ASCII-Eingabe –

so ganz nebenbei

Mikrocomputer sind leider die meiste Zeit ihres Daseins damit beschäftigt, darauf zu warten, bis irgendein verarbeitungswürdiges Ereignis eintrifft, z. B. bis der Benutzer ein paar Buchstaben auf seiner Terminal-Tastatur eintippt. Daß man diese Eingabe von ASCII-Zeichen nicht zur Hauptbeschäftigung eines Mikroprozessors machen muß, zeigt das folgende Beispiel.

Die in diesem Beitrag besprochenen Programm-Routinen entstanden aus dem Wunsch, seriell eintreffende ASCII-Zeichen als einfaches Siebensegment-Alphabet auf dem sechsstelligen Display des Mikrocomputers KIM-1 darzustellen. Die Buchstaben sehen dabei zwar etwas seltsam aus, sind aber dennoch gut lesbar. Eine Anzeigeroutine für dieses Siebensegment-Alphabet wurde bereits im HOBBYCOMPUTER-Sonderheft des Franzis-Verlages abgedruckt.

Kettet man in einem Einfachst-Programm die KIM-ROM-Empfangsroutine „GETCH" und das erwähnte Siebensegment-Programm nebst einer einfachen Routine zum Verschieben der Zeichen auf dem Display aneinander, so ist der „Erfolg" deprimierend: Das Siebensegment-Display blitzt nur nach dem Empfang jeden ASCII-Zeichens kurz auf und bleibt dann wieder dunkel, da ja der Prozessor völlig damit beschäftigt ist, auf das Start-Bit des nächsten Zeichens zu warten. Bei hohen Übertragungsgeschwindigkeiten, wie etwa 600 Baud (bit/s), ist auf dem Display zwar schon etwas zu erkennen, aber erstens laufen die Buchstaben dann so schnell durch, daß man nicht mehr mitlesen kann, und zweitens treten auch noch Übertragungsfehler auf, weil die Display-Routine so lange dauert, daß bis zu ihrem Ende der Anfang des nächsten Zeichens bereits verpaßt wurde. So funktioniert es also nicht!

Die Lösung: Der Timer-Interrupt

Zum Glück verfügen viele Mikrocomputersysteme, auch der KIM-1, über einen taktgesteuerten Timer, der den Prozessor veranlassen kann, nach Ablauf einer bestimmten, programmierbaren Zeit eine Interrupt-Routine abzuarbeiten und danach in das Hauptprogramm zurückzukehren, als wäre nichts geschehen. Beim KIM-1 ist dazu die Leitung PB7 (sie ist der Interrupt-Ausgang des Timers im IC 6530-003) mit der NMI-Leitung zu verbinden.

Der Timer gliedert sich intern in zwei Teile: einen Vorteiler, der den 1-MHz-Systemtakt durch 1, 8, 64 oder 1024 teilt, und den eigentlichen Timer, der aus einem achtstufigen Binärzähler besteht. Dieser Binärzähler läßt sich per Programm auf einen beliebigen Wert zwischen Null und 255 setzen und zählt von diesem Wert aus abwärts. Ist er bei Null angelangt, so wird – wenn dieses programmiert wurde – eine negative Flanke an PB7 erzeugt, die den Interrupt auslöst.

Bild 1 zeigt das Flußdiagramm für eine Interrupt-Routine, deren Aufgabe es ist, ein ASCII-Zeichen unabhängig von einem beliebigen parallellaufenden Hauptprogramm zu empfangen und in einen Buffer abzuspeichern. Dieser Buffer wird indirekt adressiert, nämlich durch die Zellen 00EB und 00EC, wobei 00EC den höherwertigen Adreßteil enthält und 00EB den niederwertigen. Ist ein Zeichen vollständig empfangen und ist sein Wert nicht 00, so wird der niederwertige Adreßteil um 1 erniedrigt und das Zeichen an die sich dadurch ergebende neue Buffer-Adresse abgespeichert.

Diese Methode ermöglicht es, bis zu 255 Zeichen vor einer weiteren Verarbeitung durch das Hauptprogramm zwischenzuspeichern; denn schließlich werden die Zeichen u. U. langsamer ausgewertet, als sie empfangen werden, wie dies auch in unserem Anwendungsbeispiel der Fall ist.

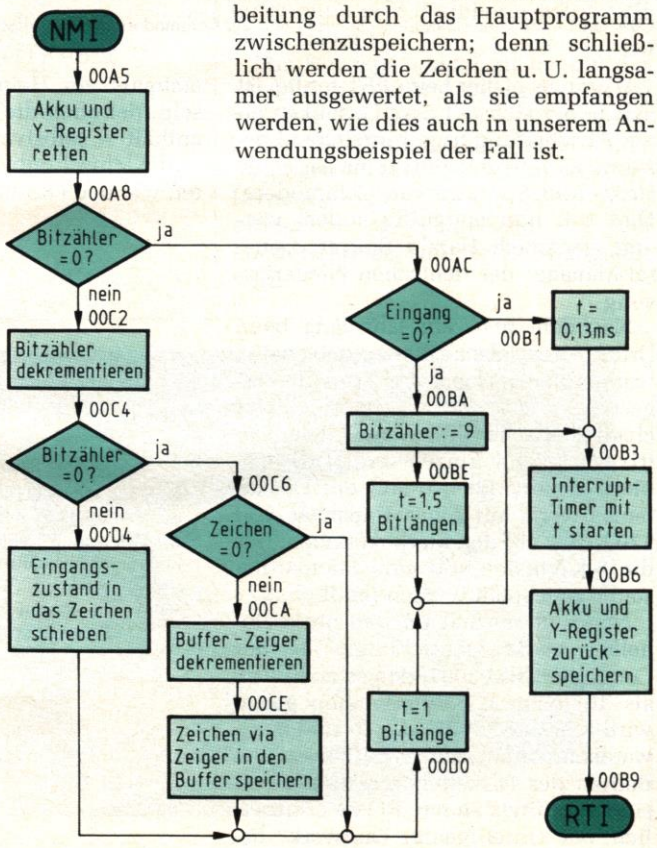

Bild 1. Flußdiagramm der Interrupt-Routine zum Empfang von seriellen ASCII-Zeichen und deren Abspeichern in einem indirekt adressierten Buffer

```
00A5 48       PHA
00A6 98       TYA
00A7 48       PHA
00A8 A5 E8    LDA E8
00AA D0 16    BNE 00C2
00AC 2C 40 17 BIT 1740
00AF 10 09    BPL 00BA
00B1 A9 02    LDA #02
00B3 8D 0E 17 STA 170E
00B6 68       PLA
00B7 A8       TAY
00B8 68       PLA
00B9 40       RTI
00BA A9 09    LDA #09
00BC 85 E8    STA E8
00BE A9 4D    LDA #4D
00C0 D0 F1    BNE 00B3
00C2 C6 E8    DEC E8
00C4 D0 0E    BNE 00D4
00C6 A5 EA    LDA EA
00C8 F0 06    BEQ 00D0
00CA C6 EB    DEC EB
00CC A0 00    LDY #00
00CE 91 EB    STA (EB),Y
00D0 A9 34    LDA #34
00D2 D0 DF    BNE 00B3
00D4 18       CLC
00D5 2C 40 17 BIT 1740
00D8 10 01    BPL 00DB
00DA 38       SEC
00DB 66 EA    ROR EA
00DD 18       CLC
00DE 90 F0    BCC 00D0
```

Bild 2. 6502-Programm als Interrupt-Routine entsprechend Bild 1

```
0000 A9 00 8D FB 17 A9 A5 8D FA 17 A2 08 A9 03 95 E6
0010 CA 10 FB 8E 0E 17 A9 7F 8D 41 17 A2 09 A0 06 84
0020 FC 98 18 65 ED 85 E6 A0 00 B1 E6 A8 C0 30 90 04
0030 C0 5B 90 02 A0 2F B9 4A 00 A0 00 8C 40 17 8E 42
0040 17 8D 40 17 A0 7F 88 D0 FD E8 E8 A4 FC 88 D0 CF
0050 20 3D 1F D0 16 C6 E9 D0 BD A9 30 85 E9 A5 ED 38
0060 E5 EB C9 FF F0 B0 C6 ED 18 90 AB 20 6A 1F C9 15
0070 10 A4 4C 00 02 00 00 00 00 00 3F 06 5B 4F 66 6D
0080 7D 07 7F 6F 00 00 00 41 00 53 00 77 7C 58 5E 79
0090 71 3D 74 05 1E 78 38 37 54 5C 73 67 50 6D 31 3E
00A0 1C 7E 76 6E 5B


0200 A2 00 8E F3 17 8E 04 17 A2 EB 8E F2 17 09 80 A2
0210 29 DD 00 02 F0 03 E8 D0 F8 E8 BD 00 02 10 03 4C
0220 00 00 20 A0 1E D0 F2 F0 F6
```

Bild 3. Dieses Programm ergibt zusammen mit der Routine aus Bild 2 einen praktischen Terminal-Tester. Startadresse ist 0000; vergessen Sie nicht, die NMI-Leitung des KIM-1 mit dem Timer-Interrupt-Anschluß PB 7 zu verbinden

Das fertige Programm

Bild 2 zeigt den disassemblierten 6502-Maschinencode für die Interrupt-Routine. Es muß noch erwähnt werden, daß sie den Interrupt-Timer selbst wieder startet; zu Beginn des Hauptprogramms muß jedoch „künstlich" ein NMI-Interrupt erzeugt werden, um die Interrupt-Routine das erste Mal anzuspringen. Das Programm in *Bild 3* tut dies; und es tut noch einiges mehr: Da, wie schon erwähnt, bei hohen Übertragungsgeschwindigkeiten die auf dem Display von rechts nach links durchlaufenden Zeichen nicht mehr mitgelesen werden könnten, wird der Zeichenbuffer recht langsam ausgelesen, nämlich mit einer Geschwindigkeit von nur etwa 3 Zeichen pro Sekunde (die genaue Zeit läßt sich in der Zelle 005A in weiten Grenzen wählen). Der „Vorlauf" der Interrupt-Eingabe-Routine kann bis zu 255 Zeichen betragen. Der Zeichenbuffer wird dabei im Adressenbereich 0300...03FF aufgebaut. Da große Teile des Hauptprogramms dem schon erwähnten Programm zur Siebensegment-Anzeige von Buchstaben sehr ähneln, wurde es lediglich in hexadezimaler Form abgedruckt. (Ausgewertet werden alle ASCII-Zeichen mit Hex-Codes zwischen 30 und 5A.)

Neben seinem ursprünglichen Verwendungszweck, nämlich für ASCII-Amateurfunkfernschreiben, läßt sich das Programm auch recht gut für den Test von Terminals und UART-Schaltungen einsetzen. Dabei lassen sich praktisch alle üblichen Übertragungsgeschwindigkeiten einstellen, wie aus *Tabelle 1* hervorgeht.

Jetzt fehlt uns für diesen Verwendungszweck also nur noch eine einfache Routine, die vorher programmierte Texte und Zeichenfolgen seriell ausgeben kann.

Das Sendeprogramm

Um auch fest programmierte Zeichenfolgen per Tastendruck über den seriellen ASCII-Anschluß des KIM senden zu können, ist im Adressenbereich 0200...0228 ein kleines Programm untergebracht, das mit Hilfe des KIM-ROM-Unterprogramms „OUTCH" Standardtexte ausgeben kann, die zwischen 0229 und 02FF stehen. Diese Standardtexte haben das Format

Tabelle 1: Wichtige Adressen im Programm

Zero-Page-Adressen
E6 Display-Zeiger L
E7 Display-Zeiger H
E8 Bit-Zähler
E9 Display-Verzögerung
EA Empfangenes Zeichen
EB NMI-Zeiger L
EC NMI-Zeiger H
ED Displ.-Zeiger-Index
FC Y-Zwischenspeicher
(Werden vom Programm gesetzt)

Geschwindigkeits-Einstellung

Baud▶	110	300	600
00BF	D4	4D	26
00D1	8E	34	1A
0201	02	00	00
0209	7D	EB	73

Tabelle 2: Hex-Code der ASCII-Zeichen

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENG	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	VS
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	∅	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	↑	_
6		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	'	}	~	DEL

Die waagrechten Ziffern stellen das niederwertige, die senkrechten das höherwertige Halbbyte dar; das Zeichen „e" ist z. B. hex 65. Die wichtigsten Steuerzeichen sind: CR = Wagenrücklauf (Carriage Return), LF = neue Zeile (Line Feed), BS = Cursor rückwärts (Back Space), SP = Leerraum (Space).

ID Text ID Text ID Text...FF.
ID ist dabei eine Hexzahl zwischen 80 und FE. Sie errechnet sich aus der Summe von 80 und dem Hex-Code derjenigen KIM-Taste, der der jeweilige Text zugeordnet werden soll. Das Byte FF sagt dem Computer, daß die Texttabelle hier zu Ende ist. Der Text selbst ist mit einer ASCII-Tabelle zu programmieren (*Tabelle 2*).

Das gewählte Format hat den Vorzug, daß zum Aufrufen des gewünschten Textes nicht die absolute Adresse dieses Textes bekannt sein muß. Das Auffinden geschieht vielmehr über eine Marke ID (Label). Eventuelle Änderungen der Texttabelle werden damit besonders einfach. *Herwig Feichtinger*

TRS-80
Viel Computer für wenig Geld

Heute, zwei Jahre nach ihrer Ankündigung, sind es immer noch zwei Geräte, die den Markt der betriebsfertigen Heimcomputer beherrschen: der PET 2001 von Commodore und der TRS-80 von Radio Shack (Tandy). Während der PET Vorteile im Hinblick auf Laborautomatisierung und automatische Meßwertverarbeitung aufweist (IEC-Bus), wird man den TRS-80 vorziehen, wenn man den Blick auf kleinere EDV-Projekte richtet.

Mit dem sehr leistungsfähigen Level-II-BASIC (Befehlssatz siehe *Tabelle*), 4 KByte RAM, 12-Zoll-Monitor, Kassettenrecorder und Netzteil kostet der TRS-80 rund 2100 DM. Die Tastatur entspricht etwa der einer Schreibmaschine, d. h. es ist keine unerträgliche Zumutung, wenn man z. B. über längere Zeit Daten eingeben muß. Im selben Gehäuse wie die Tastatur ist auch der komplette Computer mit der Logik für die Monitor-Ansteuerung untergebracht. Über einen Leiterplattenstecker an der Rückseite des Gerätes sind sämtliche wichtigen Signale der Z80-CPU herausgeführt. Hier wird eine sogenannte Erweiterungs-Box angeschlossen, mit deren Hilfe man die RAM-Kapazität auf insgesamt 48 KByte (im Grundgerät 16 KByte möglich) erweitern kann. Außerdem ist diese Box Voraussetzung für den Anschluß eines zweiten Kassettenrecorders, eines Mini-Floppy-Disk-Laufwerks (bis zu vier möglich) oder eines Druckers. Wer sich in der Schaltungstechnik von Mikroprozessoren etwas auskennt, der kann an den Erweiterungsstecker aber auch einen Peripherie-Baustein (PIO, PIA oder ähnliches) anschließen, um Ein/Ausgabe-Kanäle zur Verfügung zu haben. Bedauerlich ist, daß man in der Grundausführung ohne zusätzlichen Aufwand keine externen Einheiten oder Schaltungen steuern kann.

Der gesamte Computer ist so ausgelegt, daß die CPU alle Aufgaben übernimmt, die man ihr zumuten kann. Diesem Konzept, das natürlich aus Preisgründen gewählt wurde, ist z. B. eine genormte serielle Schnittstelle (RS232, erst mit Erweiterungs-Box möglich) zum Opfer gefallen, die den Datenaustausch mit Geräten anderer Hersteller ermöglichen würde (der Grund ist die parallele Abfrage der Tastatur). Zusätzlich muß der Mikroprozessor Teile der Video-Interface-Logik übernehmen, und er läuft „nur" mit einer Taktfrequenz von 2 MHz, obwohl 4 MHz beim Z80 möglich wären. Das alles trägt dazu bei, daß das TRS-80-BASIC nicht zu den schnellsten gehört. Trotzdem ergab der Vergleich mit einem 6800-BASIC, daß der TRS-80 Sinusberechnungen etwa um den Faktor 9 schneller ausführt.

Der TRS-80 ist inzwischen zum meistverkauften „Personal Computer" geworden

Auffallend im Befehlssatz sind einerseits sehr leistungsfähige Editierbefehle, die das Arbeiten mit dem Computer wesentlich erleichtern, andererseits Grafikbefehle, mit denen man Einzelpunkte setzen, löschen und abfragen kann. Der Bildschirm faßt im Normalbetrieb 16 Zeilen zu 64 Zeichen, im Grafik-Modus können 128 Punkte horizontal und 48 Punkte vertikal aufgelöst werden; die Position wird in einem X-Y-Koordinatensystem angegeben. Mathematisch orientierte Anwender werden das einem Satz von Sonderzeichen mit geringerer Auflösung vorziehen.

Für den späteren Anschluß des Floppy-Disk-Laufwerkes, zu dem es ein Disketten-Betriebssystem gibt, sind bereits die entsprechenden Befehle vorhanden. In beschränktem Maße lassen sich auch mit dem Kassettenrecorder Dateien aufbauen.

Ein Wort noch zu Problemen, die beim Betrieb auftauchen: Mitunter gibt es Schwierigkeiten mit Fertigkassetten, die man inzwischen in großer Zahl erwerben kann. Das liegt im allgemeinen daran, daß die Kopfstellung des Aufnahmegerätes mit der des eigenen Recorders nicht übereinstimmt. Es emp-