

EMRSIM

2022 ist ein LC-80 aufgetaucht, der in zwei ROMs einen Simulator für den **u881** (Z8-Prozessor) enthält (<https://www.robotrontechnik.de/html/forum/thwb/showtopic.php?threadid=20044>).

Ich habe die Software reassembliert und analysiert.

Download

- emrsim.zip

Benutzung

- emuliert einen Z8-Prozessor (U881)
- U881-Speicher wird im LC-80 ab 2400h abgebildet
- keine Unterscheidung CODE-Speicher, RAM-Speicher (Ide, Idc)
- Register liegen im RAM, Arithmetik etc. wird über Z80-Register A erledigt. Flags werden gesetzt/beachtet.
- vier ungültige OP-Codes dienen als Simulator-Befehle (0f, 1f, 2,f 3f).
- braucht
 - 0000-07FF 2Kx1 - System
 - 0800-123F 4K ROM EMRSIM
 - 2000-23FF 1K RAM
 - 2400-2BFF 2K RAM-Erweiterung f. Z8-Programm

Speicheradressen

2003/2004	Startadr Z8-Program (relativ, also z.B. 000c)
2005	erstes Register f. Trace
2006	zweites Register f. Trace
2007	drittes Register f. Trace
2008	0f single step, 1f trace
2100-217F	Z8-Register
2180	Peudoregister 80 = Lo PC (absolute Adresse)
2181	Peudoregister 81 = Hi PC (absolute Adresse)
2182	Peudoregister 82 = Befehlszähler
21F0-21FF	Z8-Systemregister F0..FF

Start auf Adr. 0800.

Z8-Programm wird auf Adr. (2003/4) + 2400h gelesen und gestartet.

Bei Autorun läuft das Programm autom. durch, nach jedem Z8-Befehl erfolgt eine kurze Anzeige.

Sonst wird ein Z8-Befehl ausgeführt, dann Warten auf Tastendruck.

Z8-Spezialbefehle

```
0f    ; spezial code: single step
1f    ; spezial code: trace
2f    ; spezial code: autorun
3f    ; spezial code: registeranzeige
```

Registeranzeige Der Emulator wird angehalten und der Inhalt eines Registers angezeigt.

- +,- schaltet die Register weiter
- ADR Registernummer editieren
- DAT Registerwert editieren
- 0..F ändert den Registerwert
- EX beendet die Registeranzeige und setzt das Programm fort.

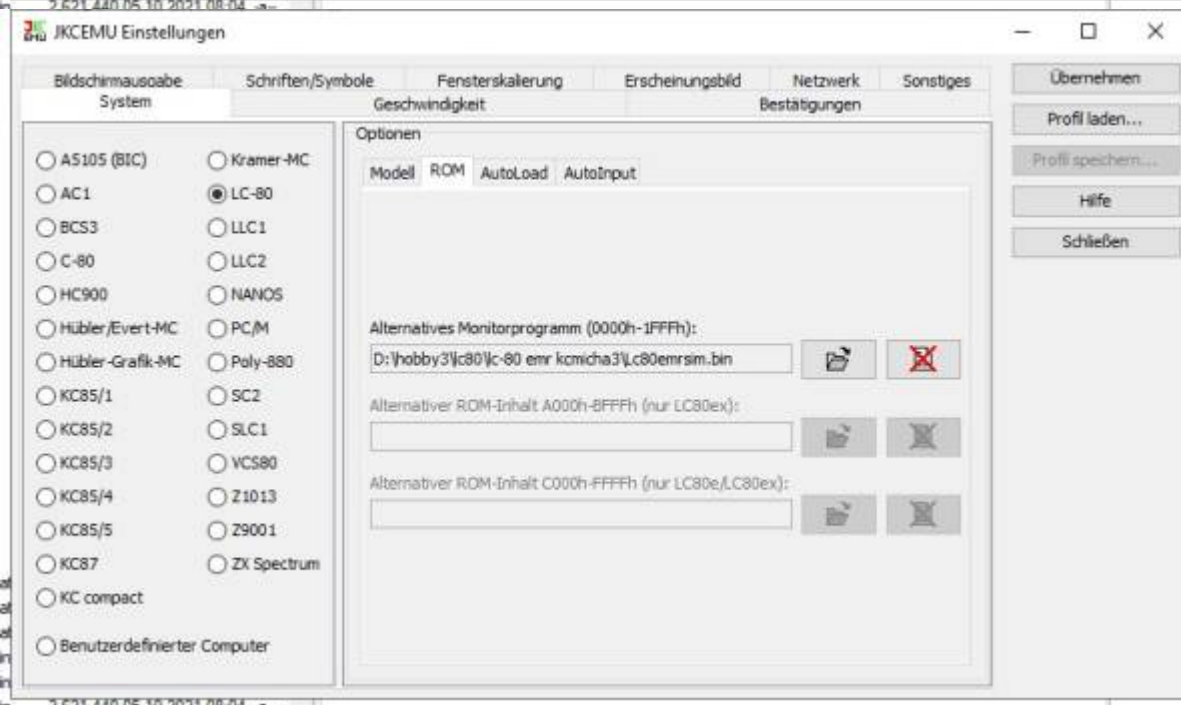
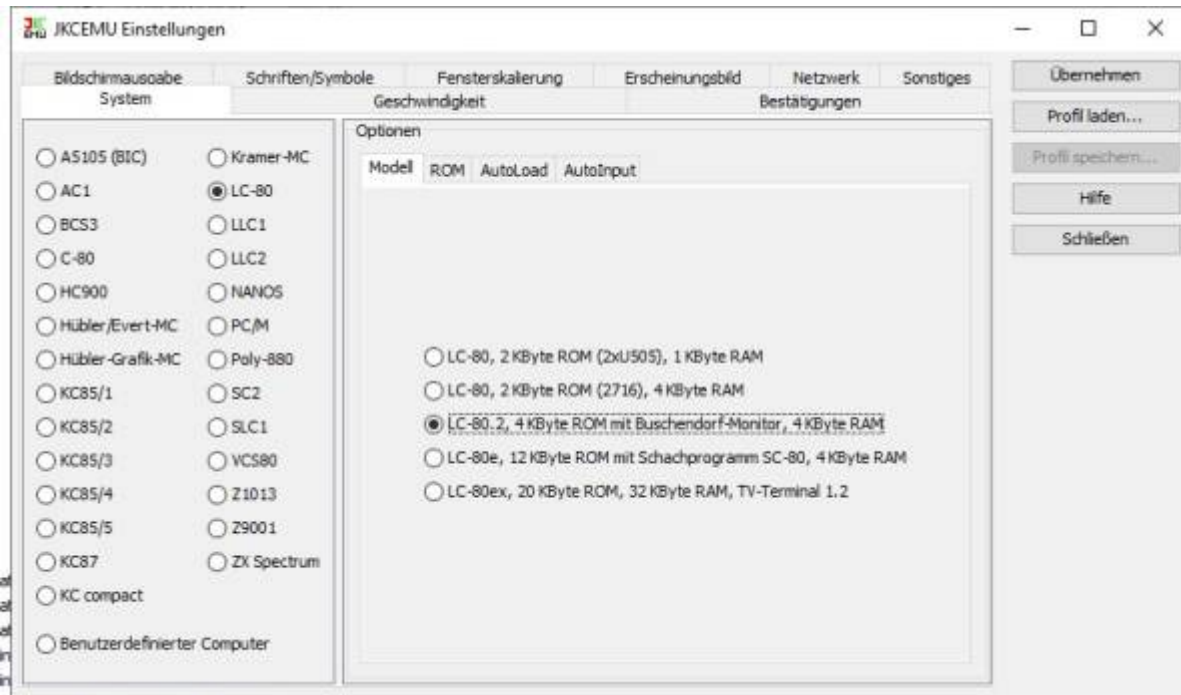
Ports Z8-Spezial-Port 2 wird emuliert → USERPORT

Nutzung im JKCEMU

Im Download-Paket liegt ein 8K-Binary lc80emrsim.bin. Dies enthält

```
0000-07FF orig 2Kx1-Monitor
0800-0FFF EMRSIM EPROM2
1000-17FF EMRSIM EPROM3
1800-1FFF U2716 EPROM4 (Brennersoftware, wird hier nicht benötigt)
```

Im JKCEMU ist in den Einstellungen der LC80-2 (4K ROM, 4K RAM) auszuwählen. Unter ROM wird als alternatives Monitorprogramm lc80emrsim.bin ausgewählt. Dann kann man den LC-80 starten.



Beispiel 1

Hochzählen eines U881-Registers. Trace-Mode, Anzeige Register 10

folgende Speicheradressen sind zu befüllen:

```

2003 : 00 00      relative Startadresse U881-Programm
2007 : 10        Anzeige Register 10 = r0
2008 : 1f        0F single step, 1F trace
                ab 2400 liegt das U881-Programm
2400 : 31 10      srp   #10h
2402 : 0C 2A      ld    r0,#42
2404 : FF        nop
2405 : 0E        m1:  inc   r0

```

```
2406 : 8B FD          jr      m1
```

Dann wird mit ADR, 0800, EX der Simulator gestartet. In der Anzeige sieht man jetzt FFFx, XX wird laufend hochgezählt. XX ist dabei der Inhalt des Registers 10h (r0).



Ändert man den Wert in 2008 auf 0f und startet das Programm, so wird nach jedem Befehl die Anzeige angehalten. Weiter geht es mit beliebiger Taste. Da die Schleife 2 Befehle umfasst, muss man 2x Taste drücken, ehe der angezeigte Wert hochzählt. Der interne Programmzähler steht in den Pseudo-Registern 80/81 Trägt man in 2005 81 und 2006 80 ein, erfolgt die Ausgabe der Adresse (absolut, 2400 ff).



Nun tragen wir auf 2008 wieder 1f ein (trace-Mode) und patchen das U881-Programm. Auf Adresse 2404 schreiben wir 3f (Registeranzeige). Starten wir das Programm, stoppt die Ausführung auf Adresse 0004=2404 und es erscheint die Registeranzeige:

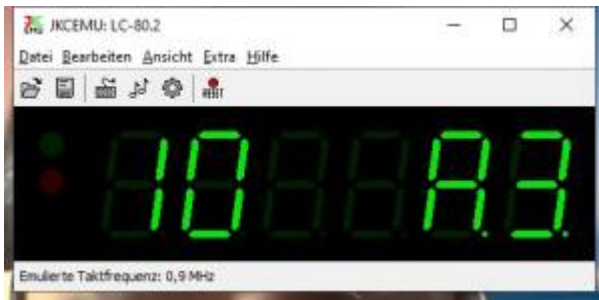


Links steht die Registernummer, rechts der Registerinhalt.

Mit den Zifferntasten wird das Register ausgewählt.
Mit DAT wechselt man zur Wert-Eingabe (Punkte rechts)



Wert auf A3 geändert:



Mit EX wird die Registeranzeige beendet, die TRACE-Anzeige läuft weiter.

From:

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

Permanent link:

<https://hc-ddr.hucki.net/wiki/doku.php/lc80/programme/emrsim?rev=1658388447>

Last update: **2022/07/21 07:27**

