Markus Schobesberger

Software-Single-Step

Am Beispiel des CT-65 von Thaler wird hier gezeigt, wie man bei 6502-Computern mit Hilfe eines Interrupt-Timers rein softwaremäßig eine Einzelschritt-Funktion implementieren kann, die z. B. zum Test von Maschinensprache-Programmen recht nützlich ist.

Das Programm in *Bild 1* ist in seiner Adressenbelegung für den CT-65 ausgelegt (siehe mc 4/1983). Beim Start an der Adresse 0340 wird der Timer 2 des VIA-Bausteins 6522 so programmiert, daß während der Ausführung des ersten Befehls im zu testenden Programm ein Interrupt erzeugt wird. In der Interrupt-Routine können nun die Inhalte der internen CPU-Register auf dem Siebensegment-Display des CT-65 angezeigt werden. Vor der Rückkehr ins zu testende Programm, um den nächsten Befehl auszuführen, werden der Stack und die CPU-Register restauriert und der Timer neu gestartet.

So wird gestartet

Vor dem Programmstart muß die Anfangsadresse des zu testenden Programms in die Zero-Page-Adressen 0067 (LSB) und 0068 (MSB) eingeschrieben werden. Dann erfolgt der Start mit der Taste G an der Adresse 0340, wonach die Vektoren für IRQ und BRK (00C3/00C5 und 00CC/00CE) automatisch gesetzt werden. Nach dem Start des zu testenden Programms und Ausführung des ersten Befehls werden auf dem Display die Registerinhalte X, Y und A angezeigt. Mit der Tastatur kann die Anzeige auch umgeschaltet werden: Taste B bewirkt die Anzeige des Programmzählers und des nächsten Befehls (Format PCH, PCL, Operationscode).

Trifft man auf einen BRK-Befehl, so ist zu beachten, daß die NMOS-CPU Break als Zwei-Byte-Befehl interpretiert und somit der Befehlszähler nicht die nächste, sondern die übernächste Adresse anzeigt.

Mit der Taste F werden die Inhalte des Statusregisters und des Stackpointers angezeigt (Format: Status, 00, Stackpointer).

Die Hex-Taste A bringt erneut die CPU-Register X, Y, A zur Anzeige. Die Enter-Taste (E) dient als Step-Taste: Der nächste Befehl wird ausgeführt, anschließend wird das Programm wieder angehalten und die Inhalte der Register X, Y, A erscheinen im Display. Wurde das Programm durch einen Break-Befehl aufgerufen, so erfolgt bei Betätigung der Enter-Taste ein Sprung zum Monitorprogramm des CT-65 (Anzeige: 0200 XX). Der Single-Step-Modus kann nur durch Drücken der Reset-Taste oder durch Ausführung eines Break-Befehles beendet werden.

Breakpoint-Bedienung

Soll das Programm lediglich zur Behandlung von Breakpoints eingesetzt werden, so müssen die Zellen für den Break-Vektor (BRKU) vor dem Start geladen werden, damit bei einem BRK-Befehl die Anzeigeroutine gestartet wird (zu ändernde Bytes: 00CC = 00, 00CD = 70, 00CE = 03).
Wenn dabei auf den Single-Step verzichtet wird, kann der Programmteil im Adreßbereich 0340...036F weggelassen werden. Bei der Fehlersuche mit Hilfe sogenannter „Breakpoints" wird an der Stelle, an der das Programm gestoppt werden soll, ein Befehl durch BRK (Code 00) ersetzt. Das zu testende Programm wird mit der Taste G gestartet, und bei Erreichen des Break-Befehls erfolgt ein Sprung zur Break-Routine, wo die Registerinhalte und der Programmstand angezeigt werden.

```
0340 lda #00
0342 sta   c3
0344 sta   cc
0346 lda #70
0348 sta   c4
034a sta   cd
034c lda #03
034e sta   c5
0350 sta   ce
0352 jsr f809
0355 sei
0356 lda #40
0358 sta a00b
035b lda #e0
035d sta a00e
0360 lda #08
0362 sta a008
0365 lda #00
0367 sta a009
036a lda   ad
036c cli
036d jmp(0067)
0370 jsr f809
0373 lda a008
0376 jsr f80f
0379 sta   63
037b pla
037c sta   61
037e pla
037f sta   6a
0381 pla
0382 sta   6b
0384 sty   64
0386 stx   65
0388 tsx
0389 stx   62
038b cli
038c nop
038d nop
038e nop
038f lda   65
0391 sta   a1
0393 lda   64
0395 sta   a0
0397 lda   63
0399 sta   a2
039b jsr f800
039e cmp #68
03a0 beq 03d0
03a2 cmp #41
03a4 beq 03c1
03a6 cmp #42
03a8 beq 03b0
03aa cmp #32
03ac beq 038f
03ae bne 039b
03b0 lda   6b
03b2 sta   a1
03b4 lda   6a
03b6 sta   a0
03b8 ldy #00
03ba lda  (6a),y
03bc sta   a2
03be clc
03bf bcc 039b
03c1 lda   61
03c3 sta   a1
03c5 lda #00
03c7 sta   a0
03c9 lda   62
03cb sta   a2
03cd clc
03ce bcc 039b
03d0 lda #10
03d2 and   61
03d4 beq 03d9
03d6 jmp f818
03d9 lda   6b
03db pha
03dc lda   6a
03de pha
03df lda   61
03e1 pha
03e2 lda #40
03e4 sta a00b
03e7 lda #e0
03e9 sta a00e
03ec lda #0d
03ee sta a008
03f1 lda #00
03f3 sta a009
03f6 lda   63
03f8 ldx   65
03fa ldy   64
03fc rti
```

Bild 1. Disassembler-Listing des Single-Step-Programms

```
0340 a9 00 85 c3 85 cc a9 70 85 c4 85 cd a9 03 85 c5
0350 85 ce 20 09 f8 78 a9 40 8d 0b a0 a9 e0 8d 0e a0
0360 a9 08 8d 08 a0 a9 00 8d 09 a0 a5 ad 58 6c 67 00
0370 20 09 f8 ad 08 a0 20 1f f8 85 63 68 85 61 68 85
0380 6a 68 85 6b 84 64 86 65 ba 86 62 58 ea ea ea a5
0390 65 85 a1 a5 64 85 a0 a5 63 85 a2 20 00 f8 c9 68
03a0 f0 2e c9 41 f0 1b c9 42 f0 06 c9 32 f0 e1 d0 eb
03b0 a5 6b 85 a1 a5 6a 85 a0 a0 00 b1 6a 85 a2 18 90
03c0 da a5 61 85 a1 a9 00 85 a0 a5 62 85 a2 18 90 cb
03d0 a9 10 25 61 f0 03 4c 18 f8 a5 6b 48 a5 6a 48 a5
03e0 61 48 a9 40 8d 0b a0 a9 e0 8d 0e a0 a9 0d 8d 08
03f0 a0 a9 00 8d 09 a0 a5 63 a6 65 a4 64 40
```

Bild 2. Hex-Dump zum Eintippen

Die Funktionsweise

Und wie funktioniert das Programm? Nach dem Start wird der Timer 2 des systeminternen Ein-/Ausgabebausteines (6522-VIA) gestartet, um nach 9 Taktzyklen einen Interrupt zu erzeugen.

Nachdem der Akku aus der dafür vorgesehenen Speicherzelle geladen wurde, erfolgt ein indirekter Sprung zur Startadresse des zu testenden Programmes.

Die Startadresse der Interrupt-Routine lautet 0370. Zuerst werden die Registerinhalte und die Inhalte des Programmzählers, des Stackpointers sowie des Statusregisters gerettet. Die Registerinhalte von A, X, Y werden in die Display-Zellen geschrieben und die Tastatur abgefragt. Je nach gedrückter Taste wird die Anzeige geändert oder bei gedrückter Enter-Taste die Interrupt-Routine verlassen. Um festzustellen, ob die Routine durch einen „normalen" Interrupt (vom Timer) oder durch einen Break-Befehl aufgerufen wurde, wird das Break-Flag (es steht in der Speicherzelle 0061) getestet: Bei B = 1 erfolgt ein Sprung zum Monitor, bei B =0 wird die Interrupt-Routine mit RTI beendet. Vor der Rückkehr in das unterbrochene Programm wird der Timer neu gestartet.

Bei Verlegung des Programms in einen anderen Speicherbereich muß lediglich die Startadresse der Interruptroutine angepaßt werden. Diese Adresse steht in den Speicherplätzen 0347 (LSB) und 034D (MSB). Ansonsten ist dieses Programm in jedem Speicherbereich lauffähig, da nur relative Sprünge verwendet werden.

Für alle, die ein anderes System benutzen, sei noch erwähnt, welche Bedeutung die vorkommenden Systemadressen haben:

Die Zeropageadressen A0, A1 und A2 dienen beim CT-65 als Displayspeicher. Der Inhalt der Adresse 00AD wird beim Start mit G in den Akku geladen, um einen Programmstart mit definiertem Registerinhalt zu ermöglichen. Die Monitor-Unterprogramme haben folgende Bedeutung:

Adresse: Funktion:

F800 Warten auf Tastendruck, Tastencode in Akku holen
F809 Einschalten des Displays
F80F Holt die Register A,X,Y vom Stack; die Register werden bei einem Interrupt vom Monitorprogramm auf den Stack gerettet.

Bild 2 zeigt den Hex-Dump des Programms zum Eintippen in den CT-65.

Interpreter-Schnittstellen beim TRS-80

Wer als Z80-Assembler-Programmierer dem Basic-Interpreter seines TRS-80, Video-Genie oder Colour-Genie Neues beibringen will, der muß auf die etwas mühsame Suche nach Software-Schnittstellen gehen.
Unabdingbare Voraussetzung hierfür ist ein möglichst mit guten Kommentaren versehenes ROM-Listing.
Doch auch dort wird meist verschwiegen, daß an verschiedenen Stellen des Interpreters bereits eine Erweiterung vorgesehen wurde, die es dem Disk-Basic ermöglicht, zusätzliche Funktionen einzufügen.
Deshalb existiert im RAM eine Liste von Sprungbefehlen, die für den diskettenlosen Betrieb jeweils durch einen Return-Befehl ersetzt sind. Durch Überschreiben mit einem Sprung zu einer Benutzerroutine kann man sich in die normale Bearbeitung des Basic-Interpreters „einhängen". Das *Bild* zeigt die Belegung dieser Sprungleiste.

1. Spalte: Lage des jeweiligen Sprungvektors im RAM. Ab der angegebenen Adresse sind drei Bytes für den Eintrag eines Sprungbefehls reserviert.
2. Spalte: Adresse innerhalb des Basic-Interpreters, von der aus die Schnittstelle aufgerufen wird.
3. Spalte: Funktion des aufrufenden Moduls in Stichworten.

Werner Wagner

Vektoradresse			Aufrufadresse			Funktion Aufrufmodul
41A6 H	16806	D	19EC H	6636	D	Klartext-Fehlermeldung
41A9 H	16809	D	27FE H	10238	D	USR Anfang
41AC H	16812	D	1A1C H	6684	D	Hauptschleife Beginn
41AF H	16815	D	0368 H	872	D	Tastatur inline
41B2 H	16818	D	1AA1 H	6817	D	Hauptschleife
41B5 H	16821	D	1AEC H	6892	D	Hauptschleife Ende
41B8 H	16824	D	1AF2 H	6898	D	Hauptschleife Ende
41BB H	16827	D	1B8C H	7052	D	NEW
41BB H	16827	D	1DB0 H	7600	D	END
41BE H	16830	D	2174 H	8564	D	PRINT Ende
41C1 H	16833	D	032C H	812	D	Zeichen ausgeben
41C4 H	16836	D	0358 H	856	D	Tastaturabfrage
41C7 H	16839	D	1EA6 H	7846	D	RUN
41CA H	16842	D	206F H	8303	D	PRINT screen Anfang
41CD H	16845	D	20C6 H	8390	D	PRINT
41D0 H	16848	D	2103 H	8451	D	CR ausgeben
41D3 H	16851	D	2108 H	8456	D	PRINT Komma auswerten
41D3 H	16851	D	2141 H	8513	D	PRINT Tab auswerten
41D6 H	16854	D	219E H	8606	D	INPUT
41D9 H	16857	D	2AEC H	10988	D	MID$ auf linker Seite
41DC H	16860	D	222D H	8749	D	INPUT decodieren
41DF H	16863	D	2278 H	8824	D	INPUT Ende
41DF H	16863	D	2B44 H	11076	D	LIST
41E2 H	16866	D	02B2 H	690	D	SYSTEM

Die Interpreter-Sprungleiste des TRS-80, nach Vektoradresse sortiert