

```

340 DATA 233,1,76,195,193,24,169,48,109,56,195,
    141,56,195,169,48,109,57,195,141 <139>
350 DATA 57,195,169,48,109,58,195,141,58,195,96,
    32,253,174,32,138,173,32,247,183 <201>
360 DATA 96,133,99,134,98,162,144,56,32,73,188,
    32,223,189,32,135,180,32,166,182 <144>
370 DATA 96,162,0,165,34,141,20,194,165,35,141,
    21,194,189,0,1,145,253,200,232,206 <202>
380 DATA 64,195,208,244,96,32,86,194,169,120,
    160,194,32,30,171,76,154,193,32,86 <161>
390 DATA 194,169,155,160,194,32,30,171,76,154,
    193,169,192,160,194,32,30,171,76,154 <058>
400 DATA 193,169,229,160,194,32,30,171,76,154,
    193,169,11,160,195,32,30,171,96,173 <015>
410 DATA 61,195,133,253,173,62,195,133,254,160,
    0,152,145,253,200,145,253,24,173 <155>
420 DATA 61,195,105,2,133,45,173,62,195,105,0,
    133,46,96,18,90,69,73,76,69,78,78 <207>
430 DATA 85,77,77,69,82,32,71,82,79,69,83,83,69,
    82,32,65,76,83,32,54,51,57,57,57 <045>
440 DATA 33,146,0,18,70,82,69,73,69,82,32,83,80,
    69,73,67,72,69,82,80,76,65,84,90 <034>
450 DATA 32,82,69,73,67,72,84,32,78,73,67,72,84,
    33,146,0,18,48,32,65,76,83,32,83 <034>
460 DATA 67,72,82,73,84,84,87,69,73,84,69,32,78,
    73,67,72,84,32,77,79,69,71,76,73 <090>
470 DATA 67,72,33,146,0,18,65,78,70,65,78,71,83,
    65,68,82,69,83,83,69,32,62,32,69 <068>
480 DATA 78,68,65,68,82,69,83,83,69,46,70,69,72,
    76,69,82,33,146,0,18,42,32,68,65 <089>
490 DATA 84,65,32,69,82,90,69,85,71,69,82,32,70,
    69,82,84,73,71,33,32,40,67,41,32 <070>
500 DATA 74,46,77,65,84,69,82,78,65,46,32,42,
    146,0,66,195,49,52,50,244,1,1,8,10 <018>
510 DATA 48,0,0 <166>

```

Single-Step für VC 20

»Single-Step« soll bei Maschinenprogrammen zum Aufdecken von Fehlern beitragen. Anfängern erlaubt es ein schnelleres Einfühlen in die Wirkungsweise der Maschinenbefehle.

Das Programm ist für den Commodore VC 20 mit mindestens 8 KByte Speichererweiterung geschrieben (ein entsprechendes Programm für den C64 finden Sie im 64'er Stammmagazin). Es ist ein DATA-Listing einer Maschinenspracheroutine. Über Zeile 9 des Listings läßt sich die gewünschte Startadresse festlegen; die Basic-Schleife ab Zeile 300 korrigiert absolute Sprungadressen. Hierzu wurden bei der Programmerstellung die absoluten Zieladressen so eingestzt, als stünde das Programm ab Adresse 0 im Speicher. Der Basic-Teil sorgt dafür, daß die Startadresse (ss) als Offset berücksichtigt wird.

Nach einmaligem Starten läßt sich die Maschinenspracheroutine mit SYS ss starten, der Basic-Teil kann demnach gelöscht werden, sollte aber sicherheitshalber vorher abgespeichert werden.

Für die Bedienung des Programmes stehen die Funktionstasten F1, F3 und F5 zur Verfügung. Ein Beispiel soll den Ablauf verdeutlichen (Tabelle):

Zunächst startet man »Single-Step« mit SYS ss, es erscheint dann die Überschrift mit Bezeichnung der einzelnen Bild-

schirmspalten und der ersten, übersetzten Zeile. Nun kann man über F3 in den Editier-Modus gelangen, es lassen sich sämtliche dargestellte Register und Flags ändern. Dazu fährt man mit dem Cursor einfach an die entsprechende Stelle und tippt die gewünschten Werte ein. Im Editier-Modus wird nun der PC (Programm-Counter) so geändert, daß er auf die Startadresse der eigenen Routine zeigt (hier im Beispiel ist dies die Adresse \$ 4000). Durch Betätigen der Taste F1 wird der hier abgelegte Befehl ausgeführt. Der PC zeigt dann auf die Startadresse des nächsten Befehles, PSW (Programm Status Word, auch Flag-Register genannt), A, X und Y sind entsprechend verändert. (Im Beispiel wurde der Akku mit den Daten #00 geladen). Jede weitere Betätigung der Funktionstaste F1 bringt einen weiteren Befehl zur Ausführung.

Jetzt noch eine Anmerkung zu Manipulationen im PSW (Flag-Register): Durch Ändern einzelner Flags lassen sich zum Beispiel Schleifen vorzeitig beenden oder, wie im Beispiel in Zeile 16 geschehen, auch verlängern; das Zero-Bit wurde zurückgesetzt, was die CPU durch einen weiteren relativen Sprung auf Adresse \$ 4000 quittieren mußte.

Weiterhin lassen sich unter Benutzung des PSW, genauer des I-Flags, Breakpoints im Programm setzen. Es ist nicht immer nötig das Programm schrittweise von Anfang bis Ende zu untersuchen. In solchen Fällen setzt man zu Beginn des inter-

** Beispiel **

\$4000	lda #0	a900	;sum gleich null
\$4002	tax	aa	;index gleich null
\$4003	sed	f8	;dezimale addition
\$4004	sum clc	18	;übertrag nicht berücksichtigen
\$4005	adc \$2,x	7502	;sum = sum+daten
\$4007	inx	e8	
\$4008	cpx \$0	e400	;alle elemente ?
\$400a	bne sum	d0f8	;nein, dann weiter
\$400c	sta \$1	8501	;lege Ergebnis ab
\$400e	rts	60	
\$0000	hex	02	;anzahl elemente
\$0001		00	;platz für summe reservieren
\$0002		793327	;zu add. elemente in packed bcd

Ablauf mit Single-Step:

```

00 : ** single — step ** :
01 : p c nv-bdizc a x y :
02 :
03 :e14e-00-00011-00-00-2c: ;ändern in —
04 :4000- - ff - ff - ;next step führt zu —
05 :4002-00-00011-00- ff -2c:
06 :4003-00-00011-00-00-2c: ;next step = set decimal flag —
07 :4004-00-01011-00-00-2c: ;clear carry —
08 :4005-00-01010-00-00-2c:
09 :4007-00-01000-79-00-2c: ;inkrementiere x-reg. —
10 :4008-00-01000-79-01-2c: ;vergleiche, setze n, z, c —
11 :400a-10-01000-79-01-2c: ;nicht null also sprung —
12 :4004-10-01000-79-01-2c:
13 :4005-10-01000-79-01-2c: ;addiere, setze n,v,z,c —
14 :4007-11-01001-12-01-2c: ;inkrementiere x-reg. —
15 :4008-01-01001-12-02-2c: ;vergleiche, setze n,z,c —
16 :400a-01-01011-12-02-2c: ;null also kein sprung —
17 :400c-01-01011-12-02-2c: ;ergebnis ablegen ...

```

mit Änderung:

```

16 :400a-01-01011-12-02-2c: ;ändern —
16' :400a- 0 ;erzwingt sprung —
17' :4004-01-01001-12-02-2c:
18' :4005-01-01000-12-02-2c: ;addiere —
19' :4007-00-01000-39-02-2c: ;inkrementiere x-reg. —
20' :4008-00-01000-39-03-2c:

```

Beispielprogramm und Ablauf mit »Single Step«

essierenden Teiles den Befehl CLI ein. Es genügt nun im Editiermodus das I-Flag zu setzen und mit F1 den nächsten Schritt zu starten. Das Programm wird erst wieder nach Abarbeiten des CLI- und des darauffolgenden Befehles angehalten.

Die Taste F5 dient zum Ausstieg aus der »Single-Step«-Routine. Ein Betätigen dieser Taste hat die gleiche Wirkung wie Stop/Restore, das heißt es wird eine teilweise Neuinitialisierung und ein Sprung zum Basic-Warmstart durchgeführt.

Abschließend noch eine Anmerkung zur Wirkungsweise:

Ich benutze hier den Timer 1 des VIA 6522 (versatile interface adapter) in der Betriebsart »One-Shot«. Diese läßt sich über das ACR (auxiliary control register) definieren. Der Timer erzeugt jetzt nach jedem Start einen einzelnen Interrupt. Durch geeignete Wahl des Timer-Zählerstandes kann erreicht werden, daß nach jedem Rücksprung aus der Interruptroutine mittels RTI jeweils ein Befehl des zu untersuchenden Programmes ausgeführt wird und darauffolgend die weitere Ablaufsteuerung wieder von der Interruptroutine übernommen wird. Im ersten Hauptteil der Routine »Single-Step« hole ich dann die interessierenden Daten vom Stack und stelle sie auf dem Bildschirm dar. Vor dem Rücksprung mittels RTI wird dann im zweiten Teil die eventuell geänderte Zeile wieder eingelesen und übersetzt.

(Hermann Weißenberger / ev)

```

0 REM ** HERMANN WEISSENBERGER **          <021>
1 REM TEL 09722/2672                        <121>
2 :                                          <060>
3 REM ::::::::::::::::::::::::::::          <002>
4 REM : ** SINGLE - STEP **                <218>
5 REM : PC NV-BDIZC A X Y :                <202>
6 REM ::::::::::::::::::::::::::::          <005>
7 :                                          <065>
8 REM STARTADRESSE --> SS                  <106>
9 :   SS=24576: REM $6000                  <086>
10 :                                        <068>
11 REM ROUTINE VERSCHIEBBAR UEBER SS       <073>
12 :                                        <070>
13 POKE 56,INT(SS/256+.001)                 <154>
14 POKE 55,INT(SS-256*PEEK(56)+.001)       <194>
15 :                                        <073>
16 REM *****                          <059>
17 REM DATALISTE DER ROUTINE (438 BYTE)    <096>
18 :                                        <076>
20 FOR I=SS TO SS+437: READ W: POKE I,W: NEXT <057>
22 DATA 160,0,185,49,0,32,210,255,200,192 <233>
24 DATA 44,208,245,120,173,43,145,41,63,141 <087>
26 DATA 43,145,169,93,141,20,3,169,0,141 <199>
28 DATA 21,3,169,128,141,138,2,169,20,141 <248>
30 DATA 36,145,169,0,141,37,145,88,96,147 <021>
32 DATA 32,42,42,32,83,73,78,71,76,69      <045>
34 DATA 32,45,32,83,84,69,80,32,42,42     <067>
36 DATA 13,80,67,32,32,32,78,86,45,66     <077>
38 DATA 68,73,90,67,32,65,32,32,88,32     <082>
40 DATA 32,89,13,173,36,145,169,0,72,40   <172>
42 DATA 32,46,1,169,13,32,210,255,160,21 <203>
44 DATA 169,45,32,210,255,136,16,248,186,160 <173>
46 DATA 0,32,57,1,32,57,1,200,32,85       <213>
48 DATA 1,200,32,57,1,200,32,57,1,200     <042>
50 DATA 32,57,1,32,159,255,32,228,255,201 <019>
52 DATA 135,240,44,201,134,240,44,201,133,208 <199>
54 DATA 238,186,160,0,32,112,1,32,112,1   <156>
56 DATA 200,32,148,1,200,32,112,1,200,32 <191>
58 DATA 112,1,200,32,112,1,169,0,141,37   <154>
60 DATA 145,104,168,104,170,104,64,120,76,210 <216>
62 DATA 254,169,145,32,210,255,160,0,177,209 <183>
64 DATA 9,128,145,209,152,72,32,46,1,32   <190>
66 DATA 159,255,32,228,255,201,0,240,246,170 <182>
68 DATA 104,168,177,209,41,127,145,209,224,133 <036>

```

```

70 DATA 240,175,224,135,240,207,224,29,240,28 <235>
72 DATA 224,157,240,37,224,48,144,206,224,71 <193>
74 DATA 176,202,224,58,144,4,224,65,144,194 <151>
76 DATA 177,209,41,127,201,45,240,186,138,32 <198>
78 DATA 210,255,200,192,22,144,177,169,157,208 <045>
80 DATA 5,192,0,240,169,138,32,210,255,136 <096>
82 DATA 16,162,160,120,162,221,202,208,253,136 <027>
84 DATA 208,248,96,189,6,1,74,74,74,74   <191>
86 DATA 32,73,1,189,6,1,41,15,202,201    <096>
88 DATA 10,144,2,233,57,105,48,145,209,200 <096>
90 DATA 96,189,6,1,72,104,10,72,169,48   <181>
92 DATA 144,2,169,49,192,7,240,2,145,209 <019>
94 DATA 200,192,13,144,236,104,202,96,177,209 <005>
96 DATA 32,139,1,10,10,10,157,6,1        <042>
98 DATA 200,177,209,32,139,1,29,6,1,157  <223>
100 DATA 6,1,200,202,96,201,48,176,2,105  <212>
102 DATA 9,41,15,96,189,6,1,41,32,157    <088>
104 DATA 6,1,169,128,72,177,209,201,49,208 <086>
106 DATA 8,104,72,29,6,1,157,6,1,200     <023>
108 DATA 104,74,72,208,236,104,202,96    <089>
200 :                                       <002>
205 REM *****                          <248>
210 REM RELATIVE PLAETZE DER              <173>
215 REM ABSOLUTEN SPRUNGADRESSEN          <071>
220 :                                       <022>
225 DATA 3,101,122,125,129,133,137,141,165 <182>
230 DATA 168,172,176,180,184,217,321,371,384 <064>
235 DATA 23,28: REM 'INTERRUPT-VEKTOR'   <085>
240 :                                       <042>
245 REM *****                          <032>
250 :                                       <052>
300 REM ABSOLUTE SPRUNGADRESSEN KORRIGIEREN <144>
305 :                                       <108>
310 FOR I=0 TO 17                          <239>
315 :   READ A                               <062>
320 :   X1=PEEK(SS+A)+PEEK(SS+A+1)*256+SS   <028>
325 :   X2=INT(X1/256+.001)                 <254>
330 :   X1=INT(X1-256*X2+.001)              <054>
335 :   POKE A+SS,X1                         <103>
340 :   POKE A+SS+1,X2                       <072>
345 NEXT I                                   <037>
350 :                                       <153>
355 READ A1: X1=PEEK(SS+A1)                 <167>
360 READ A2: X1=X1+PEEK(A2)*256+SS         <042>
365 X2=INT(X1/256+.001)                     <236>
370 X1=INT(X1-256*X2+.001)                 <036>
375 POKE A1+SS,X1                           <134>
380 POKE A2+SS,X2                           <141>
385 :                                       <188>
390 REM *****                          <178>
395 :                                       <198>
400 REM BELEGUNG DER FUNKTIONSTASTEN      <212>
405 :                                       <089>
410 REM F1 = NEXT STEP                     <208>
415 REM F3 = EDIT-MODE                     <093>
420 :   REM AKZEPTIERT WERDEN TASTEN 0...F UND <236>
425 :   REM CURSOR-RIGHT,CURSOR-LEFT       <094>
430 REM F5 = AUSSTIEG                       <091>
435 :                                       <238>
500 PRINT                                    <142>
505 PRINT"[SPACE2]START[SPACE]MIT:"        <137>
510 PRINT"SYS";SS                           <188>
515 END                                       <133>

```