

Spaß mit einem Schneider CPC (Emulator)

Lightning Talk @4FPM (Folien/Minute) – in der Zeit ist es nicht möglich alles zu lesen ;-)

Jens Thiele¹

<https://karme.de/cpc>

4. Juli 2026

¹karme@karme.de

1985

Zeitreise



Schneider CPC

eigentlich Amstrad CPC (Colour Personal Computer)

8-Bit CPU Z80 mit 4MHz, 64KiB RAM

1984 CPC464

eigentlich kein Diskettenlaufwerk, nur Kassette (das will man wirklich nicht zurück). Es gab von Vortex ein 5¼" Doppellaufwerk.

1985 CPC664

3" Diskettenlaufwerk (nicht 3½"!)

auch noch 1985 CPC6128

128KiB via Bank Switching

Amstrad CPC



Photo von Bill Bertram CC BY-SA 2.5 (der Schneider war nicht so schön bunt)

Emulator an den Start bringen

Kurz: machbar (notfalls im Webbrowser)

```
10 install Debian ;-)  
20 REM get an emulator (MAME und/oder caprice32,  
    oder via tiny8bit im Webbrowser)  
30 sudo apt install mame mame-tools dos2unix  
40 get the system ROMs  
50 build iDSK  
60 edit your first BASIC file  
$ echo '1 print "hello"'>hello.bas;unix2dos hello.bas  
70 create disk with hello.bas  
$ iDSK -n hello.dsk -i hello.bas -t 0  
80 run the emulator:  
$ mame cpc664 -flop1 hello.dsk -ab '\n\nrun "hello\n'
```

Erfolg

Amstrad 64K Microcomputer (v2)

©1984 Amstrad Consumer Electronics plc
and Locomotive Software Ltd.

BASIC 1.1

```
Ready  
run "hello  
hello  
Ready  
■
```

Locomotive BASIC²

eigentlich ganz nett, hat ein paar überraschende Features

damals bin ich auf dem CPC über BASIC nicht hinausgekommen

bleibt halt BASIC, aber das war die Einstiegsdroge

sehr langsam ⇒ also Assembler

²**B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode

Assembler - erster Versuch

keine Zeit

Assembler - zweiter Versuch (1)

moderne Mittel verwenden

keine experimentelle Archäologie

mit Pasm Cross-Assemblieren für den Z80

```
sudo apt install pasmo; cp -v 664EN_1.DSK h.dsk
```

```
cat > h.asm <<EOF
```

```
org 100h
```

```
ld de,string
```

```
ld c,9
```

```
call 5
```

```
ret
```

```
string: db 'Hello$'
```

```
EOF
```

```
pasmo h.asm hello.com; iDSK h.dsk -i hello.com -t 2
```

```
name cpc664 -flop1 h.dsk \
```

```
    -ab '\n\n|cpm\n
```

```
        hello\n'
```

Assembler - zweiter Versuch (2)

```
CP/M 2.2 - Amstrad Consumer Electronics plc
```

```
A>    hello
```

```
Hello
```

```
A>█
```



0:12 / 0:20



weitere Schritte ...

Details auf <https://karme.de/cpc>

Z80 Assembler ohne CP/M mit pasmo

ändert sich nicht wirklich viel (Firmware aufrufen anstatt CP/M)

BASIC und Assembler mischen

load+call Befehle mit Parameterübergabe

Assembler bleibt ein Zeitgrab

SDCC Small Device C Compiler (Cross-Compiler)

```
$ apt install sdcc
```

C ist schon wesentlich angenehmer

BASIC und C mischen

RSX Befehle (Erweiterungsmechanismus für das BASIC)

```
|scrollup |scrolldown |scrollleft |scrollright
```

crt0 / putchar muß man selber machen

Datenexfiltration

am einfachsten über die parallele Schnittstelle

```
print #8,"Hello world from BASIC"
```

```
_putchar::
```

```
    ld a,1
```

```
1$:
```

```
    call 0xbd2b ; mc print char
```

```
    jp nc,1$
```

MAME hat auch eine serielle Schnittstelle

Konfiguration und Benutzung aus C/Assembler ist etwas umständlicher (RSX Befehle)

(caprice32 kann auch Disketten schreiben, MAME hat auch eine Lua Schnittstelle)

Grafik / Bildschirmspeicher

kein Textmodus!, Adressierung etwas umständlich
(insbesondere, wenn man Hardware-Scrolling benutzt)

mode 0

160x200 16 aus 27 Farben, 2 Pixel/Byte, rechteckige Pixel

mode 1

320x200 4 aus 27 Farben, 4 Pixel/Byte, quadratische Pixel

mode 2

640x200 2 aus 27 Farben, 8 Pixel/Byte, rechteckige Pixel

Lisp auf den CPC bringen

C ist ja schön und gut aber, ...

benutze eigentlich fast immer (gauche) Scheme,
wenn es geht

wenigstens ein einfaches Lisp?

erster Versuch: **tinylisp**

von Robert A. van Engelen

freie Software

99 Zeilen C

zweiter Versuch: Lisp in 1k lines of C

auch von Robert A. van Engelen

auch freie Software

1018 Zeilen C

kleine Spiel-Engine für den CPC

Lisp Game Jam

zweimal im Jahr (Frühling und Herbst)

Zeit und Raum ('Adventures' / Visual Novel)



Part of the game is to find out what works. But you can go north, west, east and south (keys: n w e s or cursor keys). The next step will look like a crash but we are just using the video memory as buffer



[lisp1ksloc \(https://karme.de/cpc/lisp1ksloc\)](https://karme.de/cpc/lisp1ksloc)

zurück zum Anfang

Assembler in Lisp auf dem CPC

sehr nervig an Assembler sind ja die vielen leicht verschiedenen Dialekte/Syntaxen. Was liegt also näher als noch einen zu machen ;-)

```
(asm  
  (ld 'hl 48897)  
  (push 'ix)  
  (acall 48298)  
  (pop 'ix)  
  (ld 'a 0)  
  (rla)  
  (ld 'd 0)  
  (ld 'e 'a)  
  (ret))
```

zu schnelles Ende

Details hier: <https://karme.de/cpc>

gerne auch per E-Mail <mailto:karme@karme.de>

