

Cross-Ref 64 — Basic-Programme unter der Lupe

Wenn man zu einem Basic-Programm eine Liste aller Sprungadressen, Schleifen und Variablen hat, ist das eine sehr gute Hilfe. Nicht nur während des Programmierens, sondern auch zur nachträglichen Dokumentation. Eine solche Liste erstellt dieses Maschinen-Programm.

Cross-Ref 64 durchsucht in zwei Durchgängen (Passes) ein Basic-Programm nach Sprungbefehlen und Variablen. Sämtliche Sprungziele und Variablennamen werden dann auf einem Drucker in Form einer Tabelle ausgegeben. Eine solche Tabelle nennt man Cross-Reference-List.

Das Basic-Programm kann dazu übrigens irgendwo im Basic-Speicher liegen. Es muß nicht bei Adresse \$0801 beginnen. Das ist wichtig, denn häufig wird der untere Basic-Speicherteil für Sprites reserviert.

1. Durchgang

Im ersten Durchlauf (Pass) werden Sprunganweisungen und FOR-NEXT-Schleifen gesucht und alle Zeilennummern der Zeilen ausgegeben, die eine Sprunganweisung enthalten oder angesprungen werden. Dabei werden alle Sprünge berücksichtigt, die das C 64-Basic 2.0 bietet. FOR-NEXT-Schleifen werden durch die Laufvariable gekennzeichnet. Eventuell vorhandene, noch nicht zu Ende geschriebene FOR-NEXT-Schleifen werden mit einem »*« markiert.

2. Durchgang

Hier werden alle Variablen eines Basic-Programms gesucht und ausgegeben.

Die Ausgabe der Cross-Reference-Liste kann in beiden Durchgängen entweder auf Bildschirm oder Drucker erfolgen. Bei Bildschirmausgabe allerdings auch im Druckerformat mit 80 Zeichen/Zeile. Das bedeutet, daß zwei Bildschirmzeilen immer eine Druckerzeile darstellen. Doch wie sinnvoll ist überhaupt eine Dokumentation auf dem Bildschirm? Das Programm ist auf einen Epson-Drucker mit Görlitz-Interface angepaßt. Es kann aber leicht so verändert werden, daß es auch für andere

Drucker geeignet ist. Es müssen nur vier Speicherzellen mit POKE-Befehlen verändert werden.

Einstellung der Geräteadresse:

POKE 50517, GA : POKE 50519, GA

Einstellung der Sekundäradresse:

POKE 50521, SA+96 : POKE 50529, SA+96

Mit einem Monitor wie dem SMON oder dem Einzeler »Maschinenprogramme speichern« (Ausgabe 11/84) können Sie die auf Ihren Drucker angepaßte Programmversion speichern:

SYS57812a\$,x:POKE193,1s:POKE194,hs:POKE174,le:POKE175,he:SYS62957

Dadurch ersparen Sie sich die lästigen POKE-Befehle nach jedem Laden von Cross-Ref 64. Das Programm belegt den Speicherbereich von \$C000 (49152) bis \$CAB6 (51894). Um Ihnen eventuelle Berechnungen zu ersparen: Das Highbyte (hs) von 49152 ist 192 das Lowbyte (ls) 0. Das Highbyte von 51894 (he) ist 202, das Lowbyte (le) 182.

Übersichtliches Ausgabeformat

Wie schon erwähnt, werden alle Zeilen, in denen ein Sprung erfolgt, ausgegeben. Die gezeigte Tabelle (Tabelle 1) bezieht sich auf das Beispielprogramm (Listing 1). Man kann sich auch eine Liste aller Sprungziele drucken lassen (Tabelle 2). Die Ausgabe erfolgt mit 80 Zeichen pro Zeile, dem üblichen Druckerformat.

Sollten offene FOR-NEXT-Schleifen (ohne NEXT) vorhanden sein, wird nach einem »*« die Anzahl der fehlenden NEXT-Befehle angezeigt. Sollte die Anzahl 255 betragen, heißt das, daß ein NEXT zuviel gefunden wurde.

Sollten in einer Basic-Zeile mehrere Sprünge auftreten, wird die mehrmalige Ausgabe der Zeilennummer unterdrückt.

Bei Sprungbefehlen der Form ON X GOSUB wird das Argument nach ON durch »—« ersetzt, da auch ganze Basic-Funktionen als Argument erlaubt sind. Um diese Funktionen zu berücksichtigen, wäre ein zu großer Programmieraufwand nötig. Treten nach einem THEN mehrere Sprünge auf, wird jedem Sprungziel ein THEN vorangestellt. Eine solche Befehlsfolge ist beispielsweise

IF X=1 THEN GOSUB 100:GOSUB 200:GOSUB 300.

Bei FOR-NEXT-Schleifen wird der ganze Variablenname angegeben.

Die Variablennamen werden in einer eigenen Tabelle ausgegeben (Tabelle 3). Hier werden die Namen allerdings auf zwei Zeichen gekürzt. In der Tabelle steht nach dem Namen der Variable der Variablentyp (Real, Integer oder String). Zwei Klammern »()« zeigen an, daß es sich um eine indizierte Variable handelt. Da eine Variable in einem Programmbereich sehr häufig benutzt werden kann, werden bei Mehrfachnennung einer Variablen nur noch die Zeilennummern angegeben; so lange, bis eine andere Variable aufgerufen wird. Wird einer Variablen ein anderer Wert zugeordnet (A=A+1), wird die Zeilennummer mit einem Stern »*« gekennzeichnet. Bei der sortierten

SPRUNGBEFEHLE					
=====					
IN ZEILE	SPRUNG DURCH	NACH	IN ZEILE	SPRUNG DURCH	NACH
=====					
20	GOSUB	100		GOTO	200
40	RUN	50	60	THEN GOSUB	100
	THEN GOTO	50	80	ON - GOTO	10,20,
		30,40			
100	THEN ON - GOSUB	100,90,70,80			

Tabelle 1. Liste der Zeilen, die Sprungbefehle enthalten

Variablenausgabe (Tabelle 4), werden nach dem Variablennamen alle Zeilen aufgeschlüsselt, in denen die Variable vorkommt. Das Sortieren geschieht nach den folgenden Kriterien:

1. Typ: Real, Integer, Strings, Felder
2. Name alphabetisch
3. Zeilennummern in aufsteigender Reihenfolge.

Die Ausgabe kann jederzeit durch Tastendruck gestoppt werden. Eine begonnene Druckzeile wird zu Ende gedruckt. Bei einem weiteren Tastendruck fährt das Programm fort. Mit RUN/STOP kann die Ausgabe abgebrochen werden.

Tips zu Eingabe und Benutzung

Das Maschinen-Programm (Listing 2) sollten Sie mit dem MSE eingeben. Es wird durch »SYS 49152, Parameterliste, Bereich« gestartet. Die »Parameterliste« muß aus genau fünf Zeichen bestehen (Leerzeichen nicht mitgerechnet). Folgende Zeichen sind erlaubt:

1. Stelle: P — Ausgabe auf Drucker
2. Stelle: S — Ausgabe der Zeilen mit Sprungbefehlen (Tabelle 1)
3. Stelle: J — Ausgabe der angesprungenen Zeilen (nur zusammen mit »S« an zweiter Stelle möglich (Tabelle 2))
4. Stelle: V — Ausgabe der benutzten Variablen in der Reihen-

folge des Auftretens (Tabelle 3)

5. Stelle: J — Ausgabe der benutzten Variablen in sortierter Reihenfolge (Tabelle 4)

Werden andere Zeichen an der entsprechenden Stelle angegeben, dann wird der Programmpunkt nicht ausgeführt.

Die Bereichsangabe ist optional und hat die gleiche Syntax wie beim LIST-Befehl. Falls ein Zeilenbereich angegeben wird, analysiert das Programm auch nur den entsprechenden Abschnitt. Ein Beispiel:

Es sollen alle Sprungbefehle und alle Variablen im Zeilenbereich 100-200 gesucht werden. Die Ausgabe soll auf dem Drucker erfolgen. Der entsprechende SYS-Aufruf lautet dann: SYS 49152, PS.,100-200

Anstelle der Punkte kann dabei jedes beliebige andere Zeichen mit Ausnahme des Kommas angegeben werden.

Zum Sortieren der Variablen und für die Ausgabe der angesprungenen Zeilen wird das »Bubblesort«-Verfahren verwendet, da es besonders einfach in Assembler (Listing 3, Quellcode) programmiert werden kann. Die Variablen- und Sprungtabelle wird im Bereich \$A000 bis \$BFFF, also »unter« dem Basic-ROM, abgelegt. Falls also eine Basic-Erweiterung verwendet wird, darf dieser Speicherbereich dadurch nicht verändert werden.
(Peter Rüschoff-Nadernann/hm)

SPRUNGBEFEHLE									
=====									
* ZIEL *	SPRUNG	DURCH	* IN	** ZIEL *	SPRUNG	DURCH	* IN	*	

10	ON -	GOTO	80	20	ON -	GOTO	80		
30	ON -	GOTO	80	40	ON -	GOTO	80		
50		RUN	40	50		THEN GOTO	60		
70	THEN ON -	GSUB	100	80	THEN ON -	GSUB	100		
90	THEN ON -	GSUB	100	100		GSUB	20		
100		THEN GSUB	60	100		THEN ON - GSUB	100		
200		GOTO	20						

Tabelle 2. Liste der angesprungenen Zeilen

VARIABLEN					
=====					
VARIABLE	ZEILEN			VARIABLE	ZEILEN

A	60	80	100	100	
B	120*	120	120*	120	
D	140			G	140
E	140			K	140
E	160*			G	160
D	160			K	160
D	160			K	160

Tabelle 3. Alle im Beispiel-Programm verwendeten Variablen

VARIABLEN							
=====							
VARIABLE	ZEILEN		VARIABLE		ZEILEN		

A	60	80	100	100			
B	120	120	120*	120*			
D	140	160	160	E	140	160*	
G	140	160		K	140	160	160

Tabelle 4. Alle Variablen des Beispiel-Programms in sortierter Reihenfolge.

programm : cross-ref c000 cab7

```

c000 : 4c 03 c9 20 55 c0 20 e3 e8
c008 : c8 20 fe c5 4c ff c4 ad 8c
c010 : b0 ca 4a 4a b0 03 20 55 50
c018 : c0 20 7e c7 20 8b c2 20 2b
c020 : 36 c4 b0 01 60 20 9c c7 0e
c028 : 20 8d c7 20 e3 c8 20 fe 08
c030 : c5 4c ff c4 20 c9 c0 20 47
c038 : ff c4 4c 7e c7 20 8b c2 ad
c040 : 20 36 c4 b0 01 60 20 9c 8f
c048 : c7 20 8d c7 20 ad c8 20 4e
c050 : 16 c8 4c ff c4 a9 06 85 9a
c058 : b6 20 b1 c3 20 ba c3 4c 83
c060 : 65 c0 20 b8 c2 a9 80 8d db
c068 : ad ca a5 bd c9 ff d0 01 7d
c070 : 60 20 e5 c3 c9 20 f0 f9 27
c078 : c9 00 f0 e6 c9 22 f0 32 30
c080 : c9 8f f0 de c9 a7 f0 22 0b
c088 : c9 88 f0 1e c9 3a f0 1a fc
c090 : c9 81 f0 16 20 10 c4 90 cf
c098 : 09 20 96 c6 20 0f c7 20 0a
c0a0 : 45 c4 a2 00 8e ad ca 4c 0a
c0a8 : 71 c0 a2 80 8e ad ca 4c 4c
c0b0 : 71 c0 20 e5 c3 c9 22 f0 3b
c0b8 : 07 c9 00 f0 06 4c b2 c0 d1
c0c0 : 4c 71 c0 20 b8 c2 4c 71 af
c0c8 : c0 a9 05 85 b6 8d ae ca 77
c0d0 : 20 b1 c3 20 ba c3 a5 bd 9a
c0d8 : c9 ff f0 15 20 c3 c8 a9 16
c0e0 : 00 8d a5 ca 4c ea c0 20 c9
c0e8 : b8 c2 a5 bd c9 ff d0 01 04
c0f0 : 60 a9 01 8d a8 ca 20 e5 44
c0f8 : c3 c9 00 f0 ea c9 22 f0 25
c100 : 3b c9 8f f0 e2 c9 89 f0 a6
c108 : 1b c9 8d f0 1a c9 a7 f0 fa
c110 : 19 c9 8a f0 18 c9 91 f0 c6
c118 : 17 c9 81 f0 16 c9 82 f0 2e
c120 : 15 4c f6 c0 4c 53 c1 4c 30
c128 : 5b c1 4c 52 c2 4c 63 c1 61
c130 : 4c d6 c1 4c 7c c1 4c 9d 23
c138 : c1 4c f6 c0 20 e5 c3 c9 c9
c140 : 22 f0 07 c9 00 f0 06 4c 0e
c148 : 3c c1 4c f6 c0 20 b8 c2 cc
c150 : 4c f6 c0 20 5d c4 a9 02 f2
c158 : 4c 76 c2 20 5d c4 a9 04 3f
c160 : 4c 76 c2 20 25 c4 20 19 c7
c168 : c4 90 0e 20 5d c4 ce a4 7c
c170 : ca ac a4 ca a9 10 4c 76 4c
c178 : c2 4c f6 c0 20 5d c4 ee 14
c180 : a6 ca a9 40 8d a3 ca 20 5f
c188 : d4 c5 20 25 c4 c9 b2 f0 33
c190 : 06 20 f7 c3 4c 8a c1 20 7d
c198 : d2 c4 4c f6 c0 ad a4 ca 60
c1a0 : 48 ce a6 ca 20 e5 c3 20 d3
c1a8 : 2d c4 f0 07 c9 2c f0 f1 fa
c1b0 : 4c a4 c1 68 8d a4 ca 20 35

```

```

c1b8 : 5d c4 a9 20 8d a3 ca 20 47
c1c0 : d4 c5 20 25 c4 20 2d c4 af
c1c8 : f0 06 20 f7 c3 4c c2 c1 ef
c1d0 : 20 d2 c4 4c f6 c0 20 5d c5
c1d8 : c4 20 25 c4 c9 89 f0 07 49
c1e0 : c9 8d f0 08 4c d9 c1 a9 9b
c1e8 : 0a 4c ee c1 a9 0c 20 0b 9e
c1f0 : c8 ad a3 ca 29 fe 85 bd ed
c1f8 : 8d a3 ca a2 00 20 d4 c5 3e
c200 : 20 3c c5 20 25 c4 ce a4 b1
c208 : ca 20 d4 c7 a5 bd 20 3f 58
c210 : c2 20 ea c7 20 79 00 f0 46
c218 : 1d c9 2c d0 ef 20 f7 c3 a6
c220 : ad a5 ca c9 4b 90 e5 20 9d
c228 : f7 c4 20 3c c5 20 3c c5 eb
c230 : 20 3c c5 4c 0c c2 20 2e 1d
c238 : c5 20 f7 c4 4c f6 c0 a0 64
c240 : 02 91 fb 20 a2 c4 a0 03 e7
c248 : a5 62 91 fb c8 a5 63 91 6c
c250 : fb 60 20 25 c4 20 19 c4 63
c258 : 90 11 20 5d c4 a9 80 20 00
c260 : 0b c8 ce a4 ca ac a4 ca 52
c268 : 4c 76 c2 ce a4 ca a9 80 c2
c270 : 8d a8 ca 4c f6 c0 8d a3 80
c278 : ca 20 0b c8 20 d4 c5 20 2e
c280 : 3c c5 20 e7 c7 20 d2 c4 f6
c288 : 4c f6 c0 20 00 c4 a9 00 50
c290 : 8d a7 ca 8d a5 ca 8d a6 89
c298 : ca 8d a3 ca 85 bd 8d ad 43
c2a0 : ca 85 57 a9 a0 85 58 a9 23
c2a8 : f9 8d a9 ca 85 fb a9 bf 8a
c2b0 : 8d aa ca 85 fc 4c d2 c3 fb
c2b8 : a0 00 b1 f9 d0 05 c8 b1 c0
c2c0 : f9 f0 0f a5 f9 85 f7 a5 a1
c2c8 : fa 85 f8 20 a5 c3 f0 11 25
c2d0 : 90 0f a5 fb 8d ab ca a5 7d
c2d8 : fc 8d ac ca a9 ff 85 bd 4b
c2e0 : 60 ad a7 ca 29 0e 8d a7 e3
c2e8 : ca a0 03 8c a4 ca a0 00 78
c2f0 : b1 f7 85 f9 c8 b1 f7 85 42
c2f8 : fa 60 a9 00 8d b0 ca 8d 31
c300 : b1 ca 8d b2 ca a9 ff 8d e5
c308 : b3 ca 8d b4 ca 20 79 00 ae
c310 : c9 2c f0 05 a2 0b 4c 37 ee
c318 : a4 20 73 00 c9 50 d0 03 12
c320 : 20 56 c5 20 73 00 c9 53 e6
c328 : d0 05 a9 08 20 92 c3 20 cc
c330 : 73 00 c9 4a d0 05 a9 04 43
c338 : 20 92 c3 20 73 00 c9 56 a1
c340 : d0 05 a9 02 20 92 c3 20 23
c348 : 73 00 c9 4a d0 05 a9 01 55
c350 : 20 92 c3 20 73 00 c9 4b a3
c358 : d0 08 a9 10 20 92 c3 20 7f

```

Listing 2. »Cross-Ref 64«. Das Programm sollte mit dem MSE eingegeben werden. Beachten Sie dazu bitte die Eingabehinweise auf Seite 8.

```

c360 : 73 00 c9 2c d0 2b c9 ab b0
c368 : f0 17 20 73 00 20 6b a9 5c
c370 : a5 14 8d b1 ca a5 15 8d 02
c378 : b2 ca 20 79 00 c9 ab d0 65
c380 : 10 20 73 00 20 6b a9 a5 cd
c388 : 14 8d b3 ca a5 15 8d b4 4c
c390 : ca 60 0d b0 ca 8d b0 ca 55
c398 : 60 20 a2 c4 ec b2 ca d0 7b
c3a0 : 03 cd b1 ca 60 20 a2 c4 6b
c3a8 : ec b4 ca d0 03 cd b3 ca be
c3b0 : 60 a9 a0 85 58 a9 00 85 9b
c3b8 : 57 60 20 8b c2 20 b8 c2 4e
c3c0 : a5 bd c9 ff d0 06 20 6c 4d
c3c8 : c5 4c ae a7 20 99 c3 90 53
c3d0 : ec 60 a5 2b 85 f9 a5 2c d2
c3d8 : 85 fa 60 ce a4 ca 20 e8 bf
c3e0 : c3 ee a4 ca 60 ee a4 ca 42
c3e8 : ac a4 ca b1 f7 60 20 09 e4
c3f0 : c4 20 3e f1 4c 00 c4 ee 48
c3f8 : a5 ca 20 09 c4 20 d2 ff c4
c400 : 48 a5 01 29 fe 85 01 68 71
c408 : 60 48 a9 37 85 01 68 60 a0
c410 : 20 09 c4 20 13 b1 4c 00 da
c418 : c4 c9 3a 10 06 c9 30 30 21
c420 : 02 38 60 18 60 20 e5 c3 80
c428 : c9 20 f0 f9 60 c9 00 f0 b3
c430 : 04 c9 3a f0 00 60 ad aa d5
c438 : ca cd ac ca d0 06 ad a9 b4
c440 : ca cd ab ca 60 20 67 c7 69
c448 : b0 12 20 6c c5 a9 5a a2 ef
c450 : ca 20 84 c8 ad b5 ca 85 23
c458 : 01 4c ae a7 60 20 a2 c4 3b
c460 : a0 04 91 fb 88 8a 91 fb 01
c468 : a9 00 85 bd ad a7 ca 4a 02
c470 : b0 12 ad a7 ca 09 01 8d 9e
c478 : a7 ca 20 af c4 a2 27 a0 c2
c480 : 02 4c 88 c4 a2 22 a0 07 2f
c488 : ad a6 ca 85 63 f0 0c 85 e5
c490 : bd 20 ee c5 a9 00 85 62 47
c498 : 20 af c4 a9 00 85 bd 4c b2
c4a0 : 3c c5 a0 03 b1 f7 aa 88 de
c4a8 : b1 f7 86 62 85 63 60 20 78
c4b0 : 09 c4 a2 90 38 20 49 bc f9
c4b8 : 20 df bd 20 a8 c5 4c 00 25
c4c0 : c4 49 ff 69 06 a8 f0 09 d1
c4c8 : a9 20 20 f7 c3 88 4c c6 c8
c4d0 : c4 60 ad a7 ca 4a 4a 90 6e
c4d8 : 06 20 2e c5 4c f7 c4 ae 28
c4e0 : a5 ca e0 28 b0 08 a9 20 5a
c4e8 : 20 f7 c3 4c df c4 ad a7 a8
c4f0 : ca 09 02 8d a7 ca 60 ad 1f
c4f8 : a7 ca 29 0d 8d a7 ca a9 85
c500 : 0d 20 f7 c3 a9 00 8d a5 b0
c508 : ca 20 ee c3 f0 1f c9 03 4c
c510 : d0 12 20 6c c5 a9 6d a2 24
c518 : ca 20 84 c8 ad b5 ca 85 eb
c520 : 01 4c ae a7 20 ee c3 f0 52
c528 : fb c9 03 f0 e5 60 ce ae e1
c530 : ca d0 08 a9 05 8d ae ca a7

c538 : 20 f7 c4 60 ad a5 ca 38 35
c540 : e9 0a b0 fc 49 ff 69 01 36
c548 : aa e8 ca f0 08 a9 20 20 c6
c550 : f7 c3 4c 4a c5 60 a9 04 93
c558 : a2 04 a0 68 20 ba ff 20 4a
c560 : b1 ff a9 68 20 93 ff a9 7a
c568 : 04 85 9a 60 ad b5 ca c9 29
c570 : 35 d0 19 a9 37 85 01 a9 80
c578 : a0 85 04 a0 00 b1 03 91 ad
c580 : 03 c8 d0 f9 e6 04 a5 04 88
c588 : c9 c0 d0 f1 ad b6 ca 85 ea
c590 : 7a ad b7 ca 85 7b a9 04 0b
c598 : 20 ae ff 20 c3 ff a9 03 fc
c5a0 : 85 9a ad b5 ca 85 01 60 32
c5a8 : 20 09 c4 20 87 b4 20 a6 6e
c5b0 : b6 48 a6 bd d0 03 20 c1 15
c5b8 : c4 68 aa a0 00 e8 ca f0 c3
c5c0 : 10 b1 22 20 f7 c3 c8 c9 8a
c5c8 : 0d d0 f3 20 e5 aa 4c be a1
c5d0 : c5 4c 00 c4 a2 00 ad a8 86
c5d8 : ca 0d a3 ca 8d a3 ca 0e a8
c5e0 : a3 ca b0 08 e8 e8 e8 e8 61
c5e8 : e8 4c df c5 a0 05 bd 7a c5
c5f0 : ca 20 f7 c3 e8 88 d0 f6 45
c5f8 : ad a3 ca d0 e2 60 ad a9 7f
c600 : ca 85 fb ad aa ca 85 fc 52
c608 : ad ab ca 85 57 ad ac ca 19
c610 : 85 58 a9 05 8d ae ca 20 86
c618 : ff c4 20 67 c7 f0 6e 20 6c
c620 : 90 c6 a0 01 b1 fb 20 f7 c7
c628 : c3 c8 b1 fb c9 20 f0 03 a2
c630 : 20 f7 c3 20 ea c6 20 3c 1f
c638 : c5 a0 03 b1 fb 85 62 c8 4b
c640 : b1 fb 85 63 a9 00 85 bd e9
c648 : 20 af c4 a0 05 b1 fb 0a 67
c650 : 90 05 a9 2a 20 f7 c3 20 24
c658 : 3c c5 ee af ca 20 41 c7 6b
c660 : 08 20 0f c7 20 67 c7 f0 73
c668 : 23 28 f0 cd ad af ca c9 ac
c670 : 08 90 0c 20 f7 c4 20 2e 4a
c678 : c5 20 90 c6 4c 86 c6 c9 f2
c680 : 04 90 03 20 ee c4 20 d2 cc
c688 : c4 4c 1f c6 28 4c ff c4 81
c690 : a9 00 8d af ca 60 a0 01 c7
c698 : 91 fb 20 e5 c3 a0 02 20 75
c6a0 : 10 c4 b0 0d 20 19 c4 b0 1f
c6a8 : 08 aa a9 20 91 fb 4c b7 0d
c6b0 : c6 91 fb 20 e5 c3 aa a9 bc
c6b8 : 40 e0 24 f0 0a e0 25 f0 ae
c6c0 : 05 4a 4a 4c cd c6 4a 48 d3
c6c8 : 20 e5 c3 aa 68 e0 28 d0 f1
c6d0 : 02 09 80 a0 00 91 fb 20 48
c6d8 : a2 c4 a0 04 91 fb 88 8a b5
c6e0 : 91 fb ad ad ca a0 05 91 79

```

Listing 2. »Cross-Ref 64«. (Fortsetzung)

c6e8 :	fb	60	a0	00	b1	fb	4a	4a	f4	c8b8 :	a2	c9	20	84	c8	88	d0	f6	d9
c6f0 :	4a	4a	4a	b0	0a	a2	25	4a	e7	c8c0 :	4c	d6	c8	a9	45	a2	c9	20	af
c6f8 :	b0	01	ca	8a	20	f7	c3	b1	61	c8c8 :	84	c8	a0	02	a9	6f	a2	c9	4d
c700 :	fb	0a	90	0a	a9	28	20	f7	b2	c8d0 :	20	84	c8	88	d0	f6	a0	02	c1
c708 :	c3	a9	29	20	f7	c3	60	38	7e	c8d8 :	a9	91	a2	c9	20	84	c8	88	86
c710 :	a5	fb	e5	b6	85	fb	b0	02	02	c8e0 :	d0	f6	60	a9	15	a2	ca	20	4a
c718 :	c6	fc	60	a5	fc	85	fe	38	91	c8e8 :	84	c8	a0	02	a9	39	a2	ca	bd
c720 :	a5	fb	e5	b6	85	fd	b0	02	22	c8f0 :	20	84	c8	88	d0	f6	a0	08	ed
c728 :	c6	fe	60	a0	00	18	b1	fb	19	c8f8 :	a9	4f	a2	ca	20	84	c8	88	a5
c730 :	d1	fd	f0	04	b0	0a	90	08	6a	c900 :	d0	f6	60	20	fa	c2	a5	7a	b9
c738 :	c8	c4	b6	f0	03	4c	2d	c7	05	c908 :	8d	b6	ca	a5	7b	8d	b7	ca	f0
c740 :	60	20	1b	c7	a9	03	85	b6	a6	c910 :	a5	01	8d	b5	ca	20	00	c4	87
c748 :	20	2b	c7	08	a9	06	85	b6	3f	c918 :	ad	b0	ca	4a	4a	4a	4a	90	5a
c750 :	28	60	a0	00	c4	b6	f0	0e	b2	c920 :	0e	20	34	c0	ad	b0	ca	4a	84
c758 :	b1	fb	48	b1	fd	91	fb	68	7c	c928 :	4a	4a	90	03	20	3d	c0	ad	66
c760 :	91	fd	c8	4c	54	c7	60	a5	fc	c930 :	b0	ca	4a	4a	90	03	20	03	c9
c768 :	fc	c5	58	d0	04	a5	fb	c5	60	c938 :	c0	ad	b0	ca	4a	90	03	20	ca
c770 :	57	60	18	a5	57	65	b6	85	39	c940 :	0f	c0	4c	6c	c5	0d	ff	ff	15
c778 :	57	90	02	e6	58	60	18	ad	b9	c948 :	ff	20	20	20	20	53	50	52	e6
c780 :	ab	ca	65	b6	8d	ab	ca	90	43	c950 :	55	4e	47	42	45	46	45	48	13
c788 :	03	ee	ac	ca	60	38	ad	ab	5d	c958 :	4c	45	0d	ff	ff	ff	ff	20	ca
c790 :	ca	e5	b6	8d	ab	ca	b0	03	86	c960 :	3d	3d	3d	3d	3d	3d	3d	3d	60
c798 :	ce	ac	ca	60	ad	ab	ca	85	e9	c968 :	3d	3d	3d	3d	3d	0d	00	20	b7
c7a0 :	57	ad	ac	ca	85	58	ad	a9	77	c970 :	49	4e	20	5a	45	49	4c	45	8e
c7a8 :	ca	85	fb	c5	57	08	ad	aa	ae	c978 :	20	20	20	20	53	50	52	55	60
c7b0 :	ca	85	fc	c5	58	d0	03	28	9d	c980 :	4e	47	20	44	55	52	43	48	88
c7b8 :	f0	19	20	1b	c7	20	2b	c7	5a	c988 :	ff	20	20	4e	41	43	48	ff	b9
c7c0 :	90	03	20	52	c7	20	0f	c7	6d	c990 :	00	2a	3d	3d	3d	3d	3d	3d	ca
c7c8 :	20	67	c7	d0	ed	20	72	c7	e1	c998 :	3d	3d	2a	3d	3d	3d	3d	3d	d3
c7d0 :	4c	a6	c7	60	ad	a4	ca	18	c9	c9a0 :	3d	3d	3d	3d	3d	3d	3d	3d	a0
c7d8 :	65	f7	85	7a	08	a5	f8	85	86	c9a8 :	3d	3d	3d	3d	3d	3d	2a	3d	5c
c7e0 :	7b	28	90	02	e6	7b	60	20	e0	c9b0 :	3d	3d	3d	3d	3d	3d	3d	3d	b0
c7e8 :	d4	c7	20	09	c4	20	73	00	e4	c9b8 :	2a	00	ff	ff	ff	41	4e	47	b4
c7f0 :	20	6b	a9	a0	00	a5	15	91	e9	c9c0 :	45	53	50	52	55	4e	47	45	7d
c7f8 :	fb	85	62	c8	a5	14	91	fb	a1	c9c8 :	4e	45	20	5a	45	49	4c	45	67
c800 :	85	63	20	0f	c7	20	af	c4	e7	c9d0 :	4e	0d	ff	ff	ff	20	3d	3d	15
c808 :	4c	45	c4	a0	02	0d	a8	ca	fd	c9d8 :	3d	3d	3d	3d	3d	3d	3d	3d	d8
c810 :	8d	a3	ca	91	fb	60	ad	ab	24	c9e0 :	3d	3d	3d	3d	3d	3d	3d	3d	e0
c818 :	ca	85	57	ad	ac	ca	85	58	18	c9e8 :	3d	3d	0d	00	2a	20	20	5a	e0
c820 :	ad	a9	ca	85	fb	ad	aa	ca	72	c9f0 :	49	45	4c	20	20	2a	20	20	07
c828 :	85	fc	a9	00	8d	a7	ca	8d	f2	c9f8 :	20	53	50	52	55	4e	47	20	45
c830 :	a5	ca	a9	05	8d	ae	ca	a9	12	ca00 :	44	55	52	43	48	20	20	20	32
c838 :	f6	8d	a6	ca	20	67	c7	f0	36	ca08 :	20	2a	20	20	20	49	4e	20	0f
c840 :	42	a0	00	84	bd	b1	fb	85	c7	ca10 :	20	20	20	2a	00	0d	ff	ff	f6
c848 :	62	c8	b1	fb	85	63	20	af	4d	ca18 :	ff	20	20	20	20	20	20	56	64
c850 :	c4	20	3c	c5	a0	02	b1	fb	c5	ca20 :	41	52	49	41	42	4c	45	4e	3d
c858 :	2c	a6	ca	d0	02	29	fe	8d	24	ca28 :	0d	ff	ff	ff	ff	20	3d	3d	a5
c860 :	a3	ca	a2	00	86	bd	20	df	a8	ca30 :	3d	3d	3d	3d	3d	3d	3d	0d	d0
c868 :	c5	a0	03	b1	fb	85	62	c8	7b	ca38 :	00	20	56	41	52	49	41	42	ff
c870 :	b1	fb	85	63	20	3c	c5	20	28	ca40 :	4c	45	20	20	20	5a	45	49	b7
c878 :	af	c4	20	d2	c4	20	0f	c7	05	ca48 :	4c	45	4e	ff	ff	ff	00	2a	1f
c880 :	4c	3c	c8	60	8d	90	c8	8e	c6	ca50 :	3d	3d	3d	3d	3d	3d	3d	3d	50
c888 :	91	c8	a2	00	8e	a5	ca	ad	c3	ca58 :	2a	00	0d	53	50	45	49	43	0b
c890 :	ff	ff	f0	12	c9	ff	f0	0f	8c										
c898 :	20	f7	c3	ee	90	c8	d0	03	1b										
c8a0 :	ee	91	c8	4c	8f	c8	60	20	13										
c8a8 :	3c	c5	4c	9b	c8	a9	45	a2	82										
c8b0 :	c9	20	84	c8	a0	02	a9	ec	5e										

Listing 2. »Cross-Ref 64«. (Fortsetzung)

```

ca60 : 48 45 52 20 56 4f 4c 4c 8d
ca68 : 20 21 21 0d 00 0d 41 42 f5
ca70 : 42 52 55 43 48 20 21 21 e5
ca78 : 0d 00 54 48 45 4e 20 46 77
ca80 : 4f 52 20 20 4e 45 58 54 1e
ca88 : 20 52 55 4e 20 20 4f 4e cd
ca90 : 20 2d 20 47 53 55 42 20 61

```

```

ca98 : 47 4f 54 4f 20 20 20 20 4a
caa0 : 20 20 2a 00 00 00 00 00 5b
caa8 : 00 f9 bf 00 00 00 05 00 a9
cab0 : 00 00 00 00 00 00 00 00 b1

```

Listing 2. »Cross-Ref 64«. (Schluß)

```

* = $C000
AKOPADR = $F7
NKOPADR = $F9
CHRGET = $0073
CHRGOT = $0079
GETZEIL = $A968
TABENDL = $57
TABENDH = $58
ZE11L = $FB
ZE11H = $FC
ZE12L = $FD
ZE12H = $FE
LAENGE = $B6
TEMP = $BD
;*****
JMP SCONTROL
VAR JSR VSTART ; SUCHT VARIABLEN
JSR DRVAR ; AUSGABE V.KOPF
JSR VAROUT ; " DER VAR.
JMP CARET2 ; ZEILENVORSCHUB
VAR2 LDA SYSTAT ; VAR. IN REIHENF.
LSR A ;
LSR A ;
BCS VAR3 ;
JSR VSTART ; NEIN: SUCHEN
JSR INTEND ; TAB. UM 1 KUEZ.
JSR INIT ; INITIALISIEREN
JSR TESTTAB ; TABELLE LEER?
BCS LVAR2 ;
RTS ; JA: RTS
LVAR2 JSR SORT ; TAB. SORTIEREN
JSR DECTEND
JSR DRVAR
JSR VAROUT
JMP CARET2
SPR JSR SPRUNG ; SPRUENGE SUCHEN
JSR CARET2 ; UND AUSGEBEN
JMP INTEND
AN JSR INIT ; ANGESPRUNGENE
JSR TESTTAB ; ZEILEN AUSGEBEN
BCS LAN1 ;
RTS
LAN1 JSR SORT
JSR DECTEND
JSR DRAN
JSR ZIELE
JMP CARET2
;
VSTART LDA #6 ; = ZNR+CODE+ZNR
STA LAENGE
JSR SETTAB
JSR SETBER
JMP VZEILE2
VZEILE JSR NEXTLINE
VZEILE2 LDA #10000000
STA DBSTAT
LDA TEMP ; KODE FUER ENDE?
CMP #FF ; NEIN: WEITER
BNE VARLOOP ; JA: ENDE
RTS
VARLOOP JSR NXTCHR
CMP #20 ;
BEQ VARLOOP
CMP #0 ; ZEILENENDE ?
BEQ VZEILE
CMP #22 ; = " ODER = REM?
BEQ VARSEARCH ; DANN BIS " 0.
CMP #8F ; NEUE ZEILE
BEQ VZEILE ; WEITER
CMP #A7 ; THEN
BEQ SETDB
CMP #88 ; LET

```

```

BEQ SETDB
CMP # ; STATEMENT
BEQ SETDB
CMP #81 ; FOR
BEQ SETDB
JSR CHKALPH
BCC LTEST1
JSR VARGET
JSR INZEIL
JSR OVFLOW
LTEST1 LDX #10000000
STX DBSTAT
JMP VARLOOP
SETDB LDX #10000000
STX DBSTAT
JMP VARLOOP
VARSEARCH JSR NXTCHR
CMP #22
BEQ LWS1
CMP #0
BEQ LWS2
JMP VARSEARCH
LWS1 JSR VARLOOP
LWS2 JSR NEXTLINE
JMP VARLOOP
;
SPRUNG LDA #5 ; = ZNR+CODE+ZNR
STA LAENGE
STA LINES ; LF NACH 5 ZEILEN
JSR SETTAB
JSR SETBER
LDA TEMP
CMP #FF
BEQ SZEILE2
JSR DRSPR
LDA #0
STA ZLANG
JMP SZEILE1
SZEILE JSR NEXTLINE
SZEILE1 LDA TEMP ; KODE FUER ENDE?
CMP #FF
BNE SZEILE2
RTS
SZEILE2 LDA #100000001 ; STATUS FUER
STA THSTAT ; 5 LEERZEICHEN
SPRLOOP JSR NXTCHR
CMP #0 ; ZEILENENDE ?
BEQ SZEILE
CMP #22 ; = " ODER = REM?
BEQ SPRSUCH ; DANN BIS " 0.
CMP #8F ; NEUE ZEILE
BEQ SZEILE ; WEITER
CMP #89 ; GOTO ?
BEQ CGOTO
CMP #8D ; GOSUB ?
BEQ CGOSUB
CMP #A7 ; THEN ?
BEQ CTHERN
CMP #8A ; RUN ?
BEQ CRUN
CMP #91 ; ON ?
CMP #81 ; FOR ?
BEQ CFOR
CMP #82 ; NEXT ?
BEQ CNEXT
JMP SPRLOOP
CGOTO JMP GOTO
CGOSUB JMP GOSUB
CTHERN JMP THEN
CRUN JMP RUN
CON JMP ON

```

```

CFOR JMP FOR
CNEXT JMP NEXT
JMP SPRLOOP
SPRSUCH JSR NXTCHR
CMP #22
BEQ LSS1
CMP #0
BEQ LSS2
JMP SPRSUCH
LSS1 JSR SPRLOOP
LSS2 JSR NEXTLINE
JMP SPRLOOP
;
GOTO JSR ZEILOUT
LDA #100000010
JMP AUSGABE
;
GOSUB JSR ZEILOUT
LDA #100000100
JMP AUSGABE
;
RUN JSR CHKSPC ; NAE. ZCHN NUM.?
JSR CHKNUM ; JA: AUSGABE
BCC LRUN1 ; NEIN: WEITER
JSR ZEILOUT
DEC POINTER ; DA CHKSPC AUF N.
LDY POINTER ; ZCHN STELLT
LDA #100010000
JMP AUSGABE
LRUN1 JSR SPRLOOP
;
FOR JSR ZEILOUT
INC FANZ ; OFFENE FOR+1
LDA #100000000
STA TSTAT
JSR TEXTAUS
LFOR1 JSR CHKSPC ; TEXT BIS '=' AUS-
CMP #B2 ; GEBEN ('=':B2)
BEQ LFOR2 ;
JSR CHROUT ; GEBEN
JMP LFOR1
LFOR2 JSR LINEFEED
JMP SPRLOOP
;
NEXT LDA POINTER
PHA
NEXT1 DEC FANZ
NEXT2 JSR NXTCHR
JSR TRZCHN
BEQ NEXT3
CMP # ;
BEQ NEXT1
JMP NEXT2
NEXT3 PLA
STA POINTER
JSR ZEILOUT
LDA #100100000
STA TSTAT
JSR TEXTAUS
JSR CHKSPC
JSR TRZCHN ; SOLANGE ZCHN
BEQ LNEXT3 ; AUSG. BIS Z.ENDE
JSR CHROUT
JMP LNEXT1
LNEXT3 JSR LINEFEED
JMP SPRLOOP

```

Listing 3. Der Source-Code von »Cross-Ref 64«.

```

ON      ;
LON1    JSR ZEILOUT
        JSR CHKSPC
        CMP #89 ; GOTO ?
        BEQ ONTO
        CMP #8D ; GOSUB?
        BEQ ONSUB
        JMP LON1
ONTO    LDA #200001010
        JMP LON2
ONSUB   LDA #200001100
LON2    JSR SORCDD
        LDA TSTAT
        AND #211111110
        STA TEMP
        STA TSTAT
        LDX #0
        JSR TEXTAUS
        JSR TABS
        JSR CHKSPC
        DEC POINTER
        JSR SETGET
LON3    LDA TEMP
        JSR TSPRUNG ; SPR.ART, ZNR U.
        JSR GETSPR2 ; ZIEL IN TABELLE
        JSR CHRGET ; ENDE STATEMENT?
        BEQ LON4 ; JA: LINEFEED
        CMP #, ; WEITERE ZEILENNR.
        BNE LON3 ; NEIN: LINEFEED
        JSR CHROUT
        LDA ZLANG
        CMP #75
        BCC LON3
        JSR CARET ; LINEFEED
        JSR TABS
        JSR TABS
        JSR TABS
        JMP LON3
LON4    JSR FUNFLIN
        JSR CARET
        JMP SPRLOOP
        ;
TSPRUNG LDX #2
        STA (ZEI1L),Y
        JSR ZNR ; ZNR AUS #14/#15
        LDX #3 ; IN TABELLE
        LDA #62
        STA (ZEI1L),Y
        INY
        LDA #63
        STA (ZEI1L),Y
        RTS
        ;
THEN    JSR CHKSPC
        JSR CHKNUM ; ZIFFER?
        BCC THENDE ;
        JSR ZEILOUT ; THEN MIT ZEILE
        LDA #100000000
        JSR SORCDD
        DEC POINTER ; EIN ZCHN ZURUECK,
        LDX POINTER ; DA CHKNUM ZU WEIT
        JMP AUSGABE
THENDE  DEC POINTER ; TH-BIT SETZEN
        LDA #100000000
        STA THSTAT
        JMP SPRLOOP
        ;
AUSGABE STA TSTAT
        JSR SORCDD
        JSR TEXTAUS
        JSR TABS
        JSR GETSPR ; ZIEL IN TABELLE
        JSR LINEFEED ; UND AUSGEBEN
        JMP SPRLOOP
        ;
;*****
; UNTERROUTINEN
;*****
INIT    JSR ROMOFF
        LDA #0
        STA ZSTAT
        STA ZLANG
        STA FANZ
        STA TSTAT
        STA TEMP
        STA DBSTAT
        STA TABENDL
        LDA #A0
        STA TABENDH
        LDA #F9
        STA TABSTRTL
        STA ZE11L ; ANFANG DER TAB-
        LDA #BF ; ELLE AB #BFF9
        STA TABSTRTH
        STA ZE11H

```

```

        JMP BASTRT
        ;
NEXTLINE LDX #0
        LDA (NKOPADR),Y
        BNE LNL1
        INY
        LDA (NKOPADR),Y
        BEQ NLENDE
LNL1    LDA NKOPADR
        STA AKOPADR
        LDA NKOPADR +1
        STA AKOPADR +1
        JSR CKBEN ; ENDE BEREICH?
        BEQ LNL2 ; NEIN
        BCC LNL2 ;
        ;
NLENDE  LDA -ZE11L ; SETZT TABELLEN-
        STA TENDL ; ENDE AUF ZE11
        LDA ZE11H
        STA TENDH
        LDA #FF ; KODE FUER
        STA TEMP ; ENDE TESTLOOP
        RTS
        ;
LNL2    LDA ZSTAT
        AND #1110 ; ZNR FLAG LOE.
        STA ZSTAT
        LDX #3 ; ZEIGT AUF 0.
        STY POINTER ; BASIC-ZEICHEN
        LDX #0
        LDA (AKOPADR),Y ; ZEIGER AUF
        STA NKOPADR ; NAECHSTE B.
        INY ; ZEILE HOLEN
        LDA (AKOPADR),Y
        STA NKOPADR +1
        RTS
        ;
;*****
; HOLT SYSTEM-PARAMETER (SYSTAT)
; SETZT DRUCKER
; HOLT BEREICH (IN BERST/EN)
;*****
SYSGET  LDA #0
        STA SYSTAT
        STA BERSTL
        STA BERSTH
        LDA #FF
        STA BERENL
        STA BERENH
        JSR CHRGET
        CMP #,
        BEQ LSYS0
        LDX #0B
        JMP #A437 ; FEHLER: SYNTAX
LSYS0   JSR CHRGET ; DRUCKER
        CMP #P
        BNE LSYS1
        JSR SETPRT
        JSR CHRGET
        CMP #S
        BNE LSYS2
        LDA #201000
        JSR SETSYS
LSYS2   JSR CHRGET ; ANGESSPR. ZEILEN?
        CMP #J
        BNE LSYS3
        LDA #200100
        JSR SETSYS
LSYS3   JSR CHRGET ; VARIABLEN ?
        CMP #V
        BNE LSYS4
        LDA #200010
        JSR SETSYS
LSYS4   JSR CHRGET ; VARIABLEN SORT.?
        CMP #J
        BNE LSYS5
        LDA #200001
        JSR SETSYS
LSYS5   JSR CHRGET ; KOMMENTAR ?
        CMP #K
        BNE LSYS6
        LDA #210000
        JSR SETSYS
        JSR CHRGET
LSYS6   CMP #, ; FOLGEN ZNRS?
        BNE LSYS8 ; NEIN: RTS
        CMP #AB ; "-"
        BEQ LSYS7
        JSR CHRGET
        JSR GETZEIL
        LDA #14
        STA BERSTL
        LDA #15
        STA BERSTH
        JSR CHRGET
        CMP #AB ; "-"

```

```

        BNE LSYS8 ; NEIN: ENDE
LSYS7   JSR CHRGET
        JSR GETZEIL
        LDA #14
        STA BERENL
        LDA #15
        STA BERENH
        RTS
LSYS8   ;
        SETSYS ; ORA SYSTAT ; <AC> MIT SYSTAT
        STA SYSTAT ; VERKNUEPFEN
        RTS
        ;
;*****
CKBST   JSR ZNR ; TEST, OB AKT. ZNR
        CPX BERSTH ; GROESSER ODER
        BNE LBST1 ; GLEICH BER. ANF
        CMP BERSTL
        RTS
LBST1   ;
        CKBEN ; JSR ZNR ; TEST, OB AKT. ZNR
        CPX BERENH ; GROESSER ODER
        BNE LBEN1 ; GLEICH BER. ENDE
        CMP BERENL
        RTS
LBEN1   ;
        SETTAB ; LDA #A0
        STA TABENDH
        LDA #00
        STA TABENDL
        RTS
        ;
SETBER  JSR INIT
LBER1   JSR NEXTLINE
        LDA TEMP
        CMP #FF
        BNE LBER2
        JSR SYGOF
        JMP #A7AE
LBER2   JSR CKBST
        BCC LBER1
        RTS
        ;
BASTRT  LDA #2B ; ADRESSE DER
        STA NKOPADR ; ERSTEN BASIC
        LDA #2C ; ZEILE HOLEN
        STA NKOPADR +1
        RTS
        ;
PREVCHR DEC POINTER ; HOLT VORIGES
        JSR AKTCHR ; ZCHN, OHNE
        INC POINTER ; ZEIGER ZU
        RTS ; VERSTELLEN
        ;
NXTCHR  INC POINTER ; HOLT NAECHSTES
        ; ; ZEICHEN
        AKTCHR LDX POINTER
        LDA (AKOPADR),Y
        RTS
        ;
GET      JSR ROMON
        JSR #F13E
        JMP ROMOFF
        ;
CHROUT  INC ZLANG
        JSR ROMON
        JSR #FFD2
        ;
ROMOFF  PHA
        LDA 1
        AND #11111110
        STA 1
        PLA
        RTS
        ;
ROMON   PHA
        LDA #37
        STA 1
        PLA
        RTS
        ;
CHKALPH JSR ROMON ; TEST, OB AKKU ALPHA
        JSR #B113 ; C=1, WENN BUCHST.
        JMP ROMOFF ; C=0, WENN ANDERS
        ;
CHKNUM  CMP #3A ; GROESSER ALS '9'?
        BPL LCN1
        CMP #30 ; KLEINER ALS '0'?
        BMI LCN1 ; JA: C=0
        SEC ; NEIN: C=1
        RTS
LCN1    CLC

```

**Listing 3. Assembler-Source-Code
zu »Cross-Ref 64«
(Fortsetzung)**

```

RTS
;
CHKSPC JSR NXTCHR
CMP #20
BEQ CHKSPC
RTS
;
TRZCHN CMP #0 ; TESTET AUF NULL
BEQ LTZ1 ; UND ' '
CMP #1 ; JA: Z=1
BEQ LTZ1 ; NEIN: Z=0
LTZ1 RTS
;
TESTTAB LDA TABSTRTH
CMP TENDH
BNE LTETA
LDA TABSTRTL
CMP TENDL
LTETA RTS
;
OVFLOW JSR TESTEND
BCS LOVF
JSR SYSOFF
LDA #<TOVER
LDX #>TOVER
JSR MASKE
JMP $A7AE
LOVF RTS
;
ZEILOUT JSR ZNR ; ZNR NACH $62/$63
LDY #4 ; ZNR IN TAB
STA (ZEI1L),Y
DEY
TXA
STA (ZEI1L),Y
LDA #0
STA TEMP
LDA ZSTAT ; BEI 1 KEINE ZNR
LSR A
BCS LZ01
LDA ZSTAT ; ZNR-FLAG =1
ORA #0001
STA ZSTAT
JSR ZAUSGABE
LDX #39
LDY #2
JMP LZ02
LZ01 LDX #34
LDY #7
LZ02 LDA FANZ ; FOR OFFEN?
STA #63
BEQ LZ03 ; NEIN: RTS
STA TEMP
JSR LTA2
LDA #0
STA #62
JSR ZAUSGABE
LDA #0
STA TEMP
JMP TABS
;
ZNR LDY #3 ; HOLT NR. DER
LDA (AKOPADR),Y ; AKT. ZEILE
TXA ; NACH $62/$63
DEY ; UND A,X
LDA (AKOPADR),Y
STX #62
STA #63
RTS
;
ZAUSGABE JSR ROMON
LDX #90 ; $62/$63 WIRD ALS
SEC ; INTEGERZAHL IM
JSR #BC49 ; 16-BIT ADRESS-
JSR #BDDF ; FORMAT AUSGEBEN
JSR STROUT ; $AB1E
JMP ROMOFF
;
RECBUN EOR #FF
ADC #6
TAY
LREC1 BEQ LREC2
LDA #20
JSR CHROUT
DEY
JMP LREC1
LREC2 RTS
;
LINEFEED LDA ZSTAT
LSR A ; BEI BIT 1=0
LSR A ; KEIN LF
BCC LLF1
JSR FUNFLIN
JMP CARET
LLF1 LDX ZLANG ; KEIN LF, DANN

```

```

CPX #40 ; BIS POS. 40
BCS LLF2 ; AUFFUELLEN
LDA #20
JSR CHROUT
JMP LLF1
LDA ZSTAT
ORA #0010 ; LF-FLAG SETZEN
STA ZSTAT
RTS
;
CARET LDA ZSTAT ; LF-FLAG LOESCHEN
AND #1101
STA ZSTAT
CARET2 LDA #00
JSR CHROUT
LDA #0 ; ZEILENLAENGE:=0
STA ZLANG
CARET3 JSR GET ; TASTE GEDRUECKT?
BEQ LCAR4 ; NEIN: RTS
LCAR1 CMP #3 ; RUN/STOP?
BNE LCAR3
LCAR2 JSR SYSOFF ; JA
LDA #<TBREAK
LDX #>TBREAK
JSR MASKE
JMP $A7AE
LCAR3 JSR GET ; 2. TASTE
BEQ LCAR3 ; FUER CONT.
CMP #3
BEQ LCAR2
LCAR4 RTS
;
FUNFLIN DEC LINES
BNE LFUNF1
LDA #5
STA LINES
JSR CARET
LFUNF1 RTS
;
TABS LDA ZLANG ; ZEILENLAENGE HOL.
SEC
LTAB1 SBC #10 ; SOLANGE POSITIV
BCS LTAB1 ; 10 ADDIEREN
EOR #FF ; ERGEBNIS INVERT.
ADC #1
TXA
INX
LTAB2 DEX
BEQ LTAB3
LDA #20
JSR CHROUT
JMP LTAB2
LTAB3 RTS
;
SETPRT LDA #4
LDX #4
LDY #68 ; SEK B + #60
JSR $FFBA ; FILPAR
JSR $FFB1 ; LISTEN
LDA #68 ; SEKADR
JSR $FF93 ; SEKLST
LDA #4
STA #9A
RTS
;
SYSOFF LDA PROPORT
CMP #35
BNE GETRES
LDA #37 ; ROM AN,
STA 1 ; $A000 KOPIERT WIRD
LDA #A0
STA 4
LDY #0
LROMK LDA (3),Y
STA (3),Y
INX
BNE LROMK
INC 4
LDA 4
CMP #C0
BNE LROMK
;
GETRES LDA GETSP
STA #7A
LDA GETSP +1
STA #7B
;
PRTOFF LDA #4
JSR $FFAE ; UNLIST
JSR $FFC3 ; SCHLIESSEN
LDA #3
STA #9A
LDA PROPORT
STA 1
RTS

```

```

;
STROUT JSR ROMON
JSR #B487 ; STRINGPARAM. HOL.
JSR #B6A6 ; FRESTRING
PHA
LDX TEMP
BNE LSTR0
JSR RECBUN
LSTR0 PLA
TXA ; STR.LAENGE
LDY #0
INX
LSTR1 DEX
BEQ LSTR2
LDA ($22),Y
JSR CHROUT ; DRUCKEN
INX
CMP #00
BNE LSTR1
JSR $AAE5
JMP LSTR1
LSTR2 JMP ROMOFF
;
TEXTAUS LDX #0
LDA THSTAT
ORA TSTAT
STA TSTAT
LTA1 ASL TSTAT
BCS FUENFAUS
INX
INX ; SETZE X AUF DEN
INX ; NAECHSTEN TEXTAB-
INX ; SCHNITT
INX
JMP LTA1
FUENFAUS LDY #5
LTA2 LDA TEXT,X
JSR CHROUT
INX
DEY
BNE LTA2
LDA TSTAT
BNE LTA1
RTS
;
;=====
; AUSGABE ALLER VARIABLEN IN DER TAB
;=====
VAROUT LDA TABSTRTL ; ZEIGER AUF
STA ZE11L ; ANFANG DER
LDA TABSTRTH ; TABELLE SETZEN
STA ZE11H
LDA TENDL
STA TABENDL
LDA TENDH
STA TABENDH
LDA #5 ; STARTWERT 3
STA LINES ; FUER ABSAETZ
JSR CARET2 ; ZLANG:=0 U.
LVAR01 JSR TESTEND ; ZLANG ENDE?
BEQ LVAR09 ; JA: RTS
LVAR02 JSR SETVANZ
LDY #1 ; 1. ZEICHEN
LDA (ZE11L),Y ; AUSGEBEN
JSR CHROUT
INX ; 2. ZEICHEN
LDA (ZE11L),Y ; AUSGEBEN,
CMP #20 ; WENN <>LEER
BEQ LVAR03
JSR CHROUT
LVAR03 JSR STATOUT
JSR TABS
LVAR04 LDY #3 ; ZNR HOLEN
LDA (ZE11L),Y ; UND NACH
STA #62 ; $62/$63
INX
LDA (ZE11L),Y
STA #63
LDA #0
STA TEMP
JSR ZAUSGABE
LDY #5 ; "*" HINTER ZNR
LDA (ZE11L),Y ; WENN DEF
ASL A ; SONST WEITER
BCC LVAR05
LDA #'*
JSR CHROUT
LVAR05 JSR TABS
INC VANZ
JSR VARVERG ; ALTVAR=NEUVAR?
PHP ; JA: NUR ZNR

```

**Listing 3. Der Source-Code zu
»Cross-Ref 64«
(Fortsetzung)**

```

JSR INZEI1 ; NEIN: UND NAM
JSR TESTEND ; ENDE DER TAB?
BEQ LVAR08 ; JA: ENDE
PLP ; AUSGEBEN
BEQ LVAR04 ; (NUR ZNR)
LDA VANZ ; MEHR ALS 7
CMP #8 ; VAR AUSGEG.?
BCC LVAR06 ; JA: LF,
JSR CARET ; VANZ:=0
JSR FUNFLIN
JSR SETVANZ
JMP LVAR07

LVAR06 CMP #4 ; 4 VAR?: LF
BCC LVAR07 ; WENIGER: WEITER
JSR LLF2

LVAR07 JSR LINEFEED
JMP LVAR02

LVAR08 PLP

LVAR09 JMP CARET2

;
SETVANZ LDA #0
STA VANZ
RTS
;

;=====
; HOLEN EINER VARIABLEN AUS BASIC TEXT
;=====
VARGET LDY #1 ; ZEICHEN IN TAB.
STA (ZEI1L),Y
JSR NXTCHR ; NAECHSTES ZCHN
LDY #2
JSR CHKALPH ; BUCHSTABE?
BCS LVARGET1 ; JA: IN TAB.
JSR CHKNUM ; ZIFFER?
BCS LVARGET1 ; JA: IN TAB.
TAX ; NEIN:AC SICHERN
LDA #20 ; LEER IN TAB.
STA (ZEI1L),Y
JMP LVARGET2

LVARGET1 STA (ZEI1L),Y
JSR NXTCHR
TAX

LVARGET2 LDA #01000000
CPX #'$ ; STRING?
BEQ LVARGET4
CPX #'% ; INTEGER?
BEQ LVARGET3
LSR A ; REAL
LSR A
JMP LVARGET5

LVARGET3 LSR A ; INTEGER
LVARGET4 PHA ; AC SICHERN
JSR NXTCHR
TAX
PLA

LVARGET5 CPX #'( ; DIMENSIONIERT?
BNE LVARGET6 ; NEIN
ORA #10000000 ; JA!
LVARGET6 LDY #0 ; STATUS AN 5.
STA (ZEI1L),Y ; STELLE IN TAB.
JSR ZNR ; ZNR IN TAB.
LDY #4 ; AN 4./5. STELLE
STA (ZEI1L),Y
DEY
TXA
STA (ZEI1L),Y
LDA DBSTAT ; DEF/BEN AN 5.
LDY #5 ; STELLE
STA (ZEI1L),Y
RTS
;

;=====
; PRUEFT STATUSBYTE DER VARIABLEN
; UND GIBT TYP - $, %, () - AUS
;=====
STATOUT LDY #0
LDA (ZEI1L),Y
LSR A
LSR A ; BITS 0-3
LSR A ; UEBERLESEN
LSR A
LSR A ; REAL?
BCS LSTAT2
LDX #'% ; INTEGER?
LSR A
BCS LSTAT1
DEX ;STRING!

LSTAT1 TXA
JSR CHROUT

LSTAT2 LDA (ZEI1L),Y ; FELD?
ASL A
BCC LSTAT3 ; NEIN: RTS
LDA #'(
JSR CHROUT
LDA #'%
JSR CHROUT

LSTAT3 RTS
;

```

```

;=====
; SORTIEREN VON FELDERN NACH
; "BUBBLESORT"
; UEBERGABE:
; "LAENGE" DES FELDES
; "TABSTR" ANFANG DES FELDES
; "TABEND" ENDE
;=====
; ERHOEHE ZEIGER1 UM LAENGE
;=====
INZEI1 SEC
LDA ZE11L
SBC LAENGE
STA ZE11L
BCS LINZ1
DEC ZE11H
LINZ1 RTS
;

;=====
; ROUTINE BESTIMMT ZEIGER AUF I+1-TES
; ELEMENT; PARAMETER: LAENGE, ZE11
;=====
IZEI62 LDA ZE11H
STA ZE12H
SEC
LDA ZE11L
SBC LAENGE
STA ZE12L
BCS LADD1
DEC ZE12H
LADD1 RTS
;

;=====
; VERGLEICH DER DURCH ZE11, ZE12 UND
; LAENGE BESTIMMTEN FELDER
; C=1, WENN A(I)>A(I+1)
; Z=1, WENN A(I)=A(I+1) (Y=LAENGE)
;=====
VERGL LDY #0
LV1 CLC
LDA (ZE11L),Y
CMP (ZE12L),Y
BEQ LV2 ; GLEICH
BCS LV3 ; GROSSER
BCC LV3 ; KLEINER
LV2 INY ; NEIN: NAECHSTES
CPY LAENGE ; X=LAENGE?
BEQ LV3 ; JA: ENDE
JMP LV1 ; ELEMENT TESTEN
LV3 RTS
;

;=====
; VERGLEICH ZWEIER VARIABLEN
; Z=1, WENN NAME UND STATUS GLEICH
;=====
VARVERG JSR IZEI62
LDA #3 ; NUR 3 BYTE
STA LAENGE ; (STAT, NAME)
JSR VERGL ; VERGLEICHEN
PHP ; LAENGE WIEDER
LDA #6 ; 6
STA LAENGE
PLP
RTS
;

;=====
; TAUSCHE ZWEI FELDER, DIE DURCH ZE11
; ZE12L UND LAENGE BESTIMMT SIND
;=====
EXCHANGE LDY #0
LEXCH1 CPY LAENGE ; SOLANGE Y<LAE.
BEQ LEXCH2
LDA (ZE11L),Y
PHA ; VERTAUSCHE
LDA (ZE12L),Y ; ZWEI BYTES
STA (ZE11L),Y
PLA
STA (ZE12L),Y
INY
JMP LEXCH1
LEXCH2 RTS
;

;=====
; ENDE DER TABENDLLE ERREICHT?
; (ZE11 = TABEND? ==> Z=1)
;=====
TESTEND LDA ZE11H
CMP TABENDH
BNE LTEND1
LDA ZE11L
CMP TABENDL
LTEND1 RTS
;

;=====
; TABEND UM EIN ELEMENT ERNIED
;=====

```

```

DECTAB CLC
LDA TABENDL
ADC LAENGE
STA TABENDL
BCC LDECTAB
INC TABENDH
LDECTAB RTS
;

;=====
; TEND UM EIN ELEMENT VERRINGERN
;=====
INTEND CLC
LDA TENDL
ADC LAENGE
STA TENDL
BCC LINTEN1
INC TENDH
LINTEN1 RTS
;

;=====
; TEND UM EIN ELEMENT ERHOEHEN
;=====
DECTEND SEC
LDA TENDL
SBC LAENGE
STA TENDL
BCS LDETEN1
DEC TENDH
LDETEN1 RTS
;

;=====
; SORTIEREN
;=====
SORT LDA TENDL ; TABEND SETZEN,
STA TABENDL ; GEHT BEI SORT
LDA TENDH ; VERLOREN
STA TABENDH
LSORT1 LDA TABSTR1L ; ZEIGER1 AUF
STA ZE11L ; TABSTR
CMP TABENDL
PHP ; UND
LDA TABSTRTH ; TEST, OB
STA ZE11H ; ENDE DER
CMP TABENDH ; TABELLE
BNE LSORTY ; ERREICHT
PLP ; JA: ENDE
BEQ LSORTW

LSORTY JSR IZEI62
JSR VERGL ; VERGLEICHEN
BCC LSORT3 ; GROSSER, DANN
JSR EXCHANGE ; TAUSCHEN
LSORT3 JSR INZEI1 ; NAECHSTES
BNE TESTEND ; ENDE ERREICHT?
BNE LSORTY ; NEIN: WEITER
JSR DECTAB ; JA: NAECHSTER
JMP LSORT1

LSORTW RTS
;

;=====
; SETZT CHRGET - ZEIGER AUF AKT. ZEICH.
;=====
SETGET LDA POINTER
CLC
ADC AKOPADR
STA $7A
PHP
LDA AKOPADR + 1
STA $7B
PLP
BCC LSET1
INC $7B
LSET1 RTS
;

;=====
; HOLT SPRUNGSZIEL NACH TABELLENENDE
;=====
GETSPR JSR SETGET
GETSPR2 JSR ROMON
JSR CHRGET
JSR GETZEIL
LDY #0
LDA $15
STA (ZE11L),Y
STA $62
INY
LDA $14
STA (ZE11L),Y
STA $63
JSR INZEI1
JSR ZAUSGABE
JMP OVFLOW
;

```

Listing 3. Der Source-Code zu »Cross-Ref 64« (Fortsetzung)

```

=====
; SETZT BIT FUER SPRUNG NACH ON, THEN
;=====
SORCOD LDY #2
ORA THSTAT
STA TSTAT
STA (ZEI1L),Y
RTS
;=====
; AUSGABE DER ANGESPRUNGENEN ZEILEN
; IN DER TABELLE: H,L DES SPRUNGZIELS,
; CODE DER SPRUNGART,
; L,H DER AUFRUF-ZEILE
;=====
ZIELE LDA TENDL ; ZEIGER AUF ENDE
STA TABENDL ; TEND WURDE NICHT
LDA TENDH ; GEAEENDERT
STA TABENDH
LDA TABSTRTL ; ZEIGER AUF
STA ZE11L ; ANFANG
LDA TABSTRTH
STA ZE11H
LDA #0
STA ZSTAT
STA ZLANG
LDA #5
STA LINES
LDA #11110110
STA FANZ
LZIEL1 JSR TESTEND ; TABELLE LEER?
BEQ STPZIEL ; JA: BEENDEN
LDY #0 ; HIGH, LOW-BYTE
STY TEMP ; F. RECUN
LDA (ZEI1L),Y ; DES ZIELS
STA #62 ; HOLEN
INY
LDA (ZEI1L),Y
STA #63
JSR ZAUSGABE
JSR TABS
LDY #2 ; SPRUNGART HOLEN
LDA (ZEI1L),Y
BIT FANZ
BNE LZIEL2
AND #11111110
LZIEL2 STA TSTAT
LDX #0
STX TEMP ; F. RECUN
JSR LTA1 ; TEXTAUSGABE
LDY #3 ; LOW,HIGH-BYTE
LDA (ZEI1L),Y ; DER AUFRUFENDEN
STA #62 ; ZEILE
INY
LDA (ZEI1L),Y
STA #63
JSR TABS
JSR ZAUSGABE
JSR LINEFEED
JSR INZE11
JMP LZIEL1
STPZIEL RTS
;=====
; AUSGEBEN VON TEXT
;=====
MASKE STA ADRE+1
STX ADRE+2
LDX #0
STX ZLANG
ADRE LDA $FFFF
BEQ MASKE3
CMP #255
BEQ MASKE4
JSR CHROUT
MASKE1 INC ADRE+1
BNE MASKE2
INC ADRE+2
MASKE2 JMP ADRE
MASKE3 RTS

MASKE4 JSR TABS
JMP MASKE1
;=====
; AUSGABE DER "MASKEN" FUER SPR U. VAR
;=====
DRAN LDA #<TSPRK
LDX #>TSPRK
JSR MASKE
LDY #2
LDRAN1 LDA #<TANR
LDX #>TANR
JSR MASKE
DEY
BNE LDRAN1
JMP DRSPR1
DRSPR LDA #<TSPRK
LDX #>TSPRK
JSR MASKE
LDY #2
LDSP1 LDA #<TSPRR
LDX #>TSPRR
JSR MASKE
DEY
BNE LDSP1
LDY #2
DRSPR1 LDA #<TSPRU
LDX #>TSPRU
JSR MASKE
DEY
BNE LDSP2
RTS
;=====
; STEUERROUTINE
; RUFT DIE ROUTINEN ENTSPRECHEND
; DER PARAMETER AUF
;=====
SCONTROL JSR SYSGET ; PARAMETER HOLEN
LDA #7A ; GET-ZEIGER RETTEN
STA GETSP ;
LDA #7B ;
STA GETSP +1
LDA 1 ; PROZESSORPORT
STA PROPORT ; RETTEN
JSR ROMOFF ; ROM AUSSCHALTEN
LDA SYSTAT ; ABFRAGE DER
LSR A ; PARAMETER
LSR A
LSR A
BCC LCON1
JSR SPR
LDA SYSTAT
LSR A
LSR A
BCC LCON1
JSR AN
LCON1 LDA SYSTAT

LSR A
LSR A
BCC LCON2
JSR VAR
LCON2 LDA SYSTAT
LSR A
BCC LCON3
JSR VAR2
LCON3 JMP SYSOFF ; STARTZUSTAND
; WIEDERHERSTELLEN
;=====
; TEXTE
;=====
TSPRK .BYTE 13,255,255,255
.TEXT " SPRUNGBEFEHLE"
.BYTE 13,255,255,255,255
.TEXT " ====="
.BYTE 13,0
TSPRR .TEXT " IN ZEILE SPRUNG DURCH"
.BYTE 255
.TEXT " NACH"
.BYTE 255,0
TSPRU .TEXT "=====*"
.BYTE 0
TANK .BYTE 255,255,255
.TEXT "ANGESPRUNGENE ZEILEN"
.BYTE 13,255,255,255
.TEXT " ====="
.BYTE 13,0
TANR .TEXT "* ZIEL * SPRUNG DURCH *"
.TEXT " IN *"
.BYTE 0
TVARK .BYTE 13,255,255,255
.TEXT " VARIABLEN"
.BYTE 13,255,255,255,255
.TEXT " ====="
.BYTE 13,0
TVARR .TEXT " VARIABLE ZEILEN"
.BYTE 255,255,255,0
TVARU .TEXT "=====*"
.BYTE 0
TOVER .BYTE 13
.TEXT "SPEICHER VOLL !!
.BYTE 13,0
TBREAK .BYTE 13
.TEXT "ABBRUCH !!"
.BYTE 13,0
TEXT .TEXT "THEN FOR NEXT "
.TEXT "RUN ON - GSUB "
.TEXT "GOTO *"
TSTAT .BYTE 0
POINTER .BYTE 0
ZLANG .BYTE 0
FANZ .BYTE 0
ZSTAT .BYTE 0
THSTAT .BYTE 0
TABSTRTL .BYTE $F9
TABSTRTH .BYTE $BF
TENDL .BYTE 0
TENDH .BYTE 0
DBSTAT .BYTE 0
LINES .BYTE 5
VANZ .BYTE 0
SYSTAT .BYTE 0
BERSTL .BYTE 0
BERSTH .BYTE 0
BERENL .BYTE 0
BERENH .BYTE 0
PROPORT .BYTE 0
GETSP .BYTE 0,0
.END

```

Listing 3. Der Source-Code zu »Cross-Ref 64« (Schluß)

```

10 rem
20 gosub 100 : goto 200
30 rem
40 run 50
50 rem
60 if a=0 then gosub 100 : goto 50
70 rem
80 on a goto 10,20,30,40
90 rem
100 if a(>0) then on a gosub 100,90,70,80

```

```

110 rem
120 b=0 : if b=1 then b=b+2
130 rem
140 print d;g;e;k
150 rem
160 e=g+d+k:d=k

```

Listing 1. Dieses, für sich unsinnige, Basic-Programm soll die Leistungsfähigkeit der Dokumentationshilfe »Cross-Reference-Liste« veranschaulichen