```
                104           CPY YSAVE     03A2 A447     133           LDY YREG
                105           BCC LOOPAS    03A4 C8       134  ZAEHL    INY
                106           JSR ADRWRI    03A5 20BAFC   135  NEXTON   JSR NXTA1
                107           LDY #00       03A8 90E3     136           BCC ZWEIG
                108  FOLOOP   LDA (A1L),Y   03AA 60       137           RTS
                109           JSR COUT      03AB          138  ;
                110           INY           03AB          139  ;
                111           CPY YSAVE     03AB 2C00C0   140  ADRWRI  BIT KBD
                112           BCC FOLOOP    03AE 1016     141           BPL RETAD2
                113  NXTONE   JSR NXTA1     03B0 2028FD   142           JSR KEYREA
                114           BCC NEWONE    03B3 C9A0     143           CMP #" "
                115           RTS           03B5 D008     144           BNE CTRLC
                116  ;                      03B7 2C00C0   145  SPACE    BIT KBD
                117  ZSRCH    LDY #00       03BA 10FB     146           BPL SPACE
                118           BEQ ZWEIG     03BC 2028FD   147           JSR KEYREA
                119  ;                      03BF C983     148  CTRLC    CMP #$83
                120  ;                      03C1 D003     149           BNE RETAD2
                121  SEARCH   LDY #01       03C3 68       150           PLA
                122  ;                      03C4 68       151           PLA
                123  ZWEIG    LDA (A1L),Y   03C5 60       152           RTS
                124           CMP A4L,Y     03C6 2092FD   153  RETAD2   JSR PRA1
                125           BNE NEXTON    03C9 A9A0     154           LDA #" "
                126           DEY           03CB 4CEDFD   155           JMP COUT
                127           BMI ZWRI      03CE          156  ;
                128           LDA (A1L),Y   03CE          157  ;
                129           CMP A4L                    158           END
                130           BNE ZAEHL
                131  ZWRI     STY YREG
                132           JSR ADRWRI    ***** END OF ASSEMBLY
```

angegebenen Bereichs. Da le-II die Zeichen auch invers F) und im Flashing-Modus dargestellt werden können, be- ligt die Routine den gesamten sbereich von $00 bis $FF. male Zahlen zwischen $80 und l-Zeichen) werden als inver- dargestellt. Da man auch Pro- mit Kleinbuchstaben (ab $E0) chen könnte, den Dump auch in- ren will, erfolgt für den Bereich FF eine Konversion in Groß- ben.

ches Programm ist dem Autor annt [1], dieses ist jedoch nicht für den Apple-II-Plus geeignet, da es auf Sweet-16-Routinen zurückgreift, die nur im „alten" Betriebssystem existieren.

Suche nach ASCII-Zeichen

Hier erfolgt der Aufruf durch xxxx.yyyy CTRLY F. Die Bedingungen für die einzelnen Parameter sind die gleichen wie beim ASCII-Dump. Nach dem Aufruf springt der Cursor eine Zeile tiefer, und man kann eine Zeichenkette mit bis zu 255 Zeichen eingeben. Die Routine sucht nach diesem String im angegebenen Bereich, ob der String nun normal, invers, blinkend oder in Kleinbuchstaben im Speicher steht, und druckt dann die Startadresse des Strings (hexadezimal) sowie den String in seiner ursprünglichen Form. Kleinbuchstaben werden dabei natürlich nicht richtig dargestellt.

Suche nach Adressen

Der Aufruf geschieht mit aabb<xxxxx .yyyy CTRLY S. Die Routine sucht im Bereich xxxx bis yyyy nach der Adresse aabb. Intern steht eine Adresse in der Reihenfolge Lower Byte, Higher Byte (also bb-aa) im Speicher. Durch den Aufruf erfolgt diese Vertauschung automatisch.

Suche nach beliebigen Bytes

Dafür muß man eingeben: aa<xxxx.yyyy CTRLY Z. Die Routine sucht dann nach dem spezifizierten Byte im angegebenen Speicherbereich.
Der ASCII-Dump und alle Suchroutinen können durch das Drücken von Space oder CTRL C unterbrochen bzw. beendet werden. Will man nach Betätigung der Space-Taste weitermachen, so drückt man wieder auf Space. Will man die Suche oder den Dump ganz beenden, so drückt man CTRL C.
Das Listing im Bild wurde mit Hilfe des LISA-Assemblers erstellt und der Quellcode direkt beim Assemblieren von einer Olympia-ES100-Schreibmaschine gedruckt.

Literatur

[1] Hex-ASCII Memory Dump, The Apple Orchard, March/April 1980, Page 79.

Was tut dieses Programm?

gedruckte kleine Basic-Pro- sich im Manuskript-Stapel faktion. Möglicherweise tut Sinnvolles – am besten, Sie es selbst mal aus. Leider konn- dem Listing auch nicht welchem Zweck es dient.

```
1 REM MC 4/1982 - FE
10 FORI=1TO6:H$="":READ A
20 A=A/2
30 IFA=INT(A)THENH$=" "+H$:GOTO50
40 H$="*"+H$
50 A=INT(A):IFA=OTHENPRINTH$:NEXT:END
60 GOTO20
70 DATA253685,152213,152213
80 DATA253685,151717,606807
```

Roland Vogt

Mnemonischer Übersetzer für CBM-Computer

Zwar enthält die Computerserie CBM-3000 im Gegensatz zur 2000-Serie bereits ein fest im ROM gespeichertes Monitorprogramm; dieses ist aber nicht gerade sehr komfortabel. Die Eingabe von Maschinenprogrammen erfordert damit oft das Nachschlagen von Operationscodes in der 6502-Befehlstabelle. Das hier vorgestellte Programm gestattet es aber, Programme in der mnemonischen Assemblersprache ein- und auszugeben; es enthält dazu einen Disassembler und einen kleinen „Line-by-line"-Assembler.

Da es sich hier um keinen „symbolischen" Assembler handelt, ist es nötig, alle Adressen absolut und hexadezimal zu schreiben. Vor dem Hex-Wert muß ein Dollar-Zeichen stehen. Diese Einschränkung ist leicht zu verstehen, wenn man bedenkt, daß das Programm nur 1328 Bytes lang ist. Das Umschreiben auf einen anderen Computertyp ist leider nicht ganz einfach, da umfassender Gebrauch von den CBM-ROM-Programmen gemacht wird, was das Programm sehr verkürzte.
Der Start erfolgt mit RUN oder SYS(1040). Dann befindet man sich im CBM-Monitorprogramm. Alle normalen Monitorbefehle können weiterverwendet werden, jedoch stehen nun vier weitere Befehle zur Verfügung, die zur Ein- und Ausgabe in Assemblerschreibweise dienen. Im einzelnen bewirken die Befehle folgendes:

D = Disassemblieren

Befehlsformat: .D aaaa,eeee
Der Speicherbereich von Adresse aaaa bis Adresse eeee wird disassembliert, das heißt, in mnemonischer Form ausgedruckt. Die Ausführung des Befehls kann jederzeit durch Drücken der Stop-Taste unterbrochen werden. Trifft der Disassembler auf einen Befehl, den es nicht gibt, gibt er als mnemonischen Befehl ILL aus. Es ist darauf zu achten, daß man mit dem Disassemblieren selbstverständlich immer beim ersten Byte eines Befehls beginnen muß. Ein Beispiel für die Verwendung des D-Befehls:

```
.D 050E,0513
., 050E LDY #$00
., 0510 LDA ($FB),Y
., 0512 PHA
., 0513 JSR $FDD5
.
```

Es ist vielleicht etwas störend, daß die hexadezimalen Speicherinhalte nicht noch zusätzlich gedruckt werden. Der Autor wollte jedoch erreichen, daß ebenso wie beim M- und R-Befehl des Monitors die Ausgabe so aussieht, daß man sie z. B. nach einer Korrektur wieder als Eingabe verwenden kann. Wenn aber die hexadezimalen Werte ebenfalls mit ausgegeben würden, wäre das nicht mehr möglich, ohne die Hexcodes vorher in einer Tabelle nachzuschlagen, was ja gerade vermieden werden sollte.

, = Assembler-Eingabe

Befehlsformat: ., aaaa bbb Adresse
Der in mnemonischer Abkürzung angegebene Befehl bbb wird in die Speicherstelle aaaa geschrieben. Für den Wert der Adresse gelten die Regeln der 650[2] Assemblerschreibweise. Es können jedoch, wie bereits erwähnt, nur hexadezimale Werte verwendet werden, denen ein „$"-Zeichen vorangestellt ist.

A = Automatische Adressen-Vorgabe

Befehlsformat: .A aaaa
Von der Adresse aaaa an gibt der Computer jeweils eine vierstellige Hexadresse aus, deren Format dem ,-Befehl entspricht. Der Cursor befindet sich hinter der Adresse. Es können nun Befehle in mnemonischer Schreibweise eingegeben werden, ohne auf die Adresse achten zu müssen. Wenn eine Zeile durch Drücken der Return-Taste abgeschlossen wird, gibt der Computer die nächstfolgende Adresse aus. Dadurch entfällt das lästige Eintippen der Adressen, wenn man längere Programmstücke eingibt. Man beendet die Ausgabe von Adressen, indem man statt eines gültigen Befehls eine ungültige Folge von drei Zeichen, z. B. ., eingibt und die Zeile mit Return abschließt. Der Computer befindet sich dann wieder im Normalzustand.

0 = Rückkehr zu Basic

Befehlsformat: .0
Beim Start des Programms mit RUN oder SYS(1040) werden die Speicherstellen $03FA/$03FB geändert. Sie zeigen normalerweise auf die Fehlerroutine des Monitors und nach dem Assembler-Start auf den Anfang des Programms. Wenn man mit X das Monitorprogramm verläßt oder mit G ein Programm startet, wird diese Änderung nicht rückgängig gemacht, so daß bei einem weiteren Aufruf des Monitorprogramms durch SYS(1024) oder den BRK-Befehl das Assembler-Disassemblerprogramm wieder hin zur Verfügung steht. Wenn man das Monitorprogramm aber mit 0 verläßt, wird die Änderung der genannten Speicherstellen rückgängig gemacht, und das Programm ist ausgeschaltet. Die Benutzung des 0-Befehls ist immer dann notwendig, wenn man das Assembler-Disassembler-Programm löschen oder überschreiben will, da sonst das Monitorprogramm bei einem Eingabefehler wieder hin zu diesem Programm verzweigt und es zu unkontrollierbaren Effekten kommen kann, wenn sich dieses nicht mehr dort befindet. Nach Verwendung des 0-Befehls kann das Programm mit RUN oder SYS(1040) wieder gestartet werden.

...lump des Assemblers/Disassemblers für den CBM 3000. Auf eine disassemblierte Wiedergabe wurde hier verzichtet, da sich das Pro... ja auch selbst disassemblieren kann

```
0000 03 04 0A 00 9E 31 30 00
0008 00 00 00 00 00 00 1D 8D
0010 FA 03 A9 04 8D 03 4C 11
0018 FD C9 44 F0 4C 8B 07 20
0020 EB E7 20 E7 90 2F 20 97
0028 E7 20 A7 E7 90 24 20 E7
0030 20 01 F3 F0 19 38 FD E5
0038 FB A5 FE E5 FC 0E A0 2C
0040 20 15 FE 20 E7 20 CD FD
0048 4C 5E 04 56 FD 4C 12 09
0050 20 0E A2 97 DD C3 05 F0
0058 03 D0 F8 BD 5B 06 85 66
0060 F3 06 AA BD 18 05 20 FF
0068 BD 51 05 20 D2 FF 8A 05
0070 20 D2 FF 20 CD A5 66 C9
0078 EE D0 03 4C 05 C9 FF D0
0080 08 A9 41 20 D2 FF 4C 0B
0088 05 29 80 A9 23 20 D2 FF
0090 A5 29 40 F0 05 A9 28 20
0098 D2 FF A9 24 20 D2 FF A5
00A0 29 20 F0 11 20 0E 05 20
00A8 0E 05 20 75 E7 68 20 75
00B0 E7 4C D4 04 20 0E 20 75
00B8 E7 A5 66 29 10 F0 05 A9
00C0 29 20 D2 FF A5 29 08 F0
00C8 05 A9 2C 20 D2 FF A5 66
00D0 29 04 F0 05 A9 59 20 D2
00D8 FF A5 66 29 01 F0 05 A9
00E0 29 D2 FF 4C 3A 04 A0 00
00E8 B9 48 20 D5 FD 68 60 4C
00F0 4C 53 53 53 54 54 54 50
00F8 50 50 50 41 53 44 49 44
0100 41 4F 43 43 43 42 42 42
0108 42 42 42 42 4A 4A 41 52
0110 52 43 43 43 43 53 53 4E
0118 52 52 42 54 54 41 44 44
0120 44 44 54 54 54 41 58 59
0128 4C 48 4C 48 44 52 4F 4D
0130 50 50 49 43 43 45 4E 4D
0138 50 56 56 4D 53 53 53 4F
0140 4F 4C 4C 4C 4C 45 45 45
0148 4F 54 54 52 53 58 4C 41
0150 58 59 41 58 59 58 59 41
0158 41 41 41 50 50 43 43 43
0160 43 43 58 58 59 59 44 41
0168 52 50 58 59 54 43 53 51
0170 45 49 4C 43 53 50 52 4C
0178 52 4C 52 4C 52 43 44 49
0180 56 43 44 49 50 53 49 4B
0188 58 53 4C 00 00 01 05 06
0190 08 09 0A 0D 0E 10 11 15
0198 16 18 19 1D 1E 20 21 24
01A0 25 26 28 29 2A 2C 2D 2E
01A8 30 31 35 36 38 39 3D 3E
01B0 40 41 45 46 48 49 4A 4C
01B8 4D 4E 50 51 55 56 58 59
01C0 5D 5E 60 61 65 66 68 69
01C8 6A 6C 6D 6E 70 71 75 76
01D0 78 79 7D 7E 81 84 85 86
01D8 88 8A 8C 8D 8E 90 91 94
01E0 95 96 98 99 9A 9D A0 A1
01E8 A2 A4 A5 A6 A8 A9 AA AC
01F0 AD AE B0 B1 B4 B5 B6 B8
01F8 B9 BA BC BD BE C0 C1 C4
0200 C5 C6 C8 C9 CA CC CD CE
0208 D0 D1 D5 D6 D8 D9 DD DE
0210 E0 E1 E4 E5 E6 E8 E9 EA
0218 EC ED EE F0 F1 F5 F6 F8
0220 F9 FD FE EE EE 4D 00 00
0228 EE 80 FF 20 20 00 5A 0C
0230 0C EE 2A 2C 2C 20 4D 00
0238 00 EE 80 FF 20 20 20 00
0240 5A 0C 0C EE 2A 2C 2C 20
0248 4D 00 00 EE 80 FF 20 20
0250 20 00 5A 0C 0C EE 2A 2C
0258 2C EE 4D 00 00 EE 80 FF
0260 70 20 20 00 5A 2C 2C 00
0268 00 00 00 00 00 00 00 00
0270 00 00 00 00 00 00 00 00
0278 00 00 00 00 00 00 00 00
0280 00 00 00 00 00 00 00 00
0288 00 00 00 00 00 00 00 00
0290 00 00 00 00 00 00 00 00
0298 00 00 00 00 00 00 00 00
02A0 00 00 00 00 00 00 00 00
02A8 00 00 00 00 00 00 00 00
02B0 00 00 00 00 00 00 00 00
02B8 00 00 00 00 00 00 00 00
02C0 00 00 00 00 00 00 00 00
02C8 00 00 00 00 00 00 00 00
02D0 00 00 00 00 00 00 00 00
02D8 00 00 00 00 00 00 00 00
02E0 00 00 00 00 00 00 00 00
02E8 00 00 00 00 00 00 00 00
02F0 00 00 00 00 00 00 00 00
02F8 00 00 00 00 00 00 00 00
0300 00 00 00 00 00 00 00 00
0308 00 00 00 00 00 00 00 00
0310 00 00 00 00 00 00 00 00
0318 00 00 00 00 00 00 00 00
0320 00 00 00 00 00 00 00 00
0328 00 00 00 00 00 00 00 00
0330 00 00 00 00 00 00 00 00
0338 00 00 00 00 00 00 00 00
0340 00 00 00 00 00 00 00 00
0348 00 00 00 00 00 00 00 00
0350 00 00 00 00 00 00 00 00
0358 00 00 00 00 00 00 00 00
0360 00 00 00 00 00 00 00 00
0368 00 00 00 00 00 00 00 00
0370 00 00 00 00 00 00 00 00
0378 00 00 00 00 00 00 00 00
0380 00 00 00 00 00 00 00 00
0388 00 00 00 00 00 00 00 00
0390 00 00 00 00 00 00 00 00
0398 00 00 00 00 00 00 00 00
03A0 00 00 00 00 00 00 00 00
03A8 00 00 00 00 00 00 00 00
03B0 00 00 00 00 00 00 00 00
03B8 00 00 00 00 00 00 00 00
03C0 00 00 00 00 00 00 00 00
03C8 00 00 00 00 00 00 00 00
03D0 00 00 00 00 00 00 00 00
03D8 00 00 00 00 00 00 00 00
03E0 00 00 00 00 00 00 00 00
03E8 00 00 00 00 00 00 00 00
03F0 00 00 00 00 00 00 00 00
03F8 00 00 00 00 00 00 00 00
0400 00 00 00 00 00 00 00 00
0408 00 00 00 00 00 00 00 00
0410 00 00 00 00 00 00 00 00
0418 00 00 00 00 00 00 00 00
0420 00 00 00 00 00 00 00 00
0428 00 00 00 00 00 00 00 00
0430 00 00 00 00 00 00 00 00
0438 00 00 00 00 00 00 00 00
0440 00 00 00 00 00 00 00 00
0448 00 00 00 00 00 00 00 00
0450 00 00 00 00 00 00 00 00
0458 00 00 00 00 00 00 00 00
0460 00 00 00 00 00 00 00 00
0468 00 00 00 00 00 00 00 00
0470 00 00 00 00 00 00 00 00
0478 00 00 00 00 00 00 00 00
0480 00 00 00 00 00 00 00 00
0488 00 00 00 00 00 00 00 00
0490 00 00 00 00 00 00 00 00
0498 00 00 00 00 00 00 00 00
04A0 00 00 00 00 00 00 00 00
04A8 00 00 00 00 00 00 00 00
04B0 00 00 00 00 00 00 00 00
04B8 00 00 00 00 00 00 00 00
04C0 00 00 00 00 00 00 00 00
04C8 00 00 00 00 00 00 00 00
04D0 00 00 00 00 00 00 00 00
04D8 00 00 00 00 00 00 00 00
04E0 00 00 00 00 00 00 00 00
04E8 00 00 00 00 00 00 00 00
04F0 00 00 00 00 00 00 00 00
04F8 00 00 00 00 00 00 00 00
0500 00 00 00 00 00 00 00 00
0508 00 00 00 00 00 00 00 00
0510 00 00 00 00 00 00 00 00
0518 00 00 00 00 00 00 00 00
0520 00 00 00 00 00 00 00 00
0528 00 00 00 00 00 00 00 00
0530 00 00 00 00 00 00 00 00
0538 00 00 00 00 00 00 00 00
0540 00 00 00 00 00 00 00 00
0548 00 00 00 00 00 00 00 00
0550 49 44 44 44 54 54 54 41
0558 41 58 59 4C 48 4C 48 44
0560 42 4E 45 4E 45 4E 45 4E
0568 52 4F 4D 50 50 49 43 43
0570 45 4E 4D 50 56 56 4D 53
0578 53 53 4F 4F 4C 4C 4C 4C
0580 45 45 45 4F 54 54 52 53
0588 58 4C 41 58 59 41 58 59
0590 58 59 41 41 41 41 50 50
0598 43 43 43 43 58 58 59 59
05A0 44 41 52 50 58 59 54 43
05A8 53 51 45 49 4C 43 53 50
05B0 52 4C 52 4C 52 43 44 49
05B8 56 43 44 49 50 53 49 4B
05C0 58 53 4C 00 00 01 05 06
05C8 08 09 0A 0D 0E 10 11 15
05D0 16 18 19 1D 1E 20 21 24
05D8 25 26 28 29 2A 2C 2D 2E
05E0 30 31 35 36 38 39 3D 3E
05E8 40 41 45 46 48 49 4A 4C
05F0 4D 4E 50 51 55 56 58 59
05F8 5D 5E 60 61 65 66 68 69
0600 6A 6C 6D 6E 70 71 75 76
0608 78 79 7D 7E 81 84 85 86
0610 88 8A 8C 8D 8E 90 91 94
0618 95 96 98 99 9A 9D A0 A1
0620 A2 A4 A5 A6 A8 A9 AA AC
0628 AD AE B0 B1 B4 B5 B6 B8
0630 B9 BA BC BD BE C0 C1 C4
0638 C5 C6 C8 C9 CA CC CD CE
0640 D0 D1 D5 D6 D8 D9 DD DE
0648 E0 E1 E4 E5 E6 E8 E9 EA
0650 EC ED EE F0 F1 F5 F6 F8
0658 F9 FD FE EE EE 4D 00 00
0660 EE 80 FF 20 20 00 5A 0C
0668 0C EE 2A 2C 2C 20 4D 00
0670 00 EE 80 FF 20 20 20 00
0678 00 5A 0C 0C EE 2A 2C 2C
0680 EE 4D 00 00 EE 80 FF 20
0688 20 20 00 5A 0C 0C EE 2A
0690 2C 2C EE 4D 00 00 EE 80
0698 FF 70 20 20 00 5A 2C 2C
06A0 EE 2A 2C 2C 4D 00 00 00
06A8 EE EE 20 20 20 00 5A 0C
06B0 0C EE 2A EE 2C 30 4D 00
06B8 80 00 00 00 EE 80 EE 20
06C0 20 20 00 5A 0C 0C 0A EE
06C8 2A EE 2C 2C 2A 80 4D 00
06D0 00 00 EE 80 EE 20 20 20
06D8 00 5A 0C 0C EE 2A 2C 2C
06E0 80 4D 00 00 00 EE 80 EE
06E8 20 20 20 00 5A 0C 0C EE
06F0 2A 2C 2C 38 35 17 17 27
06F8 0D 17 27 17 27 22 17 17
0700 27 2B 17 17 27 26 16 1C
0708 16 29 0C 16 29 1C 16 29
0710 21 16 16 29 2F 16 16 29
0718 34 18 18 28 0B 18 28 25
0720 18 28 23 18 18 28 2D 18
0728 18 28 33 0E 0E 2A 0A 0E
0730 2A 25 0E 2A 24 0E 0E 2A
0738 31 0E 0E 2A 03 05 03 04
0740 15 08 05 03 04 1D 03 05
0748 03 04 09 03 37 03 02 00
0750 01 02 00 01 07 00 06 02
0758 00 01 1E 00 02 00 01 2E
0760 00 36 02 00 01 1B 19 1B
0768 19 11 14 19 13 1B 19 11
0770 20 19 19 11 2C 19 19 11
0778 1A 0F 1A 0F 10 12 0F 32
0780 1A 0F 10 1F 0F 0F 10 30
0788 0F 0F 10 C9 2C F0 03 4C
0790 D5 08 20 B6 E7 20 A7 E7
0798 B0 03 4C 12 09 20 97 E7
07A0 20 97 E7 20 EB E7 A2 00
07A8 86 42 20 EB E7 95 43 E8
07B0 E0 03 D0 F6 A2 00 A5 43
07B8 DD 18 05 D0 0E A5 44 DD
07C0 51 05 D0 07 A5 45 DD 8A
07C8 05 F0 08 E8 E0 39 D0 E6
07D0 4C 12 09 86 43 20 CF FF
07D8 C9 20 F0 F9 C9 0D D0 07
07E0 A9 EE 85 42 4C 72 08 C9
07E8 41 D0 04 A9 FF D0 F3 C9
07F0 28 D0 05 A9 40 20 B1 08
07F8 C9 23 D0 05 A9 80 20 B1
0800 08 20 B6 E7 B0 03 4C 12
0808 09 85 44 20 CF FF C9 20
0810 F0 F9 C9 0D D0 03 4C 72
0818 08 85 45 20 B9 08 B0 04
0820 A5 45 D0 1A 0A 0A 0A 0A
0828 85 45 20 EB E7 20 B9 08
0830 B0 03 4C 12 09 05 45 85
0838 45 A9 20 20 B1 08 C9 0D
0840 D0 03 4C 72 08 C9 29 D0
0848 05 A9 10 20 B1 08 C9 2C
0850 D0 05 A9 08 20 B1 08 C9
0858 58 D0 05 A9 04 20 B1 08
0860 C9 59 D0 05 A9 02 20 B1
0868 08 C9 29 D0 05 A9 01 20
0870 B1 08 A2 97 BD F3 06 C5
0878 43 D0 30 BD 5B 06 C5 42
0880 D0 29 A0 00 BD C3 05 91
0888 FB 20 D5 FD A5 42 C9 EE
0890 F0 16 C9 FF F0 12 29 20
0898 F0 07 A5 45 91 FB 20 D5
08A0 FD A5 44 91 FB 20 D5 FD
08A8 4C 08 09 CA D0 C6 4C 12
08B0 09 05 42 85 42 20 CF FF
08B8 60 C9 30 90 17 C9 47 B0
08C0 12 38 E9 30 C9 11 90 05
08C8 38 E9 07 38 60 C9 0A B0
08D0 02 38 60 18 60 C9 41 F0
08D8 02 4C 19 09 20 EB E7 20
08E0 A7 E7 20 97 E7 20 97 E7
08E8 A9 00 85 00 A0 2C 20 15
08F0 FE 20 6A E7 A2 08 86 9E
08F8 A9 1D 9D 6E 02 CA D0 FA
0900 A9 91 20 D2 FF 4C 56 FD
0908 A5 00 D0 03 4C EC 08 4C
0910 56 FD A9 4C 85 00 4C F7
0918 E7 C9 4F F0 03 4C 12 09
0920 A9 4C 85 00 A9 F7 8D FA
0928 03 A9 E7 8D FB 03 4C 89
0930 C3 97 33 33 30 35 31 2C
```

...erschiebung eines Daten-...locks im RAM des CBM

...unterschiedlichsten Anwendun... ...eine schnelle Verschiebung eines ...locks im RAM notwendig, z. B. in ...men zur Textverarbeitung oder ...ung. Im CBM enthält das 3000er ...system eine entsprechende Rou...

...kann diese von Basic aus nicht ...werden. Startet man nämlich ...49887), so wird durch das Ab... des Arguments 49887 die An... ...sse des zu verschiebenden Da... in den Zellen 5C und 5D über... Eine Ausführung der Routine ...hindert, und es treten ungewoll... ein.

Durch ein kleines Maschinenprogramm kann die CBM-Routine jedoch trotzdem verwendet werden. Dieses bringt nach Start durch SYS (988) die Anfangsadresse des Datenblocks nach 5C/5D und startet dann die CBM-Routine. Ein Überschreiben der Anfangsadresse ist jetzt nicht mehr möglich, da für das Argument 988 die Zellen 5C/5D besetzt werden, bevor die Anfangsadresse des Speicherblocks dort eingeschrieben wird.
Das aufgelistete Basic-Programm wurde erstellt, um die notwendigen Parameter einfacher bzw. dezimal eingeben zu können (*Bild*).
Der Kern des Ganzen, das Maschinenprogramm, besteht aus nur 12 Bytes und wird in den Zeilen 900 und 920 eingespeichert.
Das folgende Beispiel kann zur Kontrolle dienen, ob das Programm läuft.

```
ERSTER SPEICHERPLATZ?          32768
LETZTER SPEICHERPLATZ?         33267
LETZTER ZIELPLATZ?             33767
```

Die obere Bildschirmhälfte erscheint in der unteren Hälfte noch einmal. Dieses Beispiel zeigt nur sehr begrenzt die Möglichkeiten der beschriebenen Routine, z. B. auch für bewegte Graphik auf dem Bildschirm. Zur Anwendung sind der Phantasie jedenfalls keine Grenzen gesetzt.

Hartmut Bennöhr

Literatur

[1] ROM und RAM bei PET und CBM. Mikrocomputer-Anwendungen, Franzis Sonderheft Nr. 33.

```
  REM ******************************
  REM * MASCHINENPROGRAMM-EINGABE *
  REM ******************************
  REM
  DATA169,,133,92,169,,133,93,32,223,194,96
  FORS=988TO999:READT:POKES,T:NEXT
  REM
  REM************************
  REM* BEDIENUNGSPROGRAMM *
  REM************************
  REM
  S=256
  INPUT"ERSTER SPEICHERPLATZ";A1:A%=A1/S
  INPUT"LETZTER SPEICHERPLATZ";AN:E%=(AN+1)/S
  INPUT"LETZTER ZIELPLATZ";ZN:Z%=(ZN+1)/S
  POKE989,A1-A%*S:POKE993,A%
  POKE87,AN+1-E%*S:POKE88,E%
  POKE85,ZN+1-Z%*S:POKE86,Z%
  SYS(988)
READY.
```

...erschiebung unter Zuhilfenahme einer CBM-3000-ROM-Routine

Jörg Bayerlein

CBM-3032 als Oszilloskop

Für den bekannten Kleincomputer CBM-3032 wird in dieser Arbeit ein Maschinenprogramm beschrieben, das eine komfortable Benutzung dieses Rechners und seines Bildschirms zur grafischen Darstellung von in Datenfeldern gespeicherten Zahlen ermöglicht. Es handelt sich dabei um eine Alternative zu den zur Zeit erhältlichen Grafikerweiterungen. Das vorgestellte Programm kommt jedoch ohne zusätzliche Hardware aus und beschränkt sich auf die Benutzung der Grafik-Symbole des CBM-3032.

Will man mit dem CBM-3032-Computer große Datenmengen (1000 und mehr Worte) verarbeiten, so stellt man schnell fest, daß die Basic-Programmsprache des Rechners sehr langsam arbeitet. Will man 1000 Daten von einem externen Gerät einlesen, so kann der Vorgang gut 10 Sekunden oder länger dauern. Wenn noch zusätzliche Rechenoperationen notwendig sind, wird das Warten lästig, und eine Programmentwicklung in Maschinensprache bietet sich an. Dabei sollte die Datenübergabe vom Maschinenprogramm zum Basic-Programm schnell ablaufen, einfach zu programmieren sein und keinen zusätzlichen RAM-Speicherbereich benötigen. Im vorliegenden Fall wird ein Konzept beschrieben, in dem das Maschinenprogramm die Daten direkt aus einem vorher vom Basic eingerichteten Array holt, um sie weiterzuverarbeiten. Beschränkt man sich nun auf einen 16-Bit-Zahlenvorrat, der für die meisten Anwendungen voll ausreicht (z. B. A/D- und D/A-Wandler), so lassen sich Integer-Arrays benutzen, die vom Basic eingerichtet werden können und tatsächlich nur 16 Bits Speicherplatz pro Datenwort benötigen. Die binäre Zahlendarstellung (Festkommaformat mit Zweierkomplement) ist leicht an A/D- und D/A-Wandler anzupassen. Es lassen sich nun Maschinenunterprogramme schreiben, die Daten von Transientenrecordern, Sampling-Oszilloskopen, Meßwertaufnehmern usw. in das Array einlesen, die andererseits die Daten auf einem Plotter oder einem Oszilloskop wieder ausgeben. Sofern keine Fließkomma-Arithmetik benötigt wird, lassen sich auch einfache Maschinenprogramme schreiben, die die Daten weiterverarbeiten, z. B. Minimum- und Maximumsuche, Datenglättung, digitale Filterung oder Mittelwertbildung.

Funktion

Das Maschinenprogramm ermöglicht es, sich schnell einen grafischen Überblick über Daten zu verschaffen, die sich in einem Integer-Array befinden. Dieses Array muß vorher in Basic als erstes Array dimensioniert werden (es muß das erste Array in der Array-Tabelle sein). Der Bildschirm wird mit den CBM-Grafiksymbolen unterteilt in ein Raster von 80 × 50 Punkten, die unabhängig voneinander ansprechbar sind [1]. Das komplette Programm ist in Hex-Code aus *Bild 1* zu ersehen. Es liegt im 32-KByte-RAM-Bereich, direkt unter dem Bildschirm-RAM, und beansprucht 1,25 KByte Speicherplatz. Es bietet folgende Möglichkeiten:

- Darstellung beliebiger Ausschnitte aus den Daten des Integer-Arrays mit vollem Zahlenbereich zwischen −32 767 und +32 767.
- Automatische Ausschnittsanpassung an die Größe des Arrays und an die im Array vorhandenen Daten.
- Übergabe der Parameter für Größe und Lage des Ausschnitts, so daß im aufrufenden Basic-Programm eine Skalierung möglich ist.
- Der Ausschnitt kann per Tastendruck beliebig in seiner Lage verschoben werden. Seine Größe ist jeweils um den Faktor 2 veränderbar.
- Wenn mehr als 80 Daten dargestellt werden sollen, werden benachbarte Daten zusammengefaßt und deren Mittelwert angezeigt (digitale Glättung). Dies ist besonders nützlich bei monotonen Datenfolgen mit Störanteil. Der Array-Inhalt wird nicht verändert.
- Unabhängig von der Bildschirmdarstellung werden von einem vorhandenen Integer-Array Minimal- und Maximalwert bestimmt.
- Jeder der 4000 Punkte läßt sich nach Übergabe der x/y-Koordinaten unabhängig ansprechen (zusätzlich zur Array-Darstellung).
- Zeichnung eines Koordinatenkreuzes auf dem Bildschirm (Ursprung unten links).

Tabelle 1 zeigt die vom Programm belegten Adressen im Rechner. Die Lage des Ausschnitts wird durch die vier in *Bild 2* dargestellten Größen gewählt. Die Wahl des Ausschnitts erinnert an die Darstellung auf einem Oszilloskop, die mit Positionsreglern, Verstärkungsfaktor und Zeitablenkung variiert werden kann.
Die Größe des verwendeten Arrays hängt im wesentlichen vom verfügbaren RAM-Bereich und von der Größe des steuernden Basic-Programms ab. Das Programm kann Arrays mit mehr als 10 000 Daten verarbeiten (programmtechnisch begrenzt auf 10 239, bei größeren Arrays erfolgt Error-Meldung).
So kann man trotz geringer Auflösung des Bildschirms einen Überblick über seine Daten erhalten, aber sich je nach Wunsch auch Einzelheiten ansehen. Die zur Ausschnittsvergrößerung benötigten arithmetischen Operationen wurden bewußt einfach programmiert, um die Rechenzeiten so niedrig wie möglich zu halten. So werden die Daten horizontal nur durch ganzzahlige Potenzen von 2 dividiert, ebenso werden horizontal jeweils ein, zwei, vier, acht usw. benachbarte Daten zusammengefaßt. Diese Einschränkung erwies sich in der Praxis aber als kaum nachteilig.

Tabelle 1:
Benutzte Adressen im Rechner

von	bis	Name/Bedeutung
002C	002D	Zeiger auf Array-Beginn
0054	005E	Hilfsspeicher
0061	0063	Hilfsspeicher
0066	006B	Hilfsspeicher
0097		Letzte Taste
0098		Shift-Taste
033D	034F	Hilfsspeicher
7B00	7FFF	Programm
8000	84E8	Bildschirmspeicher
CC9A		Error-Meldung
E229		Bildschirm löschen

Hex-Dump des vollständigen Programms

```
7B00 20 75 7E B8 50 03 20 5B
7B08 7E 20 1F 7B A5 97 A6 98
7B10 F0 03 20 D9 7E 20 98 7E
7B18 20 5C 7D 20 16 7E 60 20
7B20 83 7C A0 00 B1 2C 10 05
7B28 B1 68 10 01 60 4C 9A CC
7B30 20 1F 7B A2 00 8E 4A 03
7B38 CA 8E 4C 03 A2 80 8E 4B
7B40 03 CA 8E 4D 03 A0 00 B1
7B48 66 AA C8 B1 66 A8 8A 38
7B50 ED 4B 03 50 01 6A F0 04
7B58 30 0F 10 07 CC 4A 03 F0
7B60 08 90 06 8C 4A 03 8E 4B
7B68 03 8A 38 ED 4D 03 50 01
7B70 6A F0 04 10 0D 30 05 CC
7B78 4C 03 B0 06 8C 4C 03 8E
7B80 4D 03 20 A6 7C D0 BE A0
7B88 07 AD 4B 03 91 68 C8 AD
7B90 4A 03 91 68 C8 AD 4D 03
7B98 91 68 C8 AD 4C 03 91 68
7BA0 60 20 1F 7B 38 AD 4A 03
7BA8 ED 4C 03 85 66 AD 4B 03
7BB0 ED 4D 03 85 67 A0 00 A5
7BB8 67 F0 08 C8 46 67 66 66
7BC0 18 90 F4 A5 66 C9 32 90
7BC8 04 C8 4A D0 F8 8C 3F 03
7BD0 C0 0B D0 0C A9 7F 8D 4F
7BD8 03 A9 FF 8D 4E 03 D0 3B
7BE0 18 AD 4A 03 6D 4C 03 8D
7BE8 4E 03 AD 4B 03 6D 4D 03
7BF0 70 04 18 10 01 38 6A 8D
7BF8 4F 03 6E 4E 03 A9 00 85
7C00 6B A9 19 85 6A 20 EB 7F
7C08 38 AD 4E 03 E5 6A 8D 4E
7C10 03 AD 4F 03 E5 6B 8D 4F
7C18 03 70 B9 A9 01 8D 3D 03
7C20 A0 05 B1 2C 85 57 C8 B1
7C28 2C 38 E9 01 B0 02 C6 57
7C30 EE 4E 03 D0 03 EE 4F 03
7C38 A6 57 F0 0B 46 57 6A 0E
7C40 3D 03 90 F4 4C 9A CC C9
7C48 50 90 09 4A 0E 3D 03 90
7C50 F6 4C 9A CC 60 A9 5B 8D
7C58 C0 83 A9 40 A2 28 9D C0
7C60 83 CA D0 FA A9 C0 85 6A
7C68 A9 83 85 6B A0 00 A2 18
7C70 A5 6A 38 E9 28 85 6A B0
7C78 02 C6 6B A9 5D 91 6A CA
7C80 D0 EE 60 A5 2D 85 67 A0
7C88 03 18 71 2C 85 69 A5 2C
7C90 18 69 07 85 66 90 02 E6
7C98 67 A5 2C 18 88 71 2C 85
7CA0 68 90 02 E6 69 60 18 A5
7CA8 66 69 02 85 66 90 02 E6
7CB0 67 C5 68 D0 04 A5 67 C5
7CB8 69 60 20 48 7D 20 1F 7B
7CC0 A0 0B B1 68 D0 12 C8 B1
7CC8 68 85 5B C8 B1 68 D0 08
7CD0 C8 B1 68 85 5C 20 D9 7C
7CD8 60 A9 E8 85 6A A9 83 85
7CE0 6B A5 5B C9 50 B0 50 4A
7CE8 AA A5 5C C9 32 B0 48 4A
7CF0 A8 C8 A5 6A 38 E9 28 85
7CF8 6A B0 02 C6 6B 88 D0 F2
7D00 A5 6A 8D 41 03 8D 45 03
7D08 A5 6B 8D 42 03 8D 46 03
7D10 20 40 03 A2 0F DD 38 7D
7D18 F0 03 CA D0 F8 A9 01 46
7D20 5C 90 01 0A 46 5B 90 02
7D28 0A 0A 86 5C 05 5C AA BD
7D30 38 7D A6 5B 20 44 03 60
7D38 20 7B 7E 61 6C 62 7F FC
7D40 7C FF E2 EC E1 FE FB A0
7D48 A2 07 BD 54 7D 9D 40 03
7D50 CA 10 F7 60 BD 00 00 60
7D58 9D 00 00 60 20 29 E2 20
7D60 55 7C A5 5D 85 62 A5 5E
7D68 85 63 A9 FF 85 61 AE 3E
7D70 03 A9 00 85 54 A0 00 B1
7D78 62 85 55 10 02 C6 54 C8
7D80 B1 62 85 56 E0 01 F0 22
7D88 20 FE 7D 18 A0 01 B1 62
7D90 65 56 85 56 88 84 5A B1
7D98 62 10 02 C6 5A 65 55 85
7DA0 55 A5 5A 65 54 85 54 CA
7DA8 10 DA AD 3E 03 F0 0D C9
7DB0 01 F0 09 46 54 66 55 66
7DB8 56 4A 90 F3 E6 61 A5 61
7DC0 C9 50 90 01 60 85 5B A9
7DC8 00 85 5A 38 A5 56 ED 4E
7DD0 03 85 56 A5 55 ED 4F 03
7DD8 85 55 70 03 30 1A 18 B0
7DE0 17 AE 3F 03 F0 07 46 55
7DE8 66 56 CA 10 F7 A5 55 D0
7DF0 07 A5 56 85 5C 20 D9 7C
7DF8 20 FE 7D 4C 6E 7D 18 A5
7E00 62 69 02 85 62 90 02 E6
7E08 63 C5 68 D0 08 A5 63 C5
7E10 69 D0 02 68 68 60 38 A5
7E18 5D E5 66 8D 48 03 A5 5E
7E20 E5 67 4A 8D 49 03 6E 48
7E28 03 A0 0F A9 00 91 68 C8
7E30 AD 3E 03 91 68 C8 20 E3
7E38 7F A5 6B 91 68 C8 A5 6A
7E40 91 68 C8 AD 49 03 91 68
7E48 C8 AD 48 03 91 68 C8 AD
7E50 4F 03 91 68 C8 AD 4E 03
7E58 91 68 60 20 30 7B 20 83
7E60 7C AD 48 03 0A 2E 49 03
7E68 18 65 66 85 5D AD 49 03
7E70 65 67 85 5E 60 20 30 7B
7E78 20 A1 7B 20 83 7C A5 66
7E80 85 5D A5 67 85 5E AD 3D
7E88 03 8D 3E 03 A9 00 8D 48
7E90 03 8D 49 03 20 48 7D 60
7E98 C9 2A D0 04 20 F7 7E 60
7EA0 C9 29 D0 04 20 26 7F 60
7EA8 C9 1A D0 04 20 46 7F 60
7EB0 C9 19 D0 04 20 62 7F 60
7EB8 C9 02 D0 04 20 81 7F 60
7EC0 C9 22 D0 04 20 9D 7F 60
7EC8 C9 32 D0 04 20 D1 7F 60
7ED0 C9 12 D0 04 20 B9 7F 60
7ED8 60 C9 2A F0 0D C9 29 F0
7EE0 09 C9 32 F0 05 C9 12 F0
7EE8 01 60 20 98 7E 20 5C 7D
7EF0 A5 97 A6 98 D0 E3 60 AE
7EF8 3E 03 A5 69 85 59 38 A5
7F00 68 E9 02 85 58 B0 02 C6
7F08 59 A5 5E C5 59 D0 06 A5
7F10 5D C5 58 F0 10 18 A5 5D
7F18 69 02 85 5D 90 02 E6 5E
7F20 CA F0 02 10 E4 60 AE 3E
7F28 03 A5 5E C5 67 D0 06 A5
7F30 5D C5 66 F0 10 A5 5D 38
7F38 E9 02 85 5D B0 02 C6 5E
7F40 CA F0 02 10 E4 60 20 4D
7F48 7F 8D 3E 03 60 AD 3E 03
7F50 D0 03 A9 01 60 0A F0 06
7F58 CD 3D 03 B0 01 60 AD 3D
7F60 03 60 20 69 7F 8D 3E 03
7F68 60 AD 3E 03 CD 3D 03 90
7F70 07 AD 3D 03 4A F0 05 60
7F78 C9 02 B0 03 A9 01 60 4A
7F80 60 20 88 7F 8D 3F 03 60
7F88 AD 3F 03 C9 00 B0 03 A9
7F90 01 60 C9 0B 90 03 A9 0B
7F98 60 AA E8 8A 60 20 A4 7F
7FA0 8D 3F 03 60 AD 3F 03 C9
7FA8 02 B0 03 A9 00 60 C9 0B
7FB0 90 03 A9 0A 60 AA CA 8A
7FB8 60 20 E3 7F 18 AD 4E 03
7FC0 65 6A AA AD 4F 03 65 6B
7FC8 70 06 8E 4E 03 8D 4F 03
7FD0 60 20 E3 7F 38 AD 4E 03
7FD8 E5 6A AA AD 4F 03 E5 6B
7FE0 50 E8 60 A9 01 85 6A A9
7FE8 00 85 6B AE 3F 03 E0 0C
7FF0 90 01 60 E0 01 B0 01 60
7FF8 06 6A 26 6B CA D0 F9 60
```

einen Rasterpunkt in die in *Bild 3* angedeutete Richtung. So ist auch bei großen Datenmengen die Schrittweite dem Ausschnitt angepaßt. Wird während einer Verschiebung die Shift-Taste gedrückt, so wird eine beschleunigte Verschiebung möglich. Die Tasten . (bei Tastenversion 2 die 0), 1, 3 und 5 verändern die Ausschnittsgröße. Die maximale Auflösung ist dann erreicht, wenn ein LSB pro Zeile und ein Datenwort pro Spalte angezeigt werden.

Der Ablauf des Programmes ist in *Bild 4* dargestellt. Es sind zwei Einsprungstellen möglich. Nach SYS31488 wird der Ausschnitt automatisch angepaßt, mit SYS31494 ist diese Automatik abgeschaltet, es bleiben die Ausschnittsparameter vom vorherigen Mal erhalten und können mit den Tasten verändert werden. Man sollte am Anfang sinnvollerweise immer mit der Automatik beginnen. Nach einmaligem Zeichnen erfolgt der Rücksprung ins Basic, dort ist dann eine Skalierung möglich. Die Parameter werden in einem zweiten Hilfs-Integer-Array übergeben (zweites Array in der Array-Tabelle) und sind folgendermaßen festgelegt (das Array heißt hier x%):

x%(0) Maximum des 1. Datenarrays (Integer-Array!)
x%(1) Minimum des 1. Datenarrays (Integer-Array!)

Bild 2. Lage des Bildschirmausschnittes

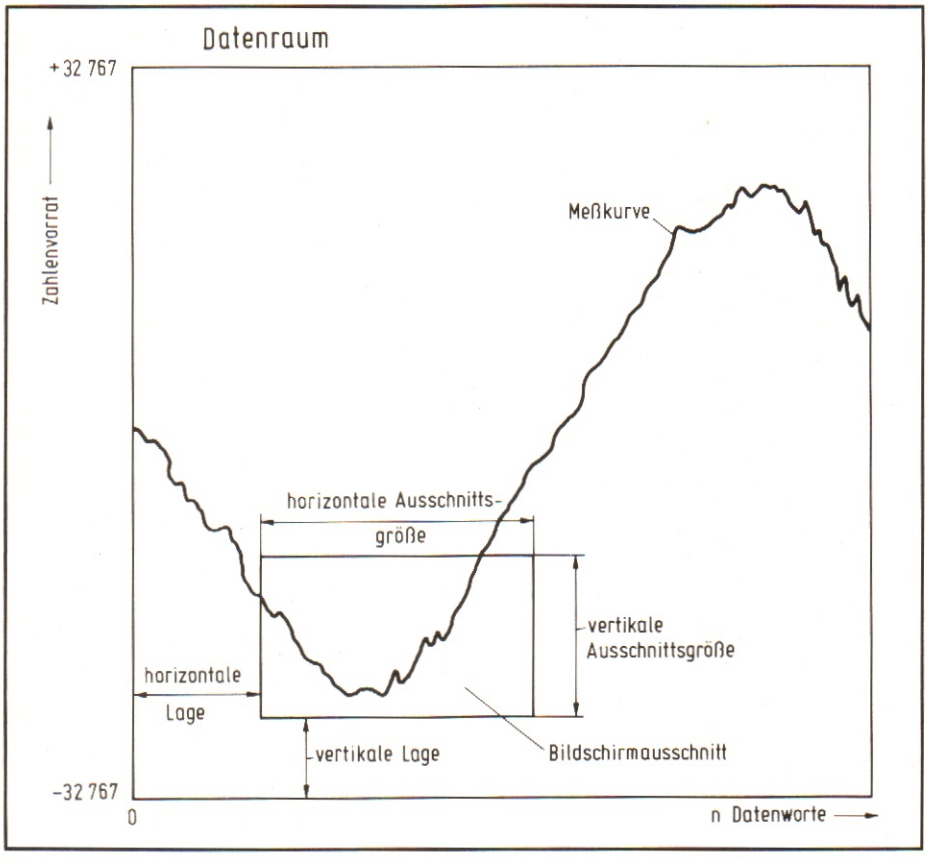
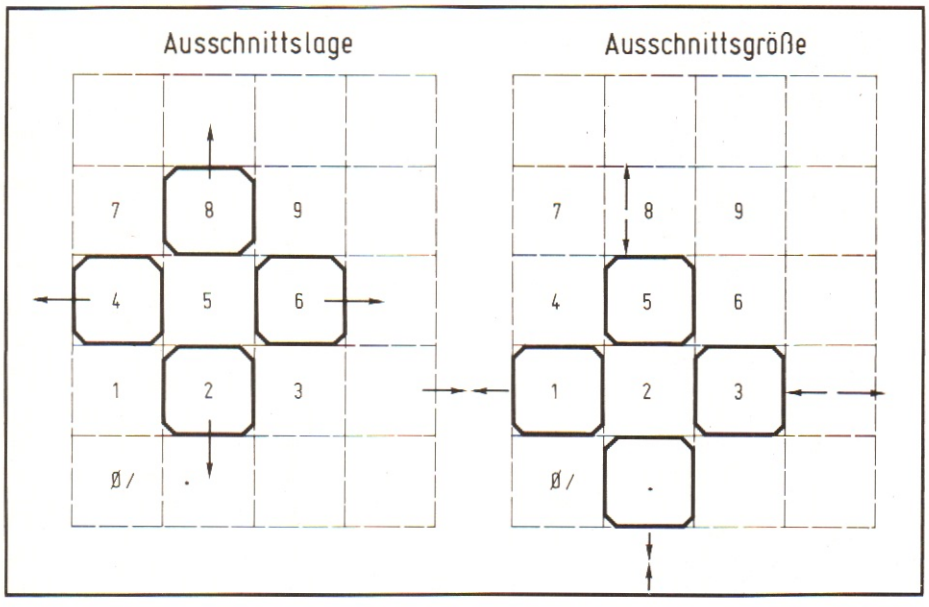


Bild 3. Belegung der Tasten zur Änderung des Bildausschnittes



x%(2) x-Koordinate für Unterprogramm zum Zeichnen eines Punktes
x%(3) y-Koordinate dto.
x%(4) Horizontale Ausschnittsgröße (1, 2, 4, 8,...128 Daten pro Spalte)
x%(5) Vertikale Ausschnittsgröße (1, 2, 4, 8, ...2048 LSB pro Zeile)
x%(6) Horizontale Position der linken Spalte
x%(7) Vertikale Position der untersten Zeile

Außer den beiden Koordinaten x%(2) und x%(3) werden die Parameter vom Maschinenprogramm bestimmt und lassen sich vom Basic nicht verändern. Die beiden Parameter für die x/y-Darstellung eines Punktes können gefahrlos alle erlaubten

Tabelle 2: Die verschiedenen Tastatur… und die zugehörigen Adressen im Prog…

Taste	Tast.-1 Dez	Tast.-1 Hex	Tast.-2 Dez	Tast.-2 Hex	Adr. Hex
0			20	14	7EB9
.	2	2			7EB9
1	26	1A	9	9	7EA9
2	18	12	17	11	7ED1, 7E…
3	25	19	25	19	7EB1
4	42	2A	33	21	7E99, 7E…
5	34	22	57	39	7EC1
6	41	29	49	31	7EA1, 7E…
8	50	32	76	4C	7EC9, 7E…

Tabelle 3: Programmlaufzeiten

Anzahl der Daten im Array	Darstellung auf dem Bildschirm
80	100 ms
1280	340 ms
10239	2020 ms

Integer-Werte annehmen, es wird nur dann ein Punkt gezeichnet, wenn x zwischen 0 und 79 und y zwischen 0 und … liegt. Dieses Programm entspricht in … Funktion etwa dem Programm in [1]. Startadresse ist dez(31930) oder hex(… BA). Weitere benutzbare Unterprogramme sind:

dez(31536), hex(7B30):
Es berechnet vom 1. Integer-Array den Minimal- und den Maximalwert (in x%(0) und x%(1)).
dez(31829), hex(7C55):
Es zeichnet das Koordinatenkreuz mit Ursprung unten links.

Falls im Zeichenprogramm das Zeichnen des Koordinatenkreuzes nicht erwünscht ist, kann der entsprechende Aufruf auf hex(7D5F) durch NOPs ersetzt werden. Auch das Löschen des Bildschirms vor jedem Zeichnen kann mit Änderung des Aufrufs auf hex(7D5C) unterdrückt werden, z. B. wenn eine Balkendarstellung erwünscht wird.
Error-Meldungen treten auf, wenn die ersten beiden Arrays keine Integer-Arrays sind oder wenn das erste Array zu groß ist (mehr als 10 239 Elemente). Das Programm ist nicht verschiebbar, sei denn, man verändert 46 Befehle (… JSRs, 1 JMP, 1 CMP,x, 2 LDA,x), die … mit einem Disassembler leicht ermittelt lassen.
Die Laufzeiten der Programme sind … ermittelt worden und sind im wesentlichen abhängig von der Anzahl der Daten des Arrays. In der *Tabelle 3* sind ein…

...n Tastat... ...en im Pro...

Adr. Hex	
7EB9	
7EB9	
7EA9	
7ED1	
7EB1	
7E99	
7EC1	
7EA1	
7EC9	

...eiten

...rstellung ... ...m Bildsch...

...00 ms
...340 ms
...020 ms

...n, es wird ... ...net, wenn ... ...wischen 0 ... entsprich... ...ogramm i... ...930) oder ... ...re Unterp...

...: ... Integer-... den Maxim... ...1)). ...: ...ordinaten... ...n links.

...ramm das ... ...reuzes ni... ...entsprech... durch NO...e zu ersehen. Bei 1280 Daten ...as Lösche... vollständiges Bild in ⅓ Sekun... ...n Zeichne...net, so daß ein fast verzöge... ...frufs auf ...es Arbeiten möglich ist. ...kt werden...nstrationsprogramm in *Bild 5* ...stellung e... der für das Unterprogramm ...en Dimensionierung DIM ...ten auf, w...(n), wobei die Namen beliebig ... keine In...10240 und n>=7 sein muß. ...n das ers... wird einfach mit Geraden-Da... 10 239 El... deren Anfangs- und Endwert ...cht versch... werden. Ab Zeile 270 folgt ...dert 46 Be...liche Darstellung, in Zeile 280 ...x, 2 LDA,x...Maschinenprogramm, in Zeile ...bler leicht...kalierungsunterprogramm auf... ...Nach einer Warteschleife (Zeile ...rogramme ...ur verlassen wird, wenn eine ...d sind im ...rückt wird, kann dann der Aus... ...der Anzahl ...t den entsprechenden Tasten ...abelle 3 si... werden.

...Das Flußdiagramm zeigt die beiden möglichen Einsprungstellen bei Aufruf aus dem ...programm

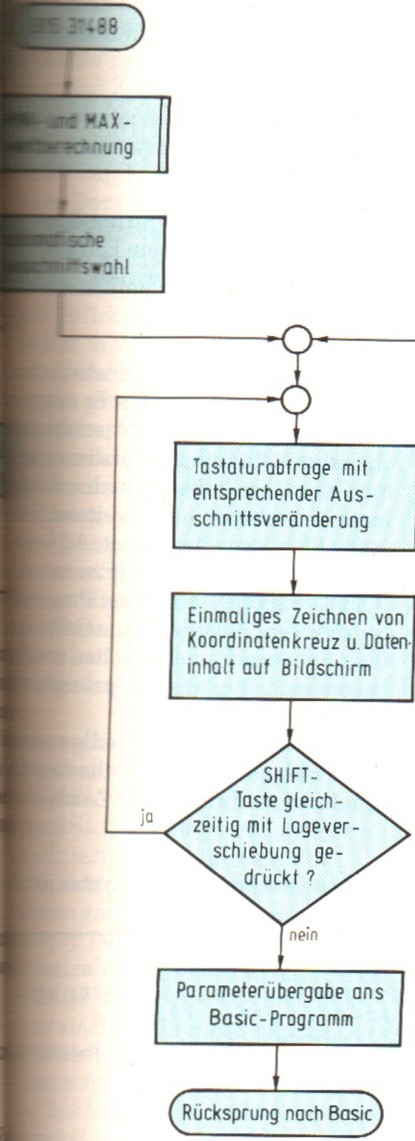


Bild 5. Demonstrationsprogramm

```
100 REM HAUPTPROGRAMM
110 DIMA%(1023),X%(7)
120 PRINT"▮GERADEN-DEMONSTRATION
130 PRINT"▮NEUE DIMENSION VON A% ▮J▮A  ▮N▮EIN
140 GETIN$:IFIN$=""THEN140
150 IFIN$="J"THENRUN170
160 IFIN$<>"N"THEN140
170 PRINT"▮ANZAHL=   ";K
180 INPUT"▮▮▮▮▮▮▮▮▮▮";K
190 IFIN$<>"N"THENDIMA%(K),X%(7)
200 PRINT"▮ANFANG=  ";A
210 INPUT"▮▮▮▮▮▮▮▮▮▮";A
220 PRINT"▮ENDE   =   ";E
230 INPUT"▮▮▮▮▮▮▮▮▮▮";E
240 PRINT"▮BITTE"INT(.028*K)"SEC WARTEN▮"
250 FORI=0TOK:A%(I)=A+(E-A)/K*I
260 PRINT"I="I"▮":NEXT
270 REM DARSTELLUNG DES ARRAYS MIT SKALIERUNG
280 SYS31488:REM ZEICHNE A% MIT AUTOMATIK
290 GOSUB350:REM SKALIERUNG
300 B=PEEK(151):IFB=255THEN300
310 GETIN$:IFIN$="E"THEN120
320 SYS31494:REM ZEICHNE A% OHNE AUTOMATIK
330 GOTO290
340 REM SKALA
350 PRINT"▮↑"X%(5)*50-1+X%(7),"MAX=";X%(0);"MIN=";X%(1)
360 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮"X%(7),X%(6)"BIS";
370 PRINTTAB(32)X%(6)+X%(4)*80-1"▮▮"
380 RETURN
READY.
```

Das Unterprogramm für die Skalierung beginnt mit Zeile 350 und macht die Verwendung der Parameterübergabe deutlich. Je nach Aufgabe lassen sich hier auch Entnormierungen durchführen (Angabe von Strom, Spannung usw.).

Mit dem Programm in *Bild 6* können von der Floppy-Disk nacheinander das Maschinenprogramm und das Basic-Hauptprogramm geladen werden. Zuerst wird der Tastaturpuffer mit den einzelnen Befehlen gefüllt. Nach Programmende werden dann diese Befehle so ausgeführt, als ob man sie über die Tastatur direkt eingegeben hätte. Das CHR$(13)-Zeichen ersetzt dabei die Return-Taste. So läßt sich dieses Programm als Urlader verwenden. Die Anzahl der Befehle ist begrenzt durch den Tastaturpuffer, der aber direkt vor dem Puffer für die Kassettenrecorder steht und diesen mitbenutzen kann (aber max. 255 Zeichen). Der Befehl in Zeile 60 setzt den Zeiger auf die höchste RAM-Adresse so zurück, daß das Maschinenprogramm nicht zerstört werden kann. Dies ist in jedem Fall vor Starten des Basic-Programms notwendig.

Literatur

[1] *Kisker, Erhard:* Höhere Auflösung bei Pet-Grafiken. FUNKSCHAU 1980, Heft 26.

[2] *Handle, Franz:* Zahlendarstellung im PET. FUNKSCHAU-Sonderheft „Hobbycomputer 2“.

Bild 6. Dieses Programm kann als Urlader für Maschinen- und Basic-Programm verwendet werden

```
10 REM LOAD MASCHINENPROGRAMM
20 C$=CHR$(13):A$=CHR$(34):S$="LOAD"+A$+"0:HEX7B00-7FFF"+A$+",8"+C$
30 S$=S$+"NEW"+C$+"LOAD"+A$+"0:HAUPTPROGRAMM"+A$+",8"+C$+"RUN"+C$
40 FORI=1TOLEN(S$):POKE622+I,ASC(MID$(S$,I,1)):NEXT
50 POKE158,LEN(S$):REM FUELLE TASTATURPUFFER
60 POKE53,123:POKE52,0:REM SETZE HOECHSTE RAM-ADRESSE ZURUECK
70 END
READY.
```

Günter Egle

IEC-Routinen für den 6502

Mit den im folgenden vorgestellten Routinen läßt sich ein 6502-System als Controller für den IEC-Bus einsetzen. Als Beispiel wird der Anschluß eines Druckers mit IEC-Bus-Schnittstelle (z. B. CBM 3022) an den AIM-65 beschrieben.

Das Handshake-Verhalten des IEC-Busses wurde schon in mehreren Artikeln beschrieben [1, 2], es wird deshalb in diesem Zusammenhang nicht mehr erläutert. Den hardwaremäßigen Anschluß des Busses an den VIA 6522 zeigt *Bild 1*. In *Bild 2* sind die Routinen wiedergegeben. Die Ports werden durch die beiden Unterprogramme ITALK (für Talker) und ILIST (für Listener) initialisiert. Den Datentransfer zwischen Bus und 6502-System übernehmen die Programme SBYTE (Zeichen ausgeben) und RBYTE (Zeichen empfangen). Das Unterprogramm SBYTE gibt das im Akkumulator stehende Datenbyte invertiert auf den Bus und wickelt den Handshake ab. Die EOI-Leitung wird entsprechend dem Inhalt des Overflow-Flags beeinflußt. (V = 1 ⇒ EOI = „High"; V = 0 ⇒ EOI = „Low".) Am Ende der Übertragung wird der Bus wieder neutralisiert, d. h. alle Leitungen werden auf High gesetzt. Kehrt das Unterprogramm mit gelöschtem Carryflag (C = 0) zum rufenden Programm zurück, so waren sowohl NDAC als auch NRFD „High". Dies bedeutet, daß sich kein Gerät am Bus befindet, und eine Datenübertragung somit nicht möglich ist.

Den Empfang eines Zeichens ermöglicht das Unterprogramm RBYTE. Das vom Bus geholte Datenbyte wird invertiert im Akkumulator übergeben. Das Overflow-Flag (V-Flag) repräsentiert wieder den Status der EOI-Leitung. Mit dem Programm ADROUT schließlich gibt man die Primär- und die Sekundäradresse aus. Die Parameterübergabe erfolgt durch den Akkumulator (Primäradresse) und das X-Register (Sekundäradresse). Das in *Bild 3* abgedruckte Programm demonstriert die Anwendung der Routinen. Es ermöglicht den Anschluß eines Druckers mit IEC-Schnittstelle an den AIM-65. Der User-Vektor UOUT ($1...) muß hierfür auf LIST gerichtet werden. Beim Aufruf des Programms durch den Monitor kann mit Hilfe des Carry-Flags die Initialisierung (C = 0) vom Datentransport (C = 1) unterschieden werden. Die Initialisierungsroutine INIT programmiert den VIA für die Funktion „Talker" (ITALK), fordert interaktiv die Primär- und die Sekundäradresse an und gibt sie auf den Bus. In der Datenphase liegt das auszugebende Zeichen auf dem Stack.

Wird zweimal in Folge ein „Carriage Return" ($0D) gesendet, so beendet das Programm die Ausgabe mit dem UNLISTEN-Kommando (3F , ATN = „Low").

Literatur

[1] IEC-Bus. Sonderheft Nr. 47. Franzis-Verlag, München.

[2] *Klein, R. D.*: EMUF bringt Strichcode-IEC-Bus. mc 1981, H. 3, S. 62...65.

Bild 1. Anschluß des IEC-Busses an den VIA 6522

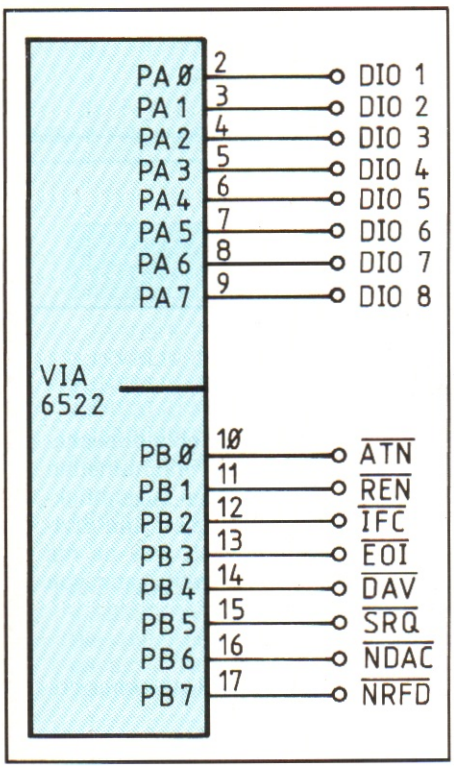


Bild 2. Die IEC-Bus-Routinen können auf jedes 6502-System übertragen werden, das über einen VIA-Baustein 6522 verfügt

```
          UORA  = DATENREGISTER  PORT A
          UORB  = DATENREGISTER  PORT B
          UDDRA = CONTROL-REGISTER PORT A
          UDDRB = CONTROL-REGISTER PORT B

; DAS UNTERPROGRAMM 'ITALK' INITIALISIERT
; DEN VIA 6522 FUER DIE FUNKTION TALKER
;
ITALK     LDA #$FF
          STA UORA          ; BUS NEUTRALISIEREN
          STA UDDRA         ; PORT A  OUTPUT
          STA UORB
          LDA #$1F
          STA UDDRB
          RTS

; DAS UNTERPROGRAMM 'ILIST' INITIALISIERT
; DEN VIA 6522 FUER DIE FUNKTION LISTENER
;
ILIST     LDA #0
          STA UDDRA         ; PORT A  INPUT
          LDA #$BF
          STA UORB
          LDA #$C7
          STA UDDRB
          RTS
```

▶

```
; DAS UNTERPROGRAMM 'ADROUT' GIBT DIE PRIMAER-
; UND DIE SEKUNDAER-ADRESSE AUS
; EINGANG:     A  ENTHAELT DIE PRIMAER-ADR.
;              X  ENTHAELT DIE SEKUNDAER-ADR.
; VERAENDERT: Y,FLAGS

ADROUT  LDY #$FE        ; ATN AUF LOW SETZEN
        STY UORB
        BIT VHIGH       ; EOI AUF HIGH SETZEN
        JSR SBYTE       ; PRIMAER-ADR AUSGEBEN
        BCC OUT         ; BRANCH BEI FEHLER
        TXA
        BIT VHIGH       ; SEKUNDAER-ADR AUS -
        JSR SBYTE       ; GEBEN
OUT     LDY #$FF        ; ATN AUF HIGH SETZEN
        STY UORB
VHIGH   RTS

; DAS UNTERPROGRAMM 'RBYTE' EMPFAENGT EIN BYTE
; AUSGANG:     A ENTHAELT DAS EMPF. ZEICHEN
;              V ENTHAELT DEN STATUS VON EOI
;                V=1 => EOI = 'HIGH'
; VERAENDERT: X , FLAGS

RBYTE   LDA UORB        ; AUF DAV='LOW' WARTEN
        AND #$10
        BNE RBYTE

        LDA #$3F        ; NRFD AUF LOW SETZEN
        STA UORB
        LDX UORA        ; DATEN UEBERNEHMEN
        LDA UORB
        AND #8          ; TEST EOI
        CLV
        BEQ EOI
        BIT VHIGH       ; EOI IST HIGH
EOI     LDA #$7F        ; NDAC AUF HIGH SETZEN
        STA UORB

DAVH    LDA UORB        ; AUF DAV='HIGH' WARTEN
        AND #$10
        BEQ DAVH
        LDA #$BF        ; NDAC = LOW
        STA UORB        ; NRFD = HIGH
        TXA
        EOR #$FF        ; DATEN INVERTIEREN
VHIGH   RTS

; DAS UNTERPROGRAMM GIBT EIN BYTE AUF DEN BUS
; EINGANG:     A  ENTHAELT DAS ZEICHEN
;              V  ENTHAELT DEN STATUS VON EOI
; AUSGANG:     C  GELOESCHT BEI FEHLER
; VERAENDERT: Y , FLAGS
;
SBYTE   EOR #$FF        ; DATEN INVERTIEREN
        TAY
        LDA UORB
        AND #$C0        ; NRFD & NDAC = HIGH ?
        CMP #$C0
        BEQ FEHLER      ; BRANCH, WENN JA

        STY UORA        ; BYTE AUF BUS GEBEN
        PHP
NRFD    BIT UORB        ; AUF NRFD='1' WARTEN
        BPL NRFD
        PLP
        LDA UORB
        BVS S1
        AND #$F7        ; EOI AUF LOW SETZEN
S1      AND #$EF        ; DAV AUF LOW SETZEN
        STA UORB

NDAC    BIT UORB        ; AUF NDAC='1' WARTEN
        BVC NDAC
        LDA UORB
        ORA #$18        ; DAV UND EOI = '1'
        STA UORB
        LDA #$FF        ; BUS NEUTRALISIEREN
        STA UORA
        SEC
        RTS

FEHLER  CLC
        RTS
```

Anschluß eines Druckers an den AIM-65

```
; ANSCHLUSS DES AIM 65 AN DEN IEC-BUS UEBER UOUT

        OUTFLG = $A413
        OUTPUT = $E97A
        RBYTE1 = $E3FD

        BCC INIT

        PLA             ; DATEN-BYTE HOLEN
        CMP #$0D        ; CR ?
        BNE WRITE
        CMP $EB         ; ZUM ZWEITEN MAL ?
        BEQ ENDE        ; BRANCH, WENN JA

WRITE   STA $EB
        BIT VHIGH       ; V='1' => EOI='1'
        JSR SBYTE       ; BYTE AUSGEBEN
        BCC ERROR
        RTS

ENDE    CLV             ; EOI = LOW
        JSR SBYTE
        JSR UNLIST
        BCC ERROR
        RTS             ; AUSGABE IST BEENDET

INIT    LDA #$20
        STA OUTFLAG     ; AUSGABE AUF D/P
        JSR ITALK       ; INIT BUS FUER TALKER
        LDY #0          ; PRIMAER-ADR ANFORDERN
        JSR OUTMSG
        JSR RBYTE1      ; UND EINLESEN
        PHA             ; ADR. RETTEN
        LDY #MSG1-MSG
        JSR OUTMSG      ; SEKUNDAER-ADR ANFORDERN
        JSR RBYTE1
        TAX
        PLA
        JSR ADROUT      ; ADR AUSGEBEN
        BCC ERROR
        LDA #$55        ; UOUT AUF 'U' SETZEN
        STA OUTFLG
        STA $EB
        RTS             ; INIT ENDE

ERROR   LDA #$20        ; OUTFLAG AUF D/P
        STA OUTFLG      ; UMSCHALTEN
        LDY #MSG2-MSG   ; FEHLER-MELDUNG AUS -
        JSR OUTMSG      ; GEBEN
        BRK             ; ZURUECK ZUM MONITOR

OUTMSG  LDY MSG,Y       ; UNTERPROGRAMM ZUR TEXT -
        PHA             ; AUSGABE AUF DISPLAY /
        JSR OUTPUT      ; PRINTER
        INY
        PLA             ; ENDE-KRITERIUM FUER DIE
        BPL OUTMSG      ; AUSGABE:  BIT 7 = '1'
        RTS

; DAS UP GIBT DAS UNLISTEN (3F) KOMMANDO AUF DEN BUS
;
UNLIST  LDA #$FE        ; ATN = 'LOW'
        STA UORB
        LDA #$3F        ; UNLISTEN CMD
        BIT VHIGH       ; EOI = 'HIGH'
        JSR SBYTE
        LDA #$FF        ; ATN AUF 'HIGH'
        STA UORB
VHIGH   RTS

MSG     .BYT $0D,'PRIMAER-ADR. ',$A0
MSG1    .BYT $0D,'SEKUNDAER-ADR. ',$A0
MSG2    .BYT $0D,'DEVICE NOT PRESENT',$A0
```