

Interrupt-Uhr mit dem KIM-1

Mikrocomputer werden oft zur automatischen Meßwertverarbeitung herangezogen, und außer der Messung analoger Eingangsgrößen spielt die Zeit dabei eine wichtige Rolle. Versteht man unter „Zeit“ die tatsächliche Uhrzeit, so spricht man auch von „Echtzeit“. Sinnvollerweise beauftragt man den Mikrocomputer selbst damit, die aktuelle Uhrzeit bereitzustellen.

Zur Realisation einer reinen Software-Uhr gibt es mehrere Möglichkeiten. Einmal könnte man eine Verzögerungsschleife so programmieren, daß sie einige Speicherzellen für Stunden, Minuten usw. fortlaufend weiterstellt, so daß

diese die Uhrzeit enthalten. Der Nachteil eines solchen Verfahrens ist so groß, daß es in der Praxis kaum angewendet wird: Der Mikrocomputer ist vollständig damit beschäftigt, die Uhr weiterlaufen zu lassen, und hat z. B. für die Verarbeitung parallel laufender Meßvorgänge keine Zeit mehr.

Die Alternative ist die Ausnutzung des Prozessor-Interrupts. Es ist z. B. möglich, jede volle Sekunde einen Interrupt auszulösen; der Prozessor springt aus dem parallel laufenden Hauptprogramm in eine Interrupt-Routine, die die Uhr um eine Sekunde weiterstellt, und kehrt dann wieder in das Hauptprogramm zurück. Da der relative Zeitbedarf für die Interrupt-Routine minimal ist, kann das Hauptprogramm ungestört z. B. Meßwerte verarbeiten oder mit einem Terminal kommunizieren; es kann aber auch die Uhrzeit auf einem Siebensegment-Display darstellen. Genau dies tut das im Bild aufgelistete 6502-Programm.

Das (KIM-spezifische) Hauptprogramm (03A0..03C0) dient lediglich dazu, den Inhalt der Zellen 00DC, 00DD

und 00DE auf dem sechsstelligen KIM-Display als Stunden, Minuten und Sekunden darzustellen. Der Tag-Zähler (00DB) wird hier nicht angezeigt, ebenso nicht der Viertelsekunden-Zähler (00DF).

Warum eigentlich ein Viertelsekunden-Zähler? Nun, der Mikrocomputer KIM-1 besitzt einen wunderschönen 8-bit-Timer, mit dem man nach einer vorprogrammierten Zeit einen Interrupt auslösen kann (vgl. FUNKSCHAU 1979, Heft 8, Seite 466, „Gebrauchsanleitung für den KIM-Timer“). Er läßt sich maximal aber nur mit einer Zeit von 255 x 1024 µs voreinstellen, so daß der Interrupt nicht jede Sekunde, sondern bereits nach einer Viertelsekunde stattfinden muß. Deswegen fungiert eine Zelle im Zero-Page-Bereich (00DF) als Viertelsekunden-Zähler; ihr Inhalt wird normalerweise nicht weiterverwendet.

Ein ähnliches Programm („Clock“) wurde bereits im „First Book of KIM“ von Charles Parsons veröffentlicht; die Interrupt-Routine umfaßt dabei hexadezimal 5B Bytes. Wendet man dagegen die hervorragenden Möglichkeiten der indizierten Adressierung an, so läßt sich diese Routine auf (hex) 28 Bytes

```
03A0 A9 03 8D FB 17 A9 D8 8D
03A8 FA 17 8D 0E 17 A5 DC 85
03B0 FB A5 DD 85 FA A5 DE 85
03B8 F9 20 1F 1F 20 6A 1F C9
03C0 00 D0 EA 4C 25 19
```

```
03D3 FF TAGE
03D4 24 STUNDEN
03D5 60 MINUTEN
03D6 60 SEKUNDEN
03D7 04 1/4 SEK.
```

```
03D8 48 PHA
03D9 8A TXA
03DA 48 PHA
03DB A2 10 LDX #10
03DD CA DEX
03DE D0 FD BNE 03DD
03E0 A9 F4 LDA #F4
03E2 8D 0F 17 STA 170F
03E5 F8 SED
03E6 A2 05 LDX #05
03E8 CA DEX
03E9 30 10 BMI 03FB
03EB B5 DB LDA DB, X
03ED 18 CLC
03EE 69 01 ADC #01
03F0 DD D3 03 CMP 03D3, X
03F3 90 02 BCC 03F7
03F5 A9 00 LDA #00
03F7 95 DB STA DB, X
03F9 F0 ED BEQ 03E8
03FB D8 CLD
03FC 68 PLA
03FD AA TAX
03FE 68 PLA
03FF 40 RTI
```

```
00DB 04 TAG
00DC 19 STUNDE
00DD 01 MINUTE
00DE 35 SEKUNDE
00DF 02 1/4 SEK.
```

Programm für die Interrupt-Uhr. Das Hauptprogramm springt beim Drücken der KIM-Taste 0 zum Monitor-Programm und erlaubt das Programmieren, während die Uhr weiterläuft. Die „Kaltstartadresse“ ist 03A0; bei laufender Uhr kann man vom Monitorprogramm aus über die Adresse 03AD wieder in das Display-Programm zurück

FUNKSCHAU-Mikro-Poster

6502-Operationscodes

MN.	/IM	/AB	/ZP	/AC	/X	/Y	/ZX	/ZY	/AX	/AY
ADC	69	6D	65	--	61	71	75	--	7D	79
AND	29	2D	25	--	21	31	35	--	3D	39
ASL	--	0E	06	0A	--	--	16	--	1E	--
BIT	--	2C	24	--	--	--	--	--	--	--
CMP	C9	CD	C5	--	C1	D1	D5	--	DD	D9
CPX	E0	EC	E4	--	--	--	--	--	--	--
CPY	C0	CC	C4	--	--	--	--	--	--	--
DEC	--	CE	C6	--	--	--	D6	--	DE	--
EOR	49	4D	45	--	41	51	55	--	5D	59
INC	--	EE	E6	--	--	--	F6	--	FE	--
LDA	A9	AD	A5	--	A1	B1	B5	--	BD	B9
LDX	A2	AE	A6	--	--	--	B6	--	BE	--
LDY	A0	AC	A4	--	--	--	B4	--	BC	--
LSR	--	4E	46	4A	--	--	56	--	5E	--
ORA	09	0D	05	--	01	11	15	--	1D	19
ROL	--	2E	26	2A	--	--	36	--	3E	--
ROR	--	6E	66	6A	--	--	76	--	7E	--
SBC	E9	ED	E5	--	E1	F1	F5	--	FD	F9
STA	--	8D	85	--	81	91	95	--	9D	99
STX	--	8E	86	--	--	--	96	--	--	--
STY	--	8C	84	--	--	--	94	--	--	--

BCC	90	BCS	B0	BEQ	F0	BMI	30
BNE	D0	BPL	10	BRK	00	BVC	50
BVS	70	CLC	18	CLD	D8	CLI	58
CLV	B8	DEX	CA	DEY	88	INX	E8
INY	C8	JMI	6C	JMP	4C	JSR	20
NOP	EA	PHA	48	PHP	08	PLA	68
PLP	28	RTI	40	RTS	60	SEC	38
SED	F8	SEI	78	TAX	AA	TAY	A8
TSX	BA	TXA	8A	TXS	9A	TYA	98

Adressierungsarten: IM = Immediate, AB = Absolut, ZP = Zero Page, AC = Accu, X) = Indirekt X, Y) = Indirekt Y, ZX = Zero Page X, ZY = Zero Page Y, AX = Absolut X, AY = Absolut Y

verkürzen, also um mehr als die Hälfte. Das Weiterzählen der aktuellen Zeit geschieht nicht für jede Zelle (00DB...00DF) einzeln, sondern in einer Schleife, bei der das X-Register dazu dient, Zero-Page-Adressen zu inkrementieren und mit den im Bereich 03D3...03D7 abgelegten Grenzwerten zu vergleichen. So werden etwa die Tage nur dann weitergezählt, wenn bei den Stunden der Grenzwert 24 erreicht ist; der Stundenzähler wird dann gleichzeitig auf Null rückgesetzt.

Der Grenzwert für die Tage ist im Beispielprogramm FF, was dazu führt, daß

– weil diese Zahl im Dezimal-Modus des 6502 nie erreicht wird – bis 99 Tage gezählt werden kann. Selbstverständlich lassen sich hier auch Grenzwerte wie 07 (Wochentag) programmieren.

Es sei erwähnt, daß hier natürlich eine Verbindung zwischen PB 7 und NMI beim KIM-1 notwendig ist, damit der Timer den Interrupt auslösen kann. Er wird dann in der Interrupt-Routine selbst wieder neu gestartet. Da der 1-MHz-Quarz etwas neben der Frequenz liegen kann, ist mit dem Inhalt der Adresse 03DC eine Feinjustierung der Uhr möglich. *Herwig Feichtinger*

Differenz von HL und DE gebildet wird. Nach Rücksprung aus diesem Unterprogramm erfolgt die Frage, ob diese Differenz Null bzw. größer Null ist. Die dadurch bedingten Sprünge erfolgen indirekt, d. h. es wird zum Befehlszähler der Sprungabstand hinzugezählt (positiv bzw. negativ). Durch diese „indirekte Adressierung“ ist es leicht möglich, das Programm in einen anderen Speicherbereich zu schreiben; man muß dann nur die zwei Unterprogrammaufrufe entsprechend ändern (3C50, 3C59).

Der Datenaustausch zwischen dem Akkumulator A und dem Speicherplatz (HL) erfolgt durch Laden von Register C mit dem Wert aus (HL) und Abspeichern von dem Wert aus A in den Platz (HL); danach wird A mit dem Wert aus C geladen.

Die Beendigung des Programms wird durch einen Sprung in das Monitorprogramm des KIT erreicht, wodurch der Inhalt von HL im Adressendisplay angezeigt wird. Bei Verwendung des Sortierprogramms als Unterprogramm muß dieser Sprung in einen Return-Befehl umgeändert werden (3C73).

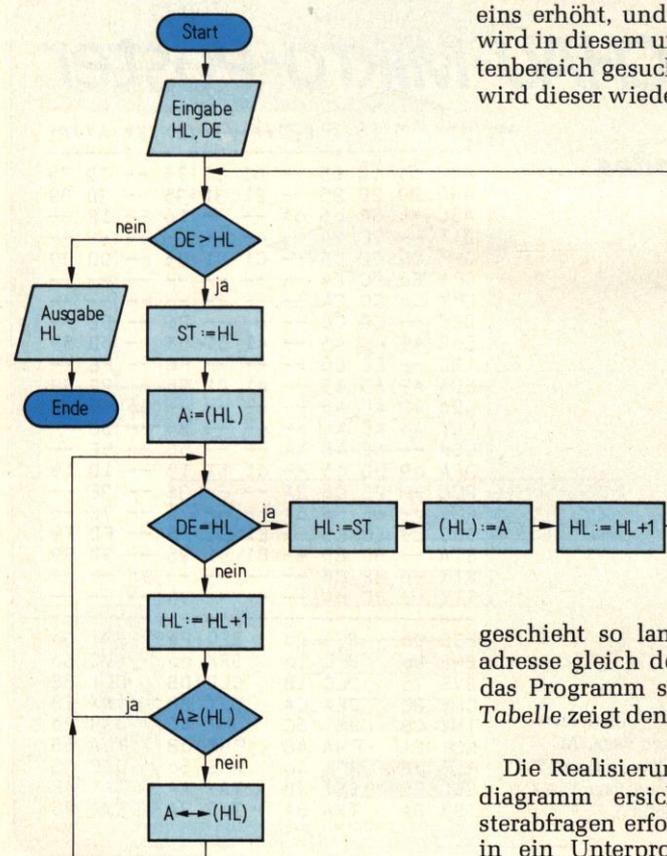
Jörg Ullrich

Sortierprogramm für den Z 80

Das nachstehend beschriebene Programm sortiert einen bis zu 64 KByte großen Datenbereich in absteigender Reihenfolge der Zelleninhalte.

Das Programm wurde auf dem Z-80-KIT realisiert. Die Anfangsadresse muß in das Doppelregister HL und die Endadresse der Daten muß in das Doppelregister DE eingegeben werden.

Nach der Eingabe und nach Drücken der Starttaste prüft das Programm (Bild), ob die Endadresse größer als die Anfangsadresse ist. Wird diese Frage verneint, springt das Programm ins Monitorprogramm des KIT und ist somit beendet. Wird die Frage bejaht, sucht das Programm im Speicherbereich von HL bis DE den größten Wert heraus und speichert ihn auf die Anfangsadresse des Datenbereiches ab. Danach wird die Anfangsadresse um eins erhöht, und ein neuer Größtwert wird in diesem um eins verkürzten Datenbereich gesucht. Ist er gefunden, so wird dieser wieder um eins erhöht. Das



Flußdiagramm des Sortierprogramms. Im Registerpaar HL steht die Anfangsadresse des Datenblockes, in DE die Endadresse; St ist das Stack-Register, A der Akkumulator und unter (HL) verstehen wir den durch HL adressierten Speicherplatz

geschieht so lange, bis die Anfangsadresse gleich der Endadresse ist und das Programm somit beendet ist. Die Tabelle zeigt den Z-80-Maschinencode.

Die Realisierung der aus dem Flußdiagramm ersichtlichen Doppelregisterabfragen erfolgt mit einem Sprung in ein Unterprogramm, in dem die

Tabelle: Maschinencode für das Sortierprogramm

Stack-Pointer	3CFFH
Startadresse	3C50H
3C50	CD 6C 3C
3C53	28 1E
3C55	30 1C
3C57	E5
3C58	7E
3C59	CD 6C 3C
3C5C	28 09
3C5E	23
3C5F	BE
3C60	30 F7
3C62	4E
3C63	77
3C64	79
3C65	18 F2
3C67	E1
3C68	77
3C69	23
3C6A	18 E4
3C6C	E5
3C6D	37
3C6E	3F
3C6F	ED 52
3C71	E1
3C72	C9
3C73	C3 0D 01