

Handbuch HRG-ms 2.7



Eigenschaften

Die Software HRG-ms ist eine Grafikerweiterung für den Sinclair ZX81. Mit dieser Software können Grafiken mit 256x192 Pixel sowohl auf dem Bildschirm als auch auf dem Drucker ausgegeben werden. Als Hardware-Voraussetzung ist lediglich ein grafikfähiger Speicher nötig.

Die Software beinhaltet folgende Grundfunktionalitäten:

- Grafiktreiber
- Funktionenbibliothek
- Basic-Schnittstelle
- Assembler-Schnittstelle
- Hotkeys für Bildschirmumschaltung
- Hotkey für Bildschirmausdruck
- Hotkey für UDG-Test

Dabei wurde besonderer Wert auf eine einfache und verständliche Bedienbarkeit gelegt. Die einzelnen Basic-Aufrufe sind übersichtlich und selbsterklärend.

Details

Lage des Programms

Die Software installiert sich zu Beginn einmalig in den Speicher über RAMTOP und braucht daher im Basic-Programm nicht enthalten zu sein. Das Programm stellt dazu RAMTOP selbsttätig um.

Tastenbedienung

Während des Programmlaufs kann zwischen Grafik- und Text-Bildschirm hin und her geschaltet werden. Das Programm läuft dabei weiter. Im Eingabemodus bzw. im Listing kann damit ebenfalls auf die Grafik geschaut werden. Das ist besonders bei der Fehlersuche recht hilfreich.

Bildschirmausdruck

In Version 2 kann per Tastendruck zu jeder Zeit ein Ausdruck des gerade angezeigten Bildschirms gemacht werden. Auch während ein Programm läuft kann der Text-Bildschirm oder eine der Grafik-Bänke ausgedruckt werden.

Eingaben

Es können Eingaben im Programmlauf und auch im Direktmodus gemacht werden, während eine Grafik angezeigt wird.

Fehlermeldungen

Bei einem Programmfehler schaltet die Anzeige automatisch in den Textbildschirm, damit man die Fehlermeldung dort sofort sehen kann.

Maschinensprache-Schnittstelle

Über eine Schnittstelle können andere Maschinenprogramme die Grafikfunktionen benutzen.

Installation

Vor der Installation der Grafikerweiterung muss RAMTOP auf 32768 stehen. Am einfachsten erreicht man das mit einem Reset oder durch Power-On.

Nach dem Laden des Programms HRG-16k.p bzw. HRG-64k.p startet dieses automatisch und zeigt das folgenden Startbild.



Sofern nun keine Fehlermeldung erscheint, hat sich das Programm erfolgreich installiert. Es wird automatisch der neue Wert für RAMTOP gesetzt. Nach dem Betätigen von BREAK wird der neue Wert von RAMTOP im Textmodus angezeigt.

Sie können danach sofort ein anderes Programm ohne NEW von Cassette laden oder mit NEW das Basic-Programm löschen und ein neues Programm eingeben.

Falls vor dem Laden des Programms RAMTOP nicht den korrekten Wert hatte, erscheint eine entsprechende Fehlermeldung. Das Programm installiert sich dann nicht und es erscheint auch kein Startbild.

Deinstallation

Nach einem Reset oder Power-On des ZX81 ist die Grafikerweiterung nicht mehr vorhanden.

Speicherbelegung

Es existieren zwei Varianten, die sich jeweils in der Nutzung des Speichers unterscheiden. Wenn nur 16k Speicher vorhanden sind, muss die 16k Variante benutzt werden. Auf die Bankumschaltung muss dann verzichtet werden. Ist mehr als 16k Arbeitsspeicher vorhanden, so kann man beide Varianten benutzen. Es kann dann also auch die 16k Variante mit Bankumschaltung genutzt werden.

	16k Variante	64k Variante
Dateiname	HRG-16k.p	HRG-64k.p
Einsprung-Adresse für Basic (bei beide Varianten gleich)	32736	
RAMTOP stellt sich beim Start automatisch auf	22960	30160
damit für Basic verbleibender Speicher	6576 Bytes	13776 Bytes
Nummern der nutzbaren Grafik-Bänke bei Programmstart aktiv*	3* (24k-32k) 4 (32k-40k) 5 (40k-48k) 6 (48k-56k) 7 (56k-64k)	4* (32k-40k) 5 (40k-48k) 6 (48k-56k) 7 (56k-64k)
kann mit den folgenden Speichererweiterungen benutzt werden	16k 32k 48k 64k	32k 48k 64k

Prinzipiell ist die 16k Variante universeller, da sie bis zu 5 Grafik-Bänke nutzen kann. Sie hat aber den Nachteil, dass weniger Speicher für das Basic-Programm verfügbar ist.

Achtung!

Die 16k Variante kann bei einem 16k Speicher auch die nicht vorhandenen Bänke einschalten. Dort wird man dann den im jeweiligen Adressraum erreichbaren Speicher grafisch sehen können. Wegen der unvollständigen Adressdecodierung des Speichers kann so auch auf den Basicspeicher zugegriffen werden.

Das Anzeigen dieses Speichers ist gefahrlos. Aber ein schreibender Zugriff in diesen Speicher (CLEAR, PIX,TEXT, etc.) führt sofort zum Systemabsturz.

Basic-Schnittstelle

Jeder Funktionsaufruf ist in einem PRINT Kommando eingebettet.
Die Syntax ist:

```
PRINT USR 32736 ,FUNKTION
```

oder:

```
PRINT USR 32736 ,FUNKTION ,X ,Y ,...
```

Die Liste der möglichen Basic Funktionen ist im Abschnitt
Funktionen Tabelle beschrieben.

Empfehlenswert ist es, für die Einsprungadresse eine Variable zu benutzen. Im folgenden Beispiel wurde wegen der besseren Lesbarkeit der Name HRG gewählt. Es ist jeder andere gültige Variablenname möglich.

Beispiel:

```
10 LET HRG=32736
20 PRINT USR HRG ,ON
30 PRINT USR HRG ,PIX ,255 ,191
```

Die Funktionsnamen (hier ON und PIX) brauchen keine Variablen zu sein. Die Basic-Schnittstelle versucht nicht, eine Variable mit diesem Namen zu referenzieren.

In dem Beispiel werden die Funktionen ON und PIX aufgerufen. Dabei benötigt die Funktion PIX weitere Parameter, die jeweils mit Kommas getrennt dahinter stehen.

Diese Parameter müssen als Zahlen oder als Variablen oder als arithmetischer Ausdruck berechenbar sein. Der sich ergebende Wert des Parameters muss zudem in einem Wertebereich liegen, der für die jeweilige Funktion erlaubt ist. Bei einer Überschreitung des Wertebereichs wird das Programm mit einer Fehlermeldung abgebrochen.

Funktionen Tabelle

Eine ausführliche Beschreibung der Funktionen ist im Kapitel Funktionen zu finden.

Funktionsname	Parameter	
PIX,	X,Y	Pixel bei x,y setzen, löschen, invertieren
UNPIX,	X,Y	
XPIX,	X,Y	
LINE	X,Y,U,V	Linie von x,y nach u,v setzen, löschen, invertieren
UNLINE	X,Y,U,V	
XLINE	X,Y,U,V	
LINETO	X,Y	Linie nach x,y setzen, löschen, invertieren
UNLINETO	X,Y	
XLINETO	X,Y	
BOX	X,Y,U,V	Rechteck mit Ecken bei x,y und u,v setzen, löschen, invertieren
UNBOX	X,Y,U,V	
XBOX	X,Y,U,V	
CIRCLE	X,Y,R	Kreis mit Mittelpunkt bei x,y und Radius r setzen, löschen, invertieren
UNCIRCLE	X,Y,R	
XCIRCLE	X,Y,R	
POLY	X,Y,X,Y,...	Zeichnet eine geschlossene Kontur mit beliebig vielen Ecken x,y
UNPOLY	X,Y,X,Y,...	
XPOLY	X,Y,X,Y,...	
ON		Schaltet Grafikanzeige an
OFF		Schaltet Grafikanzeige aus
INVERT		Invertiert die Grafik
TEXT		Textbildschirm in die Grafik kopieren, löschen, invertieren oder mit Hintergrund kopieren
UNTEXT		
XTEXT		
OVERTEXT		
UDGNEW		Setzt den Zeichensatz zurück
UDGSET	C,B0,B1,..,B7	Definiert ein Zeichen
HCOPY		Druckt die Grafik aus
CLR		Löscht die Grafik
BANK	N	Selektiert Bank

Tastatur-Funktionen

Es gibt vier praktische Tastenfunktionen, die permanent zur Verfügung stehen.

Umschalten zwischen Text und Grafik



Zum schnellen Umschalten zwischen Grafik- und Text-Bild müssen die Tasten 9 und 0 (GRAPHICS und RUBOUT) rechts oben auf der Folientastatur gleichzeitig betätigt werden. Dabei muss die SHIFT-Taste nicht gedrückt werden! Bei jeder Betätigung schaltet die Ansicht von Text zur Grafik oder umgekehrt.

Dies geht auch dann, wenn man sich das Programmlisting ansieht. So kann man das grafische Ergebnis und den dazugehörigen Programmteil abwechselnd ansehen.

Wenn ein Programm zusätzlich zur Grafik auf dem Textbildschirm weitere Daten ausgibt, kann der Nutzer damit zwischen Grafik und Zusatzinformation hin und her schalten.

Ein möglicherweise laufendes Programm wird dadurch nicht unterbrochen.

Umschalten zwischen den Grafik-Bänken

Wenn mehr als 16k Byte Speicher verfügbar ist, kann ein Basic-Programm mit der BANK Funktion zwischen mehreren Grafikseiten umschalten und so mehrere Grafiken erstellen.

Um die verschiedenen Grafik-Bänke am Bildschirm schnell durchzusehen, kann die Bildschirmansicht per Tastatur auf eine beliebige Grafik-Bank umgeschaltet werden. Dabei kann das Programm in der mit BANK ausgeählten Grafik ungestört weiterarbeiten.



Zum Umschalten muss, während die 9 und 0 (GRAPHICS und RUBOUT) gedrückt sind, eine der Zifferntasten 3, 4, 5, 6 oder 7 zusätzlich betätigt werden. Die Zifferntaste entspricht dabei der gewünschten Grafik-Bank.

Hinweis: In der 64k Variante ist Taste 3 wirkungslos.

Ein möglicherweise laufendes Programm wird dadurch nicht unterbrochen.

Bildschirm Ausdruck mit Hotkey

Zum Ausdrucken des Bildschirms muss, während die Tasten 9 und 0 (GRAPHICS und RUBOUT) gedrückt sind, die Taste C zusätzlich betätigt werden. Der zu dem Zeitpunkt angezeigte Bildschirm (Grafik-Bank oder Text-Bildschirm) wird dann auf den Drucker ausgedruckt.

Dies kann auch mitten im Programmablauf ausgelöst werden.

Sofern der Ausdruck nicht mit BREAK abgebrochen wird, läuft das Programm danach weiter.

UDG Test mit Hotkey

Zum Testen von UDGs (User Defined Graphics) kann eine spontane Textkopie in den aktuellen Grafikbildschirm ausgelöst werden. Dazu muss, während die Tasten 9 und 0 (GRAPHICS und RUBOUT) gedrückt sind, die Taste T zusätzlich betätigt werden.

Die Taste T muss dabei gedrückt bleiben! Solange die Taste gedrückt ist, zeigt das System den Grafikbildschirm mit den UDGs an. Wird sie losgelassen, schaltet das System wieder in den Textmodus zurück.

Dieser Hotkey ist immer dann hilfreich, wenn man bei der Eingabe im Textmodus prüfen möchte, ob die Darstellung mit UDGs korrekt ist.

Ein möglicherweise laufendes Programm wird dadurch nicht unterbrochen.

Beispielprogramm in Basic

Das folgende Programm zeichnet Linien auf den Bildschirm.

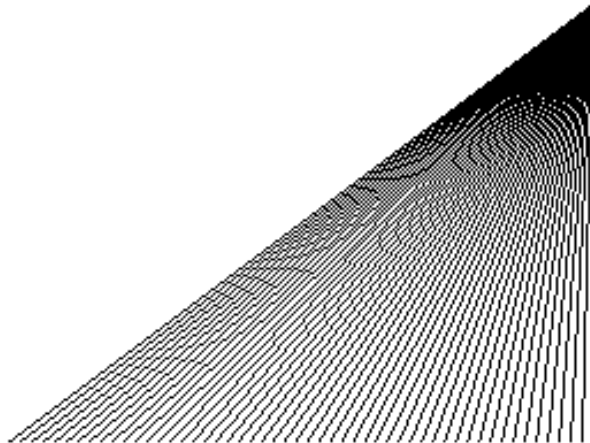
```
10 LET HRG=32736
20 PRINT USR HRG,ON
30 PRINT USR HRG,CLR
100 FOR X=0 TO 255 STEP 4
110 PRINT USR HRG,LINE,X,0,255,
191
120 NEXT X
130 LIST
140 STOP
```

Bei dem STOP in Zeile 140 erzeugt das Basic den Fehler 9/140. Die Grafiksoftware schaltet deshalb auf den Text-Modus zurück.

Mit den Tasten 9 + 0 kann man nun zwischen Grafik und Listing umschalten.

Wenn man Zeile 140 löscht, wird das Programm am Ende reglos im Grafikbildschirm stehenbleiben. Auch hier ist es äußerst praktisch, kurz im Textbildschirm nachzusehen, ob das Programm fertig ist.

Das Programm erzeugt die folgende Grafik:



Beispielprogramm in Assembler

Das folgende Assemblerprogramm macht dasselbe wie das Basicprogramm. Der Funktionscode steht jeweils im A-Register. Weitere Parameter sind in BC und DE Registern abzulegen.

```

        Id      A, HRG_On           ; activates the HRG
        call    7FE3h              ; call the HRG
        ;
        Id      A, HRG_Clr         ; clears the HRG
        call    7FE3h              ; call the HRG
        ;
        Id      A, HRG_Line        ; draw a line between start and end
        Id      bc,0000h           ; Start x=0   y=0
        Id      de,FFBFh          ; End   x=255 y=191
        ;
DrawLoop:
        push    AF                 ;
        push    BC                 ; save all needed registers
        push    DE                 ;
        call    7FE3h              ; call the HRG
        pop     DE                 ;
        pop     BC                 ; recall the registers
        pop     AF                 ;
        ;
        inc     B                  ; modify the startpoint
        inc     B                  ; x = x+4
        inc     B                  ;
        inc     B                  ;
        jr      nz,DrawLoop        ; draw next line until x is 0 again
        ;
        ret                        ;

```

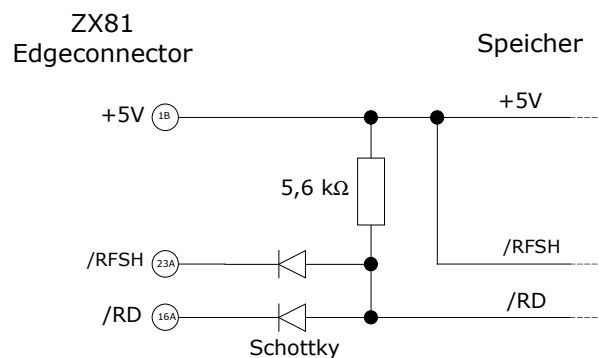
Die verfügbaren Funktionscodes:

HRG_Bank	00h	HRG_Unpoly	10h
HRG_On	01h	HRG_Xpoly	11h
HRG_Off	02h	HRG_Clr	12h
HRG_Pix	03h	HRG_Text	13h
HRG_UnPix	04h	HRG_Invert	14h
HRG_XPix	05h	HRG_HCopy	15h
HRG_Line	06h	HRG_UnText	16h
HRG_Unline	07h	HRG_XText	17h
HRG_Xline	08h	HRG_OverText	18h
HRG_Lineto	09h	HRG_Circle	19h
HRG_Unlineto	0ah	HRG_Uncircle	1ah
HRG_Xlineto	0bh	HRG_Xcircle	1bh
HRG_Box	0ch	HRG_UDGnew	1ch
HRG_Unbox	0dh	HRG_UDGset	1dh
HRG_Xbox	0eh		
HRG_Poly	0fh		

Umbau eines Speichermoduls

Nahezu jedes Speichermodul für den ZX81 oder TS1000 kann grafikfähig gemacht werden. Dazu sind nur zwei Dioden und ein Widerstand nötig.

Zuerst müssen die Signale „Read“ und „Refresh“ (/RD und /RFSH) am Edgeconnector aufgetrennt werden. Danach werden die Dioden, der Widerstand und eine Drahtbrücke wie im folgenden Schaltbild gezeigt eingelötet.



Das Refresh-Signal zum Speicher ist damit immer high und damit inaktiv. Das Read-Signal zum Speicher ist danach immer dann low also aktiv, wenn entweder ein Read oder ein Refresh von der CPU kommt. Damit wird erreicht, dass der Speicher bei einem Refresh der CPU die zur Refresh-Adresse gehörenden Daten der Speicherzelle wie bei einem Read-Zugriff auf den Datenbus legt. Für den Normalbetrieb des ZX81 sind diese Daten unerheblich. Die HRG-Software jedoch benötigt diese Datenzugriffe zur Darstellung des HRG-Bildes.

Applikations-Schnittstelle

Die Schnittstelle zu Basic oder einem aufrufenden Maschinenprogramm liegt im Speicherbereich 7FE0 – 7FFF bzw. 32736 – 32767.

Der folgende Speicherauszug zeigt die dort liegende Struktur:

```
7FE0  C3 40 60      JP HRG-BASIC-API
7FE3  C3 48 60      JP HRG-MC-API
7FE6  07 02        ;VER. LO HI
7FE8  2D 37 2C        ; "HRG"
7FEB  00            ; ENDMARK
```

Diese Struktur wird bei allen Versionen und Varianten gleich sein. Lediglich die Sprungziele und die Versionsnummer können unterschiedlich sein.

Mit dem ersten Sprung kann ein Basic-Programm, so wie in den gezeigten Beispielen, Grafikfunktionen mit Parametern aufrufen.

```
10 LET HRG=32736
20 PRINT USR HRG,ON
```

Für ein Maschinenprogramm ist der zweite Einsprung vorgesehen. Die Parameterübergabe geschieht hier über Z80-Register.

```
ld      A, HRG_On          ; function code in A
call    7FE3h             ; call the HRG
```

Um vor der Benutzung zu prüfen, ob die Grafik-Software und dessen Applikations-Schnittstelle überhaupt vorhanden ist, wird das Folgende empfohlen:

1. Prüfen, ob auf Adresse 7FE0 der Wert C3 bzw. 195 steht.
2. Prüfen, ob die beiden darauf folgenden Speicherzellen einen Wert haben zwischen RAMTOP und 7FE0.

Sind beide Bedingungen erfüllt, kann man davon ausgehen, dass die Software geladen und die Schnittstelle installiert ist.

Tests auf mehreren Systemen haben zudem gezeigt, dass zumindest der Einsprung für Basic nicht zum Absturz führt, wenn die Softwareschnittstelle gar nicht vorhanden ist. Im oben gezeigten Beispiel führt dann die Zeile 20 lediglich zum einem Fehler 2/20 weil die vermeintliche Variable ON nicht existiert.

Ein Assembler-Programm sollte hingegen unbedingt das Vorhandensein der Schnittstelle vor dem Aufruf prüfen!

Funktionen

Die Funktionen der Applikationsschnittstelle für Basic und Maschinensprache wird hier beschrieben.

Jeweils links stehen die Namen der Funktionen für Basic. Rechts die in der Header-Datei benutzten Namen und die Funktionsnummern für die Maschinensprache-Schnittstelle.

Bei der Maschinensprache-Schnittstelle wird die Funktionsnummer im A-Register übergeben. Bei dem meisten Funktionen wird der erste und zweite Parameter im BC-Register und der dritte und vierte Parameter im DE-Register übergeben. Funktionen mit mehr Parametern benutzen entweder den Stack oder einen Pointer auf eine Datenstruktur mit Parametern.

Eine Funktion wirkt immer nur auf die mit BANK selektierte Grafik-Bank. Dadurch ist es möglich, auf verschiedenen Banken unterschiedliche Grafiken gleichzeitig aufzubauen. Die Bildschirmanzeige kann dabei auf eine beliebige Grafik-Bank oder auf die Textseite geschaltet sein.

Setzen, Löschen oder Invertieren eines Pixels

PIX	HRG_Pix	A=03h
UNPIX	HRG_UnPix	A=04h
XPIX	HRG_Xpix	A=05h
X	B	0 bis 255
Y	C	0 bis 191

bearbeitet das Pixel an Position X,Y.

Gerade mit Anfang und Ende

LINE	HRG_Line	A=06h
UNLINE	HRG_UnLine	A=07h
XLINE	HRG_XLine	A=08h
X1	B	0 bis 255
Y1	C	0 bis 191
X2	D	0 bis 255
Y2	E	0 bis 191

Zieht eine Gerade zwischen dem Punkt X1,Y1 und dem Punkt X2,Y2.

Gerade mit Endpunkt

LINETO	HRG_Lineto	A=09h
UNLINETO	HRG_UnLineto	A=0ah
XLINETO	HRG_XLineto	A=0bh
X	B	0 bis 255
Y	C	0 bis 191

Zieht eine Gerade vom letzten bearbeiteten Punkt zu dem Punkt X,Y.

Rechteck mit zwei Eckpunkten

BOX	HRG_Box	A=0ch
UNBOX	HRG_UnBox	A=0dh
XBOX	HRG_XBox	A=0eh
X1	B	0 bis 255
Y1	C	0 bis 191
X2	D	0 bis 255
Y2	E	0 bis 191

Zeichnet eine Rechteck zwischen dem Punkt X1,Y1 und dem Punkt X2,Y2. Die Punkte werden als gegenüberliegende Ecken verstanden.

Kreis mit Mittelpunkt und Radius

CIRCLE	HRG_Circle	A=19h
UNCIRCLE	HRG_Uncircle	A=1ah
XCIRCLE	HRG_Xcircle	A=1bh
X	B	0 bis 255
Y	C	0 bis 191
R	D	0 bis 255

Zeichnet einen vollständigen Kreis um den Punkt X,Y mit dem Radius R. Der Kreis darf größer als der Bildschirm sein.

Polygon mit beliebig vielen Eckpunkten

POLY	HRG_Poly	A=0fh
UNPOLY	HRG_UnPoly	A=10h
XPOLY	HRG_XPoly	A=11h
X1	auf Stack	0 bis 255
Y1	auf Stack	0 bis 191
X2	auf Stack	0 bis 255
Y2	auf Stack	0 bis 191
usw.	usw.	

Zeichnet ein geschlossenes Polygon mit den angegebenen Punkten. Der Anfangspunkt muss am Ende nicht wiederholt werden. Die Funktion zieht am Ende selbsttätig eine Gerade zum Anfangspunkt zurück. Die Parameterübergabe im Assembler geschieht über den Stack. Dazu muss das aufrufende Programm vor dem Aufruf eine Liste der Punkte auf den Stack legen. Als erstes wird eine Endemarke (ffh, ffh) auf den Stack abgelegt. Danach folgen die Punktkoordinaten. Erst dann wird die Funktion aufgerufen.

Beispiel:

```

ld      A, HRG_Poly      ; draw polygone
ld      bc,FFFFh        ; endmark onto stack
push   BC                ;
ld      bc,0000h        ; Start x=0 y=0 onto stack
push   BC                ;
ld      bc,0A64h        ; Pixel x=10 y=100 onto stack
push   BC                ;
ld      bc,1432h        ; Pixel x=20 y=50 onto stack
push   BC                ;
call   7FE3h            ; call the HRG

```

Die Funktion entfernt die Daten von Stack. Das aufrufende Programm muss den Stack hinterher nicht korrigieren!

Einschalten der Grafikanzeige

ON	HRG_On	A=01h
-----------	---------------	-------

Schaltet die Bildschirmanzeige auf die momentan mit BANK ausgewählte Grafikbank. Die Anzeige muss nicht eingeschaltet sein, um in einer Grafikbank zu zeichnen! Es kann auch eine andere Grafikbank angezeigt werden als die, in die gerade gezeichnet wird.

Ausschalten der Grafikanzeige

OFF	HRG_Off	A=02h
------------	----------------	-------

Schaltet die Bildschirmanzeige zurück auf den Textbildschirm. Dabei kann weiter in einer Grafikbank gezeichnet werden.

Löschen der Grafikanzeige

CLR	HRG_Clr	A=12h
------------	----------------	-------

Überschreibt alle Pixel der angewählten Grafikbank mit Weiß.

Text in die Grafikanzeige kopieren

TEXT	HRG_Text	A=13h
UNTEXT	HRG_UnText	A=16h
XTEXT	HRG_XText	A=17h
OVERTEXT	HRG_OverText	A=18h

Kopiert den Inhalt des Textbildschirms in die momentan angewählte Grafikbank. Die Zeichen werden mit TEXT in die Grafik gesetzt, mit UNTEXT gelöscht oder mit XTEXT invertiert. Mit OVERTEXT wird auch der Zeichenhintergrund mit Weiß überschrieben.

Benutzerdefinierten Zeichensatz zurücksetzen

UDGNEW	HRG_UDGnew	A=1Ch
---------------	-------------------	-------

Der für die Text-Funktionen benutzte Zeichensatz wird hiermit zurückgesetzt. Es stehen insgesamt 127 Zeichen zur Verfügung, die mit dieser Funktion auf den Sinclair Standardzeichensatz initialisiert werden.

Benutzerdefiniertes Zeichen setzen

UDGSET	HRG_UDGset	A=1Dh
C	BC zeigt auf die	0 bis 128
B0	Struktur	0 bis 256
B1		0 bis 256
B2		0 bis 256
B3		0 bis 256
B4		0 bis 256
B5		0 bis 256
B6		0 bis 256
B7		0 bis 256

Es werden für das Zeichen C die Bytes B0 bis B7 als Bitmuster definiert. Im Sinclair-Basic sind die Zeichencodes 0 bis 63 für normale Zeichen und die Codes 128 bis 191 für dieselben Zeichen in inverser Darstellung definiert. Hier können sowohl die normalen als auch die inversen Zeichen neu definiert werden. Es stehen somit, abzüglich des Leerzeichens, 127 benutzer-definierbare

Zeichen zur Verfügung. Der Wert für C darf hierbei auch zwischen 64 und 127 oder zwischen 192 und 255 liegen. Diese Bereiche werden dann automatisch auf den Wertebereich 128 bis 191 abgebildet.

Im folgenden Beispiel wird das Pfund-Zeichen mit einem Euro-Zeichen ausgetauscht.

Beispiel in Assembler:

```

        ld      A, HRG_UDGset      ; set a character
        ld      bc, MyEuro         ; pointer to a structure of 9 bytes
        call   7FE3h              ; call the HRG
        jp      ...                ;
MyEuro:
        db     12                  ; character code for „£“
        db     0, 60, 66, 248     ; graphic bits for a „€“ character
        db     64, 248, 66, 60    ; 0      = 00000000
                                   ; 60     = 00111100
                                   ; 66     = 01000010
                                   ; 248    = 11111000
                                   ; 64     = 01000000
                                   ; 248    = 11111000
                                   ; 66     = 01000010
                                   ; 60     = 00111100

```

Beispiel in Basic:

```

10 LET HRG=32736
20 PRINT USR HRG,UDGSET,12,0,6
0,66,248,64,248,66,60
30 PRINT "123£"
40 PRINT USR HRG,TEXT
50 PRINT USR HRG,ON

```

Invertieren der Grafikanzeige

INVERT	HRG_Invert	A=14h
---------------	-------------------	-------

Keht alle Pixel der angewählten Grafikbank um.

Drucken der Grafikanzeige

HCOPY	HRG_Hcopy	A=15h
--------------	------------------	-------

Druckt die angewählten Grafikbank auf einem Sinclair kompatiblen Drucker aus.

Auswählen der Grafikbank

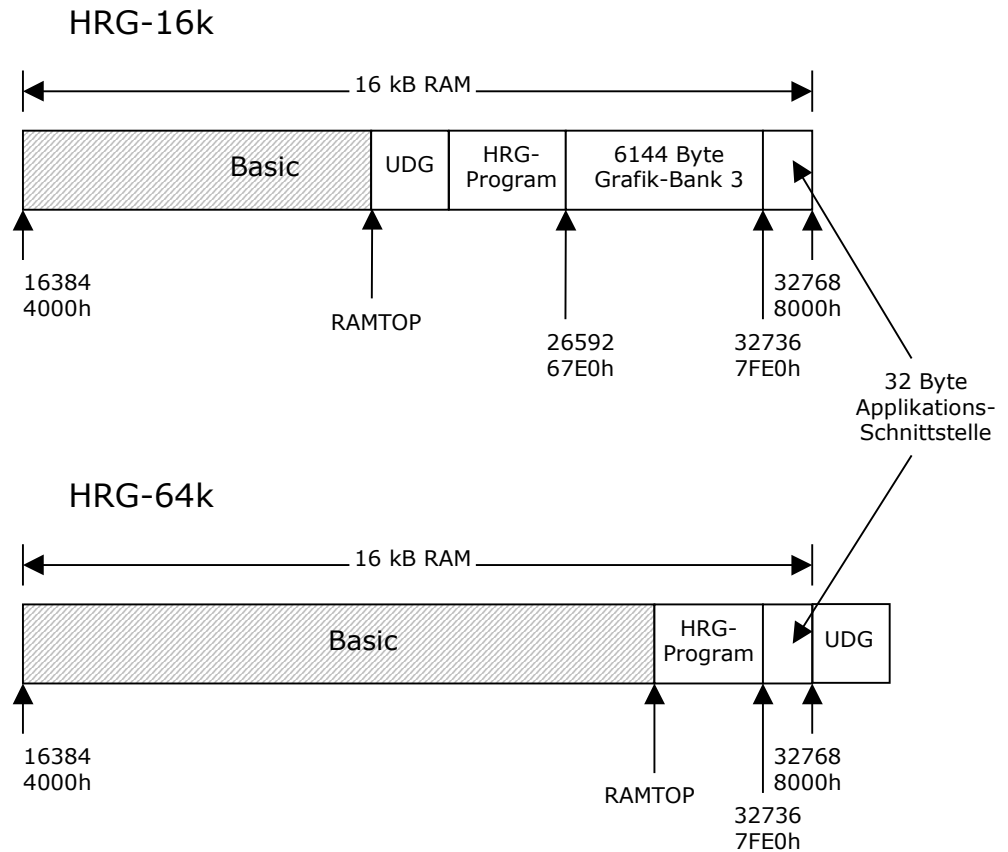
BANK	HRG_Bank	A=00h
N	BC	3 -7 oder ffffh

Selektiert eine Grafikbank.

Ein Maschinenprogramm hat zusätzlich die Möglichkeit, die aktuelle Bank und deren Startadresse zu erfragen, wenn BC=ffffh ist. Die Funktion antwortet dann mit der aktuellen Bank im A-Register und der Startadresse im HL-Register.

Speicherplan

Die genaue Lage des Programms und der einzelnen Grafik-Bänke ist in der folgenden Grafik erklärt.



Die Grafik-Bank 3 existiert in der 64k Variante nicht. Dadurch ist mehr RAM für das Basic frei.

Die Grafik-Bänke 4 bis 7 liegen um jeweils 8192 bzw. 2000h Adressen höher als die Grafik-Bank 3.

Grafik-Bank 3	67E0h bis 7FE0h -1
Grafik-Bank 4	87E0h bis 9FE0h -1
Grafik-Bank 5	A7E0h bis BFE0h -1
Grafik-Bank 6	C7E0h bis DFE0h -1
Grafik-Bank 7	E7E0h bis FFE0h -1