

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 In-

halt des Overflow-Flags beeinflusst. ($V = 1 \Rightarrow EOI = \text{„High“}$; $V = 0 \Rightarrow EOI = \text{„Low“}$.) 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 Pro-

gramm ADROUT schließlich gibt 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 (\$00) muß hierfür auf LIST gerichtet werden. Beim Aufruf des Programms durch den Monitor kann mit Hilfe des Carry-Flags die Initialisierung ($C = 0$) vom Datentransfer ($C = 1$) unterschieden werden. Die Initialisierungsroutine INIT programmiert den VIA für die Funktionen „Talker“ (ITALK), fordert interaktive 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 „Carry 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

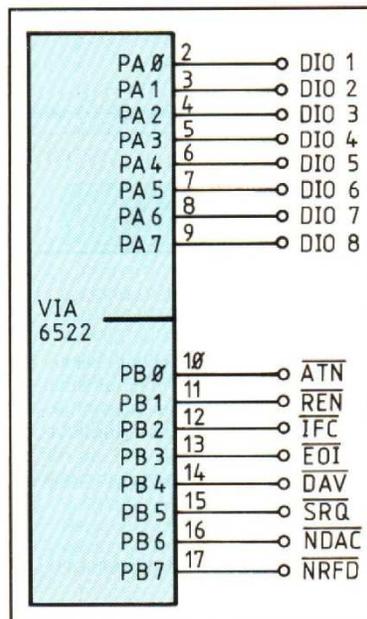


Bild 2. Die IEC-Bus-Routinen können auf jedes 6502-System übertragen werden, das über VIA-Baustein 6522 verfügt

```

UDRA = DATENREGISTER PORT A
UDRB = 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          ; BUS NEUTRALISIEREN
      STA UDRA          ; PORT A OUTPUT
      STA UDDRA
      STA UDRB
      LDA #$1F
      STA UDDRB
      RTS

; DAS UNTERPROGRAMM 'ILIST' INITIALISIERT
; DEN VIA 6522 FUER DIE FUNKTION LISTENER
;
ILIST LDA #0           ; PORT A INPUT
      STA UDDRA
      LDA #$BF
      STA UDRB
      LDA #$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

```

LDY #0FE ; ATN AUF LOW SETZEN
STY UORB
BIT VHIGH ; EDI AUF HIGH SETZEN
JSR SBYTE ; PRIMAER-ADR AUSGEBEN
BCC OUT ; BRANCH BEI FEHLER
TXA
BIT VHIGH ; SEKUNDAER-ADR AUS -
JSR SBYTE ; GEBEN
LDY #0FF ; ATN AUF HIGH SETZEN
STY UORB
RTS
    
```

DAS UNTERPROGRAMM 'RBYTE' EMPFANGT EIN BYTE
EINGANG: A ENTHAELT DAS EMPF. ZEICHEN
V ENTHAELT DEN STATUS VON EDI
V=1 => EDI = 'HIGH'
VERAENDERT: X, FLAGS

```

LDA UORB ; AUF DAV='LOW' WARTEN
AND #010
RNE RBYTE

LDA #03F ; NRFD AUF LOW SETZEN
STA UORB
LDX UDRA ; DATEN UEBERNEHMEN
LDA UORB
AND #8 ; TEST EDI
CLV
BFO EDI
BIT VHIGH ; EDI IST HIGH
LDA #07F ; NDAC AUF HIGH SETZEN
STA UORB

LDA UORB ; AUF DAV='HIGH' WARTEN
AND #010
BCC DAVH
LDA #0BF ; NDAC = LOW
STA UORB ; NRFD = HIGH
    
```

```

TYA
EOR #0FF ; DATEN INVERTIEREN
VHIGH RTS

; DAS UNTERPROGRAMM GIBT EIN BYTE AUF DEN BUS
; EINGANG: A ENTHAELT DAS ZEICHEN
; V ENTHAELT DEN STATUS VON EDI
; AUSGANG: C GELDESCHT BEI FEHLER
; VERAENDERT: Y, FLAGS
    
```

```

SBYTE EOR #0FF ; DATEN INVERTIEREN
TAY
LDA UORB
AND #0C0 ; NRFD & NDAC = HIGH ?
CMP #0C0
BEQ FEHLER ; BRANCH, WENN JA
    
```

```

STY UDRA ; BYTE AUF BUS GEBEN
PHP
BIT UORB ; AUF NRFD='1' WARTEN
BPL NRFD
PLP
LDA UORB
BVS S1
    
```

```

S1 AND #0F7 ; EDI AUF LOW SETZEN
AND #0EF ; DAV AUF LOW SETZEN
STA UORB
    
```

```

NDAC BIT UORB ; AUF NDAC='1' WARTEN
BVC NDAC
LDA UORB
DRA #018 ; DAV UND EDI = '1'
STA UORB
LDA #0FF ; BUS NEUTRALISIEREN
STA UDRA
SEC
RTS
    
```

```

FEHLER CLC
RTS
    
```

Schluß eines Druckers an den AIM-65

SCHLUSS DES AIM 65 AN DEN IEC-BUS UEBER UOUT

```

OUTFLG = #A413
OUTPUT = #E97A
RBYTE1 = #E3FD

BCC INIT

PLA ; DATEN-BYTE HOLEN
CMP #00D ; CR ?
RNE WRITE
CMP #0EB ; ZUM ZWEITEN MAL ?
BEQ ENDE ; BRANCH, WENN JA

STA #0EB
BIT VHIGH ; V='1' => EDI='1'
JSR SBYTE ; BYTE AUSGEBEN
BCC ERROR
RTS

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

LDA #020
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
TXA
    
```

```

PLA
JSR ADROUT ; ADR AUSGEBEN
BCC ERROR
LDA #055 ; UOUT AUF 'U' SETZEN
STA OUTFLG
STA #0EB
RTS ; INIT ENDE

ERROR LDA #020 ; 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 #0FE ; ATN = 'LOW'
STA UORB
LDA #03F ; UNLISTEN CMD
BIT VHIGH ; EDI = 'HIGH'
JSR SBYTE
LDA #0FF ; 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
    
```