

Grafikzusatz

Der Z9001 ist im Standard-Zustand nicht vollgrafikfähig, da er lediglich Textmodus mit 40×24 bzw. 40×20 Zeichen bietet. Gleichwohl bietet das BASIC des Plotter-Moduls bzw. das im KC87.2x eingebaute BASIC86 jedoch Grafikbefehle. Allerdings sind dies nur Platzhalter in der Liste der verfügbaren BASIC-Befehle; ein vollständiger Code für diese Befehle muss außerhalb des BASIC als Treiber [grafik.zip](#) hinzugeladen werden.



Solche Treiber gibt es von Robotron für **Plotter**, den bei U. Zander beschrieben **Grafik-Zusatz** sowie von mir für den **unverbauten KC87** mit einer Auflösung von 40×24 „Pixeln“ und 80×48 Pixeln (Viertel-Grafik, im Bild eine Analog-Uhr mit diesem Treiber). Neu hinzugekommen ist auch ein Treiber für die **KRT-Grafik**.

Der Grafik-Zusatz von robotron

Hinweis: der Nachbau dieser Vollgrafik ist nicht mehr empfehlenswert. Einfacher und vielseitiger ist die neu entwickelte **KRT-Grafik**.



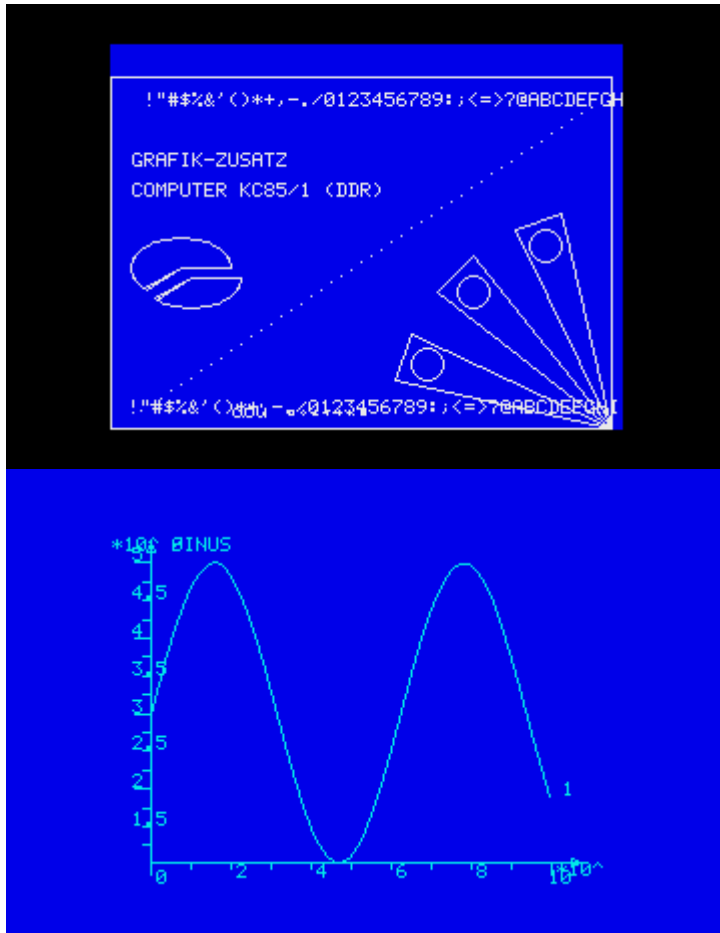
Der Grafik-Zusatz wurde leider nie von Robotron produziert, sondern stand nur zur Nachnutzung zur Verfügung (s. [Nutzerkatalog 2](#), Blatt 4). Technisch gesehen ist der Grafik-Zusatz eine zusätzliche Grafikkarte mit einer Auflösung von 256×192 Pixel und einer Farbtiefe von 1 Bit; es können eine Vorder- und eine Hintergrundfarbe aus 8 Farben gewählt werden; diese Farben gelten dann für das gesamte Bild. Die

Umschaltung zwischen interner Bildausgabe und Grafik-Zusatz erfolgt im Original über Relais. Es ist entweder die interne Bildausgabe aktiv oder der Grafik-Zusatz, eine Overlay-Möglichkeit besteht nicht. Zur Textausgabe besitzt der Treiber für den Grafik-Zusatz einen 6×8-Pixel-Zeichensatz; damit sind 42 Zeichen/Zeile möglich.



KC mit originalem Grafikzusatz. Besitzer R. Kurth

Im aktuellen JKCEMU von J. Müller (ab Version 0.6) <http://www.jens-mueller.org/jkcemu/index.html> wird der Robotron-Grafik-Zusatz emuliert:



Beispiel-Grafiken mit R+GRDEM2, meiner Adaption des Plotter-Demo-Programms. (Ein originales Testprogramm GRDEMO.SSS ist erst 2011 aufgetaucht)

Der Grafikusatz enthält einen externen Pixelspeicher von 6144 Byte (0..17FFh) (= 256×192 Pixel / 8 Pixel/Byte).

| Spalte | 0..7 | 8 | 16 | ... | 248..255 |
|--------|-------|------|------|-----|----------|
| Zeile | ----- | | | | |
| 0 | 0000 | 0001 | 0002 | ... | 001F |
| 1 | 0020 | 0021 | 0022 | ... | 003F |
| | .. | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| 191 | 17E0 | 17E1 | 17E2 | ... | 17FF |
| | ----- | | | | |

Dieser Speicher ist nicht direkt ansprechbar; er kann nur byteweise über Ports angesprochen werden.

```
port 0B8h ; Farbe + Grafik ein/aus
        ; 7 6 5 4 3 2 1 0
        ; | +-+--+ | +-+--+
        ; |   |   |   |
        ; |   |   |   PAPER (BGR)
        ; |   |   Grafik ein/aus
```

```

; | INK (BGR)
; RAND
port    0B9h    ; Adresse für internen Speicher
; die unteren Adressleitungen werden mit OUT (C)
; direkt ausgegeben, die oberen Adressdaten liegen
; über Register B auf dem Adressbus
port    0BAh    ; Daten

```

Der Zugriff nutzt dabei die Eigenheit des Z80 aus, bei IN/OUT-Befehlen das gesamte Doppelregister AF bzw. BC auf den Adressbus zu legen. Hardwaremäßig wird erst mit der Ausgabe des Speicherinhalts die Adresse an die internen RAM's aktiv. Die Ausgabe der Adresse erfolgt in zwei Schritten: Erst Ausgabe der Adresse Low-Teil auf Port B9 (Ein Latch speichert den Wert), dann Ausgabe des Speicherinhalts auf den Datenbus und gleichzeitig die Ausgabe der Adresse High-Teil auf den Adressbus. Folgender Code zeigt die notwendigen Abläufe beim Schreibzugriff:

```

write:  ld      bc, (curadr)    ; current address in external RAM
        ld      a, c           ; untere 8 Bit der Adresse
        ld      c, port_0BAh   ; ausgeben auf Port 0BAh
        out     (port_0B9h), a ; out low address, auf High-Teil des
Adressbusses
        ld      a, (curbyte)   ; liegt A, wird ignoriert
        out     (c), a         ; auszugebendes Byte
; in C steht der Port 0BAh
        out     (c), a         ; out data and high address
; jetzt steht B auf dem High-Teil des
Adressbusses,
; auf den Low-Adressen liegt der Port (wird
ignoriert)
; und auf dem Datenbus liegt A

```

Der Plotter



Der Treiber unterstützt auch die alternative Grafikausgabe auf einen angeschlossenen Kleinplotter, s. [Plotter](#).

Nutzung mit BASIC

Man benötigt: 2 RAM-Module, das Plotter-BASIC bzw. BASIC 86 (also einen KC87.2x oder das Plottergrafikmodul) und den Gerätetreiber. Zuerst muss der Gerätetreiber GRAF.COM geladen werden. Beim Start von BASIC ist memory end? **32700** (oder kleiner) anzugeben.

Folgende Grafikbefehle gibt es im BASIC:

PSET, LINE, CIRCLE, PAINT, LABEL, SIZE, ZERO, HOME, GCLS, SCALE, SCREEN, POINT, XPOS, YPOS.

Im Gegensatz zur Plotter-Anleitung können auch XPOS, YPOS und (wenig sinnvoll) PAINT und POINT genutzt werden:

- **XPOS** liefert als Rückgabeparameter die x-Koordinate des letzten erreichten Punktes,
- **YPOS** die entsprechende y-Koordinate.
- **POINT(x,y)** liefert als Rückgabeparameter 0, wenn der Abfragepunkt x,y die Hintergrundfarbe, und 1, wenn der Abfragepunkt die Vordergrundfarbe hat. Im Plottertreiber liefert POINT immer 0.
- **PAINT (x,y)[,c[,d]]** ist undokumentiert, und weder im Plotter- noch im Vollgrafiktreiber realisiert. Im Plottertreiber ist PAINT im physischen Treiber mit einem Return kurzgeschlossen.
- **LINE** akzeptiert neben dem Parameter B auch den Parameter BF (box filled). Das ist jedoch nicht implementiert.

Für die gemeinsame Programmierung von Vollgrafik und Plotter ist der **unterschiedlich große Zeichenbereich** zu beachten: Plotter: 2550×1800, Vollgrafik: 256×192. Zur Initialisierung beider Geräte dient der Befehl **SCREEN**: SCREEN 0,0 - Standardbildschirm, SCREEN 0,1 - Plotter (am E/A-Port) aktivieren, SCREEN 1 - Vollgrafik. Ein gleichzeitiger paralleler Betrieb von Vollgrafik und Plotter funktioniert nicht; es kann aber wechselweise auf beide Geräte (mit entsprechender Skalierung) derselbe Zeichenvorgang wiederholt werden.

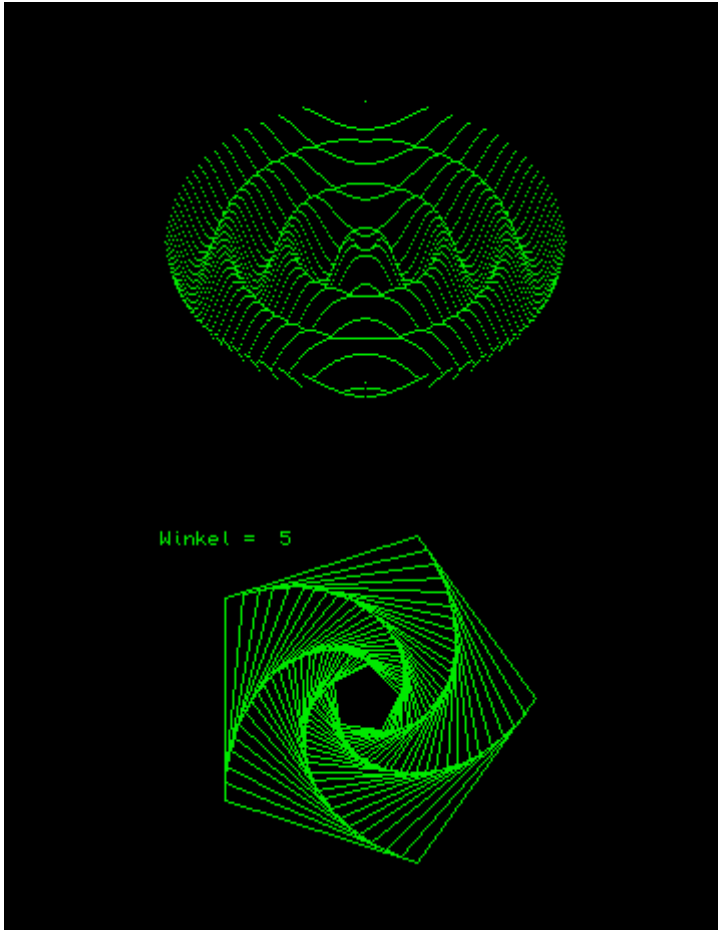
Die BASIC-Befehle sind bezogen auf die Vollgrafik in z9_gbas.pdf ([Grafik-Zusatz](#) bei U. Zander beschrieben. Für die Plotter stehen die Befehle in den zugehörigen Handbüchern [xy4131.pdf](#) bzw. [xy4140.pdf](#) (s.a. [Plotter](#)).

Mai 2009: Vermutlich stand das MSX-BASIC Vorlage für die Grafikbefehle. Auch die Auflösung von 256×192 Pixeln entspricht dem MSX-1-Standard. Die Befehle sind offenbar genauso wie im MSX-Standard definiert. (Quelle: MSX2 TECHNICAL HANDBOOK). In der Ergänzung zur Bedienungsanleitung des KC87.2x [anleitung_kc87_2x.pdf](#) steht dazu: „Die Syntax und Semantik der Grafik-Anweisungen entsprechen weitgehend dem internationalen Standard und sind im Detail den Bedienungsanleitungen der grafischen Ausgabegeräte zu entnehmen.“

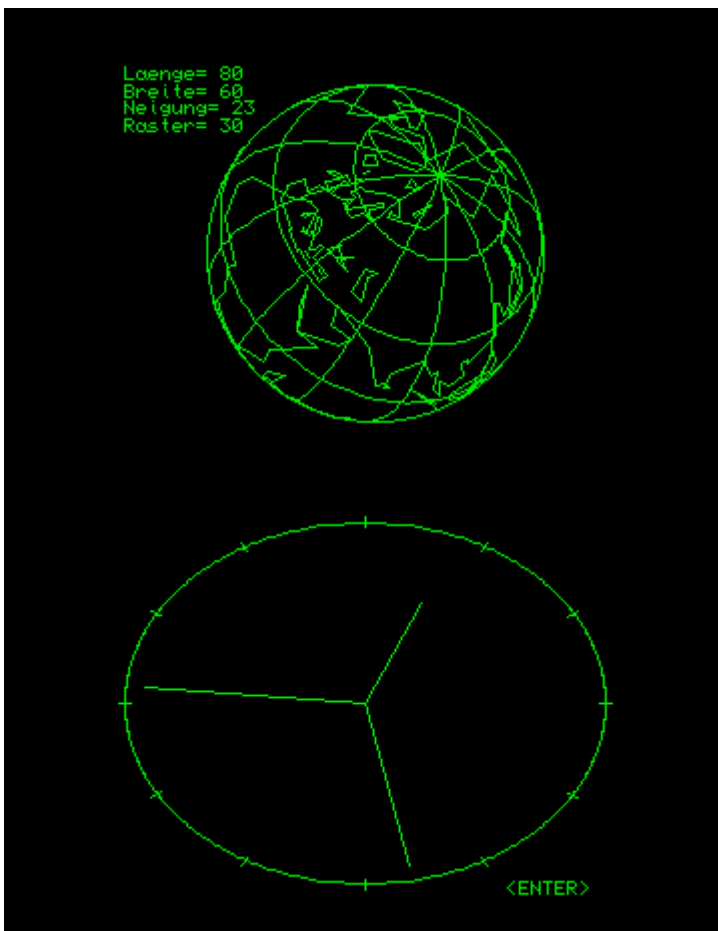
- **PSET {(X,Y) | STEP(X,Y)}[, <colour>[, <logical operation>]]** Draws the dot in the coordinate specified by (X,Y) on the graphic screen.
- **CIRCLE {(X,Y) | STEP(X,Y)},<radius>[, <colour>[, <start angle>[, <end angle>[, <proportion>]]]]** Draws a circle whose center is at (X,Y) and whose size depends on <radius>.
- **LINE [{(X1,Y1) | STEP(X1,Y1)}] - {(X2,Y2) | STEP(X2,Y2)}[, <colour>[, {B|BF}], <logical operation>]]** Draws a line or a rectangle on the screen.
- **PAINT {(X,Y) | STEP(X,Y)}[, <colour>[, <border colour>]]** Paints the area surrounded by specified <border colour> using <colour>.
- **SCREEN <screen mode>[...]** Sets the screen mode and so on.
- **function POINT (X,Y)** Returns the colour of the dot specified by coordinate (X,Y).
- **LABEL, SIZE, ZERO, HOME, GCLS, SCALE und die Funktionen XPOS und YPOS** gibt es in MSX nicht

Randbemerkung: Interessanterweise werden in der Broschüre „BASIC-Anwenderprogramme und Hinweise zum Kleincomputer robotron KC 85/1 und KC 87“ andere BASIC-Befehle beschrieben, die wohl ursprünglich geplant waren und offenbar auch schon implementiert wurden: MOVE x,y (zum Punkt bewegen); PLOT x,y (Linie zum Punkt zeichnen); POINT x,y (Punkt zeichnen); ZERO x,y; HOME;

LABEL A\$; SIZE xx,xy,yx,yy.



3D-Grafik (nach W.Hinze, Mikrocomputer, 1988); n-Eck (nach H.Völz)



Globus (nach H.Völz); Uhr (V.Pohlers)

Treiber-Intern

Von Robotron gibt es zwei **Treiber** als Erweiterung zum BASIC86: Den Treiber GRPLOT.COM von der Kassette R0114 (Plotter-Kassette) und den Treiber GRAF.COM zum Grafik-Zusatz. Der Treiber GRAF.COM umfasst phys. Treiber sowohl für die Plotter XY4131 als auch für die Vollgrafik. Für die Textausgabe enthält der Vollgrafik-Treiber einen 6x8-Pixel-Zeichensatz (der gleiche wie bei den Druckertreibern). Der Treiber GRAF.COM ist somit universell nutzbar; belegt aber auch mehr Arbeitsspeicher.

Vor Arbeit mit BASIC muss man den Grafik-Treiber GRAF.COM laden. Anschließend das Plotter-BASIC zu starten (BASIC-86). Bei Memory end ist 32700 (oder kleiner) anzugeben.

Die Basicbefehle zur Grafik-Erweiterung springen auf eine Befehlsliste außerhalb des BASIC-Bereiches: die Liste wird auf A7D6...A7FD erwartet. Auf A7D6 (dez: -22570) muss C3 (JP) stehen, sonst ist keine Erweiterung geladen und die Grafikbefehle können nicht genutzt werden.

prinzipieller Aufbau des Treibers GRAF.COM:

| |
|--|
| Sprungverteiler physische Treiber |
| Treiber für Plotter, im Anschluss Zeichentabelle |
| Treiber für Vollgrafik BASIC-Erweiterung (logischer Treiber) |
| Zeichensatztable f. Treiber für Vollgrafik |
| Sprungverteiler BASIC-Erweiterung (Adresse 0A7D6h) |

Folgende Grafikbefehle gibt es im BASIC: PSET, LINE, CIRCLE, PAINT, LABEL, SIZE, ZERO, HOME, GCLS, SCALE, SCREEN, POINT, XPOS, YPOS. Alle diese Befehle werden vom logischen Treiber vollständig verarbeitet und führen deswegen nicht zu BASIC-Fehlern. Der BASIC-Treiber übernimmt die Analyse der Befehlszeile und die Skalierung der Koordinaten. Dann wird der Sprungverteiler für die physischen Treiber aufgerufen.

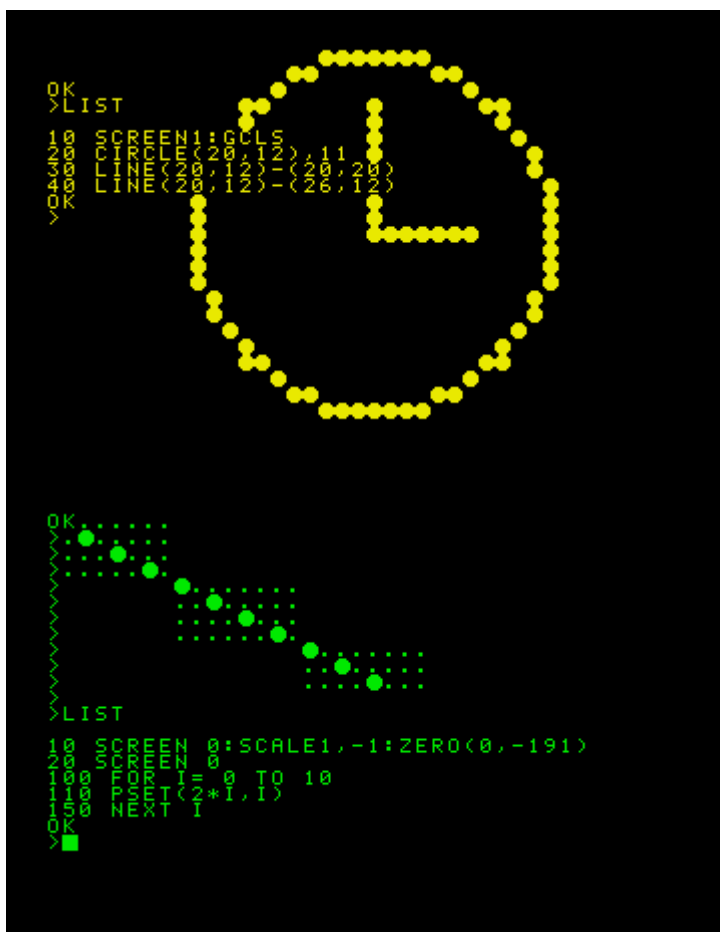
| GRPLOT.BIN Adr. 9800-A7FF | BASIC-Befehl | phys. Treiber Nr. | Bemerkung |
|------------------------------|--------------|-------------------------|---|
| 0A7D6h | pset | 1 | |
| 0A7D9h | line | 2 | LINE kennt den BOXED-Mode BF, dieser ist aber nicht implementiert |
| 0A7DCh | circle | 3 | |
| 0A7DFh | paint | 4 | PAINT (x,y)[,c[,d]] ist undokumentiert, und weder im Plotter- noch im Vollgrafiktreiber realisiert |
| 0A7E2h | label | 5 | |
| 0A7E5h | size | 6 | |
| 0A7E8h | zero | 7 | |
| 0A7EBh | home | 8 | |
| 0A7EEh | gcls | 9 | |
| 0A7F1h | scale | - | ist komplett im log. BASIC-Treiber implementiert, es gibt keinen korrespondierenden Aufruf im phys. Treiber |

| GRPLOT.BIN Adr. 9800-A7FF | BASIC-Befehl | phys. Treiber Nr. | Bemerkung |
|------------------------------|--------------|-------------------------|---|
| 0A7F4h | screen | - | ist komplett im log. BASIC-Treiber implementiert, es gibt keinen korrespondierenden Aufruf im phys. Treiber |
| 0A7F7h | point | 10 | POINT liefert für Plotter immer 0 |
| 0A7FAh | xpos | 11 | |
| 0A7FDh | ypos | 12 | |

Den reassemblierten und kommentieren Quellcode des Treibers GRAF.COM gibt es im Treiberpaket [grafik.zip](#).

Treiber für Pseudo-Grafik

Im Zeichensatz des Z9001 befinden sich spezielle Grafikzeichen, mit denen eine sogenannte „Viertel-Grafik“ mit einer Auflösung von 80x48 „Pixel“ realisiert werden kann. Mit meinem Treiber [grafik.zip](#) **GRAF14** können die BASIC-Grafik-Befehle mit dieser Auflösung genutzt werden und so ein bisschen Grafikprogrammierungsluft geschnuppert werden. Außerdem gibt es aus der Entwicklungsphase noch einen Grafiktreiber **GRAF11** mit der nativen Textauflösung von 40x24 „Pixeln“.



Grafik-Treiber GRAF11 für Testauflösung 40x24 „Pixel“. Im Download-Paket ist auch eine angepasste Version des Uhrenprogramms UHR11.; Weiteres Beispiel mit einer frühen Testversion. Die kleinen Punkte sind 'gelöschte' Pixel bei einer Ansteuerung wie beim originalen Grafikzusatz; es wurden also immer 8 „Pixel“ in einem Byte gleichzeitig bearbeitet.

