

Rolf-Dieter Klein

# EMUF bringt Strichcode zum IEC-Bus

Hier wird ein Programm vorgestellt, das es ermöglicht, mit dem Einplatinencomputer EMUF (s. Heft 2/1981) ein IEC-Bus-Interface z. B. für CBM-Rechner herzustellen, das den Anschluß eines mc-Barcode-Lesers erlaubt. Die IEC-Bus-Funktionen werden vom EMUF per Software simuliert, so daß praktisch keinerlei zusätzliche Hardware nötig ist.

Bild 3 zeigt das Listing unseres Programms. Die Barcode-Routine ist im wesentlichen dabei aus Heft 1/1981 übernommen. Nach dem Einschalten des EMUF erfolgt ein kurzes Piep-Signal als Zeichen, daß das Interface betriebsbereit ist. Danach wird in das IEC-Hauptprogramm gesprungen. Nach Empfang einer

Bild 1 zeigt die Anordnung. Der Rechner ist mit dem EMUF über den IEC-Bus verbunden, der Leser mit einem Anschluß des EMUF. Ein Lautsprecher gibt einen kurzen Piep-Ton ab, wenn eine Barcode-Zeile erfolgreich eingelesen wurde. Bild 2 zeigt die Anschlußbelegung am EMUF. Der IEC-Datenbus wird direkt mit dem Port PA verbunden. Die fünf Handshakesignale des IEC-Bus werden vom Port PB gewonnen. Dabei werden die restlichen Signale, die noch am IEC-Bus vorhanden sind, für unsere Zwecke nicht gebraucht. Im Bild sind zwei verschiedene Pinbelegungen für den IEC/IEEE-Stecker angegeben. Beim einen handelt es sich um die Version mit dem 25poligen Stecker; die 24polige Version ist am CBM-Rechner zu finden. Der Lautsprecher wird über eine einfache Schaltung mit Bit 6 des Ports PB verbunden. Der Lesestift wird an Bit 7 des Ports PB angeschlossen.

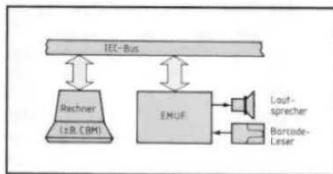


Bild 1. Am IEC-Bus muß der EMUF immer zuerst angesprochen werden. Die dabei verwendete Adresse merkt er sich

Bild 3. Assembler-Listing der EMUF-Software. Sie wurde auf einem Z-80-System mit einem 6502-Crossassembler entwickelt. Vom Basic-Rechner aus muß der IEC-Bus-Barcode-Leser mit GET (A\$) angesprochen werden

```

0000      WIDTH 80
          ;PORT A
          ; 7 .. 8 IEC DATABUS
          ;PORT B
          ; 7 6 5 4 3 2 1 0
          ; BAR BCL -EOT -NOAC -NRDF -OAC -ATH
          ;
0000      PA      EQU     0000
0001      PAD     EQU     0001
0002      PB      EQU     0002
0003      PBD     EQU     0003
          ;
0009      XTEMP1 EQU     00
0001      REF0    EQU     01
0002      REF1    EQU     02
0003      REF     EQU     03
0004      RUF     EQU     04
0005      CHCKL   EQU     05
0006      CHCKH   EQU     06
0007      XTEMP   EQU     07
0008      PFLAG   EQU     08
0009      PNT     EQU     09
000A      FLAG    EQU     0A      ;IEC MERKER ATN ..
000B      ZEICH   EQU     0B      ;ZUSPEICHER
000C      COUNT   EQU     0C      ;ZUSPEICHER
000D      PADR    EQU     0D      ;FIRST TIME
000E      ZEIT    EQU     0E
          ;
          ;
          ; INIT ROUTINE
00FC      ORG     00FC
00FC      DW     0000
          ;
00C8      ORG     00C8      ;START
00C8      A2FF    RESET: LOX  00FF      ;STACKPOINTER
00C9      9A      TXS
00C9      A999    LDA  00000000      ;ALL INPUT
00CA      8D8188 STA  PAD
00CB      A94C    LDA  00100100      ;SET UP
00CC      8D8388 STA  PBD
00CD      A999    LDA  00000000      ;NOT READY NOT ACCEPT
00CE      08      CLD
00CF      78      SEI
00D0      A9FF    LDA  00FF
00D1      8588    STA  PFLAG
00D2      858D    STA  PADR      ;START WERT
00D3      261D6C JBR  BELL1      ;AUSGEBEN TON FUER CHECK
00D4      4CA28D JRP  MAIN
          ; UPRGE
00D5      A9FF    BELL1: LDA  00FF      ;BELL AUSGEBEN
00D6      48      LOPI:  PHA
00D7      AD8268 LDA  PB
00D8      4948    EOR  00100000      ;CHANGE
    
```

```

0C25 8D8268 STA PB
0C28 A958 LDA #560
0C2A EA LOP2: NOP
0C2B EA NOP
0C2C EA NOP
0C2D E981 SBC #1
0C2F D6F9 BNE LOP2
0C31 58 PLA
0C37 E961 SBC #1
0C34 D8E9 RNE LOP1
0C36 AD8888 LDA PA
0C39 293F AND #Z18111111
0C3B 8D8888 STA PA
0C3E 68 RTS

;
0C3F 8688 ; READER: STX XTEMP1
0C41 2488 BIT PFFLAG
0C42 388E RMI START
0C45 A689 LDX PNT
0C47 E484 EING: CPX RUF
0C49 F868 BEQ START
0C4B 501888 LDA BUFFER,X
0C4F E689 INC PNT
0C50 A488 LDX XTEMP1
0C52 68 RTS

;
0C53 2C8268 ; START: BIT PB
0C56 38FB RMI START
0C58 28F36C JSR SYNCH
0C5B 8682 REF1
0C5D 8481 STX REF0
0C5F 4881 LSR REF0
0C61 28F36C JSR SYNCH
0C64 28A88C JSR AUSW
0C67 9876 BCC FEHLER
0C69 28E28C JSR BYTE
0C6C 8584 STA RUF
0C6E A288 LDX #0
0C70 A988 LDA #0
0C72 8585 STA CHCKL
0C74 8586 STA CHCKH
0C76 8487 VOR: STX XTEMP1
0C78 28E28C JSR BYTE
0C7B A687 LDX XTEMP1
    
```

```

0C7D 901888 STA BUFFER,X
0C88 18 CLC
0C81 8585 ADC CHCKL
0C83 8585 STA CHCKL
0C85 9882 BCC NULL
0C87 E686 INC CHCKH
0C89 E8 NULL: INX
0C8A E884 CPX RUF
0C8C D8E8 BNE VOR
0C8E 28E28C JSR BYTE
0C91 C585 CMP CHCKL
0C93 D84A BNE FEHLER
0C95 28E28C JSR BYTE
0C98 C586 CMP CHCKH
0C9A D843 BNE FEHLER
0C9C 281D8C JSR BELL1 ;READY MELDEN
0C9F A288 LDX #0
0CA1 8489 STX PNT
0CA3 8688 STX PFFLAG
0CA5 4C478C JMP EING

;
0CA8 A581 ; AUSW: LDA RFF0
0CAA 4A LSR A
0CAD 6581 ADC RFF0
0CA0 8583 STA REF
0CAF E483 CPX REF
0CB1 1889 BPL EINS
0CB3 8481 STX REF0
0CB5 A581 LDA RFF0
0CB7 8A ASL A
0CB8 8582 STA RFF1
0CBA 18 CLC
0CBB 68 RTS
0CBC 8482 EINS: STX REF1
0CBE A582 LDA RFF1
0CCR 4A LSR A
0CC1 8581 STA RFF0
0CC3 38 SEC
0CC4 68 RTS

;
0CC5 48 ; BITTST: PHA
0CCA A288 LDX #0
0CC8 2C8288 BIT: BIT PB
0CCR 18FB BPL BIT
    
```

Primäradresse wird verglichen, ob dies die erste empfangene ist; wenn ja, so wird ab sofort nur noch auf diese Adresse reagiert. Damit muß der EMUF das erste Gerät sein, daß auf dem IEC-BUS angesprochen wird(!). Die dabei verwendete Primäradresse wird von da ab verwendet.

Bild 4 zeigt das Handshake-Verhalten des IEC-Bus. Wird ein Datenwert auf dem IEC-Bus übertragen, so wird zunächst geprüft, ob alle Geräte fertig sind. Wenn ja, so wird das Datum angelegt und ein DAV-Signal gegeben. Dann reagieren die Geräte mit NDAC und zeigen damit, daß sie das Datum aufgenommen haben. DAV wird nun weggenommen, danach auch NDAC. Das Handshaking gilt für alle Richtungen, also CBM an EMUF und umgekehrt.

Eine Leitung ATN zeigt zusätzlich an, wann eine Adresseninformation auf dem Bus liegt. Dabei wird bei Beginn eines IEC-Zyklus zuerst eine Primäradresse übertragen. Zusätzlich gibt es die Information, ob später eine Datenein- oder Ausgabe erfolgen soll. Die Unterscheidung dafür liegt in den höherwertigen Bits: Wird als Talker (Daten-

sender) adressiert, so fühlt sich unser EMUF angesprochen, denn dann gilt es, Daten zu übertragen. Dazu wird die Routine READER aufgerufen, die ein Zeichen vom Codeleser holt. Das Zeichen wird in ZEICH gerettet. Jetzt wird die Sekundäradresse geholt; der Wert wird ignoriert. Wir brauchen die Sekundäradresse, da sich beim CBM das Handshake nicht an jeder Stelle beliebig lang anhalten läßt. Nach der Sekundäradresse will der CBM schnell seine Daten haben. Dazu senden wir den Inhalt von ZEICH. Danach ist ein Zyklus beendet. Die Daten werden mit EO1 gesendet, um zu zeigen, daß nur ein Wert kommt. Die Werte können dann mit GET# geholt werden. Der Befehl INPUT ist nicht geeignet, da er manche Zeichen („:“ und „:“) verschluckt.

Im CBM-Rechner muß bei Kleinbuchstaben ggf. eine Umrechnung erfolgen, da die Daten unverändert übertragen werden. Kleinbuchstaben in mc-Programmen haben den Wertbereich hex 60 bis 7F und müssen durch Subtraktion von hex 20 in den CBM-Bereich gebracht werden.

Programme, die nur Großbuchstaben

enthalten, können direkt eingelesen werden. Tabelle 1 und Tabelle 2 zeigen zwei mögliche Einleseprogramme beim CBM. Im ersten Fall wird immer eine Zeile eingelesen und auf dem Bildschirm ausgegeben. Danach fährt man mit dem Cursor an diese Zeile und betätigt Return, wodurch die Zeile ins Programm übernommen wird. Dann wird

Tabelle 1: Zeilenweises Einlesen beim CBM

```

10000 OPEN 1,9,15
10010 GET #1,A$
10020 PRINT A$;
10030 IF A$<> CHR$(13) THEN 10010
10040 PRINT CHR$(7)
    
```

Tabelle 2: Ausgabe aller Zeilen auf dem CBM-Schirm

```

10 OPEN 1,8,15
20 GET #1, A$
30 PRINT A$;
40 IF A$= CHR$(13) THEN PRINT CHR$(7);
50 GOTO 20
    
```

mit RUN 10000 das Programm neu gestartet und die nächste Zeile eingelesen. Tabelle 2 zeigt ein Programm, das alle Zeilen ausgibt; es muß allerdings danach irgendwie gestoppt werden, z. B. durch Herausziehen des IEC-Steckers. Die Version ist zwar von der Eingabe her bequemer, kann aber nur sehr kurze mc-Programme verarbeiten. Hier ist ein Betätigungsfeld für CBM-Spezialisten. Vielleicht ist es einfacher möglich, direkt vom IEC-Bus per Maschinenprogramm die Daten einzulesen und sofort als Programm abzulegen. Dazu muß z. B. eine Zeile in einem Puffer abgelegt werden und dann in die Interndarstellung umgewandelt werden. Das IEC-Businterface eignet sich übrigens auch für andere IEC-Bus-Rechner; ggf. kann dann die Sekundäradresse entfallen.

Das programmierte EPROM „IEC BAR“ ist von der Fa. Elektronikladen in Detmold erhältlich, die auch den EMUF-Bausatz liefert.

### Literatur

- [1] Apple-II liest Strichcode. mc 1981, Heft 1.
- [2] Mädchen für alles (EMUF). mc 1981, Heft 2.
- [3] Strichcode-Programme: mc 1981, Hefte 1...3.
- [4] IEC-Bus. Sonderheft Nr. 47, Franzis-Verlag.

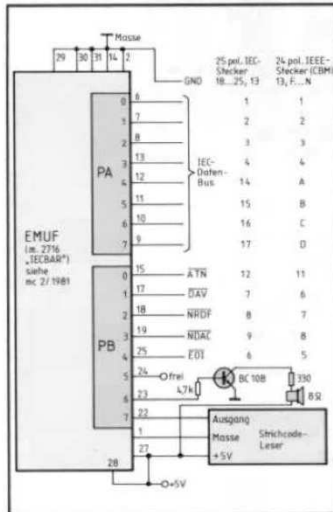


Bild 2. Anschluß von Strichcode-Leser, IEC-Bus-Stecker und Lautsprecher an den Einplatinen-Computer EMUF

```

0CC0 E8      ZAEHL: INX
0CCE A900    LDA      #ZEIT
0CD0 E901    VERZ:   SBC      #1
0CD2 D0FC    BNE      VERZ
0CD4 E0FF    CPX      #255
0CD6 F021    BEQ      UERRL
0CD8 2C0208 BIT      PB
0CD8 30F0    JMI      ZAEHL
0CDD 68      PLA
0CDE 60      RTS

0CDF 4C530C ;
; FEHLER: JMP      START
;
0CE2 40      BYTE:  PHA
0CE3 A000    LDY      #0
0CE5 20C50C NSBIT:  JSR      BITTST
0CE8 20A00C JSR      AUSW
0CE8 60      PLA
0CEC 6A      ROR      A
0CED 40      PHA
0CFE 88      DEY
0CFE D0F4    BNE      NHIT
0CF1 68      PLA
0CF2 60      RTS

0CF3 40      ;
0CF4 20C50C SYNCH: PHA
0CF7 60      JSR      BITTST
0CF8 60      PLA
0CF8 60      RTS

0CF9 60      ;
0CFA 60      UEBRL:  PLA
0CFB 60      PLA
0CFC 60      PLA
0CFD 60      PLA
0CFE 60      PLA
0CFF 4C530C JMP      START
; IEC ROUTINEN
;
0D02 A904    GETCHA: LDA      #200000100 ;RDF
0D04 B00208 STA      PB
0D07 A00208 LOPA:   LOA      PB
0D0A 2902    AND      #200000010 ;DAV WARTEN
0D0C D0F9    BNE      LOPA
0D0E A900    LDA      #200000000
0D10 B00208 STA      PB
0D13 A00008 LDA      PA ;DATA HOLEN
0D16 49FF    EOR      #211111111
0D18 48      PHA
0D19 A00208 LDA      PB
0D1C B50A    STA      FLAG
0D1E A900    LDA      #200000100 ;DAC
0D20 B00208 STA      PB
0D23 A00208 LOPB:  LOA      PB
0D26 2902    AND      #200000010
0D28 F0F9    BEQ      LOPB ;DAV HIGH
0D2A A900    LDA      #200000000
0D2C B00208 STA      PB
0D2F 68      PLA ;DATA WERT
0D30 60      RTS

0D31 A00208 ;
0D34 2901    TALKON: LDA      PB
0D36 F0F9    AND      #200000001
0D38 A912    BEQ      TALKON ;WARTEN BIS ATN WEG
0D3A B00208 LDA      #200010010
0D3D A9FF    STA      PB ;DAV HIGH
0D3F B00108 LDA      #211111111
0D42 A952    STA      PAD ;DATA CHANGE
0D44 A952    LDA      #201010010
0D47 B00308 STA      PBD
0D49 A912    LDA      #200010010
0D49 B00208 STA      PB ;SAFETY
0D4C 60      RTS

0D4D A900    ;
0D4F B00108 TALKOF: LDA      #200000000
0D52 A900    STA      PAD
0D54 B00208 LDA      #200000000
0D57 A94C    STA      PB
0D59 A94C    LDA      #2010001100
0D59 B00308 STA      PBD
0D5C A900    LDA      #200000000
0D5E B00208 STA      PB
0D61 60      RTS
    
```

```

;
;
0D62 48      SEND1: PHA
0D63 28880D SEND: JSR      CKATN ;TESTER RGF
0D64 AD0288 LDA      PB
0D69 2984      AND      #780000100
0D68 F8F6      BEQ      SEND ;WARTEN
0D4D 68      PLA
0D6E 49FF      EOR      #%11111111
0D70 808000 STA      PA
0D73 A918      LDA      #%00010000 ;DAV
0D75 808288 STA      PB
0D78 28880D CONSE: JSR      CKATN
0D7B AD8288 LDA      PB
0D7E 2988      AND      #%00001000
0D80 F8F6      BEQ      CONSE
0D82 A912      LDA      #%00010010 ;PASSIV
0D84 808288 STA      PB
0D87 68      RTS

;
0D88 68      CKATN: RTS ;7.7.
;
0D89 48      SENE01: PHA
0D8A 28880D SENE01: JSR      CKATN
0D8D AD8288 LDA      PB
0D90 2984      AND      #%00000100
0D92 F8F6      BEQ      SENE01
0D94 68      PLA
0D95 49FF      EOR      #%11111111
0D97 808000 STA      PA
0D9A A988      LDA      #%00000000 ;EOI DAV
0D9C 808288 STA      PB
0D9F 4C780D JMP      CONSE

;
0DA2      MAIN:
0DA2 28828D JSR      GETCHA ;IEC ZEICHEN
0DA5 8588      STA      ZEICH ;RETZEN
0DA7 A58A      LDA      FLAG
0DA9 2981      AND      #%00000001 ;ATN
0DAB D8F5      BNE      MAIN ;HIGH DANN NEIN
0DA0 A588      LDA      ZEICH
0DAF 2988      AND      #%00 ;TEST LISTEN TALK
0DR1 C928      CMP      #%28
0DR3 F867      BEQ      LISTPA
0DR5 C948      CMP      #%48
0DR7 F866      BEQ      TALPA
0DR9 4CA28D JMP      MAIN

;
0DR8 4CA28D LISTPA: JMP      MAIN ;HIER NICHT
;
0DRF A58D      TALPA: LDA      PADR ;PRIM ADRESSE
0DC1 C9FF      CMP      #%FF
0DC3 0889      BNE      SK2 ;WEITER SONST
0DC5 A588      LDA      ZEICH
0DC7 298F      AND      #%8F
0DC9 858D      STA      PADR ;NEUE ADRESSE
0DCB 4CD68D JMP      SK3
0DCE A588      SK2: LDA      ZEICH
0DD0 298F      AND      #%8F
0DD2 C58D      CMP      PADR ;VERGLEICH
0DD4 D8CC      BNE      MAIN
0DD6 283F8C SK3: JSR      READER ;HOLE EIN ZEICHEN
0DD9          ;FUER GET
0DDF          ;GET WEGEN ! UND ,
0DD7 8588      STA      ZEICH
0DD8 28828D JSR      GETCHA ;IGNORE SA
0DDE          ;SA MUSS DA SEIN
0DDE 28318D JSR      TALKON ;CHANGE DIR
0DE1 A588      LDA      ZEICH
0DE3 28898D JSR      SENE01 ;! ZEICHEN SENDEN NUN
0DE6 28408D JSR      TALKOF ;UMSCHALTEN
0DE9 4CA28D JMP      MAIN ;ALLES VON VORNE

;
;
; RAM GERIET
;
0818          ORG      #18
;
0818          BUFFER 08 24 ;Puffer BARCODE
0828          ENE     08 1 ;CHECK
;
0868          END

```

----- SYMBOL TABLE -----

AUSW	8CAB
BELLI	8C1D
BIT	8CC8
BITTST	8CC5
BUF	8884
BUFFER	9818
BYTE	8CF2
CHKCH	8986
CHKCL	8885
CKATN	8088
CONSE	8D78
COUNT	888C
EINS	8C47
ENIS	8C8C
ENE	8828
FEHLER	8CDF
FLAG	886A
GETCHA	8D82
LISTPA	8D8C
LOP1	8C1F
LOP2	8C2A
LOPA	8D67
LOP3	8D23
MAIN	8DA2
NOIT	8CE3
NULL	8CB9
PA	8888
PAD	8881
PADR	888D
PB	8882
PB0	8883
PF FLAG	8888
PNT	8889
READER	8C3F
REF	8883
REF8	8881
REF1	8882
RESET	8C88
SEND	8D63
SENO1	8D62
SENE01	8D89
SENE01	8D8A
SK2	8DCE
SK3	8DD6
START	8C53
SYNCH	8CF3
TALKOF	8D4D
TALKON	8D31
TALPA	8DBF
UEBR1	8CF9
VERZ	8CD8
VOR	8C76
XTEMP	8887
XTEMP1	8888
ZAHL	8CCD
ZEICH	8888
ZEIT	8888

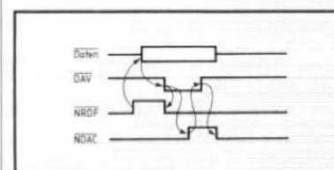


Bild 4. Handshake-Verhalten beim IEC-Bus. Die Signale können in beide Richtungen übertragen werden