KRT-Grafik

Eine Vollgrafik für den Z9001 ala Kleinstrechnertipps 11

Der Nachbau des robotron-Grafik-Zusatzes ist recht aufwändig. Günstiger erschien mir eine Idee aus den Kleinstrechnertipps Nr. 11, die dort für den Z1013 vorgestellt wurde und mit nur 4 Schaltkreisen auskommt: Der Bildwiederholspeicher der internen Grafik wird 8fach parallel ausgeführt, diese 8 Ausgänge werden anstelle der Ausgabe des Zeichengenerators direkt auf den Bildschirm ausgegeben.

Eigenschaften

- 320 x 192 Pixel
- farbfähig bei Vorhandensein des Farbzusatzes (KC87.x1). Je 8×8 Pixel haben dieselben Vorder- und Hintergrundfarben
- einfachste Hardware

Nutzung der Vollgrafik in BASIC

Für BASIC habe ich einen Treiber **GRAFP** geschrieben, der die im BASIC86 vorhandenen Befehle auf diese Hardware umsetzt, so dass vorhandene BASIC-Programme für die robotron-Grafik oder den Plotter auch mit dieser Grafik benutzt werden können. Er basiert auf dem robotron-Grafik-Treiber und unterstützt zusätzlich auch den Plotter als Grafikausgabegerät. Der Treiber ist bereits im aktuellen megamodul-ROM enthalten!

Zu beachten ist, dass diese Grafik eine höhere horizontale Auflösung besitzt, evtl. muss bei den Programmen ein SCALE-Faktor ergänzt werden, damit die volle Grafikauflösung zur Geltung kommt. Ebenfalls aufpassen muss man bei Textausgaben auf den Textbildschirm. Hier muss vor der Ausgabe mit PRINT o.ä. unbedingt mit SCREEN 0 der normale Textbildschirm zugeschaltet werden, ansonsten erscheinen statt der Buchstaben je 8 Pixel im Bild. Für Textausgabe im Grafikmodus gibt es aber noch den LABEL-Befehl. Gezeichnet wird mit den aktuellen Farben. Es kann also jederzeit mit INK, PAPER, BORDER eine andere Farbe gewählt werden.

Nebenstehendes Bild ist mit einer aktuellen Version des JKCEMU entstanden. Das zugehörige BASIC-Programm zeigt die Nutzung der BASIC-Befehle für Grafik und Farbe.

```
1 !TEST V.POHLERS 19.11.2009
10 SCREEN1:GCLS:ZERO(0,0):SCALE 1,1
20 ! Willkommen
30 LINE(0,0)-(320,192),1,B
40 PSET(160,20),0
50 LABEL "KC 87 mit Pohlers-Grafik"
60 PAUSE 50
70 SCREEN 0
100 ! 4 Kreise in Farbe
110 SCREEN 1
120 FOR FA=3 TO 6
```

```
130 CIRCLE (125,95),12*FA
140 INK FA+1
150 NEXT
160 INK 3
170 PAUSE 50
1000 SCREEN 0
1010 CLS
```

Der Treiber belegt den Speicherbereich 8D60-A800, der Treiber CRT80P (s.u.) belegt den Speicherbereich B000-B7FF, beide Treiber können damit gleichzeitig im Speicher liegen und parallel genutzt werden. Zusätzlich ist der Bereich B800-BFFF für einen Druckertreiber frei.

80-Zeichentreiber fürs OS und CP/M

Wenn man schon eine Pixelgrafik hat, kann man Text mit beliebigen Zeichensätzen ausgeben lassen. Ein 4×8-Pixel-Font und mein Treiber **CRT80P** machen CP/M zu einem 80×25-Zeichen-Display. Im normalen OS des Z9001 kann dank des modularen Systems ebenfalls mit 80 Zeichen gearbeitet werden. Dazu ist einfach der Treiber CRT80P zu starten. Alle Programme, die nicht direkt in den Bildwiederholspeicher geben, funktionieren mit diesem Treiber, also EDIT/ASM, BASIC etc. Man kann BASIC starten und hat beim Listen und in Programmen etc. die volle Bildschirmbreite von 80 Zeichen zur Verfügung. Auch die Kombination Grafiktreiber und der 80-Zeichenmodus mit BASIC ist möglich. CLS, Farbe, Windows, Tasten etc. funktionieren wie im originalen CRT-Treiber!

Da das originale Basic mit PRINT AT leider direkt in den BWS schreibt, ist es empfehlenswert, bei Nutzung des 80-Zeichen-Treibers das alternative **BASICP** zu nutzen (s.u.). Hier wird PRINT AT über die SYstemfunktionen realisiert. Außerdem ist die Fensterbreite bei WINDOW auf 80 Zeichen gesetzt.

Es können über eine neue Treiberfunktion 9 auch andere Zeichensätze genutzt werden, s. die Demo **SWITCH_ZG.**

Weiterhin gibt es von mir den analog arbeitenden Treiber **CRT40P**, der für 40 Zeichen gedacht ist und damit die gewohnte Zeichenbreite von 8 Pixeln bietet.

Anzeige der Treiberzuordnung nach Start von CRT80	Start von BASIC mit 80 Spalten
Grafik und Text sind gleichzeitig darstellbar (es muss CRT80P und GRAFP geladen sein!)	Cool: Bei CLS scrollt dann auch die Grafik nach oben weg!
Farbe funktioniert wie gewohnt	Es gibt auch einen 40-Zeichen-Treiber **CRT40P**. Es können Zeichensätze nachgeladen werden, wie die Demo **SWITCH_ZG** zeigt.
Ein BASIC-Programm in ungewohntem Zeichensatz…	… und auch in 80 Zeichen mit CRT80P

Hardware

Die Vollgrafik benötigt nur 4 einfache Schaltkreise (D175, DL275, DS8282 o.ä., U6264) !

Zum vorhandenen Bildwiederholspeicher (f. Textmodus) werden 8 weitere 1K-Bereiche parallel geschaltet. Diese 8 Bildwiederholspeicher BWSG 1..8 a 1K beginnen also alle auf Adresse EC00h und werden einzeln über Port 0B8h aktiviert:

OUT PortB8, 0	normaler Textmodus (wie nach Reset)
OUT PortB8, 0000 1000b	Vollgrafik an, Beschreiben BWS-G 1
OUT PortB8, 0000 1001b	Vollgrafik an, Beschreiben BWS-G 2
OUT PortB8, 0000 1010b	Vollgrafik an, Beschreiben BWS-G 3
…	
OUT PortB8, 0000 1111b	Vollgrafik an, Beschreiben BWS-G 8

Durch Bit4 von PortB8 wird die Vollgrafik ein- bzw. ausgeschaltet. Durch Bit0..3 von PortB8 werden die einzelnen Bildwiederholspeicher-Ebenen aktiviert. Das ist nur sinnvoll bei Bit4=1, da ansonsten die Vollgrafik ausgeschaltet ist und dann immer nur der normale Textmodus-Bildwiederholspeicher an der Adresse EC00h beschrieben wird.

Die Videoausgabe im Textmodus erfolgt dadurch, dass ein Zeichen aus dem BWS gelesen wird, der Zeichensatz-ROM auf dieses Zeichen positioniert wird und dann 8 Bit aus dem Zeichensatz-ROM für die aktuelle Bild-Zeile ausgegeben werden. Die auszugebenden Pixel der nächsten 7 Bild-Zeilen folgen im Zeichensatz-ROM auf den nachfolgenden 7 Byte.

Die Videoausgabe im Grafikmodus erfolgt ähnlich, nur entfällt der Zeichensatz-ROM. Die Pixel für die erste auszugebende Bild-Zeile stammen direkt aus dem Zeichen aus BWS-G 1, die für die zweite Zeile aus BWS-G 2 usw.

Das Grafikbild wird also zeilenweise aufgebaut:
1. Zeile = die ersten 40 Byte = 320 Pixel aus BWS-G 1. Die Pixel werden von links nach rechts von Bit7..Bit0 ausgegeben
2. Zeile = die ersten 40 Byte aus BWS-G 2
..
8. Zeile = die ersten 40 Byte aus BWS-G 8
9. Zeile = die nächsten 40 Byte aus BWS-G 1
10.Zeile = die nächsten 40 Byte aus BWS-G 2 usw.

Achtung: Da im Speicherbereich des BWS von EC00-EFFF auch die Adresstabelle der Gerätetreiber steht, wird der Z9001 bei einfachem Einschalten von PortB8, Bit4 abstürzen. Hier muss vorher der Bereich von EFC0-EFFF in alle BWSG1..8 kopiert werden, ehe eine Zeichenein- oder -ausgabe auf Kommandoebene erfolgen darf! Obiger Treiber für BASIC macht dies beim Initialisieren mit SCREEN 1.

Download

- Schaltplan, Layout → U. Zander entwickelt eine Aufsteckplatine
- grafik-krt.zip Software (BASIC-Treiber GRAFP, BASIC-Beispiele, CONST-Treiber CRT80P, CRT40P, Demos, Quellcode)

Hinweise zu den Programmen

```
Programme
---------

crt80p.tap Systemtreiber für 80 Zeichen/Zeile. Es kommt ein 4x8-Pixel-
Zeichensatz zur Verwendung. Befehle:
CRT80 80-Zeichentreiber fürs OS (CONST-UC-Treiber)
CLS Bildschirm löschen
X Textmode wiederherstellen

crt40p.tap Systemtreiber für 40 Zeichen/Zeile. Es kommt ein CGA/Z9001-8x8-
Pixel-
Zeichensatz zur Verwendung. Befehle:
CRT40 40-Zeichentreiber fürs OS (CONST-UC-Treiber)
CLS Bildschirm löschen
X Textmode wiederherstellen

switch_zg.tap 2 weitere Zeichensätze für CRT40. Befehle:
Z1 Zeichensatz 1
Z2 Zeichensatz 2
Z3 treibereigener Zeichensatz

basic_p.bin BASIC, Binary C000-E7FF
Damit können BASIC-Programme mit 80 Zeichen Breite genutzt werden!
Hintergrund: das Robotron-BASIC schreibt mit PRINT-AT direkt in den BWS,
damit funktioniert dieser Befehl nicht mit obigen CONST-Treibern.
Diese BASIC-Variante wurde im AT-Befehl auf die Nutzung der OS-Routinen
umgeschrieben. Außerdem wurde WINDOW auf 80 Zeichen gepatcht

grafp.tap Vollgrafik-Treiber für BASIC. Dieser Treiber muss vor BASIC
geladen werden. Basic ist dann mit Memory End < 32000 zu starten.
Es werden alle Befehle wie beim Grafikzusatz von robotron umgesetzt,
nur die Auflösung ist auf 320x192 Pixel vergrößert.


BASIC-Beispiele
---------------

r_grdem2_sss.tap Große Grafikdemo. Angepasste Version von R+PLDEMO (Demo f.
Plotter)
globus87_sss.tap Der Globus (H. Völz). Achtung: Lange Rechenzeit!
grafpt1_sss.tap Testprogramm 1
grafpt2_sss.tap Testprogramm 2
grafpt3_sss.tap Testprogramm 3
n-eck_sss.tap Zeichnen von n-Ecken
uhr_sss.tap Anzeige der Uhrzeit als Analoguhr. Tipp: Vorher mit TIME im OS
die Uhrzeit einstellen!
```

```
Quellcodes
----------

bm608p.asm BASIC-Erweiterung mit neuem AT-Befehl
crt40p.asm CRT40-Treiber
crt80p.asm CRT80-Treiber
grafp.asm Vollgrafik-Treiber f. BASIC
switch_zg.asm Zeichensätze wechseln f. CRT40
SPACE8.pf 8x8-Zeichensatz
TINYTYPE.pf 8x8-Zeichensatz
z9_zg.rom 8x8-Zeichensatz (original Z9001)
zg_cga.rom 8x8-Zeichensatz (CGA-Z9001)
zs80.zs 4x8-Zeichensatz



#-------------------------------------------------------------------------
---
# Befehle des Vollgrafik-Treibers f. BASIC
# (entsprechen dem MSX1-Standard)
#-------------------------------------------------------------------------
---



PSET(x,y)[,stift]
Setzen eines Punktes
stift - 0 Hintergrundfarbe (Löschen), 1 Vordergrundfarbe, (Standard: stift =
1)

LINE [(x1,y1)]-(x2,y2)[,[stift][,B[F]]]
Zeichnen von Linie oder Rechteck
stift - 0 Hintergrundfarbe (Löschen), 1 Vordergrundfarbe, (Standard: stift =
1)
B - Zeichnen eines Rechteckes (box)

CIRCLE(x,y),radius[,stift[,anf-winkel[,end-winkel[,ellip]]]]
CIRCLE(x,y),radius[,stift],[anf-winkel],[end-winkel],ellip
Zeichnen eines Kreises
stift - 0 Hintergrundfarbe (Löschen), 1 Vordergrundfarbe, (Standard: stift =
1)
anf-winkel - Anfangswinkel für Kreisbogen (Bogenmaß)
end-winkel - Endwinkel für Kreisbogen (Bogenmaß)
ellip - Ellipsenparameter (Standard: ellip = 1)

PAINT(x,y)[,c[,d]]
c default 1
d default 1

LABEL string[,stift]
Ausgabe einer Zeichenkette ab aktueller Stiftposition

SIZE b,l[,r[,s[,a]]]
```

```
SIZE b,l,[r],[s],a
Festlegung der Schriftart (nur für PLOTTER)
b Breite der Buchstaben (Geräteeinheiten)
I Länge (Geräteeinheiten)
r Schreibrichtung (Bogenmaß)
s Schräglage (Bogenmaß)
a Abstand 0 - gleichabständig (Standard), 1 - proportional

ZERO (x,y)
Der Nullpunkt des aktuellen Koordinatensystems wird in den Punkt (x,y)
gelegt.

HOME
Die Koordinaten des letzten erreichbaren Punktes werden in den
Koordinatenursprung, den Punkt (0,0) des aktuellen Koordinatensystems,
gelegt.
HOME ist gleichbedeutend mit PSET(0,0),0.

GCLS
Grafikbildschirm löschen

XPOS
XPOS liefert als Rückgabeparameter die x-Koordinate des letzten erreichten
Punktes

YPOS
YPOS liefert als Rückgabeparameter die y-Koordinate des letzten erreichten
Punktes

SCALE xfaktor,yfaktor
Mit SCALE erfolgt eine Maßstabsfestlegung in x- bzw. y-Richtung (unabhängig
voneinander).
Alle Koordinatenangaben werden vor der Verarbeitung mit xfaktor bzw. yfaktor
multipliziert.

SCREEN [0],plotter
plotter = 0 => Abschalten des Plotters
SCREEN 1 => Grafikbildschirm
SCREEN 0 => Textbildschirm

POINT(X,Y)
Bestimmen des Punktzustandes
POINT liefert als Rückgabeparameter 0, wenn der Abfragepunkt die
Hintergrundfarbe, und 1, wenn der Abfragepunkt die Vordergrundfarbe hat.



--------------

Tipp: Vollst. Init. der Grafik mit
SCREEN1:GCLS:ZERO(0,0):SCALE1,1:HOME
```

```
Tipp: Bei KCs mit Farbe funktionieren die Farbbefehle (INK, PAPER, BORDER)
auch im
Grafikmodus. Für ein 8x8 Pixel großes Feld können eigene Vorder- und
Hintergrundfarben gewählt werden (s. Demos)

Tipp: Im Grafikmodus dürfen keine Ausgaben mit PRINT oder PRINT AT erfolgen.
Es muss vorher auf SCREEN 0 geschaltet werden! Zur Textausgabe im
Grafikmodus
gibt es den Befehl LABEL.
```

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/z9001/erweiterungen/krtgrafik?rev=1279883731

Last update: **2010/07/22 22:00**