

# Bewegte Grafik und Textmischen

**Wer an Sprites denkt, stellt sich meist kleine Männchen oder Raumschiffe vor. Aber auch beim Einsatz in Anwendungsprogrammen können Sprites zur grafischen Illustration sehr nützlich sein.**

Als ich die Grafik entwarf, wollte ich ein Programm unter folgenden Gesichtspunkten schreiben: kein Spielkram und von Basic aus verwendbar. Damit diese Erweiterung auch mit anderen Basic-Erweiterungen ohne weiteres funktioniert, sollten die Befehle über »SYS« aufgerufen werden. Nun sollte es kein Allerweltsprogramm sein, sondern irgend etwas Neues. Da fielen mir die Sprites ein. Mit dem richtigen Programm könnten damit kleine aber deutliche Grafiken erstellt werden, zum Beispiel Niederschlags-Temperatur- oder Umsatztabellen. Als ideal fand ich vier Sprites hintereinander und das zweimal untereinander. Das gibt eine Auflösung von immerhin 95 x 42 = 4032 Punkten. Damit läßt sich schon einiges machen. Zur Handhabung werden vier Routinen benötigt. Eine zum Einschalten der Grafik, eine zum Setzen beziehungsweise Löschen der Punkte und eine zum schnellen Löschen der gesamten Grafik. Weiterhin noch eine, die die ganze Grafik auf eine Farbe setzt.

## Das Einschalten und Positionieren der Grafik

Dazu sind vier Parameter nötig, zwei für die x,y-Koordinaten und zwei für die x,y-Ausdehnung, also das Spreizen der Sprites. Je nachdem, ob gespreizt oder nicht, werden die Koordinaten berechnet und in die jeweiligen Register abgelegt. Zum Schluß werden noch alle Sprites aktiviert. Der Einfachheit halber kann x nur Werte zwischen 0 und 255 haben. Bei x=255 und ungespreizt ragt die Grafik sowieso schon aus dem Bildschirm heraus. Auch sollte man mit dem y-Wert nicht zu hoch gehen, da sonst die Grafik möglicherweise oben in den Bildschirm hereinkommt. Diese Einschränkungen sind aber in den meisten Fällen unwesentlich.

## Das Setzen beziehungsweise Löschen von einem Punkt

Dazu muß genau das Byte errechnet werden, in dem sich der Punkt befindet. Dafür braucht man nur die obere Spritereihe zu betrachten. Sie ist genau eine Speicherseite (256 Byte) groß. Sollte nun die y-Koordinate größer als 20 sein, so wird einfach eine Seite weitergeblättert, das heißt das Hi-Byte des Zeigers inkrementieren. Als erstes wird das Sprite errechnet. Es wird solange # 24 (= eine Spritelänge) vom x-Wert abgezogen bis der x-Wert kleiner als 24 ( $24 < x <= 0$ ) ist. So ist man praktisch im ersten Spriteblock gelandet. Nun wird vom x-Wert solange # 8 abgezogen, bis der x-Wert kleiner als 8 ( $8 < x <= 0$ ) ist. Jetzt weiß man, welches Bit gesetzt werden soll, und welches Byte in der Reihe gemeint ist (eins von dreien).

Nun der y-Wert. Ist er größer als 20, dann muß das Hi-Byte des Zeigers inkrementiert und vom y-Wert 21 abgezogen werden. Man will ja die Byteposition in einem Spriteblock errechnen. Ist der y-Wert jetzt noch größer als 0 so müssen noch einmal  $3 * y$  dazugezählt werden, denn ein Sprite hat die Ausdehnung von 3 Byte. Damit hat man das Byte, das in einem Spriteblock verändert werden soll. Nun wird das Low-Byte des jewei-

ligen Sprites dazuzugewandt und in der Zeropage für den Zeiger abgelegt. Nachdem das Bytemuster hergestellt ist, kann nun der Punkt gelöscht beziehungsweise gesetzt werden.

Das Löschen der Grafik und Setzen der Farbe versteht sich wohl von selbst. Das Ausschalten der Grafik geschieht durch POKE 53269,0.

Die Befehlsliste sieht dann so aus:

```

SYS PLOT , 0 , x-Wert , y-Wert      Punkt löschen
SYS PLOT , 1 , x-Wert , y-Wert      Punkt setzen
SYS SET   , x-Pos. , y-Pos. ,
                                     x-spreizen , y-spreizen  Grafik ein
SYS COLOR ,                               Farbcode Farbe setzen
SYS CLS   ,                               Grafik löschen
    
```

```

10 REM*****
20 REM*                S-GRAFIK                *
30 REM*                FREDERIC ESPITALIER      *
40 REM*****
50 FORA=49152T049568:READX:POKEA,X
60 C=C+X:NEXTA
70 IF C<>52597THENPRINT"FEHLER IN DATAS"
:STOP
80 SYS49547
100 DATA32,253,174,32,158,183,224,96,48,
6,162,14,32,55,164,96,142,159,193,32
110 DATA253,174,32,158,183,224,42,16,237
,142,160,193,162,0,173,159,193,201
120 DATA24,48,6,56,233,24,232,208,246,14
1,159,193,169,8,133,188,189,155,193
130 DATA133,187,173,160,193,201,21,48,11
,230,188,173,160,193,56,233,21,141
140 DATA160,193,173,159,193,162,0,201,7,
48,8,240,6,56,233,8,232,208,244,41,7
150 DATA73,7,168,169,1,136,48,3,10,208,2
50,141,161,193,173,160,193,240,8,10
160 DATA24,109,160,193,141,160,193,138,2
4,109,160,193,24,101,187,133,187,96
170 DATA32,0,192,173,161,193,160,0,17,18
7,145,187,96,32,253,174,32,158,183
180 DATA224,2,48,3,76,10,192,224,1,240,2
26,32,0,192,173,161,193,73,255,160,0
190 DATA49,187,145,187,96,169,0,162,0,15
7,0,8,157,0,9,232,208,247,96,32,253
200 DATA174,32,158,183,138,162,8,157,38,
208,202,208,250,96,160,39,162,8,152
210 DATA157,247,7,136,202,208,248,32,253
,174,32,158,183,142,159,193,32,253
220 DATA174,32,158,183,142,160,193,32,25
3,174,32,158,183,224,2,48,3,76,10
230 DATA192,189,149,193,141,29,208,169,3
4,141,162,193,189,151,193,141,161
240 DATA193,162,0,142,16,208,173,159,193
,157,0,208,157,8,208,224,6,240,28,24
250 DATA109,161,193,205,159,193,176,11,7
2,173,16,208,13,162,193,141,16,208
260 DATA104,232,232,14,162,193,76,21,193
,32,253,174,32,158,183,224,2,48,3,76
270 DATA10,192,189,149,193,141,23,208,17
3,160,193,141,1,208,141,3,208,141,5
280 DATA208,141,7,208,24,125,153,193,141
,9,208,141,11,208,141,13,208,141,15
290 DATA208,169,255,141,21,208,96,169,0,
162,0,189,0,8,73,255,157,0,8,189,0,9
300 DATA73,255,157,0,9,232,208,237,96,16
9,10,133,44,141,130,2,76,148,227,0
310 DATA255,24,48,21,42,0,64,128,192,234
,0
READY.
    
```

Bild 1. Der Basic-Lader für die S-Grafik

