

SV

JOURNAL

Die Zeitschrift des Spectra Video Club Austria



INHALT

2	CLUBNACHRICHTEN
3	SPECTRAVIDEO-BASIC
6	SV-Hardware
8	I/O-Port-Adressen
11	Z80 - Programmieren in Assembler
13	TIPS & TRICKS
15	Programme

Heft 2/84

S 15.-

Liebes Clubmitglied!

Obwohl viele Leute bis jetzt in unseren Club eingetreten sind, hört man doch manchmal kritische Stimmen: Wieso braucht man einen Computerclub? Warum ist das SVi-Journal so "einfach" geschrieben?

Ich persönlich glaube, daß ein Computerclub das Arbeiten mit einem Computer verbessert. Es gibt Spezialisten auf dem Gebiet der Hardware, auf dem Gebiet von Maschinencode und Basic, usw.. Wenn diese Leute zusammenkommen, können Sie Anfängern helfen und auch untereinander Erfahrungen austauschen. Gerade beim Spectravideo ist der Erfahrungsaustausch wichtig, denn über diesen Computer gibt es wenig Literatur, und viel, was wir über den SV-318, SV-328 und deren Peripherie wissen, haben wir uns selber erarbeitet.

In diesem Sinn ist auch die Zeitung zu verstehen! Das für Spezialisten in diesem Club das SVi-Journal keine Neuigkeiten birgt, ist wohl klar, denn gerade diese Leute machen ja die Zeitschrift! Trotzdem haben wir am Anfang versprochen, uns Beschwerden anzuhören und die Fehlerquellen, so gut es geht, zu beseitigen. Wir werden deshalb in den nächsten Folgen auch Maschincodeprogramme abdrucken und auch für Fortgeschrittene Programmpunkte bieten.

Sollten Sie übrigens noch nicht davon überzeugt sein, daß das SVi-Journal in erster Linie Kommunikationsmöglichkeit für uns Clubmitglieder ist, dann bitte ich Sie, falls Sie etwas mitteilen wollen, tun Sie dies bei uns. Und sollten Sie glauben, daß Beschwerden bei uns in den (geistigen) Mistkübel wandern, dann beschweren Sie sich. Sie werden sehen, wie ernst wir Ihre (hoffentlich) ernstgemeinte Beschwerde nehmen.

Ihr SVi-Journal-Chefredakteur Gerhard Fally!

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

Die nächsten Clubabende :

Sa, dem 18. August 1984, ab 17 Uhr
Mi, dem 29. August 1984, ab 19 Uhr
Sa, dem 16. September 1984, ab 17 Uhr
Mi, dem 26. September 1984, ab 19 Uhr

im Computerstudio, 1040 Wien, Paniglgasse 18-20.
Gäste sind willkommen. Ende jeweils ca. 22 Uhr!

Aktivitäten an den Clubabenden: Arbeiten an Spectravideo-Systemen, Informationsaustausch zwischen Clubmitgliedern
Ab Oktober zusätzliche Vorträge über Spectravideo-Computer, deren Anwendung und deren Peripherie!

Neu im "COMPUTER-STUDIO" eingetroffen!

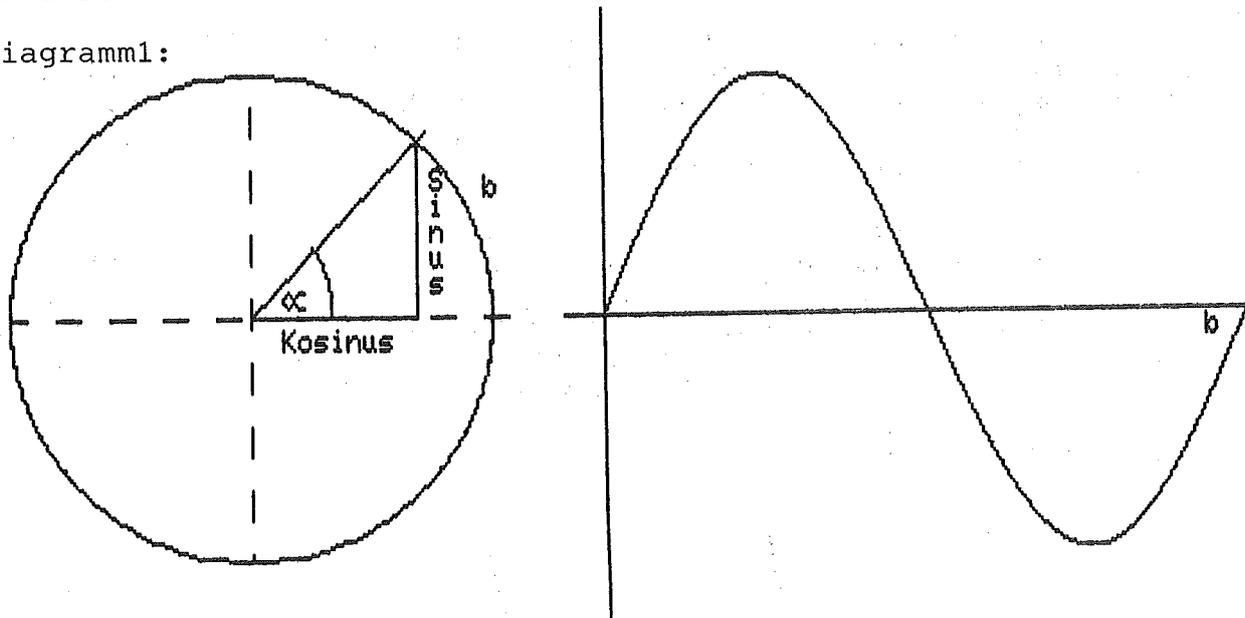
Der Super-Expander SV-605B hat zwei eingebaute Floppy-Disk-Laufwerke mit je 320 KByte Speichervermögen (formatiert!) ! Außerdem liefert die Firma Wehsner ein Software-Paket mit 5 Programmen mit (WordStar, Mailmerge, CalcStar, ReportStar, DataStar von dem Software-Haus MicroPro).

SPECTRAVIDEO-BASIC

Leider hat das SV-Handbuch auch bei Funktionen Fehler und Ungenauigkeiten. Deshalb kommen in der zweiten Folge des SV-BASIC die Funktionen an die Reihe. Der Autor wird die Reihenfolge einhalten, die im Bedienermanual verwendet ist, wird aber einige Funktionen weglassen, bei denen man nichts hinzuzufügen braucht.

ATN Der Arkustangens wird auf dem SV-328 in doppelter Genauigkeit berechnet. Was ist der Arkustangens genau? Er ist eine trigonometrische Funktion. Zur Auffrischung des Gedächtnisses möchte ich kurz die Grundlagen von Sinus, Kosinus, Tangens, usw. erklären.

Diagramm1:



Einheitskreis, Sinus, Kosinus, Sinuswelle

Sie sehen oben einen Kreis, er soll den Radius 1 haben. Man nennt so einen Kreis auch Einheitskreis. "b" ist die Abkürzung für Bogenmaß und bezeichnet einen Teil des Umfangs dieses Kreises. Vielleicht wird Ihnen aufgefallen sein, daß π öfters bei Berechnungen mit Bogenmaßen vorkommt. Wie hängt π nun mit "b" zusammen? Dazu eine kleine Rechnung: Wie groß ist der Umfang eines Kreises mit dem Radius 1 (bzw. wie groß ist das Bogenmaß von 360° , was aber das gleiche ist) ?

Die Formel für den Umfang: $U=2r\pi$

Wenn nun $r=1$ gilt, dann ergibt sich $U=2*1*\pi$. Das Bogenmaß für den ganzen Kreis ist also 2π . Hier ein kleiner Tip: Der SV-328 hat kein π . Man kann aber ganz am Anfang eines Programms mit der Variable "PI=3.14159265358979323846" festlegen. Normalerweise genügen für den Computer 14 Dezimalstellen, weil er die restlichen sowieso rundet, der Autor hat aber 20 Dezimalstellen aufgeschrieben, falls jemand Spezialberechnungen durchführen muß.

Wenden wir uns nun dem Dreieck im Kreis zu. Hier sehen Sie den Zusammenhang zwischen Sinus und Kosinus. Wenn Sie in Gedanken den Winkel α verändern, verändern sich auch die Sinus- und Kosinuswerte. Dabei müssen Sie aber beachten, daß das Dreieck immer rechtwinkelig bleiben muß.

Mit diesen beiden Funktionen kann man, wenn man alle Winkelwerte von 0-360 durchläuft, einen Kreis zeichnen. Oder aber, wenn

man keinen ganzen Kreis zeichnen will, dann kann man den Anfangs- und den Endpunkt eines Kreises mit Sinus und Kosinus eruieren. Der Computer rechnet aber den Sinus nicht von Grad, sondern von Bogenmaß aus. Der Zusammenhang zwischen Grad und Bogenmaß:

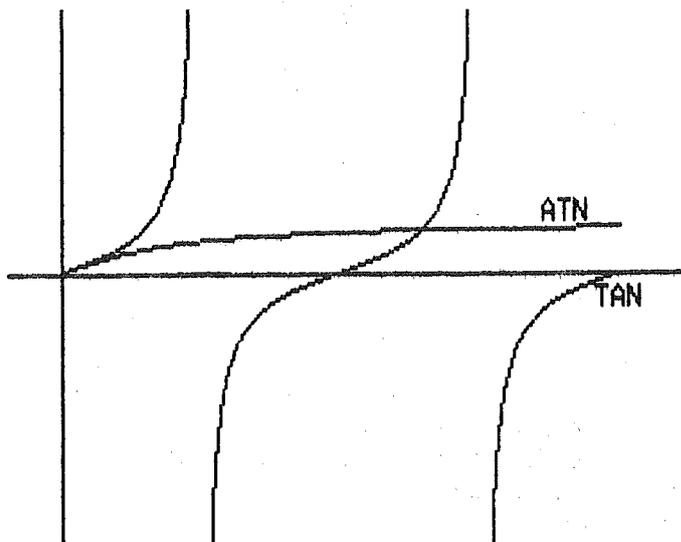
$$b = (r * \pi * \alpha) / 180 \quad \begin{array}{l} r = \text{Radius des Kreises} \\ \alpha = \text{Winkelwert in Grad} \end{array}$$

Selbstverständlich kann man den Kreis auch mit einem Radius ρ erzeugen. Das Bogenmaß bekommt dabei immer höhere Werte, während die Sinus-, Kosinus-Werte für den jeweiligen Winkel immer gleich bleiben. So entsteht auch die Sinusschwingung, wenn Sie Bogenmaß durch Zeit ersetzen. (siehe Diagramm 2)

Sinus gebrochen durch Kosinus ergibt den Tangens (siehe Diagramm2). Kosinus gebrochen durch Sinus ergibt den Kotangens. Der Kotangens ist um 90° gedreht, gegenüber dem Tangens. (Ko- ist die Vorsilbe für 90° gedreht. Sinus und Kosinus verhalten sich auch rechtwinkelig zueinander)

Nun kann man auch die Umkehrfunktionen der einzelnen Funktionen bestimmen, d.h. man vertauscht die x-Achsen- mit den y-Achsenwerten der Funktion. So entstehen die inversen Funktionen, eine davon ist der Arkustangens (=inverser Tangens= Umkehrfunktion des Tangens). Die folgende Skizze soll das Wesen des Arkustangens im Verhältnis zum Tangens verdeutlichen.

Diagramm2:



Tangens zu Arkustangens

CINT Der Basic-Interpreter hat bei der "CINT"-Funktion einen Fehler: Die Zahlen werden nicht gerundet, sondern es werden die Dezimalstellen abgeschnitten. Abgesehen davon wurde bei den Beispielen im Buch vergessen, neben das "46" das "-46" zu setzen. Dementsprechend müßten die Beispiele etwa so lauten.

```
PRINT CINT (45.67), CINT(-45.67)
45                -45
```

COS Der Kosinus wird in doppelter Genauigkeit berechnet. Die mathematischen Grundlagen stehen beim Arkustangens ("ATN").

CSNG Der Wert wird nicht gerundet, die überflüssigen Stellen werden abgeschnitten.

EXP Die Exponentialfunktion wird ebenfalls in doppelter Genauigkeit berechnet. In Zusammenhang mit "LOG(X)" möchte ich auch versu-

chen, die wichtigsten Beziehungen vom natürlichen Logarithmus zu "EXP" und "LOG" zu durchleuchten.

Bei einer Potenzzahl gilt $a^l=p$. a ist die Basis, l der Exponent. Man sagt für l : l ist der Logarithmus von p zur Basis a . Das heißt: z.B. $10^2=100$. Demnach ist 2 der Logarithmus von 100 zur Basis 10. Nun bilden alle Logarithmen mit der gleichen Basis ein Logarithmensystem. Eines von diesen Systemen hat die Basis 2.71828.... Es wird das natürliche Logarithmensystem genannt. Mathematisch Versierte haben in 2.71828... schon längst die Euler'sche Zahl erkannt, auch kurz "e". "EXP(X)" rechnet nur für a den Wert $e(2.71828...)$, für l wird X genommen. Die Formel von oben wird zu $2.71828...^X=p$ umgeformt.

"LOG(X)" rechnet den natürlichen Logarithmus von einem Wert X aus. In diesem Fall sieht die obrige Formel für den Computer so aus: $2.71828...^y=x$, wobei y in diesem Fall das Ergebnis von "LOG(X)" ist.

Der natürliche Logarithmus wird für wissenschaftliche Berechnungen gebraucht. Nach dem natürlichen Logarithmus (in der Mathematik \ln genannt) werden der dekadische und der duale Logarithmus oft verwendet. Die Umrechnung natürlicher Logarithmus (\ln)- dekadischer Logarithmus (\lg) funktioniert auf folgender Formel: $\lg p = \ln p / \ln 10$

In Worten: Der dekadische Logarithmus von p ergibt sich aus dem natürlichen Logarithmus von p , gebrochen durch den natürlichen Logarithmus von 10

Im Basic: $LG=LOG(X)/LOG(10)$ ODER $LG=LOG(X)*.434294481$

Das .434294481 kommt zustande, indem der natürliche Logarithmus von 10 ausgerechnet wird und dann $1/LOG(10)$ dividiert wird. Diese Zahl wird Modul des Logarithmus zur Basis 10 genannt.

Die gleiche Prozedur kann man übrigens auch machen, wenn man den dualen Logarithmus (ld) aus dem natürlichen berechnen will. Nur wird hier der 10er durch einen 2er ersetzt.

LOG Der natürliche Logarithmus wird mit doppelter Genauigkeit berechnet. Die mathematischen Grundlagen siehe "EXP"

RND Es muß auf jeden Fall eine Zahl mit Klammern hinter "RND" stehen, sonst gibt es einen Syntax-Error. Wenn die Zahl 0 ist, wird die zuletzt erzeugte Zufallszahl wiederholt, andernfalls eine Neue erzeugt. Die Zufallszahlen werden in doppelter Genauigkeit geliefert.

SIN liefert den doppelt genauen Sinus. Mathematische Grundlagen siehe "ATN"

SQR liefert die doppelt genaue Quadratwurzel

TAN liefert den doppelt genauen Tangens. Mathematische Grundlagen siehe "ATN"

TIME Genau alle .02 Sekunden wird ein Impuls gesendet. Das sind 50 Impulse pro Sekunde

Gerhard Fally

**** Fortsetzung folgt ****

S V - H a r d w a r e

Die Hardware der Spectra Video Computer in Fortsetzungen

2. F o l g e

Der Mikroprozessor, der unseren Computer steuert, ist der Z80A. Dieser Prozessor ist eine Weiterentwicklung des 8080. Beide werden von der Firma Zilog hergestellt.

Unsere CPU (Central Processing Unit) benötigt eine zeitliche Referenz, den Takt. Die Taktfrequenz beträgt bei den SVI-Computern 3.58 MHz. Ein Quarz mit der Frequenz von 10.738 MHz wird vom Video-Chip TMS-9929 (TMS-9918A) verwendet. Dieser generiert alle internen Taktsignale. Der CPU-Takt wird mittels Division durch 3 erhalten und vom Video-Controller zur Verfügung gestellt.

Unser System hat drei Busse: Der 8 Bit breite bidirektionale Datenbus überträgt Daten zwischen CPU und Speicher bzw. zwischen CPU und Ein-/Ausgabebausteinen. Der 16 Bit breite unidirektionale Adreßbus überträgt Adressen, die von der CPU erzeugt werden. Die Adressen legen Quelle oder Ziel der Daten fest, die über den Datenbus übermittelt werden. Der Kontrollbus schließlich überträgt jene Signale, die zur Synchronisation des Systems benötigt werden.

Das ROM (read only memory = Lesespeicher) hat einen Umfang von 32 KByte und enthält das Betriebssystem unseres Computers sowie den BASIC-Interpreter. Dieser Teil des Speichers besteht aus zwei maskenprogrammierten 16 K x 8 ROM-Chips. In den ersten Versionen der Spectra Video Computer waren es noch 4 EPROMs (8 K x 8), die sich auf einer kleinen aufgesteckten Platine befanden.

Unser Computer benötigt für die Programme, die wir erstellen möchten, und für seine Systemvariablen auch einen Schreib-/Lesespeicher, das RAM (random access memory). Es werden dynamische RAMs verwendet, und zwar beim SV-318 8 Stück des Typs 4116 (d.s. 16384 x 1 Bit-RAMs) also zusammen 16 KByte RAM, beim SV-328 sind es 8 Stück 4864, die zusammen 64 KByte RAM ergeben.

Jedes Computersystem benötigt aber auch entsprechende Ein-/Ausgabebausteine, um mit der Umwelt in Kontakt treten zu können. Den Video-Prozessor haben wir bereits erwähnt. Es handelt sich um den TMS-9929 (TMS-9918A) von Texas Instruments. Er ermöglicht die Ausgabe auf dem Fernsehapparat oder Monitor. Ein Video-RAM von 16 KByte steht zur Verfügung. Der Bildschirminhalt wird daraus ausgelesen. Acht 4116-Bausteine bilden diesen Speicher.

Für die Tonausgabe ist der "Programmierbare Tongenerator"-Baustein AY-3-8910 eingesetzt. Er hat 16 Schreib-/Leseregister. Drei Tonkanäle und ein Geräuschgenerator stehen zur Verfügung. Mixer können die Signale kombinieren, die Hüllkurve kann bestimmt werden.

Zusätzlich gibt es bei diesem Baustein zwei 8 Bit breite Ein-/Ausgabeports. Port A wird zum Einlesen der Joystick-Stellungen verwendet, Port B zum Umschalten zwischen den Speicherbanken.

Ein weiterer programmierbarer Ein-/Ausgabebaustein ist der 8255A. Er verfügt über drei 8 Bit breite Ports: Port A, B und C. Sie

dienen dem Einlesen des Feuerknopfs am Joystick, dem Einlesen von der Tastatur und dem Betrieb des Kassettenrecorders.

Die Ein-/Ausgabebausteine, die bis jetzt aufgezählt wurden, befinden sich alle im Computer selbst. Weitere Ein-/Ausgabebausteine kommen in den Erweiterungskarten zum Einsatz.

In der 80 Zeichen-Karte befindet sich der Video-Controller 6845. Er hat die Aufgabe, die 80 Zeichen-Ausgabe auf einen Monitor zu kontrollieren. Er hat 17 Register. Die 80 Zeichen-Karte hat einen eigenen Zeichengenerator in einem 4 KByte EPROM 2732.

Für die Ansteuerung der Floppy-Laufwerke sorgt der FD-1793 auf der Floppy-Controller-Karte.

Bei den seriellen Ausgabeschnittstellen RS232 und beim Modem ist jeweils ein 8250 eingesetzt.

Die Aufstellung auf der folgenden Seite gibt die I/O-Adressen für die peripheren Bausteine an, das sind jene Adressen, die man mit den BASIC-Befehlen OUT und INP anspricht, bzw. mit den Befehlen OUT und IN im Z80-Assembler.

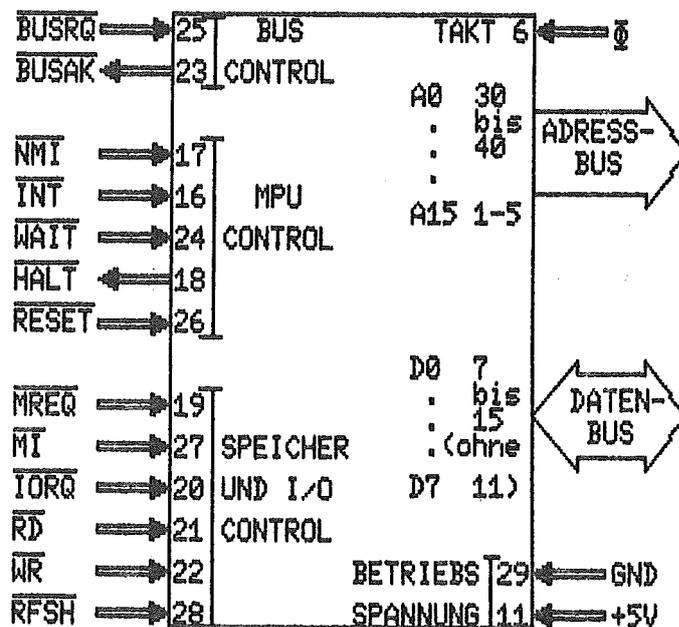
Wir wollen nun die Bausteine unseres Computers eingehender behandeln und beginnen mit dem Mikroprozessor Z80.

Der Mikroprozessor Z80

Die folgende Darstellung zeigt die Anschlußbelegung des Z80. Auf der rechten Seite findet man den Adreß- und Datenbus und die Betriebsspannung.

Die Steuersignale des Kontrollbusses kann man in vier Gruppen einteilen: das Taktsignal, die beiden Steuersignale für den Bus, sechs Steuersignale für den internen Zustand der CPU und sechs Signale zur Steuerung von Speicher und Ein-/Ausgabebausteinen.

Das Taktsignal ist in der Grafik rechts oben eingezeichnet, die übrigen Leitungen des Kontrollbusses sieht man auf der linken Seite der Darstellung.



Die Anschlußbelegung des Z80

Die I/O-Port-Adressen des SV-318 und SV-328

I/O-Port	R/W	Beschreibung	Peripherie
10H	W	WRITE DATA PORT	DRUCKER
11H	W	DATA STROBE	DRUCKER
12H	R	STATUS (READY: BIT 0 = 0)	DRUCKER
20H	R	RECEIVER BUFFER REGISTER	MODEM
	W	DIVISOR LATCH (LEAST SIGNIFICANT)	MODEM
	W	TRANSMITTER HOLDING BUFFER REGISTER	MODEM
21H	W	DIVISOR LATCH (MOST SIGNIFICANT)	MODEM
	W	INTERRUPT ENABLE REGISTER	MODEM
22H	W	INTERRUPT IDENTIFICATION REGISTER	MODEM
23H	W	LINE CONTROLL REGISTER	MODEM
24H	W	MODEM CONTROL REGISTER	MODEM
25H	R	LINE STATUS REGISTER	MODEM
26H	R	MODEM STATUS REGISTER	MODEM
28H	R	RECEIVER BUFFER REG. (LEAST SIGN.)	RS-232
	W	DIVISOR BUFFER REGISTER	RS-232
	W	TRANSMITTER HOLDING BUFFER REGISTER	RS-232
29H	W	DIVISOR LATCH (MOST SIGNIFICANT)	RS-232
	W	INTERRUPT ENABLE REGISTER	RS-232
2AH	W	INTERRUPT IDENTIFICATION REGISTER	RS-232
2BH	W	LINE CONTROL REGISTER	RS-232
2CH	W	MODEM CONTROL REGISTER	RS-232
2DH	R	LINE STATUS REGISTER	RS-232
2EH	R	MODEM STATUS REGISTER	RS-232
30H	R	FD-1793 STATUS REGISTER	FLOPPY DISK
	W	FD-1793 COMMAND REGISTER	FLOPPY DISK
31H	R/W	FD-1793 TRACK REGISTER	FLOPPY DISK
32H	R/W	FD-1793 SECTOR REGISTER	FLOPPY DISK
33H	R/W	FD-1793 DATA REGISTER	FLOPPY DISK
34H	R	READ INTRQ AND DRQ O/P PINS	FLOPPY DISK
	W	DISK SELECT REGISTER	FLOPPY DISK
		SELECT DISK 1: BIT 0 = 0	
		SELECT DISK 2: BIT 1 = 0	
		MOTOR DISK 1 ON: BIT 2 = 0	
		MOTOR DISK 2 ON: BIT 3 = 0	
38H	W	DENSITY SELECT REGISTER	FLOPPY DISK
		DOUBLE DENSITY: BIT 0 = 0	
		SINGLE DENSITY: BIT 0 = 1	
50H	W	ADRESS REGISTER SELECT	80 ZEICHEN
51H	W	CRT CONTROLLER REGISTER (R0 - R17)	80 ZEICHEN
58H	W	CRT BANK CONTROL	80 ZEICHEN
		BANK ON/OFF: FFH/00H	
80H	W	TMS-9918A WRITE MODE = 0	VDP
81H	W	TMS-9918A WRITE MODE = 1	VDP
84H	R	TMS-9918A READ MODE = 0	VDP
85H	R	TMS-9918A READ MODE = 1	VDP
88H	W	AY-3-8910 LATCH ADRESS	PSG
8CH	W	AY-3-8910 WRITE	PSG
90H	R	AY-3-8910 READ	PSG
96H	W	WRITE 8255 PORT C	PPI
97H	W	WRITE 8255 CONTROL WORD REGISTER	PPI
98H	R	READ 8255 PORT A	PPI
99H	R	READ 8255 PORT B	PPI

VDP = VIDEOPROZESSOR TMS-9918A (TMS-9929)
 PSG = PROGRAMMIERBARER TONGENERATOR
 PPI = PROGRAMMIERBARER PERIPHERIEBAUSTEIN

Die Signale des Z80 werden auch über den Expander-Bus an der Rückseite des Computers herausgeführt und gelangen so zu den Erweiterungen. Zusätzlich gibt es am Expander-Bus noch einige Steuersignale und die Spannungsversorgung + 12V und - 12V.

Wir fügen daher an dieser Stelle die Beschreibung des Expander-Busses ein, in der auch die CPU-Signale beschrieben werden.

Die Anschlußbelegung des Expander-Busses:

PIN	SIGNAL	I/O	FUNKTION
1	+ 5V	O	Spannungsversorgung + 5V, 300mA stehen für die Peripheriekarten zur Verfügung.
2	$\overline{\text{CNTRL2}}$	I	Kontrollsignal des Coleco-Adapters.
3	+ 12V	O	Spannungsversorgung + 12V, maximal 100mA für alle Peripheriekarten.
4	- 12V	O	Spannungsversorgung - 12V, maximal 50mA für alle Peripheriekarten.
5	$\overline{\text{CNTRL1}}$	I	Kontrollsignal des Coleco-Adapters.
6	$\overline{\text{WAIT}}$	I	Dient dazu, den Z80 mit langsameren Speichern oder Ein/Ausgabebausteinen zu synchronisieren.
7	$\overline{\text{RST}}$	I	Initialisiert den Mikroprozessor.
8	CPUCLK	O	Gepuffertes Systemtakt 3.58 MHz.
9-24	A15-A0		Gepuffertes 16 Bit-Adreßbus.
25	$\overline{\text{RFSH}}$	O	Gepuffertes REFRESH-Signal für die dynamischen RAMs der Speichererweiterungen. Das Signal zeigt an, daß die niederwertigen 7 Bits des Adreßbusses eine REFRESH-Adresse für die dynamischen RAMs enthalten.
26	$\overline{\text{EXCSR}}$	I	Wird nur vom Coleco-Adapter verwendet.
27	$\overline{\text{M1}}$	O	Maschinenzyklus 1, entspricht dem Fetch-Zyklus eines Befehls.
28	$\overline{\text{EXCSW}}$	I	Ebenfalls nur für den Coleco-Adapter.
29	$\overline{\text{WR}}$	O	Gepuffertes Schreibsignal. Es zeigt an, daß auf dem Datenbus gültige Daten liegen, die in den adressierten Speicher oder Baustein geschrieben werden können.
30	$\overline{\text{MREQ}}$	O	Gepuffertes Signal zur Anforderung des Speichers. Zeigt an, daß auf dem Adreßbus eine gültige Speicheradresse liegt.
31	$\overline{\text{IORQ}}$	O	Gepuffertes Signal zur Anforderung eines Ein-/Ausgabebausteins. Zeigt an, daß auf den Bits 0 bis 7 des Adreßbusses eine gültige Ein-/Ausgabeadresse liegt.
32	$\overline{\text{RD}}$	O	Gepuffertes Lesesignal. Zeigt an, daß die CPU

bereit ist, den Inhalt des Datenbusses in den Akkumulator zu laden.

33-40	D0-D7		Gepufferter bidirektionaler Datenbus.
41	CSOUND	I	Audio-Signal vom Coleco-Adapter.
42	$\overline{\text{INT}}$	I	Interruptanforderung durch Ein-/Ausgabebausteine.
43	$\overline{\text{RAMDIS}}$	I	Wenn dieses Signal auf LOW gezogen wird, wird das RAM des Computers weggeschaltet.
44	$\overline{\text{ROMDIS}}$	I	Wenn dieses Signal auf LOW gezogen wird, wird das BASIC-ROM weggeschaltet.
45	$\overline{\text{BK32}}$	O	Gepuffertes Speicherkontrollsignal. LOW bedeutet, daß die obere Hälfte von Bank 3 = Teil 32 (32KB von 8000H bis FFFFH) eingeschaltet ist. User-RAM ist mit RAMDIS abgeschaltet.
46	$\overline{\text{BK31}}$	O	Gepuffertes Speicherkontrollsignal. LOW bedeutet, daß die untere Hälfte von Bank 3 (Teil 31) (0000H bis 7FFFH) eingeschaltet ist, wobei das BASIC-ROM mit ROMDIS weggeschaltet ist.
47	$\overline{\text{BK22}}$	O	Gepuffertes Speicherkontrollsignal. LOW bedeutet, daß die obere Hälfte von Bank 2 (Teil 22) (8000H bis FFFFH) eingeschaltet ist. User-RAM ist mit RAMDIS abgeschaltet.
48	$\overline{\text{BK21}}$	O	Gepuffertes Speicherkontrollsignal. LOW bedeutet, daß die untere Hälfte von Bank 2 (Teil 21) (0000H bis 7FFFH) eingeschaltet ist, d.s. also jene 32 KByte RAM, die beim SV-328 bereits integriert sind. Das BASIC-ROM ist gleichzeitig weggeschaltet.
49-50	GND		Masse

In der kommenden Folge werden wir die Beschreibung des Z80 fortsetzen und uns dann dem Videoprozessor TMS-9918 zuwenden.

Nachtrag zur ersten Folge dieser Serie:

Mittlerweile ist der erste Super-Expander SVI-605B eingetroffen. Er verfügt wie die A-Version des SVI-605 über 2 Diskettenlaufwerke, aber mit einer Speicherkapazität von 320 KByte (formatiert) je Laufwerk. Es werden doppelseitige Disketten mit doppelter Dichte verwendet (80 Spuren). Eine Centronics-Schnittstelle zum Anschluß eines Druckers ist eingebaut.

Im Lieferumfang des neuen Super-Expanders ist auch ein umfangreiches Software-Paket enthalten: WordStar, Mailmerge, CalcStar, ReportStar und DataStar (eingetragene Warenzeichen von MicroPro International Corporation). Die Programme werden mit den dafür angebotenen Handbüchern geliefert.

Der Super-Expander SVI-605B wird im Computer-Studio zum Preis von S 29.990,- angeboten.

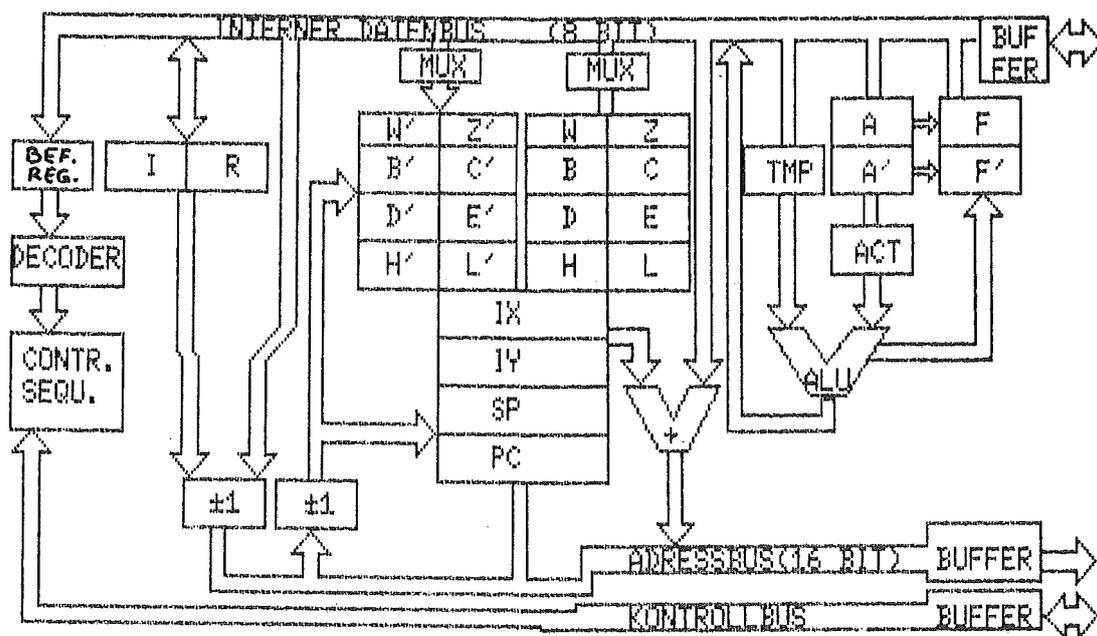
Nachdem letztesmal eine wichtige Grundvoraussetzung für die Programmierung in Assembler, das binäre und hexadezimale Zahlensystem, erklärt wurde, so werde ich jetzt die Struktur des Z-80 beschreiben.

Der Z-80 ist einer der bedeutendsten 8-Bit-Mikroprozessoren, wobei 8 Bit bedeutet, daß seine Register und sein Datenbus 8 Bit breit sind. Das heißt, daß maximal 1 Byte auf einmal übertragen werden kann. Der Z-80 besitzt Befehle, die bis zu 2 Bytes lang sein können und denen bis zu 2 Bytes an Daten folgen können. Jeder Befehl besitzt einen eigenen Operationscode, kurz Opcode, der festlegt, was der Mikroprozessor tun soll. Da die CPU jedes Byte von einem Befehl einlesen und decodieren muß, werden kürzere Befehle schneller ausgeführt.

Weil jeder Opcode im Speicher steht, und der Z-80 einen 16 Bit breiten Adressbus besitzt, braucht der Mikroprozessor ein 16-Bit-Register, in dem immer die Adresse des Befehls steht, der als nächster ausgeführt wird. Dieses Register heißt Programm-Counter, und wird "PC" abgekürzt. Der Inhalt kann nur durch Sprungbefehle verändert werden. Der "PC" wird immer dann auf den nächsten Befehl gestellt, wenn der vorige eingelesen und dekodiert wurde. Erst nach Erhöhen des Programm-Counters wird der dekodierte Befehl ausgeführt. Weiters gibt es einen Stackpointer, kurz "SP", in dem eine Adresse steht, bei welcher die Rücksprungadressen für die Unterprogramme abgelegt werden.

Doch bevor ich weitererkläre, möchte ich klarstellen, daß man unter einer Adresse 2 Bytes und unter Daten immer ein Byte versteht.

Aber der Z-80 besitzt auch andere Register, in denen Daten gespeichert und nach Belieben vertauscht werden können.



In der Mitte der Abbildung fällt ein großer Registerblock auf. Im oberen Teil sind zwei fast identische Blöcke erkennbar. Die sechs Register "B", "C", "D", "E", "H", "L" sind die Universalregister. Es kann immer nur mit einem der zwei Registerblöcke gearbeitet und

mit einem Befehl zwischen den Blöcken umgeschaltet werden. Rechts davon sind die Register "A" und "F" erkennbar. "A" und "F" bezeichnen die Zweitregister, und man kann wie bei den Universalregistern zwischen "A", "F" und "A'", "F'" umschalten. Das Register "A", wird auch Akkumulator oder kurz Akku genannt und ist das wichtigste Register der ganzen CPU. Mit ihm werden fast ohne Ausnahme alle Rechenoperationen durchgeführt. Unter ihm erkennt man die "ALU" (Arithmetik- und Logikeinheit) in der alle Rechenoperationen durchgeführt werden. "ACT" und "TMP" bezeichnen zwei Buffer, die aber für die Programmierung unwichtig sind, weil sie nicht angesteuert werden können. Das Flagregister wird beim Z-80 "F" genannt und ist rechts vom Akkumulator eingezeichnet. Das Flag-Register wird nur durch Rechenoperationen beeinflusst und zeigt an, ob bei der letzten Rechenoperation gewisse Ereignisse eingetreten sind (z.B. Übertrag).

Unter den Universalregistern sind die zwei Indexregister "IX" (Indexregister X) und "IY" (Indexregister Y) eingezeichnet. Sie sind spezielle Register und werden etwas anders gehandhabt als die Universalregister. Sie sind 16 Bit breit.

Rechts davon ist eine kleine "ALU" erkennbar. Beim Zugriff auf den Speicher mit Indexregistern wird noch ein Datenbyte angegeben, welches Offset genannt wird, und zu "IX" bzw "IY" addiert wird.

Oberhalb der Universalregister ist das Symbol "MUX" erkennbar. Es ist die Abkürzung für Multiplexer, und dieser hat die Aufgabe, die Daten, die über den Datenbus kommen, an die ausgewählten Register weiterzuleiten.

Links unterhalb des Registerblocks ist noch das Kästchen "+-1" erkennbar. Es ist der Inkrementierer/Dekrementierer. Jedes Register kann durch einen 1-Byte-Befehl (ausgenommen die Indexregister, bei denen man 2 Bytes benötigt) um 1 erhöht oder erniedrigt werden.

Zuletzt sind noch die beiden Register "I" und "R" erkennbar. Das "I"-Register ist das Interruptregister, daß ich jedoch erst später beschreiben will. Das Memory-Refresh-Register (kurz "R") wird höchstens zur Zufallsgenerierung verwendet und dient nur dazu, den dynamischen Speicher aufzufrischen.

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

```
*****
*
* Für alle, die mehr über den Mikroprozessor Z-80 wissen *
* wollen: *
* *
* Rodnay Zaks Programmierung des Z80 *
* *
* 606 Seiten, 200 Abbildungen, deutsch *
* inkl. 10 % MWSt. S 374,- *
* *
* Das Buch zum Z80. Mit ausführlicher Behandlung aller Be- *
* fehle, Z80 Hardware-Organisation, Adressierungstechniken *
* und mit Anwendungsbeispielen. *
* *
* Erhältlich im Computer-Studio, 1040 Wien, Paniglg. 18-20 *
*****
```

Variablenamen

Im Programmspeicher wird der eigentliche Variablenname immer direkt im ASCII-Code abgelegt. Bei Feldangaben werden jedoch die Indices, da sie nur indirekt zum Variablennamen gehören, als Zahlen interpretiert (ausgenommen es wird eine Variable eingesetzt). Das Gleichheitszeichen für die Variablenzuweisung wird wiederum als Token mit dem Wert &Hf1 gespeichert.

z.B: 44 32 28 12 2c 14 29 f1
 D 2 (1 , 3) =

Stringvariablen

Wer die erste Folge aufmerksam gelesen hat, dem wird aufgefallen sein, daß in der BASIC-Zeile ein String war, welcher mit den Anführungszeichen direkt im ASCII-Code abgelegt war. Genauso ist es auch bei den Stringvariablen.

z.B: 4e 24 f1 22 54 65 73 74 22
 N \$ = " T e s t "

Numerische Variablen

Bei den numerischen Variablen ist es schon etwas komplizierter, denn bei diesen muß man folgende Gruppen unterscheiden:

1. Nach dem Vorzeichen (Dies gilt für alle Gruppen)

1.1 positiv

Dem Variablenwert wird nichts vorgestellt.

z.B: 46 25 f1
 F % = Variablenwert

1.2 negativ

Dem Variablenwert wird der Token für das Minuszeichen mit dem Wert &Hf4 vorgestellt.

z.B: 5a f1 f4
 Z = - Variablenwert

2. Nach der Angabeform (dezimal,hexadezimal,octal,binär)

2.1 dezimal

2.1.1 ganzzahlig mit den Werten 0 bis 9

Der Variablenwert wird zur Basis &H11 addiert.

z.B: 4a f1 15
 J = 4

2.1.2 ganzzahlig mit den Werten 10 bis 255

Der Variablenwert wird vom Wert &H0f eingeleitet, und der Wert selbst wird in einem Byte abgelegt.

z.B: 45 f1 f4 0f 0a
 E = - 10

2.1.3 ganzzahlig mit den Werten 256 bis 32767

Der Variablenwert wird vom Wert &H1c eingeleitet, und der Wert selbst wird in zwei Byte abgelegt, wobei zuerst das niederwertige (LB) und dann das höherwertige Byte (HB) abgelegt wird.

z.B: 4b f1 1c 8f 02
 K = LB HB LB + HB = Variablenwert
 143+ 512= 655

2.1.4 ganzzahlig mit den Werten über 32767 und Gleitkommawerte

Der Variablenwert wird von zwei Bytes eingeleitet. Das

erste Byte kann folgende Werte enthalten:

&H1d Der Variablenwert ist einfacher Genauigkeit; d.h. der Variablenwert enthält max. 6 Stellen. Der Wert selbst wird in 6 Halbbyte hexadezimal abgelegt.

&H1f Der Variablenwert ist doppelter Genauigkeit; d.h. der Variablenwert enthält max. 14 Stellen. Der Wert selbst wird in 14 Halbbyte hexadezimal abgelegt.

Beim zweiten Byte wird zur Basis von &H40 der Wert, welcher angibt an wievielter Stelle der Dezimalpunkt steht, addiert.

z.B: 54 21 f1 f4 1d 42 32 76 00
T ! = - 32 76
T!=-32.76

4d f1 1f 40 10 12 34 50 00 00 00
M = 10 12 34 5
M=.1012345

2.2 hexadezimal

Der Variablenwert wird vom einem Byte mit dem Wert &H0c eingeleitet. Der Wert selbst wird in zwei Byte, mit der Reihenfolge niederwertiges (LB) und höherwertiges (HB) Byte abgelegt.

z.B: 48 45 f1 0c 7a c3
H E = &H LB HB
HE=&HC37A

2.3 octal

Der Variablenwert wird vom einem Byte mit dem Wert &H0b eingeleitet. Der Wert selbst wird in zwei Byte, mit der Reihenfolge niederwertiges (LB) und höherwertiges (HB) Byte abgelegt.

z.B: 4f 35 f1 0b f4 0b
O 1 = &O LB HB
O1=&O5764

2.4 binär

Die Abspeicherung von binären Werten weist eine Besonderheit auf. Es ist nämlich die einzige Form bei den numerischen Variablen, welche direkt im ASCII-Code abgelegt wird.

z.B: 4d 35 f1 26 42 31 30 30 30 30
M 5 = & B 1 0 0 0 0
M5=&B10000

Fortsetzung folgt

Wolfgang Rotschek

SOMMERENGPASS AN BLUTKONSERVEN BEI DER BLUTSPENDEZENTRALE DES ÖSTERREICHISCHEN ROTEN KREUZES

Die Blutspendezentrale braucht monatlich ca. 15.000 Blutkonserven. Das ROTE KREUZ ruft aus diesem Grund alle gesunden Männer und Frauen im Alter von 18-60 Jahren auf, Blut zu spenden.

BITTE HELFEN AUCH SIE !

Die Blutspendezentrale des Österreichischen Roten Kreuzes für Wien, Niederösterreich und Burgenland in Wien 4., Gußhausstraße 3, hat Montag bis Freitag von 8.00 Uhr bis 17.30 Uhr und Samstag von 8.00 Uhr bis 11.00 Uhr geöffnet.

BYTES-SUCHPROGRAMM

Das Programm sucht eine Folge von bis zu 64 Bytes im ROM und RAM der Bank 0 des SV-328. Die Übergabe der Bytes erfolgt in zwei-stelliger, hexadezimaler Darstellung. Es sind nur die Ziffern '0' bis '9' und die Buchstaben 'A' bis 'F' zugelassen.

Im folgenden Beispiel wird die Bytefolge 21,2B,F9 gesucht (entspricht 'LD HL,F92B').

```
PRINTUSR("212BF9")
```

Das Programm durchsucht nun die 64K der Bank 0 nach dieser Bytefolge und zeigt die Adresse(n) an, wo sich diese Bytefolge befindet. Jede Adresse wird als vierstellige Hexadezimalzahl dargestellt.

Nach dem Suchen wird der übergebene String noch einmal angezeigt.

Erfolgt das Suchen mit der 'LPRINT'-Anweisung, so wird das Ergebnis an den Drucker geschickt, zum Beispiel:

```
LPRINT USR("212BF9")
```

Der Einstieg in das Maschinenprogramm mit einer Zahl als Argument erzeugt die Fehlermeldung 'Type mismatch'. Wird eine ungerade Anzahl von Zeichen übergeben, dann antwortet der Computer mit 'Missing operand'. Ist der übergebene String leer oder wird ein unzulässiges Zeichen mitübergeben, dann meldet sich der Computer mit 'Illegal function call'. Ist die Länge des Strings größer als 128, dann erfolgt die Fehlermeldung 'String too long'.

Nach dem Laden und Starten des BASIC-Programmes wird in Zeile 150 und 160 festgestellt, ob mit Diskette oder Kassette gearbeitet wird. Dementsprechend wird mit der 'CLEAR'-Anweisung der Platz für das Maschinenprogramm geschaffen.

Mit 'GOTO 460' kann man das fertige und angepaßte Maschinenprogramm abspeichern. Hierbei überprüft das Programm, ob das Disk-BASIC vorhanden ist oder nicht. Wenn es vorhanden ist, wird automatisch das Laufwerk 1 angesprochen, im andern Falle der Kassettenrecorder.

Die Zeilen 510 bis 540 enthalten eine kurze Routine, um maximal 256 Bytes zu überprüfen. Diese Routine ist allgemein für jede Checksum-Prüfung geeignet und ist durch die 'XOR'-Funktion sicherer als eine einfach gebildete Summe.

Das Maschinenprogramm verwendet außer den ROM-Routinen für die Fehlermeldungen auch die 'PRINT'-Routine des ROMs. Der Einsprung in diese Routine von Maschinsprache her erfolgt mit 'RST 18H', wobei der Akkumulator den ASCII-Wert des auszugebenden Zeichens enthält. Das Zeichen wird bei der laufenden Cursorposition angezeigt und danach der Cursor um eine Position nach rechts verschoben. Das Suchprogramm verwendet diese Routine, um die vierstelligen Adressen anzuzeigen.

In der nächsten Ausgabe bringen wir ein dokumentiertes Assembler-listing zu diesem Maschinenprogramm.

Das Listing zu dem BASIC-Programm finden Sie auf der folgenden Seite.

```

100 '
110 ' Bytes suchen
120 ' Version 2.5 vom 11.08.1984
130 ' Philipp DTT
140 '
150 GOSUB 510
160 IF B=145 THEN CLEAR 200,&HD000:D=0 ELSE CLEAR 200,&HF000:D=1
170 RESTORE 310:DEFINT A:DEFSTR I
180 A=&HD000+D*B192
190 READ I
200 IF I="**" THEN 260
210 IF I="++" THEN I=HEX$(&HD0+D*32)
220 IF I="+-" THEN I=HEX$(&HD1+D*32)
230 POKE A,VAL("&H"+I)
240 A=A+1
250 GOTO 190
260 DEFUSR=&HD000+D*B192
270 PRINT "Das Programm wurde eingelesen."
280 PRINT "RAMTOP liegt nun bei ";HEX$(&HD000+D*B192)
290 PRINT USING "##### Bytes frei,davon ##### fuer Strings. ";FR
E(O)+FRE("");FRE("")
300 END
310 DATA C3,08,++,00,00,00,00,00,FE,03,C2,05,09,23,23,4E
320 DATA 23,46,C5,E1,7E,A7,CA,9E,0F,1F,1E,18,DA,07,09,FE
330 DATA 41,1E,0F,D2,07,09,32,07,++,23,4E,23,46,ED,43,03
340 DATA ++,F3,2A,03,++,11,00,+-,3A,07,++,47,CD,0B,17,CD
350 DATA DB,++,DA,9E,0F,D6,30,FE,0A,DA,4E,++,D6,07,87,87
360 DATA 87,87,4F,23,CD,0B,17,CD,DB,++,DA,9E,0F,D6,30,FE
370 DATA 0A,DA,66,++,D6,07,81,12,23,13,10,++,21,00,00,22
380 DATA 05,++,FB,2A,05,++,23,22,05,++,7C,B5,CA,D1,++,11
390 DATA 00,+-,3A,07,++,47,1A,BE,13,23,C2,77,++,10,F7,2A
400 DATA 05,++,7C,FE,+-,C2,9E,++,7D,FE,00,CA,73,++,3A,06
410 DATA ++,CD,BE,++,3A,06,++,CD,C4,++,3A,05,++,CD,BE,++
420 DATA 3A,05,++,CD,C4,++,3E,20,CD,CF,++,C3,73,++,E6,FO
430 DATA 1F,1F,1F,1F,E6,0F,C6,30,FE,3A,FA,CF,++,C6,07,DF
440 DATA C9,3E,0D,CD,CF,++,3E,0A,C3,CF,++,FE,30,DB,FE,3A
450 DATA 3F,D0,FE,41,DB,FE,47,3F,C9,00,00,00,00,00,00,**
460 ' Speichern nach einlesen
470 I$="SEEBYT"
480 GOSUB 510:IF B<>145 THEN D=1 ELSE D=0:I$="1:"+I$
490 BSAVE I$,&HD000+D*B192,&HD0FF+D*B192,&HD000+D*B192
500 END
510 ' Checksum
520 DEFINT A,B:FOR A=&HE000 TO &HEOFF
530 B=B XOR (PEEK(A)):NEXT
540 RETURN

```

```

*****
*   Das richtige BASIC-Lehrbuch zum Spectravideo-Computer   *
*   *                                                         *
*   Werner Chmel           324 Seiten, 300 Abbildungen     *
*   Basic-Kompendium      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 *
*****

```

PRIMZAHLENBERECHNUNGEN

Mit diesem Programm wird es für Mathematiker leicht, Primzahlen zu berechnen. Man gibt einfach den Zahlenbereich ein, von dem man die Primzahlen wissen will. Der Computer rechnet danach alle aus und zeigt sie selbsttätig an.

Kommen wir nun zur Programmbeschreibung: Nach einer kurzen Einleitung, die bis Zeile 160 geht, fragt der Computer nach dem Zahlenbereich. Die Variable "AB" speichert die erste Zahl und die Variable "BIS" die letzte Zahl des Bereichs. Danach wird geprüft, ob vernünftige Werte eingegeben wurden, andernfalls springt der Computer wieder zur Eingabe zurück.

Um nicht unnötig viel Speicherplatz zu verschwenden, wird das Variablenfeld, in das die berechneten Primzahlen eingespeichert werden, in der Zeile 220 möglichst knapp dimensioniert. Da keine geraden Zahlen (außer 2) Primzahlen sein können, wird für die höchstzulässige Anzahl der Zahlen die Hälfte des eingegebenen Bereichs verwendet.

Wie oben erwähnt ist die Zahl 2 die einzige gerade Primzahl, und wird deshalb im Programm gesondert berechnet. Mit einer "IF...THEN" Anweisung wird geprüft, ob die 2 im Testbereich ist. Wenn sie es ist, speichert sie der Computer als erste Primzahl ab. Mit der Variable NR zeigt der Computer übrigens immer auf das erste freie Feld, in das Primzahlen eingespeichert werden können.

Nun kommt der eigentliche Berechnungsteil, der aus zwei ineinander geschachtelten Schleifen besteht. Die erste Schleife mit der Variable "I" läuft den ganzen Zahlenbereich durch und erzeugt so für die zweite Schleife die zu testenden Zahlen. Da nur ungerade Zahlen ohne Teiler sein können, wird aus Gründen der Geschwindigkeit jede gerade ausgelassen. Dies wird mit einer "IF...THEN"-Anweisung gemacht.

Um das Programm völlig zu verstehen, muß man die mathematischen Grundlagen der Primzahlenberechnung kennen. Die einfachste Methode ist, die zu testende Zahl X nacheinander durch die Werte 2-X zu dividieren (also X/2, X/3, X/4 usw.). Sollte bei einer Division kein Rest vorhanden sein, ist die Zahl X keine Primzahl.

Eine schnellere Art basiert auf der Methode, das von zwei Teilern der Zahl X, die miteinander multipliziert X ergeben, immer ein Teiler kleiner oder gleich der Quadratwurzel von X sein muß. Da aber ein Teiler, der größer als die Quadratwurzel von X ist, immer einen Teiler kleiner \sqrt{X} besitzt, braucht man lediglich die Zahlen von zwei bis \sqrt{X} nacheinander durch X dividieren. Und genau dies wird in der zweiten, der "J"-Schleife gemacht. Sollte sich eine Primzahl "zu erkennen geben", wird sie im Variablenfeld PRIM(NR) abgespeichert und der Zeiger NR inkrementiert.

Nach der Berechnung fragt der Computer, ob noch einmal gerechnet werden soll, oder ob die ausgerechneten Primzahlen angezeigt werden sollen. Es kann auch nur eine Primzahl abgefragt werden.

Interessant ist hier die Eingaberoutine, die mit INKEY\$ arbeitet. Dieser Befehl wartet nicht selbständig, bis eine Eingabe erfolgt, sondern man springt mit einer "IF...THEN"-Anweisung immer wieder zum INKEY\$ zurück, solange keine Taste gedrückt wird.

Gerhard Fally

```

10 REM *****
20 REM ***
30 REM *** PRIMZAHLENPROGRAMM VERS 1.1
40 REM ***
50 REM ***      VON T. MIKULA      1984
60 REM ***
70 REM *** EDITED BY G. FALLY      1984
80 REM ***
90 REM *****
100 CLS
110 SCREEN2:PRINT
120 PRINT" PRIM
130 PRINT" ZAHLEN
140 PRINT" PROGRAMM
150 FORI=1TO1000:NEXT
160 SCREEN0,0
170 INPUT "INTERVAL VON ";AB
180 PRINT
190 INPUT" BIS ";BIS
200 LOCATE,,0
210 IFBIS<ABTHEN230
220 ANZ=INT((BIS-AB)/2):DIMPRIM(ANZ)
230 IFAB<=2THEN240ELSEGOTO260
240 IF BIS>1THENPRINT"2 IST EINE PRIMZAHL"ELSE260
250 NR=1:PRIM(NR)=2
260 FORI=AB TO BIS
270 IF I/2=INT(I/2)THEN330
280 FORJ=2TOSQR(I)
290 IF I/J=INT(I/J) OR I=1THEN330
300 NEXTJ
310 PRINTI"IST EINE PRIMZAHL
320 NR=NR+1:PRIM(NR)=I
330 NEXTI:PRINT"ENDE"
340 FORS=1TO1000:NEXTS
350 CLS:PRINT"ZWISCHEN"AB"UND"BIS"SIND"NR:PRINT"PRIMZAHLEN"
360 PRINT"NEUE BERECHNUNGEN?
370 A$=INKEY$:IFA$=""THEN370
380 IFA$="J"ORA$="j"THENRUN
390 PRINT"1)ALLE PRIMZAHLEN ZWISCHEN
400 PRINTAB"UND"BIS"!
410 PRINT"2)EINE PRIMZAHL!
420 PRINT"3)AUSSTIEG AUS DEM PROGRAMM!
430 A$=INKEY$:IFA$=""THEN430
440 IFA$="1"THEN470
450 IFA$="2"THEN510
460 IFA$="3"THENCLS:ENDELSE340
470 FORI=1TO NR
480 PRINTPRIM(I)"IST EINE PRIMZAHL
490 FOR W=1TO200:NEXTW:NEXTI
500 GOTO 340
510 PRINT"DIE WIEVIELTE PRIMZAHL ?
520 PRINT"BITTE GEWUENSCHTE ZAHL EINGEBEN!
530 INPUTD
540 IF D>NR THEN PRINT"ES GIBT ZWISCHEN"AB"UND"BISELSE560
550 PRINT"NUR"NR"PRIMZAHLEN
560 PRINTPRIM(D)"IST DIE PRIMZAHL NUMMER"D
570 PRINT"ZWISCHEN"AB"UND"BIS"!
580 FORI=1TO2000:NEXT:GOTO 340
590 IF ERR=7THENCLSELSE620
600 PRINT"ZWISCHEN"AB"UND"BIS"LIEGEN
610 PRINT"ZU VIELE PRIMZAHLEN
620 RESUME150

```

IMPRESSUM:

Chefredakteur: Gerhard Fally

Ständige freie Mitarbeiter: Wolfgang Rotschek, Heinz Schmid, Georg Wolfbauer

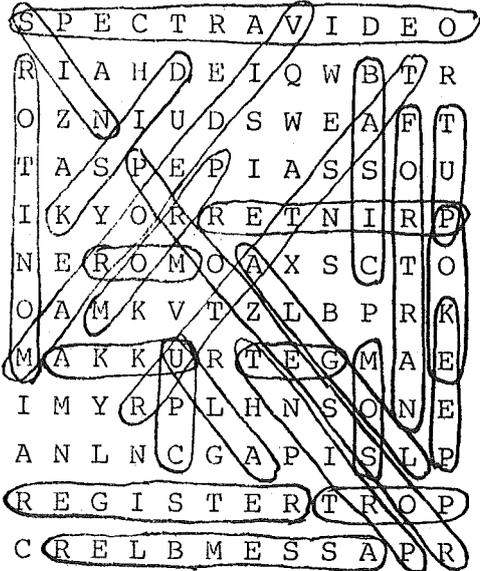
Medieninhaber (Verleger): Spectra Video Club Austria, p.A. Computerstudio, A-1040 Wien, Paniglg. 18-20, Tel. (0222) 65 88 93

Hersteller: Offsetschnelldruck Riegeltechnik, 1080 Wien

Herausgeber: Spectra Video Club Austria, p.A. Computerstudio, A-1040 Wien, Paniglg.18-20, Tel.65 88 93

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

Lösung des Rätsels aus Heft 1:



 * **SONDERANGEBOT** Gültig bis 7. September 1984 *
 * + Datenrecorder SV-903 *
 * + 2 Joysticks *
 * + 4 Programmkassetten *
 * **SV-328** *
 * Paketpreis nur 7990.- *
 * **Computer-Studio** *
 * 1040 Wien, Paniglgasse 18-20 (neben TU) *
 * Tel. 65 78 08, 65 88 93, Mo-Fr 9-18, Sa 9-12 *



Was bietet Ihnen der Spectra Video Club Austria?

Regelmäßige Clubabende mit Gelegenheit zum Informationsaustausch! Kostenloses Arbeiten an SV-318 und SV-328-Systemen und Möglichkeiten zum Verwenden von Druckern!

Freier Bezug des SVi-Journals, unserer Clubzeitschrift! Verbilligte Angebote von Spectravideo-Produkten!

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

Ich interessiere mich für die Mitgliedschaft beim "Spectra Video Club Austria" und wünsche die Zusendung näherer Informationen und der Anmeldungsunterlagen.

Name

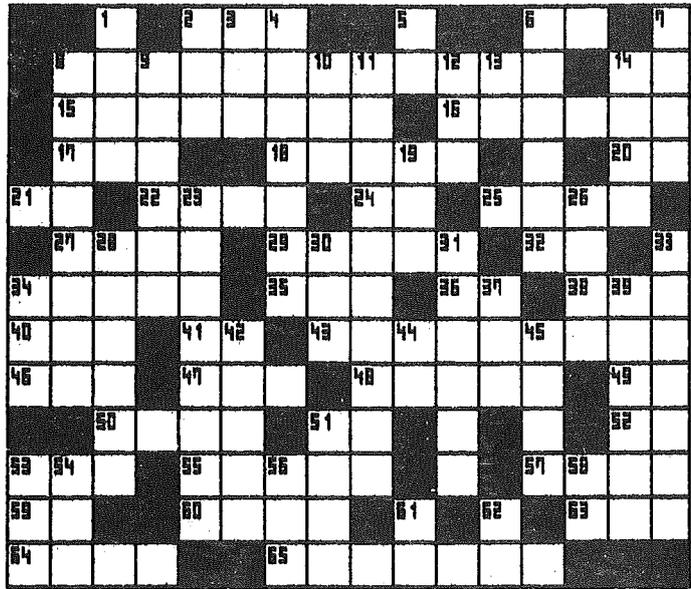
Adresse

Bitte per Post an "Spectra Video Club Austria", p.A. Computer-Studio, 1040 Wien, Paniglgasse 18-20 senden oder persönlich abgeben. Die Abgabe dieses Coupons ist unverbindlich!

KREUZWORTRÄTSEL

Waagrecht:

2) ganzes Streicherorchester (italienische Abk.)-6) pers. Fürwort-8) kleiner Spielcomputer-14) chem. Zeichen für Tantal-15) Abnormalität-16) Fehler-17) selten-18) Geistesgestörter-20) chem. Zeichen für Neon-21) lat. zu-22) völlig-24) Vorwort-25) Behälter für Asche-27) europ. Hauptstadt-29) männliche Ente-32) Abk. für Eingabe/Ausgabe-34) Fluß in Frankreich-35) pers. Fürwort-36) Abk. für Amplitudenmodulation-38) jetzt-40) BASIC-Funktion-41) Abk. für "knock out"-43) Variation-46) Göttin (lat.)-47) Vorsilbe (drei-)-48) geometrische Figur-49) chem. Zeichen für Selen-50) chemisches Element-51) Faultier-52) Stadt in der UdSSR-53) logische Verknüpfung-55) Nadelbaum-57) Ziffer (engl.)-59) Doppelvokal-60) Staatshaushalt-63) engl. Mädchenname-64) Gegenteil von "gefangen"-65) Verdienste



Senkrecht: 1) die Gute (lat.)-2) BASIC-Befehl-3) Ort in Peru-4) staatliche Institution-5) Feutbiotop-6) irren (lat.)-7) vornehme Frau-8) anfaßbare Computerteile-9) Tageszeit-10) pers. Fürwort (3.Fall)-11) Peripherie-12) Kennzeichen von LKWs mit internationaler Ladung-13) pers. Fürwort-14) Feinabstimmung (engl.)-19) Fluß in Peru-23) kurze heitere Begebenheit-26) erstaunter Ausruf-28) europ. Staat-30) amerikanischer Elektronikkonzern-31) elektr. Einheit-33) aufschneiden-34) Abk. für Medizin-37) er traf (engl.)-39) Nonsense-42) Festgewand-44) ugs. nein-45) sauber-51) vor (lat.)-53) Vorwort-54) lediglich-56) denn (lat.)-58) Eselslaut-61) Keimling-62) Abk. für Musikerziehung

SPECTRAVIDEO

SV-318/328

Die Computer für alles.

Freizeit und Beruf.

SPECTRAVIDEO