

# Forth 83

2024/2025 habe ich das FG FORTH83 d. DDR an den AC1 angepasst. Es läuft auf allen Systemen, speziell für den picoAC1 gibt es I/O-Routinen zur Nutzung der SD-Karte.



```

Z80 FORTH 1.3 (AC1)
AG FORTH
Th. Beierlein / V. Pohlers

WORDS
(WHERE) EDIT ED EDITOR DARK TA AT CUROFF
YESNO EMPTY REA REASSEMBLER SAVE-SYSTEM FILES
EMPTY SEE .4 DECOMPILER ASSOCIATIVE: CASE: MAP
OUT EMPTY PROCLAIM LMARGIN MARK GET PUT DU
DUMP ? LINE HELLO ROOT P! Pg COPY FH \S
\ INDEX LIST THRU LOAD --> \SCR C/L WORDS
LARGEST ID ?LINE RMARGIN ?ENOUGH S DEPTH
CONTROL ASCII ? WHILE ELSE IF REPEAT AGAIN
UNTIL +LOOP LOOP ?? DO THEN BEGIN ?LEAVE
LEAVE ?<RESOLVE ?>RESOLVE <RESOLVE >RESOLVE <MARK
>MARK ?CONDITION BYE WARM COLD BOOT VERSION
FORTH-83 ABORT IS <IS> QUIT USER #USER
DEFINITIONS VOCABULARY LABEL CODE AVOC FORGET
#THREADS DEFER CRASH ZVARIABLE <CONSTANT VARIABLE
CONSTANT RECURSE ; DOES >DOES ;CODE ;USES
ASSEMBLER ] [ CREATE REVEAL HIDE >LINK >NAME
>BODY >LINK NAME BODY >NAME N>LINK ?CSP
!CSP ABORT" ?ERROR WHERE " [COMPILE] [ ]
ALLOT LITERAL LITERAL IMMEDIATE COMPILE C.
INTERPRET ?STACK STATUS DEFINED FIND HASH
< < >TYPE WORD PARSE PARSE-WORD SOURCE PLACE
>STRING SCAN SKIP D.R D. ok
ok
VOCS : EDITOR REASSEMBLER DECOMPILER ROOT USER ASSEMBLER I/O HID
DEN FORTH ok

```

## Download

- forth\_ac1.zip
- Forth, Docs, Zusatzpakete, Beispiele, picoAC1-GET/PUT  
Der Ordner forth wird beim picoAC1 auf eine SD-Karte kopiert.

## Kurzanleitung

- Start mit „f“
- 1 EDIT startet den Full-Screen Editor, Ende mit ^C
- 1 GET filename lädt eine Datei in die RAM-Disk
- 1 n THRU compiliert Screen 1 bis n
- Worte können mit SEE name decompiliert werden
- Vokabulare: VOCS und ORDER
- Dictionary: WORDS
- Ende mit BYE

## Beschreibung

Dateiname	FORTH83
Laden in	OS (mon3.1, ...)
Programmstandort	2000h-46FFh

	8000h-BFFFh RAM-Disk
OS-Kommando	f

Beschreibung des Forth s. [FG FORTH83 d. DDR](#)

- [Assembler](#)
- [CFORTH](#)
- [Debugger](#)
- [Decompiler](#)
- [Erinnerungen](#)
- [Fachgruppe FORTH](#)
- [Glossar](#)
- [Installation](#)
- [Installation Z1013](#)
- [Internes](#)
- [Kassettenarbeit](#)
- [Multitasking](#)
- [Screen-Druck](#)
- [Screeneditor](#)
- [Software](#)
- [Vokabulare](#)
- [Zeileneditor](#)

## picoAC1

Das Forth (incl. Varianten) und Programme werden als Z80-Dateien auf der SD-Karte abgelegt, z.B. in einem Ordner „forth“.

im Mon wird zuerst die Datei `getput_pico.Z80` laden (nach 8000h), dann `forth83`. Forth wird gestartet mit einer der drei Möglichkeiten:

```
f
J:
J 2000
```

In Forth werden die bereits in der RAM-Disk stehenden I/O-Routinen compiliert mit `1 6 THRU`. Ab jetzt stehen die Worte `.FILES`, `GET`, `PUT` zur Verfügung (s. [I/O](#)). Mit `.FILES` wird das aktuelle Verzeichnis gelistet. Mit `1 GET filename` lädt man eine Datei in die interne RAM-Disk. Nach dem Laden wird die Zahl der geladenen Screens angezeigt (`#scr n`). Mit `1 n THRU` erfolgt üblicherweise das Compilieren des geladenen Programms, es gibt aber auch Ausnahmen!

Gespeichert wird mit `1 n PUT filename`. Dieses Kommando speichert Screen 1 bis n auf SD-Karte unter dem Namen „filename“ im Z80-Format. Als Typ wird „F“ (für Forth-Datei) eingetragen. Ein Dateityp wird nicht automatisch an den Filenamen angehängt!

## Systemzusammenstellung

```
as_z80 forth83.asm
erzeugt forth83.z80, Adresse 2000-46FF

starten mit "f" (oder J 2000), Restart mit "r"
```

Fullscreen-Editor  
 Sedit\_AC1.Z80  
 extern laden (nach 8000), Forth starten  
 1 LOAD  
 Anwenden: n EDIT (editiert Screen n, n = 1..xx)

SEE - Dekompiler  
 SEE vp.z80  
 extern laden (nach 8000), Forth starten  
 1 10 THRU  
 Anwenden: SEE xxx

Datei-Arbeit  
 getput\_ac1pico.Z80  
 extern laden (nach 8000), Forth starten  
 1 6 THRU  
 Anwenden: .FILES (DIR)  
 n GET fname (Lädt File ab Screen n)

MARK EMPTY HERE FENCE !
SAVE-SYSTEM FORTH83A3
speichert das gesamte Forth incl. Pakete

## Beispiele

Die Beispiele können auch am originalen AC1 getestet werden. Dazu verlässt man Forth, lädt die Datei im Monitor-Modus nach 8000h, und geht dann zu Forth zurück (mit „f“).

Am picoAC1 werden die Dateien direkt unter Forth geladen, z.B. „1 GET LINED.Z80“. Groß-/Kleinschreibung des Dateinamens ist egal.

### **lined.z80**

#### [Zeileneditor](#)

```
1 6 THRU
1 EDIT      \ Screen 1 bearbeiten
...
DONE
```

## turmvonhanoi.z80

demonstriert, wie der „Turm von Hanoi“ optimal gelöst wird. n HANOI startet das Programm für n ( $\leq 5$ ) Scheiben.

```
1 5 THRU
3 HANOI      \für 3 Scheiben
```

```
Zug Nr: 0
#
#####
Turm 1    Turm 2    Turm 3
Zug Nr: 1

#####
#####
#
Turm 1    Turm 2    Turm 3
ging von Turm 1 nach Turm 3  Zug Nr: 2

#####
#####
#
Turm 1    Turm 2    Turm 3
ging von Turm 1 nach Turm 2 #
```

## unsinnstext.z80

nach L. Brodie, erzeugt witzige Texte

```
1 5 THRU      \ Screen 6..10 enthalten Textbausteine
TEXT          \ nach jedem Absatz ENTER drücken
NOCHMAL       \ gibt letzten erzeugten Text nochmal aus
```

```
TEXT

In dieser Arbeit werden wir zeigen, dass es unter Verwendung von repräsentativer moderner Interaktion, nicht zu verwechseln mit zufälliger warnender Kritik, möglich ist, in den meisten Fällen von funktioneller moderner Banalität eine Komplexität zu erreichen, wie sonst nur bei funktioneller Konkurrenz.

Einerseits haben Studien gezeigt, dass es unter Ausnutzung vertiebbarer Hilfsmittel, wie wahlreicher analoger Konzeption, in Verbindung mit paralleler digitaler Software, nicht durchführbar ist, mit integrierter einseitiger Hardware nur so zu scheinen, als wüsste man von systematischer dogmatischer Reserve.

Andererseits lehrt jedoch die Erfahrung, dass es mit strukturiertem Einsatz von partieller einseitiger Konkurrenz, in Verbindung mit zufälliger transitionaler Intelligenz, möglich ist, in den meisten Fällen von totaler Mobilität eine Komplexität zu erreichen, wie sonst nur bei qualifizierter warnender Flexibilität.

Zusammenfassend können wir sagen, dass es unter Verwendung von systematischer Kapazität, nicht zu verwechseln mit funktioneller moderner Reserve, nicht durchführbar ist, mit synchronisierter obligater Hardware nur so zu scheinen, als wüsste man von kompatibler Kapazität.

ok
```

## demorechner.z80

Großanzeige von Ziffern und Rechenzeichen. Enthält in Screen 1 und 2 Eaker-Case, nutzt AT vom Paket SEDIT (Screen-Editor).

```
1 8 THRU  
DEMO      \ erwartet korrekte Fortheingabe, z.B. 15 6 *
```



Zusammen mit formi2.f83 kann die Aufgabe auch in infix-Notation eingeben werden: { 12 \* 4 + 3 \* 2 }

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



Permanent link:  
<https://hc-ddr.hucki.net/wiki/doku.php/homecomputer/ac1/forth>

Last update: **2025/07/21 08:36**