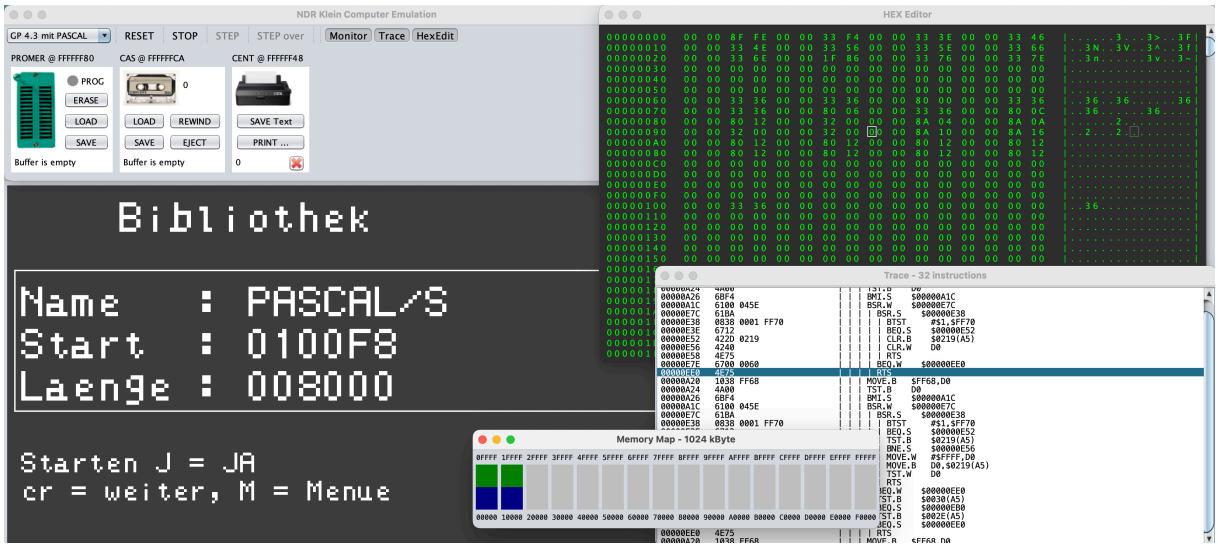
NDR Klein Computer Emulation

Version 1.2



Vorwort

Seit meiner Jugendzeit hat mich der von Rolf Dieter Klein konzipierte und im deutschen Fernsehen vorgestellte Computer fasziniert. All mein Taschengeld ist in den Erwerb der Baugruppen geflossen. Der Computer hat mich viele Jahre lang begleitet und mein weiteres Berufsleben geprägt.

Später habe ich beschlossen, mein angehäuftes Wissen anderen Menschen weiterzugeben und habe eine Homepage kreiert, die sich bis heute ausschließlich dem NKC widmet. Leider hat mich das Berufsleben sehr häufig davon abgehalten, die umfangreiche Seite angemessen zu pflegen.

Schon vor gut sechs Jahren hatte ich angefangen, eine Software-Emulation für den NKC umzusetzen, das Projekt ist dann leider liegengeblieben und vergessen worden. Im Sommer des Jahres 2023 hat mich Martin Merck kontaktiert, der einen eigenen Emulator entwickelt. Davon inspiriert, habe ich mein Projekt wieder aufgenommen und versucht so schnell wie möglich zumindest in einen Zustand zu bringen, den man als erste Version veröffentlichen kann. Ich danke Martin auch für die Bereitstellung seiner Quellen sowie Tipps bei der Lösung von Problemen.

Für die Umsetzung des Emulators waren die Vorarbeiten von Tony Headford im Rahmen seines Projektes „Java Amiga MachineCore“ essenziell. Die Emulation des Mikroprozessors MC 68000 ist gut aber war noch nicht perfekt. Das Projekt „Hexadecimal File Editor“ von Keith Fenske ist ebenfalls in den Emulator eingeflossen, um einen direkten Einblick in den Hauptspeicher der Emulation zu gewähren. Beide Projekte mussten in den Emulator kopiert werden, da etliche Erweiterungen zur Integration in den Emulator notwendig waren. Quellen und Links finden sich im Anhang dieses Dokuments.

Im aktuellen Installationspaket ist ein neues Grundprogramm GP-EDAS enthalten. Der darin enthaltene Editor und Assembler EDAS*4 des AC1 wurde erstmals in der Zeitschrift Funkamateur in den Ausgaben 01 bis 04 von 1987 veröffentlicht und von Steffen Reimer (DL2LCE) in das Grundprogramm 2018 integriert.

Die Umsetzung eines so komplexen Projektes benötigt seine Zeit. Ich würde mich wirklich sehr über Feedback freuen. Egal ob Lob, Kritik, Wünsche oder Fehlermeldungen. Ich bin für fast alles offen.

Ein abschließender Dank geht an meine Frau, die es erträgt, wenn ich stundenlang ohne für sie erkennbares Ziel vor meinem MacBook sitze.

Inhaltsverzeichnis

Versionshistorie

Version 1.0 (April 2024)

Die initiale Version mit der Emulation der Motorola 68000 und 68008 Mikroprozessoren mit Unterstützung der Baugruppen

- BANKBOOT (Speicher ab Adresse 0x0000)
- FLO2 (Diskketteninterface)
- CAS (Kassetteninterface)
- PROMER (Programmiergerät)
- CENT (Druckerschnittstelle)
- UHR (Echtzeituhr)

Version 1.10 (November 2024)

Es ist viel Arbeit in die zweite Version geflossen, um einige Fehler zu beheben und etliche neue Features umzusetzen. Die wichtigsten Neuerungen nachfolgend als Liste. Die Beschreibungen in den folgenden Kapiteln wurden entsprechend angepasst und erweitert.

- Emulation des Zilog Z80 Mikroprozessors
- Unterstützung unterschiedlicher Diskettenformate
- Emulation der Baugruppe IOE (Ein- und Ausgabe)
- Überarbeitung des Oberfläche (Verwendung von Icons)
- Anzeige einer beliebigen Bildschirmseite der GDP64K
- Installationsdateien für MacOS und Windows
- Fehlerbehebung beim Setzen der Taktfrequenz
- Überarbeitung der vorgegebenen Konfigurationsdateien
- Erweiterte und korrigierte Dokumentation
- Zusätzliche Disketten-Images für CP/M 68K
- Dateinamen von Disketten-Images dürfen Leerzeichen enthalten
- Unterstützung von Label-Dateien für das Trace-Fenster
- Erweiterung des Menüsystems
- Konfiguration der DIL-Schalter auf der Baugruppe KEY
- Allgemeine Verbesserungen und Fehlerbehebungen

Version 1.20 (September 2025)

Es ist mal wieder viel Zeit in die Erweiterung des Emulators geflossen. An dieser Stelle möchte ich mich bei Jürgen bedanken, der viel Zeit mit Tests verbracht und einige Ideen beigesteuert hat.

- Einstellungsdialog mit Sprachauswahl deutsch/englisch
- Wiederherstellungsmöglichkeit der enthaltenen Dateien und Konfigurationen
- Verbesserung des Fenster-Handlings
- Verbesserungen bei Single-Step
- Verbesserung der GDP64K Emulation, verringerter Energiebedarf
- Baugruppe BANKBOOT funktioniert jetzt auch mit Z80
- Umsetzung der Baugruppe SER (Beta-Status)
- Baugruppe GDP64HS unterstützt RMW-Modus und das Rücklesen der Bildspeichers
- Verbesserte Darstellung von bewegter Grafik im Monitor-Fenster
- Frei einstellbare Vordergrund- und Hintergrundfarbe des Monitors
- Freie Skalierbarkeit des Monitor-Fensters
- Erneutes automatisches Laden der ROMs nach einem RESET des Prozessors
- Tool zum Konvertieren von Intel HEX Dateien ins Binärformat
- Tool zum Konvertieren von Binärdateien in das Intel HEX Format
- Tool zum Zerlegen von Binärdateien in verschiedene Größen
- Tool zum Zusammensetzen mehrerer Binärdateien
- Tool zum Bearbeiten von CP/M68K Disketten Images
- Frei einstellbares Verzeichnis für Benutzerdaten
- Erststart nach Installation wählt eine sicher funktionierende Standard-Konfiguration

Weitere Features sind bereits in der Planung …

Die Emulation

Der zentrale Grund, warum Java als Grundlage für diese Software gewählt wurde, liegt in seiner Plattformunabhängigkeit. Die Fähigkeit von Java, auf verschiedenen Systemen ausgeführt zu werden, ermöglicht es Benutzern, die Software sowohl auf Macintosh-Betriebssystemen als auch unter Windows auszuführen.

Darüber hinaus bietet Java eine stabile und sichere Umgebung, was zu einer zuverlässigen Leistung und einem erhöhten Schutz vor potenziellen Sicherheitsrisiken führt.

Hardware-Treiber

Zur Emulation der Hardware-Baugruppen des NKC wurde ein Konzept entwickelt, mit dem sich verschiedene Treiber aktivieren lassen, die jeweils genau eine Baugruppe behandeln. Die Treiber sind fest in den Emulator integriert und lassen sich bei zukünftigen Programmversionen erweitern.

Die Basisadressen der Hardware-Baugruppen können frei festgelegt werden, allerdings ist die allgemein verfügbare Software des NDR Klein Computers nicht darauf ausgelegt. Daher sollten die Adressen in der Konfigurationsdatei gemäß dem NKC-Standard vergeben werden.

Hardware-Treiber bestehen mindestens aus dem Treiber selbst und einem Memory-Handler. Zusätzlich kann ein Darstellungsbereich (Frame) und ein parallel zum Emulator ablaufendes Programm (Thread) implementiert werden. Die Frames dienen zur Darstellung im Haupt-Fenster des Emulators, die Threads zur Aktualisierung des Inhaltes dieser Frames.

Systemvoraussetzungen

Die Emulation läuft auf jedem Computer, auf dem das JAVA Runtime Environment (JRE 17) lauffähig ist. Getestet wurden die Installationsdateien unter

- macOS Sequoia auf einem MacBook Pro mit Apple Silicon M1 / M3
- macOS Catalina auf einem iMac 27 mit Intel Core i7
- Windows 10 und 11 auf einem Dell Desktop Computer
- Windows 11 unter VMWare Fusion auf einem MacBook Pro M3

Zukunftsaussichten

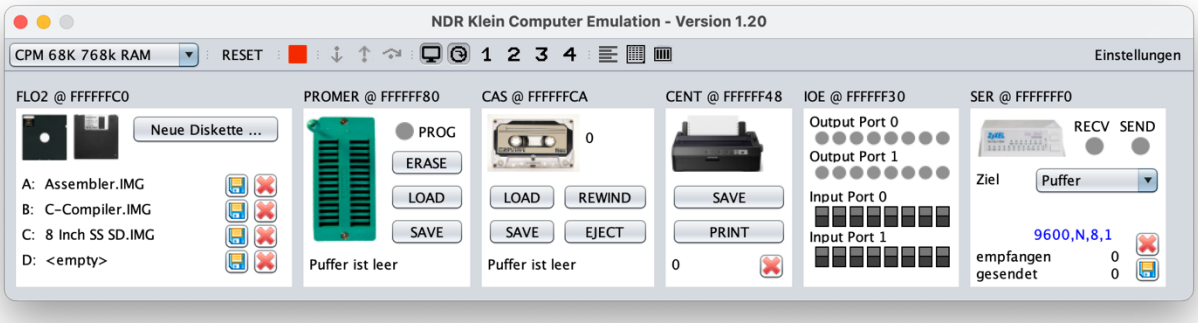
Die Entwicklung ist lange noch nicht abgeschlossen. Hier eine kleine Liste der Sachen, die ich noch auf dem Plan habe:

- Umsetzung weiterer Baugruppen
- Integration weiterer Tools
- Umsetzung des Mikroprozessors 8088
- Integration eines Disassemblers
- Programmieren von EEPROM im Hauptspeicherbereich

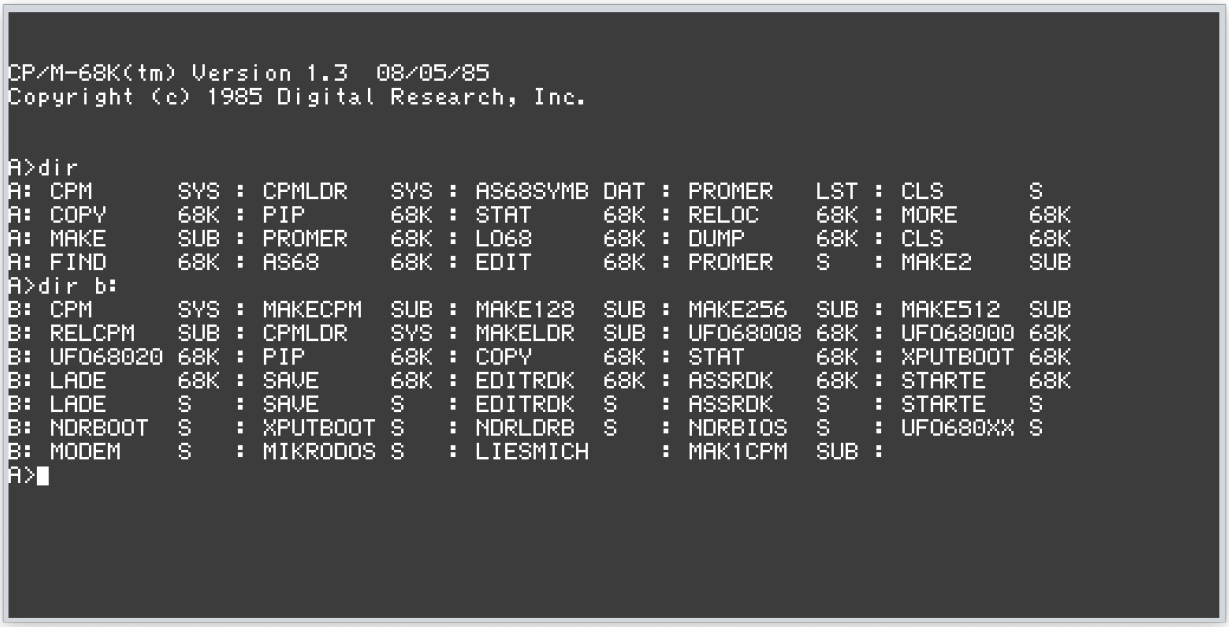
Ich bin auf Eure Rückmeldungen und Wünsche gespannt.

Das Hauptfenster

Das Hauptfenster dient zur Steuerung der Emulation. Unter der Symbolleiste werden die Baugruppen angezeigt, welche durch die Konfigurationsdateien ausgewählt wurden. Im Bild sind die Rahmen der Baugruppen FLO2, PROMER, CAS, CENT und IOE sichtbar. Reihenfolge und Umfang der Rahmen lassen sich über die Konfigurationsdateien anpassen.



Direkt unter dem Hauptfenster wird der Darstellungsbereich für den Bildschirm der Baugruppe GDP64K des NKC angezeigt. Dieses Fenster bleibt beim Verschieben des Hauptfensters immer unter dem Hauptfenster positioniert.



Die Symbol-Leiste

Über die Symbolleiste im Hauptfenster lassen sich wichtige Funktionen des Emulators steuern.

Umschalten der Konfiguration

Das Drop-Down-Menü auf der linken Seite zeigt alle Konfigurationsdateien aus dem Ordner CONFIG im Benutzerverzeichnis an und gestattet das schnelle Umschalten der Hard- und Softwarezusammenstellung, die während der Emulation genutzt wird.

Beim ersten Start des Emulators wird eine Standardkonfiguration mit dem Namen „DEFAULT“ aktiviert. Sobald eine andere Konfiguration ausgewählt wurde, wird die Standardkonfiguration aus der Liste ausgeblendet und kann nicht mehr aktiviert werden.

RESET

Setzt das emulierte System zurück und startet die Emulation erneut mit der gleichen Konfiguration. Bereits geladene Dateien und Pufferinhalte bleiben erhalten. Ab Version 1.20 werden alle in der Konfigurationsdatei angegebenen ROMs auf BANKBOOT und ROA64 neu eingelesen.

 *STOP*

Hält die Emulation an und aktiviert den Einzelschrittmodus. Wenn die Emulation angehalten ist, wird an der gleichen Stelle das Symbol für das Starten der Emulation angezeigt.

 *RUN*

Beendet den Einzelschrittmodus und startet den automatischen Ablauf der CPU. Wenn die Emulation gestartet ist, wird an der gleichen Stelle das Symbol für das Anhalten der Emulation angezeigt.

 *Step In*

Die Einzelschritt-Funktionen stehen nur dann zur Verfügung, wenn die Emulation angehalten wurde. STEP IN führt einen einzelnen Befehl der CPU aus. Jeder STEP-Befehl aktiviert automatisch das Trace-Fenster, in dem die im Einzelschrittmodus ausgeführten Instruktionen protokolliert werden.

 *Step Out*

Führt die Instruktionen so lange aus, bis die aktuelle Unterprogramm-Ebene verlassen wird. Im Trace-Fenster werden alle ausgeführten Instruktionen angezeigt, die bis zum Verlassen des Unterprogramms ausgeführt wurden.

 *Step Over*

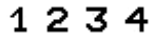
Führt einen Befehl der CPU aus. Falls in ein Unterprogramm gesprungen wird, wird dieses bis zum Ende des Unterprogrammes ausgeführt. Im Trace-Fenster werden alle ausgeführten Instruktionen protokolliert.

 *Bildschirm*

Zeigt oder verbirgt den Bildschirm, der durch die Baugruppe GDP64K angesteuert wird. Änderungen am Bildschirminhalt erfolgen auch, wenn das Fenster verborgen ist. Mit den daneben angezeigten Symbolen kann der Inhalt des Monitor-Fensters beeinflusst werden.

 *Automatik-Modus*

Das Monitor-Fenster verhält sich wie bei einem originalen Computer, der Inhalt des Monitor-Fensters wird durch die aktuell gesetzte Ausgabeseite bestimmt. Weitere Informationen hierzu finden sich im Abschnitt über die Baugruppe GDP64K.

 *Anzeigeseite*

Die Baugruppe GDP64K unterstützt die Verwaltung von vier unabhängigen Bildschirm-Seiten. Normalerweise erfolgt die Steuerung über einen speziellen Ausgabekanal der GDP, mit der die Schreibseite und die Leseseite getrennt festgelegt werden können. Mit den vier Symbolen kann die automatische Umschaltung deaktiviert werden, wodurch sich auch die Bildschirm-Seiten einsehen lassen, die normalerweise nicht sichtbar wären.

Trace Fenster

Zeigt oder verbirgt das Trace-Fenster zur Anzeige der Instruktionen im Einzelschrittmodus. Das Fenster wird automatisch sichtbar, sobald zum ersten Mal einer der Einzelschritt-Befehle ausgeführt wird.

HEX Editor Fenster

Zeigt oder verbirgt das Fenster zur Anzeige und Manipulation des Speicherinhaltes.

Memory-Map Fenster

Zeigt oder verbirgt die Anzeige der Speicherbelegung.

Einstellungen

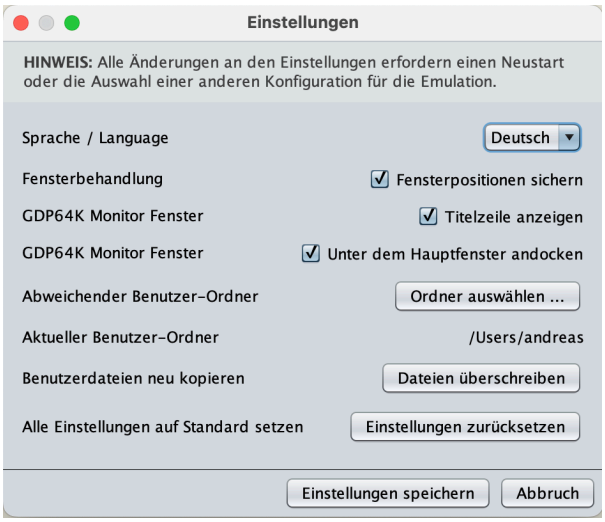
Auf der rechten Seite der Toolbar findet sich ein Button zum Aufruf des Dialoges mit den Einstellungsmöglichkeiten.

Der Einstellungsdialog

Zunächst ein wichtiger Hinweis: Bei jedem Speichern der Einstellungen wird der Emulator komplett neu gestartet. Jede nicht gespeicherte Arbeit geht dabei verloren.

Sprache / Language

Als Sprachen lassen sich deutsch und englisch auswählen. Die Sprachauswahl wirkt sich auf die Oberfläche des Emulators aus, aber nicht auf die emulierten Programme.



Fensterpositionen sichern

Mit „Fensterpositionen sichern“ werden beim Beenden des Emulators die Positionen der Fenster gespeichert und bei einem Neustart wiederhergestellt.

Die Position des Monitor-Fensters wird nicht gesichert, wenn es angedockt ist.

GDP64K Monitor Fenster – Titelzeile anzeigen

Die Option „Titelzeile anzeigen“ versieht das Monitor-Fenster mit einem Rahmen, der das unabhängige Verschieben des Fensters ermöglicht. Gleichzeitig wird hierdurch die Anpassung der Größe des Monitor-Fensters möglich.

GDP64K Monitor Fenster – Unter dem Hauptfenster andocken

Wenn „Unter dem Hauptfenster andocken“ gewählt ist, wird das Monitor-Fenster zusammen mit dem Hauptfenster verschoben, sobald die Position des Hauptfensters verändert wird.

Abweichender Benutzerordner

Standardmäßig werden die vom Benutzer veränderbaren Dateien (Konfigurationsdateien, ROMs, Disketten-Images und einige weitere) im Benutzerverzeichnis gespeichert. Die Software erstellt dort einen Ordner NKCEmu und kopiert die vorgegebenen Dateien dorthin. Mit dem Button „Ordner auswählen“ kann man einen abweichenden Speicherort für den Ordner NKCEmu festlegen (zum Beispiel auf einem Netzwerk-Laufwerk). Direkt darunter wird der aktuelle Benutzerordner angezeigt.

Benutzerdateien neu kopieren

Mit dem Button „Dateien überschreiben“ werden alle im Installationspaket mitgelieferten Dateien erneut in das Benutzerverzeichnis kopiert. Dabei gehen Änderungen, die an den Originalen gemacht wurden, verloren. Selbst erstellte Dateien oder Kopien der Originale mit abweichendem Dateinamen bleiben erhalten. Wiederhergestellt werden:

- ROM-Image-Dateien für BANKBOOT
- ROM-Image-Dateien für ROA64
- Konfigurationsdateien
- Diskettenimages für FLO2
- Verzeichnisse und Beispieldateien

Dateien, die im persönlichen Benutzerverzeichnis gelöscht wurden, egal ob absichtlich oder versehentlich, werden ebenfalls wieder in das Benutzerverzeichnis kopiert.

Einstellungen zurücksetzen

Mit diesem Button werden alle Einstellungen auf den Standard-Wert zurückgesetzt. Im Prinzip verhält sich der Emulator danach, wie nach der Installation.

- Alle Einstellungen in diesem Dialog werden zurückgesetzt
- Der Benutzerordner befindet sich wieder unterhalb des Installationsverzeichnisses
- Alle Fensterpositionen werden auf Standard zurückgesetzt
- Die zuletzt benutzten Ordner in Dateiauswahl-Dialogen werden zurückgesetzt

Die im Benutzerverzeichnis erstellten oder veränderten Dateien bleiben jedoch erhalten.

Das Menü

Die Funktionen des Emulators können auch über das Menü gesteuert werden. Neben den Menüeinträgen sind die Tastaturkürzel angegeben, mit denen sich die Emulation auch ohne Zuhilfenahme der Maus steuern lässt.

Edit

- Paste to NKC – Inhalt der Zwischenablage wird an die Baugruppe KEY gesendet.
- Take Screenshot – Der Inhalt des Monitors wird in die Zwischenablage kopiert.

Run

- Start Emulation – Startet die Emulation und beendet den Einzelschrittmodus.
- Stop Emulation – Hält die Emulation an und aktiviert den Einzelschrittmodus.
- Step in – Führt eine einzelne Instruktion der CPU aus.
- Step out – Führt Instruktionen aus, bis ein Unterprogramm beendet ist
- Step over – Führt eine einzelne Instruktion oder ein ganzes Unterprogramm aus.
- Reset – Setzt die CPU und die Hardware zurück.

View

- Monitor Window – Zeigt den Monitor an
- Automatic Mode – Bildschirminhalt durch die Emulation gesteuert
- Lock Page 1 – Bildschirm zeigt immer Page 0 der GDP64K an
- Lock Page 2 – Bildschirm zeigt immer Page 1 der GDP64K an
- Lock Page 3 – Bildschirm zeigt immer Page 2 der GDP64K an

- Lock Page 4 – Bildschirm zeigt immer Page 3 der GDP64K an
- Trace Window – Zeigt das Trace-Fenster an
- HexEdit Window – Zeigt das das Hex-Editor-Fenster an
- Memory Map Window – Zeigt die Speicherbelegung an

Tools

- Intel HEX Konverter – Binärdatei aus Intel-Hex-Format erzeigen
- Dateien zerlegen – Große Dateien in Blöcke zu z.B. 8 kByte zerlegen
- Dateien zusammensetzen – Mehrere Dateien zusammenführen
- Disketten Image bearbeiten – Löschen und Hinzufügen von Dateien in das Image

Help

- About – Zeigt Informationen zum Emulator an

Das Monitor-Fenster

Der Monitor wurde in Version 1.20 umfangreich überarbeitet. Das Fenster lässt sich jetzt frei auf dem Bildschirm positionieren und skalieren. Damit ist es möglich, eine realistische Darstellung des Seitenverhältnisses von 4:3 zu erhalten.

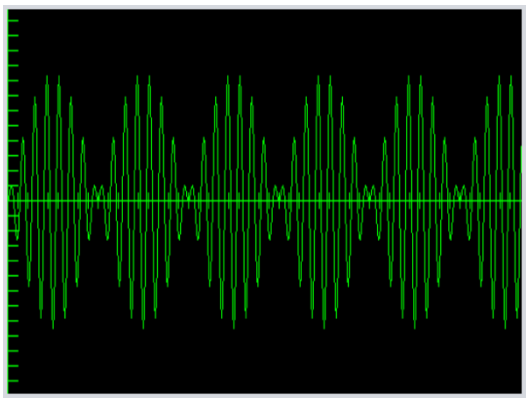
Jedes Pixel der Baugruppe GDP64K oder GDP64HS erzeugt genau 1 Pixel der Grafikausgabe des Host-Computers. Moderne Computer arbeiten jedoch mit verschiedenen Auflösungen je nach Monitor und Einstellung der Bildschirmauflösung. Dadurch kommt es zu einer verzerrten Darstellung des Inhaltes des Monitor-Fensters.

Eine weitere Neuerung betrifft die im Monitor-Fenster verwendeten Farben für den Vorder- und Hintergrund.

Farb-Einstellungen

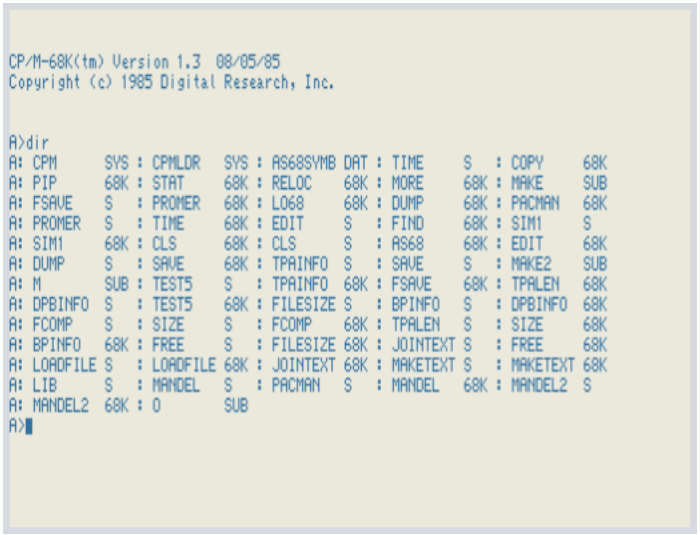
Mit einer zusätzlichen Zeile in den Konfigurations-Dateien lässt die die Farbe für den Vorder- und Hintergrund beliebig anpassen.

Die freie Festlegung der Farben gestattet das Look-and Feel von alten Röhrenmonitoren, sowohl in Grün oder Amber. Wer es mag, kann auch gerne gelbe Schrift auf blauem Hintergrund einstellen, so wie es zum Beispiel die Schneider CPC Computer taten.



Da die Einstellung in den verschiedenen Konfigurations-Dateien erfolgt, lässt sich für jede Konfiguration ein anderes Farbprofil setzen. Im Kapitel über die Konfigurations-Dateien ist näher beschrieben, wie die Einstellung vorgenommen werden kann. Die Konfiguration „NKC 68K DEMO“ wurde zur Demonstration wie im Bild gezeigt angepasst.

Position und Skalierung



Das neue Monitor-Fenster lässt sich frei auf dem Bildschirm positionieren.

Die letzte Position wird beim Beenden der Software oder beim Wechsel der Konfiguration gespeichert und beim Neustart wiederhergestellt, wenn diese Funktion in den Einstellungen aktiviert ist.

Falls das Monitor-Fenster unter dem Hauptfenster angedockt ist, wird die letzte Position ignoriert.

Bisher konnten für die Breite und Höhe des Fensters nur die ganzzahligen Vergrößerungsstufen von 1 bis 3 eingestellt werden. Mit der neuen Version lassen sich auch Werte dazwischen

angeben, damit eine realistische Darstellung des Bildes erreicht werden kann. Das Fenster kann jedoch nicht kleiner als 512x256 Pixel und nicht größer als 1536x768 eingestellt werden.

Emulierte Hardware

CPU – Central Processing Unit (68000 und 68008)

MC68008 Registers

```
D0 = 00030000  A0 = 0003CEA0   PC  = 000C0A20
D1 = 00030080  A1 = 0000000B
D2 = 00038000  A2 = 00040000   SSP = 0003E554
D3 = 00000400  A3 = 00000000   USP = 00000000
D4 = 00038000  A4 = FFFFFFFE
D5 = 00000000  A5 = 000C8000   SR  = 00002704
D6 = 0000F078  A6 = 000C0A00
D7 = 000005D0  A7 = 0003E558   FLG = xnZvc

[000C0A20] MOVE.B $FF68,D0
```

Die Emulation basiert auf der Umsetzung des Motorola MC 68000 Prozessors von Tony Headford mit einigen Erweiterungen, welche die Nutzung im NKC-Emulator erleichtern und verbessern. Die Änderungen beziehen sich im Wesentlichen auf das Berechnen des Timings der Prozessor-Varianten und auf die Protokollierung der Unterprogramm-Ebene.

Über die Konfigurationsdateien können verschiedene Prozessoren ausgewählt werden. Dies bestimmt sowohl den maximal nutzbaren Speicherbereich als auch die Ablaufgeschwindigkeit des Emulators.

Die Geschwindigkeit der Emulation entspricht etwa der realen Geschwindigkeit im NKC. Da sich die Emulationsgeschwindigkeit auf die Systemzeit des emulierenden Computers bezieht, kann die Geschwindigkeit der Emulation je nach verwendetem Computer leicht abweichen. In der Regel beträgt die Differenz jedoch unter 2%.

In der Konfigurationsdatei kann optional ein Frame aktiviert werden, der die Inhalte aller Register der CPU in Echtzeit anzeigt. Dabei werden die aktuell geänderten Registerinhalte hervorgehoben dargestellt sowie der gerade ausgeführte Befehl in disassemblierter Form angezeigt. Im Zusammenhang mit dem Trace-Fenster und den Breakpoints ist das sehr hilfreich bei der Entwicklung eigener Programme.

CPU 68000

- Maximaler Speicherbereich 2 MByte 0x00000000 – 0x001FFFFF
- Nominelle Taktfrequenz 16 MHz

CPU 68008

- Maximaler Speicherbereich 1 MByte 0x00000000 – 0x000FFFFF
- Nominelle Taktfrequenz 8 MHz

CPU 68010

- Maximaler Speicherbereich 2 MByte 0x00000000 – 0x001FFFFF
- Nominelle Taktfrequenz 20 MHz

Optional kann die emulierte Taktfrequenz der CPU abweichend von der Vorgabe eingestellt werden. Dazu wird das Schlüsselwort CLOCK in den Konfigurationsdateien verwendet.

Z80A Registers

```
A = 0x00  : A' = 0xFF   PC = 0x3EC4
B = 0x00  : B' = 0xFF   SP = 0x67F5
C = 0x00  : C' = 0xFF
D = 0x00  : D' = 0xFF   IX = 0x1302
E = 0x0A  : E' = 0xFF   IY = 0x624D
H = 0x13  : H' = 0xFF
L = 0x2E  : L' = 0xFF   Flags
F = 0x90  : F' = 0xFF   S z H pv n c

[3EC4] LD A,($60CF)
```

Die Emulation des Z80 Mikroprozessors wurde so weit wie möglich kompatibel ausgeführt, so dass jederzeit eine Umschaltung zwischen den Systemen möglich ist.

Über die Konfigurationsdateien kann die Version des Prozessors ausgewählt werden. Im Gegensatz zum 68000 wird hierdurch nur die Taktfrequenz beeinflusst.

CPU Z80A

- Maximaler Speicherbereich 64 kByte 0x0000 – 0xFFFF
- Mit BANKBOOT 1 MByte 0x00000 – 0xFFFFF
- Nominelle Taktfrequenz 4 MHz

CPU Z80B

- Maximaler Speicherbereich 64 kByte 0x0000 – 0xFFFF
- Mit BANKBOOT 1 MByte 0x00000 – 0xFFFFF
- Nominelle Taktfrequenz 6 MHz

CPU Z80H

- Maximaler Speicherbereich 64 kByte 0x0000 – 0xFFFF
- Mit BANKBOOT 1 MByte 0x00000 – 0xFFFFF
- Nominelle Taktfrequenz 10 MHz

MEMORY – Hauptspeicher

Bei der Emulation des Hauptspeichers wird nicht zwischen den verschiedenen Baugruppen wie ROA64, RAM64/256 usw. unterschieden. Stattdessen wird der gesamte Speicherbereich wie ein zusammenhängender Bereich mit unterschiedlichen Eigenschaften betrachtet.

Die Größe des verwalteten Hauptspeichers richtet sich nach der CPU und umfasst den gesamten möglichen Adressbereich des gewählten Mikroprozessors. Der Speicher kann durch die Konfigurationsdatei frei definiert werden. Dadurch ist es möglich, die Bestückung aller Speicherbaugruppen festzulegen.

Über die Konfigurationsdateien ist es möglich, die Größe des maximal adressierbaren Speichers für die Emulation anzupassen. Falls diese Möglichkeit nicht genutzt wird, werden die Vorgaben des gewählten Mikroprozessors genutzt.

BANKBOOT – RAM-Speicher ab Adresse 0

Die Emulation verwaltet neben dem Hauptspeicher 32 kByte Speicher der BANKBOOT-Karte, die nach dem Start des Systems dazu dient, RAM-Speicher ab Adresse 0 einzubinden. Auch hier kann die Bestückung mit ROM oder RAM über die Konfigurationsdateien frei festgelegt werden. Die Baugruppe BANKBOOT hat keinen eigenen Anzeigebereich im Hauptfenster.

Ab Version 1.20 des Emulators können 16 unterschiedliche RAM-Bänke in den 64 kByte umfassenden Speicherbereich des Z80 Prozessors eingeblendet werden. Das Grundprogramm 2018 bietet dazu entsprechende Funktionen.

GDP64K – Grafik-Interface und Monitor

```
GDP64K @ 0x70
STATUS = 00  PEN/ERASER = UP
CTRL1  = 03  PEN MODE   = PEN
CTRL2  = 00  WRITE MODE = NORMAL
CSIZE  = 21  WRITE AREA = 4096X4096
DELTAX = 80  LINE MODE  = CONTINUOUS
DELTAY = 00
X MSB  = 01  XPOS = 436
X LSB  = B4  YPOS = 030
Y MSB  = 00
Y LSB  = 1E  WRITEPAGE 0  READPAGE 0
```

Der Treiber emuliert das Verhalten des Grafik-Prozessors EF9366 auf der Baugruppe GDP64K des NKC. Ein 64 kByte umfassender Bildschirmspeicher wird mit entsprechenden Befehlen an den Treiber manipuliert.

Ein eigenes Fenster stellt den Inhalt dieses Bildschirm-Speichers dar. Der Inhalt des Fensters wird durch einen getrennten Thread 50-mal pro Sekunde aktualisiert und läuft unabhängig von der Emulation des restlichen Systems. Um die Proportionen der Ausgabe einigermaßen korrekt darzustellen, wird jedes Pixel als 3x2 Block dargestellt.

In der Konfigurationsdatei können optional andere Skalierungen angegeben werden, die zu einer mehr oder weniger verzerrten Ausgabe führen. Ebenfalls optional kann ein Frame aktiviert werden, der in Echtzeit die Inhalte der Register des Grafikprozessors im Hauptfenster darstellt.

GDP64HS – Grafik-Interface und Monitor

```
GDP64HS @ 0x70
STATUS = 00  PEN/ERASER = UP
CTRL1  = 03  PEN MODE   = PEN
CTRL2  = 00  WRITE MODE = NORMAL
CSIZE  = 21  WRITE AREA = 4096X4096
DELTAX = 80  LINE MODE  = CONTINUOUS
DELTAY = 00
X MSB  = 01  XPOS = 436  RMW = OFF
X LSB  = B4  YPOS = 030  SCROLL = 000
Y MSB  = 00
Y LSB  = 1E  WRITEPAGE 0  READPAGE 0
```

Im NDR Klein Computer kann anstelle der GDP64K die Baugruppe GDP64HS eingesetzt werden. GDP64HS unterstützt zusätzliche Funktionen, die auf mit GDP64K nicht möglich sind.

- Read-Modify-Write-Modus (RMW)
- Auslesen des Bildschirmspeichers
- Hardware-Scrolling

Der Status des RMW-Modus und der Scroll-Position wird zusätzlich zu den Standardmäßigen Ausgaben der GDP64K angezeigt.

RMW-Modus (Read-Modify-Write)

Vor der Ausgabe eines Pixels auf dem Monitor wird getestet, ob dieses bereits gesetzt ist.

Neues Pixel	**altes Pixel**	**Ergebnis**
schwarz	schwarz	schwarz
schwarz	weiß	schwarz
weiß	schwarz	weiß
weiß	weiß	schwarz (invertiert)

Auslesen der Bildschirmspeichers

Nachdem eine Pixelposition an den Grafikprozessor EF9366 übertragen wurde, kann der Inhalt des Bildspeichers, welches dieses Pixel enthält über den Port 60 (Seitenregister) zurückgelesen werden.

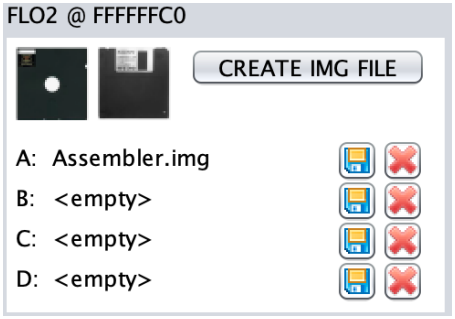
Hardware-Scrolling

Über den bei der GDP64HS zusätzlich verfügbaren Port 61 kann eine Verschiebung der Anzeige in vertikaler Richtung angegeben werden. Die Verschiebung wirkt sich ausschließlich bei der Darstellung auf dem Monitor aus. Die Schreibpositionen bleiben unverändert.

Dadurch ist ein sanftes Scrolling des Bildschirminhaltes möglich, da nicht immer der komplette Inhalt des Bildschirms neu aufgebaut werden muss. Bei einer Verschiebung um 8 Pixel nach oben reicht es aus, die letzte Zeile des Bildschirms neu zu beschreiben.

HINWEIS: in der aktuellen Version 1.20 ist diese Funktion noch nicht verfügbar.

FLO2 – Floppy-Disk-Interface



Mit dem Treiber FLO2 können insgesamt vier Diskettenlaufwerke emuliert werden. Welche Formate erkannt werden hängt dabei natürlich von der Implementierung innerhalb der betriebenen Software ab. Um das Arbeiten mit dem Emulator einfacher zu gestalten, können leere Diskettenimages im Standard-Format des NKC direkt über die Oberfläche erstellt werden.

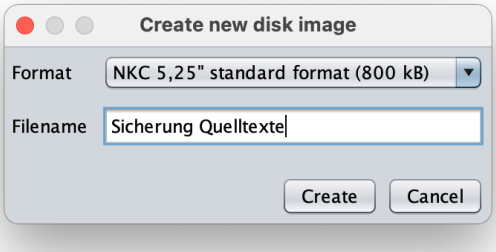
Das Standard-Format des NKC bietet eine Bruttokapazität von 800 kByte entsprechend der Speicherkapazität einer 5,25 Zoll Diskette im Double Density Format. Eine Formatierung wie bei physischen Disketten ist in der Emulation nicht notwendig. Der Treiber arbeitet unmittelbar auf den Dateien des emulierenden Computers. Nach der Arbeit muss also nicht manuell gespeichert werden.

In der beim NKC vorgesehenen Konfiguration sind die Laufwerke unter CP/M wie folgt belegt

- A: NKC 5,25“ Standard Format (800 kB)
- B: NKC 5,25“ Standard Format (800 kB)
- C: IBM 3740 8“ Standard Format (250 kB)
- D: IBM 3740 8“ Standard Format (250 kB)

Die mitgelieferten Disketten-Images liegen ausschließlich im Standard-Format vor. Im Anhang findet sich eine Liste der Inhalte dieser Images.

Create IMG File



Erzeugt ein neues leeres Disketten-Image, welches danach sofort in eines der Laufwerke geladen werden kann.

Es werden einige Formate zur Auswahl gestellt, die nur dann Sinn ergeben, wenn die Software diese Formate nutzen kann. Unter CP/M müsste dazu das BIOS entsprechend angepasst werden.

Disketten Image laden

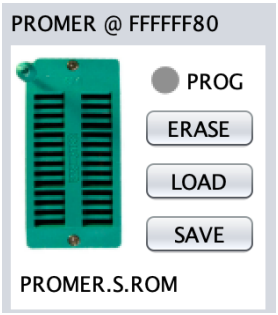
Mit den Diskettensymbol-Buttons neben den Laufwerksbuchstaben kann ein Image in eines der Laufwerke geladen werden. Nach der Auswahl wird der Dateiname der Image-Datei neben dem Laufwerksbuchstaben angezeigt. Eine Neuauswahl ist jederzeit auch während des Betriebs möglich, ohne dass ein Datenverlust auftritt.

Image-Dateien können über das Betriebssystem schreibgeschützt werden. Sie lassen sich ganz normal laden und benutzen. Jeder Schreibversuch führt zu einer Fehlermeldung im Emulator, die Image-Datei wird nicht verändert.

Disketten Image auswerfen

Mit dem zweiten Button neben dem Dateinamen der Imagedatei kann dieses Diskettenimage ausgeworfen werden. Auch hier sind keine Datenverluste zu erwarten, da der Treiber direkt auf den Image-Dateien arbeitet.

PROMER – EPROM-Programmier-Interface



Der Treiber für die Baugruppe PROMER ist auf die übliche Konfiguration für 8 kByte EPROMS des Typs 2764 eingestellt. EPROMS können gelesen, programmiert und gelöscht werden. Die Bedienung der Baugruppe erfolgt über den Frame im Hauptfenster.

Die Programmierung erfolgt entweder über das Grundprogramm oder über das Programm PROMER.68K direkt aus CP/M. Dieses Programm befindet inklusive Quellcode auf der NKC-Zusatzdiskette. Wie im Original leuchtet während des Programmierens eine LED auf.

LOAD – Einsetzen bestehender EPROMs

Lädt den Inhalt des Puffers mit einer ROM-Datei aus dem Unterverzeichnis PROMER. Nach dem Laden wird der Dateiname der geladenen Datei unten im Frame angezeigt. Der Inhalt des Puffers kann jetzt mit dem Grundprogramms in den Hauptspeicher übertragen werden.

SAVE – Speichern programmierter EPROMs

Speichert den Inhalt des Puffers in das Unterverzeichnis PROMER. Nach dem Speichern wird der Dateiname der gesicherten Datei unten im Frame angezeigt. Dateien, die hiermit erzeugt wurden, können in die Unterverzeichnisse ROMS und BOOT kopiert werden, um diese bei der Konfiguration des Hauptspeichers bzw. des Speichers auf der Baugruppe BANKBOOT zu verwenden.

ERASE – Löschen von EPROMs

Löscht den 8 kByte umfassenden Puffer, alle Bytes enthalten den Wert 0xFF. Anstelle des Dateinamens wird der Text „EPROM is empty“ angezeigt. Im Gegensatz zu einem realen EPROM lassen sich auch gesetzte Bits in den Puffer schreiben, man muss also nicht den Puffer löschen, um erneut programmieren zu können.

CAS – Kassetten-Interface

Der Treiber für die Baugruppe CAS emuliert das Kassetteninterface des NKC und verwaltet einen Daten-Puffer unbestimmter Größe. In diesem Puffer werden alle an den Baustein 6850 der Baugruppe CAS übertragenen Bytes gesichert beziehungsweise von dort geladen.

Der Frame zeigt neben den vier Buttons zur Bedienung ein Zählwerk (Byte-Zähler) und den Dateinamen der aktuell geladenen oder gespeicherten CAS-Datei aus dem Unterverzeichnis CAS des Emulators an.

Die Bedienung der Baugruppe erfolgt über die Menüpunkte **Sichern CAS**, **Laden CAS** und **Pruefen CAS** des Grundprogramms. Beim Speichern wird das Timing des Grundprogramms nachgebildet, das Laden und Prüfen erfolgt hingegen sehr schnell, auf die Umsetzung einer Verzögerung wurde absichtlich verzichtet.

LOAD – Einlegen von Kassetten

Lädt eine zuvor mit CAS gesicherte Datei in den Puffer. Das Zählwerk wird automatisch auf 0 zurückgesetzt. Das Grundprogramm wartet nach Aufruf des Menüpunkts „Laden CAS“ so lange bis eine CAS-Datei geladen wurde. Da das Grundprogramm beim Laden keinen Dateinamen abfragt, kann man effektiv nur das erste gesicherte Programm einer CAS-Datei verwenden.

SAVE – Sichern des Pufferinhaltes

Sichert den Pufferinhalt in eine CAS-Datei im Unterverzeichnis CAS des Emulators zur späteren Verwendung. Nach dem Speichern zeigt das Zählwerk die Anzahl der in den Puffer geschriebenen Bytes an.

REWIND – Zurückspulen von Kassetten

Das Zählwerk wird auf 0 zurückgesetzt, die geladenen oder gespeicherten Daten im Puffer bleiben erhalten.

EJECT – Auswerfen von Kassetten

Löscht den Puffer. Es ist zu beachten, dass möglicherweise geschriebene Daten zuvor gesichert werden müssen.

CENT – Centronics-Drucker-Interface

Der Treiber für die Baugruppe CENT emuliert das Druckerinterface des NKC und erlaubt das Ausdrucken sowie das Speichern aller an die Baugruppe gesendeten Daten. Der Treiber enthält einen Puffer, in dem alle gesendeten Daten gesammelt werden. In der unteren linken Ecke wird der Füllstand des Puffers angezeigt.

Der Pufferinhalt kann wahlweise als Textdatei gespeichert oder direkt zu einem physikalisch vorhandenen Drucker gesendet werden. Nach dem Speichern oder Ausdrucken kann der Puffer manuell gelöscht werden. Als Textdatei gesicherte Druckerausgaben werden im Unterverzeichnis CENT des Emulators abgelegt. Diese Funktion ist hilfreich bei der Erstellung von Dokumentationen eigener Programme.

Unter CP/M kann mit der Tastenkombination Ctrl+P erreicht werden, dass alle Ausgaben auf dem Bildschirm gleichzeitig zum Drucker gesendet werden. Die Tastenkombination wechselt ein Flag im Treiber und zeigt abhängig von Status einen roten Punkt neben dem Drucker-Symbol an.

Da der Emulator keine Möglichkeit hat, festzustellen wann CP/M gestartet wird oder ob das Betriebssystem aktuell ausgeführt wird, sollte man darauf achten, dass das Flag nicht aktiv ist, bevor man CP/M startet.

SER – Serielle Schnittstelle

Die Baugruppe SER bietet eine serielle Schnittstelle für die Kommunikation mit dem Hostcomputer oder mit einem realen NDR Klein Computer.

Als Ziel der Ausgabe können die Optionen

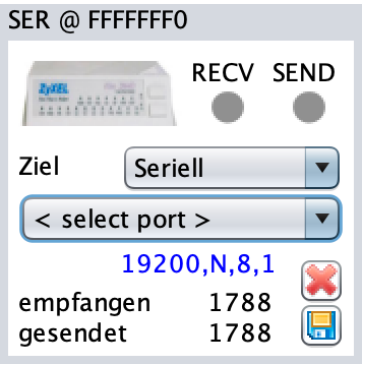
- Puffer
- Seriell

eingestellt werden.

Puffer-Modus

Im Puffer-Modus werden die vom NKC an SER gesendeten Daten in einem Sendepuffer gesammelt. Bei aktiviertem ECHO-Modus der Baugruppe SER werden gesendete Daten zusätzlich in den Empfangspuffer kopiert und können innerhalb der Emulation wieder empfangen werden. Die beiden Buttons unten rechts erlauben das Löschen der Puffer und das Sichern des Sende-Puffers in das Verzeichnis SER der Benutzerdaten.

Seriell-Modus



Wenn als Ziel der Ausgabe Seriell angegeben wird, erschein eine neue Auswahlmöglichkeit mit physikalisch vorhandenen seriellen Schnittstellen des Host-Computers. Bei modernen Computern benötigt man dazu einen USB nach SERIELL-Wandler.

Die Baudrate und das Datenformat lassen sich im Controller nicht festlegen, sie werden durch die in der Emulation ablaufende Software bestimmt. Im Beispiel links 19200 Baud, No Parity, 8 Datenbits, 1 Stopp-Bit.

Falls wie oben gezeigt kein serieller Port ausgewählt wurde, kann keine Übertragung von Daten aus der Emulation erfolgen. Software, die dies nicht abfängt, hängt sich auf.

Die beiden Zähler zeigen an, dass 1788 Bytes vom NKC gesendet wurden. Durch den aktiven ECHO-Modus wurden die gesendeten Bytes gleichzeitig im Empfangs-Puffer abgelegt und könnten durch eine Software in der Emulation wieder eingelesen werden.

UHR – Real Time Clock

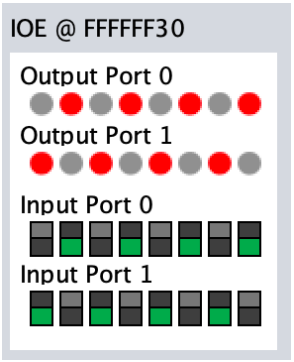
Die Baugruppe UHR stellt eine batteriegestützte Echtzeituhr im NKC zur Verfügung.

In der Emulation ist nur der lesende Zugriff auf die Uhrzeit umgesetzt, es wird immer die Zeit des Hostcomputers zurückgegeben.

Das Stellen der Uhrzeit durch den NKC hat keinen Effekt, führt aber auch nicht zu einem Fehler.

Es gibt unterschiedliche Hardware für den NKC, um eine Echtzeituhr zu realisieren. Die aktuelle Implementation funktioniert nicht mit den Grundprogrammen 6.22 und 7.10

IOE – Eingabe und Ausgabe



Der Treiber für die Baugruppe IOE emuliert die Eingabe und Ausgabe zur Steuerung externer Geräte. Die Baugruppe stellt zwei Ausgabe-Ports und zwei Eingabe-Ports zur Verfügung.

In den Grundprogrammen können die Ports mit den Menüpunkten **I/O setzen** und **I/O lesen** angesteuert werden.

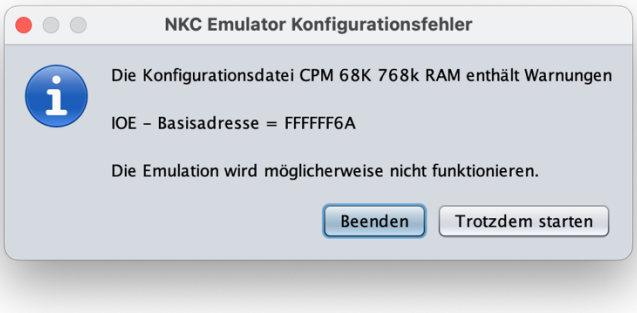
Die Basisadresse der Baugruppe lässt sich über die Konfiguration einstellen. Aktuell kann nur eine einzige IOE-Baugruppe innerhalb einer Konfiguration verwendet werden.

Mögliche Adresskonflikte

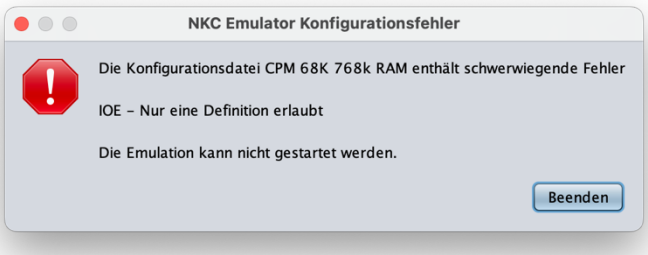
Wenn in den Konfigurationsdateien abweichende Basisadressen für die Baugruppen definiert werden, kann dies andere Baugruppen in der Funktion beeinträchtigen oder zu einer nicht lauffähigen Emulation führen.

Bevor eine Konfiguration gestartet wird, wird die Konfigurationsdatei analysiert. Dabei wird zwischen Warnungen und Fehlern unterschieden.

Bei Warnungen, die zum Beispiel durch abweichende Basisadressen der Baugruppen ausgelöst werden, erscheint ein Dialog, der auf diesen Umstand hinweist. Die Emulation kann trotzdem gestartet werden.



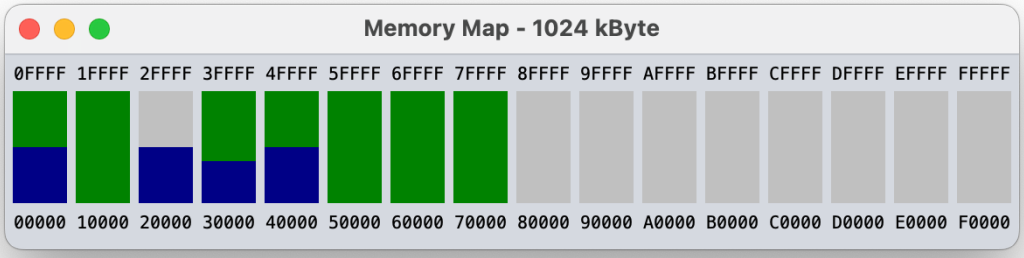
Bei Konfigurationsfehlern, zum Beispiel bei mehrfacher Definition gleicher Baugruppen, wird eine entsprechende Meldung angezeigt und die Emulation kann nicht gestartet werden. In diesem Fall muss die Konfiguration korrigiert werden.



Weitere Funktionen

Memory-Map Fenster

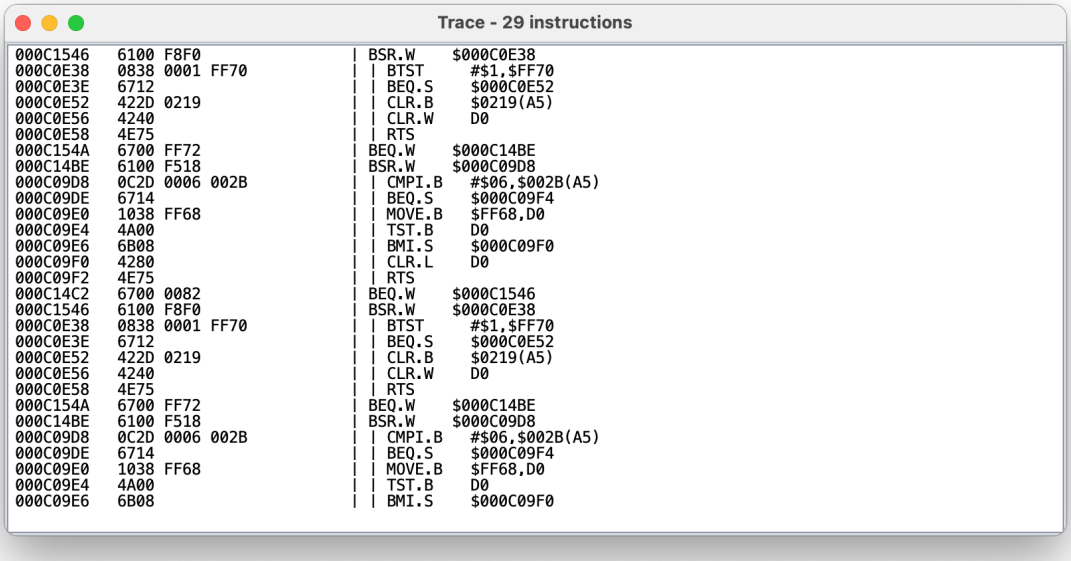
Im diesem Fenster wird die Belegung des Hauptspeichers dargestellt. So kann man jederzeit schnell überprüfen, an welchen Adressen ROMs geladen und an welchen Adressen RAM definiert wurde. Grundsätzlich wird nur das erste Megabyte des Speichers angezeigt. Bereiche mit RAM werden grün, ROM-Bereiche werden blau dargestellt. Unbelegte Bereiche erscheinen in grauer Farbe.



Das Fenster kann über das Menü View des Hauptfensters oder mit der Tastenkombination ALT+4 beziehungsweise Command+4 unter OSX aufgerufen werden.

Trace Fenster

Das Trace-Fenster dient zur Anzeige der ausgeführten Befehle des Mikroprozessors. Sinnvoll ist diese Anzeige jeder einzelnen Instruktion nur im Einzelschrittmodus, damit man den Ablauf eines Programmes exakt nachvollziehen kann. Im normalen Betrieb werden die Instruktionen auch bei geöffnetem Trace Fenster nicht wiedergegeben. Die Ausgabe der disassemblierten Befehle verbraucht sehr viel Zeit, so dass die Geschwindigkeit der Emulation deutlich geringer wäre.



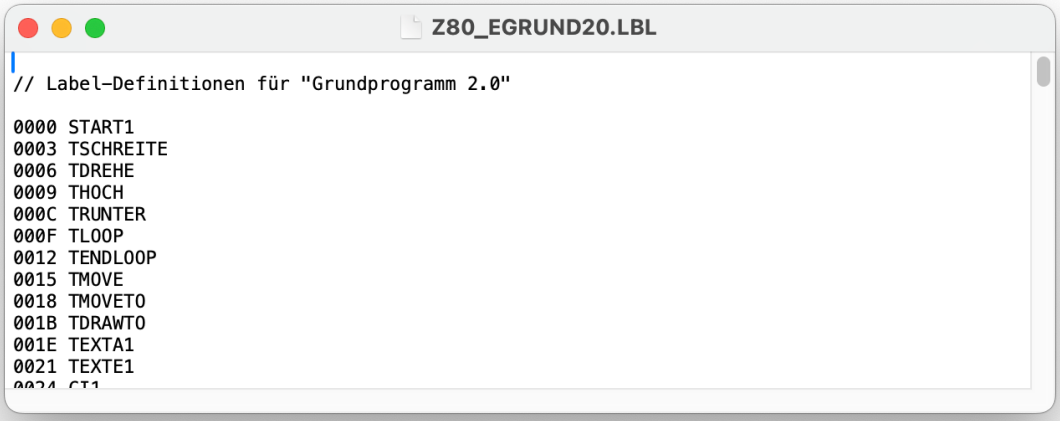
Jeder von der CPU ausgeführte Befehl wird in einer Zeile dargestellt. Die Zeilen bestehen aus

- Aktueller Programmzählerstand
- Bytes des ausgeführten Befehls
- Disassemblierte Instruktion

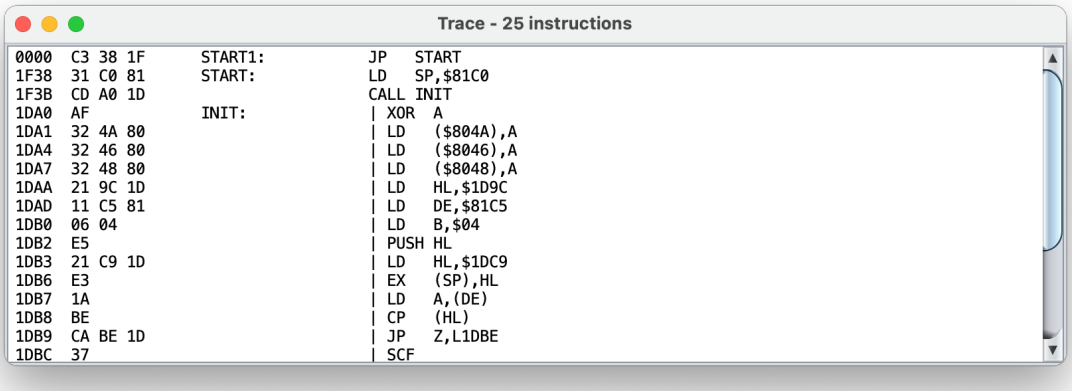
Die Instruktionen sind entsprechend der Verschachtelung der Unterprogramme eingerückt. Über das Menü kann der gesamte Text ausgewählt und in die Zwischenablage kopiert werden. Im Fenster ist zudem die Auswahl von Teilbereichen des Listings möglich.

Anzeige von Labels

Seit der Version 1.2 wird die Ausgabe von Labels unterstützt. Bei allen geladenen ROM-Images wird geprüft, ob eine Datei mit dem gleichen Dateinamen und der Endung .LBL vorhanden ist. Falls ja, lädt der Emulator die dort enthaltene Tabelle mit den Adressen und klarschriftlichen Namen der Labels.



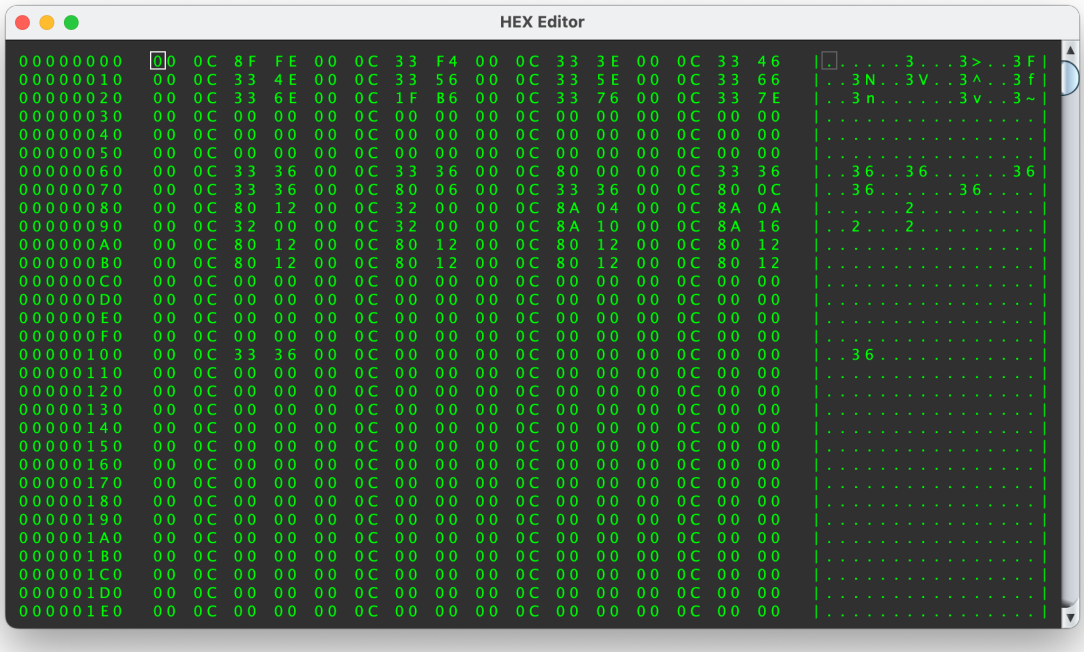
Die angegebenen Adressen müssen der Adresse entsprechen, in der das ROM tatsächlich geladen wird. Labels dürfen eine Länge von bis zu 16 Zeichen haben. Hier die ersten paar Zeilen bei der Ausführung des Grundprogramms 2.0 für den Z80 Prozessor.



Für den 68000 Prozessor funktioniert dies auch, es sind aber aktuell keine LBL-Dateien im Auslieferungsumfang.

Hex-Editor Fenster

Der Hex-Editor bietet einen direkten Zugriff auf den gesamten Hauptspeicher der Emulation. Die Anzeige wird bei sichtbarem Fenster immer aktuell gehalten und bietet so wertvolle Informationen beim Debuggen von Programmen.



Zusätzlich kann der gesamte Speicher unmittelbar überschrieben werden. Dabei sollte man Vorsicht walten lassen, da Änderungen im aktuell ausgeführten Programmbereich fast immer zu ungültigen Befehlssequenzen führen.

Der rechteckige Cursor kann frei im HEX-Bereich oder im ASCII-Bereich positioniert werden. Eingaben werden unmittelbar in den Hauptspeicher der Emulation geschrieben.

Integrierte Tools

Mit Erscheinen dieser neuen Version gibt es eine neuen Menüpunkt Tools, unter dem diverse kleine Hilfsprogramme erreichbar sind. Mehrere dieser Tools können gleichzeitig geöffnet und genutztt werden, während die Emulation läuft. Alle Tools merken sich die zuletzt benutzten Pfade.

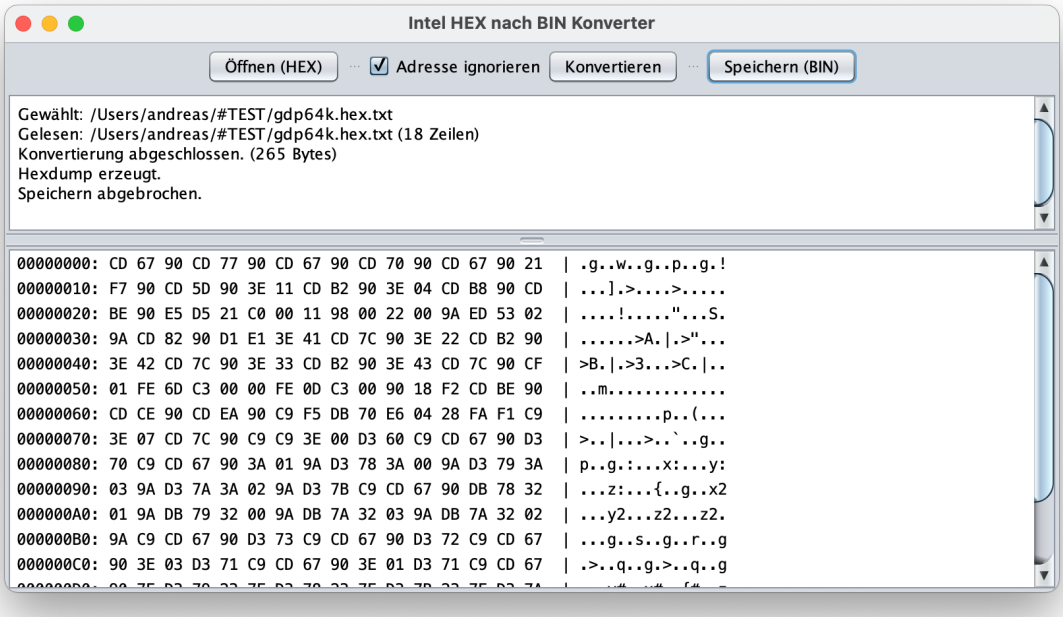
Intel HEX nach BIN-Konverter

Das Programm dient zum Konvertieren von Dateien im Intel HEX-Format in reine Binärdateien. Das Intel Hex-Format ist eine ASCII-Textdatei mit Textzeilen, die jeweils einen bestimmten Speicherbereich im Hexadezimalformat repräsentieren.

Normalerweise wurde dieses Format genutzt, um Maschinenprogramme zum Beispiel über ein Modem zu übertragen. Die einzelnen Zeilen können auch absolute Adressen des so codierten Programms enthalten.

```
:1C015000E61028FA79D3ECC93E53D3CA3E50D3CAC93A2A13CB67CA0801DBCAE651
:1C016C000128FADBCBC93A2A13CB67CA2301DBCAE60228FA79D3CBC90E07CD1CC1
:1C01880001C90600E5D17EE6DFFEDDCA00027EFECBCAFA01FEEDCAEE017EFEC3FC
:1C01A400CA3702FECDCA3702E6EFFE22CA3702FE2ACA3702E6CFFE01CA3702E6AE
:1C01C000C7FEC2CA3702FEC4CA37027EE6F7FE10CA3802FED3CA3802E6E7FE20A7
```

Nach dem Öffnen einer korrekt formatierten HEX-Datei wird die sofort die Wandlung in das Binär-Format ausgeführt, die Vorschau zeigt die binäre Repräsentation der Intel HEX-Datei.



Mit der Checkbox „Adresse ignorieren“ kann man die eventuell in der HEX-Datei angegebenen Adressen ignorieren du die Ausgabe immer auf die Adresse 0 festsetzen. Im oberen Bereich wir die Größe der erzeugten Ausgabe angegeben. Die Checkbox kann jederzeit ohne erneutes Laden der Quelle umgeschaltet werden.

Beim Speichern wird der Ordner der Quelldatei vorgegeben. Der Dateiname erhält die Erweiterung .BIN. Beides kann im Sichern-Dialog abgeändert werden.

ACHTUNG:

Es gibt keine Überprüfung, ob die generierte Datei schon vorhanden ist. Bereits bestehende Dateien mit dem gleichen Namen werden ohne Nachfrage überschrieben.

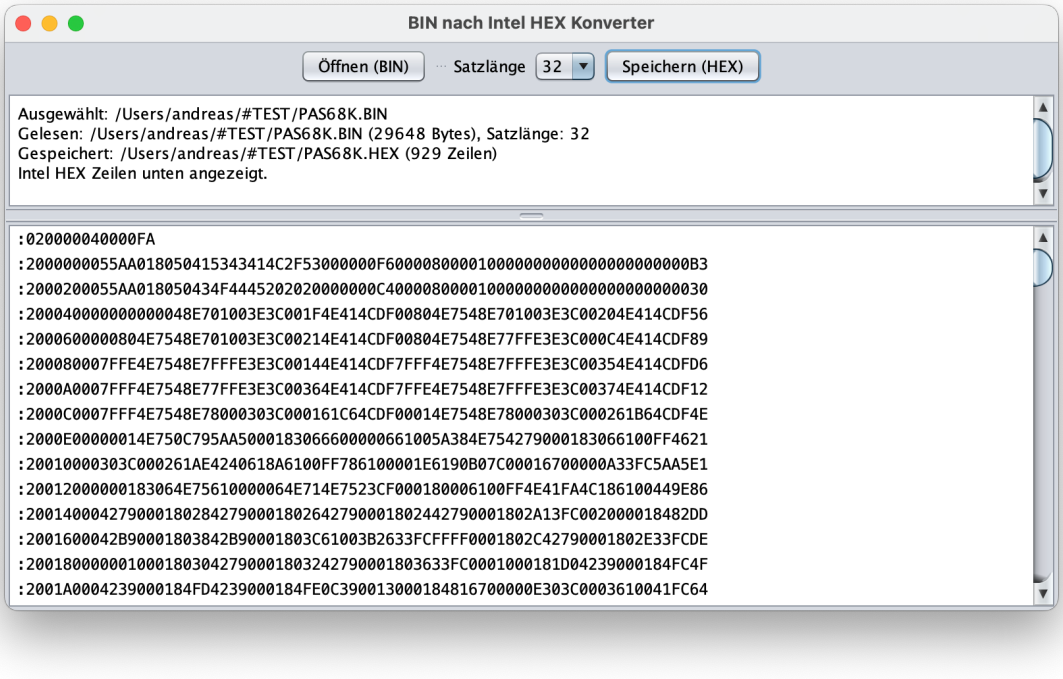
BIN nach Intel HEX Konverter

Dieser Konverter bildet das Gegenstück zum Intel HEX nach BIN-Konverter. Beliebige Dateien lassen sich in das Intel HEX Format wandeln. Nach dem Öffnen der Datei wird diese automatisch mit der eingestellten Satzlänge konvertiert und im Vorschaubereich als Hex-Dump angezeigt. Die Satzlänge kann jederzeit vor dem Speichern geändert werden, die Vorschau wird unmittelbar aktualisiert.

Als Vorgabe für den Speicherort wird der Ordner der zuvor geladenen Binärdatei vorgegeben. Die Datei-Endung der Zieldatei wird um die Datei-Endung .HEX erweitert.

ACHTUNG:

Es gibt keine Überprüfung, ob die generierte Datei schon vorhanden ist. Bereits bestehende Dateien mit dem gleichen Namen werden ohne Nachfrage überschrieben.



Binärdateien zerlegen

Beim Arbeiten mit dem realen NKC kommt es häufig vor, dass man EPROMs zum Einsatz auf ROA64 brennen möchte. Software, die größer als die zu verwendenden EPROMs sind, muss dazu in Blöcke der korrekten Größe zerlegt werden.

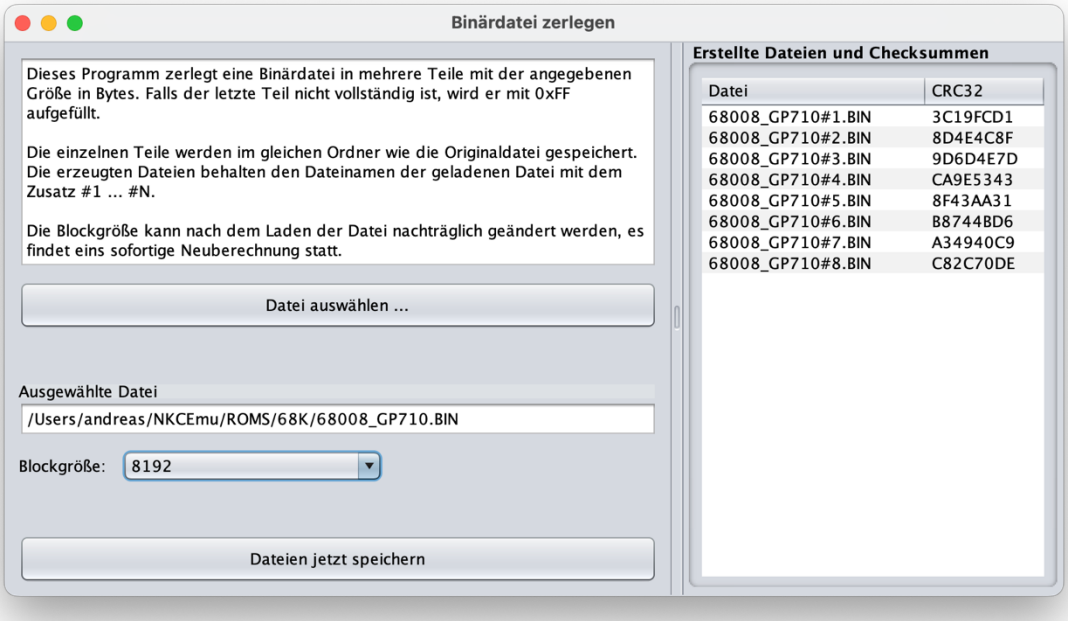
Nach der Auswahl der zu zerlegenden Datei über den Button „Datei auswählen“ wird die Datei unmittelbar in gleich große Teile in der angegebenen Blockgröße zerlegt. Auf der rechten Seite des Fensters wird eine Vorschau aller Teile mit der CRC32-Prüfsumme angezeigt.

Die Blockgröße kann in den typischen Größen von EPROMs gewählt werden und kann noch geändert werden, wenn die zu zerlegende Datei bereits geladen ist. Die Vorschau wird sofort aktualisiert.

- 1024 Byte EPROM 2708
- 2048 Byte EPROM 2716 Verwendung auf SBC2
- 4096 Byte EPROM 2732
- 8192 Byte EPROM 2764 Verwendung auf ROA64 Standard
- 16384 Byte EPROM 27128
- 32768 Byte EPROM 27256

Mit dem Button „Dateien jetzt speichern“ werden die in der Vorschau angezeigten Dateien im gleichen Verzeichnis wie die geladene Datei abgelegt. Die Namen der neuen Dateien werden aus dem Dateinamen der Originaldatei generiert und mit einer fortlaufenden Nummer ergänzt.

Für den Fall, dass die Länge der Quelldatei kein Vielfaches der Blockgröße ist, wird der letzte Teil der erzeugten Dateien bis zur Blockgröße mit 0xFF aufgefüllt.

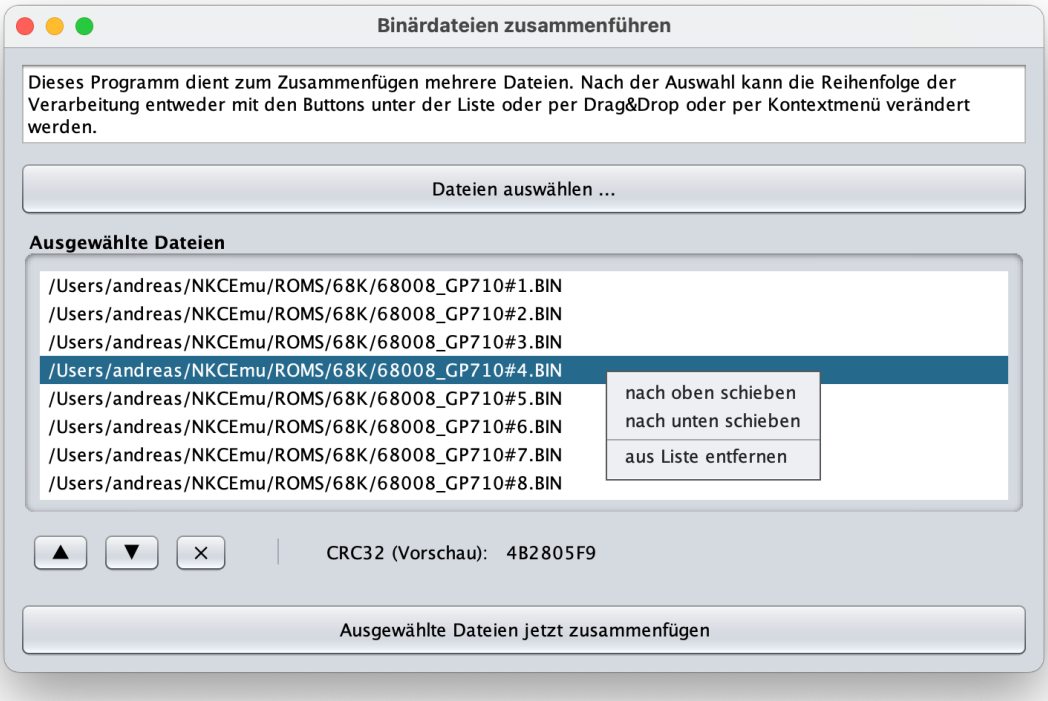


ACHTUNG: Eventuell bestehende Dateien mit gleichem Dateinamen werden ohne Nachfrage überschrieben.

Binärdateien zusammenführen

Dieses Tool bildet das Gegenstück zu „Dateien zerlegen“, es können also mehrere Dateien zu einer einzigen Datei zusammengeführt werden. So ist es zum Beispiel möglich, kleine ROMs zur Verwendung auf ROA256/1M umzuwandeln.

Die zusammenzuführenden Dateien müssen alphabetisch sortiert in einem gemeinsamen Ordner vorliegen. Im Öffnen-Dialog können mehrere Dateien markiert werden, die nach Abschluss der Auswahl in einer Liste dargestellt werden.



Der Dateiname der erzeugten Datei mit den zusammengeführten Daten kann frei festgelegt werden, Vorgabe ist „output.bin“.

Nach dem Klick auf „Ausgewählte Dateien zusammenführen“ wird die neue Datei erstellt und im gleichen Ordner wie die Quelldateien gespeichert. Auf der rechten Seite wird der Name der erzeugten Datei und die CRC32 Prüfsumme angezeigt.

ACHTUNG

Es gibt keine Überprüfung, ob die generierte Datei schon vorhanden ist. Eine vorhandene Datei mit dem gleichen Namen wird ohne Nachfrage überschrieben.

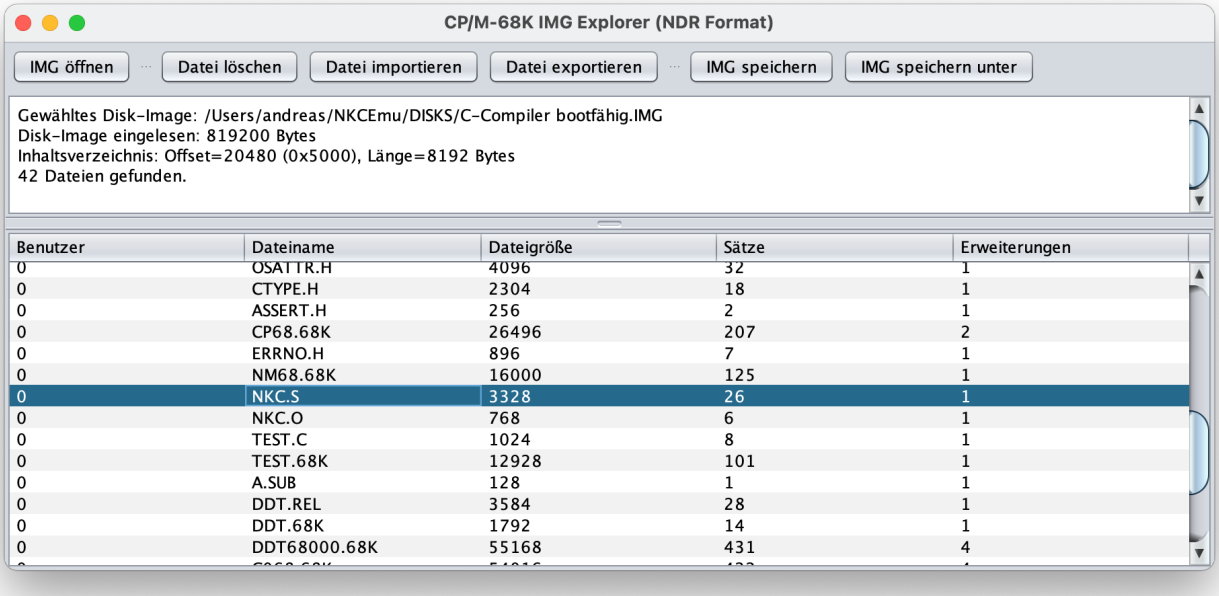
Disketten-Image bearbeiten

In der Emulation werden Diskettenlaufwerke über Image-Dateien realisiert, die ein genaues Abbild aller Sektor-Inhalte einer Diskette in einer Datei zusammenfassen. Das Format der Image-Dateien ist im Anhang beschrieben.

Es können zwar neue Disketten-Images mit dem Emulator erzeugt und mit den Bordmitteln der emulierten Software beschrieben werden, neue Dateien vom Host-Computer zu übertragen ist ohne Hardware und darauf zugeschnittene Software nicht so einfach möglich.

Dieses Tool bietet die Möglichkeit, Disketten-Images des Emulators zu öffnen und die Inhalte zu manipulieren. Im derzeitigen Zustand des Tools gibt es folgende Einschränkungen:

- Beschränkt auf das NDR-Format 800 kByte
- Beschränkt auf das Dateisystem des CP/M 68K
- Bereits vorhandene Dateien können nicht überschrieben werden



IMG öffnen

Öffnet einen Dateiauswahl-Dialog, mit dem zum Beispiel eine Datei aus dem Verzeichnis NKCEmu/DISKS ausgewählt werden kann. Nach der Analyse des Images werden die auf der Diskette enthaltenen Dateien in einer Liste angezeigt.

Datei löschen

Wird in der Liste eine der aufgeführten Dateien markiert, kann diese aus dem Dateisystem des CP/M 68K gelöscht werden. Es erfolgt keine Nachfrage, die Datei wird unmittelbar aus dem im internen RAM-Puffer gelöscht.

Datei importieren

Öffnet einen Dateiauswahl-Dialog, mit dem eine beliebige auf dem PC vorhanden Datei in das Disketten-Image und das Dateisystem von CP/M integriert werden kann. Hierbei gibt es natürlich Einschränkungen:

- Es muss noch entsprechend viel Platz im Dateisystem vorhanden sein
- Der Dateiname wird auf das CP/M Format umgerechnet
- Die zu importierende Datei darf noch nicht vorhanden sein

Datei exportieren

Öffnet einen Dateiauswahl-Dialog, mit dem die in der Liste markierte Datei auf dem Host-Computer gespeichert werden kann. Alle Dateien werden als Binärdateien gehandhabt. Da die exakte Dateilänge unter CP/M68K nicht ermittelt werden kann, sind die gesicherten Dateien immer ein Vielfaches von 128 Bytes. Die letzten Bytes des letzten Sektors können Daten von zuvor gespeicherten Dateien enthalten.

IMG speichern

Sichert die Image-Datei unter dem gleichen Namen und überschreibt die zuvor geladene Image-Datei.

ACHTUNG

Es erfolgt keine Nachfrage, ob dies erwünscht ist.

IMG speichern unter

Öffnet einen Dateiauswahl-Dialog, in dem ein neuer Ablageort für die bearbeitete Image-Datei angegeben werden kann. So gespeicherte Disketten-Images können sofort wieder in den Emulator (FLO2) geladen werden.

ACHTUNG

Auch hier erfolgt keine Nachfrage, wenn eine bereits bestehende Datei ersetzt wird.

Konfigurationsdateien

Die Konfigurationsdateien sind zeilenorientierte Textdatei, die zur Konfiguration der Hard- und Software für den Emulator dienen. Beim Start des Emulators wird die zuletzt genutzte Konfigurationsdatei zuerst analysiert und danach der Emulator mit den darin enthaltenen Vorgaben gestartet. Es können mehrere Konfigurationsdateien definiert werden, die im Unterverzeichnis CONFIG abgelegt sein müssen.

Während der Laufzeit des Emulators kann man im Hauptfenster über eine Auswahl der gewünschten Konfiguration zwischen den verschiedenen Konfigurationen wechseln, was jeweils einen Neustart der Software auslöst.

Bei Fehlern in einer ausgewählten Konfiguration wird eine Meldung mit den Fehlern angezeigt und die zuvor gewählte Konfiguration bleibt aktiv. Bei der Auswahl einer gültigen Konfiguration wird diese gespeichert. Beim nächsten Start der Emulation wird automatisch die letzte Konfiguration wiederhergestellt.

Konfigurationsdateien sind zeilenorientierte Textdateien mit den nachfolgend beschriebenen Möglichkeiten. Die Dateien dürfen Kommentare enthalten, die mit // eingeleitet werden oder optional mit * am Anfang einer Zeile. Die Reihenfolge der Schlüsselworte ist nicht relevant, die Reihenfolge der Parameter hingegen muss eingehalten werden.

Prozessordefinition

CPU

Dient zur Auswahl zwischen den emulierten Mikroprozessoren. Es ist genau eine CPU-Angabe in einer Konfigurationsdatei zulässig und notwendig.

```
CPU 68008          // CPU Motorola MC 68008 (8 MHz)
CPU 68000          // CPU Motorola MC 68000 (16 MHz)
CPU Z80A           // CPU Zilog Z80 (4 MHz)
CPU Z80B           // CPU Zilog Z80 (6 MHz)
```

CLOCK

Mit diesem Eintrag kann optional die von der CPU vorgegebene Taktfrequenz überschrieben werden. Die Angabe erfolgt in Hertz.

```
CLOCK 4000000      // 4 MHz Taktfrequenz der CPU
```

Auf einem MacBook Pro M1 liegt die maximale Emulationsgeschwindigkeit bei ca. 300 MHz. Zu niedrige Werte sollte man nicht angeben, Werte unter 100 kHz sind nicht sinnvoll.

Speicherkonfiguration

MEMORY

Legt abweichend von der durch die CPU vorgegebenen maximalen Speicherbereich eine andere Größe des Hauptspeichers fest. Die Angabe erfolgt in kByte.

```
MEMORY 64          // 64 kByte adressierbarer Speicher
MEMORY 1024        // 1 MByte adressierbarer Speicher
```

Die Angabe von MEMORY ist nicht notwendig, je nach gewähltem Mikroprozessor erfolgen sinnvolle Vorgaben, die jeweils den maximalen Adressbereich für diese CPU im NKC umfassen.

ROM

Definiert einen Adressbereich als Read Only Memory. Es müssen zwei Parameter für den Adressbereich und ein Parameter für den Namen der Image-Datei mit dem Inhalt des ROM folgen. Die beiden Adressen können dezimal oder hexadezimal angegeben werden. Die Image-Dateien werden im Unterverzeichnis ROMS des Emulators erwartet.

```
ROM 0x000000 0x007FFF EASS43.ROM      // GP ab Adresse 0x00000
ROM 0x0C0000 0x0C7FFF EASS43.ROM      // GP ab Adresse 0xC0000
```

Es dürfen beliebig viele Zeilen zur Definition der verschiedenen ROM-Blöcke benutzt werden. Werte, die außerhalb des physikalischen Adressraums der CPU liegen, werden ignoriert. Später definierte ROM-Blöcke können vorherige Definitionen überschreiben.

RAM

Definiert einen Adressbereich als Random Access Memory. Es müssen zwei Parameter folgen, welche die Startadresse und Endadresse des Speichers angeben. Die beiden Adressen können dezimal oder hexadezimal angegeben werden.

```
RAM 0x008000 0x009FFF                 // 8 kByte RAM ab Adresse 0x08000
RAM 0x000000 0x0BFFFF                 // 768 kB RAM ab Adresse 0x00000
```

Es dürfen beliebig viele Zeilen zur Definition der verschiedenen RAM-Blöcke benutzt werden. Werte, die außerhalb des physikalischen Adressraums der CPU liegen, werden ignoriert. Später definierte RAM-Blöcke können vorherige Definitionen überschreiben.

BOOTROM

Wie ROM, jedoch für die Baugruppe BANKBOOT. Der maximale Adressbereich liegt zwischen 0x0000 und 0x7FFF, da die Baugruppe BANKBOOT genau 32 kByte Speicherbereich bietet.

```
BOOTROM 0x0000 0x1FFF BOOT.ROM        // Boot ROM ab Adresse 0x0000
```

BOOTRAM

Wie RAM, jedoch für die Baugruppe BANKBOOT. Auch hier liegt der zulässige Adressbereich zwischen 0x0000 und 0x7FFF.

```
BOOTRAM 0x2000 0x7FFF                 // RAM auf BANKBOOT ab Adresse 0x2000
```

Modifizieren geladener Speicherbereiche

PATCHBYTE / PATCHWORD / PATCHLONG

Dient zum Patchen bereits geladener ROM-Bereiche. Es ist zu beachten, dass die Adressen geändert werden müssen, wenn die ROMs in abweichende Speicherbereiche geladen werden.

Beispiele:

```
PATCHWORD 0x003432 0x000A     // schnellere Copyright Meldung
PATCHWORD 0x0030BC 0x0011     // 80 Zeichen pro Zeile im Editor
```

```
PATCHLONG 0x000000 0x00F000      // Abweichender initialer Stack-Pointer
```

PATCHNOP

Dient zum Patchen bereits geladener ROM-Bereiche mit No-Operation-Befehlen (NOP). Als zweiter Parameter wird die Anzahl der zu speichernden NOP-Befehle angegeben. Jeder NOP-Befehl belegt 2 Bytes des Speichers.

```
PATCHNOP 0x00342C  5             // GP 4.3 kein Copyright ausgeben
```

PATCHBOOTBYTE / PATCHBOOTWORD / PATCHBOOTLONG

Wie oben, jedoch für die Baugruppe BANKBOOT

PATCHBOOTNOP

Wie oben, jedoch für die Baugruppe BANKBOOT

Konfiguration der Baugruppen

Jede Baugruppe darf aktuell nur einmal definiert werden. Die Angabe der Basisadressen ist immer optional, ohne Angabe wird die Standard-Adresse wie im NKC vorgesehen verwendet. Abweichende Basisadressen der Baugruppen führen in der Regel zu einer nicht korrekt funktionierenden Emulation, da die gesamte Software darauf nicht ausgerichtet ist.

BANKBOOT

Instanziiert den Hardware-Treiber für die Baugruppe BANKBOOT.

```
BANKBOOT

BANKBOOT 0xFFFFFFC8
```

Ohne Angabe des Treibers kann kein RAM-Speicher ab Adresse 0x0000 definiert werden und der Betrieb von zum Beispiel CP/M 68k ist nicht möglich. Wenn BANKBOOT angegeben wird, muss mindestens auch ein ROM ab Adresse 0x0000 mit einem BANKROM-Eintrag definiert werden.

KEY

Instanziiert den Hardware-Treiber für die Baugruppe KEY. In fast allen Konfigurationen wird die Baugruppe KEY benötigt, um das System überhaupt bedienen zu können.

```
KEY

KEY 0xFFFFFF68                   // Angabe der Basisadresse
```

KEYSWITCH

Dient zur Konfiguration der Einstellung der DIP-Schalter auf der Baugruppe KEY, die unter der Adresse 0xFFFFFF69 gelesen werden können. Einige Programme werten diese Schalter aus, um unterschiedliche Funktionen zur Verfügung zu stellen.

```
KEYSWITCH 0xFF                   // Alle Schalter auf ON

KEYSWITCH 0b011100111            // Standardeinstellung
```

GDP64K

Instanziiert den Hardware-Treiber für die Baugruppe GDP64K. Es sind 0 bis 3 Parameter gemäß den nachfolgenden Beispielen möglich. Die Standard-Werte für die Vergrößerung sind 2 für horizontal und 2 für vertikal. Daraus ergibt sich eine Größe des Monitor-Fensters von 1024*512 Pixel.

Für den korrekten Betrieb der Emulation mit den Grundprogrammen muss die Baugruppe GDP angemeldet sein. Ohne GDP erscheint nur das Hauptfenster und der Monitor lässt sich nicht aktivieren.

```
GDP64K                          // Standardeinstellung
GDP64K 0xFFFFFF70               // abweichende Basisadresse
GDP64K 2 3                      // geänderte Vergrößerung 2x3
GDP64K 2 3 0xFFFFFF70           // abweichende Adresse und Vergrößerung
```

Bei einer Vergrößerung von 2 horizontal und 3 vertikal ergibt sich ein weitestgehend dem NKC entsprechendes Seitenverhältnis. Mir persönlich gefällt die vorgegebene Auflösung besser.

GDP64HS

Aktiviert zusätzlich zu den Möglichkeiten der GDP64K die erweiterten Funktionen der GDP64HS. Der Aufruf funktioniert exakt wie bei GDP64K.

```
GDP64HS                         // Standardeinstellung
GDP64HS 0xFFFFFF70              // abweichende Basisadresse
GDP64HS 2 3                     // geänderte Vergrößerung 2x3
GDP64HS 2 3 0xFFFFFF70          // abweichende Adresse und Vergrößerung
```

Es ist zu beachten, dass nur entweder eine GDP64K oder GDP64HS in der Konfigurationsdatei angegeben werden darf. Beide Karten einzufinden ist auch mit unterschiedlichen Adressen nicht möglich.

Bei Verwendung der GDP64HS werden folgende Funktionen aktiviert

- RMW-Modus (Read-Modify-Write)
- Auslesen des Bildspeichers auf der GDP64HS-Baugruppe
- Hardware-Scrolling (geplant)

HINWEIS

Seit der Version 1.20 werden die Vergrößerungsfaktoren ignoriert, da dich die Größe des Monitor-Fensters frei einstellen lässt.

GDPFONT

Ermöglicht das Austauschen des im Grafikprozessor EF9366 integrierten Zeichensatzes. Ohne Angabe wird der originale Zeichensatz verwendet.

```
GDPFONT 0                       // Originaler Zeichensatz
GDPFONT 1                       // Verbesserter Zeichensatz
```

GDPCOLOR

Mit dieser Einstellung lassen sich die Vordergrund- und Hintergrundfarben des Monitor-Fensters festlegen. Die Farben lassen sich über einige feste Farbnamen oder über RGB-Werte einstellen. RGB-Werte können hexadezimal oder dezimal eingetragen werden.

```
GDPCOLOR YELLOW BLUE          // gelbe Schrift auf blauem Hintergrund
GDPCOLOR 0x800000 WHITE       // Dunkelrot auf weiß
GDPCOLOR WHITE 0              // Weiß auf schwarz
```

Die vordefinierten Farbnamen für den Vordergrund lauten:

```
WHITE, BLACK, GREEN, YELLOW, ORANGE, BLUE, NETBLUE
```

Die vordefinierten Farbnamen für den Hintergrund lauten:

```
WHITE, BLACK, BLUE, GRAY, SAND
```

Es sollte darauf geachtet werden, dass sich ein ausreichender Kontrast ergibt. Bei schwarzer Schrift auf schwarzem Grund, wird man nichts sehen können. Im Falle von nicht bekannten Farbnamen oder ungültigen RGB-Werten wird die entsprechende Farbe auf den Standardwert weißer Vordergrund oder schwarzer Hintergrund zurückgesetzt.

FLO2

Instanziiert den Treiber für die Baugruppe FLO2. Dieser Treiber stellt vier Laufwerke zur Verfügung, die mit Image-Dateien aus dem Unterverzeichnis DISKS geladen werden können.

Aktuell werden nur die ersten beiden Laufwerke im Mini-Format unterstützt.

```
FLO2
FLO2 0xFFFFFFC0               // Angabe der Basisadresse
```

CAS

Instanziiert den Treiber für die Baugruppe CAS zum Speichern von Dateien auf einem externen Kassettenrekorder.

```
CAS
CAS 0xFFFFFFCA                // Angabe der Basisadresse
```

PROMER

Instanziiert den Treiber für die Baugruppe PROMER zum Programmieren von EPROMS.

```
PROMER
PROMER 0xFFFFFF80             // Angabe der Basisadresse
```

CENT

Instanziiert den Treiber für die Baugruppe CENT zur Ansteuerung von Druckern.

```
CENT
CENT 0xFFFFFF48               // Angabe der Basisadresse
```

SER

Instanziiert den Treiber für die Baugruppe SER zur seriellen Datenübertragung.

```
SER
SER 0xFFFFFFC0                    // Angabe der Basisadresse
```

Bedienelemente

FRAMES

Einige der Baugruppen stellen Bedienelemente zur Verfügung, die im Hauptfenster der Emulation dargestellt werden können. Die Reihenfolge der Bedienelemente kann unabhängig von den Treibern eingestellt werden.

```
FRAMES FLO2 CAS PROMER CENT       // Reihenfolge der Bedienelemente
FRAMES CPU PROMER CAS
FRAMES FLO2 CENT
```

Die Angabe von FRAMES ist essenziell für die Bedienung des Emulators. Ohne Bedienelemente lassen sich die meisten Funktionen des Emulators und der geladenen Baugruppen nicht korrekt einsetzen. Schreibfehler oder nichtexistierende Bedienelemente werden vom Emulator ignoriert.

Die folgenden Bedienelemente sind aktuell verfügbar

FLO2	- Laden, auswerfen und Erstellen von Disk-Images
CAS	- Laden und Speichern von Kassetten-Dateien
PROMER	- Programmieren von EPROMS mit Speichermöglichkeit
CENT	- Druckerausgabe mit Speichermöglichkeit
CPU	- Anzeige der Register der CPU, 68000 oder Z80
GDP	- Anzeige der Register der Baugruppe GDP64K bzw. GDP64HS
IOE	- Anzeige der Baugruppe IOE mit LED und Schaltern
SER	- Anzeige der Baugruppe SER, siehe Beschreibung der Baugruppe

Weitere Einstellungen

DISK

Dient zum automatischen Einlegen von Disketten-Images für die Baugruppe FLO2. Es muss das Laufwerk und die Image-Datei angegeben werden. Der Wertebereich für die Nummer des Laufwerks liegt zwischen 0 und 3, andere Angaben werden ignoriert.

```
DISK 0 Assembler.IMG          // Belegen des ersten Laufwerks
DISK 1 NKC1CPM68K.IMG         // Belegen des zweiten Laufwerks
```

Nicht existente Image-Dateien führen zu einem Fehler während der Emulation, ähnlich als sei keine Diskette in das betreffende Laufwerk eingelegt.

Falls der Dateiname Leerzeichen enthält, muss der Dateiname in einfache oder doppelte Anführungszeichen gesetzt werden.

DISK 1 'Assembler Quelltexte.IMG'

DISK 2 "8 Zoll Testdiskette.IMG"

KEYS

Dient zum Hinterlegen einer Tastatursequenz, die sofort nach dem Start der Emulation in den Tastatur-Puffer übertragen wird. Dadurch ist es zum Beispiel möglich, bestimmte Menüpunkte des Grundprogramms automatisch aufzurufen

```
KEYS w cr w cr 4 cr          // Autostart CP/M im Grundprogramm 4.3
KEYS w cr 3 cr               // Anzeige der Bibliotheksprogramme
```

BP (Breakpoints)

Dient zum Definieren von beliebig vielen Breakpoints. Sobald die Instruktion an einer Adresse in der Liste der Breakpoints ausgeführt wurde, wird die Emulation in den Einzelschrittmodus versetzt und das Trace-Fenster mit der zuletzt ausgeführten Instruktion angezeigt.

Es können mehrere Schlüsselworte BP in einer Konfigurationsdatei verwendet werden.

```
BP 0x0033F4                  // 1 Breakpoint
BP 0x003C40 0x001012         // 2 Breakpoints
```

Mit Hilfe der weiteren Funktionen des Emulators kann der korrekte Ablauf von Programmen leicht kontrolliert werden. Die Ausführung des Programmes kann im Einzelschrittmodus weitergeführt werden, die Fortführung des automatischen Ablaufs ist jederzeit möglich.

NOAUTOSTART

Bestimmt, dass die Emulation nach der Konfiguration nicht automatisch gestartet wird. Dadurch ist es möglich, die Emulation ab dem Start im Einzelschrittmodus zu betreiben.

```
NOAUTOSTART
```

Die CPU muss manuell über den START-Button im Hauptfenster gestartet werden.

Vorgegebene Konfigurationen

Im Lieferumfang befinden sich mehrere vorgegebene Konfigurationsdateien, welche direkt über das Hauptfenster ausgewählt werden können. Die Auswahl einer neuen Konfiguration bewirkt einen sofortigen Neustart der Emulation.

Achtung: Eventuell nicht gespeicherte Dateien und Pufferinhalte gehen dabei verloren.

Alle in den vorgegebenen Konfigurationsdateien verwendeten EPROMs und Disketten-Images sind Bestandteil der Installationsdateien und werden beim ersten Start in das Benutzer-Verzeichnis kopiert.

Konfigurationen für Motorola MC 680x0

CP/M 68K 768k RAM

- CPU 68000 bei 16 MHz Taktfrequenz
- 768 kByte RAM ab Adresse 0x00000
- Grundprogramm 4.3 ab Adresse 0xC0000
- 32 kByte RAM hinter dem Grundprogramm
- PASCAL/S ab Adresse 0xD0000
- 32 kByte RAM hinter PASCAL
- EPROM Demo ab Adresse 0xEA000
- Baugruppen KEY, GDP, FLO2, PROMER, CAS, CENT, UHR, IOE und SER
- Autostart des CP/M

CP/M 68K mit GP 6.22

- CPU 68008 bei 8 MHz Taktfrequenz
- 256 kByte RAM ab Adresse 0x00000
- Grundprogramm 6.22 ab Adresse 0xC0000 bis 0xCDFFF
- 64 kByte RAM hinter Grundprogramm ab Adresse 0xD0000
- Baugruppen KEY, GDP64K, FLO2, PROMER, CAS, CENT und IOE

CP/M 68K mit GP 7.10

- CPU 68008 bei 8 MHz Taktfrequenz
- 256 kByte RAM ab Adresse 0x00000
- Grundprogramm 7.10 ab Adresse 0xC0000 bis 0xCFFFF
- 64 kByte RAM hinter Grundprogramm ab Adresse 0xD0000
- Baugruppen KEY, GDP64K, FLO2, PROMER, CAS, CENT und IOE

GP 4.3 Minimal

- CPU 68008 bei 8 MHz Taktfrequenz
- Grundprogramm 4.3 ab Adresse 0x00000
- 8 kByte RAM hinter Grundprogramm
- Baugruppen KEY, GDP64K, PROMER, CAS und IOE

GP 4.3 mit PASCAL

- CPU 68008 bei 8 MHz Taktfrequenz
- Grundprogramm 4.3 ab Adresse 0x00000
- 32 kByte RAM hinter Grundprogramm
- PASCAL/S ab Adresse 0x10000
- 32 kByte RAM hinter Pascal/S

- Baugruppen KEY, GDP64K, PROMER, CAS und CENT
- Keine Einschaltmeldung
- Automatisch 80 Zeichen pro Zeile

GP 4.3 mit RL-BASIC

- CPU 68008 bei 8 MHz Taktfrequenz
- Grundprogramm 4.3 ab Adresse 0x00000
- 32 kByte RAM hinter Grundprogramm
- RL-BASIC ab Adresse 0x10000
- 168 kByte RAM hinter RL-BASIC
- Baugruppen KEY, GDP64K, PROMER, CAS und CENT
- 32 kByte RAM hinter Pascal/S
- Keine Einschaltmeldung
- Automatisch 80 Zeichen pro Zeile

NKC 68K DEMO

- CPU 68008 bei 8 MHz Taktfrequenz
- Grundprogramm 4.3 ab Adresse 0x00000
- 8 kByte RAM hinter Grundprogramm
- Bibliothek DEMO ab Adresse 0x0A000
- Baugruppen KEY, GDP
- Keine Einschaltmeldung
- Autostart der Demo
- Live-Anzeige der CPU- und GDP-Register

Konfigurationen für Zilog Z80

Z80 GP 2.0 Debugger

- CPU Z80B bei 6 MHz Taktfrequenz
- RDK-Grundprogramm ab Adresse 0x0000
- Assembler & Debugger ab Adresse 0x6000
- 32 kByte RAM ab Adresse 0x8000
- Baugruppen KEY, GDP64K, PROMER, CAS, CENT und IOE

Z80 GP 2.0 Vollausbau

- CPU Z80B bei 6 MHz Taktfrequenz
- RDK-Grundprogramm ab Adresse 0x0000
- Programmiersprache GOSI ab Adresse 0x2000
- Programmiersprache BASIC ab Adresse 0x4000
- Assembler & Debugger ab Adresse 0x6000
- 32 kByte RAM ab Adresse 0x8000
- Baugruppen KEY, GDP64K, PROMER, CAS, CENT und IOE

Z80 GP 3.0

- CPU Z80A bei 4 MHz Taktfrequenz
- RDK-Grundprogramm ab Adresse 0x0000
- Assembler & Debugger ab Adresse 0x6000

- 32 kByte RAM ab Adresse 0x8000
- Baugruppen KEY, GDP64K, PROMER, CAS, CENT und IOE

Z80 GP 3.1

- CPU Z80A bei 4 MHz Taktfrequenz
- RDK-Grundprogramm ab Adresse 0x0000
- RL-Basic ab Adresse 0x4000
- Assembler & Debugger ab Adresse 0x6000
- 32 kByte RAM ab Adresse 0x8000
- Baugruppen KEY, GDP64K, PROMER, CAS, CENT und IOE

Z80 GP 2018 BankBoot

- CPU Z80 bei 4 MHz Taktfrequenz
- RDK-Grundprogramm auf BANKBOOT ab Adresse 0x0000
- 16 kByte RAM auf BANKBOOT ab Adresse 0x4000
- 1 MByte RAM ab Adresse 0x0000
- Baugruppen KEY, GDP64HS, PROMER, CAS, CENT, UHR, IOE und SER
- Geänderte Farbdarstellung: gelbe Schrift auf blauem Hintergrund

Anhänge

Installation

Mac OS X

Da ich keine Apple-Developer Lizenz besitze, ist die Installation des Emulators auf neueren OSX-Versionen (Sequoia und später) nicht mehr ohne Weiteres möglich.

Das Öffnen der PKG-Datei wird zunächst verweigert und es gibt nur die Möglichkeit, die heruntergeladene Datei in den Papierkorb zu legen.

Um das Installationsprogramm trotzdem öffnen zu können, muss man die Systemeinstellungen öffnen, und auf der linken Seite „Datenschutz und Sicherheit“ auswählen. Dort relativ weit nach unten scrollen, bis der folgende Abschnitt sichtbar ist.

Jetzt kann die ursprüngliche Warnung (erstes Bild) geschlossen werden um direkt danach auf den Button „Dennoch öffnen“ in den Systemeinstellungen zu klicken.

Es erscheint wieder die Warnung, dass Apple das Programm nicht verifizieren kann. Jetzt jedoch erweitert um den Button „Dennoch öffnen“. Nach der Eingabe des Passwortes öffnet sich das Installationsprogramm.

Die Apple-Developer Lizenz kostet 99 € pro Jahr. Für ein reines Hobby ist es mir das nicht wert.

Quellenangaben

Bücher und Hefte

Hauptsächliche Quelle für die Umsetzung der Hardware-Treiber waren die Bücher und Hefte, die von Rolf Dieter Klein über den Francis-Verlag herausgegeben wurden.

- Mikrocomputer selbstgebaut und programmiert ISBN 3-7723-7161-2
- Rechner modular ISBN 3-7723-8721-7
- Die Prozessoren 68000 und 68008 ISBN 3-7723-7651-7
- Anwenderhandbuch CP/M 68K ISBN 3-7723-9751-4

Zusätzlich die Datenblätter der im System verwendeten integrierten Schaltkreise.

- Motorola M68000 Microprocessors User's Manual
- Motorola M68000 Programmer's Manual
- Thomson EF9366 Data Sheet
- 6850 Data Sheet
- Western Digital FD 179X Floppy Disk Controller

68000 Emulation

Grundlage für die Emulation der 32 Bit Motorola Mikroprozessoren 68000 und 68008 war die Vorarbeit von Tony Headford, der eine gut funktionierende Emulation der Mikroprozessoren für Java umgesetzt und veröffentlicht hat.

- Projektseite https://github.com/tonyheadford/m68k

Es waren einige Fehlerbehebungen, Erweiterungen und Anpassungen notwendig, um eine möglichst gute Integration in diese Emulation des NKC zu sichern. Die modifizierten Stellen sind im Quellcode entsprechend markiert.

- Anpassung der Zyklen gemäß Motorola Manual
- Ableitung des 68008 Prozessors
- Zusätzliche Methoden der CPU
  - getProcessorName()
  - getNominalFrequency()
  - getAddressMask()
- Anpassung der Klasse DisassembledInstruction()
- Verarbeitung der Unterprogramm-Ebenen

Hex Editor

Der integrierte Hex-Editor für den Hauptspeicher basiert auf dem Hexadecimal File Editor von Keith Fenske. Auch hier waren einige Erweiterungen notwendig, um dem Editor Zugriff auf den Hauptspeicher der Emulation zu gewähren.

- Projektseite https://kwfenske.github.io

Die zusätzlichen Funktionen wurden in einer eigenen Klasse umgesetzt, welche die originale Klasse erweitert und als Ersatz für die main() Methode dient.

GP-EDAS

Das für den Z80 konzipierte Grundprogramm GP-EDAS ist eine Weiterentwicklung des GP 1018 von Steffen Reimer (DL2LCE), welche einen Editor und Assembler bereitstellt. Die jeweils aktuelle Version kann von seiner Homepage geladen werden.

- Projektseite http://www.entschweben.de

Der darin enthaltene Editor und Assembler EDAS*4 des AC1 wurde erstmals in der Zeitschrift Funkamateur Ausgaben 01 bis 04 von 1987 veröffentlicht.

Dateiformate

Baugruppe FLO2

Die Image-Dateien für die Baugruppe FLO2 sind jeweils in einer durchgehenden Binärdatei im Unterverzeichnis DISKS des Emulators abgelegt. Neben den reinen Daten sind keine weiteren Informationen in der Datei enthalten. Das Format der Image-Datei wird anhand der Dateigröße erkannt. Jede Diskette ist unterteilt in Seiten, Spuren und Sektoren mit einer bestimmten Anzahl an Bytes. Aktuell werden die folgenden Diskettenformate unterstützt:

NKC Mini Format – 5,25 oder 3,5 Zoll – 800 kByte Brutto

Dieses ist das standardmäßig im NKC verwendete Diskettenformat welches unter anderem auch für den Betrieb von CP/M 68K verwendet wird. Dieses Format wird von der Baugruppe FLO2 auf den ersten beiden Laufwerken A: und B: unterstützt.

- Spuren 80 pro Seite
- Seiten 2 - doppelseitig
- Sektoren 5 pro Spur
- Kapazität 1024 Bytes pro Sektor
- Imagedatei 2 x 80 x 5 x 1024 = 819200 Bytes = 800 kByte

Anhand des Beispielcodes kann man den Aufbau der Image-Dateien und die Reihenfolge der Sektoren nachvollziehen.

```
for (int track = 0; track < 80; track ++)
     for (int side = 0; side < 2; side++)
          for (int sector = 0; sector < 5; sector ++)
               for (int data = 0; data < 1024; data++)
                    // Datenbyte schreiben oder lesen
```

NKC Maxi Format – 8 Zoll – 250 kByte Brutto

Dieses Format wurde auf älteren 8 Zoll Laufwerken verwendet, die nur einen Schreib- und Lesekopf besitzen. Die Disketten konnten beidseitig beschrieben werden, nachdem man sie andersherum eingelegt hatte. Das Format wird von der Baugruppe FLO2 auf den Laufwerken C: und D: unterstützt.

- Spuren 77 pro Seite
- Seiten 1 - einseitig
- Sektoren 26 pro Spur
- Kapazität 128 Bytes pro Sektor
- Imagedatei 1 x 77 x 26 x 128 = 256256 Bytes = 250 kByte

Anhand des Beispielcodes kann man den Aufbau der Image-Dateien und die Reihenfolge der Sektoren nachvollziehen.

```
for (int track = 0; track < 77; track ++)
      for (int sector = 0; sector < 26; sector ++)
            for (int data = 0; data < 128; data++)
                  // Datenbyte schreiben oder lesen
```

Baugruppe CAS

Die Emulation der Baugruppe CAS ermöglicht das Speichern von Daten, die an die Baugruppe gesendet werden und in realer Hardware auf einem Kassettenrekorder gespeichert worden wären. Ein Puffer speichert alle gesendeten Bytes inklusive der Synchronisationssequenzen. Beim Speichern wird der gesamte Puffer als Binärdatei im Unterverzeichnis CAS des Emulators gesichert. Diese Dateien können auch wieder geladen werden. Der Aufbau einer Aufzeichnung wird durch die Implementation in den Grundprogrammen vorgegeben.

Schreiben von Textdateien

- 40 Bytes 0xFF Synchronisation
- 2 Bytes 0x00 0x2F
- N Bytes DATEINAME 0x0D
- 40 Bytes 0xFF Synchronisation
- 2 Bytes 0x00 0x3B
- 4 Bytes Startadresse
- 4 Bytes Endadresse
- N Bytes Daten Nutzdaten
- 4 Bytes Prüfsumme
- 20 Bytes 0xFF Ende

Schreiben von Datendateien

- 40 Bytes 0xFF Synchronisation
- 2 Bytes 0x00 0x2F
- N Bytes DATEINAME 0x0D
- 32 Bytes 0xFF Synchronisation
- 1 Bytes 0x00
- N Bytes Text Nutzdaten
- 1 Byte 0x00 Textende
- 2 Bytes Prüfsumme
- 20 Bytes 0xFF Ende

Baugruppe PROMER

Die Baugruppe PROMER hat einen internen Pufferspeicher mit 8 kByte Umfang. Das Bedienfeld bietet die Möglichkeit, Binärdaten zu laden und zu speichern. Beim Speichern werden immer genau 8 kByte umfassende Binärdateien erzeugt, beim Laden sind abweichende Dateigrößen möglich. Bei kleineren Dateien wird der Rest des Pufferspeichers mit 0xFF aufgefüllt, bei größeren Dateien wird der Rest abgeschnitten. Die Datenablage erfolgt im Unterverzeichnis PROMER des Benutzerordners des Emulators.

Baugruppe SER

Daten, die über eine in der Emulation laufende Software an die serielle Schnittstelle gesendet werden, landen in einem Sendepuffer, dessen Inhalt auf dem Host-Computer gesichert werden kann. Alle Daten werden als reine Binärdatei abgelegt und erhalten die Dateiendung .SER. Diese Dateien werden im Unterordner SER des Benutzerordners für den Emulator abgelegt.

Enthaltene Dateien

Unterverzeichnis ROMS/68K

68008_GP430.BIN

Grundprogramm Version 4.3 vom Entwickler des Computers Rolf Dieter Klein als einzelnes 32 kByte Image mit grundlegenden Funktionen und einem integrierten Editor und Assembler. Das Grundprogramm erwartet mindestens 8 kByte RAM ab der Startadresse + 0x08000

68008_GP622.BIN

Grundprogramm Version 6.22 von Ralph Dombrowski. Ursprünglich besteht diese Version aus sieben Images zu 8 kByte, die für den Emulator zu einem einzigen 56 kByte umfassendes Image zusammengefasst wurden. Das Grundprogramm erwartet mindestens 8 kByte RAM-Speicher ab der Startadresse + 0x10000.

68008_GP710.BIN

Grundprogramm Version 7.10 von Jens Mewes. Ursprünglich besteht diese Version aus acht Images zu 8 kByte, die für den Emulator zu einem einzigen 64 kByte umfassendes Image zusammengefasst wurden. Das Grundprogramm erwartet mindestens 8 kByte RAM-Speicher ab der Startadresse + 0x10000.

68008_GOSI.BIN

Umsetzung der Grafisch Orientierten Sprache I (GOSI) in Form eines Bibliotheksprogramms. Zum Einsatz wird eines der Grundprogramme benötigt. Das ROM-Image hat einen Umfang von 24 kByte und kann an einer beliebigen freien Stelle des Hauptspeichers geladen werden.

68008_PASCAL.BIN

PASCAL/S ist eine Umsetzung des PASCAL Compilers der ETH Zürich von Rolf Dieter Klein. Das Bibliotheksprogramm besteht aus zwei Teilen, dem Compiler und der Run-Time-Umgebung. Quellcodes werden mit dem Editor des Grundprogramms eingegeben, danach mit PASCAL/S übersetzt und mit PCODE ausgeführt. Die beiden Bibliotheken belegen 32 kByte Speicher.

68008_BASIC.BIN

RL-Basic ist eine Umsetzung der Programmiersprache BASIC für den NKC. Nach dem Start muss zunächst der Arbeitsbereich gesetzt werden, da das Programm nicht automatisch danach sucht. Falls der Arbeitsbereich (RAM-Bereich) nicht auf einen gültigen RAM-Bereich gesetzt wird, funktioniert die Software nicht.

68008_DEMO11.BIN

Demonstration der grafischen Möglichkeiten des NKC von Rolf Dieter Klein. Das ROM kann an einer beliebigen freien Stelle des Hauptspeichers geladen werden und über die Bibliotheks-Funktion der Grundprogramme gestartet werden.

Unterverzeichnis ROMS/Z80

Z80_EGRUND20.BIN, Z80_EGRUND20.LBL, Z80_EGRUND20.ASM

Das Grundprogramm in der Version 2.0 vom Entwickler Rolf Dieter Klein. Das ROM muss ab Adresse 0x0000 geladen werden und erwartet mindestens 8 kByte RAM ab der Adresse 0x8000.

Im Einzelschrittmodus wird automatisch die LBL-Datei geladen und die klarschriftlichen Labels im Trace-Fenster angezeigt. Zusätzlich befindet sich der komplette Quelltext des GP im Lieferumfang.

Z80_EGRUND30.BIN

Das von mir im Jahre 2013 erweiterte Grundprogramm in der Version 3.0 mit einem einseitigen Menü. Die Menüeinträge zum Laden und Speichern auf einem USB-Stick funktionieren in dieser Version des Emulators noch nicht, da hierfür eine angepasste Version des IOE-Treibers notwendig wäre.

Z80_EGRUND31.BIN

Version 3.1 des Z80 Grundprogramms mit 3 weiteren Menüpunkten zum direkten Starten der Programmiersprachen GOSI und BASIC und zum Starten des Assemblers/Debuggers. Die entsprechenden ROM müssen an bestimmte Adressen geladen werden. (siehe unten)

Z80_GP2018V3.BIN, Z80_GP2018V3.ASM

Ein neues Z80 Grundprogramm mit integriertem Monitor, welches ich im Laufe des Jahres 2018 in Zusammenarbeit mit Steffen Reimer umgesetzt habe. Das Grundprogramm unterstützt vier Bildschirmseiten und besitzt einen integrierten Disassembler.

Z80_EZASS31.BIN

Assembler und Debugger von Rolf Dieter Klein. Das ROM muss ab der Adresse 0x6000 geladen werden und benötigt das Grundprogramm in der Version 2.0 oder Version 3.1. In Verbindung mit dem Grundprogramm 3.0 funktioniert der Rücksprung in das GP nicht.

Unterverzeichnis BOOT/68K

BOOTORIG.ROM

Dient zum Einsatz auf der Baugruppe BANKBOOT und sucht innerhalb des gesamten Speichers nach der Signatur des Grundprogrammes. Wenn es gefunden wurde wird BANKBOOT deaktiviert und das Grundprogramm gestartet. Zur Nutzung muss in der Konfiguration Hauptspeicher ab Adresse 0x0000 definiert werden. RAM ist auf der BANBOOT Baugruppe nicht notwendig.

Unterverzeichnis BOOT/Z80

Z80_GP2018V3.ROM

Das Grundprogramm in der Version 2018 kann auch auf der Baugruppe BANKBOOT verwendet werden. Beim Start des Systems wird der Inhalt des ROM in den RAM-Speicher kopiert und danach gestartet.

Unterverzeichnis DISKS

Alle Diskettenimages können derzeit nur unter CP/M 68K genutzt werden.

NKC1CPM68K.IMG bis NKC8CPM68K.IMG

Diese Images beinhalten das eigentliche CP/M System, wie es im originalen Umfang für den NKC zusammengestellt wurde.

1. Startdiskette mit NKC spezifischen Programmen
2. Assembler und Debugger
3. Linker und Relocator
4. Terminal, Formatier-Programm
5. BIOS, CP/M Relocator
6. Winchester-Treiber
7. C-Compiler und Tools
8. CP/M Z80 Emulation

NKC1CPM68K.IMG

```
B: CPM      SYS : MAKECPM  SUB : MAKE128  SUB : MAKE256  SUB : MAKE512  SUB
B: RELCPM   SUB : CPMLDR   SYS : MAKELDR  SUB : UFO68008 68K : UFO68000 68K
B: UFO68020 68K : PIP      68K : COPY     68K : STAT     68K : XPUTBOOT 68K
B: LADE     68K : SAVE     68K : EDITRDK  68K : ASSRDK   68K : STARTE   68K
B: LADE     S   : SAVE     S   : EDITRDK  S   : ASSRDK   S   : STARTE   S
B: NDRBOOT  S   : XPUTBOOT S   : NDRLDRB  S   : NDRBIOS  S   : UFO680XX S
B: MODEM    S   : MIKRODOS S   : LIESMICH     : MAK1CPM  SUB :
```

NKC2CPM68K.IMG

```
B: AS68     REL : RELOC    REL : PIP      REL : INIT     REL : COPY     REL
B: STAT     REL : RELOC1   SUB : README   TXT : DDT      REL : DDT68000 68K
B: CPM      SYS : AS68INIT     : DUMP     REL : DDT10    REL : DDT68010 68K
B: RELOC2   SUB
```

NKC3CPM68K.IMG

```
B: LO68     REL : S        O   : CLIB         : LIBE     A   : LIBF     A
B: CLINK    SUB : CLINKE   SUB : CLINKF   SUB : RELOC3  SUB : ED       REL
B: CP68     REL : ASSERT   H   : CTYPE    H   : ERRNO    H   : OPTION   H
B: OSIFERR  H   : PORTAB   H   : SETJMP   H   : SIGNAL   H   : STDIO    H
B: AR68     REL : FIND     REL : MORE     C   : MORE     REL : NM68     REL
B: OSATTR   H   : OSIF     H   : RELOC4   SUB : C        SUB : CE       SUB
B: C068     REL : C168     REL : RELOC5   SUB
```

NKC4CPM68K.IMG

```
B: LINK68   REL : LOADR    O   : OVHDLR   O   : SENDC68  REL : SIZE68   REL
B: TERM     C   : TERMA    S   : TERM     REL : XFER86   C   : XFER86   REL
B: FORMAT   S   : FORMAT   REL : INIT     S   : RELOC6   SUB : CONFIG   C
B: CONFIG   REL : BDOS     S   : BIOS     C   : BIOS     O   : BIOSA    S
B: BIOSA    O   : BIOSTYPS H   : CBIOS    S   : ELDBIOS  S   : ERGBIOS  S
B: LDBIOSA  S   : LDBIOSA  O   : LDBIOS   O   : LOADBIOS H   : LOADBIOS SUB
B: NOBIOSHI S   : NOBIOSLO S   : NORMBIOS H   : NORMBIOS SUB : PUTBOOT  S
B: PUTBOOT  REL : VT52     C   : VT52     O   : BOOTER   S   : BOOTER   O
B: RELOC7   SUB
```

NKC5CPM68K.IMG

```
B: CPMLIB       : LDRLIB       : CPM      REL : CPM15000 MAP : CPM15000 SR
B: CPMLDR   SYS : LCPM     SUB : LCPM10   SUB : MAKELDR  SUB : RELCPM   SUB
B: RELOC8   SUB : XPUTBOOT S   : CPM400   SR  : XPUTBOOT REL : XBOOTER  S
B: XBOOTER  O   : XBIOS    C   : XBIOSA   S   : XBIOSA   O   : XLDBIOSA S
B: XLDBIOSA O   : XLOADBIO H   : XNORMBIO H   : XFLDBIOS H   : XFNMBIOS H
B: XNORMBIO SUB : XMAKELDR SUB : XLCPM    SUB : XBIOS    O   : XCPM     SYS
B: XCPMLDR  SYS : RELOC9   SUB : XCPM     REL : CPM400   MAP : XLOADBIO SUB
```

NKC6CPM68K.IMG

```
B: C068     REL : LIBF     A   : LIBE     A   : WIFORM68 A68 : WI68BIOS S
B: WIFORM68 68K : WIFORM68 S   : LIESMICH BAK : LIESMICH TXT : C168     REL
```

NKC7CPM68K.IMG

```
B: PIP      68K : STAT     68K : EDITRDK  68K : SIZE68   68K : RELOC    68K
B: CLINK    SUB : LOADR    O   : CPM      SYS : C068     68K : C168     68K
B: CP68     68K : CLINKF   SUB : C        SUB : CE       SUB : MORE     C
B: ERRNO    H   : LO68     68K : SETJMP   H   : SIGNAL   H   : STDIO    H
B: OSATTR   H   : MORE     O   : ASSERT   H   : XBIOS    O   : OSIF     H
B: S        O   : CTYPE    H   : LIBF     A   : OPTION   H   : OSIFERR  H
B: PORTAB   H   : AS68     68K : AS68SYMB DAT : CLIB         : AS68INIT
B: CLINKE   SUB : LIBE     A
```

NKC8CPM68K.IMG

```
B: READ     ASM : DU       DOC : REZ80    DOC : EMUIO    S   : MOVPAT   ASM
B: XSUB     COM : CRCGEN   COM : DDTZ     COM : DU       COM : @        COM
B: INIDIR   COM : REZ80    COM : GEMU     DOC : EMU      DOC : LOOP     TXT
B: READ     COM : DDTZ     DOC : SCOPY    COM : MOVPAT   BAT : TMT      COM
B: RCV      COM : CPMZ80   68K : EMUIO    DVR : GRUND4,3 68K
```

ASSEMBLER.IMG

Beinhaltet notwendige Anwendungen, um eigene Assembler-Programme zu erzeugen. Von diesem Image kann gebootet werden. Alle Tools zum Übersetzen von Quelltexten sind enthalten. Unter Anderem findet dich hier ein paar Programme die nützlich sein können:

- PROMER.S Baugruppe PROMER direkt aus CP/M ansteuern
- TIME.S Anzeige der aktuellen Uhrzeit
- CLS.S Löschen des Bildschirms
- TPAINFO.S Anzeige der Größe der TPA (Transient Program Area)
- DPBINFO.S Anzeige von Infos über den DPB (Disk Parameter Block)
- BPINFO.S Anzeige von Infos über die BP (Base Page)
- FREE.S Anzeige der Belegung der Diskette
- MANDEL.S Berechnung der Mandelbrotmenge (Apfelmännchen)

```
A: CPM      SYS : CPMLDR   SYS : AS68SYMB DAT : TIME     S   : COPY     68K
A: PIP      68K : STAT     68K : RELOC    68K : MORE     68K : MAKE     SUB
A: FSAVE    S   : PROMER   68K : LO68     68K : DUMP     68K : PACMAN   68K
A: PROMER   S   : TIME     68K : EDIT     S   : FIND     68K : SIM1     S
A: SIM1     68K : CLS      68K : CLS      S   : AS68     68K : EDIT     68K
A: DUMP     S   : SAVE     68K : TPAINFO  S   : SAVE     S   : MAKE2    SUB
A: M        SUB : TPAINFO  68K : FSAVE    68K : TPALEN   68K : DPBINFO  S
A: FILESIZE S   : BPINFO   S   : DPBINFO  68K : FCOMP    S   : SIZE     S
A: FCOMP    68K : TPALEN   S   : SIZE     68K : BPINFO   68K : FREE     S
A: FILESIZE 68K : JOINTEXT S   : FREE     68K : LOADFILE S   : LOADFILE 68K
A: JOINTEXT 68K : MAKETEXT S   : MAKETEXT 68K : LIB      S   : MANDEL   S
A: PACMAN   S   : MANDEL   68K : MANDEL2  S   : MANDEL2  68K : O        SUB
```

BIOS & CPM Reloc.IMG

Beinhaltet alle Tools um die Systemdateien des CP/M erneut zu übersetzen und an die aktuelle Speichergröße anzupassen. Verschiedene SUBMIT-Dateien gestatten das Erstellen des Systems für 128, 256, 512, und 768 kByte RAM.

```
B: COPY     68K : PIP      68K : STAT     68K : RELOC    68K : MORE     68K
B: CLS      68K : LO68     68K : DUMP     68K : AS68SYMB DAT : PROMER   68K
B: FIND     68K : AS68     68K : EDIT     68K : NDRBIOS  S   : NDRBIOS  O
B: MAKE256  SUB : CPM      REL : CPMLIB       : NDRBIOS2 O   : CPM      SYS
B: CPMLDR   SYS : MAKECPM2 SUB : NDRBIOS2 S   : MAKE512  SUB : TPALEN   68K
B: SIZE     68K : MAKE768  SUB : MAKE128  SUB
```

8 Inch SS SD.IMG

Ein Beispiel-Image für die Laufwerke C: und D: des Emulators. Das aktuelle CP/M BIOS unterstützt hier ausschließlich Images in diesem Format. Das Beispiel-Image enthält Kopien einiger Quelltexte, die auch auf Assembler.IMG zu finden sind.