

SV

JOURNAL

Die Zeitschrift des Spectra Video Club Austria



INHALT

3	SPECTRAVIDEO-BASIC
5	Wichtige ROM-Adressen
6	SV-Hardware
10	Z80 - Programmieren in Assembler
13	TIPS & TRICKS
14	GRAFIK AUF DEM SPECTRAVIDEO SV-328
16	Programme

Heft 1/84

S 15.-

Liebes Clubmitglied!

Ob ein Club gut funktioniert, hängt oft von der Kommunikation zwischen den Interessenten ab. Gute Dienste kann hier eine Zeitschrift leisten. Deshalb haben einige Mitglieder des Spectra Video Club Austria das SVi-Journal gegründet.

Was im Konkreten wird die Zeitschrift bieten? Da die Computer von Spectravideo ein sehr umfangreiches Basic besitzen und Befehle haben, die in keinem Buch erklärt und im Spectravideo-Handbuch schlecht beschrieben sind, gibt es eine Serie über die Basic-Befehle des SV-318 und SV-328. Weiters haben wir noch, zum besseren Verständnis des Computers, einen Maschincode-Kurs und eine Hardware-Beschreibung in die Zeitung hineingenommen. Damit noch nicht genug! Es gibt ja immer wieder Leute, die Programmteile erfinden, mit denen man herrlich den Computer überlisten kann. Solche Programmteile finden Sie unter der Rubrik "Tips & Tricks".

Wenn Sie wissen wollen, wer was wann im Club macht, dann können Sie bei "Clubaktivitäten" nachschauen. Natürlich bringen wir auch 2-3 Programme des Monats, die Sie ablaufen lassen können, falls Sie zu wenig Zeit und Lust haben, selber Programme zu entwickeln. Wie in jeder besseren Zeitschrift finden Sie auch Rätsel und Witze in den Seiten des SVi-Journals.

Da es aber immer wieder Spezialwünsche gibt, möchten wir Sie, liebes Clubmitglied, bitten, kommen Sie zu einem Mitarbeiter des SVi-Journals und bringen Sie uns Ihre Programme oder Ihre Beschwerden und Wünsche über den Club, den Computer und nicht zuletzt über die Zeitschrift selber!

Ihr SVi-Journal-Chefredakteur Gerhard Fally!

CLUBNACHRICHTEN***NEUIGKEITEN***CLUBNACHRICHTEN***NEUIGKEITEN***

Die nächsten Clubabende :

- Sa, dem 23. Juni 1984, ab 17 Uhr (Ende ca. 22 Uhr)
- Mi, dem 4. Juli 1984, ab 19 Uhr (Ende ca. 22 Uhr)
- Sa, dem 21. Juli 1984, ab 17 Uhr (Ende ca. 22 Uhr)
- Mi, dem 1. August 1984, ab 19 Uhr (Ende ca. 22 Uhr)

im Computer-Studio, 1040 Wien, Paniglgasse 18-20.
Gäste sind willkommen.

Die Computer von Spectravideo sind BILLIGER geworden!

SV-318	5490.- inkl. MWSt.
SV-328	7990.- inkl. MWSt.

Für Clubmitglieder gibt es beim "COMPUTER-STUDIO WEHSNER" Sonderangebote: Beim Kauf eines Spectravideo-Computers erhält man den Mono-Kassettenrekorder gratis dazu. Alles andere ist um 5% unter dem Kaufpreis zu haben.

SPECTRAVIDEO-BASIC

Wußten Sie, wie viele Befehle und Funktionen der SV328 hat? Gezählte 206, wobei die Makrobefehle von "DRAW", "PLAY", "SOUND" usw. nicht mitgezählt wurden. Da kann es schon vorkommen, daß ein Befehl in keinem BASIC-Lehrbuch zu finden und im SV-Handbuch schlecht erklärt ist. Aus diesem Grund wurde diese Serie geschrieben. Sie soll kein Kurs in der üblichen Art sein, in dem man lernt, was ein Flußdiagramm, Variablen oder ähnliches sind. Dafür gibt es ausführliche Bücher. Ich möchte jeden Befehl und jede Funktion einzeln erklären und eventuell Feinheiten zu dem einen oder anderen Befehl packen, die nicht im Handbuch stehen.

Anfangen will ich mit dem Befehl "PRINT". Mit ihm kann man Daten auf den Bildschirm schreiben. "PRINT" kann übrigens durch ein '?' abgekürzt werden. Welche Formate kann dieser Befehl haben? Wenn Sie den Wert einer Variablen (z.B. A oder A\$) auf dem Bildschirm sehen wollen, müssen Sie die Variable nach dem "PRINT" ansetzen (z.B. PRINT A oder PRINT A\$). Um zwei oder mehr Strings auf den Bildschirm zu bringen, können Sie entweder das ';', das ',', das '"' oder ein Space schreiben (z.B. PRINT A;A\$,B;B\$A). Ein ',' bedeutet, daß die Anfänge der beiden Strings 15 Zeichen voneinander entfernt sind. Ist der erste String länger als 15 Zeichen wird der zweite String in die nächste Zeile geschrieben. Bei einem ';' wird kein Platz zwischen den Strings gelassen, ebenso bei einem '"' oder bei einem Space. Beachten Sie bitte, daß bei numerischen Variablen vor und nach der Zahl immer ein Leerzeichen zusätzlich ausgeschrieben wird. Wird ein ',' oder ein ';' nach der letzten Variablen in einer "PRINT"-Anweisung geschrieben, wird die nächste "PRINT"-Anweisung an die letzte angehängt. Hier gilt wieder: ';' = ohne Zwischenraum und ',' = 15 Zeichen von Anfang zu Anfang.

Wenn Sie Direkttext ausschreiben lassen wollen, müssen Sie den Text lediglich zwischen zwei '"' stellen (z.B. ?"HALLO"). Das SV-Basic macht es Ihnen sogar möglich Funktionen in den "PRINT"-Befehl einzuschleusen. Sie können z.B. PRINT SIN (X) programmieren und der Computer rechnet den Sinus von X aus und zeigt Ihnen danach an. Natürlich kann man Direkttexte, Strings usw. in einem "PRINT"-Befehl kombinieren (z.B. PRINT"HALLO"A\$"IHNNEN"SIN(X)).

Will man formatierte Bildschirmausgaben, kann man dies mit "PRINT USING" erreichen. Der Befehl wird in zwei Sektionen eingeteilt, die erste Sektion ist ein String, der über die Formatierung der Daten, die ausgedruckt werden sollen, Auskunft gibt, die Zweite ist der eigentliche "PRINT"-Befehl, der genauso funktioniert, wie das normale "PRINT", mit der Einschränkung, daß zwischen zwei Variablen kein Space oder ein '"' stehen darf. Außerdem wird zwischen den beiden formatierten Variablen am Bildschirm kein Platz gelassen, egal ob ein ';' oder ein ',' im "PRINT"-Befehl steht. Wie sieht der Formatierungsstring nun eigentlich aus? Zuallererst kann man diesen String in Form von Direkttext, also z.B. "###" oder in Form von Stringvariablen, z.B. F\$, einsetzen. Außerdem muß man zwischen numerischen und alphanumerischen Variablen unterscheiden. Steht in der zweiten Sektion eine numerische Variable, gibt man mit '#' die Anzahl der Stellen an, die man anzeigen will. Gesetzt der Fall, man will 5 Stellen, ohne Kommastellen, anzeigen, dann muß man "#####" eingeben. Ist die Zahl dann kleiner als 5 Stellen, wird sie so gesetzt, daß die Einerstelle ganz rechts steht. Vor der Zahl wird

der Formatierungsplatz mit Blanks ausgefüllt. Will man zusätzliche Kommazahlen anzeigen, setzt man zwischen die '#' einen Punkt (z.B. PRINT USING"###.###";F). Die Zahl wird dann nach diesem Punkt ausgerichtet (z.B. PRINT USING"###.###";34.239 wird so ausgedruckt:' 34.239'). Sind weniger Kommastellen da, als vom Format her möglich, werden die Stellen links an das Komma angehängt und der Rest mit Nullen aufgefüllt (z.B.:###.####: 34.23900). Will man zwischen zwei Variablen, vorausgesetzt sie kommen im selben "PRINTUSING"-Befehl vor, keine Blanks, sondern Sterne, muß man am Anfang des Formatierungsstrings '**' setzen, will man vor der Zahl ein Dollarzeichen '\$' haben, muß man '\$\$' einsetzen. '**\$' kombiniert die beiden Fälle. Das Symbol '^'^'^', angeschlossen an die Formatierung, erzeugt Exponentialschreibweise der Variable. Dabei werden die Mantisse so groß, wie es das Format erlaubt, gemacht und die 10er-Potenzen berechnet und angezeigt. Zuletzt muß noch die Vorzeichenregelung geklärt werden. Wird kein '+' oder '-' im Formatstring verwendet, dann wird nur ein '-' vor einer negativen Zahl ausgegeben und ein Blank vor einer positiven. Will man vor einer positiven Zahl ein '+' und ein '-' vor einer negativen, muß man ein '+' an den Anfang des Strings stellen. Ein '-' am Anfang bewirkt ein Ausdrucken eines Minus vor jeder Zahl, unabhängig vom Vorzeichen. Ein '+' am Ende des Strings bewirkt, daß das Vorzeichen hinter der Zahl ausgegeben wird, ein '-' am Ende generiert ein Minuszeichen hinter einer Zahl, wenn diese negativ ist.

Bei alphanumerischen Variablen gibt es nur zwei verschiedene Symbole für den Formatstring. Das eine ist ein '!'. Es bewirkt, daß nur das erste Zeichen der Variable ausgedruckt wird. Das andere ist ein '\'. Danach folgen in der Formatierungsanweisung um zwei Blanks weniger, als Zeichen ausgedruckt werden sollen. Ein '\' schließt den String ab. Ist die alphanumerische Variable kürzer als im Formatstring vorgesehen, wird sie linksbündig abgelegt, der Rest mit Blanks gefüllt. Ist sie größer, wird der überstehende rechte Teil abgeschnitten.

Man kann numerische und alphanumerische Formatsymbole in einem String kombinieren, dann muß man allerdings darauf achten, daß den Variablen die richtigen Format-Anweisungen zugeordnet werden (z.B. PRINTUSING"###,\ \";A\$,A ist falsch;A gehört vor A\$). Sie können übrigens auch Direkttext in den Formatstring schreiben. Er wird so ausgedruckt, wie Sie ihn programmieren. Bitte beachten Sie aber dabei, daß dieser Text nicht eine Formatierungsanweisung für eine Variable teilt. Der Computer würde aus dieser einen Anweisung zwei falsche machen (z.B. PRINT USING "###WER#";A ist falsch;"####WER" ist richtig).

Weitere mit "PRINT" verwandte Befehle sind "LPRINT" und "LPRINTUSING". Sie funktionieren wie "PRINT" und "PRINTUSING", nur das nicht der Bildschirm, sondern der Drucker angesprochen wird. Außerdem wird angenommen, daß der Drucker 132 Zeichen pro Zeile drucken kann. "PRINT#" und "PRINT#USING" werden unter dem Kapitel DATEI & DATEIVERARBEITUNG besprochen.

Genauso wichtig wie die Ausgabe ist die Eingabe. Der bekannteste und wichtigste Eingabebefehl ist "INPUT".Mit "INPUT" können numerische und alphanumerische Variable eingegeben werden (z.B. INPUT A(\$)). Wenn Sie zwei oder mehr Variable in einem "INPUT"-Befehl verbinden wollen, müssen Sie zwischen die Variablen Beistriche setzen. Im laufenden Programm wird bei jedem "INPUT" ein Fragezeichen am Bildschirm sichtbar, die Variablen können dann eingegeben werden. Sollten mehrere in einem "INPUT"-Befehl ste-

hen, kann man bei der Eingabe der Variablenwerte zwischen den Werten Beistriche setzen, wird der erste mit "Enter" abgeschlossen, erscheinen zwei Fragezeichen am Bildschirm. Sie können dann die nächsten Variablen eingeben.

Vor die Variablen kann man übrigens Direkttext stellen, eingekapselt in zwei "'", gefolgt von einem Semikolon. ACHTUNG! Sollte der Fall mit den zwei Fragezeichen eintreten, dann wird der Text kein zweites Mal vor die beiden Zeichen gestellt. - "INPUT#" wird im Kapitel DATEI & DATEIVERARBEITUNG besprochen.

Ein weiterer sehr wichtiger Befehl ist das "GOTO". Mit "GOTO" kann man zu einer Zeile springen, unabhängig von irgendwelchen Bedingungen oder Variablenwerten. Man nennt "GOTO" deshalb auch unbedingten Sprungbefehl. Im wesentlichen besteht dieser Befehl nur aus dem GOTO und einer Zahl, die den Wert einer Zeilennummer hat, Variablen und Funktionen sind nicht zulässig.

Ein weiterer wichtiger Sprungbefehl ist "GOSUB". Mit "GOSUB" kann man in ein Unterprogramm springen, welches mit "RETURN" abgeschlossen wird. Hinter "GOSUB" darf wieder nur eine Zahl stehen. Der Unterschied zwischen den beiden Sprungbefehlen besteht darin, daß sich der Computer bei "GOSUB" die Zeilennummer merkt, in der das "GOSUB" steht und bei Erreichen von einem "RETURN" zu der Zeilennummer zurückspringt und in der nächstfolgenden Zeile weiterarbeitet. Wenn ein Programmteil öfters gebraucht wird, bietet das "GOSUB-RETURN" einen großen Vorteil, weil das "RETURN" keine bestimmte Zeilennummer anspringt, sondern immer nur die nach dem letzten "GOSUB".

WICHTIGE ROM-ADRESSEN

4198H	Zeichengenerator (bis 4550H)
02C9H	Tabelle BASIC-Wörter (bis 0586H)
05D6H	Tabelle Fehlermeldungen (bis 084BH)
0018H	Ausgabe eines Zeichens auf dem Bildschirm (RST 18H)
003EH	Zeichen von Tastatur lesen
0044H	Druckertreiber
0047H	Screen 0
004AH	Screen 1
004DH	Screen 2
0053H	Screen ,0 (Funktionstastenanzeige aus)
0056H	Screen ,1 (Funktionstastenanzeige ein)

Wir nehmen Informationen über das ROM gerne zur Weiterleitung an unsere Leser entgegen.

```

*****
*      Das richtige BASIC-Lehrbuch zum Spectravideo-Computer      *
*                                                                 *
*      Werner Chmel          324 Seiten, 300 Abbildungen          *
*      Basic-Kompodium       S 296,-                              *
*                                                                 *
*      Neben den reinen Syntax-Beschreibungen und zahlreichen     *
*      Beispielprogrammen werden auch detaillierte Program-     *
*      miertechniken vorgestellt. Der beschriebene Befehls-     *
*      satz entspricht dem Erweiterten Microsoft-Basic der     *
*      Spectravideo-Computer (nur wenige Ausnahmen).           *
*                                                                 *
*      erhältlich im Computer-Studio, 1040 Wien, Paniglg.18-20   *
*****

```

S V - H a r d w a r e

Die Hardware der Spectra Video Computer in Fortsetzungen

1. F o l g e

In dieser Serie wollen wir dem Leser die Hardware der Spectra Video Computer näherbringen. Wenn in dieser Serie vom Computer die Rede ist, bedeutet dies, daß es sich um Informationen handelt, die sowohl für den SV-318 als auch für den SV-328 Gültigkeit haben. Unterschiede werden ausdrücklich vermerkt.

Der Spectra Video Computer ist mit dem Mikroprozessor Z80A der Firma Zilog ausgestattet. Der Prozessor arbeitet mit einer Taktfrequenz von 3,58 MHz und steuert das gesamte System.

Bevor wir uns näher mit der Systemarchitektur befassen noch ein Blick auf periphere Geräte und Erweiterungen, die an den Computer anschließbar sind. Eine graphische Übersicht finden Sie auf der folgenden Seite.

Direkt an den Computer anschließbar sind ein bis zwei Joysticks und ein Datenrecorder. Es werden derzeit zwei verschiedene Datenrecorder von Spectra Video angeboten: der SV-903, ein Stereo-Kassettenrecorder mit eingebautem Mikrophon (mit Ein/Aus-Schalter) und der SV-904, ein Mono-Recorder.

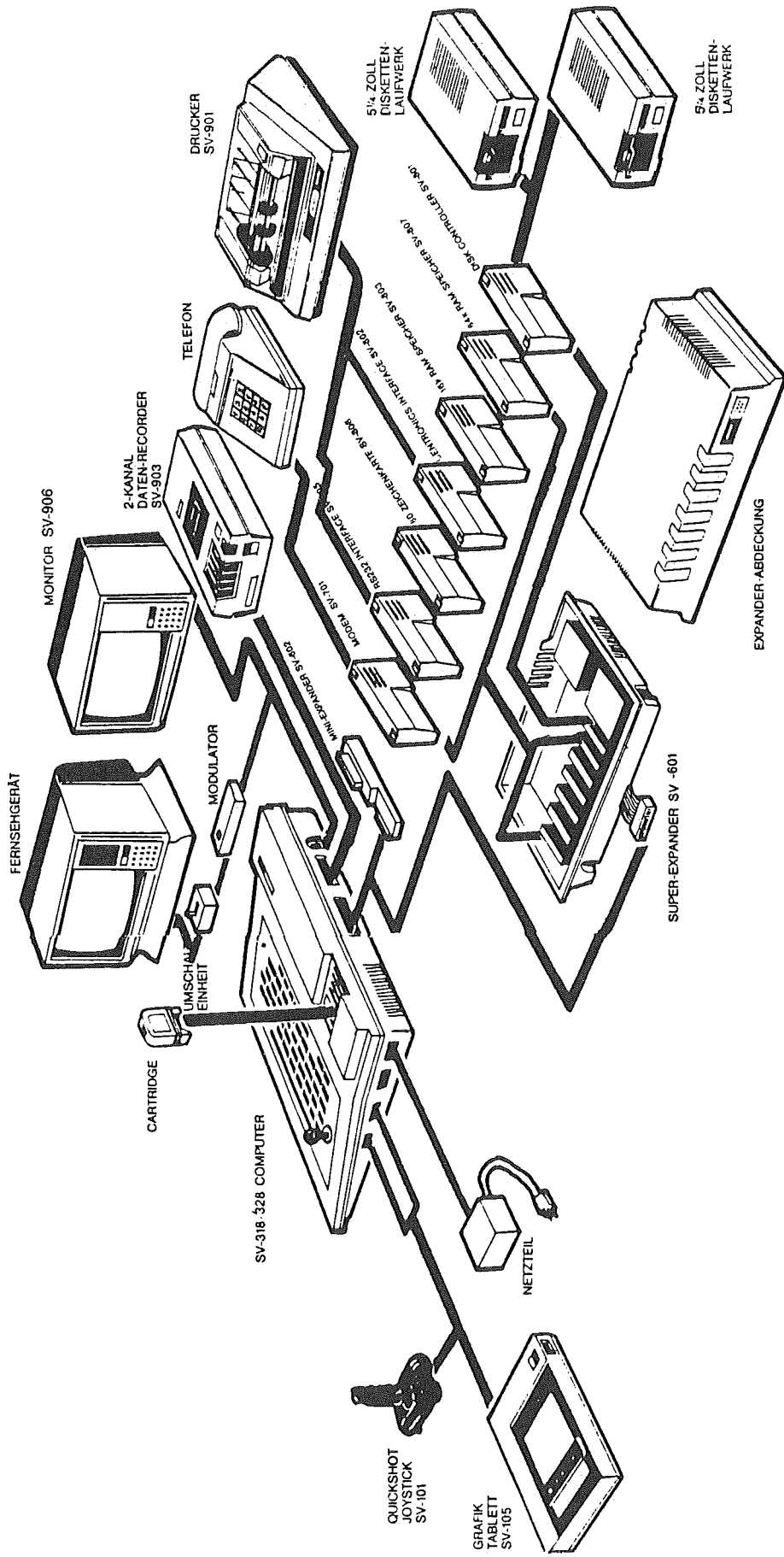
An die Joystick-Anschlüsse kann auch das Grafik-Tablet SV-105 angeschlossen werden.

Ein Steckschacht steht an der Oberseite des Computers (mit einem Klappdeckel verschlossen) zum Einstecken von ROM-Cartridges zur Verfügung. Diese Anordnung ist vorteilhaft, weil der Zugang zum Steckschacht auch bei angesteckten Erweiterungen möglich ist.

Weiters befindet sich ein fünfpoliger Videostecker an der Rückseite des Computers zum Anschluß des beigegebenen UHF-Modulators und über eine ebenfalls beigegebene Antennenweiche (umschaltbar Computer und TV-Antenne) an den Fernsehapparat oder zum direkten Anschluß eines Videokabels an den Videoeingang des TV-Apparats. Es kann auch ein Farbmonitor (muß über Eingang für Video-Signal verfügen) oder ein monochromer Monitor (Farben werden als unterschiedliche Grauwerte dargestellt) angeschlossen werden.

Der Systembus ist an der Rückseite des Computers als fünfzigpolige Platinensteckerleiste nach außen geführt. Zum Anschluß weiterer Peripheriegeräte können sogenannte Expander angeschlossen werden. Derzeit stehen zur Verfügung: Der Super-Expander SV-601 zum Anschluß von maximal 7 Erweiterungskarten, der Mini-Expander SV-602 zum Anschluß einer Erweiterungskarte (der Anschluß des Floppy-Controllers ist in diesem Fall nicht möglich), weiters die Super-Expander mit Metallgehäuse SV-605 und SV-605A.

Alle Super-Expander sind mit einem eigenen Netzteil ausgestattet. An Steckkarten stehen zur Verfügung: Centronics-Interface zum Anschluß eines Druckers, eine RS232-Schnittstelle, eine 80-Zeichenkarte, eine Disk-Controller-Karte (wird mit CP/M 2.2 Systemdiskette und Disk-BASIC-Diskette geliefert), weiters 64 KByte-RAM-Karte und 16 KByte-RAM-Karte. Letztere kann nur beim SV-318 zur Erweiterung des Speicherbereichs in der Bank 0 eingesetzt werden.



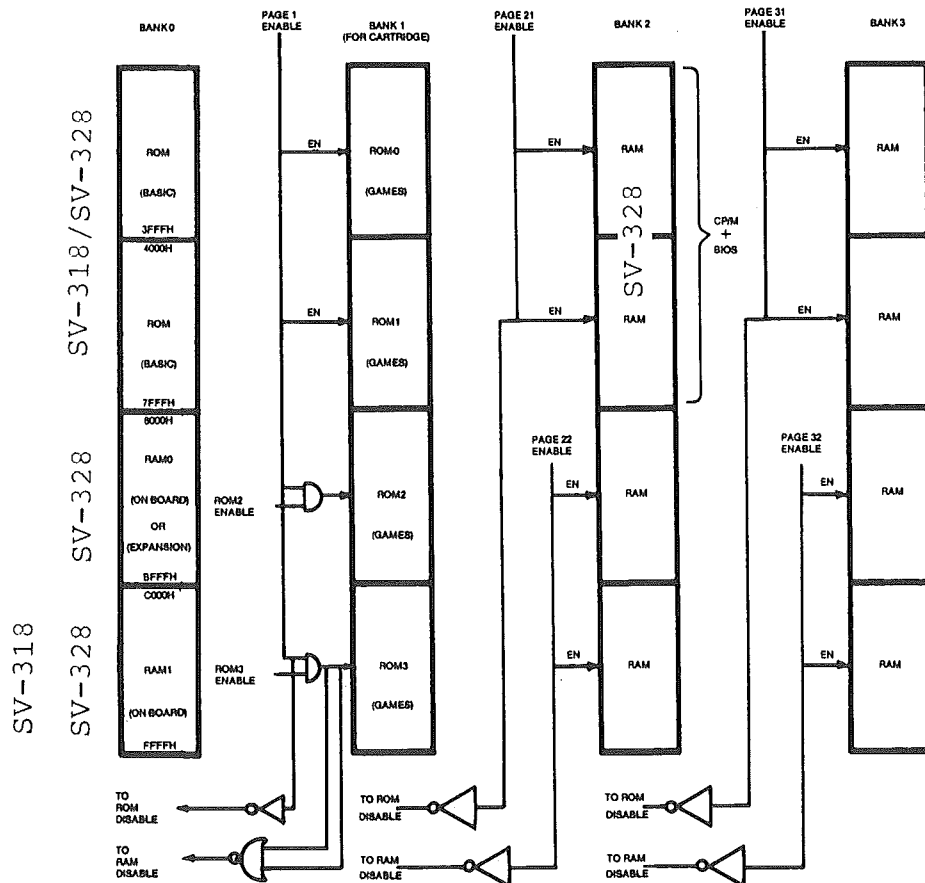
Beim Modell SV-605 sind Floppy-Controller und Centronics-Schnittstelle zum Anschluß eines Druckers bereits eingebaut, weiters ein Diskettenlaufwerk 5 1/4 " (beim SV-605A zwei Laufwerke).

SV-318 und SV-328 unterscheiden sich nur durch die Tastatur und den eingebauten Arbeitsspeicher. Der SV-318 ist mit einer geräumigen Gummitastatur ausgestattet. Bemerkenswert ist eine runde Steuerplatte an der rechten Seite, mit deren Hilfe der Cursor in acht Richtungen gelenkt werden kann. Ein roter Joystick kann in diese Platte eingesetzt werden.

Der SV-328 besitzt eine schreibmaschinenähnliche Tastatur mit einem Zehnerblock mit integrierten Cursorsteuertasten und Rechentasten. Der Cursor kann übrigens auch hier durch gleichzeitiges Drücken zweier Steuertasten diagonal bewegt werden. Dies wird durch ein eingebautes sog. "Key-roll-over" ermöglicht: es können gleichzeitig mehrere Tasten gedrückt werden, die dann auch auf dem Bildschirm ausgegeben werden.

Die folgende Darstellung zeigt Ihnen die Speicherbelegung. Beim SV-318 und SV-328 befindet sich in der unteren Hälfte der Bank 0 der BASIC-Interpreter (32 KByte). Im SV-318 ist ein 16 KByte-RAM-Bereich eingebaut (Adreßbereich C000-FFFFH) und ein Video-RAM von ebenfalls 16 KByte. Im SV-328 sind außer dem Video-RAM 64 KByte RAM eingebaut und zwar befinden sich 32 KByte in der oberen Hälfte von Bank 0 und 32 KByte in der unteren Hälfte von Bank 2.

Letztere können jedoch nur für Maschinenprogramme, zur Speicherung von Daten im Maschinencode (Pseudo-Floppy) und vom CP/M-Betriebssystem verwendet werden, da der BASIC-Interpreter für diesen Bereich nicht zur Verfügung stehen kann.



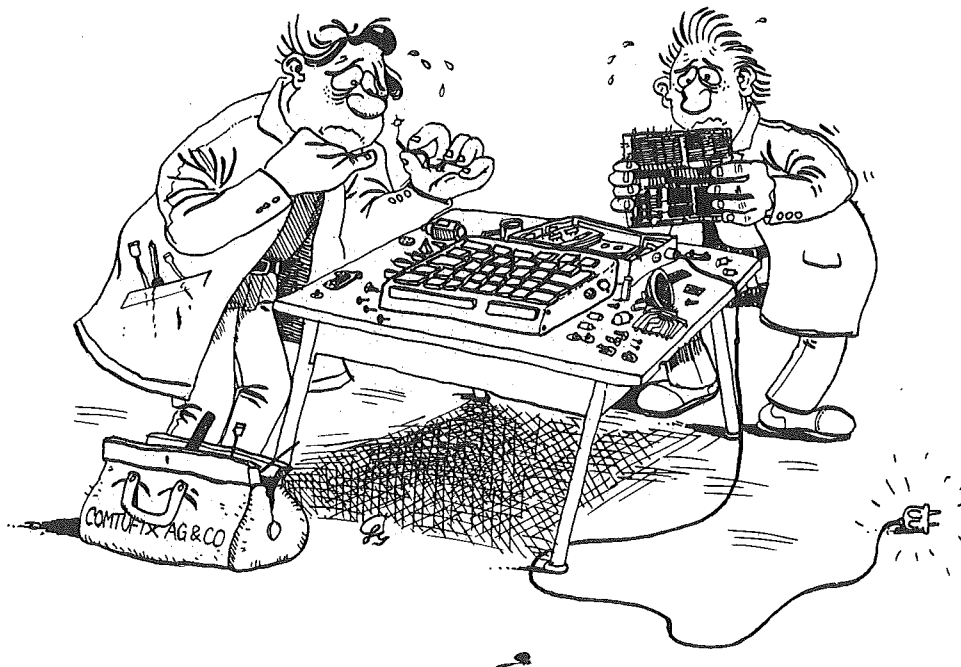
Speicheraufteilung bei SV-318 und SV-328

Die 64 KByte-RAM-Karte kann mit DIP-Schaltern auf verschiedene Bereiche eingestellt werden. Wenn man so die obere Hälfte der Bank 2 aktiviert, besteht die Möglichkeit, diesen Bereich auch in BASIC anzusprechen. Der SV-328 meldet sich bei einer so eingestellten Erweiterung mit "58398 Bytes free".

Mit dem Befehl "switch" kann man in die zweite Bank wechseln. Es können sich dann zwei BASIC-Programme gleichzeitig im Speicher befinden, zwischen denen man beliebig wechseln kann. Vorsicht, Variable können dabei nicht übergeben werden.

Über die Möglichkeiten der 64 KByte-Karte werden wir demnächst einen eigenen Artikel bringen.

In der kommenden Folge dieser Serie bringen wir die Systemarchitektur, die Beschreibung des herausgeführten Systembusses und beginnen dann mit genauen Beschreibungen der Systembausteine, beginnend mit dem Mikroprozessor Z80.



Inserate, Vorschläge und Programme bitte an die Redaktion senden! Kleinanzeigen werden kostenlos abgedruckt. Für jedes Programm, das abgedruckt wird, wird ein T-Shirt, von der schwedischen Firma "DATA TRACK" gestiftet, an den Entwickler geschickt.

Mitgliedsbeiträge für den Spectra Video Club Austria:

für Schüler, Studenten und Lehrlinge	S 250.-
ansonsten	S 500.-

Dieser Beitrag wird jährlich eingehoben. Näheres am Anmeldeformular und in den Clubstatuten.

Z 80 - PROGRAMMIEREN IN ASSEMBLER

Assemblerfortsetzungskurs Nr.:001

Wer ist noch nicht daumendrehend vor dem Computer gesessen, fluchend über die langsame Geschwindigkeit des BASIC? Der einzige Ausweg und die einzig richtige Devise lautet "Assembler". Empfohlene Lektüre : "Programmierung des Z-80" von Rodney Zaks.

Da das dezimale Zahlensystem äußerst unpraktisch ist, wird in der Assembler-Programmierung das duale und hexadezimale Zahlensystem verwendet.

Wie Sie bestimmt schon oft gehört haben, ist ein BIT die Abkürzung von "Binary Digit" und die kleinste Speichereinheit. In einem BIT kann entweder "0" oder "1" gespeichert sein. Bei den gängigen Mikroprozessoren werden diese BITS in Achtergruppen zusammengefaßt. Eine Gruppe von 8 BITS wird BYTE und eine Gruppe von 4 BITS NIBBLE genannt.

Die Darstellung von ganzen Zahlen im Dualsystem ist sehr einfach. Ein BYTE sieht nun so aus :

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

Jedes BIT stellt eine 2er Potenz da. Das 7.BIT repräsentiert 2⁷, das 6.BIT 2⁶ ... 2⁰. Die einzelnen Werte sind also 128, 64, 32, 16, 8, 4, 2, 1. Von der Dualzahl "0 1 0 0 1 0 1 1" kann man den dezimalen Wert wie folgt berechnen :

0	*	128	=	0
1	*	64	=	64
0	*	32	=	0
0	*	16	=	0
1	*	8	=	8
0	*	4	=	0
1	*	2	=	2
1	*	1	=	1

dezimal				75

Die Rechenregeln für Dualzahlen sind einfach :

0 + 0 = 0
1 + 0 = 1
0 + 1 = 1
1 + 1 = (1)0

Hierbei bedeutet (1) einen Übertrag (Carry), das binäre 10 ist dezimal 2.

Nun wollen wir zwei Dualzahlen addieren :

0 1 1 0 1 1 0 0	108
0 1 0 0 0 1 1 0	70

1 0 1 1 0 0 1 0	178

Was passiert nun, wenn wir folgende Zahlen addieren :

1 0 1 0 0 1 1 0	166
1 1 0 0 1 1 0 1	205

(1) 0 1 1 1 0 0 1 1	371

Der Übertrag (1) zeigt, daß die Summe der Zahlen 255 nicht übersteigen darf.

Bei der vorzeichenbehafteten Dualdarstellung ist das 7. BIT signifikant. Ist es "0", so kennzeichnet es eine positive und "1" eine negative Zahl.

z.B. 0 0 0 0 1 0 1 1 dezimal 11
 1 0 0 0 1 0 1 1 dezimal -11

Doch wenn man eine negative und eine positive Zahl addiert, kommt man nicht auf ein richtiges Ergebnis.

z.B. +41 + (-12) 0 0 0 1 1 0 0 1 41
 1 0 0 0 1 1 0 0 -12

 1 0 1 0 0 1 0 1 -53

Was machen wir jetzt ? Wir werden es mit dem Einerkomplement versuchen. Die positiven Zahlen bleiben gleich, jedoch bei den negativen Zahlen, wird jedes einzelne BIT invertiert.

+9 = 0 0 0 0 1 0 0 1 -9 = 1 1 1 1 0 1 1 0

Wir wollen jetzt 9 von 12 subtrahieren. 0 0 0 0 1 1 0 0 12
 1 1 1 1 0 1 1 0 -9

 (1) 0 0 0 0 0 0 1 0 2

Nachdem die Subtraktion mit dem Einerkomplement nicht funktioniert, bleibt uns nur noch das Zweierkomplement übrig. Das Zweierkomplement unterscheidet sich vom Einerkomplement nur durch die negativen Zahlen. Zu den negativen Zahlen wird das Einerkomplement erstellt und dann eins addiert und schon hat man das Zweierkomplement.

6 = 0 0 0 0 0 1 1 0 -6 = 1 1 1 1 1 0 0 1
 0 0 0 0 0 0 0 1

 1 1 1 1 1 0 1 0

Nun wollen wir prüfen, ob die Subtraktion mit dem Zweierkomplement funktioniert.

0 0 0 0 1 1 1 0 14
 1 1 1 1 1 0 1 0 -6

 (1) 0 0 0 0 1 0 0 0 = 8

Hurra, wir haben ein richtiges Ergebnis! Es funktioniert, obwohl ein Übertrag entsteht, den wir aber nicht zu beachten brauchen. Zum Schluß wollen wir noch einen Versuch zur Sicherheit machen.

0 0 0 1 0 1 0 0 20
 1 1 1 1 0 1 1 0 -10

 (1) 0 0 0 0 1 0 1 0 = 10

Ohne mathematische Beweise, können wir sagen, daß die Addition und Subtraktion mit dem Zweierkomplement funktioniert.

Achtung: Bei der vorzeichenbehafteten Dualdarstellung, beim Einerkomplement und beim Zweierkomplement sind nur Zahlen im Bereich von -127 bis +127 möglich !!!

Da ich jetzt die binäre Zahlendarstellung hoffentlich ausreichend erklärt habe, will ich jetzt zur Hexadezimalen Zahlendarstellung übergehen.

Bei der hexadezimalen Zahlendarstellung, wird ein BYTE immer in 2 NIBBLES, d.h. $2 \cdot 4$ BITS, unterteilt und jedes NIBBLE mit einem Symbol belegt. Da aber ein NIBBLE sechzehn verschiedene Zustände haben kann, kommen wir mit unseren zehn Ziffern nicht mehr aus, und müssen daher die Buchstaben A,B,C,D,E,F zu Hilfe nehmen. "A" entspricht 10 dezimal, "B" entspricht 11 dezimal, und "F" entspricht 15.

Um jetzt das NIBBLE "1 0 1 1" ins Hexadezimale umzuwandeln, bringen wir es zuerst in dezimale Darstellung: "1 0 1 1" = 11. Da 11 dezimal "B" hexadezimal entspricht, hat das NIBBLE "1 0 1 1" den hexadezimalen Wert "B". Um nun ein ganzes BYTE hexadezimal umzuwandeln, rechnet man sich das obere NIBBLE des BYTES ins Hexadezimale um und dann das untere.

Als Beispiel nehmen wir das BYTE "1 1 0 1 0 1 0 1". Das obere NIBBLE "1 1 0 1" hat den dezimalen Wert 13, welcher dem hexadezimalen Wert "D" entspricht. Mit dem unteren NIBBLE "0 1 0 1" geht es genauso: "0 1 0 1" = 5 dezimal. 5 dezimal entspricht wiederum "5" im hexadezimalen Zahlensystem.

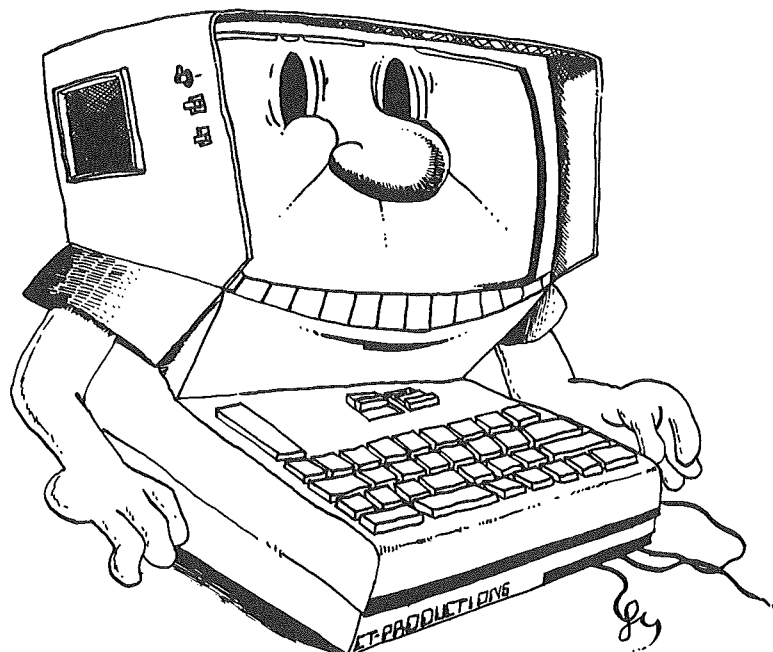
Die Dualzahl "1 1 0 1 0 1 0 1" ist daher mit der hexadezimalen Zahl "D5" identisch.

Zum besseren Verständnis zeige ich noch einige Beispiele:

Dual	Dezimal	Hex
1 0 1 1 1 0 1 0	186	BA
0 1 0 1 1 1 0 0	92	5C
1 0 0 1 0 0 1 1	147	93

Editor: Georg Wolfbauer

***** Fortsetzung folgt *****



Mein Name ist Wolfgang, und ich möchte eine Serie einleiten, in der Tips und auch Tricks für den SV beschrieben werden. Ich bin auch immer für Vorschläge und Gedankenanstöße dankbar.

Die interne Struktur des Basic-Programmspeichers

1. Teil

Zu Beginn dieser Serie möchte ich erklären, wie in einem Basic-Programm die Programmzeilen selbst im Programmspeicher abgelegt werden. In der nächsten Folge beschreibe ich, wie im Programmspeicher Variablenzuweisungen abgelegt werden. Die Fortsetzung besteht dann darin, mit Hilfe von Beispielen, die Codes der Basic-Befehle zu erklären. Das Ende dieser ersten Serie wird darin bestehen, daß ich ein Programmlisting vorstelle, womit man, direkt aus einem Programm heraus, mathematische Formeln eingeben kann.

Als Erstes also eine grobe Übersicht über die Struktur einer Basic-Zeile im Programmspeicher.

Zur Erläuterung einmal ein Speicherauszug, wobei die erste Kolonne die erste Adresse der nachfolgenden Speicherzellen ist und die weiteren acht Kolonnen die Werte aus den Speicherzellen selbst sind. Alle Werte sind im Hex-Code angegeben.

8000	00	1A	80	0A	00	91	22	44
8008	69	65	73	20	69	73	74	20
8010	65	69	6E	20	54	65	73	74
8018	22	00	00	00				

Nun die Erklärung dazu:

Die erste Speicherzelle mit der Adresse &H8000 ist bei einem Basic-Programm immer 0. Die nächsten beiden Speicherzellen mit den Adressen &H8001 und &H8002 geben zusammen die Startadresse der nächsten Basic-Zeile an, wobei jeweils die erste Speicherzelle die nieder- und die zweite die höherwertige Hälfte der Adresse enthalten. In unserem Beispiel würde die nächste Basic-Zeile bei der Adresse &H801A beginnen. Die nächsten beiden Speicherstellen enthalten die Zeilennummer der Basic-Zeile, dabei verhält es sich mit der Wertigkeit genauso, wie bei der Adresse der nächsten Zeile. Das heißt, daß die Zeilennummer den Wert 10 hat. Erst danach folgt nun die eigentliche Basic-Zeile, wobei der Wert &H91 für den Basic-Befehl PRINT steht, und danach folgt der String begrenzt durch die Anführungszeichen. In unserm Fall steht von der Adresse &H8006 bis zur Adresse &H8018 "Dies ist ein Test". Die nächste Speicherzelle mit dem Wert &H00 markiert das Ende der Basic-Zeile. Da die nächsten beiden Speicherzellen, die eigentlich die Adresse des Beginns einer Basic-Zeile enthalten müßten, den Wert &H00 enthalten weiß der Computer, daß das Ende des Basic-Programms erreicht ist. Diese Programmzeile würde also lauten:

```
10 PRINT"Dies ist ein Test"
```

Soweit also für den Anfang. In der nächsten Folge behandle ich, wie versprochen, die Variablenzuweisungen. Wer will, kann ja schon einmal mit dem, was er aus dieser Folge erfahren hat, ein bißchen probieren.

Fortsetzung folgt.

Wolfgang Rotschek

GRAFIK AUF DEM SPECTRAVIDEO SV-328

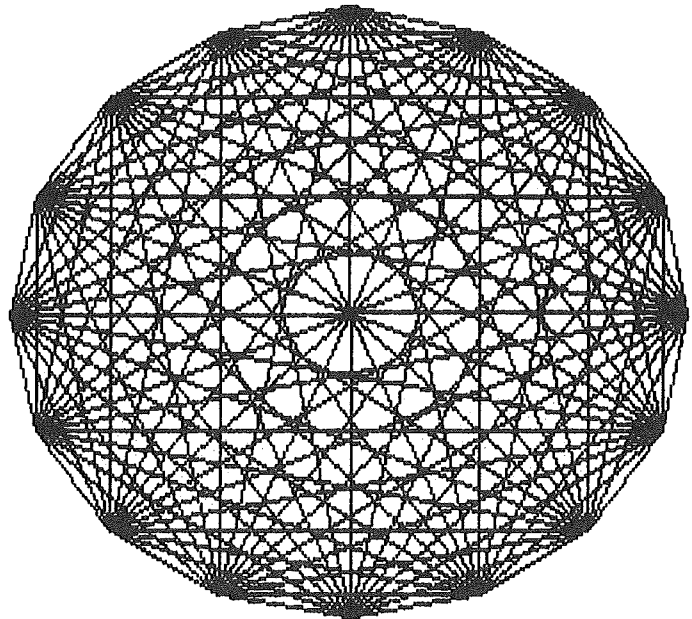
Die Spectravideo-Computer verfügen über ausgezeichnete Grafikfähigkeiten. Wir bringen Ihnen zwei Demonstrationsprogramme als Anregung zu eigenen Programmen. Mit dem Programm "Hardcopy" können Sie den Bildschirminhalt auch auf dem Drucker ausgeben. Das Programm ist für den Matrixdrucker EPSON RX-80 geschrieben und sollte auch auf EPSON-kompatiblen Druckern laufen.

Hängen Sie das Programm "Hardcopy" als Unterprogramm an das jeweilige Demonstrationsprogramm an und fügen Sie entsprechend die Zeile "GOSUB 1000" ein.

Bitte beachten Sie, daß in Zeile 1260 der Farbwert der eingestellten Zeichenfarbe des Bildschirms angegeben sein muß. Dies ist in unseren Beispielen bereits berücksichtigt.

Rechts sehen Sie einen Ausdruck des Programms GRAFIK-DEMO #1.

Wir werden uns bemühen, in den weiteren Folgen unserer Clubzeitschrift immer wieder auch Grafikprogramme abzu- drucken.



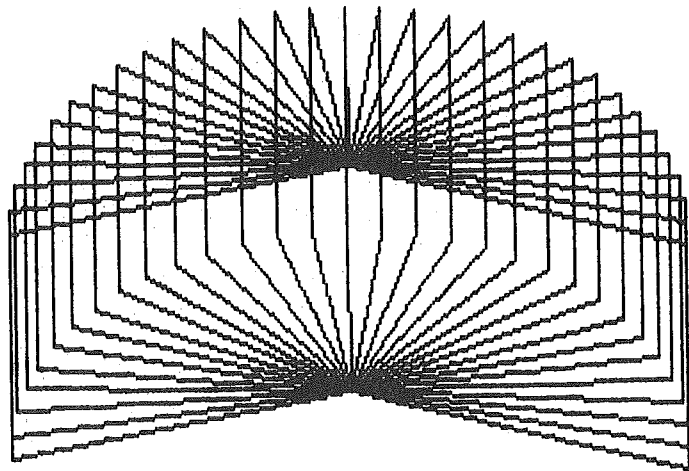
```
10 REM *****
20 REM *
30 REM *          GRAFIKDEMO #1          *
40 REM *
50 REM *    GEORG WOLFBAUER, WIEN    *
60 REM *
70 REM *****
80 SCREEN,0:CLS
90 INPUT"Eingabe der Dichte";B
100 IFB=0GOTO90
110 DIMA(B,1)
120 COLOR15,1,1
130 SCREEN1:C=0
140 FORA=0TO2*3.141592654#STEP6.283185308#/B
150 A(C,0)=SIN(A)*127+128:A(C,1)=COS(A)*95+96
160 C=C+1
170 NEXTA
180 B=B-1
190 FORA=0TOB
200 FORC=A+1TOB
210 LINE(A(A,0),A(A,1))-(A(C,0),A(C,1))
220 NEXTC,A
230 REM HARDCOPY EINFUGEN MIT GOSUB 1000
240 A#=INPUT$(1):RUN
```

```

10 REM *****
20 REM *
30 REM *          GRAFIKDEMO #2
40 REM *
50 REM *    GEORG WOLFBAUER, WIEN
60 REM *
70 REM *****
80 SCREEN,0:CLS
90 INPUT"Eingabe der Dichte";B
100 IFB=0GOTO90
110 COLOR15,1,1
120 SCREEN1
130 FORA=3.141579#/2-.1TO3.141579#/2+3.141579#+.1STEP3.141579#/B
140 X=SIN(A)*127+128:Y=COS(A)*69+70
150 LINE(128,50)-(X,Y):LINE-(X,Y+70):LINE-(128,120):NEXT
160 REM HARDCOPY EINFÜGEN MIT GOSUB1000
170 A$=INPUT$(1):REM BELIEBIGE TASTE = WEITER
180 GOTO90

```

GRAFIKDEMO #2



```

1000 REM *****
1010 REM *
1020 REM *          HARDCOPY
1030 REM *
1040 REM *    SV-328 MIT EPSON RX-80
1050 REM *
1060 REM *    HEINZ SCHMID, WIEN
1070 REM *
1080 REM *****
1090 DEFUSR=&HA000
1100 RESTORE 1130
1110 READA$
1120 IFA$="*"THEN1140ELSEPOKE&HA000+I%,VAL("&H"+A$):I%=I%+1:GOTO1110
1130 DATA 23,23,7E,06,08,17,CB,1A,10,FB,7A,CD,15,39,C9,*
1140 LPRINTCHR$(27);"3";CHR$(24);
1150 FORX%=31TO0STEP-1
1160 LPRINTCHR$(27);"L";CHR$(128);CHR$(1);
1170 FORY%=0TO23:FORZ%=0TO7
1180 R%=VPEEK(256*Y%+8*X%+Z%)
1190 IFR%=0THENGOSUB1250
1200 AA=USR0(R%):AA=USR(R%)
1210 NEXTZ%,Y%
1220 LPRINTCHR$(13)
1230 NEXTX%
1240 RETURN
1250 LP%=VPEEK(8192+256*Y%+8*X%+Z%)
1260 IFLP%=15THENR%=255:REM 15 = ZEICHENFARBE WEISS
1270 RETURN

```

HISTOGRAMM mit DURCHSCHNITT

Dieses Programm verarbeitet die Noten einer Schularbeit. Sie können die Anzahl der Noten eingeben, vorausgesetzt, die Anzahl der Schüler jeder Note ist unter 17. Danach zeichnet der Computer ein Histogramm mit Perspektive. Der Notendurchschnitt wird ebenfalls ausgerechnet und graphisch und numerisch angezeigt.

Die Eingabe wird mit einer Schleife erledigt, die auch gleichzeitig kontrolliert, ob die Eingabe unter 17 liegt. C, die Variable für die Durchschnittsberechnung wird ebenfalls in der Schleife berechnet. Danach wird in der Variable B die Gesamtanzahl der Schüler bestimmt. Die nächste Schleife zeichnet die 5 Türmchen des Histogramms. V ist die Hilfsvariable für die Höhen der Blöcke, W dient dem richtigen Abstand zwischen den Blöcken. Zum Schluß wird der Durchschnitt berechnet und mit fast der gleichen Routine, die für die ersten fünf Blöcke gebraucht wurde, angezeigt. In der Zeile 350 steht der Wartebefehl, der das Bild solange angezeigt hält, solange keine Taste gedrückt wird.

```

10 REM *****
20 REM *
30 REM * HISTOGRAMM MIT DURCHSCHNITT *
40 REM *      5.JUNI 1984. *
50 REM *      MADE BY GERHARD FALLY *
60 REM *      1020WIEN *
70 REM *
80 REM *****
90 SCREEN0,0:FORI=1TO5
100 PRINT"WIE OFT NOTE" I;
110 INPUT A(I):IF A(I)>16THENPRINT"BITTE GEBEN SIE VERNUENFTIGE
    ZAHL EIN!":GOTO100ELSEC=C+A(I)*I
120 NEXT
130 B=A(1)+A(2)+A(3)+A(4)+A(5)
140 COLOR 15,4:SCREEN1:C=C/B
150 REM ***HISTOGRAMMAUFBAU***
160 FORI=1TO5:V=150-A(I)*8:W=35+I*16
170 LINE(5+W,150)-(12+W,V),10,BF
180 LINE(W,143)-(W,V-7),6
190 LINE(4+W,150)-(W,143),6
200 LINE(8+W,V-8)-(W,V-8),6
210 LINE(4+W,V)-(4+W,150),6
220 LINE(12+W,V-1)-(8+W,V-8),6
230 LINE(12+W,V-1)-(4+W,V-1),6:PAINT(3+W,145),6
240 LOCATEW,160:PRINTI
250 LOCATEW,8:PRINTUSING"##":A(I):NEXT
260 REM ***DURCHSCHNITTSANZEIGE***
270 LINE(54,179)-(39+C*16,187),10,BF
280 LINE(53,179)-(53,180),6
290 LINE(53,187)-(50,180),6
300 LINE(50,172)-(50,180),6
310 LINE(34+C*16,172)-(50,172),6
320 LINE(39+C*16,178)-(34+C*16,172),6
330 LINE(53,178)-(39+C*16,178),6:PAINT(51,174),6
340 LOCATE137,180:PRINT"D="C
350 S$=INKEY$:IFS$=""THEN350ELSERUN

```



```

*****
*
*           BUBBLE-SORT AUF DEM SPECTRA VIDEO COMPUTER
*
*
*****

```

Bubble-Sort heißt das einfachste Verfahren zum Sortieren einer Liste in steigender oder fallender Reihenfolge.

Die beiden untersten bzw. obersten Elemente einer Liste werden verglichen. Ist das untere kleiner (größer) als das obere so werden die beiden Elemente vertauscht. Das "leichtere" Element steigt also wie eine Luftblase nach oben, daher kommt auch der Name "Bubble-Sort", auf deutsch "Blasensortieren".

Dank des Befehls "SWAP", den uns das Erweiterte Microsoft-BASIC der Spectra Video Computer zur Verfügung stellt, läßt sich ein derartiges Programm auf einfache Art realisieren. Das folgende Programm soll Ihnen eine Anregung für eigene Programme geben.

```

10 REM *** EINGABE DER LISTE
20 PRINT"Geben Sie nun eine Liste von Begriffen"
30 PRINT"ein, die Sie sortieren moechten:"
40 PRINT"(* = Ende der Liste, max. 30 Elemente)"
50 DIM A$(30)
60 FOR I = 1 TO 30
70 INPUT A$(I)
80 IF A$(I) = "*" THEN L = I-1 : GOTO 100
90 NEXT I
95 L = 30
100 REM *** BUBBLE-SORT
110 FOR I = 1 TO L-1
120 FOR J = I+1 TO L
130 IF A$(I)>A$(J) THEN SWAP A$(I),A$(J)
140 NEXT J
150 NEXT I
200 REM *** AUSGABE DER SORTIERTEN LISTE
205 PRINT:PRINT"SORTIERTE LISTE"
210 FOR I = 1 TO L : PRINT A$(I) : NEXT

```

Probelauf:

```

Geben Sie nun eine Liste von Begriffen
ein, die Sie sortieren moechten:
(* = Ende der Liste, max. 30 Elemente)
? Birne
? Marille
? Zwetschke
? Pfirsich
? Apfel
? *

```

```

SORTIERTE LISTE
Apfel
Birne
Marille
Pfirsich
Zwetschke

```

```

100 REM *****
110 REM *
120 REM *          SWITCH BASIC-ROM IN RAM BANK 2
130 REM *
140 REM *          REINHOLD WENZL,  HÖRSCHING
150 REM *
160 REM *****
170 FORX=&HA000T0&HA02A
180 READA#
190 POKE X, VAL("&H"+A#)
200 NEXT
210 DEFUSR=&HA000
220 X=USR(0)
230 CLS
240 NEW
250 DATA F3,E5,D5,C5,01,00,80,21,00,00,3E,0F,D3,8B,3E,DF,D3,8C
260 DATA 56,3E,0F,D3,8B,3E,DD,D3,8C,72,23,0B,05,04,20,E8,0D,0C
270 DATA 20,E4,C1,D1,E1,FB,C9

```

Dieses Programm haben wir von Herrn Wenzl, einem Mitglied des "Computer Club SV-328" in Linz-Hörsching erhalten.

Beim Ablauf des Programms wird das BASIC-ROM (32 KByte) in die zweite Bank (untere 32 KByte) übertragen. Das Programm löscht sich danach selbst.

Sie haben nun die Möglichkeit, Änderungen im BASIC-Interpreter vorzunehmen. Eine kleine Demonstration stellt das folgende kurze Programm dar:

```

10 FORI=&H5EET0&H5F3:READA#:POKEI,VAL("&H"+A#):NEXT
20 DATA66,65,68,6C,65,72

```

Lassen Sie dieses Programm laufen und geben Sie danach irgendwelche Zeichen ein und drücken Sie die ENTER-Taste: Ihr SV-328 meldet sich jetzt mit "Syntaxfehler".

Das Programm funktioniert selbstverständlich nur auf dem SV-328, nicht jedoch auf dem SV-318.

Melden Sie sich bitte bei der Blutspendezentrale oder einer anderen Dienststelle des

ÖSTERREICHISCHEN
ROTEN KREUZES

Dem Blutspender werden etwa 350 Gramm Blut aus der Armvene entnommen. Dies entspricht etwa 1/14 der Gesamtblutmenge eines gesunden, erwachsenen Menschen und beeinträchtigt das Wohlbefinden des Spenders nicht.

rette
**ein
Leben,
spende
Blut**

Intelligentes Wortfinden

Was ist das? Wenn Sie schon einmal probiert haben, ein Adventure-Spiel zu schreiben, muß der Computer die von Ihnen eingegebene Zeile erst überprüfen, um festzustellen, welche Anweisungen er befolgen kann und welche nicht. Deshalb muß er zuerst die Wörter in einem String (in unserem Falle A\$) finden. Diese Routine liefert sowohl den Anfangspunkt als auch den Endpunkt eines Wortes in A\$. Eine vordefinierte Funktion (FN WRT\$) liefert zusätzlich das Wort Nr. X in A\$.

Ein kleines Beispiel: Wenn die Meldung "SATZ:" erscheint, können Sie einen Satz mit bis zu 25 Wörtern eingeben wie z.B.: "Guter Rat ist teuer". Danach wird der Computer einen Ausdruck von allen in A\$ vorkommenden Wörtern liefern mit den dazugehörigen Wortnummern.

Wenn Sie dann aus dem Programm mit "ENTER" aussteigen, können Sie mit der Funktion WRT\$(X), wobei X die Wortnummer ist, jedes Wort aus dem String A\$ bekommen. Im obigen Falle würde "? FN WRT\$(1) " das Wort "Guter" liefern, "? FN WRT\$(4) " das Wort "teuer" liefern.

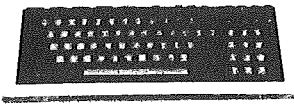
```
9000 ' Woerter finden
9010 '
9020 CLEAR 768
9030 ' fn wrt$(x): liefert Wort Nr. x in a$
9040 DEFFN WRT$(X)=MID$(A$,X(X,0)+1,X(X,1)-X(X,0))
9050 ' fn md$(x) :=mid$(a$,x,1)
9060 DEFFN MD$(X)=MID$(A$,X,1)
9070 ' fn frg(x) : stellt fest,ob Satzzeichen vorhanden ist
9080 DEFFN FRG(X)=- (FN MD$(X)<>" " AND FN MD$(X)<>"," AND FN MD$(X)<>". " AND FN MD$(X)<>"!" AND FN MD$(X)<>"'" AND FN MD$(X)<>": " AND FN MD$(X)<>"; " AND FN MD$(X)<>"?")
9090 MW=25 ' maximal 25 Woerter
9100 WR=1 ' Wortnummer
9110 WA=0 ' Wortanfang
9120 DIM X(MW,1)
9130 CLS
9140 A$="";B$=""
9150 PRINT"SATZ:";
9160 LINEINPUTB$
9170 IFB$=""THENENDELSEA$=B$
9180 FOR X=1 TO LEN(A$)
9190 IF WA<>FN FRG(X) THEN X(WR,WA)=X-1:WR=WR+WA:WA=1-WA:IF WR>MW THEN 9210
9200 NEXT
9210 IF X(WR,0)<>0 THEN X(WR,1)=X-1
9220 FOR I=1 TO WR
9230 PRINT USING"WORT NR.:###";I;
9240 PRINT TAB(20);FN WRT$(I)
9250 NEXT
9260 GOTO 9150
```

Haben Sie's gewußt?

Zum Spectravideo-Computer gibt es den Coleco-Adapter. Damit ist es möglich, auf dem Computer die zahlreichen Spiele von CBS Coleco Vision zu spielen, z.B. Zaxxon, Smurf, Ladybug, Gorf, Donkey Kong. Der Adapter kostet S 1.790,- im Computer-Studio, Paniglg.18-20.

SPECTRAVIDEO**SV-328**

SV-328 nur 7990.-
 SV-318 nur 5490.-
 COBOL, FORTRAN eingetroffen.



CPU Z80, 3,56 Mhz,
 erw. Microsoft-BASIC,
 CP/M 2.2

Der Ausbaufähige

Wir sind Spezialisten für den SV-328. Daher bieten wir immer aktuellste Preise und ausführlichste Information über diesen leistungsfähigen Personalcomputer.

Zahlreiche Programme für geschäftliche Anwendung (z. B. Textverarbeitung, Adreßverwaltung, Datenbank, Fakturierungsprogramm, Etikettendruck) mit deutscher Dokumentation. Mit deutschem Zeichensatz lieferbar. BASIC-Kurse.

Matrixdrucker, Typenraddrucker, Monitore.

Täglicher Postversand.

Vorführungen und Verkauf in unserem neueröffneten

Computer-Studio

1040 Wien, Paniglgasse 18-20 (neben TU)
 Tel. 65 78 08, 65 88 93, Mo-Fr 9-18, Sa 9-12

Elektronik
 WEHSNER GESELLSCHAFT M.B.H.

SUCHRÄTSEL

In diesem Wirrwarr von Buchstaben sind 25 Begriffe aus der Computertechnik versteckt. Versuchen Sie, sie zu finden.

S P E C T R A V I D E O
 R I A H D E I Q W B T R
 O Z N I U D S W E A F T
 T A S P E P I A S S O U
 I K Y O R R E T N I R P
 N E R O M O A X S C T O
 O A M K V T Z L B P R K
 M A K K U R T E G M A E
 I M Y R P L H N S O N E
 A N L N C G A P I S L P
 R E G I S T E R T R O P
 C R E L B M E S S A P R

Medieninhaber (Verleger): Spectra Video Club Austria, p.A. Computer-Studio, A-1040 Wien, Paniglgasse 18-20, Telephone 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. Chefredakteur: Gerhard Fally

SPECTRAVIDEO

SV-318/328

**Die Computer für alles.
 Freizeit und Beruf.**

SPECTRAVIDEO