

»Windows« für den C 64

Dieses Programm für den C 64 stellt vier Speicher für Bildschirminhalte im Text/Blockgrafikmodus zur Verfügung. Da diese Speicher im RAM »unterhalb« des Betriebssystems liegen, beanspruchen sie keinen zusätzlichen Speicherplatz. Das Programm ist in Maschinensprache geschrieben und durch Verwendung eines Verschiebeladers in verschiedenen Speicherbereichen lauffähig.

Besonders in größeren Programmen ist es oft wünschenswert, zwischen mehreren Bildschirmen umschalten zu können oder aber Fenster mit Zusatzinformationen in den laufenden Bildschirm einblenden zu können, ohne jedoch seinen ursprünglichen Inhalt zu zerstören. Um dies zu ermöglichen, stellt das Programm vier Speicher zur Verfügung, in denen komplette Bildschirminhalte oder Teile daraus abgelegt werden können. Ein Speicher belegt einen Bereich von 2 KByte, jeweils ein KByte für Zeichen und Farbe. Da diese Speicher unterhalb des Betriebssystems im RAM ab Adresse \$E000 untergebracht sind, beanspruchen Sie keinerlei zusätzlichen Platz.

Anwendung

Das Programm bietet folgende Betriebsarten:

— STORE:	SYS BA, NR, XO, YO, FB, FH	Speichern
— RECALL:	SYS BA + 3, NR	Rückholen
— SWAP:	SYS BA + 6, NR	Austauschen

Die Parameter haben folgende Bedeutung:

BA	= Basisadresse des Programms (siehe unten)
NR	= Nummer des angesprochenen Speichers (0...3)
XO, YO	= Position der linken oberen Ecke des zu speichernden Bereiches (0..39, 0..24)
FB	= Breite des Bereiches (1...40)
FH	= Höhe des Bereiches (1...25)

Die Fensterseite (Höhe) darf nur bis zum rechten (unteren) Bildrand reichen. Wenn zum Beispiel die linke obere Ecke auf die Koordinate X = 10 und Y = 5 gesetzt ist, so ist die maximale Fensterbreite 30 und die maximale Fensterhöhe 20. Falsche Angaben für die Parameter erzeugen die Fehlermeldung »ILLEGAL QUANTITY ERROR«.

STORE: Um einen kompletten Bildschirm abzuspeichern, geben Sie folgendes ein: SYS BA, NR, 0, 0, 40, 25. Mit anderen Parametern erzeugen sie ein Bildschirmfenster, das bei RECALL auch nur den durch diese Parameter definierten Bereich über-

schreibt. Darum bleibt der ursprüngliche Bildschirm erhalten. **RECALL:** Es ist nur die Angabe der Speichernummer nötig. Der gespeicherte Bereich wird an seine ursprüngliche Position zurückgeschrieben.

SWAP: Wenn Sie ein schon vorher vorbereitetes und im Speicher stehendes Informationsfenster in den aktuellen Bildschirm einblenden, wird dessen ursprünglicher Inhalt zerstört. Um ihn zu erhalten, müßten Sie ihn vor der Einblendung in einen noch freien Speicher retten. SWAP erleichtert die Arbeit und spart Speicherplatz, indem es die Inhalte von aktuellem Schirm und Speicher austauscht. Mit einmal SWAP rufen sie den Speicher ab, noch einmal SWAP erzeugt wieder das ursprüngliche Bild. Die Information über Position und Größe des auszutauschen Bereiches liefert das Bild im Speicher. SWAP wird daher nur mit der Speichernummer aufgerufen.

Noch ein Hinweis: Vermeiden Sie es, mit RECALL oder SWAP einen Speicher aufzurufen, in den noch nichts hineingeschrieben wurde, dies würde zum Absturz des Programms führen (siehe: Kompatibilität zu Simons-Basic). Der Basic-Lader initialisiert alle 4 Speicher, um einer Fehlbedienung vorzubeugen.

Laden des Programmes

Die Verschieberoutine des Basic-Laders gestattet es, das Programm auf den Anfang jeder (sinnvollen) Speicherseite zu legen. Nach dem Start bietet der Lader die Optionen:

1	RAM-Ende normales Basic	(\$9E00)
2	RAM-Ende Simons-Basic	(\$7E00)
3	Geschützter Bereich	(\$C000)
> 7	Eingabe der Startseite	

Die Optionen 1 bis 3 setzen den Programmanfang auf die angegebene Adresse, bei Eingabe einer Zahl > 7 wird diese als die Nummer derjenigen Speicherseite interpretiert, wo das Programm abgelegt werden soll.

Aus der Nummer der Speicherseite erhält man durch Multiplikation mit 256 die noch fehlende Basisadresse. Das Ladeprogramm gibt sie am Schluß mit aus.

Der Vektor für die Basic-Obergrenze in (55), (56) wird entsprechend der Startadresse des Maschinenprogramms heruntersetzt, sofern das Programm nicht im geschützten Bereich untergebracht ist. Beachten Sie, daß Sie diesen Vektor nach einem Reset beziehungsweise STOP/RESTORE korrigieren, sonst wird das Programm überschrieben.

Im Programmlisting erkennen Sie zwei Unterschiedlich umfangreiche Blocks von DATA-Zeilen. Der erste, große Block stellt das eigentliche Programm dar, der kleine ist eine Liste von Adressen, welche relativ zum Programmanfang diejenigen Bytes angeben, die bei einer Verschiebung korrigiert werden müssen. Der Korrekturwert ergibt sich aus der Differenz zwischen der Startadresse, mit der das Programm assembliert wurde (\$9E00=Speicherseite 158) und der gewünschten Startseite. Die beiden Blocks besitzen eigene Checksummen und OK-/Fehlermeldungen.

Kompatibilität zu Simons-Basic

»Bildspeicher« ist auch eine hilfreiche Erweiterung der Simons-Basic-Befehle für die Bildschirmsteuerung. Es sind jedoch folgende Punkte zu beachten:

— Die Hires-Grafik überschreibt alle vier Speicher. Diese müssen daher vor einem Zugriff mit »RECALL« oder »SWAP« unbedingt neu beschrieben werden.

— Der geschützte Bereich ab \$C000 wird ebenfalls von Hires überschrieben, so daß »Bildspeicher« in Programmen, welche die Hires-Grafik benutzen, am Basic-RAM-Ende untergebracht werden muß (Option 2).

Ob Sie jedoch nun mit oder ohne Simons-Basic arbeiten, in jedem Falle ist »Bildspeicher« eine nützliche Routine, die auch dem C 64-Besitzer das Fenster zum »Windowing« aufstößt.

(Hans-Herbert Hagedorn / ev)



Listing »Bildspeicher«

```

100 REM ***** <143>
110 REM * * * * * <081>
120 REM * BILDspeICHER * <201>
130 REM * * * * * <101>
140 REM * H. H. HAGEDORN * <163>
150 REM * * * * * <121>
160 REM * RUPPRECHTSTR. 30 * <202>
170 REM * * * * * <141>
180 REM * 83 LANDSHUT * <101>
190 REM * * * * * <161>
200 REM * TEL. 0871/67337 * <199>
210 REM * * * * * <181>
220 REM ***** <007>
230 : <032>
240 PRINT "[CLEAR]":PRINT:PRINT <006>
250 PRINT "[SPACES, RVSON]BILDspeICHER-POSITION <249>
[SPACE]"
260 PRINT:PRINT "[SPACE3]NORMAL-BASIC[SPACE]= <206>
[SPACE]#9E00[SPACE3](1)"
265 PRINT:PRINT "[SPACE3]SIMONS-BASIC[SPACE]= <226>
[SPACE]#7E00[SPACE3](2)"
270 PRINT:PRINT "[SPACE3]GESCH. BEREICH=[SPACE] <001>
#C000[SPACE3](3)"
275 PRINT:PRINT "[SPACE3]PAGE-ADRESSE[SPACE] <024>
EINGEBEN[SPACE](>7)"
280 PRINT:PRINT TAB(25):INPUT L <195>
290 IF L=1 THEN L=158 <032>
300 IF L=2 THEN L=126 <038>
305 IF L=3 THEN L=192 <047>
310 IF L>7 AND L<255 THEN D=158-L:GOTO 320 <202>
315 PRINT"???:GOTO 280 <051>
320 POKE 2,D+100:POKE 780,L:IF L<161 THEN POKE <177>
56,L:POKE 55,0
325 CLR:I=PEEK(780)*256:J=I <145>
330 READ A:IF A>=0 THEN POKE I,A:I=I+1:S=S+A <207>
:GOTO 330
340 IF S<>S1530 THEN PRINT"DATENFEHLER",S:END <116>
345 PRINT"PROGRAMMDATEN[SPACE]OK":S=0 <239>
:D=PEEK(2)-100:IF D=0 THEN 380
350 READ P:IF P>=0 THEN A=J+P:POKE A,PEEK(A)-D <036>
:S=S+P:GOTO 350
360 IF S<>8737 THEN PRINT"BLOCKKORREKTURFEHLER", <043>
S:END
    
```

```

370 PRINT:PRINT"BLOCKKORREKTUR[SPACE]OK" <241>
380 PRINT:PRINT"BASISADRESSE[SPACE]J=[SPACE]J" <104>
<104>
385 FOR I=0 TO 3:SYS J,I,0,0,40,25:NEXT <051>
390 DATA 076,009,158,076,087,158,076,091 <035>
400 DATA 158,032,110,158,224,004,176,066 <017>
410 DATA 142,163,159,032,110,158,224,040 <019>
420 DATA 176,056,142,159,159,032,110,158 <047>
430 DATA 224,025,176,046,142,160,159,032 <047>
440 DATA 110,158,224,000,240,036,024,138 <041>
450 DATA 109,159,159,201,041,176,027,202 <069>
460 DATA 142,161,159,032,110,158,224,000 <063>
470 DATA 240,016,024,138,109,160,159,201 <080>
480 DATA 026,176,007,142,162,159,032,116 <097>
490 DATA 158,096,162,014,076,058,164,169 <130>
500 DATA 000,240,002,169,255,141,166,159 <114>
510 DATA 032,110,158,224,004,176,235,142 <118>
520 DATA 163,159,032,197,158,096,032,253 <156>
530 DATA 174,076,158,183,032,005,159,032 <156>
540 DATA 076,159,160,003,185,159,159,145 <177>
550 DATA 091,136,016,248,024,165,091,105 <170>
560 DATA 004,133,091,174,162,159,172,161 <180>
570 DATA 159,177,087,145,091,177,089,145 <222>
580 DATA 093,136,016,245,032,112,159,202 <193>
590 DATA 208,236,096,172,161,159,177,087 <234>
600 DATA 072,177,091,145,087,104,145,091 <229>
610 DATA 177,089,072,177,093,145,089,104 <255>
620 DATA 145,093,136,016,233,032,112,159 <236>
630 DATA 202,208,224,240,059,120,165,001 <233>
640 DATA 072,169,053,133,001,032,076,159 <003>
650 DATA 160,003,177,091,153,159,159,136 <023>
660 DATA 016,248,032,005,159,024,169,004 <021>
670 DATA 101,091,133,091,174,162,159,173 <035>
680 DATA 166,159,208,183,172,161,159,177 <067>
690 DATA 091,145,087,177,093,145,089,136 <078>
700 DATA 106,245,032,112,159,202,208,236 <056>
710 DATA 104,133,001,088,096,032,067,159 <076>
720 DATA 024,172,160,159,240,016,173,164 <082>
730 DATA 159,105,040,141,164,159,144,003 <088>
740 DATA 238,165,159,136,208,240,024,173 <112>
750 DATA 164,159,109,159,159,141,164,159 <137>
760 DATA 133,087,133,089,144,003,238,165 <132>
770 DATA 159,024,173,136,002,109,165,159 <141>
780 DATA 133,088,024,169,216,109,165,159 <161>
790 DATA 133,090,096,169,000,141,164,159 <159>
800 DATA 141,165,159,096,169,000,133,091 <139>
810 DATA 133,093,169,224,133,092,169,228 <189>
820 DATA 133,094,174,163,159,240,016,024 <186>
830 DATA 160,008,152,101,092,133,092,152 <182>
840 DATA 101,094,133,094,202,208,243,096 <203>
850 DATA 024,169,040,101,087,133,087,144 <214>
860 DATA 002,230,088,024,169,040,101,089 <218>
870 DATA 133,089,144,002,230,090,056,173 <231>
880 DATA 161,159,101,091,133,091,144,002 <233>
890 DATA 230,092,056,173,161,159,101,093 <255>
900 DATA 133,093,144,002,230,094,096,000 <253>
905 DATA -1 <109>
910 REM KORREKTURTABELLE <226>
920 : <213>
930 DATA 002,005,008,011,018,021 <129>
940 DATA 028,031,038,041,050,058 <158>
950 DATA 061,070,077,080,095,098 <187>
960 DATA 105,108,118,121,126,141 <174>
970 DATA 144,158,165,191,207,214 <202>
980 DATA 220,230,233,238,252,263 <200>
990 DATA 267,272,277,282,289,292 <246>
1000 DATA 295,304,311,319,327,330 <229>
1010 DATA 348,393,405,-1 <049>
    
```