MONX



MONX-pico ist die Erweiterung des Monitors 3.1 um zusätzliche Monitorkommandos zur Arbeit mit der Hardware des picoAC1, insbesonders zur Arbeit mit der SD-Karte.

Neben den Kommandos gibt es auch Routinen zur Dateiarbeit mit der SD-Karte des picoAC1. Damit können Programme über eine einheitliche Schnittstelle auf Massenspeicher zugreifen, eine Funktionalität, die den bisherigen Systemen (AC1-ACC, AC1-SCCH, AC1-2010) leider fehlte.

Ich hab das Konzept ein wenig erweitert, so dass auch andere Systeme und andere Monitore von MONX partizipieren können. Man lädt einfach einen passenden MONX-Treiber (MONX-pico, MONX-VDIP). Mit monx_inc.asm kompilierte Programme suchen eine MONX-Erweiterung im RAM und bieten dann Massenspeicherzugriff, ohne dass individuelle Programme erzeugt werden müssen.

Es gibt derzeit

- BASIC8K (BASIC8K-SD, mit DSAVE, DLOAD, DLOAD„ “ für Dir)

- nascom-Emulator-Emulator

Folgende Software nutzt MONX, unterstützt aber nur MONX auf Adresse 0800h.

- picoEDAS
- Pascal
- Forth 83

Außerdem gibt es funktionskompatible Erweiterungen für andere Speicher:

- USB-Stick via VDIP (s.u.)
- CH376 (in Entwicklung)

Downloads

- monx.zip

Version 0.5 10.9.2026. Alle Quellen und Binaries

- MONX-pico auf Adresse C000 und E000, für die picoAC1-SD-Karte
- MONX-VDIP für eine VDIP-USB-Stick-Hardware (AC1-201, picoAC1, JKCEMU)
- monx_inc.asm Include-Datei, linkt dynamisch eine MONX-Erweiterung
- Testprogramme test1, test2
- BASIC8K (angepasst auf monx_inc)

Kommandos

```
picoAC1 (MonX 1.03)

H           Help        Hilfetexte Bedienung "Ur-"Monitor"
a AADR Dateiname            Laden von Dateien an Absolut-Adresse im RAM
c [Zeichen-:Hintergrund-:Rahmenfarbe]   Bildschirmfarben anzeigen/setzen
d [Verzeichnisname|Nr]  dir        Verzeichnisse anzeigen / wechseln
h           help        Hilfetexte Bedienung "Monitorerweiterungen"
l Dateiname|Nr      load        Daten von SD-Karte laden
m Verzeichnisname   mkdir        Verzeichnis anlegen
r Dateiname|Nr      remove        Datei in Papierkorb verschieben
s AAAA EEEE SSSS Dateiname/Infotext Daten auf SD-Karte schreiben
t [YYYY-MM-DD*hh:mm:ss] time        Systemzeit anzeigen/setzen

picoAC1 (Mon 1.10)

Z           Minibasic
v           version        Version anzeigen
?           kdolist        Kommandos auslisten
```

Dokumentation

Im codeberg-Repository zum picoAC1 findet man

```
Bedienung.md
monx_interface_v110.txt     Beschreibung der lo-level-Funktionen
                zum Einbau des SD-Zugriffs in eigene Programme
monx_noac1.E (-> test1.asm)  Laden einer Nicht-AC1-Datei
monx_test.S (-> test2.asm)   Testroutinen fuer die AC1-Monitorerw.
monx_fsize.E            Abfrage der Länge einer Datei auf SD-Karte
```

Die MONX-Monitorkommandos habe ich auch hier beschrieben.

MONX für andere Monitore

Original ist MONX fest auf Adresse 800h mit dem ACC-Monitor 3.1 verbunden. Dieser Adressbereich ist bei anderen Monitoren leider belegt. Damit MONX auch mit anderen Monitoren genutzt werden können, bietet es sich an, diese auf andere Adressen zu linken. Das betrifft auch die genutzten Speicherzellen. Zum Glück werden diese nicht direkt angesprochen, sondern über einen Funktionsaufruf berechnet.

```
monx_E000.z80       ist MONX 1.1.0 für Adresse E000...E7FF
            die Speicherzellen liegen in diesem Bereich
```

Diese Monitorerweiterung wird im MON3.1-Monitor geladen. Dann kann der Monitor gewechselt werden, z.B. MON SCCH 10/88.

Die Monitorerweiterung monx_E000 steht nun im SCCH-Monitor zur Verfügung.

ABER: Damit Programme ihn auch nutzen können, müssen die Programme natürlich darauf vorbreitet werden, statt der originalen Systemrufadressen auf 800h ff nun andere Adressen zu nutzen. Dafür muss man die Programme mit der include-Datei

```
include monx_inc.asm
call    xinit        ; ret A=FF, NZ, wenn keine Monitoerweiterung gefunden
```

assemblieren. Diese enthält statt einfacher EQU's einen kleinen Sprungverteiler, der je nach vorhandener Monitorerweiterung auf 800h, oder E000h zeigt oder - wenn gar keine Monitorerweiterung vorhanden ist, Fehler ausgibt.

Die Routine xinit ist zwingend beim Programmstart o.ä. einzubinden, mindestens vor dem ersten Aufruf einer MONX-Systemroutine.

Ein Test auf eine vorhandene Monitorerweiterung durch den Vergleich, ob auf 0FFEh „pi“ steht, ist leider nicht mehr allgemein möglich und muss auskommentiert werden. Stattdessen liefert xinit den Wert FF zurück, wenn keine Monitorerweiterung gefunden wurde, sonst 0.

MONX-VDIP



Für die VDIP-USB-Erweiterung am AC1-2010 und am picoAC1 habe ich einen kompatiblen Treiber MONX-VDIP geschrieben, der statt der SD-Karte einen USB-Stick via VDIP unterstützt. Programme mit monx-inc laufen dann mit USB als Speichermedium.

Es sind folgende Systemkommandos implementiert:

```
a AADR Dateiname            Laden von Dateien an Absolut-Adresse im RAM
d [Verzeichnisname] dir        Verzeichnisse auflisten / wechseln
l [Dateiname]       load        Dateien auflisten / Datei von USB laden
(Headersave-Format)
                    Am Ende steht die Startadr. in ARG1 (für J:)
m Verzeichnisname   mkdir        Verzeichnis anlegen
r Dateiname     remove        Datei in Papierkorb verschieben
s AAAA EEEE SSSS Dateiname/Infotext Datei auf USB schreiben (Headersave-
Format)
s: Dateiname                erneut speichern mit gleichen Adr.
?[:]            kdolist        Kommandos auflisten / mit Adressen

Nicht enthalten sind:

H           Help        Hilfetexte Bedienung "Ur-"Monitor"
h           help        Hilfetexte Bedienung "Monitorerweiterungen"
t [YYYY-MM-DD*hh:mm:ss] time        Systemzeit anzeigen/setzen
Z           Minibasic
v           version        Version anzeigen
```

Änderungen gegenüber MONX-pico:

* Die Auswahl von Verzeichnissen oder Dateien über Nummern gibt es nicht, es wird immer mit dem Dateinamen gearbeitet * Es werden nur 8.3-Dateinamen unterstützt * Es sind beliebig geschachtelte Unterverzeichnisse möglich * Wechsel ins übergeordnete Verzeichnis mit 'd ..' * Wechsel ins Root-Verzeichnis mit 'd /' * Beim SCCH-Monitor wird das Zeichen '_' (0x5F) als Backspace genutzt und erscheint nicht auf dem Bildschirm. Man sollte daher auf '_' in Dateinamen verzichten. * Das ?-Kommando zeigt ohne Parameter nur die Kommandobuchstaben an

Dateien

Die Dateien auf dem Massenspeicher werden im Headersave-Format gespeichert, s. usb_z80-dateien. Den eigentlichen Daten ist ein 32 Byte großer Startblock (aka Kopfblock/Header) vorangestellt.

Header	Programm

Byte	Bedeutung
0, 1	Anfangsadresse der Datei
2, 3	Endadresse
4, 5	Startadresse bei Fall eines ausführbaren Programms (Dateityp: „C“)
6-11	frei (wird manchmal benutzt für CRC oder Autor)
12	Dateityp (z.B. „C“: Ausführbares Maschinencodeprogramm)
13-15	Headersave-Kennung „magic bytes“ (3x D3h)
16-31	Dateiname/Beschreibung, mit Leerzeichen aufgefüllt

Der Dateityp ist prinzipiell frei wählbar, MONX nutzt 'D'. Verbreitet sind

- P Lauffähiges Maschinenprogramm (z1013: C !!)
- D Maschinendatei
- B Basicprogramm
- F Datenfeld (BASIC)

Die 16 Zeichen „Dateiname“ können frei für lange Programmnamen oder zusätzliche Informationen genutzt werden, der eigentliche Dateiname liegt bereits im Filesystem vor.

Originale Kassettenbasierte Programme arbeiten immer in 32-Byte-Blöcken. Sind die letzten Daten kürzer als 32 Byte, wird der letzte Block auf volle 32 Byte erweitert. Bei den heute üblichen Massenspeichern (SD-Karte, USB) ist das nicht mehr notwendig, aus Komaptibilitätsgründen sollte es aber trotzdem gemacht werden.

MONX-pico nutzt FAT32-formatierte SD-Karten, unterstützt aber nur 16 Zeichen für den Dateinamen (incl. Dateityp) und nur eine Unterverzeichnis-Ebene.

MONX-VDIP nutzt FAT16-formatierte USB-Sticks, unterstützt aber nur kurze 8.3-Dateinamen, dafür mehrere Unterverzeichnis-Ebenen.

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

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/homecomputer/ac1/software/monx?rev=1789118504

Last update: **2026/09/11 09:21**