Herwig Feichtinger

Strichcode lesen mit AIM-65 und PC-100

Zwei verbreitete Computer fehlen uns in der Sammlung der Strichcode-Einlese-Software: Die intern baugleichen Geräte AIM-65 (Rockwell) und PC-100 (Siemens). Am folgenden Programm, das unabhängig von der ursprünglichen 6502-Version des Apple-II entwickelt wurde, sieht man auch gleich, wie unterschiedlich Programme für den gleichen Zweck ausfallen können.

Der Strichcode-Leser wird bei AIM-65 und PC-100 am Port PA5 angeschlossen (*Bild 1*), der am Application Connector herausgeführt ist. Das Programm (*Bild 2*) verwendet den Timer T1 desselben VIA-Bausteins; er ist nach einem Reset stets im gewünschten One-Shot-Modus, so daß keine weitere Initialisierung der Steuerregister nötig ist. Die Software testet ausgiebig, ob es sich wirklich um Strichcode handelt, z. B. durch wiederholte Zeitabfragen. Dabei wird zur Übergabe einer Fehlermeldung aus Unterprogramm-Ebenen das Carry-Flag der CPU verwendet, außer bei TST, wo es anzeigt, daß sich der Leser auf einem schwarzen Strich befindet. Wird eine Zeile der Länge Null gelesen (letzte Zeile), so erfolgt ein Rücksprung zum Monitor über RTS. Dies setzt den Programmstart mit einer der Tasten F1...F3 voraus.

Die Bedienung des Programms

Zunächst geht man mit Taste E in den Texteditor und wählt die gewünschte Anfangsadresse, ab der die Zeichen abgespeichert werden sollen; bei Basic-Programmen gewöhnlich 0200, bei Maschinenprogrammen die Startadresse. Als Endadresse (der Editor fragt automatisch danach) sollte man stets eine Adresse unter 0F20 eingeben. (Die Einhaltung dieser Endadresse wird allerdings beim Einlesen des Strichcode nicht geprüft.) Ist die Initialisierung des Editors erfolgt, so geht man z. B. mit Escape in das Monitorprogramm und beginnt das Strichcode-Lesen nach Druck auf F1. Sobald der Leser auf weißem Papier liegt, erscheint im Display die Nummer der jeweils einzulesenden Zeile (zuerst 01). Ein Fehler wird mit ERROR quittiert, und der Zeilenzähler wird dann nicht inkrementiert. Das Programm paßt sich in einem sehr großen Bereich der Lesegeschwindigkeit an, und es ist belanglos, ob man in einer oder in zehn Sekunden über eine Zeile fährt. Ein goldenes Mittelmaß für erste Versuche ist aber eine Zeit von etwa drei Sekunden für eine Zeile.
Nach dem Einlesen der letzten Zeile erscheint links im Display wieder der Monitor-Prompt, und man kann mit T in den Editor zurück, um nachzusehen, was eigentlich geladen wurde.
Sollte in der ersten Zeile nur Merkwürdiges stehen, sollte man mit dem M-Befehl das erste Textbyte (gewöhnlich bei 0200) untersuchen. Ist es hex 0D, sollte man es mit dem „/"-Befehl durch hex 20 ersetzen.

Übernahme in den Basic-Interpreter

Hat man eventuell nötige Anpassungen des Programms mit dem C-Befehl des Editors durchgeführt, so kann man den Programmtext (wenn es sich um ein Basic-Programm handelt) nun auf Kassette abspeichern; man sollte sich vorher aber vergewissern, ob der Gap-Wert in Zelle A409 mindestens hex 18 ist. Ferner ist es nötig, als letzte Basic-Programmzeile ein CTRL-Z einzugeben; es ist im Display unsichtbar. Dann initialisiert man Basic und lädt den Test von der Kassette per LOAD in den Basic-Interpreter. Dieses Verfahren mag auf den ersten Blick umständlich erscheinen, hat aber den großen Vorteil, daß man vor der Übernahme in Basic die komfortablen Editiermöglichkeiten des Texteditors in Anspruch nehmen kann.

In Kürze wird es bei mc zu interessanten Programmen auf Anforderung Strichcode-Listings geben.

Literatur

[1] ROM und RAM in AIM-65 und PC-100. Franzis-Sonderheft Nr. 33, „Mikrocomputer-Anwendungen".
[2] Apple-II liest Strichcode. mc 1981, Heft 1.

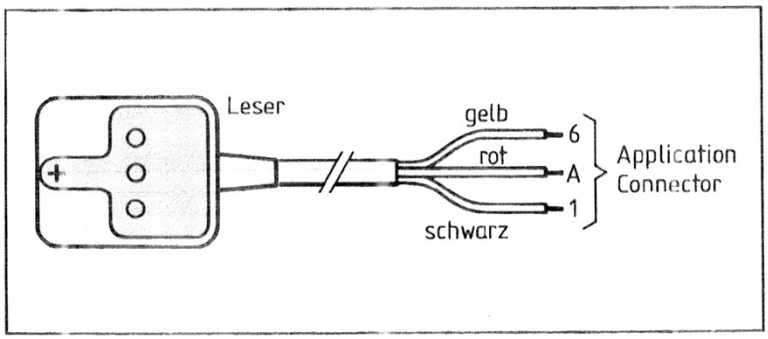
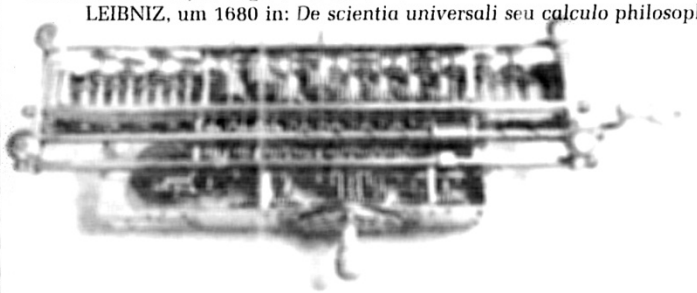


Bild 1. Anschluß des Strichcodelesers an AIM-65 oder PC-100

Bild 2. Assemblerlisting der Software zum Strichcode-Lesen

```
0FFC          PASS 1
0FFC          PASS 2
0000
0000          ;BARCODE READER
0000          ;MC/FE/11-1981/AIM
0000 PA             =$A00F
0000 TIMH           =$A005
0000 IFR            =$A00D
0000 NUMA           =$EA46
0000 CKER           =$E394
0000 CRLF           =$E9F0
0000 DISP           =$E9BC
0000 BUF            =$116
0000                *=$E1
00E1
00E1          ;TEXT EDITOR
00E1 ADL            *=*+1
00E2 ADH            *=*+1
00E3 TPL            *=*+1
00E4 TPH            *=*+1
00E5                *=$E7
00E7 CHR            *=*+1
00E8 REF            *=*+1
00E9 LEN            *=*+1
00EA CHKL           *=*+1
00EB CHKH           *=*+1
00EC CNT            *=*+1
00ED                *=$10C
010C          4C200F JMP INIT
010F                *=$F20
0F20
0F20          ;INIT FOR 1ST LINE
0F20 INIT     A5E3   LDA TPL
0F22          85E1   STA ADL
0F24          A5E4   LDA TPH
0F26          85E2   STA ADH
0F28          A901   LDA £1
0F2A          85EC   STA CNT
0F2C          ;WAIT FOR WHITE
0F2C WWH      A9FF   LDA £255
0F2E          8D05A0 STA TIMH
0F31 WWH1     20B00F JSR TST
0F34          B0F6   BCS WWH
0F36          2C0DA0 BIT IFR
0F39          50F6   BVC WWH1
0F3B          ;SHOW LINE NR.
0F3B          20F0E9 JSR CRLF
0F3E          A5EC   LDA CNT
0F40          2046EA JSR NUMA
0F43
0F43          ;WAIT TILL BLACK
0F43 SYN      20B00F JSR TST
0F46          90FB   BCC SYN
0F48          ;CHECK 2 SYNCHS
0F48          20CB0F JSR BLT3
0F4B          B0DF   BCS WWH
0F4D          4A     LSR A
0F4E          85E8   STA REF
0F50          20BC0F JSR BLT
0F53          B037   BCS ERR
0F55          C5E8   CMP REF
0F57          9034   BCC ERR
0F59          ;READ LENGTH BYTE
0F59          20E40F JSR RDB
0F5C          B02E   BCS ERR
0F5E          86EA   STX CHKL
0F60          86EB   STX CHKH
0F62          A8     TAY
0F63          F02D   BEQ RDY
0F65          C914   CMP £20
0F67          B023   BCS ERR
0F69          85E9   STA LEN
0F6B          ;READ DATA BYTES
0F6B RDD      20E00F JSR RDB
0F6E          B01C   BCS ERR
0F70          991601 STA BUF,Y
0F73          65EA   ADC CHKL
0F75          85EA   STA CHKL
0F77          9002   BCC RDD1
0F79          E6EB   INC CHKH
0F7B RDD1     88     DEY
0F7C          D0ED   BNE RDD
0F7E          ;GET CHECKSUM
0F7E          20E00F JSR RDB
0F81          C5EA   CMP CHKL
0F83          D007   BNE ERR
0F85          20E00F JSR RDB
0F88          C5EB   CMP CHKH
0F8A          F009   BEQ STO
0F8C
0F8C          ;DISPL.'ERROR'
0F8C ERR      2094E3 JSR CKER
0F8F          4C2C0F JMP WWH
0F92          ;BACK TO MONITOR
0F92 RDY      91E1   STA (ADL),Y
0F94          60     RTS
0F95          ;STORE IT TO MEMORY
0F95 STO      A6E9   LDX LEN
0F97 STO1     BD1601 LDA BUF,X
0F9A          91E1   STA (ADL),Y
0F9C          E6E1   INC ADL
0F9E          D002   BNE STO2
0FA0          E6E2   INC ADH
0FA2 STO2     CA     DEX
0FA3          D0F2   BNE STO1
0FA5          F8     SED
0FA6          A5EC   LDA CNT
0FA8          6900   ADC £0
0FAA          85EC   STA CNT
0FAC          D8     CLD
0FAD          4C2C0F JMP WWH
0FB0          ;
0FB0          ;SUBROUTINES
0FB0          ;
0FB0          ;TEST IF BLACK
0FB0 TST      48     PHA
0FB1          18     CLC
0FB2          AD0FA0 LDA PA
0FB5          2920   AND £$20
0FB7          F001   BEQ TST1
0FB9          38     SEC
0FBA TST1     68     PLA
0FBB          60     RTS
0FBC          ;WAIT FOR BLACK
0FBC BLT      A9FF   LDA £$FF
0FBE          8D05A0 STA TIMH
0FC1 BLT0     2C0DA0 BIT IFR
0FC4          7018   BVS BLT2
0FC6          20B00F JSR TST
0FC9          90F6   BCC BLT0
0FCB          ;TEST BLACK TIME
0FCB BLT3     A9FF   LDA £$FF
0FCD          8D05A0 STA TIMH
0FD0 BLT1     2C0DA0 BIT IFR
0FD3          7009   BVS BLT2
0FD5          20B00F JSR TST
0FD8          B0F6   BCS BLT1
0FDA          4D05A0 EOR TIMH
0FDD          60     RTS
0FDE BLT2     38     SEC
0FDF          60     RTS
0FE0          ;READ A BYTE,
0FE0          ;ADJUST SPEED
0FE0 RDB      A208   LDX £8
0FE2 RDB2     20BC0F JSR BLT
0FE5          B014   BCS RDB0
0FE7          C5E8   CMP REF
0FE9          66E7   ROR CHR
0FEB          C5E8   CMP REF
0FED          B003   BCS RDB1
0FEF          0A     ASL A
0FF0          65E8   ADC REF
0FF2 RDB1     4A     LSR A
0FF3          85E8   STA REF
0FF5          CA     DEX
0FF6          D0EA   BNE RDB2
0FF8          A5E7   LDA CHR
0FFA          18     CLC
0FFB RDB0     60     RTS
0FFC                 .END
0FFC           ERRORS= 0000
```

Spruch des Monats

„Es wird dann beim Auftreten von Streitfragen für zwei Philosophen nicht mehr Aufwand an wissenschaftlichem Gespräch erforderlich sein als für zwei Rechnerfachleute. Es wird genügen, Schreibzeug zur Hand zu nehmen, sich vor das Rechengerät zu setzen und zueinander (wenn es gefällt, in freundschaftlichem Ton) zu sagen: Laßt uns rechnen."

LEIBNIZ, um 1680 in: *De scientia universali seu calculo philosophico*