

RAM-Floppy RAF2008

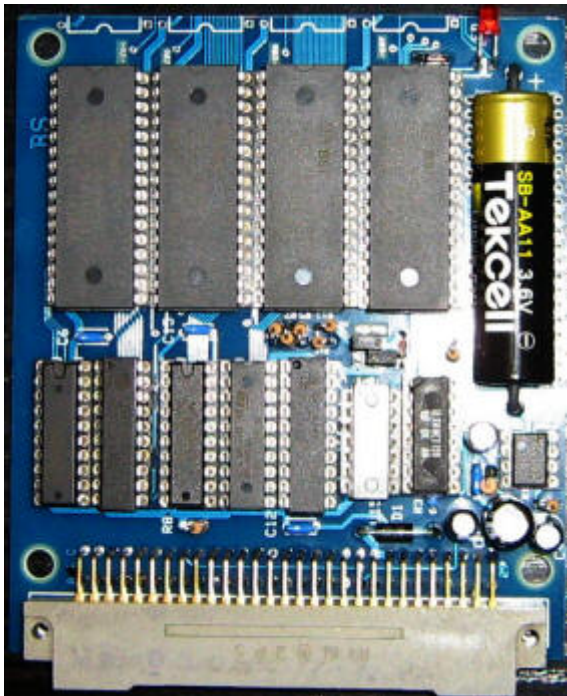
Die 2MB/4MB-RAM-Floppy RAF2008

In der DDR gab es verschiedene RAM-Floppys; auch als Alternative zu schwer beschaffbaren „richtigen“ Floppylaufwerken. An meinem Z1013 habe ich z.B. 2 NANOS-RAM-Floppys a 256 KByte angeschlossen. Auch von der Akademie der Wissenschaften gab es eine RAM-Floppy für K1520-Systeme, die sich durch modularen Ausbau, nachladbare Treiber u.a. hervorhob und so eine recht große Verbreitung fand (s. www.robotrontechnik.de)

Vorgestellt wurde die RAM-Floppy in der rfe 4/1987, S. 268.

2008 wurde diese Vorlage mit modernen Bauelementen und Batteriestützung durch Enrico Grämer als RAF2008 nachentwickelt. Auf obiger robotrontechnik-Seite ist das Projekt komplett beschrieben.

Auch am Z9001 lässt sich die RAF2008 als 2 MB/4 MB-RAM-Floppy problemlos betreiben. Die originalen Treiber, um Fehler korrigiert, laufen ohne große Änderungen auch direkt im Z9001-CP/A.





Meine RAF2008, mit 2MByte SRAM bestückt. Rechts ist der CP/M-Start und der Start des nachladbaren Treibers RAF2X20P.COM zu sehen.

Unterlagen

Auf <http://www.robotrontechnik.de/html/eigenbau/raf2008.htm> gibt es die Geschichte, Treiber und die Unterlagen des Entwicklers zur RAF2008. Bei U. Zander liegen aufbereitete Stromlaufpläne u.v.a.m.: <http://www.sax.de/~zander/z9001/module/raf.html>.

Downloads

- [prl2com.zip](#) Binary und Quellcode zu Standard-Treiber und zum Treiber mit Parität
- die kompletten orig. Treiber gibt's bei robotrontechnik (s.o.)

Wie immer gibt es im Downloadpaket die kompletten Quelltexte und weitere Informationen.

Einstellungen

Auf der Karte gibt es zwei Jumper. Für den Betrieb am Z9001 ist JP 2 zu stecken. Das begrenzt den Port auf Adressen 2xH. Der JP 1 bleibt für die erste RAM-Floppy offen, damit ist Port 20H eingestellt. Bei geschlossenem Jumper JP 1 gilt Port 24h.

Treiber

Neben einem angepassten BIOS, das Treiber für die RAM-Floppys enthält, kann auch ein nachladbarer Treiber genutzt werden:

RAF2X20P.COM

wird von der Floppy-Diskette geladen und installiert die RAF2008 als Laufwerk P. Beim ersten Start wird dabei die RAM-Floppy formatiert, d.h. der Directory-Bereich mit 0E5h überschrieben. Beim erneuten Start unterbleibt die Formatierung, so dass die Daten auf der RAM-Floppy erhalten bleiben. Dank der Stützbatterie bleiben die Daten auf der RAM-Floppy bis zu 1 Jahr erhalten, ohne das der Rechner angeschaltet sein muss!

Für die zweite Portadresse (bei einer zweiten RAM-Floppy) gibt es den Treiber **RAF2X240.COM**. Dieser bindet die zweite RAM-Floppy als Laufwerk O ein.

Die nachladbaren Treiber schreiben sich ans obere RAM-Ende TPA und verringern so dauerhaft den für Programme zur Verfügung stehenden Speicher. Erst mit einem Kaltstart (Reset) ist der Treiber wieder entladen!

Wem die beiden Treiber nicht reichen, der kann sich weitere Treibervarianten kompilieren. Ich habe die Programme reassembliert und neue Assemblerquellen geschrieben. Beim Treiber kommt das PRL-Format (→ [Nachladbare Treiber unter CP/M](#)) zum Einsatz, um den Treiber ans aktuelle TPA-Ende zu schieben. Beim Kompilieren eigener Treibervarianten muss dieses Format erzeugt und um einen Loader ergänzt werden.

Der eigentliche Treiber wird mit dem M80-Assembler und CSEG (ohne ORG-Adresse!) assembliert. Der Digital Research- Linker LINK131.COM erzeugt daraus beim Aufruf mit dem Parameter *op* eine PRL-Datei; diese umfasst den Treiber (auf Adresse 100h gelinkt) und die Verschiebe-Informationen. Mit meinem Patcher PRL2COM wird in diese PRL-Datei der Loader gepatcht und der Dateityp in COM geändert, so entsteht die gewünschte residente Systemerweiterung.

```
m80.com =treiber
link.com treiber[op]
prl2com treiber.prl
```

Ansteuerung

Die zur Ansteuerung nötige Adresse wird ähnlich wie bei der RAM-Disk nach mp 3/88 gebildet:

- Raf_C = RAF-Control Port zum Laden der Track- & Sector-Adresse „HiAdr“ via D-bus 0...7 und A-bus 8...14 (B-Reg).
- Raf_D = RAF-Data-I/O Port zum Schreiben/Lesen von max. 128 zusammenhängenden Bytes mit einem INIR-/OTIR-Befehl mit einem 7-bit Byte-Index „LoAdr“ auf dem A-bus 8...14 (B-Reg)

