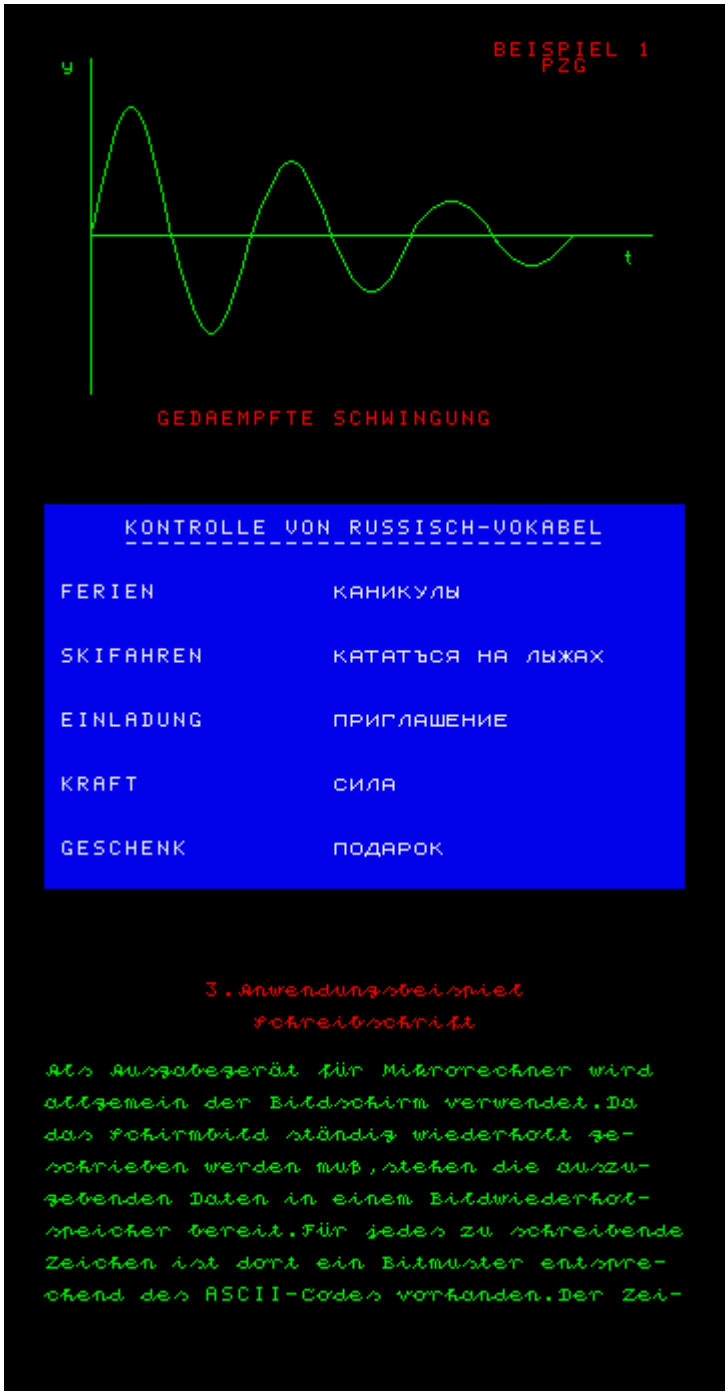
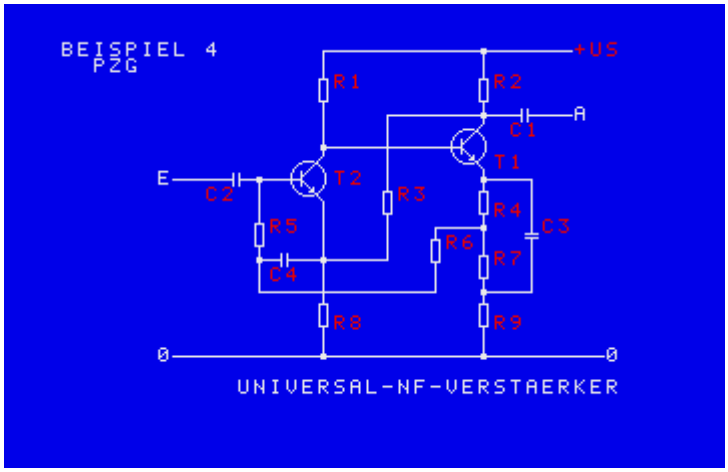
Programmierbarer Zeichengenerator

In „Mikroprozessortechnik“ 7/1988, S. 221 wurdE ein Programmierbarer Zeichengenerator für den KC 87 vorgestellt:

Programmierbarer Zeichengenerator für den KC 85/1 und KC 87

Die Hardwarekonfiguration des Kleincomputers KC85/1 ermöglicht bisher eine beschränkte, nicht frei programmierbare, quasigrafische Darstellung der auf einem EPROM vorhandenen Grafikzeichen (ASCII-Code 128 bis 255 Dez.). Mit der Hardware-Ergänzung „Programmierbarer Zeichengenerator (PZG)“ können durch den Nutzer 126 Grafikzeichen frei programmiert (Erzeugung der gewünschten Bitmuster), auf dem Bildschirm dargestellt und ggf. über einen Drucker (K6313 o. ä.) ausgegeben werden.





Der PZG besteht aus einer kleinen Leiterplatte mit einem zusätzlichen 1- KByte-RAM (2xU 214), einem herkömmlichen Zeichensatz auf einem 2-KByte-EPROM (224 programmierte ASCII-Zeichen 32 bis 255 Dez.) und einem Flip-Flop (DL074) zur wechselseitigen Umschaltung zwischen der Standard-Grafik und der frei programmierbaren Grafik.

Die Ansteuerung des PZG erfolgt über 3 Schaltadressen, die sich im normalerweisen nicht zugänglichen Farb-RAM als Kurzmerkspeicher befinden. Sie dienen zum Einschalten des PZG, zum Beschreiben des PZG-RAM und zum Aktivieren des PZG-RAM.

Die realisierte Installierung der PZG-Leiterkarte, die zweckmäßigerweise an Stelle des Farbmoduls (der Farbmodul befindet sich auf der PZG-Karte) unter Nutzung der Standard-Steckverbindung vorgenommen wurde, verändert äußerlich den KC85/1 nicht und sichert die herkömmliche Funktionsfähigkeit des Kleincomputers.

Es bestehen die Möglichkeiten der Verwendung verschiedenartiger Zeichensätze, z. B.:

- grafische Funktionsdarstellungen
- kyrillische Buchstaben (Russisch-Zeichensatz)
- lateinische Schreibschrift und
- Darstellung elektronischer Schaltungen.

Ergänzende Software ermöglicht die Programmierung von Grafikzeichen (MC-Programme) und das Ausdrucken der erzeugten Bildschirminhalte.

Die Betreibung des PZG ist sinnvollerweise mit Farbmodul (aber auch ohne Farbfernsehgerät und RGB-Satz) zu realisieren. Neben einer selbst gefertigten PZG-Leiterkarte liegen eine Kurzdokumentation zur Hardwarelösung und Programmbeschreibung beispielhafter Demonstrations-Software vor.

TU Magdeburg, Büro für Neuererwesen, Bundrock

Th. Bundrock erinnert sich: *Meiner heutigen Erinnerung nach entstand der „Programmierbare Zeichengenerator“ aus der der Darstellungsnot (ein Plakat war uns zu „poplig“) für ein „Jugendobjekt“ in der Chemischen Industrie (Steuerung und Regelung einer SKL-Ofenanlage in Böhlen/Leipzig) bei einer (Jugend-) MMM in Leipzig. Von der Hardware und der PZG-Programmierung habe ich als ausgebildeter Verfahrenstechniker und Hobbyprogrammierer naturgemäß weniger Kenntnis. Aber da gab es noch einen KC85/1-Jugendwettbewerb, an dem ein 16-jähriger Schüler (nicht ich !) der Abiturklasse der Uni Magdeburg teilnahm, in Assembler programmierte und die Aufgabe(n) löste, aber wegen der „Basic“-Forderung (von uns/mir unerkannt) aber scheiterte... Lehrreiche Geschichte. Der PZG entstand als Idee von Herrn Licht (Leipzig ?, sicher müßte man ihn zum PZG als Erstentdecker befragen, habe allerdings keinen Kontakt), die Nachnutzung wurde seinerzeit von mir abgewickelt (wieviele ?, vielleicht bis zu 20 ?). Die Leiterplatte wurde von dem Schüler in Assembler auf dem KC85/1(Z9001) gezeichnet (Fehler wurden mit gelöteten Kabeln „überwunden“), im Leiterplatten-Labor hergestellt, bestückt und „unsichtbar“ eingebaut. Wahrscheinlich habe ich noch die Unterlagen, aber vor Mitte August komme ich nicht dazu, diese zu sichten. Das „Ur-Tier“ habe ich mit Sicherheit noch (oder auch den 2. noch). Bestimmt sind noch leicht defekte Leiterplatten (oder die Folien) da. Den Monitor-Anschluß kann ich auch noch realisieren, denn der Robotron-TV mit RGB-Satz läuft (zwar selten) noch immer seit 19 Jahren in unserer Küche - köstlich - das kriege ich schon irgendwie hin ! Aber ob die Software noch läuft ? Wenn ich denn alles zusammen habe, sende ich gern informative Unterlagen. Eingesetzt haben wir im Sportbereich den Z9001 (immer gleich mit Farbmodul) zur Turnierauslosung bei Schach-Turnieren bis 1993 (Basic-Programmierung) ohne vordringlichen Einsatz des PZG. Im weiteren hatten wir im Einsatz: Sprachmodul, den ersten Plotter, Programmiermodul, E/A-Modul, Textmodul und noch ???* (von Juli 2007)

Downloads

- k02_pzg.zip

Programme zur PZG, s.u.

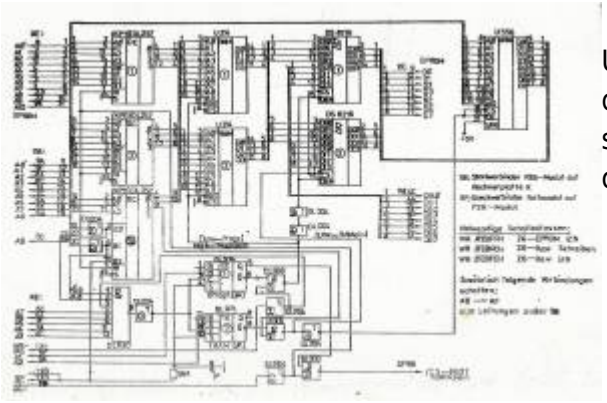
- unterlagen_pzg.pdf

02/2026, Danke an A.Senf Unterlagen zur PZG

- software_pzg.zip

02/2026 Analyse von PZG=ALL u.a.m.

Hardware

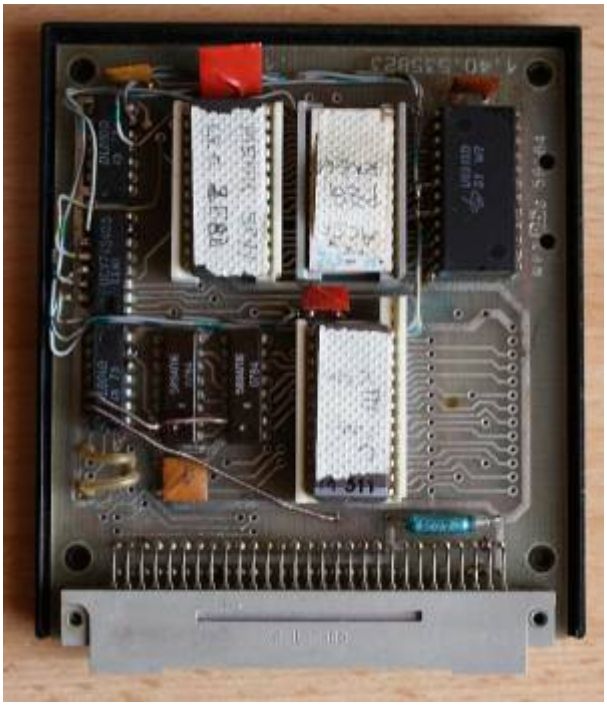


Bis Feb. 2011 hatte ich noch keinerlei Schaltungs-Unterlagen zur Hardware. Dank A.S. haben wir nun auch den Stromlaufplan (links). Folgendes ließ sich jedoch schon aus der Software ableiten und stimmt offensichtlich mit der Hardware überein:

Es gibt einen zusätzlichen 1K-RAM-Bereich. Die Ansteuerung wird wie folgt realisiert:

- Bei Schreibzugriff auf EBFEh wird die PZG aktiv, d.h. der zusätzliche 1K-RAM dient als Zeichengenerator für die Zeichen 80h..FFh. Zeichen 00h..7Fh kommen weiterhin aus dem normalen Zeichensatz-ROM
- Bei Schreibzugriff auf EBFFh wird die PZG deaktiviert, d.h. normaler Z9001-Betrieb mit Zeichensatz-ROM
- Bei Schreibzugriff auf 0EBFCh wird der zusätzliche 1K-RAM im Bereich E800h..EBEFh(EBFF?) eingeblendet; bei nachfolgendem Schreibzugriff auf EBFEh wird der PZG-RAM ausgeblendet und wieder der normale Farb-RAM zum Beschreiben aktiviert.

```
; Zeichensatz laden
LD HL,ZG-BUFF
LD DE,0E800H
LD BC,1008 ;3F0
LD (-5124),A    ;EBFC
LDIR
LD (-5122),A    ;EBFE
RET
```



Ein umgebautes ROM-Modul mit RAM (Besitzer A.S.), enthält die Programm-Pakete

- PZG/RNEW,
- MENU (HC-CAOS),
- C Copy 3.1 (R.Wobst),
- E Eprommer-Software (robotron)

Die Datei PZGOK.TAP enthält einen 16K-Speicher-Abzug eines KC87 mit gestecktem Modul. Offenbar sind die Adress-Bereiche im ROM-Modul auch geändert, sie passen nicht zu einem originalen ROM-Modul.

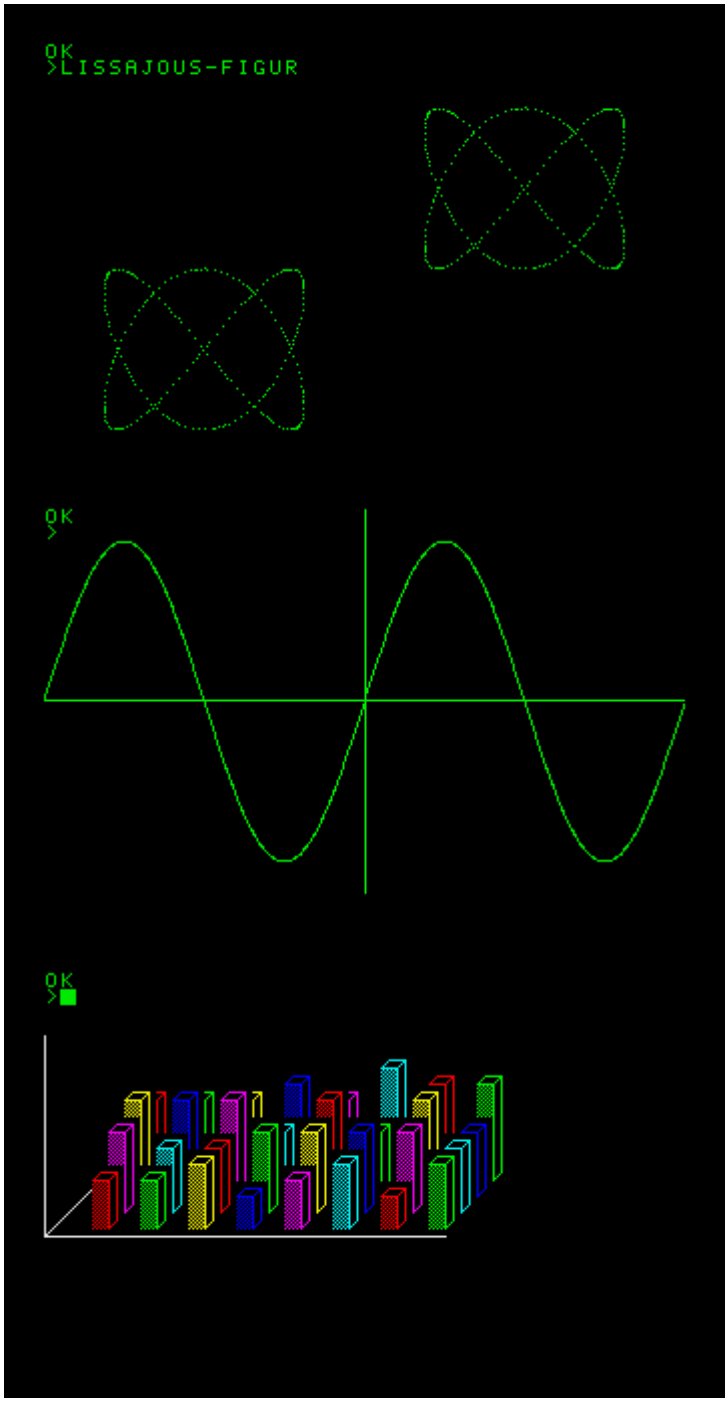
```
Platz 1:    C, E            A000..A3FF COPY 3.1, A400-AFFF EPROM
Platz 3:    MENU            9800..9FFF HC-CAOS
Platz 4:    PZG/RNEW        A800..AFFF RNEW, PZG
Platz 5:    2K RAM            ????..????
```

2023: U. Zander besitzt eine PZG-Platine: https://www.sax.de/~zander/z9001/module/pzg.html

Software

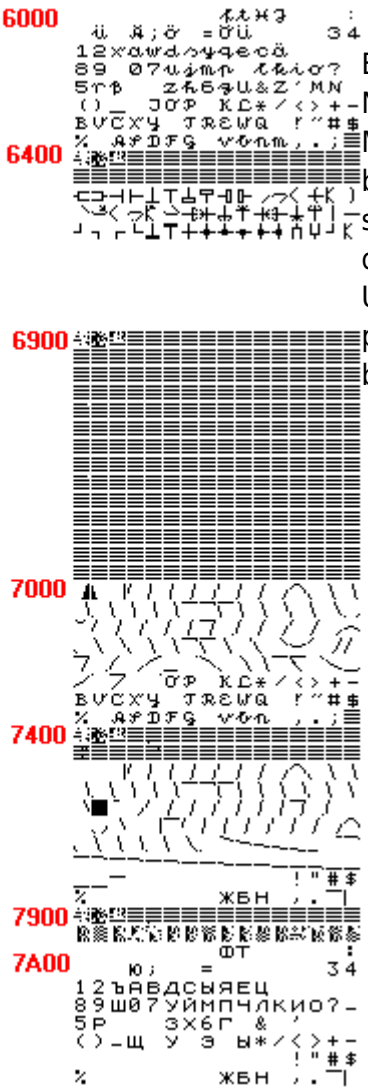
PZG_COM.TAP	Editieren bzw. Erstellen von Zeichen/Zeichensätzen
PZG=ALL_sss.tap	das große Demonstrationsprogramm (laden im BASIC)
PZ-ENDE_COM.TAP	fertige Zeichensätze call*6400 Schreibschrift call*6900 Schaltzeichen call*7400 Funktionen/Schreibschrift call*7900 Funktionen call*7e00 kyrillische Buchstaben call*7e20 EPROM-EIN
ZG-ROM_COM.TAP	Zeichensatz Z9001 mit Umlauten

ZG-ROMA_COM.TAP	Zeichensatz Z9001
FUNKT_TXT.TAP	Quellcode MC-Anteil
FUNKT_COM.TAP	compilierter MC-Anteil
FUNKT_SSS.TAP	Funktionskreuz, Lissajous-Figur
HARDC_TXT.TAP	Quellcode MC-Anteil
HARDC_COM.TAP	compilierter MC-Anteil
SINUS_SSS.TAP	DOKE 863,5000:RUN, erzeugt Sinus-Kurve
DIAG_SSS.TAP	DOKE 863,5000:RUN erzeugt 3D-Stapel-Diagramm
RNEW_PZG.TAP	wie im ROM. Achtung! hier stimmen die Adresse nicht! Das Prog. muss auf A800 geladen werden!



PZ-ENDE_COM call*7e20 EPROM-EIN kann nicht stimmen.
RNEW_PZG.TAP Die Ladeadresse passt nicht! Das Prog, muss auf A800 geladen werden. Vermutlich fürs EPROM-Brennen auf 1000 abgelegt?

PZG=ALL



So sehen die Zeichensätze im Speicher abgelegt aus. (PZ-ENDE_COM.TAP, ist auch in PZG=ALL _sss.tap so enthalten). Nach einem Zeichensatz folgt ein kurzes Maschinencodeprogramm zum Laden des Zeichensatzes. Man beachte, dass die Zeichen NICHT in ASCII-Reihenfolge aufgebaut sind. Die Zeichen werden über Grafikzeichen ausgegeben, und deren Zuordnung zu den ASCII-Zeichen erfolgt im OS über eine Umsetzungstabelle. In der Praxis bedeutet das, dass die passenden Zeichen des Kursiv-Zeichensatzes dadurch trotzdem beim Druck auf die jeweilige Taste erscheinen.

Das BASIC-Programm PZG=ALL enthält diverse Demos in Endlosschleife. Das Programm ist aus Maschinecode-Teilen und drei BASIC-Teilen zusammengesetzt. Durch Manipulation des Programmanfang-Zeigers (DOKE 863,xxxx) werden die BASIC-Programme selektiert. Zwischen den Teilen steckt Maschinencode:

```
0401h  BASIC-Programm, startet Programm 3
1100dez MC-Unterprogramme zur Laufzeitgenerierung eines Zeichensatzes
(Vollgrafik für Funktionsplotter)
2000dez BASIC-Programm (Funktionsplotter)
6000h..7EFFh Zeichensätze und Ladeprogramme (s.a. PZ-ENDE.COM)
8000h  das Haupt-BASIC-Programm mit 4 Beispielen
```