

Informationen zum KK-FORTH

Version 1.2/0

- **Allgemeine Hinweise**
- **Kurzbeschreibung**
- **Kurzglossar**

**Klaus Kohl
Buchenweg 11
D-71299 Wimsheim
Tel. 07044/908789
Email: klaus.kohl@designin.de**

Allgemeines

Beim KK-FORTH oder kurz KKF handelt es sich um eine einheitliche FORTH-Version für diverse Prozessoren und Betriebssysteme. Es basiert auf dem FORTH83-Standard, hat aber viele Erweiterungen. Beispielsweise sind schon einige Befehle des geplanten ANSI-Standard implementiert. Darüber hinaus wurden viele einfache, aber wirkungsvolle Konzepte aus verschiedenen anderen FORTH-Versionen übernommen und erweitert.

Trotzdem bleibt es durch seine grundlegende 16Bit-Struktur und dem abgerundeten Befehlsumfang eine FORTH-Version, die sich gut zum Erlernen der Programmiersprache FORTH eignet. Deshalb ist auch ein Kapitel zur FORTH-Einführung im Handbuch vorhanden. Zusätzlich werden mit dem KKF viele Sourcen wie Bildschirmeditor, Assembler oder Beispiele zur Ansteuerung der vorhandenen Hardware mitgeliefert.

Bei den KKF-Versionen für Einplatinencomputer (z.B. 84C015- oder PC-EMUF mit serieller Schnittstelle) übernimmt ein Terminalprogramm auf dem PC die Verwaltung der Files und das Editieren der Programme.

Da die einzelnen KKF-Versionen bis auf den rechnerspezifischen Teil gleich sind, gibt es ein großes Handbuch mit FORTH-Einführung, Beschreibung des KK-FORTH und dem ausführlichen Glossar. Für jede Version ist dann ein Handbuchzusatz mit spezifischen Informationen verfügbar. Darin werden die einzelnen Files beschrieben und weitere Angaben zur Implementation (z.B. Befehlsaufbau und Registerbelegung) geliefert.

Features des KKF

- Minimale Hardwarevoraussetzung (ca. 16KByte EPROM, 8KByte RAM, Schnittstelle für Programmentwicklung)
- Weitgehende Kompatibilität mit dem FORTH83-Standard
- Viele Befehlserweiterung aus volksFORTH und ANSI-Vorschlag
- Durch viele Assemblerbefehle sehr schnell
- Durch Auslagerung der Variablen und Vektoren auch ROM-Fähig
- Das Fileinterface und alle Ein-/Ausgabe sind vektorisiert
- Viele Systemwörter können durch eigene Befehle ersetzt werden
- Fehlermeldungen als Text; Fehlerbehandlung umleitbar
- Komfortabler Zeileneditor für die Eingabe (speichert die letzten 8 Zeilen)
- Autostart-Applikationen sind möglich
- Selbst auf EMUF's kann das Fileinterface verwendet werden (PC-Terminal)
- Viele Tools wie Assembler und Disassembler werden mitgeliefert
- System für Multitasking und Interruptverarbeitung vorbereitet
- Verfügbar für die unterschiedlichsten Systeme und Prozessoren
 - IBM-PC oder Kompatible (ab 8088-Prozessor und DOS 2.11)
 - PC-EMUF (Terminal über SIO1)
 - 84C015-EMUF (Terminal über SIO2)
 - RTX-2001a (FG-Board oder mc-RISC-EMUF)
 - 68332-Board
- Keine Lizenzgebühren bei privaten Applikationen

Es sind auch Anpassungen der vorhandener KKF-Versionen an ihre spezielle Hardware möglich. Dies ist aber im Einzelfall zu klären. Bitte richten Sie entsprechende Anfragen an die auf der ersten Seite angegebene Adresse.

Für einige Controller existieren auch Target-Compiler z.B. für headerlose Compilierung größerer Programme teilweise in Modultechnik (z.B. beim 84C015).

KKF_PC

Das KK-FORTH für den IBM-PC ist ein COM-Programm, daß vollständig an das Betriebssystem angepaßt ist. Alle Ein-/Ausgaben werden über das Standardgerät abgewickelt und können deshalb vom Betriebssystem umgeleitet werden. Das Fileinterface erlaubt den Zugriff auf das aktuelle Verzeichnis. Zusätzlich ladbare Befehle dienen dem Aufruf externer DOS-Befehle oder sogar eines DOS-Shell's. Der ebenfalls direkt zum FORTH ladbare Editor ist sehr kompakt und trotzdem so schnell, daß er sogar auf einem langsamen PC-Laptop gut verwendbar ist. Weitere Features im Überblick:

- Minimale Systemvoraussetzung:
 - 8086/88-Prozessor mit mindestens 64KByte freies RAM
 - MSDOS ab Version 2.11
- Zusätzliche Befehle des KKF-Kerns:
 - Reservieren und Ansprechen des externen RAM's
- Viele Zusatzprogramme werden mitgeliefert:
 - 8086-Assembler
 - 8086-Disassembler
 - Debugger für KKF_PC
 - Zusatzbefehle für DOS-Aufrufe
 - Ladbarer Editor (und ein Editor als eigenständiges Programm)
 - Ladbares Terminalprogramm (für die EMUF's)
 - Druckertreiber für NECP6 (zum Erstellen der Listings)
 - Befehle für Bildschirmsteuerung (BIOS und direkte Ansteuerung)

KKF_V20

Für den V20-EMUF werden auf einer Diskette mehrere Versionen ausgeliefert. Je nach verfügbarem Speicher oder verwendetem Betriebssystem kann entweder die Minimalversion oder die DOS-Version verwendet werden. Wie beim KKF_PC werden Assembler, Disassembler und zusätzliche Tools mitgeliefert.

Die Minimalversion benötigt ein 32KByte-EPROM am Speicherende des V20 und ein 32KByte RAM am Speicheranfang. Da vom EPROM nur 17KByte verwendet werden, bleibt genügend Raum für weitere Befehle des Anwenders. Die Ausgabe erfolgt auf der ersten seriellen Schnittstelle. Das Fileinterface wird ebenfalls über diese Schnittstelle transparent für den Anwender mit dem mitgelieferten Terminalprogramm abgewickelt. Da das Programm zum Teil im EPROM läuft, muß für die Verwendung des Debuggers ein anderes, etwas langsames Bitimage gebrannt werden.

Die DOS-Version entspricht dem auf einem PC lauffähigen Programm. Es erwartet wie bei einem COM-Programm, daß es auf Adresse \$0100 eines RAM-Bereiches mit mindestens 64KByte kopiert wird. Die Fileverwaltung wird aber wie in der Minimalversion über das Terminal abgewickelt.

Die PC-Version entspricht der schon beim KKF_PC beschriebenen Programm, daß sowohl Ein-/Ausgabe als auch das Fileinterface über das Betriebssystem abwickelt.

KKF_8415

Die KKF-Version für den 84C015-EMUF wird ebenfalls auf einer PC-lesbaren 360K-Diskette ausgeliefert. Auf der Diskette ist ein Binärfile, daß in ein 16K- oder 32K-EPROM gebrannt und an Stelle des Betriebssystem-EPROM's in die Platine gesteckt werden muß. Nach dem Einschalten meldet sich das FORTH (wie das Monitorprogramm) mit 9600 Baud an der gleichen Schnittstelle. Es kann jedes beliebige Terminalprogramm zum Arbeiten mit dem KKF verwendet werden. Jedoch können erst mit dem ebenfalls mitgelieferten Terminalprogramm die File- und Editor-Befehle genutzt werden. Da aber sowohl das Terminalprogramm als auch die dazu notwendige KKF-Version (für PC) mitgeliefert wird, kann das Terminalprogramm den eigenen Wünschen angepaßt werden.

Neben dem Binärfile werden weitere Tools mitgeliefert. Nach dem Erstellen des vollständigen Programmes kann mit `SAVESYSTEM <Filename>` ein Image der zusätzlichen Befehle zum Terminal geschickt werden. Dieses Image kann dann ebenfalls in das EPROM übernommen oder in ein EEPROM gebrannt werden. Danach sind dann entweder weitere Befehle beim Einschalten verfügbar oder ein vorher angegebener Befehl wird unmittelbar ausgeführt.

- Systemvoraussetzung:
 - 16KByte EPROM (evtl. 32KByte für zusätzliche Befehle)
 - mindestens 8KByte RAM (System für 32KByte RAM vorbereitet)
 - Serielle Schnittstelle (nur für Programmierung notwendig)
- Viele Zusatzprogramme werden mitgeliefert:
 - Terminalprogramm mit allen notwendigen Quellen und Programmen
 - Z80-Assembler
 - Z80-Disassembler
 - Debugger
 - Beispielprogramme für die 84C015-Hardware

KKF_FG

Diese FORTH-Version ist vom Befehlsumfang und der Anwendung identisch mit den anderen KKF-Versionen. Leider ist es deshalb nicht kompatibel zu dem vorher ausgelieferten FG-FORTH. Jedoch dürfte der Aufwand zur Anpassung minimal sein. Da die Multiplikationsroutine für den RTX-2001A angepaßt wurde und die Stacks prozessorspezifisch behandelt werden, kann ohne Änderung des Programmes der teurere RTX-2000 durch den RTX-2001A ersetzt werden.

- Systemvoraussetzung:
 - Wahlweise RTX-2000 oder RTX-2001A (wird beim Start erkannt)
 - 2*8KByte BOOT-EPROM (evtl. 32KByte für zusätzliche Befehle)
 - mindestens 64KByte RAM (Bank 0 vollständig bestückt)
 - Serielle Schnittstelle über G-Port 7 (Bit 0 und 1)
- Viele Zusatzprogramme werden mitgeliefert:
 - Terminalprogramm mit allen notwendigen Quellen und Programmen
 - UCode-Definitionen, optimierender Kompiler, Dekompiler
 - Beispiel für Interrupt-Verwendung

KKF-Befehlsübersicht

Konstanten

TRUE	FALSE	#ESC	
#BL	#DEL	#CR	#BRK
#B/BLK	#L/BLK	#C/L	#TIB-MAX

Systemadressen und -variablen

SYSCON	SYSVAR	(SYSVAR	SYSVARLEN@
HERE	VHERE	HEAP	HEAP?
TASKADDR@	TASK0	FIRST	LIMIT
VOC-LINK	DP	VDP	VLEN
HDP	HLEN	TDP	TASKADDR
TLEN	TASK-LINK	TASKS	MAXTLEN@
SFLAG	UFLAG		

USER-Variable und zugehörige Befehle

WDP@	PAD	S0	R0
STATE	CURRENT	CONTEXT	
BLK	>IN	SCR	R#
SPAN	>TIB	TIB	
#TIB	BASE	HLD	DPL
INPUT	OUTPUT	DISC	ERRORHANDLER

Variable

FILE-ID	FILE-FCB	FILE-LINK	INDENT?
NEXT-LINK	CODE?	LAST	>HEAP?
CAPS			

Datenstack- und Returnstackbefehle

SP@	SP!	DEPTH	
'DCLEAR	DCLEAR		
ROLL	SWAP	ROT	-ROT
	2SWAP	2ROT	-2ROT
PICK	DUP	OVER	TUCK
	2DUP	2OVER	2TUCK
DROP	NIP	2DROP	2NIP
?DUP	S>D	D>S	
RP@	RP!	RDEPTH	
'RCLEAR	RCLEAR		
>R	R>	R@	RDROP
2>R	2R>	2R@	2RDROP
PUSH			

Arithmetik, Logik und Vergleiche

+	-	NEGATE	
?NEGATE	ABS		
2-	1-	1+	2+
AND	OR	XOR	NOT
ASHIFT	2*	2/	
SHIFT	U2/		
FLIP			
0<	0=	0<>	0>
<	=	<>	>
U<	U>	WITHIN	CASE?
MIN	MAX	UMIN	UMAX

D+	D-	DNEGATE	
?DNEGATE	DABS		
D2*	D2/	DU2/	
D0<	D0=	D0<>	D0>
D<	D=	D<>	D>
DU<	DU>	DWITHIN	
DMIN	DMAX	DUMIN	DUMAX
M+	M-		
UM*	UM/MOD		
M*	*	M/	/
M/MOD	/MOD	MOD	
*/MOD	*/		

Speicherzugriffe

CHAR+	CHARS	CELL+	CELLS
ALIGNED	>NEXTTASK	>VADDR	
C@	C!	@	!
2@	2!		
CMOVE	CMOVE>	MOVE	
FILL	ERASE	BLANK	
+!	-!	ON	OFF
SKIP	SCAN	SKIP>	SCAN>
-TRAILING	/STRING	/\$	
COUNT	COUNT>0		
>\$			
UPC	UPPER		
PC@	PC!	P@	P!
PTR>D	D>PTR		
CS@	DS@	SS@	

Zugriff auf externen Speicher

(MALLOC	(MRELOC	(MFREE	
LC@	LC!	L@	L!
L2@	L2!		
LCMOVE	LCMOVE>	LMOVE	
LFILL	LWFILL		
LDUMP			

Ein-/Ausgabebefehle

STANDARD-IO	(INPUT	(OUTPUT	(DISC
EMIT?	EMIT	TYPE	
CR	DEL	BELL	
CLS	MAXAT	AT	AT?
SPACE	SPACES	INDENT	
-TYPE	.ID		
KEY?	KEY	STRING	
EDITSTRING	QUERY	EXPECT	
STOP?			
BINARY	DECIMAL	HEX	
>NUMBER	NUMBER	CONVERT	NUMBER?
<#	HOLD	#	#S
SIGN	#>		
D.R	D.		
U.R	0U.R	U.	
.R	.?		
?	U?	D?	

.STATE	.BLK	
.S	DUMP	
ORDER	VOCS	WORDS
FILES	FILE.	
LIST	IDENT	
-IDENT		

Kontrollstrukturen

EXECUTE	PERFORM		
EXIT	?EXIT	0=EXIT	
BRANCH	?BRANCH	NEXTBRANCH	
>MARK	>RESOLVE	<MARK	<RESOLVE
RECURSE			
IF	ELSE	THEN	
BEGIN	WHILE	UNTIL	REPEAT
?DO	DO	LOOP	+LOOP
BOUNDS			
I	I'	J	J'
LEAVE	?LEAVE	UNLOOP	
FOR	NEXT		
CASE	OF	ENDOF	ENDCASE

Definition und Dictionary

ALLOT	ALIGN		
C,	,A	,C	
LITERAL	2LITERAL	ASCII	NUMBER,
COMPILE	[COMPILE]	POSTPONE	
-\$,	\$,	>\$,	
"	"		
VALLOT	VALIGN	VC,	V,
HALLOT	HALIGN	HC,	H,
EVALUATE	INTERPRET	PARSER	
[]		
DEFINITIONS	ALSO	SEAL	
ONLYFORTH	FORTH		
-WORD	WORD		
FIND	(FIND	'	[']
L>NAME	N>LINK		
LINK>	BODY>	NAME>	
>LINK	>BODY	>NAME	
FORGET	(FORGET	REMOVE	'REMOVE
HCLEAR	SAVE	EMPTY	
(.(
\	\	\NEEDS	
IMMEDIATE	RESTRICT	INDIRECT	
HIDE	REVEAL		
-HEADERS		HEADERS	
:	;		
CREATE	DOES>	VCREATE	VDOES>
VARIABLE	2VARIABLE	CONSTANT	2CONSTANT
VOCABULARY	USER	ALIAS	
>LABEL	LABEL	DEFER	IS
UVECTOR	UTABLE:	(;CODE	

Fileverwaltung

MAKE	DELETE	
OPEN	(OPEN	CLOSE
MORE	CAPACITY	
BLOCK	BUFFER	UPDATE
SAVE-BUFFERS	EMPTY-BUFFERS	FLUSH
LOAD	THRU	-->
INCLUDE	LOADFROM	
SAVESYSTEM		

Direkte Filebefehle

(FILE?	(FILE-CREATE	(FILE-DELETE	
(FILE-OPEN	(FILE-CLOSE	(FILE-SIZE	
(FILE-POS!	(FILE-POS@	(FILE-READ	(FILE-WRITE
(FILE-FREE	(FILE-FIRST	(FILE-NEXT	

Multitasker

PAUSE	SINGLETASK	MULTITASK
-------	------------	-----------

Fehlerbehandlung

ERROR	?ERROR		
?STACK	?DEPTH	?ALLOT	?PAIRS
?OPEN	?BLK	?NAME	
NOTFOUND	NODEFER		
ABORT"	ERROR"		
>ERRORTXT	ERRORTXT@	(ERRORHANDLER	' ERROR

Sonstige Befehle

NOOP	QUIT		
BOOT	' BOOT	COLD	' COLD
ABORT	' ABORT	BYE	' BYE