-Interpreter verwendet. Hier prüft er, welche Variablentypen verwendet wird.

Die MSX-Computer können natürlich auch zwischen ihren Banken switchen. Genauso wie der SVI-328 hat auch der "728er" vier Banken, die allerdings im MSX-Jargon Slots genannt werden. Jede dieser vier Banken ist in vier weitere Pages zu je 16 KBytes gegliedert. Dies ist notwendig, da es MSX-Computer gibt, die nicht gleich 80 KBytes RAM in der Grundversion besitzen. Da kann man nicht 32-KByte-Pages "herumschalten", wenn nur 16 KBytes vorhanden sind. Diese Banken müssen auch angesprochen werden, der RST 030H ermöglicht dies auf einfache Weise. Da erste nachfolgende Byte gibt an, in welche Bank man möchte, die beiden Bytes danach die Adresse, wo genau das Sprungziel liegt. In dem einen Byte muß man folgendes angeben:

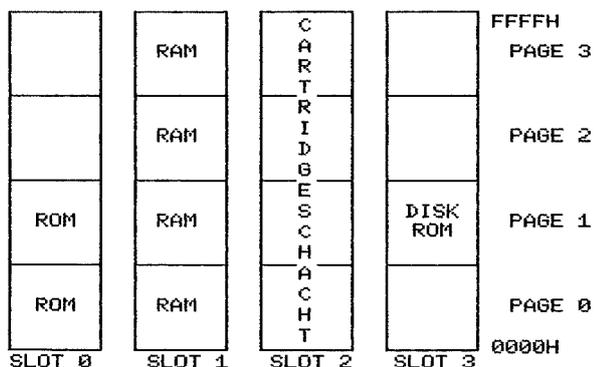
```
7 6 5 4 3 2 1 0
T n n n Z Z E E
```

T bedeutet ein Testbit, das aussagt, ob "Secondary Slots" verwendet werden. Diese Secondary Slots werden von einer der vier Banken angesprochen.

ZZ zeigt einen Wert an, der zwischen 0 und

3 schwankt, und gibt die Slot-Nummer des Secondary Slots an.

EE gibt ebenso einen Wert zwischen 0 und 3 an, der die Nummer der Primary Bank, einer der vier Banken im Computer, angibt.



n steht für "nichts" und kennzeichnet nicht gebrauchte Bits.

Im Anschluß an diesen Restart kommt noch eine Abbildung der vier Slots des SVI-728.

In Slot 0 befindet sich das MSX-ROM. Slot 1 beinhaltet das gesamte User-RAM, und Slot 2 ist für Cartridges vorgesehen. Das Disk-BASIC befindet sich in Slot 3.

RST 038H ist mit einer Interruptbehandlungsroutine belegt. Wenn diese Routine angesprochen wird, testet der Computer die Feuertasten eventuell angeschlossener Joysticks und die Tastatur auf eine gedrückte Taste. Die innere Uhr des Computers wird dekrementiert und die Daten im PSG werden auf den neuesten Stand gebracht.

Diese 8 Restarts werden durch RST XX angesprochen. In einer der nächsten Ausgaben sehen wir uns die weitere Sprungtabelle an, die bis zur Adresse 0160H geht. Diese Routinen werden mit einem CALL angesprochen und beinhalten auch einige nützliche Funktionen.

## Ein- und Ausgabe an Peripheriebausteine

Fortsetzung von Seite 19

Dieser Baustein belegt die Ports &H96, &H97, &H98, &H99 und &H9A. Der Port &H96 wird auch als Port C, &H99 als Port B und &H98 als Port A bezeichnet. Port &H9A dient zum Lesen des Ports C.

Die Bedeutung der Bits der Ports A, B und C haben Sie im Heft 6/84 aufgelistet. In diesem Heft haben Sie auch unsere Tastaturmatrix aufgezeichnet.

Das folgende Programm wandelt die oberste Reihe der Tastaturmatrix, die Reihe, die durch ein OUT (099H),0 adressiert wird, in eine Zahl um, deren Wert beim Rücksprung im Akku steht.

```

DI
PUSH BC
WAIT IN A,(9AH)
AND F0H
OR 00
OUT (96H),A
IN A,(99H)
LD B,8
LOOP SLA A
JR NC,EX_LOOP
DJNZ LOOP
JR WAIT
EX_LOOP DEC B
LD A,B
POP BC
EI
RET

```

Dieses Programm wartet solange, bis Sie eine Zifferntaste drücken. Bitte beachten Sie, daß nur die normalen Zifferntasten erlaubt sind und nicht der Zehnerblock.

Am Anfang ( WAIT ) liest das Programm den Port C ein, löscht die Tastaturreihenbits, und mit OR 0 wähle ich die Reihe 0 an. In diesem Beispiel ist OR 0 eigentlich unnütz, aber um zum Beispiel die Tasten der Reihe 5 einzulesen, müßten Sie OR 5 schreiben. Nach dem OUT (&H96),A wird im Port B angezeigt, welche Tasten gedrückt sind. Es ist dem hinzuzufügen, daß das Port B invertiert ist, eine gedrückte Taste also als 0 und eine

nicht gedrückte Taste als 1 angezeigt wird. Das Register B des Z80 dient als Schleifen- und Bitzähler. Nachdem Port B eingelesen wurde, wird der Akku solange nach links verschoben und B dekrementiert, bis eine gedrückte Taste gefunden wird ( das Carryflag ist dann nicht gesetzt ). Ist dies der Fall, springt der Computer zu EX\_LOOP, wo der Wert der gedruckten Taste in B-1 steht. Ist keine Taste gedrückt, springt das Programm zu WAIT.

Oder ein anderes Beispiel in Verbindung mit dem PSG zum Joystickeinlesen.

```

R_JOY DI
CP 2
LD A,14
OUT (88H),A
IN A,(90H)
JR C,R_JOY_1
RRA
RRA
RRA
RRA
AND 15
R_JOY_1 RES 4,A
LD C,A
IN A,(98H)
JR C,R_JOY_2
RRA
R_JOY_2 AND 16
OR C
AND 31
EI
RET

```

Sie rufen das Programm mit der Joysticknummer im Akku auf ( 1 oder 2 ) und bekommen im Akku die Richtungen und Feuerknöpfe zurück. Bit 0 bis 3 geben die Richtung an, Bit 4 gibt an, ob Feuer gedrückt ist oder nicht. Nach dem CP 2 am Anfang zeigt das Carryflag an, ob das Programm Joystick 1 oder 2 angewählt hat. Mit RRA verschiebt das Programm dann nur im Falle von Joystick 2 die Bits in die Positionen der Bits von Joystick 1. Bei R\_JOY\_1 liest das Programm die Feuerknöpfe ein, mit RRA schiebt es dann bei Carry gleich 0 das Bit 5 in das Bit 4. Schließlich verknüpft R\_JOY\_2 die Ergebnisse und kehrt zurück.

Ich hoffe, Ihnen mit dieser Übersicht über die I/O-Ports doch ein wenig geholfen zu haben.

```
*****
*
*      Graphik für alle Sprachen      *
*
*****
```

Der Spectravideo kann viele Sprachen sprechen. Allerdings sind in den wenigsten Graphikbefehle implementiert. Ein Clubmitglied hat sich nun des Betriebssystems CP/M angenommen und für die verschiedensten SVI-Sprachen ein mächtiges Graphikmodul entwickelt.

War es bis vor kurzem noch üblich, daß Mitglieder ihre Programme auf Diskette oder Kassette entwickelt haben, so gibt es jetzt eine neue Art des Speichermediums, das Modul. Das entsprechende Programm wird auf Disk entwickelt und im fertigen Zustand in ein EPROM gebrannt. Dieses EPROM setzt man dann auf eine Platine, die in den Modulsteckschacht paßt und fertig ist das Modul. So eine Entwicklungsmethode hat auch Rafael Razim angewandt.

Das Programm wird komplett als Modul geliefert, einige Demos sind beigelegt. Außer dem Modul ist noch eine kleine Startroutine notwendig, die auf die jeweilige Sprache angepaßt ist. Eine ausführliche Dokumentation ist beigelegt.

Man kann dieses Programm sogar schon als kleine Sprache sehen, da sie einen eigenen Befehlssatz hat und mit ihm eigene Programme geschrieben werden. Wir haben die Graphik anhand des Turbo-PASCALS getestet. Hier unsere ersten Versuche:

Zuerst wird das Modul in den Schacht gesteckt, dann der Computer eingeschaltet. Danach kann man problemlos Turbo-PASCAL laden. Es gibt folgende Befehle:

Text: schaltet Textmodus ein

Screen1: schaltet die hochauflösende Graphik ein (wichtigster Usermode)

Screen2: schaltet die niederauflösende Graphik ein

pset1(x,y,colour): setzt einen Punkt auf der Position x,y und mit der Farbe colour

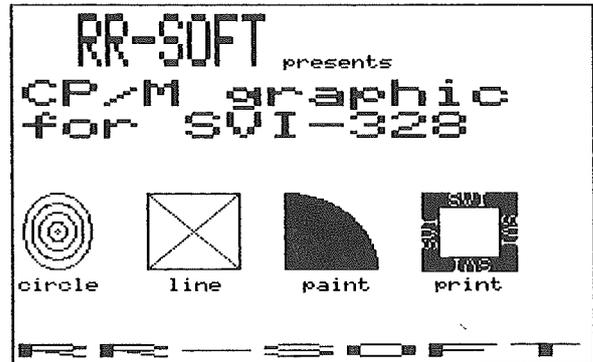
pset2(x,y,colour): setzt ebenso einen Punkt, allerdings im SCREEN 2-Modus

moveturtle(x,y): funktioniert wie das LOCATE im SVI-BASIC. Setzt den Graphik-Cursor auf die Position x,y, zeichnet jedoch keinen Punkt. Vorbereitungsbefehl für CIRCLE, LINE und PAINT.

circlepar1(cbegin,cend): legt Anfang und Ende des Kreises fest, Winkel wird in Altgrad, auf Wunsch auch in Neugrad eingestellt.

circlepar2(cxmax,cymax): legt die Verzerrung des Kreises zu einer Ellipse fest. cxmax ist der Zähler, cymax der Nenner des Bruches, mit dem die x-Koordinate multipliziert wird.

circle(radius,colour): setzt einen Kreis mit dem Radius 'radius' und der Farbe 'colour'. Die Mittelpunktkoordinaten wurden mit moveturtle festgelegt.



line(x,y,colour,switch): zeichnet eine Linie bis zum Punkt x,y mit der Farbe COLOUR. switch gibt Optionen an:

0EOH: Linie wird als Diagonale eines Rechtecks angesehen und nichtgezeichnet, sondern das errechnete Rechteck (analog zum LINE...B im BASIC).

0FOH: wie 0EOH, nur wird das Rechteck angemalt.

alle anderen Werte werden zum Zeichnen einer Linie verwendet.

printpar1(dir, xmax, ymax): dir gibt die Schreibrichtung an; xmax und ymax die Vergrößerungen des Textes.

printpar2(colour,x,y): x und y positionieren den Text, der mit PRINT geschrieben wird.

print(stp): schreibt den Text der Variable stp auf den Bildschirm.

In der Variablen stp können auch Escape-Sequenzen verwendet werden, die alle wichtigen Parameter während des Druckes ändern. Abgeschlossen wird der String mit ESC+'c'. Man kann übrigens auch entscheiden, ob man Inverssschrift haben möchte und ASCII-Codes oder Videocodes auf den Bildschirm schicken möchte.

Man kann natürlich die Namen der angegebenen Parameter selbst bestimmen. Diese Befehle können nun in jedes PASCAL-Programm eingebaut werden. Dabei gelten die PASCAL-Regeln. Während des Betriebes hat man übrigens immer die Möglichkeit, die Folge der Parameter in einem Befehl zu überprüfen. Es muß nämlich bei Verwendung der Graphik ein PASCAL-File am Anfang jedes Programms stehen, welches die Routinen im Modul aufruft. In diesem Programmteil stehen auch alle Befehle samt zugehöriger Syntax.

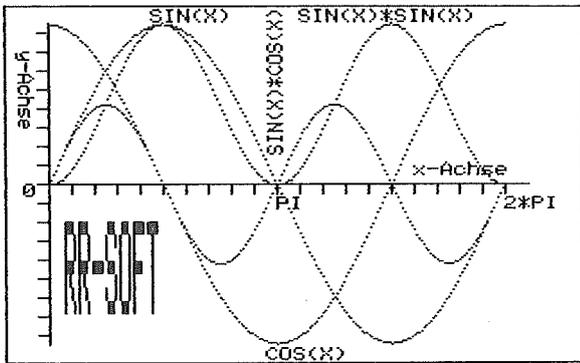
In diesem Programmteil kann man auch sehen, welche Variablentypen wo verlangt werden. Generell kann man jedoch sagen, daß als genauester Variablentyp Integer verwendet wird, Byte jedoch am häufigsten vorkommt.

Das Programm hat natürlich einige Features. So kann man den Text sowohl in die X- als auch in die Y-Richtung vergrößern. Ebenso läßt sich der Text in 4 verschiedenen Rich-

tungen ausrichten. Daraus ergeben sich vielfältigste Möglichkeiten für die Textgestaltung. Schade ist, daß ein Text, der über einen Bildschirmrand geht, in der gleichen Zeile wieder auf der anderen Seite erscheint. Es wird jedoch bald eine verbesserte Version herauskommen, die den Text eine Zeile nach unten verschoben ausgibt, wenn er über eine Bildschirmbreite geht. Außerdem arbeitet das Programm am 40-Zeichenschirm. Der 80-Zeichenschirm bleibt gleichzeitig für Arbeiten am Programm oder sonstige Editiermöglichkeiten frei.

Ebenso sind zwei Hardcopies implementiert. Eine arbeitet ganz normal, während die andere in Grauwerten die einzelnen Farben wiedergibt und das ganze Bild vergrößert. Sprites kommen für die verkaufsfertige Version ebenso wie die Unterstützung des Graphiktablets dazu.

Das Programm ist eine sehr interessante Entwicklung. Es ermöglicht ein bequemes Zeichnen mit vielen Features in den verschiedensten Sprachen.



Die Graphikerweiterung ist bei Rafael Razim erhältlich und kostet rund 1000 Schilling. Die Demos während des Testberichtes sind Hardcopies von erzeugten Graphiken und sollen die Qualität zeigen, in der Zeichnungen möglich sind.

```

10 *****
11 *****
12 *****
13 *****
14 ***** TELEFONNUMMERN - DATEI *****
15 *****
16 ***** 19850429 by M.TOTH *****
17 *****
18 *****
19 *****
20 *****
21 '
22 DATENFELD DIMENSIONIEREN
23 '
24 SP=INT(FRE(0)/12-100)
25 DIMA$(SP,3)
26 COLOR4,15,15:WIDTH39:TROFF:SCREEN0,0:CLS
27 STOPON:ONSTOPGOSUB89
28 KEY1,CHR$(11)+CHR$(13)+CHR$(13)+CHR$(13):
KEY5,"RUN"+CHR$(13)
29 RESTORE30:READP$
30 DATA$SAVE"TEL"
31 LOCATE11,10,0:PRINT"Bitte Warten"
32 '
33 'EINLESEN DER DATEN AUS DATA-ZEILE
34 '
35 I=I+1

```

```

36 FORA=0T03:READA$(I,A):NEXT
37 IFA$(I,0)<"ZZ"THEN35
38 I=I-1:CLS
39 '
40 'SUCHKRITERIUM AUSWAEGHLEN
41 '
42 PRINT"Waehlen Sie ":PRINT:PRINT
43 PRINT"A      Ausgabe aller Daten":PRINT
44 PRINT"      Sortieren nach ":PRINT
45 PRINT"B      Zunamen":PRINT
46 PRINT"C      Vornamen":PRINT
47 PRINT"D      Telefonnummern":PRINT
48 PRINT"E      Anfangsbuchstaben":PRINT
49 PRINT"F      Eingabe von Daten":PRINT
50 PRINT"G      Loeschen von Daten":PRINT
51 PRINT"H      Aussteigen aus dem Programm"
52 A$=INKEY$:IFA$=""THEN52
53 IFA$="A"ORA$="a"THEN67ELSEIFA$="B"ORA$="b
"THENA=OELSEIFA$="C"ORA$="c"THENA=1ELSEIFA$=
"D"ORA$="d"THENA=3ELSEIFA$="E"ORA$="e"THEN66
ELSEIFA$="F"ORA$="f"THEN73ELSEIFA$="G"ORA$="
g"THEN83ELSEIFA$="H"ORA$="h"THEN89ELSE52
54 '
55 'EINGABE UND SUCHEN
56 '
57 CLS:LOCATE,,1:LINEINPUT" Eingabe : ";G$:L
OCATE,,0:CLS
58 FORB=1TOI
59 IFB$<>A$(B,A)THEN61
60 E=E+1:PRINTA$(B,0);" ";:PRINTTAB(14)A$(B,
1);" ";:PRINTTAB(24)A$(B,2);" ";:PRINTTAB(30
)A$(B,3):E=E+1:IFE=22THENF=F+1:FORE=1TO2000:
NEXT:E=0:CLS
61 NEXT
62 IFE+F=0THENPRINTG$;" nicht gefunden !"
63 PRINT:PRINT"Druucken Sie eine Taste um fo
rtzusetzen";
64 IFINKEY$=""THEN64
65 E=0:F=0:G$="":CLS:GOTO42
66 CLS:LOCATE,,1:LINEINPUT" Eingabe : ";G$:L
OCATE,,0
67 CLS:L=LEN(G$):FORB=1TOI:IFLEFT$(A$(B,0),L
)=G$THEN:PRINTA$(B,0);" ";:PRINTTAB(14)A$(B,
1);" ";:PRINTTAB(24)A$(B,2);" ";:PRINTTAB(30
)A$(B,3):E=E+1:IFLEFT$(A$(B,0),1)<>LEFT$(A$(
B+1,0),1)ANDE<>22THENPRINT:E=E+1
68 IFE=22THENF=F+1:FORE=1TO2000:NEXT:E=0:CLS
69 NEXT:GOTO62
70 '
71 'EINGABE VON NEUEN DATEN
72 '
73 IFSP-I<2THENCLS:PRINT"Der Speicherplatz i
st leider zu Ende. Geben Sie bitte keine ne
uen Daten mehr ein!":PRINT
"Der Systemverwalter";:PRINT:GOTO63
74 CLS:LOCATE,,1:LINEINPUT"      ZUNAME : "
;A$(0,0):PRINT
75 LINEINPUT"      VORNAME : ";A$(0,1):PRINT
76 LINEINPUT"      VORWAHL : ";A$(0,2):PRINT
77 LINEINPUT"TELEFONNUMMER : ";A$(0,3):LOCAT
E,,0:B=0
78 B=B+1:IFA$(0,0)>A$(B,0)THEN78ELSEIFA$(0,0
)=A$(B,0)ANDA$(0,1)>A$(B,1)THEN78
79 CLS:PRINTB*10+85;:PRINT"DATA";A$(0,0);",
";A$(0,1);",";A$(0,2);",";A$(0,3):GOTO88
80 '
81 'LOESCHEN VON DATEN
82 '
83 CLS:LOCATE,,1:LINEINPUT"      ZUNAME : "
;A$(0,0):PRINT
84 LINEINPUT"      VORNAME : ";A$(0,1):LOCAT
E,,0
85 B=0
86 B=B+1:IFB>ITHENCLS:G$=A$(0,0):GOTO62
87 IFA$(B,0)=A$(0,0)ANDA$(B,1)=A$(0,1)THENCL
S:PRINTB*10+90ELSE86
88 PRINT"RENUM100,95":PRINT:PRINT"RUN":PRINT
:PRINT"Wenn Sie Daten ein- oder ausgeben
wollen, druecken Sie bitte auf Fun
ktion 1, ansonsten auf Funktion 5!":END
89 LOCATE,,1:CLS:KEY1,P$+CHR$(13):PRINT"Verg
essen Sie nicht das neue Programm abzuspic
hern ! A Funktion 1 ü":PRINT:PRINT"
1985 by M.TOTH":END
90 '
91 'DATA-ZEILEN MIT DATEN
92 '
100 DATAHUBER,HANS,0222,5685416
110 DATAZ,,

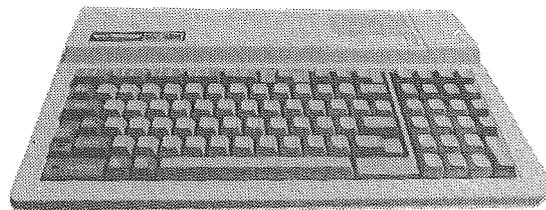
```

**Der tragbare  
Bondwell 14**



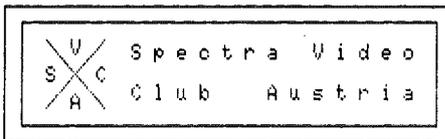
**SVI-328  
und Peripherie**

**SVI-728 und MSX**



# Computer-Studio

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



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 Spectravideo-Computer
- kostenloser Bezug der monatlich erscheinenden Clubzeitschrift SVI-JOURNAL
- verbilligte Angebote von Spectravideo-Produkten

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

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

## IMPRESSUM:

Chefredakteur: Gerhard Fally

Ständige freie Mitarbeiter: Constantin Gagnas, Philipp Ott, Rafael Razim, Wolfgang Rotschek, 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.