

PETER'S ASSEMBLIERECKE

für ATARI -Computer

Scrolling Teil 1

Das Scrollen des Bildschirms ist eine der ältesten aber gleichzeitig auch interessantesten Fähigkeiten, über die Computer verfügen. Grundsätzlich kann man Scrolling dann brauchen, wenn man mehr Information am Bildschirm darstellen will, als in eine Füllung des Bildschirmspeichers paßt, etwa wenn ein langes Basic-Listing am Bildschirm ausgegeben werden soll. Daß die Ausgabe eines BASIC-Listings keine besonders anspruchsvolle Art des Scrollens ist, braucht nicht extra erwähnt zu werden, denn es handelt sich dabei um sogenanntes Software-Scrolling.

Die Möglichkeit des Scrollings reichen noch viel weiter. Programme zur Textverarbeitung können damit die Beschränkung auf 40 Zeichen pro Zeile elegant umgehen, die Darstellung von umfangreichen elektronischen Schaltungen wäre denkbar und vieles, vieles mehr. Und natürlich wäre eine ganze Anzahl der besseren Spiele ohne Scrolling überhaupt nicht vorstellbar (Scramble, Caverns of Mars, Zaxxon...).

Jeder wird nun einsehen, daß das ruckartige Scrolling eines Basic-Listings mit dem eines Zaxxon recht wenig gemein hat. Daher ist es sinnvoll, einige verschiedene Arten des Scrollens zu unterscheiden: Software- und Hardware-Scrolling, vertikales und horizontales Scrolling, Grob- und Fein-Scrolling.

Artenvielfalt

Beginnen wir von vorne: Ein Beispiel für Software-Scrolling haben wir bereits besprochen. Charakteristisch dafür ist, daß jedes Byte des Bildschirmspei-

chers per Software an seine neue Stelle verschoben wird. Zu diesem Zweck ist es oft nötig, mehrere hundert oder gar mehrere tausend Byte zu bewegen. Wesentlich eleganter ist da schon das Hardware-Scrolling, eine Fähigkeit, die der Atari wie kein anderer auf dem Homecomputer Sektor beherrscht. Der wesentliche Grundgedanke dabei ist es, nicht die Bildinformation (in Form von Bytes) durch den Bildschirmspeicher zu schaufeln, sondern den Bildschirmspeicher über die Bildinformation zu verschieben. Damit kann man (im Extremfall) durch die Veränderung von nur zwei Bytes denselben Effekt herbeiführen, für den man beim Software-Scrolling nahezu 8000 Bytes verschieben müßte.

Vertikales und horizontales Scrolling unterscheiden sich einfach in der Richtung der Bewegung. Dabei ist horizontales Scrolling beim Atari etwas aufwendiger als rein vertikales. Selbstverständlich können auch beide Formen kombiniert auftreten (2D-Scrolling).

Grob- und Fein-Scrolling bezieht sich auf die kleinste Bewegungseinheit des Scrollens. Grob heißt dabei, daß als Grundschrift ein Zeichen dient, als Fein-Scrolling wird eine pixelweise Bewegung des Bildes bezeichnet.

Vertikales Grob-Scrolling

Nach soviel trockener Theorie kann ein wenig Praxis nicht schaden. Das Atari-Basic-Programm in Listing 1 demonstriert, wie hardwaremäßiges, vertikales Grob-Scrolling funktioniert. Es benutzt den Charakter-Modus GRAPHICS 1 und scrollt den Bildschirm

durch Veränderung der Displaylist. Bei einer durch einen GRAPHICS-Aufruf erzeugten Displaylist ist die Adresse des Bildschirmspeichers immer im fünften (niederwertiges Byte) und sechsten (MSB) Byte vom Anfang der Displaylist aus zu finden. Ändert man nun diese Adresse um jeweils die Anzahl der Bytes pro Zeile des benutzten Grafikmodus (im Beispiel 20 Bytes), so erzeugt man vertikales Scrolling. Obgleich es nicht unbedingt einen ästhetischen Genuß verspricht, wird dabei der Einfachheit halber das O.S. ROM von \$E000 bis \$FFFF als Bildspeicher mißbraucht, über das Sie mit einem Joystick in Port 0 hinwegscrollen können.

Warum das mit diesem Programm erzeugte Scrolling nicht gerade einzigartig ist, hat zwei Gründe. Erstens haben wir nur Grob-Scrolling verwendet, so daß die Bewegung recht ruckartig wirkt. Zweitens müssen gerade beim Scrolling alle Veränderungen der Bildschirmgrafik immer im Gleichschritt mit der Bilderzeugung des Computers (und des Fernsehgerätes bzw. Monitors) sein, sonst treten vielerlei Störungen in der Grafik auf. Diesen zweiten Punkt können wir im Rahmen eines Basic-Programmes nur schwerlich erfüllen. Mit anderen Worten: hier muß Maschinensprache her.

Feinheiten

Wenn wir nun schon mal bei Assembler gelandet sind, dann machen wir auch Nägel mit Köpfen und gehen gleich zu Fein-Scrolling über. Dazu sind zwei weitere Schritte notwendig: In der Displaylist muß bei jeder ANTIC-Anweisung das Vscroll-Bit (Bit Nr. 5) gesetzt sein. Wir werden daher diesmal keine durch GRAPHICS erzeugte Displaylist, sondern eine "handgestrickte" verwenden. Die Feinverschiebung des Bildes wird dann durch Einschreiben eines Wertes von 0-15 in das VSCROL-Register (\$D405) von ANTIC erreicht. Wie Sie richtig vermuten, ist damit eine maximale Bildverschiebung von 15 Pixels möglich, also genau so weit, wie ein GRAPHICS-2-Zeichen hoch

ist. Eine Fortsetzung des Scrollvorganges ist dann mit Hilfe des Grob-Scrollings, also der Veränderung der Bildschirmadresse (der sog. LMS-Bytes, Load Memory Scan-Bytes) möglich. Konkret bedeutet dies, daß zum pixelweisen Scrolling über einen größeren Speicherbereich Grob- und Feinscrolling kombiniert angewendet werden müssen.

Scrolling im VBI

Ein Beispiel für den gerade geschilderten Sachverhalt können Sie dem Assemblerlisting entnehmen. Wer die Assemblercke bisher schon verfolgt hat, der wird dort einen alten Bekannten wiederfinden: den Vertical Blank Interrupt, der schon beim PM-Helfer zum Einsatz kam. Qualitativ hochwertiges Scrolling ist nur durch den VBI möglich, da wie schon erwähnt, unbedingt der Gleichtakt zur Bilderzeugung eingehalten werden muß. Grafikänderungen während des VBIs führen bekanntlich nie zu Störungen, da während dieser Zeit kein Bild ausgegeben wird.

Die VSINIT-Routine trägt zunächst die anfängliche Bildschirmadresse in die "handgestrickte" Displaylist ein, aktiviert die neue Displaylist, setzt das VSCROL-Register auf 0 und fügt die Routine VSVBI in den VBI ein. Da es sich bei VSCROL um ein Write-only Register handelt, wird ein SVSCR Schattenregister eingerichtet, das immer am Ende des VBIs in das Hardwareregister übertragen wird.

Das VBI-Programm, das von nun an 50 mal in der Sekunde durchlaufen wird, fragt zunächst den Joystick Nr.0 ab. Wird dieser nach oben gedrückt, kommt das Unterprogramm OBEN zum Zuge, umgekehrt ist das UP UNTEN dran. Beide Programmteile enthalten die gleiche Strategie, so daß es genügt, eines davon näher zu erläutern. Im Unterprogramm OBEN wird zunächst geprüft, ob eine Zeichengrenze erreicht ist, d. h. ob das Fein-Scrollregister den Wert 0 besitzt. Nur in diesem Fall muß überprüft werden, ob das Ende des Scroll-Bereiches erreicht

wurde. Ist das nicht der Fall, so wird das Fein-Scrollregister um eins erhöht und damit der Bildschirm um ein Pixel nach oben verschoben. Falls SVSCR nun größer als VMAX geworden ist, dann reicht die Feinverschiebung nicht aus, es muß also grob gescrollt werden. Dazu wird zur Adresse des Bildschirmspeichers die Länge einer Zeile addiert, wodurch die bisher zweite Zeile an die erste Position rückt.

Gährende Leere

Haben Sie das Assemblerlisting eingetippt und starten es an der Adresse \$0601, dann sehen Sie - nichts! Das ist auch nicht weiter verwunderlich, denn der als Bildschirmspeicher verwendete Bereich von \$7000

bis \$7FFF ist ja noch jungfräulich unbenutzt. Sie können diesen Speicherbereich nun z. B. mit dem Fill-Befehl des Maschinensprache-Monitors füllen. Oder Sie benutzen das BASIC-Listing 3, das außer dem Basic-Loader für das Maschinenprogramm noch einen Programmteil zur Erzeugung einer Zufallslandschaft enthält, durch die Sie mittels eines Joysticks hindurchscrollen können. In diesem Programm können Sie übrigens gut beobachten, wie Basic- und Maschinenprogramm gleichzeitig laufen:

Während Sie sich durch das Terrain bewegen, werden Sie vom Basic eingeholt, das immerzu neue Landschaften generiert.

Stellen Sie sich nun vor, das Scrolling-Programm in geeigneter Weise mit dem PM-Helfer der März-CK zu kombinieren. Damit hätten Sie dann ein Programm wie "Caverns of Mars" schon fast fertig - die Assembler-Ecke macht's möglich. Allerdings, und das muß zur Dämpfung der Euphorie gleich gesagt werden, ist dies nur mit den Assemblerprogrammen und nicht mit den Basic-Loadern machbar.

In einer der nächsten Ausgaben werden wir uns im zweiten Teil mit einigen Feinheiten sowie mit horizontalem Scrolling beschäftigen, bis dahin - Tschüß!

Peter Finzel

Listing 1

```

100 REM * Listing 1:
110 REM *      DEMO fuer vertikales
120 REM *      Grobscrolling
130 GRAPHICS 1
150 ? :? "SCROLL-DEMO/ GRAPHICS 1"
160 LMS=PEEK(560)+PEEK(561)*256+4:REM LMS-Adresse (Display-List)
170 S=57344:REM *      $E000 (OS-ROM)
180 Z=20:REM *      GR.1 Zeile ist 20 Bytes lang
190 BA=57344:BE=65295:REM Anfang & Ende des Scrollbereiches
200 IF STICK(0)=13 THEN S=S-Z:IF S<BA THEN S=BA
210 IF STICK(0)=14 THEN S=S+Z:IF S>BE THEN S=BE
220 H=INT(S/256):REM *      Anfangsadresse des
230 L=S-H*256:REM *      Bildschirmspeichers festlegen
240 POKE LMS,L:POKE LMS+1,H
290 GOTO 200

```

Listing 2

```

0100 ;*****
0110 ;LISTING 2:
0120 ;Vertikales Fein-Scrolling
0130 ;      in Maschinensprache
0140 ;
0150 ;      Peter Finzel '85
0160 ;*****
0170 ;
0180 ; Konstante
0190 ;
0200 ;
0210 YMIN = $7000
0220 YMAX = $7000+$3840 Scroll-Bereich=192 Zeilen
0230 VMAX = 15      Maximalwert des Fein-Scroll Reg.
0240 ZLAENG = 20    Zeilenlänge in Byte
0240 MDB = 7+520    ANTIC Modus 7 und Vscroll-Bit
0250 ;
0260 ; Operating System
0270 ;
0280 ;
0290 ;
0300 ;
0310 ;
0320 ;
0330 ; Hardware
0340 ;
0350 ;
0360 ;
0370 ;*****
0380 ;Programm-Einsprung
0390 ;*****
0400 ;      = $0600 wie immer in Page 0
0410 ;
0420 ;      PLA      BASIC Einsprung KaJBR($36)
0430 ;      JMP VINIT Sprung zum NP-Aufang
0440 ;
0450 ;*****
0460 ;Display-List f. Scrollendes
0470 ;GRAPHICS 2-Display
0480 ;*****
0490 ;
0500 DLIST .BYTE #70,#70,#70 ;3 Leerzeilen
0510 .BYTE #MD+540 ;erste Zeile mit LMS-Bit
0520 LMSADR .WORD YMIN ;Adresse Bildspeicher
0530 .BYTE #MD,#MD,#MD,#MD
0540 .BYTE #MD,#MD,#MD,#MD
0550 .BYTE #MD,#MD,#MD,#MD-$20
0560 ;Achtung: letzte Zeile immer ohne Vscroll-Bit.
0570 .BYTE #61 ;ANTIC JMP-Anweisung
0580 .WORD DLIST ;zum Anfang der Disp.-List
0590 ;
0600 ; Interne Variable
0610 ;
0620 SVSCR .BYTE 0 ;Shadow fuer VSCRDL
0630 ;
0640 ;*****
0650 ;Initialisierungsroutine zum
0660 ;Einrichten der neuen Disp.-List
0670 ;und der VBI-Routine
0680 ;*****
0690 ;
0700 VINIT LDA BVHINQ255 LMS-Adresse auf
0710 STA LMSADR Anfang des Bildspeichers
0720 LDA #YMIN/256 richten
0730 STA LMSADR+1
0740 LDA #DLIST+255 neue Disp.-List
0750 STA #DLIST einrichten
0760 LDA #DLIST/256
0770 STA #DLIST+1
0780 LDA #0 Fein-Scrolling in Ausgangs-
0790 STA SVSCRDL position
0800 STA SVSCR auch Schattenregister!
0810 LDY #SVBIQ255 Scroll-Routine in.
0820 LDY #SVBI/256 den VBI einfügen
0830 LDA #? deferrted VBI genutzt
0840 JBR #SVBI ohne Kommentar...
0850 RTS
0860 ;
0870 ;*****
0880 ;VBI-Routine f. vert. Scrolling
0890 ;*****
0900 ;
0910 VBI CLD      wer rechnet schon decimal?

```

```

0641 AD7B02 0920 LDA STICK0 hat jemand am Knueppel gezogen?
0644 2901 0930 AND #1 etwa nach oben?
064b D006 0940 BNE TSTUNT nein, aber vielleicht nach unten?-->
064B 206106 0950 JSR DBEH Bildschirm nach oben scrollen-->
064B AC3B06 0970 JWP VBIEND Fertig!==>
064E AD7B02 0980 JSTUNT LDA STICK0 Fruedanknuappel Nr.0
0651 2902 1000 AND #2 nach unten gezogen?
0653 D003 1010 BNE VBIEND nichts los -->
0655 209206 1020 JSR UNTEN nach unten scrollen-->
065B AD1906 1040 VBIEND LDA SVSCR Schattenregister Fein-Scrolling
065B BD05B4 1060 STA VSCRDL in Hardwarereg. uebertragen
065E 4C42E4 1070 JWP XITUVV Ende der Vorstellung.
1080 ;
1090 ;
1100 ;
1110 ;
0661 AD1906 1120 DBEH LDA SVSCR Zeichengrenze?
0664 D00E 1140 BNE DBI nein, kein Anschlag zu befuerchten
0664 AD0B06 1150 LDA LMSADR sind wir schon
0669 C900 1170 CMP #YHIN/256 am oberen Anschlag?
066B D007 1180 BNE DBI nein, Scrolling moeglich-->
066D AD0906 1190 LDA LMSADR+1 MSB ueberpruefen
0670 C97F 1210 CMP #YHIN/256
0672 F01B 1220 BEQ DBEND Anschlag erreicht, kein Scrolling-->
0674 8E1906 1230 DBI INC SVSCR zuerst Fein-Scrolling versuchen
0677 AD1906 1250 LDA SVSCR
067A C910 1260 CMP #SVSPAX+1 Grob-Scrolling noetig?
067C 9013 1270 BCC DBEND nein, Fein-Scr. genuegt -->
067E A900 1290 LDA #0 Fein-Scr. ruecksetzen
0680 BD1906 1300 STA SVSCR
0683 1B 1310 CLC Grob-Scrolling vorbereiten
0684 AD0B06 1320 LDA LMSADR Bildschiraddress neu
0687 0914 1330 ADC #ZLAENB berechnen (eine Zeile
0689 BD0B06 1340 STA LMSADR nach oben)
068C 9003 1350 BCC DBEND
068E 8E0906 1360 INC LMSADR+1 MSB
0691 40 1370 DBEND RTS
1380 ;
1390 ;
1400 ;
1410 ;
0692 AD1906 1420 UNTEN LDA SVSCR Zeichengrenze?
0695 D00E 1440 BNE UNI nein, kein Anschlag zu befuerchten
0697 AD0B06 1450 LDA LMSADR sind wir schon
069A C900 1470 CMP #YHIN/256 am unteren Anschlag?
069C D007 1480 BNE UNI nein, Scrolling moeglich-->
069E AD0906 1490 LDA LMSADR+1 MSB ueberpruefen
06A1 C970 1510 CMP #YHIN/256
06A3 F01B 1520 BEQ DBEND Anschlag erreicht, kein Scrolling-->
06A5 CE1906 1530 DBI DEC SVSCR zuerst Fein-Scrolling versuchen
06A8 1013 1550 BPL UNEND Fein-Scr. genuegt -->
06AB A90F 1560 LDA #SVSPAX Fein-Scr. ruecksetzen
06AC BD1906 1570 STA SVSCR
06AF 3B 1580 SEC Grob-Scrolling vorbereiten
06B0 AD0B06 1590 LDA LMSADR Bildschiraddress neu
06B3 8E1A 1600 SBC #ZLAENB berechnen (eine Zeile
06B5 D006 1610 STA LMSADR nach unten)
06B8 D003 1620 BCC UNEND
06BA CE0906 1630 DEC LMSADR+1 MSB
06BD 40 1640 UNEND RTS
1650 ;
ASSEMBLY ERRORS: 0 33020 BYTES FREE

```

Listing 3

```

100 REM * LISTING 3:
110 REM * Feinscrolling in
120 REM * Maschinensprache
130 REM * Demo in BASIC PF'85
150 BA=28672:REM * Scrollber.:=$7000
160 ZL=20:REM * Zeilenlaenge
170 POKE 756,226:REM * Graphikzeichen
180 GOSUB 30000:REM * Maschinengm
190 A=USR(1536):REM * VBI ein!
200 REM * Scroll-Bereich mit Bild fuellen
220 FOR A=BA TO BA+204*ZL STEP ZL
230 E=INT(RND(0)*8)+A
240 FOR I=A TO E:POKE I,10:NEXT I
250 FOR I=E+1 TO E+10:POKE I,92:NEXT I
260 FOR I=E+11 TO A+19:POKE I,8:NEXT I
290 NEXT A:GOTO 200:REM und neu fuellen...
30000 REM * Maschinenprogramm fuer Feinscrolling
30010 S=0:RESTORE 30100
30020 FOR A=1536 TO 1725:READ D:POKE A,D:S=S+D:NEXT A
30030 IF S<>14814 THEN ? "DATEN-FEHLER!":STOP
30090 RETURN
30100 DATA 104,76,26,6,112,112,112,103,0,112,39,39,39,39,39,39,39
30110 DATA 39,39,39,39,7,65,4,6,0,169,0,141,8,6,169,112,141,9,6,169
30120 DATA 4,141,48,2,169,6,141,49,2,169,0,141,5,212,141,25,6,160
30130 DATA 64,162,6,169,7,32,92,228,96,216,173,120,2,41,1,208,6,32
30140 DATA 97,6,76,88,6,173,120,2,41,2,208,3,32,146,6,173,25,6,141
30150 DATA 5,212,76,98,228,173,25,6,208,14,173,8,6,201,0,208,7,173
30160 DATA 9,6,201,127,240,29,238,25,6,173,25,6,201,16,144,19,169
30170 DATA 0,141,25,6,24,173,8,6,105,20,141,8,6,144,3,238,9,6,96,173
30180 DATA 25,6,208,14,173,8,6,201,0,208,7,173,9,6,201,112,240,24
30190 DATA 206,25,6,16,19,169,15,141,25,6,56,173,8,6,233,20,141,8
30200 DATA 6,176,3,206,9,6,96

```