

6502 Programmieren - Teil 14#

von Uwe Röder#

In dieser Folge möchte ich Ihnen die Programmierung von Interrupts zeigen, die ja eh nur in Maschinensprache zu realisieren ist.

Es ist zwar in meiner Artikelserie über das Betriebssystem und die Hardwarefähigkeiten der Ataris auch schon auf diese Thema eingegangen worden, doch rechtfertigt der enorme Nutzen der Interrupts die mehrmalige Behandlung dieses Themas.

Da ich hier sowohl auf den DLI, die VBIs und den Timer-Interrupt eingehen möchte, werde ich immer nur kurze allgemein gefasste Beispiele geben und die Interrupts nur einführend vorstellen.

Zuerst zu den einfachen Interrupts.

Dies sind erst einmal die beiden Vertical Blank Interrupts. Ein solcher Interrupt wird jedesmal nach dem Aufbau eines Bildschirms also immer nach 1/50 Sekunde ausgeführt.

Es gibt, wie gesagt, zwei verschiedene VBIs, den Immediate und den Deferred VBI. Der Immediate VBI wird immer, auch während zeitkritischer Operationen durchgeführt. Im Gegensatz dazu wird der deferred VBI bei zeitkritischen Operationen (wenn CRITIC \$42/66 <>0) nicht ausgeführt.

Um einen VBI zu setzen muss der ACCU eine Kennung für die Art des VBI enthalten. 6=immediate
7=deferred

Das X-Register muss das High-Byte und das Y-Register das Low-Byte der Adresse der Interrupt-Routine enthalten. Dann braucht nur noch ein JSR nach SETVBV (\$E45C) ausgeführt zu werden. Bei den VBI-Routinen muss nicht, wie bei einem DLI, darauf geachtet werden, dass die Register bei Eintritt und Verlassen der Routine die gleichen Inhalte aufweisen. Dafür muss man aber anstelle des RTI-Befehls als Endkennung der Interrupt-Routine zu einem je nach Interrupt verschiedenen Rücksprungvektor springen.

Immediate VBI: SYSVBV \$E45F

Deferred VBI: XITVBV \$E462

Dies sieht dann im Programm so aus:

```
00010          .LI OFF
00020 -----
00030 START    LDA #6      ; IMMEDIATE
00040          LDX /IVBI
00050          LDY #IVBI
00060          JSR $E45C
00070 -----
00080 ; IMMEDIATE IST JETZT GESETZT
00090 -----
00100          LDA #7      ; DEFFERED
00110          LDX /DVBI
00120          LDY #DVBI
00130          JSR $E45C
00140 -----
00150 ; DEFERRED IST JETZT GESETZT
00160 ;
00170 ; FORTFUEHRUNG DES HAUPTPROG.
00180 ; ODER EINFACH RTS
```

```

00190 -----
00200         RTS
00210 -----
00220 IVBI     ... ;IMMEDIATE-ROUTINE
00230         ...
00240         ...
00250         JMP $E45F ;ENDE
00260 ;       WICHTIG !!!
00270 -----
00280 DVBI     ... ;DEFERRED-ROUTINE
00290         ...
00300         ...
00310         JMP $E462 ;ENDE
00320 ;       WICHTIG !!!
00330 -----

```

Soviel zu den VBIs. Wenden wir uns jetzt den fünf SYSTEM-Timern zu. Diese Timer bestehen aus je zwei Bytes und werden jede 1/50 Sekunde um eins vermindert.

Der erste Timer ist dabei der einzige, der auch während zeitkritischer Operationen vermindert wird.

Die ersten zwei Timer verfügen über einen Sprungvektor durch den gesprungen wird wenn der Timer von 1 auf 0 herunter gezählt wurde. Die dadurch ausgelöste Routine muss mit einem RTS enden. Bei den anderen drei Timern wird dafür nur ein Flag gesetzt. Die Handhabung ist damit also klar:

1. Sprungvektor einrichten oder Flag=0
2. Timer setzen

Die Adressen der Timer lauten wie folgt:

```

CDTMV1 $218/$219
CDTMV2 $21A/$21B
CDTMV3 $21C/$21D
CDTMV4 $21E/$21F
CDTMV5 $220/$221

```

Die beiden Sprungvektoren für die ersten zwei Timer sind:

```

CDTMA1 $226/$227
CDTMA2 $228/$229

```

Die drei Flags für die Timer 3 bis 5 sind:

```

CDTMF3 $22A
CDTMF4 $22C
CDTMF5 $22E

```

Die einzelnen Timer können auch über die schon oben erwähnte Routine SETVBV gesetzt werden. Der ACCU muss dafür die Timernummer (1-5), das X-Register das High-Byte und das Y-Register das Low-Byte der Zeitspanne in 1/50 Sekunden enthalten.

```

00010         .LI OFF
00020 -----

```

```

00030 SETVBV      .EQ $E45C
00040 CDTMA2     .EQ $228
00050 -----
00060 S          LDA #TIM
00070           STA CDTMA2
00080           LDA /TIM
00090           STA CDTMA2+1
00100           LDA #2
00110           LDX #1 ;10 SEK.
00120           LDY #244
00130           JSR SETVBV
00140           RTS
00150 -----
00160 TIM         LDA #$30
00170           STA $2C6
00180           RTS
00190 -----

```

Bei diesem Beispiel kehrt die Kontrolle sofort nach Ausführung des kurzen Hauptprogramms wieder an den Assembler zurück, so dass man wieder etwas anderes machen kann. Nach 10 Sekunden ändert sich dann 'spontan' die Hintergrundfarbe von blau in rot.

Der letzte und wohl auch am häufigsten verwendete Interrupt, den ich hier besprechen möchte, ist der Display List Interrupt, kurz DLI. Mit Hilfe des DLI können wir bekanntlich während des Bildschirmaufbaus die Farben ändern oder den Zeichensatz wechseln.

Der Vektor für den DLI ist VDSLST \$200/\$201. Um einen DLI zu nutzen muss in der Display List (\$230/\$231) in der entsprechenden Zeile das höchstwertige Bit (7) gesetzt werden. Dies kann durch eine "Oderierung" mit #\$80 geschehen. Des weiteren muss in NMIEN \$D40E ebenfalls Bit 7 gesetzt werden. Dies erreicht man durch Einschreiben von #\$C0. Um einen DLI einzurichten muss man erst den Interrupt sperren (NMIEN=0), dann müssen die Adresse der DLI-Routine in VDSLST eingetragen und die entsprechenden Bits in der Display List gesetzt werden. Der letzte Schritt besteht nun darin den DLI freizugeben (NMIEN=#\$C0).

Bei der DLI-Routine ist zu beachten, dass die drei Register (A,X,Y) vor und nach Eintritt der Routine die gleichen Inhalte haben. Man muss diese Inhalte also direkt am Anfang der Routine auf dem Stapel ablegen und später am Ende der Routine wieder einlesen. Die DLI Routine muss mit einem RTI enden.

Beispiel:

```

00010           .LI OFF
00020 -----
00030 NMIEN      .EQ $D40E
00040 VDSLST     .EQ $200
00050 SDLSTL     .EQ $230
00060 COLOR      .EQ $D018
00070 DLV        .EQ $D0
00080 WSYNC      .EQ $D40A
00090 -----
00100 START      LDA #0
00110           STA NMIEN
00120           LDA #DLI
00130           STA VDSLST
00140           LDA /DLI
00150           STA VDSLST+1
00160           LDA SDLSTL
00170           STA DLV

```

```

00180          LDA SDLSTL+1
00190          STA DLV+1
00200          LDY #15
00210          LDA (DLV),Y
00220          ORA #$80
00230          STA (DLV),Y
00240          LDA #$C0
00250          STA NMIEN
00260          RTS
00270 -----
00280 DLI          PHA ;REGISTER
00290          TXA ;RETTEN
00300          PHA
00310          TYA
00320          PHA
00330 -----
00340          LDA #$30
00350          STA WSYNC ;SYNCHRON.
00360          STA COLOR ;FARBE
00370 -----
00380          PLA ;REGISTER
00390          TAY ;ZURUECKHOLEN
00400          PLA
00410          TAX
00420          PLA
00430          RTI
00440 -----

```

So, das war's dann wieder mal.

Ihr Uwe Röder
CSM 09/1989

Der Artikel entstammt der Kursreihe ?6502 Programmieren? des Compy Shop Diskettenmagazins. Die Kursreihe besteht aus 14 Kursen, die im Laufe des Jahres 2011 in unregelmäßigen Abständen einzeln veröffentlicht werden, bzw. anschließend als Zusammenzug als ABBUC-Buch ?6502 Programmieren? erscheinen.

Koordination: Volkert Barr (volkert@nivoba.de)

Version 1.1 / 2011-01-23