Simons Basic

2 Teil

Nachdem in der letzten Ausgabe die Strukturbefehle und Programmierhilfen behandelt wurden, wollen wir nun die Grafik-, Musik-, Bildschirm- und sonstigen Befehle mit Beispielen erläutern.

Die Grafik-Befehle sollen nicht in ihrer alphabetischen Ordnung besprochen werden, sondern in ihrer natürlichen Anordnung, wie sie eventuell auch eingesetzt werden könnten. Die Grafik-Befehle sind meines Erachtens für die Programmierung der hochauflösenden Grafik eine unabdingbare Voraussetzung. Dies heißt nicht, daß diese unbedingt dem Simons Basic entnommen werden müssen, jedoch sollten diese oder ähnliche Befehle unbedingt zur Programmierung herangezogen werden.

Ihr Vorteil liegt nicht nur in der einfacheren Programmierung, sondern — und dieser Vorteil ist nicht unerheblich — auch in der schnelleren Bearbeitung. Bei komplizierten Grafiken, in denen viele Punkte gesetzt werden müssen, wartet man nicht selten mehr als eine Stunde, bis diese mit Basic erstellte Grafik fertig ist. Durch die Grafik-Befehle ergibt sich hier eine zigfach höhere Geschwindigkeit.

HIRES

Mit diesem Befehl wird im Simons Basic die Grafik eingeschaltet. Gleichzeitig werden für einfarbige Grafiken die Hintergrund- und Punktfarbe festgelegt. Vergleichen Sie hierzu Bild 1. Es zeigt ein Basic-Programm, das das gleiche bewirkt, und ein entsprechendes Assemblerprogramm, das dem Buch »Das Commodore 64-Buch, Band 1« entnommen ist. Das Basic-Programm läuft zirka zwei Minuten, wohingegen das Assemblerprogramm und auch der Befehl HIRES keine merkbare Ausführungszeit beanspruchen. Auf die verschiedenen Speicher des Commodore 64 für die Farbgrafikdarstellung soll hier nicht näher eingegangen werden, jedoch wollen wir anmerken, daß beim Einschalten der Grafik zunächst der Farbspeicher (RAM-Bereich 1023-2022) und der Grafikspeicher für die Darstellung der einzelnen Punkte (RAM-Bereich 8192-16383) mit der Farbe beziehungsweise mit Nullen vorbesetzt werden. Näheres zum Thema Grafik finden Sie in der oben angeführten Buchreihe.

MULTI

Da der Befehl HIRES nur einfarbige Grafiken zuläßt, kann man mit dem Befehl MULTI weitere drei Farben gleichzeitig auf dem Bildschirm darstellen.

LOW COL

Mit diesem Befehl sind noch weitere drei Farben für den Multi-Color-Modus zuschaltbar. Während des Programms können zwar immer nur drei Farben angesprochen werden, jedoch erscheinen bis zu sechs Farben auf dem Bildschirm.

HI COL

Mit dem Befehl HI COL können die ursprünglich mit dem Befehl MULTI dargestellten Farben im Programm wieder eingeschaltet werden. Für sechs Farben sind also die Befehle LOW COL und HI COL entsprechend im Programm zu setzen.

PLOT

Der Befehl PLOT dient zur Ausgabe eines einzigen Punktes auf dem Bildschirm. Daß dies kein einfaches Unterfangen ist, wird an dem Basic-Programm in Bild 2 deutlich. Hierzu trägt der Aufbau des Grafikspeichers erheblich bei. Ebenfalls in Bild 2 dargestellt ist die Version bei Verwendung des Befehls aus Simons Basic.

LINE

Der Befehl LINE zeichnet eine Linie mit den angegebenen Parametern auf dem Bildschirm. In Bild 3 sind wieder die entsprechenden Programme in Basic und bei Verwendung des Befehls aus Simons Basic dargestellt. Wie man an dem Basic-Programm sieht, kann man natürlich den Befehl zum Setzen eines Punktes als Unterprogramm aufrufen. Das macht aber die Ausgabe der Linie nicht wesentlich schneller. Daß es auch in einem Basic-Programm nicht einfach mit der Ausgabe einer geraden Linie getan ist, sondern auch noch einige Umwandlungen zur korrekten Pro-

```
20006 REM ****************************************
20007 REM *    G R A F I K    E I N              *
20008 REM ****************************************
20020 POKEV+17,59
20030 POKEV+24,24
20040 FORLV=1024TO2023
20050 POKELV,HF+16*ZF
20060 NEXT
20070 FORLV=8192TO16383
20080 POKELV,0
20090 NEXT
20100 POKE53280,HF
20110 RETURN

READY.
```

Bild 1a. Einschalten der hochauflösenden Grafik in Basic

```
HIRES 6,7
```

Bild 1b. Grafik einschalten mit Simons Basic

```
20400 REM ***************************************
20401 REM *    P U N K T E   S E T Z E N        *
20402 REM ***************************************
20410 IF PX<0 OR PX>319 THEN RETURN
20420 IF PY<0 OR PY>199 THEN RETURN
20430 HX=8*INT(PX/8)
20440 HY=320*INT(PY/8)+(PY AND 7)
20460 BX=2↑(7-(PX AND 7))
20470 H1=8192+HX+HY
20480 POKEH1,PEEK(H1) OR BX
20490 RETURN
PLOT PX,PY
```

Bild 2. Punkt setzen in Basic und in Simons Basic

grammausführung notwendig sind, ist aus dem Listing ersichtlich.

REC, CIRCLE, ARC, ANGL, BLOCK

Ähnlich dem Ziehen einer Linie werden durch diese Befehle ein Rechteck (REC), eine Ellipse (der Kreis ist ein Sonderfall der Ellipse-CIRCLE), Segmente (ARC), Radien (ANGL) und ausgefüllte Rechtecke (BLOCK) gezeichnet. Für Rechtecke und Blöcke können wieder die entsprechenden Befehle zum Ziehen von Linien und zum Zeichnen eines Punktes als Unterprogramme herangezogen werden, wenn das ganze in Basic programmiert werden soll. Dabei kann man einen Block als Darstellung mehrerer paralleler Linien auffassen. Für die Ausgabe von Kreisen, Ellipsen und Segmenten sowie Radien müssen die entsprechenden trigonometrischen Funktionen Sinus und Cosinus herangezogen werden. Das Zeichnen eines Radius benötigt nur zur Bestimmung des Endpunktes diese Winkelfunktionen, und ist ansonsten analog dem Ziehen einer Linie.

PAINT

Mit dem Befehl PAINT läßt sich eine vorgegebene, geschlossene Figur (Rahmen) mit einer Farbe ausfüllen. Bei entsprechend komplizierten Figuren (Konvex und konkav gebogene Randstücke, Aussparungen in der Mitte) ist das schnelle Ausfüllen eines vorgegebenen Rahmens im Basic auch ein topologisches Problem, was selbst einem einigermaßen geübten Programmierer auf Anhieb nicht gelingen wird.

DRAW, ROT

Mit dem Befehl DRAW kann eine Figur aus lauter Linien zusammengesetzt werden, und mit dem Befehl ROT kann eine derartig gezeichnete Figur um verschiedene Winkel gedreht werden.

CSET

Dieser Befehl hat gleich drei Funktionen:
- Zeichensatz umschalten (von Grafik auf Groß/Kleinschrift beziehungsweise umgekehrt)
- Grafik zurückholen
- Eine mehrfarbige Grafik zurückholen und dabei diese Grafik mit anderen Farben versehen.

```
20600 REM ***************************************
20601 REM *    L I N I E N   Z I E H E N        *
20602 REM ***************************************
20610 IF EX=AX OR EY=AY THEN H2=1:GOTO20640
20620 IF ABS(EY-AY) < ABS(EX-AX) THEN H2=1: GOTO 20640
20630 H2=ABS(EX-AX)/ABS(EY-AY)
20640 IFEX<AXTHENH2=H2*-1
20650 H3=1
20660 IFEY<AYTHENH3=-1
20670 IFEX=AXTHEN20740
20680 FORLL=AXTOEXSTEPH2
20690 PX=INT(LL+.5)
20700 PY=((LL-AX)*(EY-AY)/(EX-AX))+AY
20710 GOSUB20400
20720 NEXT
20730 RETURN
20740 PX=AX
20750 FORPY=AYTOEYSTEPH3
20760 GOSUB20400
20770 NEXT
20780 RETURN
LINE AX,AY,EX,EY
```

Bild 3. Linie ziehen in Basic und in Simons Basic

CHAR, TEXT

Diese beiden Befehle sind besonders wichtig, da im Modus der hochauflösenden Grafik keine Buchstaben, Zahlen und sonstige Zeichen ausgegeben werden können. Diese beiden Befehle ermöglichen einerseits das Ausgeben einzelner Zeichen (CHAR) sowie ganzer Textzeilen (TEXT), wobei die Position, die Größe und die Farbe der Zeichen in dem Befehl selbst angewählt werden können.

Sprite-Befehle

Auch die Sprite-Befehle wollen wir in der Reihenfolge ihrer Verwendung besprechen.

DESIGN

Mit diesem Befehl wird dem Sprite Speicherplatz zugeteilt. Dies ist gegenüber dem normalen Basic keine wesentliche Erleichterung, da dies dort auch mit einem einzigen POKE-Befehl erledigt werden kann. Da man bei Simons Basic aber so gut wie gar nicht mehr auf POKE-Befehle zurückgreifen muß, ist dieser Befehl doch sinnvoll.

@

Nachdem der Speicher für ein Sprite zugeteilt wurde, muß als nächstes dieses definiert werden. Dies kann man durch 21 Zeilen, denen ein Klammeraffe vorangestellt ist und die jeweils eine Zeile des Sprites (12 oder 24 nebeneinanderliegende Punkte) enthalten.

CMOB

Dieser Befehl definiert die Farben eines Sprite.

MOB SET

Im Simons Basic bedeutet die Abkürzung MOB Moveable Objekt Block. Mit dem Befehl MOB SET werden die Eigenschaften eines Sprites festgelegt. Das sind die Farbe des Sprites, die Priorität gegenüber dem Hintergrund und ob hochauflösende Grafik oder Multicolor Grafik gewünscht wird.

MMOB, RLOCMOB

Diese beiden Befehle dienen zur Darstellung beziehungsweise zur Bewegung des Sprites. Dazu können Start- und Zielposition des Sprites angegeben werden, sowie die Geschwindigkeit mit der sich das SPRITE über den Bildschirm bewegen soll.

MOB OF

Hiermit wird das Sprite wieder ausgeschaltet.

DETECT, CHECK

Diese beiden Befehle dienen zur Vorbereitung einer Kollisionsprüfung und zur Kollisionsprüfung selbst. Auch diese beiden Befehle sind in Basic durch einen einfachen POKE-Befehl sehr schnell zu realisieren, sie ersparen einem aber die umständliche Suche, welches Byte im Video-Controller-Chip die entsprechende Aufgabe wahrnimmt, und das Auseinanderziehen der einzelnen Bits.

Musik-Befehle

Die Anwendung der Musik-Befehle und ihrer Eigenschaften werden am besten deutlich, wenn man sich das Beispiel aus dem Handbuch ansieht (Bild 4).

Befehle für Zeichenreihen

Die Befehle für Zeichenreihen ergeben sich aus der Befehlsübersicht im ersten Teil. Wir wollen hier nur zwei Befehle besonders hervorheben:

PLACE

Der Befehl PLACE wird immer da verwendet, wo eine kleinere Zeichenreihe in einer größeren gesucht wird. Dieser wichtige Befehl fehlte bisher bei allen Commodore Basic-Generationen. Sicherlich hat jeder Programmierer, der regelmäßig Programme schreibt, sich für diesen Befehl schon ein eigenes Unterprogramm geschrieben.

USE

Mit dem Befehl USE wird den Programmierern ein noch größerer Gefallen getan als mit dem Befehl INST. In vielen Programmen wünscht man sich eine formatierte Ausgabe von Zahlen. Bisher mußte dazu immer mühsam ein Basic-Unterprogramm geschrieben werden, das eine Zahl als Zeichenreihe behandelt und in eine kaufmännische Zahlendarstellung umwandelt. Was bei anderen Basic-Dialekten selbstverständlich ist, wurde für den Commodore Computer in Simons Basic realisiert.

Befehle für Zahlen

EXOR, MOD, DIV, FRAC, % und $

Die sechs auf Zahlen anwendbaren Befehle sind wohl für jeden Programmierer im mathematisch-technisch-wissenschaftlichen Bereich unentbehrlich. Dabei ist der Befehl EXOR kein eigentlicher Zahlenbefehl. Er bildet im Prinzip nur eine weitere logische Verknüpfung neben den schon vorhandenen UND, ODER und NOT. Trick-Programmierer benutzen diesen Befehl um zwei Zahlen bzw. Bit-Muster zu vertauschen ohne Zuhilfenahme einer dritten Variablen.

Der Befehl MOD ergibt den Rest einer Zahl nach einer Division (Mod (30,4) = 2, das heißt, 30/4 = 7 REST 2) DIV entspricht dem Basic-Befehl INT, kann jedoch nur die Vorkommazahl nach einer Division bestimmen (DIV (30,4) = 7)

FRAC hingegen ergibt die Nachkommastellen einer Zahl (FRAC (3.56) = .56)

Mit % wird eine Binärzahl in eine Dezimalzahl umgewandelt und $ bewirkt das Gegenteil.

Gerade bei den manigfaltigen Adressen im Video-Controller des Commodore 64, wo einzelne Bits in Registern gesetzt oder gelöscht werden müssen, um bestimmte Tätigkeiten zu erreichen oder zu unterlassen, sind diese Funktionen sehr nützlich.

Wie zu Beginn bereits erwähnt, werden die Befehle zur Bildschirmsteuerung von Programmierern verwendet, die auch das letzte aus ihrem Computer herausholen wollen. Simons Basic bietet dazu sehr viele Möglichkeiten.

FLASH, OFF, BFLASH, BFLASH 0

Diese Befehle bewirken das Blinken von Bildschirmfarben bzw. des Rahmens. Auch die Blinkfrequenz kann variiert werden.

FCHR, FCOL, FILL, INV

Mit diesen vier Befehlen können bestimmte Bildschirmbereiche mit Zeichen und/oder Farben gefüllt bzw. invertiert werden.

MOVE

Der MOVE-Befehl dupliziert Bildschirmbereiche. Bei geschickter Bildschirmausgabe durch Cursorsteuerung läßt sich auf diese Art und Weise auch eine fensterweise Ausgabe wie bei den Computern der Serie 8000 erreichen. In diesem Falle können sogar die Fenster in einen anderen Bereich kopiert werden. Viertelt man zum Beispiel den Bildschirm, so kann man ein Viertel für die normale Bildschirmausgabe verwenden, und der Anwender kann sich bei Bedarf diese Bildschirmausgabe in ein anderes Viertel kopieren um eventuell Datenvergleiche durchzuführen.

LEFT, RIGHT, UP, DOWN

Simons Basic erlaubt weiterhin Bildschirmbereiche zu rollen. Dies kann analog zu den oben angeführten Befehlen nach rechts, links oder nach oben und unten geschehen. Sehr interessant ist zum Beispiel, daß auch Bereiche gerollt werden können, so daß der Bildschirm teilweise erhalten bleibt und in anderen Bereichen des Bildschirms fortlaufend Daten angezeigt werden können. Das Bildschirmrollen kann sowohl zyklisch als auch durch Nachziehen von Leerzeilen bzw. -spalten erfolgen.

```
100 REM ********************************
110 REM *** BEISPIEL!                  ***
120 REM *** MUSIK                      ***
130 REM ********************************
140 REM *** LAUTSTAERKE EINSTELLEN     ***
150 VOL 10
160 REM *** WELLENFORM EINSTELLEN      ***
170 WAVE 1,00010000
180 REM *** HUELLKURVE EINSTELLEN      ***
190 ENVELOPE 1,8,8,8,0
200 REM *** FESTLEGEN DER NOTEN        ***
210 A$= "▒IZ■C5■E5■F5■"
220 A2$="G5▒C5■E5■F5■G5▒C5■E5■F5■G5▒"
230 A2$=A2$+"E5▒C5▒E5▒D5▒E5■E5■D5■C5▒"
240 A2$=A2$+"C5■E5▒G5▒G5■F5▒F5▒E5■F5■"
250 A2$=A2$+"G5▒E5▒C5▒D5▒C5▒C5■C5■E5■"
260 A2$=A2$+"F5■"
270 A3$="C5▒C5■"
280 MUSIC 8,A$+A2$+A2$+A3$
290 REM *** MUSIK SPIELEN              ***
300 PLAY 1
310 REM *** LAUTSTAERKE 0              ***
320 VOL 0
330 END
```

Ergebnis:

"WHEN THE SAINTS GO MARCHING IN"

wird gespielt.

Musik mit Simons Basic (entnommen aus dem Handbuch)

SCRSV, SCRLD

Bildschirminhalte im normalen Modus können mit diesen beiden Befehlen auf Diskette gespeichert beziehungsweise wieder geladen werden. Dies kann zum Beispiel interessant sein, wenn auf dem Bildschirm Balkengrafiken dargestellt wurden, deren Daten erst mühsam errechnet werden mußten. Zur nochmaligen Anzeige eines solchen Diagramms braucht dann keine neue Berechnung durchgeführt, sondern lediglich der Bildschirminhalt von Diskette geladen werden.

COPY, HRDCPY

Mit den beiden Copy-Befehlen sind sowohl die Bildschirmausgabe von hochauflösender Grafik (COPY) als auch eines Bildschirmes im Normalmodus (HRDCPY) auf einen Drucker möglich. Dies sind zwei interessante Befehle, besonders das Hardcopy der hochauflösenden Grafik, da dies eine relativ umständliche Druckerprogrammierung erfordern würde.

Befehle für LIGHTPEN, JOYSTICK und PADDLE

Mit den vier Befehlen zur Abfrage des Status der obengenannten externen Hilfsgeräten lassen sich sicherlich sehr einfach Spiele programmieren. Statt umständlicher IF-Abfragen im Programm können die Werte der externen Geräte durch diese Befehle sofort erfragt und somit auch gleich weiter verarbeitet werden.

Sonstige Befehle

DESIGN

Ähnlich den Sprites können auch die Zeichen des normalen Zeichensatzes neu definiert werden. Mit dem Befehl DESIGN wird zunächst festgelegt welches Zeichen erstellt werden soll und mit dem Befehl @ wird dieses Zeichen auch ähnlich wie bei den Sprites kreiert.

DIR, DISC

Mit dem DISC-Befehl können Befehle an die Floppystation unmittelbar übergeben werden, ohne OPEN, CLOSE und die entsprechenden Fehlerkanäle anzugeben. Mit DIR kann das Inhaltsverzeichnis einer Diskette ausgegeben werden, wobei auch Jokerzeichen zugelassen sind, so daß ein sogenanntes »Pattern Matching« möglich ist. Dies bedeutet, daß man sich Programme oder Dateien, die ganz bestimmte Buchstabenkombinationen enthalten, auszugsweise auflisten lassen kann.

FETCH

Der FETCH-Befehl ermöglicht es, kontrollierte Eingaben im Programm zuzulassen, ohne daß aufwendige INPUT-Routinen geschrieben werden müssen.

INKEY

Sehr wichtig ist auch der INKEY-Befehl, da er abfragt ob eine Funktionstaste gedrückt ist. Dies ermöglicht eine schnelle Bearbeitung durch den Anwender, da nicht immer RETURN nach einer Eingabe gedrückt werden muß, beziehungsweise eine umständliche Überprüfung über den GET-Befehl entfällt.

LIN

Normalerweise beginnt eine Routine zum Positionieren des Cursors immer in der linken oberen Bildschirmecke und steuert die gewünschte Cursorposition dann durch eine entsprechende Anzahl von »Cursor Right« und/oder »Cursor Down« an. Dies muß bei absolutem Cursorpositionieren auch dann erfolgen, wenn die Ausgabe bereits in der nächsten Zeile erfolgen soll. Mit dem LIN-Befehl ist auch eine relative Zeilenpositionierung des Cursors möglich.

MEM

Für Anwender, die ihren Zeichensatz selbst programmieren beziehungsweise diverse Zeichen verändern wollen, ist der MEM-Befehl interessant, da er ohne komplizierte USR-Funktion den ROM-Bereich des Zeichensatzes in den entsprechenden RAM-Bereich verlegt.

PAUSE

Sicherlich hat jeder Programmierer in seinem Programm irgendwo eine »leere« FOR...NEXT-Schleife, um eine Anzeige eine gewisse Zeit am Bildschirm aufrecht zu erhalten.

Einerseits ist das ein Programmiertrick, der die Dokumentation des Programms nicht wesentlich erleichtert, andererseits ist in diesen Schleifen nur eine ungenaue Zeitangabe möglich. Mittels des PAUSE-Befehls kann man nun das Programm für eine genau definierte Anzahl von Sekunden anhalten und sogar zusätzlich noch eine Meldung drucken.

RESET

Der RESET-Befehl ist nicht zu verwechseln mit irgendwelchen RESET-Tasten und/oder -Schaltern. In Simons Basic dient er zum Setzen eines Zeigers auf eine beliebige DATA-Zeile. Dies wird dann benötigt, wenn in einem Programm mehrere verschiedene DATA-Blöcke vorkommen, die teilweise oder ganz neu eingelesen werden müssen.

Zusammenfassung

Als Ganzes gesehen ist Simons Basic mit Sicherheit eine sinnvolle Basic-Erweiterung für den Commodore 64. Erfahrene Programmierer können die Befehle von Simons Basic sogar in den meisten Fällen ohne genaues Studium des Handbuches verwenden, da die Syntax der Befehle vollkommen ausreichend zu ihrer Erklärung ist. Jedoch empfiehlt es sich zu Beginn ein bißchen mit den Befehlen herumzuspielen. Auch für den Anfänger ist Simons Basic eine wertvolle Hilfe, jedoch dürfte für diesen es Anfangs neu sein, daß auch Basic-Befehle Parameter erhalten können, was zum Beispiel in hohem Maße bei den Grafik- und Sprite-Befehlen der Fall ist.

Die unter Programmierhilfen zusammengefaßten Befehle bilden eine weitgehend ausreichende Basis um Programme sehr schnell und methodisch austesten zu können beziehungsweise die Stuktur-Befehle ermöglichen eine übersichtlichere Programmstruktur, die sich dann einfacher dokumentieren läßt.

Insbesonders erwähnenswert scheinen mir auch die Programmschutz-Befehle, die zwar an sich keinen eigenen Programmschutz darstellen (unerlaubtes Kopieren ist immer noch möglich), aber in Verbindung mit anderen Schutzmaßnahmen sicherlich einen ausreichenden Schutz gegen unbefugte Programmnutzung darstellen können. Man denke hier nur an ein Passwort, was im Programm unkenntlich gemacht wird, so daß es nicht einfach dem Programm-Listing zu entnehmen ist.

Das Handbuch ist zwar klar und übersichtlich gegliedert und mit ausreichend erklärenden Beispielen versehen, jedoch wären manche Übersichtstabellen (wie zum Beispiel eine Zusammenfassung aller Befehle nach Bereichen geordnet mit Angabe ihrer Syntax/Übersicht von Befehlen mit Parametern) als Nachschlagewerk für den fortgeschrittenen Programmierer wünschenswert.

Fazit: Simons Basic ermöglicht es, den Commodore 64 effizient in allen seinen Möglichkeiten zu nutzen.

(H.-L. Schneider)