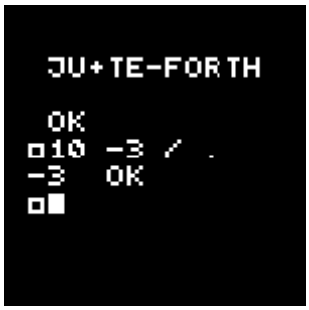
FORTH

- tiny_forth.zip

JU+TE-FORTH für 4K ES1988 und 6K ES4.0, incl. Assembler-Quellcodes

- forth-83.pdf (allg.) FORTH-Lehrgang aus Jugend+Technik, von Dr. H. Hoyer
- spezifische Beschreibung der Implementation s. Systembeschreibung (Auszug aus JU+TE 01/1990)





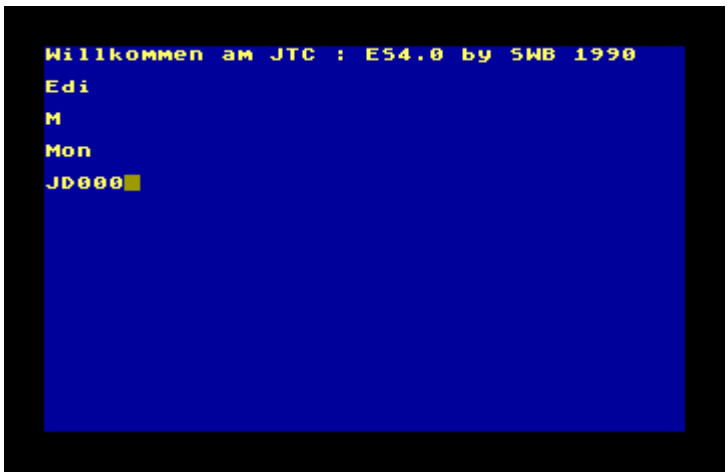
Das JU+TE-FORTH-System setzt das 4K-System ES1988 sowie 8 Kbyte RAM voraus. Das FORTH-System wird von Kassette geladen und läßt sich ab %E000 über PROG oder BASIC starten.

Basic: einfach mit RUN (+) starten (Basic-Programm CALL %E00A ist im Forth enthalten)
PROG: GE000, L

- Start auf %E000
- ab %FC00 Magnetbandpuffer, bis dahin frei für Prg. (ca. 2KByte)
- Speichern von %E000 bis DP, am einfachsten bis %FBFF
- 173 Systemworte

Beim 6K-ES4.0 wird Forth auf Adresse %D000 geladen. Der Start erfolgt mit

```
M
Mon
JD000
```





Das FORTH für ES4.0 ist eine gepatchte Version des FORTH für ES1988. Die Tastaturabfragen erfolgen anders, z.B. haben QUERY und EXPECT neue Implementierungen. Dafür musste VIEW weichen.

Weitere Hinweise s.es40.

Neue Version Ich habe die Version fürs ES4.0 überarbeitet. Nun ist ein VIEW mit drin (Im Unterschied zum ES1988-VIEW wird VIEW beim Erreichen von CODE beendet, statt in den Monitor zu springen). Außerdem ist diese Version kürzer als das Original!

Systembeschreibung

aus JU+TE 01/1990, Seiten 70-73

FORTH-83-System für JU+TE-Selbstbau-Computer

Mit der leistungsstarken Programmiersprache FORTH-83 erhält der JU+TE-Computer neue Anwendungsmöglichkeiten. Mit diesem Beitrag wird ein von Kassette ladbares FORTH-System beschrieben. Nur für Leser, die diese Sprache bereits kennen, ist es allerdings in allen Einzelheiten sofort verständlich. Diejenigen, die zum Beispiel mit unserem ABC den Einstieg beginnen (vgl. dazu S. 67 ff. in diesem Heft), sollten diese Artikel in Griffnähe behalten, um sich gleichzeitig mit FORTH-83 und der Handhabung des JU+TE-Systems vertraut zu machen. Die Redaktion vermittelt übrigens gern das kostenlose Überspielen des JU+TE-FORTH auf eingesandte Magnetbandkassetten (möglichst CrO2; Rückporto in Briefmarken bitte nicht vergessen).

Wichtige FORTH-Worte außerhalb des Standards

VIEW (–) Anzeige einer Wortdefinition: VIEW (name)
LINK (adr –) Fortsetzung des Wörterbuchs ab Adresse adr.
INPUT (– d) Zahleneingabe mit doppelt genauem Ergebnis. Die Variable DPL wird aktualisiert.
BUSIN (adr – n) Lesen des Registers mit der Adresse adr (untere acht Bit) in die unteren acht Bit von n.
BUSOUT (n adr –) Laden des Registers mit der Adresse adr (untere acht Bit) mit unteren acht Bit von n.

Konzeption

Der JUGEND+TECHNIK-Selbstbau-Computer ist durch geringen Hardwareaufwand gekennzeichnet und bietet keinen großen Spielraum für höhere Programmiersprachen. Er eignet sich eher für

Steuerungsaufgaben als zur Textverarbeitung oder Datenbankverwaltung. Unter diesen Bedingungen ist FORTH die geeignetste Hochsprache, da sie sowohl maschinennahe Programmierung als auch allgemeine Formulierungen unterstützt. Sehr willkommen sind natürlich weitere Vorzüge wie die gute Ausschöpfung der Rechenleistung zur Ausführung von Anwenderprogrammen. Das JU+TE-FORTH System setzt 8 Kbyte RAM (voll bestückter Speichermodul auf Platz 1) voraus. Die ersten fünf Kilobyte benötigt der FORTH- Kern, für den Anwender bleiben die restlichen beiden Kilo bis %FC00, wo der Magnetbandpuffer beginnt. Da FORTH deutlich weniger Speicherkapazität als BASIC in Anspruch nimmt, bieten 2 Kbyte Anwender-RAM sehr viel Platz. Das FORTH-System wird von Kassette geladen und läßt sich ab %E000 über PROG oder BASIC starten. Beim Sichern (SAVE) von Anwenderprogrammen muß der Kern mitgenommen werden, da hier z. B. die Systemvariable DP, die die Endadresse speichert, enthalten ist. Am einfachsten ist es, stets den RAM Bereich von %E000 bis %FBFF auf Kassette zu speichern.

Das JU+TE-FORTH-System ist ein FORTH-83- Standard-Subset mit Standarderweiterungen für doppelt genaue Zahlen. Von den 132 Worten des sogenannten notwendigen Wortschatzes fehlen die für die Screenarbeit (Quelltextspeicherung) und für die Teilung des Wörterbuchs in Vokabulare nötigen sowie die zwanzig in Abb. 1 aufgeführten. Diese werden zum größten Teil implizit vom System realisiert, lassen sich aber nicht völlig standardgerecht in Anwenderprogramme einbauen.

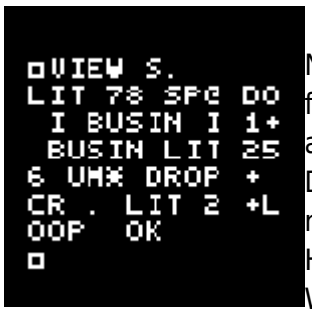
Abb 1:

```
‘ , -TRAILING .( >BODY ABORT
ABORT" COMPILE CONVERT CREATE
DOES> EXECUTE IMMEDIATE LITERAL
PAD SPAN [ ['] [COMPILE] ]
```

Besonderheiten

Auf die Screenarbeit wurde verzichtet, da mit der Bildschirmgröße und RAM- Austattung des JU+TE-Computers zu enge Grenzen gesetzt sind. Statt dessen können mit dem Wort VIEW alle Wörterbucheinträge rückübersetzt werden.

Die Vorsätze LIT und DLIT kennzeichnen dabei mit einfachen (16 bit) und mit doppelter (32 bit) Genauigkeit gespeicherte Zahlen. Sonst erzeugt VIEW die gleiche Anzeige, die für die Eingabe von Doppelpunktdefinitionen nötig ist. Maschinenprogramme kennzeichnet VIEW mit CODE. Anschließendes Betätigen der ENTER-Taste bringt den Maschinencode im DATA-Modus, der mit OFF wieder verlassen werden kann, zur Anzeige. Jede andere Taste unterdrückt diese Ausgabe.



Die Übersicht über das Wörterbuch ist damit ganz einfach: WORDS bringt die Namen (jeweils sieben untereinander) zur Anzeige, wobei mit der ENTER-Taste fortgesetzt und mit jeder anderen abgebrochen werden kann. VIEW mit anschließender Namenseingabe zeigt die Wortdefinition auf dem Bildschirm. Damit kann der Quelltext mit Papier und Bleistift leicht verwaltet und sogar nachträglich erzeugt werden. FORTH ist flexibel genug, um auch mit minimaler Hardware auszukommen. Das JU+TE-FORTH-System speichert alles im Wörterbuch (inline) und benötigt daher keine weiteren Dateien. Beim Übersetzen von Eingaben entsteht direkt das entsprechende Maschinenprogramm, was sogar den sonst üblichen Abstand zwischen der Spitze des Wörterbuchs und dem Pufferspeicher PAD überflüssig macht. Das erste Byte eines Wörterbucheintrags enthält nur den Count, die Kennzeichnung als normales (%00) oder als Immediate Wort (%01) folgt als zusätzliches Byte dem Linkfeld. Es wurde zum Kennzeichnen von Strukturwörtern (z. B. DO) und zur Steuerung des Rückübersetzens eingeführt. Der Rumpf ist ein ausführbares Maschinen-Unterprogramm, das sowohl Codefeld als auch

Parameterfeld realisiert (Abb. 2). Das Wort >BODY ist damit ohne Sinn. Zum sofortigen Ausführen wird die Eingabe in einen solchen Rumpf an die Spitze des Wörterbuchs übersetzt und anschließend gerufen. Wenn sich das Wörterbuch nach vielen Erweiterungen einmal der oberen Anwender-RAM-Grenze %FBFF gefährlich nähert, kann mit dem Wort LINK die Fortsetzung an beliebiger anderer Stelle (auf einem weiteren Speichermodul) ausgelöst werden. LINK nimmt die Fortsetzungsadresse vom Stack. Die aktuelle Endadresse des Wörterbuchs läßt sich mit Hilfe von HERE jederzeit anzeigen (Abb. 3).

Abb 2:
<ditaa noedgesep> +————————-+

Countbyte 100XXXXX

+————————-+

Name ASCII

+————————-+

Linkfeld (High, Low)

+————————-+

Kennbyte

+————————-+

Maschinencode
(Rumpf, = CFA = PFA)

+————————-+ </ditaa>

Abb 3: Anzeige der Wörterbuchgrenze:

```
HEX HERE U. DECIMAL
```

Für die Zahleneingabe gelten allgemein übliche Regeln: Die Ziffern werden mit Hilfe der aktuellen Zahlenbasis (Variable BASE) in eine einfach genaue Dualzahl gewandelt. Enthält das Zahlwort jedoch einen Dezimalpunkt, erfolgt die Weiterverarbeitung mit doppelter Genauigkeit. Die Variable DPL erhält die Anzahl der rechts vom Dezimalpunkt stehenden Ziffern (DPL := -1, wenn kein Punkt). Ein Minuszeichen im Zahlwort bewirkt das Negieren des gewandelten Wertes. Die in Standardprogrammen übliche Zahleneingabe mit SPAN und CONVERT läßt sich mit dem JU+TE-FORTH-Kern nicht standardgerecht ausführen. Die Schwierigkeiten resultieren aus der Nutzung des Bildwiederholspeichers als Eingabepuffer. Statt dessen gibt es hier das Wort INPUT, das eine Zahleneingabe mit ENTER als Abschluß fordert und als doppelt genaue Zahl auf dem Stack hinterläßt.

FORTH arbeitet mit Speicheradressen. Zum Standard gehören keine Worte, die sich auf Register beziehen. Passend zum FIG-FORTH für die Kleincomputer KC 85 enthalt das JU+TE-FORTH-System die Wörter BUSIN und BUSOUT, die wie @ und ! wirken, aber mit Register- statt Speicheradressen arbeiten (KC: E/A-Adressen). Da sich der Stack in den Registern %20 bis %4F des Einchip-Mikrorechners befindet, muß auch BUSIN statt @ für die Anzeige des Stackinhalts verwendet werden. Abb. 4 gibt zwei dafür gut geeignete Definitionen. Das Zeichen @ wird übrigens mit SHIFT P

eingegeben. Das Reservieren von Freibereich im Wörterbuch läßt sich allgemein mit n ALLOT (ENTER) ausführen. Bei n = 9 und n = 10 muß beim JU+TE-FORTH nach n die ENTER-Taste zusätzlich betätigt werden, damit sich die compilierte Eingabe und das zu verschiebende Linkfeld nicht gegenseitig stören. Das hängt damit zusammen, daß als Eintritt für die Suche nach Wortdefinitionen an der Spitze des Wörterbuchs (HERE) einfach die Adresse des zuletzt eingetragenen Wortnamens (Linkfeld) steht.

Abb 4: Anzeige des Stackinhalts:

```
: S. 78 SP@ DO I
  BUSIN I 1+ BUSIN
  256 UM* DROP + CR
  2 +LOOP ;
: STACK DEPTH 0> IF
  S. ELSE ." LEER"
  THEN ;
```

FORTH lebt grundsätzlich vom Erweitern des Wörterbuchs. Fehlende Standardworte sind kein großes Problem, da sie sich grundsätzlich nachtragen lassen. (Das Überführen von TINY-BASIC in extended BASIC ist dagegen z. B. völlig unmöglich.) Die in Abb. 1 aufgezählten Wörter werden zumindest im JU+TE-FORTH-System kaum benötigt (vgl. INPUT für SPAN und CONVERT). Am ehesten wird CREATE vermißt. Es läßt sich mit VARIABLE (name) -2 ALLOT ersetzen. Das in Verbindung mit CREATE interessante Komma kann, wie in Abb. 5 dargestellt, nachgetragen werden. Das Wort J, das bei verschachtelten DO ... LOOP-Strukturen den Index der nächstäußeren Schleife zum Stack kopiert, setzt voraus, daß die innere Schleife als Wort definiert wurde und so nur mit dem Namen in der äußeren erscheint. Das trägt den Stilforderungen Rechnung, die nur eine Struktur wie DO ... LOOP in einer Doppelpunktdefinition zulassen.

Abb 5: Eintragen von TOS ins Wörterbuch (n –):

```
  : , HERE 2 ALLOT ! ;
```

Maschinenprogramme

Das Wort CODE gestattet Wortdefinitionen in Maschinensprache. Nach Eingeben des Wortnamens und Betätigen der ENTER Taste ruft das JU+TE-FORTH-System das Betriebsprogramm DATA mit dem ersten Byte des Rumpfes auf der Eingabeadresse. Ab dort kann das Maschinenprogramm eingetragen werden. Der Registerpointer steht beim Rufen von Maschinenprogrammen auf %10. Die Arbeitsregister r2 und r3 speichern den TOS. Zum Rechnen stehen r4 bis r11 zur Verfügung. Auch die Register %02 bis %0F (einschließlich der Ports) dürfen benutzt werden. Beim Zugriff auf den Bereich ab %50 ist das Betriebssystem zu berücksichtigen. Die Register %20 bis %4F bleiben für den Stack reserviert und damit tabu. Stackpointer ist das Arbeitsregister r15. Diese Aufteilung gestattet das Verwenden der allgemein nutzbaren Betriebs-Unterprogramme (vgl. JU+TE 5/89 S. 380) auch für das FORTH-System.

Nach Eingabe des letzten Bytes des Maschinen-Unterprogramms (meist AF) wird die Definition mit OFF abgeschlossen. Dabei verwendet das FORTH-System die aktuelle Eingabeadresse als HERE (Adresse des ersten freien Bytes). Es ist daher nicht gleichgültig, in welchem Zustand der DATA-Modes verlassen wird. Soll die automatische Adreßberechnung (Binden) zum Einsatz kommen, müssen die betreffenden Operandenbytes bei der Eingabe zunächst frei bleiben (z.B. %FF). Nach Eingabe dieses unfertigen Programms unter FORTH kann es mit VIEW angezeigt werden. Über RESET und PROG ist das Nachtragen der Marken und schließlich das Binden nötig. Nach dem Neustart des

FORTH-Systems steht das neue Wort dann zur Verfügung. Die Adresse der Markentabelle sollte bei diesem Verfahren vor Beginn der Definition des Maschinenprogramms bereits vereinbart sein, damit PROG gleich zur richtigen Adresse führt.

Anwendung

Das JU+TE-FORTH-System besitzt eine hohe Ausführungsgeschwindigkeit. Im Vergleich zu anderen FORTH-Systemen dauert jedoch die DO-LOOP-Schleife länger, da das erforderliche Verwalten von Index und Grenze im Return-Stack beim Einchip-Mikrorechner größere Umstände macht. Das Übersetzen von Eingaben erledigt ein Interpreter. Die hier spürbaren Laufzeiten beeinträchtigen aber nicht die Geschwindigkeit beim Ausführen auf diese Weise erzeugter Maschinenprogramme (im Rumpf).

Die Übertragbarkeit von FORTH-Programmen auf andere Rechner setzt voraus, daß in den Definitionen ausschließlich der Standardwortschatz verwendet wird. Maschinenprogramme, die mit CODE erfaßt wurden, zählen grundsätzlich nicht dazu. Übertragbar ist stets der Quelltext, nicht jedoch das auf Kassette speicherbare Wörterbuch. Das hat auf anderen Rechnertypen mit Sicherheit Abweichungen, auch wenn das gleiche FORTH-Programm kompiliert wurde.

Viel Spaß beim Programmieren in FORTH!

Dr. Helmut Hoyer

HINWEIS

JU+TE 01/1990, Seiten 70-73

Zunächst noch ein Hinweis zum FORTH-System, das auf an JU+TE eingesandte Magnetbandkassetten geladen wird: Dieses Programm nutzt den RAM-Modus des 4-K-Betriebssystems zur Anzeige und zur Eingabe von Maschinenprogrammen in FORTH-Wörtern. Befindet sich das Betriebssystem in einem U 2732 C oder U 2764 C, gilt die Startadresse %1352. Ist das Betriebssystem in zwei U 2716 C geteilt, beginnt der RAM-Modus jedoch auf %2352. In diesem Fall müssen im FORTH-System die Bytes auf den Adressen %EBEB und %EDD3 von %13 auf %23 geändert werden, damit die Anwendung des Wortes CODE klappt. Für diese Änderung eignet sich der RAM-Modus, nach Laden des FORTH-Systems aus dem Resetmenü gestartet, am besten.

Einstieg

Als kleines „Programm“ zum Einstieg kann man die Malfolge nehmen

```
: MALFOLGE 11 1 DO DUP I * . LOOP DROP ;
```

Damit hat man ein Programm (eine neues FORTH-Wort) „MALFOLGE“ definiert. Mit einer Zahl davor wird die Malfolge 1*n .. 10*n ausgegeben:

```
3 MALFOLGE
9 MALFOLGE
-2 MALFOLGE
111 MALFOLGE
```

Als weiteren Einstieg kann ich das auch Handbuch zum KC85-Forth empfehlen. Es ist sehr gut geschrieben. Über weite Teile kann man das auch für den TINY nehmen, z.B. „Elefanten füttern“!
http://www.mpm-kc85.de/dokupack/M026_FORTH.pdf

ES4.0

- angepasstes FORTH, Laden mit LD000
- Starten mit JD000, entspricht im wesentlichen dem FORTH des 4K-Systems, wurde aber auf %D000 verschoben
- Befehl VIEW existiert nicht
- Befehl CODE ruft den MON auf, in Register %20/21 steht, wo man mit Eintragen des Codes beginnen kann, vor Rueckkehr mit Q in %20/21 die Adresse hinter dem Code eintragen
- Nutzt den 80-Zeichen-Modus des EDI
- Vor einer Eingabezeile braucht kein so merkwuerdiges Zeichen stehen wie bei 4K-Version
- Letzte Position der Eingabezeile wird normalerweise nicht genutzt, wenn dort aber ein \ Zeichen steht, wird noch die Eingabe der naechsten Zeile abge wartet, bevor sie verarbeitet wird. Das kann man bis zu 2 mal machen, beim 3. mal erscheint \#MSG als Fehler also max. 237 Zeichen pro Zeile
- %F300-F3FF wird als Zeilenpuffer genutzt.

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

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/tiny/software/forth?rev=1705992427

Last update: **2024/01/23 06:47**