

Farben-Tip

# Oben 16 Farben im Bild und gleichzeitig unten 80 Zeichen Text

Mehr Mode  
in einem  
Bild

Die Schneider-Computer können bekanntlich auf 3 verschiedene Arten Grafik und Text darstellen: 16 Farben und eine so geringe Auflösung, daß nur noch 20 Zeichen/Zeile dargestellt werden ODER ein Kompromiß aus 40 Zeichen/Zeile bei 4 Farben pro Bildschirmpunkt ODER die höchste, aber einfarbige Auflösung von 80 Zeichen/Zeile. Wäre es nicht schön, wenn man z.B. im oberen Bildschirm Drittel ein 16farbiges Bild UND im unteren Drittel den Begleittext mit 80 Zeichen/Zeile darstellen könnte? Sicher haben Sie es schon vermutet: Weil es so schön ist, werden wir es jetzt auch machen.

Unser Problem ist es also, verschiedene MODEs gleichzeitig auf dem Bildschirm darzustellen. Prinzipiell gibt es zwei Lösungen dafür: Wir könnten einen Computer, z.B. einen Atari 130 kaufen, dessen Videoprozessor das schon von Haus aus kann, oder wir müßten uns eines Software-Tricks bedienen. Da die erste Möglichkeit aus praktischen Erwägungen ausscheidet, beschäftigen wir uns besser mit



der notwendigen Software. Bei jedem Interrupt wurde das Farb-Änder-Programm aufgerufen, wodurch 6 farbige Balken untereinander entstanden. Damals störte es uns, daß die Balken genau an der gleichen Stelle stehenblieben, ohne zu flackern oder sich zu bewegen, aber heute ist das genau das, was wir brauchen.

## Interrupts sind die Lösung

Angenommen, wir wollten die obere Hälfte des Bildes in Mode 0 (16 Farben) darstellen und die untere Hälfte in Mode 2 (80 Zeichen).



Wir brauchen dazu lediglich Mode 0 einzuschalten, zu warten, bis der Elektronenstrahl das halbe Bild gezeichnet hat, und dann Mode 2 einzuschalten. So weit, so gut. Aber wir wollen ja nicht nur das tun, sondern zwischendurch sollen ja auch noch unsere normalen Programme weiterlaufen, also rufen wir das Programm über Interrupts auf. Das hat noch einen weiteren Vorteil: 300 Interrupts (Unterbrechungen) werden pro Sekunde erzeugt, 50 mal pro Sekunde wird ein Bild gezeichnet. Wir haben also 6 Unterbrechungen in einem Bild, können theoretisch also 6 mal den Bildschirmmodus ändern. Vorausgesetzt, daß der erste Interrupt zeitlich mit dem Beginn eines neuen Bildes übereinstimmt, brauchen wir also nur bei jedem 3. Interrupt nach dem Bildanfang auf Mode 2 und synchron mit dem Bildanfang auf Mode 0 zu schalten, und schon sind wir da, wo wir hin wollten.

Gesagt, getan. Listing 1 ist das Ergebnis unserer bisherigen Überlegungen. Es wird auf die übliche Weise ein Interruptblock eingebunden, so daß ROUTI dreihundertmal pro Sekunde aufgerufen wird. (Zeichen 70/80. Im Listing entspricht + bd19 &bd19, ist also hexadezimal. Das Zeichen + wird vom Schneider-Devpac-Assembler verwendet; wer einen anderen Assembler hat, muß das Symbol entsprechend ändern.) Bei jedem Aufruf von ROUTI wird mitgezählt, der

## Listing 1

wieviele Aufruf es war. Handelt es sich um Nr. 1, wird Mode 0 eingeschaltet, bei Nr. 3 wird Mode 2 eingeschaltet, bei Nr. 6 wird der Zähler wieder zurückgesetzt, weil ja 6 Interrupts pro Bild auftreten. Allerdings wird der Zähler abwärts, also nicht von 0-5 gezählt. Zeilen 240-380 erledigen das und wählen den entsprechenden Mode an. Sehr wichtig sind aber die Zeilen vor und nach dem Einbinden in die Interruptkette: Da die obere Bildschirmhälfte in Mode 0 dargestellt werden soll, schalten wir erst einmal auf diesen um und warten dann auf den Strahlrücklauf (40-60). Erst dann wird KL-ADD-FASTTICKER aufgerufen. Anschließend werden die Interrupts gesperrt, da wir jetzt noch einmal warten müssen, und zwar länger als die Zeitperiode, die zwischen zwei Interrupts

```

A000      10      org  #a000                ;LISTING Nr.2
A000      20      ent  $                    ;Startadresse &a000
A000 210AA0      30 init: ld  hl,free4      ;4 freie Bytes zur Verwaltung
A003 010EA0      40      ld  bc,tab1       ;Sprungadressentabelle
A006 CDD1BC      50      call #bcd1        ;RSX'es einbinden
A009 C9         60      ret                ;das war es
A00A         70 free4: defs 4
A00E 16A0       80 tab1: defw ntab
A010 C31CA0     90      jp  init2          ;Modeswitch einschalten
A013 C335A0    100      jp  aus            ;'' ausschalten
A016 4FCE      110 ntab: defb "0","N"+#80   ;Tabelle der Namen
A01B 4F46C6    120      defb "0","F","F"+#80
A01B 00        130      defb 0            ;Ende der Tabelle
                        140 ;
A01C 3E00     150 init2: ld  a,0           ;Mode 0
A01E CD1CBD    160      call #bd1c         ;einschalten
A021 CD19BD    170      call #bd19        ;MC_WAIT_FLYBACK
A024 2141A0    180      ld  hl,block       ;binde in Interruptkette ein
A027 CDE3BC    190      call #bce3        ;KL_ADD_FAST_TICKER
A02A F3        200      di                ;keine Stoerung bitte...
A02B 3E06     210      ld  a,6            ;Zaehler auf Startwert
A02D 3269A0    220      ld  (auf),a       ;(wird heruntergezählt)
A030 CD19BD    230      call #bd19        ;warte auf Strahlruecklauf
A033 FB        240      ei                ;ist synchronisiert
A034 C9        250      ret
                        260 ;
A035 2141A0    270 aus: ld  hl,block       ;Block wieder aus...
A03B CDE6BC    280      call #bce6        ;...Interruptkette herausnehmen
A03B 3E02     290      ld  a,2
A03D CD1CBD    300      call #bd1c         ;Mode 2 einschalten
A040 C9        310      ret
                        320 ;
A041 0000     330 block: defw 0           ;Interruptblock
A043 0000     340      defw 0
A045 00        350      defb 0
A046 B1        360      defb #B1
A047 4CA0     370      defw routi
A049 00        380      defb 0
A04A 0000     390      defw 0
                        400 ;
A04C 3A69A0    410 routi: ld  a,(auf)      ;Das eigentliche Programm
A04F 3D        420      dec  a            ;erniedrige Zaehler und...
A050 3269A0    430      ld  (auf),a
A053 FE00     440      cp  0              ;...ueberpruefe welcher Interrupt
A055 280B     450      jr  z,mod0        ;bei Nummer 0 ->Mode 0
A057 FE02     460      cp  2
A059 C0        470      ret  nz          ;auch nicht Nr.2
A05A 3E02     480      ld  a,2
A05C C31CBD    490      jp  #bd1c         ;sonst...
A05F 3E06     500 mod0: ld  a,6          ;Mode 2!
A061 3269A0    510      ld  (auf),a       ;Zaehler wieder auf Anfang
A064 3E00     520      ld  a,0
A066 C31CBD    530      jp  #bd1c         ;Mode 0 einschalten
A069 06        540 auf: defb 6           ;Zaehlvariable
A06A         550      end

```



liegt. Nun werden der Zähler auf seinen Startwert gesetzt und die Interrupts — und somit der Aufruf unseres Programms — wieder eingeschaltet, sobald ein neuer Frame-Fly aufgetreten ist (90-140). Diese komplizierte Prozedur ist notwendig, um eine möglichst gute Synchronisation bzgl. des Umschaltprogramms und der tatsächlichen Darstellung auf dem Bildschirm zu erreichen. Sonst könnte es leicht passieren, daß unser Mode-0-Bereich irgendwo in der Bildmitte „hängt“ und die Bereiche darunter und darüber in Mode 2 angezeigt werden. Wenn Sie nun Listing 1 laufen lassen, werden Sie allerdings feststellen, daß wir keine exakte Teilung des Bildschirms in der Mitte haben, sondern daß unten 7 Zeilen Text und oben 18 Zeilen Grafik angezeigt werden. Das liegt aber nicht etwa daran, daß nun beim falschen Interrupt umgeschaltet wurde, sondern daran, daß ja nicht nur die eigentliche Bildfläche, sondern die gesamte BILDSCHIRMfläche, also auch der Border vom Elektronenstrahl mitgezeichnet wird, so daß die Interrupts nicht synchron zur Bildschirmposition 0,0, sondern zur oberen Bildschirmecke liegen. Um also tatsächlich in der Bildmitte umzuschalten, muß die Nummer in Zeile 290 entsprechend erniedrigt werden.

## Andere Modes/Farben...

Listing 1 ist natürlich noch nicht das Nonplusultra. Ärgerlich ist z.B., daß Mode 0 noch etwas in die erste Textzeile hineinreicht. Man kann hier noch nachregulieren, obwohl der nächste Interrupt erst ein ganzes Stück tiefer kommt: Nach Setzen des neuen Modes noch ein bißchen mit einer Verzögerungsschleife a'la LD Bx:DJNZ \$ warten. 50 wäre hier ein guter Wert für x, aber die genaue Zahl zu ermitteln, überlasse ich Ihnen (denken Sie aber daran, daß dadurch andere Programme, die normal ablaufen, verzögert werden können).

Kein Problem ist es auch, andere Modes einzuschalten (einfach die Zahlen in Zeile 40,310 und 350 ändern) oder noch mehr Modes auf dem Bildschirm zu mischen. Ebenfalls eine interessante Idee wäre

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

A000      10      org  #a000      ;LISTING NR.1
A000      20      ent  $          ;hier Pr-Start
                                30 ;

A000 3E00      40 init: ld  a,0      ;Initialisieren:erst Mode 0
A002 CD1CBD      50      call #bdlc ;einschalten
A005 CD19BD      60      call #bd19 ;warte auf Strahlruecklauf
A008 2119A0      70      ld  hl,block ;binde in Interruptkette ein
A00B CDE3BC      80      call #bce3 ;KL_ADD_FAST_TICKER
A00E F3          90      di          ;dauert laenger als 1/300 Sek
A00F 3E06      100     ld  a,6      ;Zaehler auf Startwert
A011 3241A0      110     ld  (count),a ;(wird heruntergezählt)
A014 CD19BD      120     call #bd19 ;warte auf Strahlruecklauf
A017 FB          130     ei
A018 C9          140     ret
                                150 ;

A019 0000      160 block: defw 0      ;Block mit Daten fuer
A01B 0000      170     defw 0      ;Interrupthandler
A01D 00          180     defb 0
A01E 81          190     defb #81
A01F 24A0      200     defw routi
A021 00          210     defb 0
A022 0000      220     defw 0
                                230 ;

A024 3A41A0      240 routi: ld  a,(count) ;Das eigentliche Programm
A027 3D          250     dec  a      ;erniedrige Zaehler und...
A02B 3241A0      260     ld  (count),a
A02B FE00      270     cp  0      ;...ueberpruefe welcher Interrupt
A02D 280B      280     jr  z,mod0 ;bei Nummer 0 ->Mode 0
A02F FE02      290     cp  2
A031 C0          300     ret  nz      ;auch nicht Nr.2
A032 3E02      310     ld  a,2      ;sonst...
A034 C31CBD      320     jp  #bdlc ;Mode 2!
A037 3E06      330 mod0: ld  a,6      ;Zaehler wieder auf Anfang
A039 3241A0      340     ld  (count),a
A03C 3E00      350     ld  a,0
A03E C31CBD      360     jp  #bdlc ;Mode 0 einschalten
A041 06          370 count: defb 6 ;Zaehlvariable
A042          380     end

```

Pass 2 errors: 00

Table used: 71 from 211

Executes: 40960

```

1 REM LISTING Nr.3
2 DATA 21,0A,A0,01,0E,A0,CD,D1,BC,C9,00,00,00,00,1
6,A0,C3,1C,A0,C3,35,A0,4F,CE,4F,46,C6
3 DATA 00,3E,00,CD,1C,BD,CD,19,BD,21,41,A0,CD,E3,B
C,F3,3E,06,32,69,A0,CD,19,BD,FB,C9
4 DATA 21,41,A0,CD,E6,BC,3E,02,CD,1C,BD,C9,00,00,0
0,00,00,81,4C,A0,00,00,00,3A,69,A0,3D
5 DATA 32,69,A0,FE,00,28,08,FE,02,C0,3E,02,C3,1C,B
D,3E,06,32,69,A0,3E,00,C3,1C,BD,06
6 MEMORY &9FFF:MODE 2
7 RESTORE 2:FOR x=&A000 TO &A069:READ w$:POKE x,VA
L("&"+w$):NEXT:CALL &A000
8 PRINT"öON - schaltet Modesplit ein":PRINT"öOFF -
schaltet Modesplit aus"

```



es, gleichzeitig mit dem Umschalten des Modus auch die angezeigten Farben zu ändern, so daß man in Mode 0 andere Farben verwenden könnte als in Mode 1, die aber doch gleichzeitig auf dem Bildschirm erscheinen. Es gibt hier noch viele weitere Möglichkeiten, und ich bin sicher, Sie haben genug Phantasie, um das Programm selbst weiterzuentwickeln.

## Neue BASIC-Befehle

Damit wären wir bei Listing 2. Hiermit ist es möglich, die „gesplittete“ Darstellung mit den Befehlen ION ein- und mit IOFF auszuschalten. INIT erfüllt hier eine ganz andere Funktion als in Listing 1: Es werden lediglich die beiden Befehle ON und OFF generiert, und erst INIT2 bindet dann ROUTI in die Interruptkette ein. Bei einem OFF-Kommando wird dann AUS aufgerufen und der Interruptblock wieder entfernt, so daß wieder alles einheitlich in einem Mode dargestellt wird.

Das letzte Programm, Nr. 3, ist eine absolute Ausnahme: Hier haben wir auch mal einen „Happen“ für reine BASIC-Programmierer: Es tut im Prinzip dasselbe wie Nr. 2, aber man braucht keinen Assembler, um es zu nutzen. Also: Listing 3 an den Anfang des eigenen Programmes, und man kann mit ION den Modesplit ein- und mit IOFF wieder ausschalten.

Sicherlich fällt jedem beim Herumexperimentieren mit dem Modesplit schon nach sehr kurzer Zeit auf, daß die Schrift recht merkwürdig aussieht, wenn Sie in den Mode 0-Bereich gerät: Der Grund dafür ist klar: Der Zeichengenerator nimmt natürlich immer noch an, daß auf dem Bildschirm Mode 2 (beim Herumexperimentieren mit all unseren Beispiellistings sollte in Mode 2 gearbeitet werden) dargestellt wird, und so werden nicht die richtigen Bitmuster für Mode 0 erzeugt. Um das zu verhindern, kann man WINDOW +0,1,80,19,25 definieren oder mit POKE &blc8,0 die Schrift an Mode 0 anpassen (an dieser Adresse steht der Bildschirm-

modus bei 464. Siehe Vergleich zwischen 464 und 664), aber leider funktioniert die Grafik deshalb trotzdem nicht. Mit einem POKE &blc8,0:call &bbba ist es schon besser, aber immer noch nicht perfekt. Man muß also den Zeichengenerator komplett an Mode 0 anpassen, um oben normal zeichnen zu können. Die Lösung dafür werde ich nachreichen. Allerdings wird

## Vorschau

man normalerweise auch direkt Mode 0-Bilder von Diskette oder Tape in den Bildschirm laden und so keine Probleme mit dem Zeichengenerator bekommen. Man sollte Modesplit nach Tape- oder Diskettenbenutzung neu einschalten, da bei Tape/Disk-Betrieb die Interrupts ausgeschaltet werden und die Synchronisation gestört wird.

TMB

# Der Schneider Tip

ROM-CALL's ohne Einspruchbedingungen zum Ausschneiden und Sammeln

CALL &BB00 Initialisierung der Tastatur. CALL &BB03 Rücksetzung der Tastaturverwaltung. Leeren des Tastaturpuffers.  
CALL &BB06 Warte auf das nächste Zeichen von der Tastatur.  
CALL &BB19 Warte auf nächste Tastatureingabe.

CALL &BB3F in Direkteingabe: keine Repeatfunktion mehr.  
CALL &BB48 BREAK ausschalten.

CALL &BB45 BREAK einschalten.  
CALL &BB4E Setzt Textcursor auf Bildschirmfang, oben links.  
CALL &BB57 Verbietet, daß Zeichen am Bildschirm dargestellt werden.  
CALL &BB54 Erlaubt, daß Zeichen am Bildschirm dargestellt werden.

CALL &BB7B Lasse Cursordarstellung (Anwender) zu. Z.B. bei Input.  
CALL &BB7E Verbiete Cursordarstellung (Anwender). Z.B. bei Input.

CALL &BB9C Tausche Ink's für Stift und Papier.  
CALL &BB14 Löscht Bildschirm bis Cursorposition.  
CALL &BB6B Verhindert Bildschirmausgabe folgender Meldungen: Press PLAY then any Key: Press REC and Play then any Key: Found „FILENAME“ Block „N“, Loading „FILENAME“ Block „N“, Saving „FILENAME“ Block „N“.