

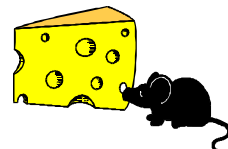
# VIERTE DIMENSION

3/1995

11. Jahrgang 1995, 3. Quartal, DM 10.-



- ultraFORTH83 rev3.82
- Meßtechnik mit dem PC. Teil II (Timer)
- DOER & MAKE
- FORTH optimieren
- fLux - Brief aus der Diaspora
- Freiburg-Zurich-Nashville-Story
- Forth-Computer, ICR, ...



# FORTH MAGAZIN

Organ der Forth Gesellschaft e.V.

## FORTH - Shirt



**T - Shirt:** hellgrau / grün  
in Größe: M-L-XL **25 DM**

**Sweat-Shirt:** grau / grün  
in Größe: M-L-XL **40 DM**  
(+ Porto)

### ForthWORKS

Ulrike Schnitter  
Nelkenstr. 52  
85716 Unterschleißheim  
Tel.: 089-310 33 85

## In-Circuit Emulation für die 68HC11E Serie

- besonders für 68HCE11 Singlechiplösungen
- geringe Abmessungen, kurze Signalleitungen
- direkt in eine 68HC711E9-Fassung einsteckbar
- alle 40 IO-Leitungen des 68HC711E9 verfügbar
- Hintergrunddebugg über 2. UART in Echtzeit
- hohe Debugdatenrate mit 57k Baud
- 32kByte Emulationsspeicher
- Entwicklungsumgebung MONI11E, TCOM6811
- kostengünstig ab **599,- DM**

68HC11F1-Board, komplett	299,- DM
Debugger MONI11A1 (f. A,E,L-Typ), MONI11F1 oder MONI11K4 (f. K,N-Typ)	149,- DM
Handbuch zu MONI11.. und TCOM6811	49,- DM

Die Software ist auf PC/XT/AT lauffähig. Zu der Software wird der Forth-Compiler TCOM6811(Library v. H. Dyja) mitgeliefert.  
Alle Preise incl. Mwst. ohne Porto und Verpackung.

**Holger Dyja - Naumannstr. 13 - 10829 Berlin**  
Tel./Fax. 030 / 784 12 57

## Dienstleistungen und Produkte von Forthlern und/oder für Forthler (Anzeige)

### Ingenieurbüro Dipl.-Ing. Wolfgang Allinger

Tel. (+Fax.) 0+212-66811  
Brander Weg 6  
D-42699 Solingen

Entwicklung von µC, HW+SW, Embedded Controller, Echtzeitsysteme 1 bis 60 Computer, Forth+Assembler PC / 8031 / 80C166 / RTX2000 / Z80 ... für extreme Einsatzbedingungen in Walzwerken, KKW, Medizin, Verkehr / >20 Jahre Erfahrung.

### ETA Elektrotechnische Apparate GmbH

Tel.: 0+9187 -10.0 (Fax: -10.397)  
Industriestr. 2-8  
D-90518 Altdorf (b. Nürnberg)

Produkte für Echtzeitanwendungen  
FRP1600: Echtzeitprocessor optimiert für Forth  
RP-PB1: FRP1600 Prototyping Board.

### FORTech Software GmbH

Tel.: 0+381 -405 94 71 (Fax: -4059.471)  
Joachim-Jungius-Str. 9  
D-18059 Rostock

PC-basierende Forth-Entwicklungswerkzeuge, System comFORTH für DOS und Windows, Cross- und DownCompiler für diverse Microcontroller, Controllerboards mit 80C196, 80C537 und H8, Softwareentwicklung für Microcontroller und PC's, auch unter Windows (und fremdsprachig)

### Dipl.-Ing. Arndt Klingelberg

Tel.: 0+2404 -61648 (Fax: -63039)  
Strassburgerstr. 12  
D-52477 Alsdorf (b. Aachen)

Computergestützte Meßtechnik und Qualitätskontrolle, Fuzzy, Datalogger, Elektroakustik (HiFi), MusiCassette High-SpeedDuplicating, Tonband, (engl.) Dokumentationen u. Bed.-anl.

### Ing.Büro Klaus Kohl

Tel.: 0+8233-30 524 (Fax: --9971)  
Postfach 11 73  
D-86406 Mering

FORTH-Software (volksFORTH, KKFORTH und viele PD-Versionen). FORTH-Hardware (z.B. Super8) und -Literaturservice. Professionelle Entwicklung für Steuerungs- und Meßtechnik.

Möchten auch Sie oder Ihre Firma hier aufgeführt werden? Bitte wenden Sie sich an die Anzeigenverwaltung (s. Impressum).

Ihre Anzeige plus 3 Zeilen je 45 Zeichen Text kosten 90.-DM (incl. 20.-DM Einrichtung/Änderung, je Zusatzzeile 10.-DM) und das komplett für ein ganzes Jahr.

## Dienstleistungen und Produkte von Forthlern und/oder für Forthler (Anzeige)

**IMPRESSUM**

**Name der Zeitschrift**

FORTHMAGAZIN - VIERTE DIMENSION  
Organ der Forth-Gesellschaft e. V.

**Herausgeberin**

FORTH-Gesellschaft e. V.  
Postfach 1110  
85701 Unterschleißheim  
Tel./Fax: 089/3173784

**Redaktion & Layout**

Claus Vogt  
Ebersstr. 10  
D-10827 Berlin  
Tel.: 030 / 782 81 79 (Fax, BBS nach Bedarf)  
Mail: vd@FORTH-ev.de

Anzeigenverwaltung: Ulrike Schnitter c/o Forth-Ges.; PF 1110; 85701 Unterschleißheim.

ANS-Forth: Ulrich Hoffmann; uho@informatik.uni-kiel.de; Sehestädter Str. 26; 24340 Eckernförde.

Forth international: Fred Behringer; Planegger Str. 24; 8124 München.

Zeichnungen: Rolf Kretzschmar; rolf.kretzsch@forth-ev.de

**Redaktionsschluß**

Jeweils 2 Monate vor Erscheinen des Heftes.

**Erscheinungsweise**

Viermal im Jahr.

**Preis**

Einzelpreis: DM 10,-

**Manuskripte und Rechte**

Berücksichtigt werden alle eingesandten Manuskripte von Mitgliedern und Nichtmitgliedern. Leserbriefe können ohne Rücksprache gekürzt wiedergegeben werden. Für die mit dem Namen des Verfassers gekennzeichneten Beiträge übernimmt die Redaktion lediglich die presserechtliche Verantwortung. Die in diesem Magazin veröffentlichten Beiträge sind urheberrechtlich geschützt. Übersetzung, Vervielfältigung, Nachdruck sowie Speicherung auf beliebige Medien ist auszugsweise nur mit genauer Quellenangabe erlaubt. Die eingereichten Beiträge müssen frei von Ansprüchen Dritter sein. Veröffentlichte Programme gehen - soweit nicht anders vermerkt - in die Public Domain über. Für Fehler im Text, in Schaltbildern, Aufbauskizzen etc., die zum Nichtfunktionieren oder evtl. Schadhafwerden von Bauelementen oder Geräten führen, kann keine Haftung übernommen werden. Sämtliche Veröffentlichungen erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.



**Vorsicht Kultur!**

Wer weiß schon, daß Forth im Bereich der Musik eine wichtige Sprache ist? Sicher ist die Sprache der Musiker nicht das, was Forthler gemeinhin verstehen. Letzteren wäre mit einem Listing zur Programmierung der MIDI-Schnittstelle vielleicht besser gedient.

Die Echtzeit'95 brachte es ans Tageslicht: Keine Forth-Programmierer beim Programmierwettbewerb! Und das bei einem Wettbewerb, der traditionell von Forthlern gewonnen wird. Gar keine Forthler? Nun ja, nur solche, die mit C oder Pascal gewinnen wollten. Man soll dem Gegner ja eine Chance geben, sonst spielt er nicht mehr mit. Schade, daß keines der anwesenden FG-Mitglieder sich traute, einen Bericht für die VD zu verfassen. Dafür berichtet Philip Zembrod über seinen Sieg beim ACM-Wettbewerb. Auch er gewann mit sakraler Programmiersprache. Seinen Artikel schreibt er passenderweise in englisch.

Völlig sprachneutral aber umso verständlicher beginnt Bernd Paysan eine Serie über 'Optimierungstechniken'. Die hohe Kunst der Programmgestaltung als harte Technik.

Bei soviel Kultur in einem Heft will der Editor nicht nachstehen und hat sich erlaubt die verbleibende Fläche mit einer Clipart zu füllen. Zu einem richtigen Selbstportrait reichten leider weder die zeichnerischen Fähigkeiten noch die Geländegängigkeit des Handscanners.

*Claus Vogt, Editor des Forthmagazins 'Vierte Dimension'*

**Schwerpunktthema "embedded Control"**

Berlin, 10.8.95

Liebe Forthler,

in den zwei vergangenen Ausgaben wurde für die Jahreswende das Schwerpunktthema "embedded Control" angekündigt. Leider ist dieses Thema bisher auf wenig Gegenliebe gestoßen. Es wäre schade, wenn es den Bach runtergehen würde. Es hätte mich auch interessiert, ob die Leute, die ich im April angeschrieben habe, sich noch mit Contollerentwicklung beschäftigen.

Ich bin auch weiterhin der Meinung, daß das Thema für Forthler interessant und diskussionswürdig ist. Vielleicht war ja der Zeitraum von März bis heute für die potentiellen Autoren zu kurz, zumindestens eine Absichtserklärung von sich zu geben.

Auf jeden Fall scheint mir das Erscheinen des Thema "embedded Control" in unbekannter Zukunft zu liegen.

*Holger Dyja*

# Sie schenken uns einen Artikel..

Dafür möchten wir Ihnen auch ein kleines Präsent überreichen

Um die Schublade der Redaktion besser mit Artikeln zu füllen, gibt es jetzt für jeden Autor ein Geschenk als Anerkennung. Und damit es besser auf seinen/ihren Geschmack abgestimmt ist, kann es aus einer Liste ausgewählt werden. Diese wird durch Spenden gefüllt. Einige Geschenke sind schon gebraucht, aber gut erhalten - manchmal längst vergriffen und damit richtige Raritäten. Manches ist aber auch brandaktuell. Sollte man ein Geschenk nicht in sein Herz geschlossen haben, kann man es ja wieder spenden.

## Jupiter ACE - FORTH Programm-Handbuch

S. Vickers 1983, deutsch, 159 Seiten

Das Handbuch des einzigen Homecomputers, der FORTH als Betriebssystem hatte. Für Minimalisten und Oldtimer.

- gestiftet von Rafael Deliano

## Steve Jobs

J.S.Young 1988, deutsch, 495 Seiten

Von der Presse zum Kultbuch hochstilisiert, ist es jedenfalls ein sehr gut geschriebenes Buch. Beleuchtet die Microcomputerbranche zwischen 1975 und 1985 aus der Sicht von Apple.

- gestiftet von Rafael Deliano

## Vom Mythos des Mann-Monats

F.P.Brooks 1972 (1987) deutsch 165 Seiten

Der Titel wurde zum Schlagwort. Das Buch wirkt auf den ersten Blick antiquiert, weil sich der Autor auf Erfahrungen bei IBM aus den 60er Jahren stützt. Die Probleme der Softwareentwicklung haben sich jedoch kaum geändert.

- gestiftet von Rafael Deliano

## The Macintosh Way

Guy Kawasaki 1988 englisch 209 Seiten

Der Inhalt befaßt sich mit Marketing von Standardsoftware für Tischcomputer. Lesenswert wird es durch seine umfassende Sammlung industriespezifischer Jokes.

- gestiftet von Rafael Deliano

## ... aus Nebaj

Schal 2m lang, rot und gelb

Wer Nebaj nicht kennt, sollte es kennenlernen. Als Vorgeschmack gibt's einen Schal. Die Muster und Farben der Kleidung ist typisch für jedes Dorf des guatemaltekischen Hochlandes.

- gestiftet von Holger Dya

## Programmers at Work

S.Lammers 1986, englisch, 385 Seiten

Enthält Interviews, die Lammers mit 20 Top-Programmierern geführt hat. U.a. Gates, Bricklin, Hertzfeld, Kildall, Page, Warnock, Raskin,... Interessanter Einblick in die Innereien der PC-Softwareindustrie der 80er. Sowie in Ansichten und Arbeitstechnik derer, die Erfolg hatten. Auffällig wieviele von ihnen mal FORTH programmiert haben.

- gestiftet von Rafael Deliano

## Forth lernen mit dem ZF

Das ideale Geschenk für Forth-Anfänger

Friederich Prinz hat hier ein wirklich schönes Päckchen gepackt. Es enthält das ZF, ein Forth aus der Hand von Tom Zimmer, das aber überschaubarer als das F-PC ist und sich gut für Anfänger mit DOS-Erfahrung eignet. Dazu gibt es zahlreiche Aufsätze von Friederich Prinz, die sich wiederholt in den Anfängerkursen der Moerser Forthgruppe bewähren mußten.

- gestiftet von Friederich Prinz

## Das kleine Zeitgeistpaket

Wer den Timer des PC mit der Trial-And-Error-Methode bezwingen möchte, braucht dieses Geschenk nicht. Doku-Freunde finden hier: Timer-Routinen des AT-Rom-Bios, 8254-Datenblatt, Register der RT-Clock...

- gestiftet von Claus Vogt

## "Schlagzeuger"

Linolschnitt 21 x 17 cm

Ein wirklich schönes Kunstwerk aus der geübten Hand unseres langjährigen Editors. Eine Zierde jeder Wohnung.

- gestiftet von Rolf Kretzschmar

## "Fisch"

Linolschnitt 21 x 17 cm

Vermutlich aus der gleichen Periode seines künstlerischen Schaffens stammt auch dieses Werk. Der Kulturbegeisterte erkennt das unschwer an der Dynamik des Striches, der Rezensent schließt es aus der Identität des Formats.

- gestiftet von Rolf Kretzschmar

## Forth-Magazin Vierte Dimension

von hohem archivarischen Wert

Diese größte deutschsprachige Zeitschrift im Forth-Bereich dürfte den meisten Lesern bekannt sein. Weniger bekannt ist, daß einige längst verschollen geglaubte Jahrgänge dieses Blattes völlig überraschend bei jüngsten Ausgrabungen im Keller des Forth-Büros gefunden wurden. InteressentInnen sollten die gewünschte Nummer angeben.

- gestiftet vom Forth-Büro & der Familie Schmitter

## F-PC - Das Buch zum Forth

Für Ein-, Um- und Aufsteiger

Das F-PC ist sicher das derzeit meistbenutzte Forth für PC. Es stammt wie das ZF aus der Hand von Tom Zimmer und hat der Forth-Gemeinde neue Dimensionen geöffnet. Jörg Staben ist es gelungen eines seiner Einführungsbücher, das monatlich in der c't beworben wird, vor dem Verkauf zu retten. Vom Forth-Zweizeiler bis zur Optimierung der win.ini kann hier jeder was lernen.

- gestiftet vom Autor Jörg Staben

## Sponsoren gesucht!

Sicher möchten auch Sie etwas für die fleißigen Autorinnen und Autoren des Forth-Magazins locker machen. Das geht ganz einfach:

Schicken Sie der Redaktion eine kleine Ankündigung, um was es sich bei dem Geschenk handelt - bitte schicken Sie nicht gleich das ganze Geschenk!

Wenn ein Autor sich Ihr Geschenk wünscht, dann benachrichtigen wir sie, wer der Glückliche ist. Sie senden es dann - zusammen mit Ihrem persönlichen Glückwunsch - direkt an den resp. die Beschenkte.

**FORTH.** Ein Geschenk für gute Freunde. In Gesellschaft am schönsten.

<p><b>Forth-Computer...</b> Der Traum vom Forth-Computer: Vom Jupiter Ace, Wirth'schen Stackmaschinen und anderem</p>	<p>von Rafael Deliano..... 10</p>
<p><b>ultraFORTH83 rev3.82</b>  Eine neue Version des volksFORTH83 für Commodore C64, C16 und Plus4</p>	<p>von Philip Zembrod..... 11</p>
<p><b>Meßtechnik mit dem PC. Teil II</b> Programmierung des Timer0-Interrupts.</p>	<p>von Klaus Kohl..... 15</p>
<p><b>DOER &amp; MAKE kurz vorgestellt</b>  <b>DOER drehen</b> Leo Brodie hat - piffig wie er ist - den DOER erfunden. Thomas Höhenleitner dreht ihn weg, um ihn gleich wieder herzdrehen.</p>	<p>von Thomas Höhenleitner ..... 17, 19</p>
<p><b>FORTH optimieren</b> Forth ist etwa doppelt so schnell, wie alles andere, was es gibt. Wem es immer noch nicht reicht, der lernt hier das ABC der Optimierungstechnik.</p>	<p>von Bernd Paysan.....21</p>
<p><b>ICR - Intermediate Code for Robots</b> Wenn die Kaffeemaschine morgens mit dem Briefmarkenautomaten Forth spricht, versteht der Roboter nur Bahnhof. Internationale Normung bemüht sich um ihn.</p>	<p>von Rafael Deliano.....23</p>
<p><b>fLUX - Brief aus der Diaspora</b> Ein privater Brief enthüllt den Zusammenhang von Forth, den Toronto-Charts und den verschlungenen Pfaden des amerikanischen Planeten-Genius-Projekts.</p>	<p>von Frank Rothkamm...24</p>
<p><b>The Freiburg-Zurich-Nashville-Story</b> Das Kulturprogramm setzt ein Wettbewerbsgewinner fort. Um der Konkurrenz eine Chance zu lassen, wurde er fremdsprachig.</p>	<p>von Philip Zembrod .....34</p>
<p><i>Es geht - ANS Forth!</i> 'Final Draft' dpANS94 im WWW, mit Kurzlisting schneller ins ANS.</p>	<p>von Ulrich Hoffmann ..... 14</p>
<p><i>Forth International</i> Die holländische "Forth-Gebruikersgroup", 'Gehaltvolles' aus den 'Forth Dimensions' 8v16 und 'Het Vijgeblad' 45 &amp; 46.</p>	<p>von Fred Behringer .....28</p>

**Inserenten**

- 2 Forthworks
- 2 Holger Dyja
- 2 Produkte von und für Forthler
- 36 FORTech

**Rubriken**

- 3 Editorial
- 3 Impressum
- 4 Autorengeschenke
- 6 Leserforum
- 7 Korrektur 'Kreis' vd2/95
- 9 Virus 'tewi' (vd2/95)
- 32 Termine
- 35 Adressen und Ansprechpartner

**Forth-Gesellschaft**

- 12 Brief aus der Provinz
- Forth mirror in Bremen
- Excel-Kurs in Moers
- 13 ZF back to its roots
- Forth-Vertrieb
- Microcontrollerverleih
- Forth-Büro
- ELRAD in der KBBS
- Forth-Magazin sucht

**Projekte**

- 34 Vorschlag AG Marc4

**Forth-Systeme**

- 8 until251 - Forth in C
- win32forth
- 31 fastgraf für Turbo-Forth
- 34 win32forth unter os/2
- Mops 2.6 released

**Metasprachen**

- 8 C+@ von AT&T
- Open Boot

**Forth inside**

- 8 rapidFile

**Prozessorgeflüster**

- 10 Feldbuscontroller 'iX1'

**Bücher**

- 9 Noble/Rodriguez: Hornbuch
- 9 vd sucht Rezensionen

**Zeitschriften**

- 9 Byte:MPE ProForth
- M&T:ooP von FORTech?
- c't: Open Firmware
- 10 Elrad:Forth-Multitasker
- Elrad:Round Robin

**Medien**


- 10 Z-Netz:Netzbericht
- 33 EuroForth94: Abstracts

**Was noch**

- 8 ACM löst SigForth auf
- 33 The lesser known ..

**Was fehlt:**

- Die Rubriken 'Anfänger', 'ANS', 'Forth-Quellen' und 'Produktinfos',
- Die Auflösung vom Tausche-Rätsel '94
- Der Artikel zum Gray Parser

 = Listing auf Diskette



## Das kleinste Forth

Liebe Redaktion,

wie sieht das kleinste Forth aus, das in der Lage ist, "sich selbst hochzuziehen"? Kein Tiny-Forth á la Ting. Ich meine einen Pre-Kernel mit kleinstmöglicher Wortzahl, ein Bootstrapped-Forth, in das man schalten kann und das dann den Rest des Gesamtsystems hinzulädt und compiliert. (In einem solchen Fall würde man nicht mehr von Metacompilation reden können. Es wäre eine einfache Compilation, so wie ein Anwenderprogramm nicht metacompiliert, sondern einfach nur compiliert wird.) Natürlich hängt jedes Forth an der Nabelschnur eines Betriebssystems und man müßte erst einmal eine genaue Begriffsbestimmung vornehmen, bevor man das Problem präzise formulieren kann. In welcher Form soll der hinzuzuladende und zu compilierende Teil des Restsystems vorliegen? Als Datei oder soll er sich schon im RAM befinden? Von mir aus darf er sich ruhig schon im RAM befinden. Die Datei-Organisation ist nicht unbedingt die Aufgabe Forthens. Sicher ist, daß es mehrere Lösungen gibt. Um anzudeuten, in welcher Richtung ich denke: Das Minimalsystem enthält natürlich kein , (Komma), sondern kommt mit C, aus - und compiliert selbstverständlich , (Komma) als allererstes beim Hinzuladen des eigentlichen KERNELS.

*Fred Behringer, München*

## FORTH und AutoCAD

Grüß euch alle!

Ich verdiene meine Brötchen (mit Belag :-)) durch das Entwickeln von Software unter AutoCAD. Ich stoße in LISP auf Geschwindigkeitsprobleme und in C auf Geldprobleme (Entwicklung viel teurer als in LISP).

Deshalb möchte ich jetzt ein Forth unter ADS implementieren, um wieder einen Interpreter zu haben, der aber flotte Programme liefert. Völlig FORTH - untypisch möchte ich das lieber kaufen als erfinden!

Falls ich das doch erfinden muß, wird es wohl LMI- 386 - URFORTH werden.

Wenn jemand mitliest, der ein eigenes FORTH hat, das als Pharlapkompatible Library existiert oder als als C-Quelltext vorliegt, bitte melden (Lizenzvereinbarung)!

Kommentare? Meinungen?  
Wünsche? Bitte reichlich!

*Gruß, Herbert  
in de/comp/lang/forth  
hkoenig@hkoenig.forth-ev.de*

Hi, there,

VD 2/95 am 7. problemlos erhalten! Der Cover war diesmal etwas trübselig nüchtern. Irgend ein Bildchen würde der Sache ein Gesicht geben. Der Artikel über Kreisalgorithmus wäre von der Sorte, die FORTH und der VD ein ernsthaftes Image geben könnte. Auch der Artikel Plewe/Allinger war angenehm seriös in der Feststellung, daß FORTH auf PCs nicht immer optimal ist.

Die Postscript-Grafiken in meinen Artikeln sind offensichtlich danebengegangen. Sie wurden nur als Outlines dargestellt. Wohl besser, wenn man in Zukunft alles scannt.

Die Adressenseite ist immer noch nicht überarbeitet. Einige fehlen. Z.B. die Münchner Gruppe. Viele Karteileichen sind dagegen aufgeführt.

*Rafael Deliano, 16.7.95*

Hallo Claus,

Deine Version der VD kann sich wirklich sehen lassen. Man merkt, daß es Dir Spass macht! Das ist wichtig!

Zu den Bildern: Wie kommen die Verzerrungen zustande? Soll ich lieber wieder per PapierFAX schicken? Unter der Qualität des Bildes zu Meßtechnik sollte es nicht sein, wenn mein Name dran steht.... :->

Ansoensten: weiter so.....

*rolf.kretz@rkaachen.forth-ev.de  
(Rolf Kretzschmar)*

## Zu 'Brief aus der Provinz' in VD 2/95, S.18

... Da fällt mir übrigens eine Korrektur zur letzten VD ein. Darin stand, daß ich der einzige Prinz in Moers bin - oder so ähnlich. Das ist völlig falsch. In unserer Forthgruppe gibt sich mein Bruder Ulrich gelegentlich die Ehre (und uns auch). Mitglied in der FG ist er genau so lange wie ich. Und in Berlin war er auch dabei. Der Ulrich ist sogar mit auf dem Photo.

# Leserforum

## *Glückauf, Friederich Prinz [die red. entschuldigt sich untertänigst bei den vergessenen Prinzen]*

Lieber Claus Vogt,

jedesmal freue ich mich, wenn die VD in meinem Briefkasten gelandet ist und neugierig schaue ich dann hinein. Selber habe ich noch nie einen Beitrag zur Vereinzeitschrift geleistet. Die Ausrede mit der Zeit will ich nicht anführen - wir haben ja alle gleich viel Zeit; so etwa 24 Stunden jeden Tag.

Anbei übersende ich Dir zwei Files, die vielleicht das Interesse der Einen oder des Anderen erwecken könnten. [s. DOER in diesem Heft / red]

*th@marie.graphikon.com  
(Thomas Hoehenleitner), 7/95*

## **Zu 'Das war die Forth-Tagung' in VD 2/95, S.15**

Hallo allerseits,

ich habe gerade die neue VD bekommen und durchstöbert. Ist gut geworden, finde ich (dickes Lob an Claus!!!).

Jörgs Artikel über die Forthtagung bietet auch eine gute Zusammenfassung der Geschehnisse. Meinen Vortrag über die automatische Generierung von On-Line-Glossaren betreffend, finde ich aber nicht unbedingt, daß Leute, die das comFORTH für Windows nicht haben "in die Röhre gucken müssen".

Eigentlich ist es ein einfaches Verfahren, bei dem zur Zeit der Compilation die Informationen über das gerade compilierte Wort (Name, Kommentar, Vokabular,...) in eine Datei geschrieben werden. Das sollte wohl mit den meisten Forth-System durch entsprechende Verbiegungen möglich sein. Um die Informationen über das Hilfesystem von Windows als On-Line-Glossar verfügbar zu machen, ist die bewußte Datei ledig-

lich im RTF-Format mit einigen speziellen Formatierungen angelegt worden. So kann sie anschließend mit dem MS-Hilfecompiler in eine On-Line-Hilfe übersetzt werden.

Aber Du hast Recht Jörg: im comFORTH für Windows Professional wird dieses Tool fertig ausprogrammiert zur Benutzung bereitstellen. Ausprobieren, und dazu sollte der Vortrag anregen, kann es allerdings jeder, der ein Forth-System hat (und ein wenig daran herumbiegen kann), sich ein bißchen mit dem RTF-Format auseinanderzusetzen und den Hilfecompiler von MS besitzt (gibt's glaube ich auch via FTP).

MfG Udo

*us@egd.igd.fhg.de (Udo Schütz), 7/95*

Ja, nur irgendwie sind in Jörgs Artikel einige Orthographieflaws(tm) 'reingerutscht. Ob das an Jörgs Pentium liegt? Und was ist aus dem Monospaced-Font geworden? Sieht immer noch nach Courier aus. [der Font wird fleißig für Listings verwendet. Besten Dank / red]

*paysan@informatik.tu-muenchen.de (Bernd Paysan) in de/comp/lang/forth/7/95*

## **Zu 'Hochsprachen in Silizium' in VD 2/95, S.31**

Liebe VD Redaktion,

ich bin kein großer Leserbriefschreiber, aber der Artikel von Rafael Deliano "Hochsprachen in Silizium" enthält einige gravierende Fehler über die PDP-11. In der Reihenfolge des Auftauchens im Artikel:

1.) Die VAX ist keine aufgebohrte PDP, sondern besitzt eine vollkommen andere Architektur. Das einzige,



Leserbriefe: Am liebsten kurz. Sonst trifft uns die Pflicht zur Kürzung. Die Redaktionsadresse lautet:

**Forth-Magazin 'Vierte Dimension'**  
**Claus Vogt, Ebersstr. 10,**  
**D-10827 Berlin,**  
**vd@FORTH-ev.de**

**Forth. Kurz und Knapp. Das**

das sie mit der PDP verbindet, ist - genau genommen: war - daß alle Modelle der VAX, die eine 11 vor ihrem Namen hatten, einen Kompatibilitätsmodus hatten, in dem sie PDP-11 Programme ausführen konnten. Seit 1986 wird keine VAX mehr gebaut, die diesen Modus implementiert hat.

2.) Die PDP hat 8 Register, wovon 2 von der Architektur eine zusätzliche Bedeutung bekommen haben. Dies sind Register 6, das der Systemstack-pointer ist. Man kann dieses Register aber auch selbst verwenden, solange man beachtet, daß bei einem CALL, einer Exception oder einem Interrupt die Rückkehradresse auf den Stack gepusht wird. Register 7 ist der Programmzeiger. Es wird während der Ausführung automatisch verändert. Aber der Programmierer kann dieses Register genauso wie alle anderen in jeder Operation als Quelle oder Ziel verwenden. Ob das in einem speziellen Fall Sinn macht oder nicht. Für C und andere Hochsprachen, die ihre Parameter über den Stack übergeben, hat die PDP einige nette Features. Sie kennt nämlich - und das gilt für ALLE Register - den Adressierungsmodus "Register indirekt mit Offset". Damit kann eine Routine auf Werte mitten im Stack zugreifen ohne den SP zu verändern! Verwendet man diese Addressierung zusammen mit dem PC, dann ergibt das Code, der a priori unabhängig von der Adresse ist, an die er geladen wurde.

3.) Die PDP kennt genau zwei Additionsbefehle, einen mit Carry und den anderen ohne. Die 64 Varianten des Befehls sind schlichter Quatsch! Oder will der Autor jemandem verlickern, daß der Befehl ADD (R0), R1 eine andere Variante des Befehls ADD (R5),(R6)+ ist. Bei den beiden unterscheidet sich nur die Addressierung. Wenn das aber eine Variante sein soll, dann kennt die schwachsinnige Architektur der Intelchips gar 96 Varianten!

4.) Die Behauptung, daß die PDP darauf ausgelegt war, Memory-zu-Register- Bewegungen zu machen, ist

ebenfalls grob verzerrt. Sicher kann sie das. Aber sie kann ohne weiteres einen Wert aus dem Speicher auf einen anderen Wert im Speicher addieren (oder was auch immer), ohne daß einer der beiden Werte vom Programmierer in irgendein Register geladen werden muß. Das nennt man aber eine 2-Adress-Maschine!

5.) Es waren nicht nur einige 2-Adressbefehle vorhanden, sondern alle Befehle, die zwei Operanden hatten, konnten in der oben geschilderten Addressierungsart ausgeführt werden. Logischerweise ist ein NOT eine unäre Operation und hat daher nie einen zweiten Operanden, aber das ist trivial.

Es sind noch ein paar weitere Wischi-Waschi-Aussagen im selben Kapitel, aber um die zu korrigieren, müßte man das Kapitel neu schreiben.

Ich finde, jemand sollte nur dann über etwas schreiben, wenn er auch hinreichend Ahnung von dem Stoff hat. Mir jedenfalls langte es nach dem schauderlichen Kapitel, und ich habe nicht weitergelesen in der Befürchtung, daß weiter hinten ebenso falsche Dinge behauptet werden und ich es aber nicht bemerke, weil ich die Materie noch nicht kenne.

Dieser Artikel hat m.E. mehr Schaden angerichtet als Nutzen. Hoffentlich enthält er nur Zeug, das die Leser schon kennen und daher die Qualität selbst beurteilen können. Wenn nicht, hat der Autor wunderbar zur Volksverdummung beigetragen.

Mfg

Ulrich Paul,

100022.3315@compuserve.com,  
7/95

**Stellungnahme der**  
**Redaktion:**

Schade, daß Paul den Artikel nicht mochte. Im Gegensatz zu ihm kann ich die PDP-11 nicht programmieren. Ich habe auch nie mit der DEC-Welt zu tun gehabt. Der Artikel wollte aber nur sehr knapp Architekturen gegenüberstellen und nicht die PDP-11 erschöpfend behandeln.

DEC hat ein 580 Seiten starkes Buch herausgegeben [1], das die geschichtliche Entwicklung und den Hintergrund der Design-Entscheidungen der DEC-Systeme erläutert. Darauf habe ich mich gestützt und DEC's Meinung zur PDP-11 scheint mir immer noch relevant. Im Einzelnen:

1) Als "VAX" ist hauptsächlich der Typ 11/780 bekannt. Dieser wurde von DEC als Weiterführung der PDP-11 Produktlinie entwickelt. Vgl. in [1] das Kapitel: "VAX11/780: A Virtual Address Extension to the DEC PDP-11 Family". Insoweit man einen 80486 als aufgebohrten 8086 bezeichnen darf, trifft das dann auch für VAX und PDP-11 zu.

2) Ulrich hat recht damit, daß die PDP-11 sechs und nicht wie falsch im Text angegeben fünf universelle Register hat. Daß die beiden anderen Register auch universell verwendet werden könnten, ist ein technisches Detail, das mir in diesem Zusammenhang nicht wesentlich schien.

3) Ich legte Wert auf die Herausstellung der CISC-Philosophie: "The PDP-11 provides more addressing modes than nearly any other computer. The eight modes for source and destination with dyadic operators provide what amounts to 64 possible ADD instructions" [1] S. 386.

4) 5) "The basic design decision that sets the PDP-11 apart was based on the observation that by using

TRULY general registers and by suitable addressing mechanisms, it was possible to consider the machine as a 0-address (stack), 1-address (general register) or 2-address (memory-to-memory) computer." [1] S.251.

D.h. die PDP-11 wurde als sehr flexible General-Register-Maschine entworfen. Das kam im Text aber auch zu Ausdruck, glaube ich.

Für Hinweise auf sachliche Fehler sind Autor, Redaktion und Leser jederzeit dankbar.

Soweit es um verschiedene Interpretationen geht, sollte man vermeiden, seinen Standpunkt als den einzig denkbaren anzusehen.

[1] "Computer Engineering. A DEC View of Hardware Systems Design" Digital Press 1978

Rafael Deliano, Juli '95

**Korrektur zu**  
**'Kreialgorithmus'**  
**in VD 2/95, S.9**

Lieber Claus,

hier mein Vorschlag zur Verbesserung des Artikels Kreialgorithmus. Er sollte möglichst kurz sein, deshalb lasse ich alles Unnötige weg:

In unserem Artikel haben sich ein paar Fehler eingeschlichen, die wir hiermit richtigstellen.

Autoren sind: Fritz Prinz; Hombergerstraße 335; 47443 Moers und Ulrich Richter; Oberwallstraße 4;; 47441 Moers

Auf Seite 10 Spalte 2 Mitte muß es heißen:  
 $(y-1)^2 = (y-1)(y-1) = y^2-2y+1$

Seite 10 Spalte 3 oberes Drittel  
 $r^2-x^2-y^2-2y-1 = \Delta_1$   
 $r^2-x^2-y^2+2x-2y-2 = \Delta_2$

Seite 11 Spalte 1 obere Hälfte  
 $-2y-1 \rightarrow -2(y-1)-1 = y-1-y-1-1 = -2y-3$

Mit besten Grüßen  
 - Ulrich -

## Forth inside: C+@

AT&T's Bell Labs entwickelte in den letzten Jahren außer C++ eine weitere objektorientierte Programmiersprache mit dem Namen C+@. Während man C++ mehr oder weniger als eine Erweiterung von C um objektorientierte Mechanismen ansehen kann, handelt es sich bei C+@ dagegen um eine rein objektorientierte Sprache. Die Klassen selbst werden als Objekte betrachtet und Methodenaufrufe ausschließlich über Message Passing abwickelt. Eine Kopplung mit vorhandenem C-Code ist problemlos möglich.

Die Entwicklungsumgebung selbst ist in C+@ geschrieben, läuft unter X11 und bietet die gewohnten graphischen und textuellen Werkzeuge zum Class-Browsing, dem Generieren neuer Codes und zum Debugging. Als Besonderheit bietet die Umgebung weiterhin eine Shell, die einerseits den interpretativen Aufruf von Klassen und Methoden erlaubt und andererseits aber auch eine inkrementelle Compilation unterstützt, mit der neue Klassen und Methoden bei laufender Applikation hinzugefügt werden können.

Um eine Binärkompatibilität auf verschiedenen Plattformen zu erreichen, verwendet man zur Implementation 'threaded Code'. In einem ersten Schritt wird das Programm in Token-Ketten umgesetzt. Diese werden mit einer separat vorcompilierten, maschinenspezifischen Bibliothek für den Token-spezifischen Maschinen-Code gelinkt (Diese Primitiva werden hier 'Beads' genannt). Analog zu bekannten Fädungstechniken schließen alle Beads auch hier mit dem Aufruf des nächsten Primitives, sprich mit einem NEXT.

Ausführlichere Informationen zur Sprache findet man in:

- [1] Jim Fleming; The C+@ Programming Language; Dr. Dobbs Journal, 10/93, S.24 ff.
- [2] Michael Wagner; Katze aus dem Sack; iX, 6/95, S.152 ff.

*tb@tb.forth-ev.de(Thomas Beierlein)*

## Forth inside: Was ist 'Open Boot'

The idea of openboot is you can put a primitive driver on ROM on a PCI card. The same card could run with an Intel CPU or a PowerPC. It would be enough to get you going. Normally after you were fired up you

would replace with a native mode driver. OpenBoot is basically a portable Forth interpreter. The nucleus of ASM words has to be implemented in ROM BIOS in a CPU specific manner. The drivers are written in pure high-level code put into ROM on the peripheral cards.

You could also put the code that does the equivalent of configuring jumpers into such OPENBOOT ROMS.

I have never played with OpenBoot directly. This knowledge is cobbled together from reading articles and asking questions.

The OpenBoot people were careful not to call this Forth since it would scare the horses.

*Roedy@bix.com in comp/lan/forth 7/95*

## Forth inside: rapidFile

... I guess it's time to start this thread again ... Marketing FORTH. I recall having a discussion many years ago with a member of the RapidFile Development Team at [then] Ashton-Tate. RapidFile was one of the only applications turned out by a major software house which was written in FORTH. During the discussion I mentioned that I noticed the

*natkin@clark.net(Todd Natkin) in comp/lang/forth 6/95*

## Forth-Systeme: until251 - Forth in C

I have uploaded to SimTel, the Coast to Coast Software Repository (tm), (available by anonymous ftp from the primary mirror site ftp.coast.net and the secondary mirrors of SimTel): <ftp://ftp.coast.net/SimTel/msdos/forth/until251.zip> SimTel/msdos/forth/until251.zip.

Until V2.5.1 is also available via anonymous FTP and on the Web on [Taygeta.oc.nps.navy.mil in /pub/Forth/Reviewed/until251.zip](http://Taygeta.oc.nps.navy.mil/pub/Forth/Reviewed/until251.zip).

Until is an almost Forth-83 written in C. Its internals are described in "Write Your Own Programming Language Using C++", ISBN # 1-55622-264-5. The book is available from Fig or your local book store.

I use Until as a scripting language for SGML applications at the moment. It has many features, such as a string search and replace word set, null terminated strings (useful with C strings), and the C I/O and string libraries, that make it very comfortable for developing text processing applications.

Changes to v2.5.1 include:

- Updated documentation. The printed doc is now over 175 pages.
- C-level I/O redirection code
- added error checking for all malloc() calls
- backslash quoted C-style characters in "..." string constants
- new words include getenv, case, and printf
- the S-Engine SGML document interpreter
- a minimal floating point word set
- the string search and replace word set is now fully implemented

The distribution contains full C source code that is very portable. It compiles on several other platforms in addition to MS-DOS. The MS-DOS version of Until does not require a DOS extender; it will run on PCs as minimal as HP-95LXs.

until251.zip has replaced until22.zip.

Uploaded by the author.

*72745.1566@CompuServe.CO M (Norm Smith) in comp/lang/forth*

## Forth International: ACM löst SigForth auf

Auf der ACM SIGPLAN PLDI-Konferenz vom 18.6.95-21.6.95 in La Jolla, CA verlautebarte die Präsidentin der SIGPLAN (Special Interest Group on Programming Languages), daß SigForth durch SIGPLAN übernommen wird. SigForth hatte finanzielle Probleme. SIGPLAN wird den SigForth-Newsletter nicht fortführen, der aber ohnehin keinen Herausgeber mehr hatte und dessen Zukunft daher auch im Rahmen der SigForth unklar war. Vermutlich wird die SigForth-Konferenz ebenfalls eingestellt.

ACM SigForth wurde um 1990 als Organisation für professionelle Forth-Benutzer gegründet. Die Gründer sahen FIG als Hobbyisten-Verein an. Treibende Kräfte waren u.a. George Shaw, Irv Montanez und Paul Fregner. Leider konnte SigForth den professionellen Anspruch nicht erfüllen, soweit ich das aus der Lektüre der Newsletters beurteilen kann. Die Newsletters waren zwar schön gestaltet, aber sie kamen unregelmäßig und enthielten meines Erachtens relativ wenige Artikel professioneller Qualität. Offenbar hat sich die professionelle Forth-Welt kaum in der SigForth engagiert. Im Herbst 1994 schlug SigForth der FIG eine Vereinigung der beiden Gruppen vor.

In der Usenet-Newsgroup comp.lang.forth engagierte sich vor allem Jack Woehr fuer die Vereinigung.

Letztendlich wurde sie offenbar von FIG abgelehnt oder so lange hinausgezögert, bis es fuer SigForth zu spät war.

*Anton Ertl*

## Forth-Systeme: Tom Zimmers win-32FORTH Version

Henry Vinerts, ein Mitglied der FIG Silicon Valley, war so liebenswürdig, mir auf meine Bitte hin eine Kopie von Tom Zimmer's jüngstem Werk direkt aus den USA zukommen zu lassen. Die Diskette erreichte mich Mitte Juli '95.

Laut Begleittext benötigt win-32FORTH WINDOWS 3.1 und WIN32S. Unter OS/2 bekam ich es nicht zum Laufen, obwohl es ab Version 3.0 selbst mit einem virtuellen WIN32S 'gesegnet' ist: "DOS-Fehler 23" - laut Hogan's Tabellenwerk ein CRC-Fehler beim Lesen der Festplatte. Sehr geheimnisvoll. Also gut, dachte ich - WINDOWS halt dich fest, du kriegst eine allerletzte Chance! Aber auch unter Windows 3.11 klappte nichts.

In den beiliegenden Quelltexten schreibt Tom Zimmer im File FORML94.TXT (\*.WRI), daß WIN-32FOR für Windows95 und WindowsNT entwickelt worden sei. Der Ulrich Richter 'fährt' auf seinem Rechner Windows95 in der x-ten Version der "Final"-Betarelease. Also, Termin mit Ulrich abgesprochen - jetzt wollten wir es wirklich wissen. Und tatsächlich - nach 1 1/2 Wochen des Herumärgerns zeigt sich WIN-32FOR erstmalig auf dem Bildschirm.

Es sieht sogar ausgesprochen gut aus, richtig professionell. So problemlos sich das Forth in Windows's Oberfläche (TaskBar) einbinden ließ, so problemlos gestaltet sich auch das 'Zurechtfinden' in der zunächst ungewohnt komfortablen Entwicklungsumgebung. Ungewohnt ist die Umgebung aber nur für mich. Ulrich, der mehr mit C++ arbeitet als mit Forth, entfährt das Statement: "Mensch, das kommt aber haarscharf an Borland heran".

Die ersten Versuche eine 'alte' \*.SEQ Applikation laufen zu lassen, sind zum Scheitern verurteilt. WIN-32FOR kennt kein CMOVE. Das ist schnell dazugeschrieben, aber dann fehlt C!L und C@L und... Das lassen wir für's erste mal lieber sein. Stattdessen wenden wir uns den Quellen zu - und ich erlebe eine Art Kulturschock! Jetzt 'rächt' es sich,

*Fortsetzung auf Seite 34 unten*





### Presseschau: Byte

In "Byte" 6/95 Seite 401S23 findet sich eine kurze Vorstellung von MPEs ProForth für Windows. Kann 10 Jahre her sein, seit dort die Produktvorstellung eines FORTH gemeldet wurde. Ansonsten ist in "Byte" zu lesen, daß sich Open Firmware besonders bei Apple immer mehr durchsetzt.

*jrd (Rafael Deliano), Juli 95*

### Presseschau: M&T 15/1995

Daß Forth in der Öffentlichkeit mittlerweile unbekannt ist, hat auch seine guten Seiten. Jedenfalls konnten Diessner und FORTech ihre FORTHs problemlos in der M&T 15/95 in

der "Marktübersicht: Objektorientierte Programmiersprachen" zwischen all den C++ und Smalltalks unterbringen. Für die Redakteure genügte es offenbar, daß wenn man exotisch ist, man gewiß auch objektorientiert, fuzzy, neuro, multimedial ist ...

*jrd (Rafael Deliano), Sept. 95*

### Presseschau: c't Forth in 'Open Firmware'

Zum Artikel über den neuen PowerMac in c't 8/95 wird auf Seite 148 in einem Kasten Open Firmware erläutert. Dabei wird auch verraten, daß Open Firmware auf Forth beruht (vgl. VD 1/95, S.7). Einige Forth-Prinzipien werden erläutert. Es wird mitgeteilt, daß zur PCI-Firmware-

Entwicklung auf dem Mac ein ANSI-konformes Forth-System namens CForth-93 (von Mitch Bradley ?) empfohlen wird.

*Joerg Plewe, Juli '95*

### Presseschau: Elrad

In ELRAD 7/95 S48 findet sich von mir ein einführender Artikel wie man den PAUSE-Multitasker auf einem Mikrocontroller implementiert. Verwendet wurde wieder nanoFORTH

auf einem Mitsubishi-Controller (vgl. VD 1/95 S.6). Für Unterlagen zu anderen PAUSE-Implementierungen bin ich Klaus Kohl zu Dank verpflichtet. Die ELRAD hat im Artikel auch eine kurze Notiz über die Forth-Gesellschaft untergebracht.

*jrd (Rafael Deliano), Juli 95*

Wie manche schon gesehen haben, ist in der neuen (7/95) ELRAD ein Artikel zu "Multitasking per RoundRobin unter Forth" von Rafael Delia-

*Fortsetzung nächste Seite*

## Bücher

### J.V. Noble, R.J. Rodriguez The Apprentice Programmer's Hornbuch

(engl., 'Das Buch für Programmierlehrlinge')  
ein Lehrbuch

In einem ca. 15 KByte langen Text hat J.V. Noble das Buch in COMP/LANG/FORTH vom Juli vorgestellt. In seiner amüsanten Weise versucht er FORTH als Werkzeug zu charakterisieren, das beim Verstehen des Computers ganz allgemein hilft: Beim Begreifen einfacher, effizienter Algorithmen, beim Schreiben fehlerarmer Programme und auf dem Weg vom Studenten zum 'guten Ingenieur' -so J.V Noble. Er vergißt auch nicht mit seinem Buch ein 'ausgezeichnetes PD-Forth' (nämlich F-PC 3.6) mitzuliefern.

Das Buch beginnt bei den Grundlagen der Digitaltechnik, behandelt allgemeine Hochsprachenkonzepte, Fragen der Programmgestaltung und -kommentierung, diskutiert 'wichtige Algorithmen' wie Sortiermethoden. Ab Kapitel 8 beginnt die Assemblerprogrammierung mit hardwarenaher und Echtzeitprogrammierung, Floatingpointprozessoren und Schnittstellen. Applikationen von Expertensystemen über Textformatierer bis zum 'Number Crunching' werden angerissen und zu guter Letzt bietet die 'Optimierung' einen willkommenen Anlaß tief in die Innereien des Forth-Interpreters einzusteigen, diverse Fädelungstechniken und die Metacompilation vorzustellen.

Von Null auf Hundert in 12 Kapiteln? Haben die Autoren jemals mit Anfängern gearbeitet? Sind die Kapitel sehr lang und umfassend? Oder sind die Amis alle viel schlauer als wir Niederrheiner...

*Friedrich Prinz*

### Virus zu 'Forth auf CD' in VD 2/95, Seite 25

Folgenden Output bekam ich beim Scannen der ersten CD aus dem tewi-Paket "Power-Programmierung" (ja, des mit den 19FORTH-Systemen drauf) mit McAfee VirusScan V221, also Vorsicht:

```
n
--- begin of original log file ---
Scan V.2.2.0 Copyright (c) McAfee, Inc. 1994,1995.
All rights reserved.
Virus data file V221 created 04/13/95 18:31:41

07/02/95 19:38:48

Options:
/REPORT e:\scan.log
F:\MODULA2\COMPILER\TAYLMOD2\TMODULA.EXE
Found the VC-SNVF virus
F:\PASCALLIBRARY\DOS\DATABASE\UNITDB\TEST.EXE
Found the BCV virus

Summary report on F:

File(s)
Analyzed: ..... 38588
Scanned: ..... 3978
Possibly Infected: ..... 2
Time: 00:41:19

--- end of original log file ---
```

Der tewi-Verlag hat inzwischen auf den Virus-Alarm reagiert. Sie meinen, sie hätten mit 4 verschiedenen Virensclannern geprüft, u.a. von IBM und McAfee und es müßte ein Fehlalarm sein. Ein bißchen Vorsicht ist meines Erachtens trotzdem angezeigt.

*Philip Zembrod, Juli'95*

### Rezensionen gesucht!

Das Forth-Magazin möchte gerne mehr Rezensionen veröffentlichen. Außerdem suchen wir noch Personen, die ihre Zeitschriften hinsichtlich forthrelevanter Themen lesen und darüber Kurzberichte verfassen!

**Bitte bei der  
Redaktion melden / Claus**

no (Seite 48ff). Ganz überraschend dann auf Seite 50 ein kleiner Hinweis auf den Forth-e.V., diese Newsgroup und die Mailbox.

Exklusiv für Norddeutschland gibt es seit 3 Tagen hier einen Abzug der Elrad-Mailbox (Stichtag 6. Juni). Dort sind zum Thema Forth auch ein paar Sachen:

CORDIC.F65 Forth-Listing zu "Zahlenmühle", Stephan Groáe-Holtforth

F51.ZIP Forth Entwicklungssystem für IBM

PC, FORTH11.ARC FIG-Forth für 68HC11 Stephan Groáe-Holtforth

Gruss, Holger (hp@kbbs.org)

## Netzbericht: Z-NETZ und COMP/LANG/FORTH

Im Internetforum DE/COMP/LANG/FORTH sind sich ein paar 'Kampfhähne' in die Haare geraten. Ein schwaches Echo des Streites über das alljährlich neu aufflammende Thema 'Kommerzware vs. PD' war noch in Z-NETZ/SPRACHEN/FORTH zu lesen. Ursprünglich hatte der 'Streit' mit einem (beinahe) philosophischen Thema begonnen - nämlich mit der (unausgesprochenen) Fragestellung, inwieweit es 'Rechters' ist, daß ein Forther beim anderen 'abschreibt'.

Ich habe noch Jörg Plewe's Worte im Ohr, die er auf der Jahrestagung der Forthgesellschaft von sich gegeben hat, und die davon sprachen, daß Forther alle Räder immer wieder neu erfinden, weil das Erfinden einfach so großen Spaß macht. Mit dieser 'Grundhaltung' gerät man natürlich niemals in die Gefahr zu irgendeinem Projekt als 'Plagiator' tätig zu werden.

Wer aber seinen Lebensunterhalt damit verdient, daß er seinen Kunden funktionierende (Programm-) Lösungen schnell und preiswert anbieten kann, der müßte schon ziemlich 'beklopft' sein, wenn er nicht bereits funktionierende Dinge aufgreift, beund/oder überarbeitet und weiterverwendet - gewissermaßen recycelt.

Letztlich hat sich der 'Streit' auch in etwa auf dieser Schiene aufgelöst. Der Aufforderung, sich in die dazu angestregten Diskussionen einzumischen, sind nicht sehr viele Forther gefolgt.

Interessanter als die beiden deutschsprachigen FORTH-Foren war für mich in der letzten Zeit das englischsprachige Forum COMP/LANG/FORTH. Dort 'tummeln' sich ohnehin einige der bekanntesten Na-

men der internationalen Forthszene, wie z.B. Elizabeth Rather. Aber dort gibt's auch, wenn man sich nur zu fragen traut, kompetenten Rat und Hilfe zum Nulltarif in kürzester Zeit. Und dort lesen sehr viel mehr Menschen mit, als sich an den täglich zu lesenden Diskussionen beteiligen.

Ein Beispiel: Ich habe vergeblich versucht, Tom Zimmer's WIN32FOR in einer WINOS/2 Session zum Arbeiten zu bewegen. Tom Zimmer hat aber keine (zumindest keine mir bekannte) Adresse im Internet. Auch von seinen Co-Autoren zu WIN32FOR habe ich im Internet bisher nie etwas 'gehört'. Also habe ich eine allgemeine Anfrage gestartet, in der

Hoffnung, daß irgendjemand Rat weiß. Und prompt hat mir Andrew McKewan geantwortet, DER Co-Autor zu WIN32FOR. Andrew liest COMP/LANG/FORTH ständig mit, mischt sich aber selten in allgemeine Diskussionen ein.

F.PRINZ@MHB.gun.de  
(Friedrich Prinz) im Juli '95

## Prozessorgeflüster: Der programmierbare Feldbuscontroller iX1

Die Ablösung des Kabelbaums in der Maschinen- und Anlagenverkabelung durch den Feldbus wird immer noch durch den Normenwirrwarr der verschiedenen Systeme (CAN, FIP, IEC-Feldbus, INTERBUS-S, PROFIBUS, P-NET etc.) behindert.

Mit dem programmierbaren Feldbuscontroller IX1 hat DELTA t ein ASIC (PLCC-68 Gehäuse) entwickelt, das als single-chip Controller durch Umprogrammieren an alle Feldbusse angeschlossen werden kann. Der Prozessor besitzt eine dual-stack Architektur mit einem forthigen Befehlsatz, der für die Verarbeitung serieller Protokolle optimiert ist. Intern besitzt der IX1 ein 1k x 12 großes DatenRAM und der interne Programmspeicher ist 4k x 12 bits groß.

Programme werden mit einer interaktiven Entwicklungsumgebung auf dem PC in Forth und in Zukunft auch mit C erstellt.

Klaus Schleisiek-Kern;  
DELTA t GmbH; Adenauer Allee 54;  
20097 Hamburg;  
Tel: 040-280152-20;  
Fax: 040-280152-90

# FORTH-Computer..

Es gab vor vielen Monaten in d.c.l.f eine kurze aber heftige Diskussion darüber, ob man nochmal einen FORTH-Computer bauen könnte oder sollte. Also einen Tischcomputer mit Betriebssystem in FORTH, der dann zumindest in Kleinserie produzierbar ist.

## Jupiter Ace & Co

Der einzige explizite FORTH-Computer, der Verbreitung gefunden hat, war 1982 der Homecomputer Jupiter Ace. Der kann allerdings heute kein direktes Vorbild mehr sein. Er entspricht den technischen Anforderungen nicht mehr. Zudem besteht keine Möglichkeit für Verbreitung auf breiter Basis, wie sie ein solches Consumer-Produkt erfordert. Angeregt wohl durch Dirk Brühl (vgl.VD 89/1) hat 1991 Dr. Olaf Haeggquist eine Maschine auf RTX2000-Basis gebaut, mit Tastatur, Floppy, Harddisk, Videointerface. Das beweist zumindest, daß es geht. Das System blieb aber ein Unikat, Verbreitung in Kleinserie scheint nie angestrebt gewesen zu sein.

Die Idee, einen Tischcomputer um eine Programmiersprache herum zu bauen, ist nicht neu und auch nicht auf FORTH beschränkt.

## Wirths Maschinen

Prof. Wirth hat mehrmals eine neue Workstation gebaut, wenn er eine neue Programmiersprache entwickelte. Er hält den Aufwand für gerechtfertigt. Ernsthafte Softwareentwicklung würde seiner Meinung nach sonst durch ungeeignete Hardware behindert.

Nach einem Besuch im Forschungszentrum PARC von Xerox (1976/77), begann er mit dem Bau des Lilith-Rechners auf dem dann gleichzeitig Modula-2 entwickelt wurde. Dieser Pseudo-Alto verwendete Bit-Slice-CPU's (AM2901), hatte bit-mapped Grafik und eine Maus. Mit 6 Mitarbeitern, die allerdings nur zeitweise daran arbeiteten, dauerte es 2 Jahre bis Hardware und Software einsatzfähig wurden. Sie war damit vor Lisa und Macintosh fertig und erregte zumindest in Fachkreisen Aufsehen. Aber genau wie der Alto von Xerox war das Gerät nicht fürs gemeine Volk bestimmt, sondern für den Eigenbedarf der Hochschule.

Der 32-Bit-Rechner für seine nächste Sprache Oberon heißt Ceres und wurde von den Studenten der ETH entwickelt. Als CPU entschied man sich (ca. 1985) für den NS-32GX32 von National Semiconductor. Ein CISC ähnlich dem 68000, aber mit wesentlich orthogonalerem Befehlssatz. Wirth schätzt, daß sein Codegenerator für den 32000 deshalb nur halb so umfangreich wie für den 68000 ist. Das Gerät hatte Farbschirm mit 1024x800 Pixel. 3-Tasten-Maus, 4-8 MB Speicher und einen simplen Netzwerkanschluß über RS485.

Für Pascal gab es keinen Computer von Prof. Wirth. Dafür aber um 1980 die "Pascal Microengine" von Western Digital. Die CPU bestand aus einem Chipsatz von fünf NMOS-ICs, die darauf optimiert waren P-Code-Pascal auszuführen. Der Programmierer sollte mit allem Komfort portable Programme in Hochsprache auf der Microengine entwickeln. Und wenn sie liefen, auf marktgängige Mikrocomputer portieren. Damals waren noch CP/M-Zeiten und Tischcomputer hatten den Z80 als CPU. Wodurch die strenge Unterscheidung zwischen Entwicklungs- und Zielsystem teilweise verständlich wird.

Zeitgleich 1980 startete bei Apple die Arbeit an Lisa. Erste Konzepte sahen einen bit-slice-Rechner, der auf P-Code optimiert war, vor. Die Verfügbarkeit des kosteneffektiveren 68000 änderte das Hardwarekonzept. Die Ausrichtung auf Pascal hingegen wurde beibehalten. Auch weite Teile des Macintosh-OS waren ursprünglich in Pascal geschrieben.

Pascal und Nachfolger sind kein Einzelfall. Auch für LISP hat es eigene Workstations von Symbolics und TI gegeben. Und früher oder später nimmt auch wieder jemand einen neuen Anlauf, einen neuen FORTH-Computer zu bauen.

Rafael Deliano



# ultraFORTH83 rev.3.82

von Philip Zembrod

Philip Zembrod; In den Muehlmatten 5; 79288 Gottenheim; zem@iw.m.fhg.de

Lang ist es her, daß eine neue Version des ultraFORTH83/volksFORTH83 der Forth-Gesellschaft erschien. Im Jahre 1988 sprang das ultraFORTH83 von Level 3.5 auf 3.8 und unterstützte hinfort außer dem Commodore C64 auch den C16 und Plus4. Später kam das volksFORTH83 3.81.41 für PC hinzu.

Philip Zembrod hat bereits vor Jahren ein paar lang erwartete Fehlerkorrekturen und einige Veränderungen im ultraFORTH vorgenommen. Das Ergebnis ist die release 3.82, die wir hiermit der Öffentlichkeit vorstellen. Sie wird beim Forth-Vertrieb 'Klaus Kohl' und in der Mailbox auf der VD-Sammeldiskette erhältlich sein. Ob sie die Version 3.8 im Vertrieb ersetzen wird, stand bis Redaktionsschluß noch nicht fest.

*Stichworte: ultraFORTH volksFORTH C64 C16*

*Dokumentation der Änderungen  
im ultraFORTH für C64/C16  
von Version 3.80 zu Version 3.82  
(Stand: Philip Zembrod, 2.3.91)*

Meiner Erfahrung nach sehr praktisch.

**Scr. 128:** PENCOL von 0540 auf 0536 geändert. -> C16-INKPOT -Be-fehl

**Scr. 143:** In index und find wurde 2 .r zu 3 .r und 1+ 25 type zu 28 type verändert.

**Scr. 139:** DERROR? wurde verändert: In die IF- Abfrage wurde ein ELSE-Zweig eingeschoben, der den Rest der "00,OK,00,00"-Meldung abholt und verwirft. Dies ist nötig, damit die Floppy eine neue Meldung bereitstellt.

**Scr. 123:** (64 bzw. (16 prüfen jetzt auf Länge= von c)

**Scr. 113:** in endpoints wurde das @ nach voc-link gestrichen, da es bewirkte, daß das neueste Vokabular nicht durchsucht wurde.

**Scr. 16:** Logo auf "ultraFORTH-83 3.82 ..." geändert

**Scr. 49:** Martin Tracy's UNLOOP wurde eingebaut. Es entfernt die Parameter von DO-LOOP vom Returnstack. In Forth implementiert, sähe es für uF so aus:

```
: UNLOOP rdrop rdrop rdrop ;
```

Es wird verwendet, um innerhalb einer DO-LOOP-Schleife ein Wort zu verlassen.

```
: suchen ( xx -- yy )
  vorbereiten
  ende anfang DO
    found? IF I irgendwas
      UNLOOP exit
    THEN
  LOOP leidernichtgefunden ;
```

**Scr. 35:** Das Wort 4+ wurde sichtbar gemacht.

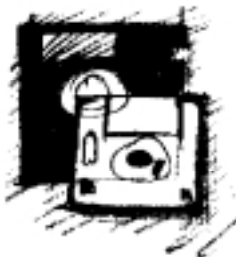
**Scr. 139: Scr. 140 - 148** wurden nach **141ff.** verschoben und der load/thru - Befehl in Scr. 123 angepaßt.

DERROR? wurde nach **Scr. 140** verschoben, in **Scr. 139** wurden die Worte I/O-STATUS? (liest das I/O-Statusbyte an Adresse \$90) und BUS-READ ( was der genau macht, weiß ich im Moment leider selbst nicht mehr) eingefügt.

Der geänderte Quellcode befindet sich in der Datei uf-mysrc.d64 Das ist ein Disk-image für den C64-emulator C64S.EXE. Es enthält einfach in aufsteigender Reihenfolge alle Sektoren aller Spuren: (T1S0, T1S1, ..., T1S20, T2S0, ..., T17S20, T18S0, ..., T18S18, ..., T24S18, T25S0, T25S17, ...T30S17, T31S0, ..., T31S16, ..., T35S16).

Die uF-Screens liegen direkt auf die Disk-Sektoren abgebildet; lediglich die Sektoren T18S0 und T18S1 werden ausgespart. Ein gutes Utility zum Zurückspielen einer .d64-Datei auf eine 1541-Diskette gibt es meines Wissens derzeit noch nicht, sorry. Ich habe aber den C-Quelltext eines kurzen Programms beigefügt, das aus einer .d64-Forth-Screen-Disketten-Image-Datei eine les- und bearbeitbare Ascii-Datei macht.

Die Dateien c64uf382.prg und c16uf382.prg sind die ausführbaren Forth-Kernsysteme für C64 bzw. C16.



**Die VD-Sammeldiskette VD95-3  
enthält zu diesem Beitrag:**

C64UF382.PRG 15,099 .a.. 07-07-95 Das System uF3.82 für C64  
UF-MYSRC.D64 174,848 .a.. 26-09-94 Die Quellen zu uF3.82  
UF382CHG.DOC 2,613 .a.. 07-07-95 Die Änderungen in uF3.82  
C16UF382.PRG 15,168 .a.. 07-07-95 Das System uF3.82 für C16 & Plus4  
4FORMAT.C 2,177 .a.. 13-07-95 C-Programm zum Screenumformen

## Forth-Gesellschaft: Brief aus der Provinz

Moers. Sommerzeit - Urlaubszeit... Der Michael ist schon in der Bretagne. Ich werde in wenigen Tagen in Süd-Tirol in den Bergen wandern und in alten Bergwerksstollen im 'Dreck herumkratzen. Und der 'große Rest der Gruppe' freut sich doch tatsächlich darauf, Ferien von Michael und mir zu machen. Ist das denn die Möglichkeit? Sollten wir die Freunde so gestrebt haben? Na - zum Glück hat Michael im ganzen letzten Jahr die 'Kursanteile' (und damit die Arbeit) in der Gruppe gehabt. Da weise ich erst einmal alle Schuld präventiv weit von mir... :-)

Sommerzeit - Biergartenzeit... Wenn sich die Moerser Gruppe an sonnigen Samstagen kollektiv für den mittlerweile häufig erwähnten Biergarten am Rhein entscheidet, dann heißt das nicht, daß wir uns dort NUR zu Kaffee, Kuchen und Eis niederlassen (außer vielleicht der Harald). Im Gegenteil - z.B. ein Blick auf die Schornsteine auf der anderen Rheinseite bringt das niederheinische Hirn auf allerlei merkwürdige Ideen, und liefert Diskussionsgrundlagen für Wochen...

Wer kennt sie nicht, die Bilder von gesprengten Kaminen in den Nachrichten.

Aber wen hat es je interessiert, warum die (meistens) während des Fallens noch einmal irgendwo (meistens) außerhalb der Mitte brechen. Uns hat das insgesamt drei Samstage beschäftigt - mich selbst sogar zwei Wochen.

Statik hatte ich doch mal, zwei Semester immerhin. Da sollte sich doch etwas 'forthiges' draus machen lassen. Also haben wir die während des Fallens im Kamin auftretenden Kräfte diskutiert und beschrieben. Für diejenigen von uns, die im Juni unseren 4-wöchigen EXCEL-Kurs mitgemacht haben (Ja, so etwas machen wir hier auch - und 'nebenher'), ist der Aufbau einer Tabelle zu entsprechenden Untersuchungen eine ausgezeichnete Übung. Für die 'puren' Forther sollte die Entwicklung entsprechender Berechnungen und die anschließende 'Visualisierung' (am besten mit ZF's neuen Graphikroutinen) eine Herausforderung sein...

Sommerzeit - Rückblickzeit... Nach den Sommerferien, wenn das Wetter schnell herbstlich wird, und das Verbleiben in 'unseren' Räumen wahrscheinlicher, beginnt hier in Moers immer ein neuer Abschnitt mit neuen 'Projekten', neuen Kursen und neuen Themen. Dementsprechend ist die Zeit kurz vor dem Urlaub

(wenn die bis zum Reiseantritt verbleibenden Arbeitstage zu den längsten im Jahr werden) die ideale Zeit für Rückblicke - sozusagen für ein forthiges Silvester.

Mittlerweile ist mein Schreck über die unerwartete Ehrung anlässlich der Jahrestagung 'verdaut'. Ich kann mich sogar darüber freuen - ebenso für unsere Gruppe, wie auch für mich selbst - und mich aufrichtig bei den Menschen bedanken, die mich dieser Ehrung für 'würdig' befunden haben...

Mir wird es immer wichtig sein, für Forth und für die "sozial-philosophischen" Ideen, die ich damit verbinde, zu arbeiten. Deshalb freuen mich die vielen Kontakte, die ich seit dem letzten Sommer durch Forth und über das ZF knüpfen konnte, ganz besonders. Und am meisten freuen mich natürlich die Kontakte, die über den 'großen Teich' reichen und das ZF 'back to the roots' führen.

Henry Vinerts hat Tom Zimmer darauf angesprochen, daß hier in Moers dessen ZF noch immer gerne genutzt wird. Tom Zimmer hat "hardly remembered" das ZF irgendwann als erstes Forth mit sequentiellm Editor entwickelt zu haben. ZF's Quellen hat er allerdings seit langem nicht mehr zur Verfügung. Er war, so hat Henry mir geschrieben, sichtlich verwundert, daß überhaupt noch jemand mit dem ZF arbeitet, und daß wir damit "pretty small applications" entwickeln würden.

Nun, wir würden das ZF gerne einmal 'von Grund auf' überarbeiten, den Assembler als Overlay optional einbinden, redundante Definitionen entfernen, den Editor noch ein wenig komfortabler machen, die Erweiterungen von XZF dauerhaft einbinden und, und, und... Leider fehlt es uns dazu an der notwendigen Zeit. Auch unsere Kurstexte müßten noch einmal komplett überarbeitet werden. Ich bin sicher, daß wir nicht nur den Tom Zimmer am Ende in Staunen versetzen könnten...

F.PRINZ@MHB.gun.de  
(Friedrich Prinz), Juli '95

## Forth-Gesellschaft: Jagen und Sammeln

### Forth mirror at ftp.uni-bremen.de

dku@zarniwoop.pc-labor.uni-bremen.de (Dirk Kutscher) wrote:

" ftp.uni-bremen.de now provides a mirror of Skip Carter's Forth Archive from taygeta.oc.nps.navy.mil. The url:ftp://ftp.uni-bremen.de/pub/languages/programming/forth/Taygeta-Mirror(for direct access)".

No such file or directory. You meant Taygeta-Archive -- that does exist %-)

Cheers,

Jayfar@netaxs.com (Jay Farrell) in comp/lang/forth, Juli '95 in Antwort auf dku@zarniwoop.pc-labor.uni-bremen.de (Dirk Kutscher)

## Forth-Gesellschaft: Kurse

### EXCEL für Einsteiger - Werbung für die FORTHER

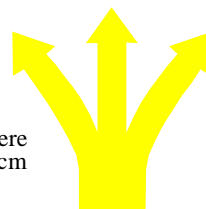
Arbeiten am und mit dem Computer ist eben, zumindest manchmal, doch Mathematik. Und die Mathematik entzieht sich eben immer noch den meisten Menschen einfach dadurch, daß sie sich schlecht - im Sinne des Wortes - "begreifen" läßt. Es sei denn, man nutzt ein Werkzeug, mit dessen Hilfe sich mathematische Zusammenhänge einfach und schnell 'visualisieren' lassen. EXCEL, die Tabellensoftware von MS, ist zwar eigentlich von seinen Entwicklern für etwas andere Aufgaben gedacht. Es stellt aber durchaus ein Werkzeug dar, das die Forderung nach SCHNELLER und EINFACHER VISUALISIERUNGsmöglichkeit erfüllt. Mitglieder der Moerser Forthgruppe hatten angeregt, einen Einsteiger Kurs zu EXCEL durchzuführen. Das haben wir selbstverständlich schnellstmöglich aufgegriffen. Das Institut für Lehre und Bildung der DMT (Deutsche Montan Technologie) hat uns in der Moerser Bergfachschule (wo angehende 'technische Aufsichten' für den Steinkohlenbergbau aber auch 'Umwelttechniker' ausgebildet werden) einen hervorragend ausgestatteten Computerraum zur

Verfügung gestellt. Und schon konnte an vier Samstagen, von 8:00 bis 12:00 der Kurs 'laufen'. Selbstverständlich war das Angebot auch für Nichtforther offen. Die Christel hat eine Lehrerkollegin mitgebracht, die beiden haben SchülerInnen in den Kurs 'geloockt' und einer der Schüler hat seinen Vater davon überzeugt, daß man sich eine solche Gelegenheit nicht entgehen lassen dürfe. Schließlich werden Kurse dieser Art und dieses Umfangs gewöhnlich gegen > 500,- DM angeboten. Bei uns war's natürlich kostenlos! Und das hat gehörig beeindruckt. Da gibt es doch tatsächlich Leute, die am 'heiligen' Samstagmorgen einen Workshop unentgeltlich anbieten und durchführen und andere Menschen an ihrem Wissen 'partizipieren' lassen. "Normal" ist das nicht - aber welcher FORTHER ist das schon...

Fazit: Fröhlich war's, locker und lehrreich (für die Teilnehmer und den Referenten) und das Interesse an den FORTHERn ist geweckt (was wir ganz sicher auszunutzen versuchen werden

Glückauf  
F.PRINZ@MHB.gun.de  
(Friedrich Prinz)

Hier wäre Platz für eine weitere Aktivität von genau 4,46 x 2,4 cm Größe gewesen. Nächstesmal?



Geschichte  
wird gemacht  
- es geht voran!



**Forth-Gesellschaft: ZF back to its roots**

Größer, Schöner, Bunter ? Lieber kleiner, schneller, handlicher...

Der Forthfreund Henry Vinerts aus dem sonnigen Californien, eifriger Leser der VD und Mitglied der FIG SILICON VALLEY, hat bei der Moerser Forthgruppe das ZF angefordert. Hintergrund seiner Bitte war, daß 'alle anderen Systeme' wenig bis gar nicht für Anfänger geeignet sind. Die FIG im Silicon Valley hat aber Anfänger in ihren Reihen, für die Henry Vinerts nach einem geeigneten Forth suchte. Das meint er mit dem ZF gefunden zu haben. Die in Moers im Rahmen der bisherigen Kurse und Vorträge entstandenen Texte will er ins amerikanische übersetzen. 'Seine' FIG hat das ZF lange

'vergessen', schreibt er, und liefert eine Hintergrundinformation am Rande - Tom Zimmer, bisher wohl Mitglied der gleichen Interest Group, wird diese verlassen und nach Austin, Texas gehen.

Soll ich leugnen, daß wir (die Moerser) uns freuen, wenn wir uns heute darin bestätigt sehen, daß wir uns der auch bei den Entwicklungssystemen allenthalben schmerzhaft spürbaren Gigantomanie beständig verweigert haben? Ich versuch's erst gar nicht ;-)

*Glückauf  
Friederich Prinz*

**Forth-Gesellschaft: Microcontrollerverleih**

Seit 1.1.1995 hat Thomas Prinz den Microcontroller-Verleih übernommen. Bis jetzt hat er zwei Anfragen bekommen.

Dem Verleih stehen folgende unterschiedliche Geräte zur Verfügung: Super-8 / Mini-Muck und zur Abschreckung 8052-AH-Basic. Es sind jeweils enthalten eine Leiterplatte mit dem entsprechenden Microcontroller, Netzteil, Beschreibung und Software. Demnächst wird noch ein 68HC11 hinzukommen. Dieser wird zur Zeit von Rafael Deliano zusammengebaut.

Der F65k wird in alter Form nicht mehr hergestellt und ist nicht mehr erhältlich.

Wer dem Verleih Geräte zur Verfügung stellen könnte, sei hiermit ermuntert.

Interessant wären auch ältere Geräte EMUF, MOPS aber mit Unterlagen, Software etc..

Nach Möglichkeit möge man mich zwischen 19:00 Uhr und 22:00 Uhr kontaktieren, außer Montag und am 1. Mittwoch jeden Monats.

*Thomas Prinz  
Microcontroller-Verleih der Forth  
Gesellschaft e.V. und  
FG-e.V. Rhein Neckar-Gruppe  
Adalbert-Stifter-Str.15  
69412 Eberbach a/N  
Tel: 06271 / 2830  
Fax: 06220 / 7065  
Mail: TOPEBAN@RNG.Forth-  
ev.de*

**Forth-Gesellschaft: Vertrieb**

**Neu im Vertrieb:**

- VD95-1 - Diskette zum Forthmagazin
  - VD95-2 "
  - VD95-3 "
  - Dokumentation ANS X3.215 - 1994 im Word 2.0-Format
  - Crossassembler für Z8 und PIC
- von Bernd Hinze
- EFORTH für PIC 1742

je Diskette: 10.-DM

**Ing. Büro Klaus Kohl**  
PF 1173 Mering  
Zeppelinstr. 10  
86415 Mering  
Tel.: 08233 / 30 524  
Fax.: 08233 / 99 71



**Forth-Gesellschaft: Das Forth-Büro**

Das Forth-Büro fällt vor allem durch seine Bescheidenheit auf. Trotz mehrmaliger Rückfrage mochte es nichts aus dem Büroalltag im schönen Unterschleißheim berichten.

Wer etwas über das Forth-Büro wissen möchte, muß wohl doch dort anrufen. Die Adresse steht im Impressum.

**Forth-Gesellschaft: ELRAD in der KBBS**

Immer öfter freuen wir uns, etwas über Forth in der ELRAD zu lesen. Diesmal freuen wir uns außerdem, etwas von der ELRAD in der KBBS, der Stamm-Mailbox der Forth-Gesellschaft, lesen zu können. Seit Juli liegt hier auf der Platte ein Abzug der ELRAD-Mailbox unter /usr/spool/uucppublic/ELRAD/\*. Die Dateien sind besonders für Hardware-Entwickler interessant. Das Inhaltsverzeichnis gibt es im File /usr/spool/uucppublic/elrad.zip und Unix-like in /usr/spool/uucppublic/elrad.gz (beide ca. 17kB). Wer's lieber ausgepackt saugen möchte, findet 60 kB /usr/spool/uucppublic/elrad.txt.

Den Mailbox-Abzug auf CD erhielten nur 8 handverlesene Boxen. Sysop Holger freut sich über

diese Ehre, zumal es seine erste 'goldene' CD ist.

Wir bedanken uns bei der ELRAD und würden uns gerne angemessen revanchieren. Da sich die Fertigstellung der CD der Forth-Gesellschaft ('Die schönsten Forth-Quellen aus zwei Jahrhunderten') leider um ein weiteres verzögert hat, wird die Wahl der zuständigen Gremien wohl auf die Singleauskopplung 'Geh mit mir in die FG' (drei Direktoren im Quartett) fallen, die wegen ihres ungewohnten Blockformats schon manchen Plattenspieler zur Verzweiflung getrieben hat. nAber den Jungs und Mädels von der ELRAD fällt da bestimmt was ein....

*Claus Vogt*

**Forth-Gesellschaft: Forth-Magazin**

**Das Forth-Magazin sucht ...**

**An Sachmitteln fehlen noch:**

- Ein Englisch-Wörterbuch
- Ein Duden, neuer als 1968

An Hardware fehlt noch:

- Ein Rechner, schneller als 386dx33
- Eine Gigabyte-Festplatte
- Ein A4-Scanner

**An Software fehlt noch:**

- Ein DTP-System, teurer als 50 DM

- Ein Archivsystem, vielleicht auf Basis des F-PC-Hypertextes?

**An Beratern fehlen noch:**

- Spezialisten aus dem Satz-, Grafik-, Layoutbereich
- DTP-Erfahrene zur Beratung bei der Neuanschaffung
- Wer kann mir Hinweise zum Belichten von Diskette geben?
- Wer kennt sich mit Macs aus und kann beim Anschaffen helfen

*Claus 030 / 782 81 79*



Puh! Von Sommerloch keine Spur. Mir steht die Arbeit bis zum Kragen. Nun denn, will ich über jüngste Entwicklungen rund um ANS-Forth berichten. Mehr und mehr Forth-Programmierer setzen sich mit dem Standard auseinander und so ist es nicht verwunderlich, daß die Fragen, die bezüglich der

diskutiert. Sogar die von mir betreuten Studenten kommen mittlerweile zu mir und fragen, ob sie ihre Ausarbeitungen nicht in der HyperText Markup Language HTML verfassen dürfen, die Grundlage des World Wide Webs ist. Was hat das nun mit dem Standard zu tun? Nun, seit einiger Zeit ist

*Jetzt geht's ANS FORTH!*  
von Ulrich Hoffmann



eben auch der Final Draft dpANS94 im World Wide Web unter der Adresse <http://taygeta.oc.nps.navy.mil/forth/dpans.html> zu erreichen. Dieses Dokument, das dankenswerterweise von Norman

Auslegung des Standards gestellt werden, zunehmend knifflig werden. Hier seien nur die Punkte "separater Floating Point Stack oder Floating Point auf dem Datenstack" und "Dateneingabe von doppelgenauen Zahlen" genannt... Um diese Diskussionen verfolgen zu können, kann ich nur wieder raten, den Dienst der Forth Gesellschaft zu nutzen und die Nachrichten in der Usenet-Gruppe comp.lang.forth zu verfolgen. Nach anfänglicher Flaute hat sich auch de.comp.lang.forth zu einem regen Diskussionsforum entwickelt, an dem wohl alle Beteiligten viel Spaß haben.

1995 scheint aber auch das Jahr des Internets zu sein. An allen Ecken wird groß über WWW und Netsurfen berichtet und

Smith (smithn@orvb.saic.com) in HTML aufbereitet wurde, nutzt die Hypertextmöglichkeiten von HTML aus, um die im Final Draft vorhandenen Querverweise zu Hyperlinks zu machen. Sehr bequem kann man so mit dem WWW-Browser navigieren und das zuweilen lästige Blättern und Suchen nach den benötigten Querverweisen entfällt. Wer online mit dem Internet verbunden ist, braucht nur obige Adresse anzuwählen und kann schon im Final Draft stöbern. Leider ist Online-Internet zur Zeit noch ein recht kostspieliges Unterfangen und damit wohl nur etwas für Freaks. Um aber per WWW-Browser im Final draft zu stöbern ist es prinzipiell gar nicht nötig, eine direkte Internet-Verbindung zu haben. Ko-

piert man die nötigen Files auf seinen lokalen Rechner, so kann der Hypertext auch ohne bestehende Verbindung - also offline - gelesen werden. Dazu bietet sich z. B. der WWW-Browser Netscape an, den es sowohl unter Windows als auch für Unix und den Mac gibt. Alles was letztlich fehlt, ist ein Freiwilliger mit Internet-Zugang, der die Mühe nicht scheut, die für den Standard relevanten Files aus dem Internet zu kopieren und anderen zur Verfügung zu stellen...

Die F-PC Anhänger höre ich gerade murmeln "Ach, wozu HTML. Das hat F-PC alles ja schon lange!". Richtig! Auch für F-PC gibt es Dank C. H. Ting einen Satz Files mit dem Draft-Proposed-Standard No. 6 (dpANS6), die als Text-Files mit F-PC Hyperlinks ausgeführt sind. Damit ist beim Programmieren in F-PC der Standard nur ein paar Mausklicks oder Tastendrucke entfernt. Meine Implementierung des Core und Core Extensions word sets für F-PC ist ja seit längerer Zeit erhältlich, so daß erste ANS-Forth Schritte auch mit F-PC gegangen werden können. Die Implementierung weiterer word sets (File, String, Double, Facility, ...) befindet sich nun eigentlich schon zu lange in der Beta-Testphase und wartet auf ein baldiges Release.

Die Zahl der ANS-Forth Standard Systeme steigt stetig. Brad Rodriguez (bj@genie.com) hat sein ANSI Camel-Forth für Z80, 8051 und 6809 vorgestellt und zeigt damit, daß auch kleine Forth-Systeme für Single-Board-Computer dem Standard genügen können. Diese Systeme sind für den privaten Gebrauch und für Ausbildungszwecke kostenlos zu benutzen. Einen Blick auf ANSI Camel-Forth zu werfen ist allemal sinnvoll, um sich an dessen Eleganz zu erfreuen. Brad ist halt ein Meister seines Faches.

```
:place( c-addrln c-addr-- )
\ Put the string given by C-ADDRLEN
\ as counted string at address C-ADDR'
OVER>R ROT OVER CHAR+R> CHARSMOVE C!;
```

```
:skip( addrln c-- addr' len' )
\ Within the string given by ADDR and LEN
\ skip leading occurrences of character C. Return the
\ remaining string denoted by ADDR' and LEN.
>R
BEGIN( addrln )
  DUP
  WHILE( addrln )
    OVERC@ R@ =
  WHILE
    1 /STRING
  REPEAT THEN
  R> DROP;
```

```
:scan( addrln c-- addr' len' )
\ Within the string given by ADDR and LEN
\ scan for the first occurrences of character C. Return
\ the remaining string denoted by ADDR' and LEN.
>R
BEGIN( addrln )
  DUP
```

```
WHILE( addrln )
  OVERC@ R@ =
  WHILE
    1 /STRING
  REPEAT THEN
  R> DROP;

:have( <spaces>ccc<spaces> flag ) BL WORDFINDNIP;
: ?exit( flag-- )
  POSTPONEIF POSTPONEEXIT POSTPONETHEN; IMMEDIATE
: umin( u1 u2-- u3 ) 2DUP U< IF DROPEXIT THEN NIP;
: umax( u1 u2-- u3 ) 2DUP U< IF NIP EXIT THEN DROP;
: -rot( x1 x2x3-- x2x3x1 ) ROT ROT;
: on( a-addr-- ) TRUE SWAP!;
: off( a-addr-- ) FALSE SWAP!;
: crash( -- ) TRUE ABORT "uninitialized execution vector";
```

```
:Defer( <spaces>name- )
  CREATE [] crash,
  DOES> ( !*x--!*x ) @ EXECUTE;
```

```
:is \ Interpretation: ( xt <spaces>name- )
  \ Compilation: ( <spaces>name- ) Runtime: ( xt-- )
  '>BODY STATE @ IF POSTPONELITERAL POSTPONE! EXIT THEN
  !; IMMEDIATE
```

Nun zum heutigen Abschluß noch die Definition ein paar mir lieb gewordener Worte, die im Standard zwar nicht auftauchen, die aber als Folklore weit verbreitet sind und sich außerdem leicht in ANS-Forth formulieren lassen. In den Definitionen von SKIP und SCAN zeigt sich auch gleich einmal, wie die das letzte mal besprochenen WHILE Schleifen mit mehreren Ausgängen nützlich eingesetzt werden können.

*Viele Grüße und bis zum nächsten Mal.*





## Serie: PC-Meßtechnik, Teil II

# Programmierung des Timer0-Interrupt

Autor: Klaus Kohl / BMC Systeme GmbH  
 Zeppelinstr. 10; 86406 Mering; Tel.: 08233 / 30 524

Wenn nicht durch externe Hardware nachgeholfen wird, hat man im PC nur durch die Verwendung des Hardware-Timers 0 eine Quelle für einen zeitsynchronen Interrupt. Der folgende Artikel beschreibt, hier noch ohne direkten Bezug zu einem Meßproblem, die Verwendung dieses auch vom Betriebssystem genutzten Interrupts in eigenen FORTH-Programmen. Alle Beispiele wurden im KK-FORTH für PC erstellt. Da dieses FORTH an volksFORTH angelehnt ist und die Programmbeispiele sehr kurz sind, dürfte eine Anpassung an andere FORTH-Versionen einfach sein. Zur Erleichterung der Anpassung hier einige Hinweise:

- Das Zeichen & kennzeichnet eine dezimale Schreibweise
- Um feste Adressen zu erhalten, wurden Datenfelder mit CREATE und ALLOT angelegt
- CS@ liefert Codesegment des FORTH-Programmes

Stichworte: Meßtechnik Timer PC

## Programmierung des Timer 0 im PC

Wie den meisten PC-Anwendern bekannt ist, verfügt der PC über einen Timerbaustein. Der dazu meist verwendete 8253 hat drei unabhängige 16Bit-Timer, die mit einem Grundtakt von 1,193180MHz angesteuert werden.

Da der Timer 1 nicht in allen PC's zur Verfügung steht und der Timer 2 für die Tonausgabe reserviert bleiben sollte, blieb für eine interruptgesteuerte Meßdatenerfassung nur der Timer 0. Da dieser Timer auch vom Betriebssystem z.B. Verwaltung der Softwareuhr verwendet wird, müssen eigene Routinen nach entsprechenden Zeitintervalle auch wieder die Betriebssystem-Interruptroutine aufrufen.

Das folgende Programmbeispiel verändert das Verhalten des Timers 0, ohne daß gleichzeitig die Interrupts umgeleitet werden. Es wird deshalb die Softwareuhr falsch laufen und evtl. vorhandene Bildschirmschoner mit noch nie gesehenen Eigenschaften aufwarten.

```
\ Timer0 auslesen
: ti0@  ( -- val )
  $40 pc@ $40 pc@ flip or ;

\ Timer0 verändern
: ti0!  ( div -- ) \ Timer0 auf neuen Teilerwert stellen
  %00110110 $43 pc! \ Timer0; LSB/MSB; Mode3; binär
  dup $40 pc! flip $40 pc! \ LSB/MSB ausgeben
  %00000000 $43 pc! ;
```

Timer	Verwendung
0	System-Timer mit 18,2Hz (1,19MHz/65536) über Interrupt
1	früher zusammen mit DMA-Kanal für DMA- Refresh verwendet
2	meist für Frequenzerzeugung des Piepsers verwendet

### Portadresse Bedeutung

\$40	Daten für/von Timer 0
\$41	Daten für/von Timer 1
\$42	Daten für/von Timer 2
\$43	Steuerwort

Das Auslesen des Timers ist unkritisch und liefert den aktuellen 16Bit-Wert, der von 0 bis zum maximalen Wert-1 geht. Bei der Standardeinstellung dauert es genau  $65536 / 1,193180\text{MHz} = 54,9254$  ms bis der Zähler von 0 bis 65535 läuft.

Bei der Veränderung des Timers muß auch das Steuerregister angesprochen werden. Der FORTH-Befehl **TI0!** will nur die Obergrenze neu festlegen und stellt deshalb den auch vom Betriebssystem verwendete Mode 3 ein. Andere Betriebsarten sind in den entsprechenden Unterlagen (siehe Literaturverzeichnis) zu finden. Da der Timer vom DOS normalerweise nicht umprogrammiert wird, bleibt die neue Einstellung auch nach dem Verlassen des Programmes erhalten. Eine einfache Anwendung wäre z.B. das Herunterbremsen des Computers. Bei einem Wert unter **\$100** wird sogar ein 486DX2 so langsam, daß man bei **DIR** mitlesen kann. Zurückgestellt wird der Timer dann wieder durch einen Hardware-Reset oder durch **0 ti0!** .

## Interruptprogrammierung in FORTH

Nun zur praktischen Anwendung des Timers in der Meßdatenerfassung. Da man nicht die gesamte Zeit auf den Ablauf des Timers warten will, ist die Einbindung eigener Routinen in die Interruptsteuerung des Betriebssystems notwendig. Hier noch einige Hintergrundinformationen dazu:

Wenn der Timer 0 abgelaufen ist, wird der Interrupt \$08 ausgelöst. Falls Interrupts erlaubt sind, beendet der Prozessor seinen gerade laufenden Befehl und merkt sich Adresse des nächsten Befehls auf dem Prozessorstack (entspricht meist dem Datenstack in FORTH). Gleichzeitig blockiert der Interruptcontroller alle weiteren Interrupts, bis er wieder durch Einschreiben des Wertes \$20 in Portadresse \$20 freigegeben wird. Aus den Speicheradressen \$0020..\$0024 des Segmentes \$0000 wird dann die Adresse der zugehörigen Interruptroutine geholt und gestartet. Der normale DOS-Timerinterrupt wird dann neben einigen internen Aktionen auch noch den Interrupt \$1C aufrufen, bevor er den Interruptcontroller wieder freigibt und das unterbrochene Hauptprogramm fortsetzt. Der Interrupt \$1C (Adresse dazu ab \$0000:\$0070 im Segment \$0000) wird z.B. vom BASIC zur Einbindung eigener Routinen wie die Steuerung der Tondauer genutzt.

Da wir vermeiden wollen, daß das normale Timing des Betriebssystems verändert wird, werden wir an Stelle des normalen Interrupts \$08 eine eigene Routine einbinden. Diese hat folgende Aufgaben:

- Verwendete Register retten
- Eine eigene Routine ausführen
- Anzahl der Interrupts zählen und
  - alle 52 ms die alte Interruptroutine des Betriebssystems anspringen
  - sonst Interruptcontroller freigeben und zum Hauptprogramm zurückkehren

Um das Prinzip dieser Interruptsteuerung zu zeigen, werde ich hier aber keine aufwendige Abtastroutine einbinden, sondern einfach eine Variable hochzählen, die dann mit dem Hauptprogramm abgefragt und ausgegeben werden kann. Leider kann ich hier nicht mehr die Verwendung eines Assemblers vermeiden. Es handelt sich um einen leicht modifizierter 8086-Assembler mit folgenden Eigenheiten:

- Assemblerbefehle werden mit Komma beendet ( z.B. CLI)
- Befehlsaufbau: Quelle mit Adressierungsart Ziel mit Adressierungsart Befehl
- Adressierungsarten:
  - Register (AX, BX, ...) brauchen keine zusätzlichen Angaben
  - Mit # nach dem Wert werden Konstanten angegeben
  - Mit #) werden direkte Adreßangaben gekennzeichnet
  - Segmentprefix werden unmittelbar zuvor mit Segment seg, angelegt
- Mit Kontrollstrukturen können bedingte Kurzsprünge angegeben werden
  - BEGIN, ... immer mit abschließenden Komma
  - Bedingungen (z.B. 0=) werden genau invertiert als Sprungbefehl kompiliert

```

\ Timerwert n = 1193182Hz / Interruptfrequenz
\ &1193 Constant #divt      \ Timerwert für 1000.15 Hz
&298 Constant #divt      \ Timerwert für 4003.95 Hz
\ DOS-Interrupt alle t = 65536 / n Interrupts
\ &85 Constant #div        \ Vorteiler für 1000.15 Hz
&220 Constant #div        \ Vorteiler für 4003.95 Hz
;

\ Arbeitsvariablen
2Variable oldint$08      \ Zeiger auf Standardroutine
Create int8div 2 allot   \ Vorteiler mit fester Adresse
;

\ Interrupt ein-/ausschalten
Code di ( -- ) \ Interrupt abschalten
cli, next, End-Code
Code ei ( -- ) \ Interrupt einschalten
sti, next, End-Code
;

\ Eigenes Zählerprogramm
Create counter 2 allot   \ Zählervariable
PROC userint ( -- ) \ Zählerinterrupt
pushf,                  \ Flags retten
cs seg, counter #) inc, \ Zähler erhöhen
cs seg, 1 # int8div #) sub, \ Teiler erniedrigen
0= IF,                  \ Null erreicht ?
cs seg, #div # int8div #) mov, \ alter Wert
popf,                  \ Flag holen
0 $20 I@ 0 $22 I@ # far jmp, \ DOS-Interrupt
THEN,
ax push,               \ Register AF retten
$20 # al mov, $20 #) byte out, \ Interruptcontroller freigibt
ax pop,                \ AF holen
popf,                 \ Flags holen
iret,                 \ Rückkehr aus Interrupt
End-Proc
;

\ Interrupts auf eigene Routine und zurück zum DOS stellen
: inton ( -- ) \ Interrupt umleiten
di #divt ti0!          \ Timer beschleunigen
#div int8div !        \ Zähler zurückstellen
0 $20 I2@ oldint$08 2! \ Alte Vektoren merken
cs@ (int++ 0 $20 I2! \ Neue Vektoren setzen
ei ;
: intoff ( -- ) \ Interrupt zurückstellen
di $0000 ti0!         \ Timer zurückstellen
oldint$08 2@ 0 $20 I2! \ alte Interruptroutine
ei ;
;

\ Hauptprogramm
: main ( -- ) \ Anzeigen der Zählerwertes im Interrupt
0 counter !          \ Zähler auf 0
inton                \ Interrupt ein
BEGIN
cr counter @ u. key?
UNTIL
intoff ;

```

Wenn das Programm mit **MAIN** gestartet wird, erfolgt eine laufende Ausgabe des aktuellen Wertes in **COUNTER**. Da dieser Zähler vom Hauptprogramm nicht verändert wird, ist dies eine Bestätigung des laufenden Interruptprogrammes. Man kann jetzt nach Rettung der verwendeten Register auch andere Routinen einbinden, die einen A/D-Wandler abfragt oder einen Digitalport verändert. Hier noch eine Empfehlung vor dem Start dieser Versuche:

- Schreib-Caches für Laufwerke und Harddisks sollten abgeschaltet sein (z.B. Smartdrive)
- Windows möglichst nicht laufen lassen
- RESET-Schalter in Griffweite halten

## Literatur zur Timerprogrammierung

- (1) c't-Serie PC-Bausteine; Rund um den Timer. c't 1988, Heft 4, Seite 196ff
- (2) Thom Hogan: Die PC-Referenz für Programmierer. Microsoft Press
- (3) Frank van Gilluwe: The undocumented PC. The Andrew Schulmann Programming Series. Addison-Wesley Publishing Company



# DOER & MAKE kurz vorgestellt

von Thomas Höhenleitner

th@graphikon.com; Fax: 030-42104750 Tel: 030-42104733  
Graphikon GmbH; Mandestr. 16;10409 Berlin

Wenn man eine Funktion ausführt, ist im allgemeinen der Algorithmus derselben fest vorgegeben. Das Funktionsverhalten ergibt sich allein aus den übergebenen Daten und bestimmten Variablen. Eine Funktion beginnt normalerweise an einer definierten Adresse im Speicher. Der Funktionsaufruf ist letztendlich ein Sprung des Prozessors an diese Funktionsadresse. Ein indirekter Sprung, ein Sprung an die in einer Variablen gespeicherte Adresse ist auch möglich. Wenn man die in dieser Funktionsvariablen befindliche Adresse austauschen kann, stellt der indirekte Sprung eine vektorisierte Funktion dar. Damit ist das Verhalten von Funktionen frei veränderlich, also umschaltbar oder vektorisierbar.

*Stichworte: DOER, MAKE, vektorisierte Funktion*

Nachfolgende Informationen sind für interessierte Leser, die diese beiden Worte nicht kennen und das Buch "THINKING FORTH" von Leo Brodie weder in Deutsch noch in Englisch griffbereit haben. Weiter unten wird es dann für mehr Leute interessant.

## FORTH Mikrokurs

In FORTH-Implementationen sind die Funktionsadressen (**execution token**) oft im sogenannten Fadencode (direct, indirekt oder token threaded code) aufgefädelt und der Prozessor ist der innere Interpreter, der diesen Fadencode abarbeitet. Normalerweise besorgt der äußere Interpreter nach der Wörterbuchsuche die Adresse einer benannten Funktion, den **execution token** des FORTH-Wortes. Dann erfolgt ein Maschinencode-Call auf diese Adresse und der innere Interpreter wird angeworfen, falls es sich um ein Secondary handelt, also ein i.a. mit dem Doppelpunkt definiertes Wort. Der innere Interpreter rattert nun den Fadencode entlang. Dabei wird bei jedem Primitive (Assemblerwort) dessen code sofort ausgeführt. Fadencodemotionen von Secondaries landen auf dem Returnstack und dann geht es in diesen genau so weiter. Irgendwann ist der letzte Maschinencode erreicht. Wenn das Fadencodende eines Secondary erreicht wird, steht dort z.B. ein vom ; kompiliertes UNNEST (F83) oder EXIT (comFORTH) welches vom Returnstack die

letzte Secondaryadresse holt. Dann geht alles wieder eine Ebene höher weiter. Wenn der Returnstack leer geworden ist, landen wir wieder im äußeren Interpreter und erfreuen uns des OK des Systems, falls wir weich gelandet sind.

## Funktionsaufruf in FORTH

Normalerweise brauchen wir uns um diese Dinge wenig kümmern - FORTH wird's schon richten. Aber manchmal will man mal tricksen oder sonstigen Unsinn anstellen. Da ist CALL eins dieser nützlichen comFORTH Worte. Es ermöglicht einen direkten Fadencoderaufruf ohne weiteres Brimborium. Dabei kapselt es wie z.B. auch JUMP (ähnliches comFORTH-Wort) die FORTH-Implementationsart. Bei Verwendung dieser Worte hängt also der FORTH-Code nicht an der Fadensorte, ist also portabel. Seine Nützlichkeit zeigt sich z.B. im Quelltext REMAKE.CFW. Fadencodenden können nämlich einfach so im Speicher rumhängen und darauf warten, daß CALL ihre Adresse aufruft. Die Adressen dieser Fadencodestücke sind es, die im Body der Doer liegen, austauschbar sind und bei Verwendung von REMAKE auf Stacks gerettet werden.

## Leo Brodies Erfindung: DOER und MAKE und ;AND

Man definiere ein Wort, dessen Verhalten erst später zugewiesen wird. Bis

dahin macht dieses Wort, auch einfach Doer genannt, nichts außer dazusein und sich in andere Worte hineincompilieren zu lassen.

DOER GRUSSok  
GRUSSok  
: FRAGE GRUSS." Wieviel Uhr ist es?"; ok  
FRAGE Wieviel Uhr ist es? ok

Nun kann man einen Gruß einbauen:

MAKE GRUSS." Hallo!"; ok  
GRUSS Hallo! ok  
FRAGE Hallo! Wieviel Uhr ist es? ok

Das ganze funktioniert auch mit Compilation.

: MORGENS MAKE GRUSS." Guten Morgen!"; ok  
: ABENDS MAKE GRUSS." Guten Abend!"; ok  
MORGENS FRAGE Guten Morgen! Wieviel Uhr ist es?  
ok  
ABENDS FRAGE Guten Abend! Wieviel Uhr ist es? ok

Dann gibt es noch ein ;AND welches es ermöglicht, mehreren Doern innerhalb eines Wortes mit MAKE ein Verhalten zuzuweisen und unabhängig davon auch normalen Code auszuführen.

DOER PARTNER ok  
: AUFSTEHEN GRUSS PARTNER." aufstehen!"; ok  
: MUFFLIG MAKE GRUSS ;AND MAKE PARTNER  
" Eh, ";AND AUFSTEHEN; ok  
MUFFLIG Eh, aufstehen! ok  
: NETT MAKE GRUSS." Guten Morgen ";AND  
MAKE PARTNER." Lieblich, ";AND AUFSTEHEN;  
ok  
NETT Guten Morgen Lieblich, aufstehen! ok

## Implementation

Wer comFORTH oder fieldFORTH benutzt hat diese Worte bereits. Für

- FORTH-83 Modell Laxen/Perry/Harris
- FORTH-83 Lab.Micro PC/FORTH 2.0
- FIG-Modell
- 79er-Standard MVP-FORTH

sind im Anhang B des Buches von Leo Brodie "In FORTH denken" die Public Domain Quelltexte (jeweils 1 Screen) angegeben. Eine Implementation in andere FORTH-Systeme dürfte damit nicht schwer fallen. Vielleicht ist die VD so nett und druckt die PD Sourcen ab. Wer gut ist und auf Schnelligkeit Wert legt, definiert den Laufzeit-Part seiner DOER nicht mit DOES>, sondern wie in comFORTH

vorgemacht mit `;CODE ... END-CODE`.

## PREDEFINED im comFORTH und fieldFORTH

So schön diese Doer sind, möchte man doch bald mehr und mehr damit anfangen. Das Wort **PREDEFINED** sorgt dafür, daß das bisherige **DOER**-Verhalten nicht verloren geht, sondern mit ausgeführt wird. Beispielsweise gibt es im comFORTH ein Wort **ANSWER**, welches sich im Vokabular **COM-FORTH** befindet. Dieses ist ein **DOER** und für die Systemantwort auf ein Enter zuständig. Beispielsweise sagt es i.a. **okay** und zeigt den Parameterstack an - sehr nützlich. Wenn aber auch Interesse am Zustand des Stringstacks besteht, dann sagt man einfach:

```
MAKE ANSWER PREDEFINED."S;
```

Leider funktioniert das nur beim interaktiven **MAKEn**. Der Grund dafür ist eine gewisse Unklarheit was denn mit **PREDEFINED** gemeint ist, wenn man innerhalb einer Definition **MAKE** benutzt. Das einzige was da zur Verfügung steht um in den Body des gerade erzeugten Babies eingebaut zu werden, ist das Verhalten des betreffenden **DOER's** zur Compilezeit. Während der Laufzeit kann dieser **DOER** ja gerade sonst was anstellen. Um bei dieser Unklarheit bösen Bugs das Leben schwer zu machen, darf **PREDEFINED** nur beim interaktiven **MAKEn** benutzt werden. **PREDEFINED** kommt übrigens aus der OOP-Implementations Kiste der Rostocker. Riecht ja auch stark nach Vererbung.

## REMAKE und UNREMAKE

Wenn man sich nun der Stringstack-anzeige genügend erfreut hat und das alte Verhalten zurückhaben will hat man erstmal Pech. **MAKE ANSWER PREDEFINED**; hilft nicht zurück, denn **PREDEFINED** enthält nun per Definition bereits **."S**. An dieser Stelle kann man das **ANSWER** nur wieder völlig neu **MAKEn**, was hier noch gehen mag, aber bei komplizierteren Worten sehr anstrengend werden kann; insbesondere wenn diese als Haken im

System liegen und man deren Quelltext nicht hat. Sagt man also vorher nicht **MAKE** sondern

```
REMAKE ANSWER PREDEFINED."S;
```

hat man den gleichen gewünschten Effekt aber nun die Möglichkeit

```
UNREMAKE ANSWER
```

zu sagen und den jungfräulichen Zustand des Systems (fast) wieder zu haben. Wenn man **REMAKE** wie das **MAKE** in ein Wort einbaut ist darauf zu achten, daß dieses Wort kontrolliert oft aufgerufen wird bzw. immer wieder **REMAKEd** wird, da sonst der zugehörige Stack überläuft.

**PS:** Es soll nicht verschwiegen werden, daß auch **DEFER** und **IS** für die Vektorisierung von Funktionen weit verbreitet sind. Der wesentliche Unterschied zu **DOER** und **MAKE** ist:

• **DEFER**-Worte enthalten **execution token**

• **DOER**-Worte enthalten Zeiger auf Fadencode

Für jedes Verhalten eines **DEFER**-Wortes muß ein eigenes Wort definiert werden, um den zuzuweisenden **execution token** zu bekommen, was zur Namensinflation beiträgt. **DOER**-Worte können innerhalb einer Definition ein Stück Fadencode generiert und dessen Adresse eingetrichtert bekommen ohne dafür ein neues Wort zu brauchen. □

### Listing 1: DOER.SCR Definition von DOER/MAKE

entnommen aus "In Forth denken" L. Brodie, Anhang B  
Beispiele für andere Forth Implementationen sind dort auch zu finden.  
Dieser Screen läuft z.B. auf volksForth83 rev 3.81.41 und F83

```
0 ( DOER/MAKE Schattenblock )
1 NOTHING Ein Non-Operator, der nichts tut.
2 DOER Definiert ein Wort mit vektorfaehigem Verhalten.
3 MARKER Sichert addr fuer den optionellen Fortsetzungszeiger
4 (MAKE) Bringt die Adresse von weiterem CODE in das Parameter-
5 feld eines DOER-Wortes
6 MAKE Wird interpretativ benutzt! MAKE DOER-Name ForthCode ;
7 oder in einer Definition:
8 : Def MAKE DOER-Name Forth-Code ;
9 ;AND Gestattet die Fortsetzung der Definition mit MAKE.
10 UNDO Gebrauch: UNDO DOER-Name ; stellt die Ausführung
11 sicher
12
13
14
15
```

```
0 \ DOER/MAKE FORTH83 Modell Laxen/Perry/Harris
1 : NOTHING ;
2 : DOER CREATE [] NOTHING >BODY , DOES> @ >R ;
3 VARIABLE MARKER
4 : (MAKE) R> DUP 2+ DUP 2+ SWAP @ >BODY !
5 @ ?DUP IF >R THEN ;
6 : MAKE STATE @
7 IF COMPILE (MAKE) HERE MARKER ! 0 ,
8 ELSE HERE [COMPILE] ' >BODY ! [COMPILE] ]
9 THEN ; IMMEDIATE
10 : ;AND COMPILE EXIT HERE MARKER @ ! ; IMMEDIATE
11 : UNDO [] NOTHING >BODY [COMPILE] ' >BODY ! ;
12
```



Zu diesem Beitrag auf Diskette:

DOER .SCR 8,192 .a.. 11-09-95 Doer nach Leo Brodie (Screens)  
DOER .LST 1,195 .a.. 11-09-95 dasselbe für Drucker  
DOER .P2F 4,020 .a.. 11-09-95 dasselbe zum Lesen  
REMAKE .CFW 5,245 .... 14-07-95 Doer verdrehen  
REMAKET .CFW 386 .... 14-07-95 Beispiel zu REMAKE.CFW  
STACKS .CFW 2,638 .... 25-07-95 Kontextbezogene Stacks

## DOER drehen

Dies ist eine Idee bezüglich der DOER/MAKE-Erfindung von Leo Broodie. DOER, MAKE und ;AND sind weiter oben beschrieben und Bestandteil manchen FORTH Systems (z.B. comFORTH und fieldFORTH von FORTEch). Es sind keine Standardworte. Einigen FORTHler(inne)n sind sie gewiß vertraut. Wer einmal daran gewöhnt ist, mag sie nicht mehr missen.

Im comFORTH gibt es sehr viele Doer im System. Das sind Haken, wo man sich einklinken und das Systemverhalten nach Herzenslust und eigenen Wünschen bis zur Unkenntlichkeit verbiegen kann. Ein typischer Vertreter ist z.B. das Wort ANALYZER innerhalb von INTERPRET. Dieses stellt fest, ob es sich bei einem aus dem Eingabestrom separierten String um ein Wort, einen Dateinamen, (zweites Zeichen u.a. ein Doppelpunkt), ein Zeichen ('a') eine doppelgenaue oder einfach genaue Zahl handelt. Dieses Verhalten will man manchmal temporär verändern. Wenn man aber mit dem Speichern des bisherigen DOER-Verhaltens keinen Streß haben möchte und vielleicht auch noch Wert auf reentranten Code legt, dann ist REMAKE sehr nützlich.

Die Listings dokumentieren sich weitestgehend von selbst. Viel gibt's dazu nicht zu sagen. Ich habe mich um ausführliche Kommentierung des Quelltextes bemüht. Das ganze kann man bestimmt noch besser hinkriegen. Dann wird daraus wohl so etwas wie dynamische Vererbung. Dieses Feld überlasse ich aber gern Gurus wie Egmont Woitzel.

Wer comFORTH für Windows besitzt dürfte die Files problemlos laden können. Ansonsten kann man auch nur die eigentlichen Quelltexte laden. Es müßte ziemlich systemunabhängig funktionieren.

Das Wort CALL ist ein comFORTH-Wort und könnte das einzige fehlende Wort sein. Es hat die Funktion, ein Stück Fadencode ab Adresse auszuführen. Es ist systemspezifisch und

kann etwa so in Assembler Code definiert werden:

```
CODE CALL (ps: addr ==>)
      (Ausführen Fadencode ab Adresse addr)
...   \Instruktionspointer zum
      \Returnstack bringen
...   \Instruktionspointer mit addr laden
...   \Stack adjustieren
...   \Sprung nach NEXT
END-CODE
```

Die kontextbezogenen Stacks sind mitentstanden, da sie für das REMAKE gebraucht werden. Alle gedrehten Doer auf einen gemeinsamen Stack zu retten hat den Nachteil, beim Zurückdrehen genau auf die Reihenfolge achten zu müssen. Jetzt bekommt jeder Doer seinen eigenen Stack, einen auf seinen Kontext bezogen Stack. Damit aber nicht zuviel Speicherplatz flöten geht, wird nur dann ein Stack angelegt, wenn der entsprechende Doer gedreht wird. Die Drehtiefe ist im allgemeinen gering. STACKS.CFW wurde ausfaktoriert, da es sich hierbei um einen separat verwendbaren Konstrukt handelt.

Wie das ganze funktioniert, dürfte am Quelltext zu erkennen sein. Die Grundidee ist, daß jeder Stack anhand einer doppelgenauen Zahl identifiziert wird. Diese besteht zur einen Hälfte aus der Stackklassennummer (einfach die CFA) und zur anderen Hälfte wiederum einer CFA, der des jeweiligen DOER's. Es

kommt ja nur auf Eindeutigkeit an. Sozusagen wird erst ein Handle konstruiert und dann dazu ein Stack aufgemacht. Für die Doerdreherei werden nur die drei Zugriffsworte PUSH, TOS und POP gebraucht. Wer z.B. ein DROP haben möchte, kann es sich ja selber basteln.

Für jeden zu drehenden Doer wird also ein separater Stack benutzt. Dieser Stack wird einmalig beim ersten REMAKE für den jeweiligen Doer angelegt und bleibt in seiner Benutzung für den Anwender unsichtbar.

HISTORICAL und [HISTORICAL] sind Zugaben zum Experimentieren und für Leute, deren FORTH kein PREDEFINED hat. Eine Anmerkung ist vielleicht nötig:

Wenn man in einem Wort, also im Compile-Mode, ein REMAKE mit HISTORICAL verwendet, darf man den gedrehten Doer nicht einfach noch einmal mit HISTORICAL REMAKEN. Dann gibt's eine Rekursion, da per Definition immer das aktuelle Laufzeitverhalten geholt wird. Sicherheiten sind nicht eingebaut. Wer den Stackbereich überschreitet . . . . .  
 . . . . .  
 . . . . .(ABSTURZ!)

*Viel Spaß! Thomas Höhenleitner*

### Listing: STACKS.CFW

```
\ STACKS.CFW
\ =====
\ Kontextbezogene Stacks
\ Die Anwendung kann man sich in REMAKE.CFW ansehen.
\ .....
\ Bearbeiter: Thomas Höhenleitner
\ Stand: 25-Jul-95
\ © 1994 graphikon GmbH
\ =====

{ORDER STRING ALSO OS ALSO FORTH DEFINITIONS
BASE @ DECIMAL

HAVE stacks.cfw 0=
.IF

CR .( ...lade kontextbezogene Stacks )
CR

CREATE stacks.cfw

\ .....

: STACK ( ps: n ==> )( ib: name ==> )( definiert Einzelstack name der Tiefe n )
  CREATE ,
  DOES> ( ps: ==> dhd )
  DUP ;

: STACKS ( ps: n ==> )( ib: name ==> )( definiert Stackklasse name der Tiefe n )
  CREATE ,
  DOES> ( ps: ==> hdl ) ;

100 CONSTANT #STACKS \ so viele Stacks sollten genügen
```



# DOER&MAKE

```
VARIABLE STACKS# \ Anzahl der erzeugten Stacks
0 STACKS# !

4 CELLS CONSTANT FG \ Feldgröße der Stackindexliste
\ dhdl (2) Stackhandle
\ saddr (1) Stackadresse
\ sprt (1) Stackpointer

CREATE SINDEX ( Indexfeld für die angelegten Stacks, letztes Feld = 0, 0 0 )
#STACKS FG * ALLOT
SINDEX #STACKS FG * ERASE

: NEWSTACK ( ps: dhdl ==> )( Baut Stackhandle dhdl in SINDEX ein. )
  DUP @ >R \ Stackgröße auf Returnstack
  1 STACKS# @+! \ Stackzahl und inkrementieren
  FG * SINDEX + \ SINDEX-Ende
  R! 2! \ dhdl eintragen
  HERE DUP R>
  2 CELLS + 2! \ Stackadresse und -pointer eintr.
  R> ALLOT ; \ Platz reservieren

: SIADDR ( ps: n ==> addr )( Liefert Adresse des Feldes n )
  FG * SINDEX + ;

: SIEMPTY? ( ps: n ==> flag )( TRUE, wenn Feld n leer ist )
  SIADDR 2@ 0 0 D= ; \ wird am Inhalt 0 erkannt

: SINDEX# ( ps: dhdl ==> n )( liefert Indexnummer in SINDEX )
  0 0 DO \ "Endlos"schleife mit Schleifenzähler
    2DUP I SIEMPTY?
    IF \ ist Stackhandle dhdl nicht in SINDEX,
      2DUP NEWSTACK \ wird dhdl automatisch eingebaut
    THEN
      I SIADDR 2@ D=
      IF
        2DROP I LEAVE \ Hier ist in jedem Fall einmal Ende
      THEN
        LOOP ;

: SPOINT^ ( ps: dhdl ==> addr )( Adresse des Stackpointers von Stackhandle dhdl )
  SINDEX# SIADDR 3 CELLS + ;

: TOS ( ps: dhdl ==> n )( Liefert eine Kopie des obersten Elementes des Stacks dhdl )
  SPOINT^ @ 1 CELLS - @ ;

: PUSH ( ps: n dhdl ==> )( Bringt n zum Stack dhdl )
  SPOINT^ 1 CELLS SWAP @+! ! ;

: POP ( ps: dhdl ==> n )( Holt oberstes Element vom Stack dhdl )
  SPOINT^ -1 CELLS SWAP @+! 1 CELLS - @ ;

\ -----
\ ( ...fertig )
CR
.THEN \ --- stacks.cfw

2 UMAX BASE ! }ORDER
```

```
CR .( ...generische Definitionen )

VARIABLE [STATE] ( Merkmalszelle für Systemzustand bei Ausführung von REMAKE )
( Wird von HISTORICAL benutzt. )

: REMAKE ( ib: name ==> )( weist einem DOER ein neues Verhalten zu )
( Altes DOER-Verhalten wird gemerkt, Wiedereintrittsfähig. )
( REMAKE kann als fast uneingeschränkter Ersatz für MAKE fungieren. )
( Es bewahrt den bisherigen DOER-Faden jedoch auf. )
( Es ermöglicht die Abarbeitung des bisherigen DOER-Kodes an )
( beliebiger Stelle im neuen DOER-Kode. )
( PREDEFINED und ;AND dürfen ebenfalls mit REMAKE benutzt werden, )
( auch gemischt mit MAKE. Mehrfach-REMAKEs sind kein Problem, )
( mit jedem UNREMAKE geht es dann wieder eine Stufe zurück. )
( Auch doppelte Zuweisung des identischen Kodes, wie es z.B. )
( bei sich überlagernden Interrupts vorkommen kann, ist möglich. )
( Für jeden DOER wird ein eigener Stack benutzt. So braucht man keine )
( Rücksicht auf die Reihenfolge beim Biegen mehrerer DOER zu nehmen. )
>IN @ \ Quelltextposition merken
, \ 'cfa des DOER name holen
SWAP >IN ! \ Quelltextposition restaurieren
STATE @ DUP [STATE] ! \ Systemzustand
IF \ kompilierend
  [COMPILE] LITERAL \ 'cfa als Literal in Fadenkode einbauen
  COMPILE \ kompiliere: DOER>STACK
THEN \ (macht keinen Fadenkodeeintrag)
DOER>STACK \ bisheriges DOER-Verhalten sichern
[COMPILE] MAKE ; IMMEDIATE \ MAKE übernimmt alles weitere

: UNREMAKE ( ib: name ==> )( Dreht DOER-Verhalten eine REMAKE-Ebene zurück. )
( UNREMAKE schaltet wieder auf das ursprüngliche DOER-Verhalten. )
( Das Zusammenwirken kann in freiem Wechsel zwischen Compile-Mode )
( und Interpret-Mode erfolgen. )
STATE @ \ 'cfa des DOERS name besorgen
IF \ Systemzustand
  \ kompilierend
  [COMPILE] LITERAL \ 'cfa als Literal in Fadenkode einbauen
  COMPILE \ kompiliere: STACK>DOER
THEN \ (macht keinen Fadenkodeeintrag)
STACK>DOER \ DOER-Verhalten restaurieren
; IMMEDIATE

: [HISTORICAL] ( ib: name ==> )( Altes DOER-Verhalten in REMAKE-Kode einbauen )
( Wird REMAKE interpretierend aufgerufen, wird das alte DOER-Verhalten )
( vom DOER-Stack geholt. )
( Wird REMAKE im Compile Mode aufgerufen, wird das DOER-Verhalten zum )
( Kompilationszeitpunkt als altes DOER-Verhalten eingebaut )
STATE @ \ 'cfa des DOERS name
IF \ Systemzustand
  \ kompilierend
  [STATE] @ \ Systemzustand bei REMAKE
  IF \ war kompilierend
    >BODY @ \ jetziges DOER-Verhalten holen
  ELSE \ war interpretierend
    THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
    TOS \ DOER-Verhalten vom Stack holen
  THEN
    [COMPILE] LITERAL \ Es wird das Verhalten des DOERS kompiliert,
    COMPILE CALL \ welches zur Laufzeit von HISTORICAL gilt
  ELSE \ interpretierend
    THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
    TOS \ DOER-Verhalten vom Stack holen
    CALL \ und ausführen
  THEN
    ; IMMEDIATE
```

## Listing: REMAKE.CFW

```
\ REMAKE.CFW
\ =====
\ DOER verdrehen mittels REMAKE und UNREMAKE
\ -----
\ Bearbeiter: Thomas Höhenleitner
\ Stand: 14-Jul-95
\ © 1994 graphikon GmbH
\ =====
\ ORDER STRING ALSO OS ALSO FORTH DEFINITIONS
BASE @ DECIMAL
\ zu ladende Include-Files
@:STACKS WINCLUDE
HAVE remake.cfw 0=
.JF
CR .( ...lade REMAKE und UNREMAKE )
CREATE remake.cfw
\ -----
CR .( ...Laufzeitkode )
20 STACKS THREAD ( Stackklasse für DOER-Fäden, maximale Tiefe 20 )
: DOER>STACK ( ps: 'cfa==> )( DOER-Faden von 'cfa auf Stack )
  DUP >BODY @ \ DOER-Faden
  SWAP THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
  PUSH ; \ auf den Stack
: STACK>DOER ( ps: 'cfa ==> )( DOER-Faden restaurieren )
  DUP THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
  POP \ DOER-Faden vom Stack
  SWAP >BODY ! ; \ in den DOER zurück
```



```
: HISTORICAL ( ib: name ==> )( Altes DOER-Verhalten in REMAKE-Kode einbauen )
( Wird REMAKE interpretierend aufgerufen, wird das alte DOER-Verhalten )
( vom DOER-Stack geholt. )
( Wird REMAKE im Compile Mode aufgerufen, wird das DOER-Verhalten zur )
( Laufzeit des kompilierten REMAKE als altes DOER-Verhalten eingebaut )
STATE @ \ 'cfa des DOERS name
IF \ Systemzustand
  \ kompilierend
  [STATE] @ \ Systemzustand bei REMAKE
  IF \ war kompilierend
    [COMPILE] LITERAL \ 'cfa kompilieren
    COMPILE THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
    COMPILE TOS \ DOER-Verhalten zur Laufzeit holen
  ELSE \ war interpretierend
    THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
    TOS \ DOER-Verhalten vom Stack holen
    [COMPILE] LITERAL \ kompilieren
  THEN
    COMPILE CALL \ CALL führt Fadenkode ab Adresse auf Stack aus
  ELSE \ interpretierend
    THREAD \ 'cfa und THREAD sind Handle dhdl für Stack
    TOS \ DOER-Verhalten vom Stack holen
    CALL \ und ausführen
  THEN
    ; IMMEDIATE
\ -----
CR .( ...fertig )
CR
.THEN \ --- remake.cfw

2 UMAX BASE ! }ORDER
```

# FORTH optimieren

von Bernd Paysan

paysan@informatik.tu-muenchen.de; Stockmannstr. 14; D-814777 München;

Rechenzeit ist knapp. In mehreren Folgen werden Optimierungstechniken beschrieben: In native-code-FORTH, wie auch am typischen, gefädelten FORTH. In diesem ersten Teil soll auf traditionelle Optimierungstechniken und ihre Bedeutung für FORTH eingegangen werden. Der zweite Teil wird verschiedene Implementierungstechniken und ihre Leistungsfähigkeit beschreiben. Weitere Teile könnten z.B. Anton Ertls Vorschlag eines hochoptimierenden FORTH-Compiler nach dem Stand des Compilerbaus behandeln.

*Stichworte: Optimieren Forth-Compiler*

Eines von Murphy's Gesetzen könnte etwa so lauten: Die CPU ist immer zu langsam. Wenn WinWord mit einem Pentium90 gerade flüssig scrollt, lechzt der Benutzer schon nach dem P6. Als FORTH-Programmierer hat man es nicht so einfach. Schließlich verlangen die Kunden, daß in FORTH alles mit einem Z80 geht. Und da hilft nur eines: Optimieren.

Optimieren heißt hier nicht etwa, ein optimales Programm zu generieren, sondern nur näher an das Optimum heranzukommen. Dabei gibt es mindestens zwei Ziele, die ein optimales Programm kennzeichnen: Es soll möglichst schnell laufen und möglichst wenig Speicherplatz belegen. Meistens sind diese beiden Ziele gleichzeitig erreichbar, manche Optimierungen haben aber einen Space-Time-Tradeoff, wie Loop Unrolling oder manche komplexen Assemblerbefehle, die durch mehrere einfache (und deshalb schnellere) ersetzt werden können.

## Optimierung am Algorithmus

Nun hat man also sein Programm, das nach Meinung des Kunden einfach zu langsam ist. Zunächst gilt es zu klären, ob der verwendete Algorithmus die Bremse darstellt. Wer ein großes Array mit Bubblesort sortiert, sollte sich nicht wundern, wenn das ewig dauert. Mergesort oder Quicksort sind da sicher bessere Alternativen. Werden große Stringlisten durchsucht (wie etwa das FORTH-Wörterbuch), so sollte man sich eine geeignete Hash-Codierung einfallen lassen und statt einer Liste eine ange-

messene Menge gehashter Listen nehmen. Solche algorithmischen Optimierungen können die Geschwindigkeit eines Programms durchaus um den Faktor 1000 oder mehr beschleunigen. Sie sind ihr Geld also wirklich wert.

Gute Algorithmen findet man etwa in Nikolaus Wirths Buch "Algorithmen und Datenstrukturen". Das gibt es sowohl für Pascal als auch für Modula II - leider nicht für FORTH. Für FORTH existiert lediglich im Bereich der Numerik ein Buch "Scientific FORTH" von Julian V. Noble (vgl. VD 1/95, S.14). Ansonsten gilt Knuth's mehrbändiges "The Art of Computer Programming" als Standardwerk. Trotz des für die Computergeschichte hohen Alters (der erste Band entstand 1962!) behandelt es viele Standardprobleme. Bei neueren Problemen bleibt einem aber nur die Recherche in aktuellen Zeitschriften und neuen (spezialisierten) Büchern sowie die eigene Forschung. Hier umfassende Lösungen zu präsentieren, würde mehr Platz benötigen, als in dieser Zeitschrift jemals zur Verfügung stehen wird.

Ein nicht unwichtiger Nebenschau- platz der Optimierung betrifft die optimale Nutzung der verschiedenen schnellen Speicher. Der Cache-Speicher ist schneller als der Hauptspeicher, der wiederum schneller als die Platte. Bei riesigen Datenmengen sollte die innerste Bearbeitungsschleife möglichst nur ein etwa cachegroßes Teilstück abarbeiten. Die nächstäußeren Schleifen widmen sich den entsprechend langsameren Speicherarten. Durch solche Optimierung der Zugriffe sind Geschwindigkeitssteigerungen um einen Faktor 10-100 möglich

- auch bei PCs, nichtnurbei großen Workstations. Dies betrifft sowohl Datenbankzugriffe als auch die Multiplikation von großen Matrizen.

## Klassische Optimierungstechniken und FORTH

Eine Reihe von klassischen Optimierungstechniken sind in FORTH entweder gar nicht nötig, oder können mit minimalen Aufwand dem Programmierer überlassen werden. Es sind dies Optimierungen, die nicht besonders maschinennah sind, und sich auch als Sourcecodetransformationen beschreiben lassen können. Diese Optimierungen beeinträchtigen höchstens die Lesbarkeit des Programms und sollten natürlich möglichst nur an Performance-Engpässen angewendet werden. Als Abschnittsüberschriften habe ich die üblicherweise verwendete Nomenklatur gewählt, damit man sich auch als FORTHer im üblichen Jargon ausdrücken kann.

### Constant Arithmetic

Konstantenberechnung kann man in FORTH einfach den Interpreter erledigen lassen. Wozu hat man ihn denn? Dank der Worte [ (compile off) und ] (compile on) kann man ihn sogar mitten in einer Wortdefinition einschalten und Konstanten zwischendurch berechnen. Als Beispiel soll die Zahlenwandlung erhalten. Hier wird (in einer Definition) die Differenz zwischen dem Zeichen A und dem Zeichen 9 berechnet und um 1 vermindert:

```
ASCII A ASCII 9 - 1-
```

welches natürlich jedesmal ausgerechnet wird. Besser ist:

```
[ ASCII A ASCII 9 - 1- ] LITERAL
```

Das wird nur einmal ausgerechnet. Im ANSI-Standard muß man übrigens oben [CHAR] verwenden, unten aber CHAR - in diesem Fall hat State-Smart durchaus seine Vorteile.

### Constant Propagation

Es kommt durchaus vor, daß man mit Konstanten weiterrechnet und dabei

wieder konstante Operationen verwendet. Diese sollte man natürlich auch nicht zur Laufzeit ausführen, wenn eh' immer dasselbe herauskommt. Obiges Beispiel weitergeführt: Angenommen, wir wollen die Zahlenwandlung so weiterbasteln, daß nach Z a ausgegeben wird (wenn BASE > 36). Wir müssen also zu Zahlen > 35 noch etwas dazuzählen. Wir haben beim ersten Mal 'ASCII A ASCII 9 - 1-' ausgegerechnet. Nun kommt noch 'ASCII a ASCII Z - 1-' dazu. Das könnte etwa so gehen:

```
[ ASCII A ASCII 9 - 1- DUP ] LITERAL
(...anderer Code..)
[ ASCII a ASCII Z - 1- + ] LITERAL
```

Leider sind beide Teile hier vielleicht in verschiedenen IF-ELSE-Zweigen. Dann spielen uns die meisten FORTH-Implementierungen einen Streich: Die Kontrollstrukturen verwenden ebenfalls den Stack. Deshalb ist diese Optimierung leider nicht immer so einfach zu benutzen. Immerhin erreicht man mit Cut&Paste-Programmierung in diesem Fall etwa dasselbe.

## Common Subexpression Elimination

Dieser klassische Fall, daß mehrere Teilausdrücke gleich sind, entsteht beim FORTHeln kaum. Er ist typisch für Programmiersprachen mit benannten Variablen. Betrachten wir den Ausdruck:

$$(a-b)^2$$

In C formuliere ich ihn als

$$(a-b)*(a-b)$$

und hoffe, daß der Compiler bemerkt, daß er  $a-b$  nur einmal ausrechnen muß.

In FORTH schreibe ich:

$$a b - DUP *$$

und die "Subexpression" ist verdoppelt.

Bei komplexeren Ausdrücken muß man sich natürlich schon mal überlegen, ob da nichts doppelt ausgerechnet wird. Ist die Berechnung noch dazu faktorisiert, können die doppelten Berechnungen in verschiedenen Wörtern vorkommen. Tip: Faktorisieren Sie doppelte Berechnungen aus und versuchen Sie anschließend, den Aufruf durch einen übergebenen Parameter auf dem Stack zu ersetzen.

## Copy Propagation und Dead Store Elimination

Ebenfalls eher selten kommt Copy Propagation vor. Kopiert wird in FORTH immer explizit mit DUP, OVER und Verwandten. Diese Kopiererei kann man leider kaum vermeiden. Ausnahmen gibt es natürlich und zwar Copy Propagation in Zusammenhang mit Dead Store Elimination. "Dead Stores" sind in FORTH am DROP, NIP oder 2DROP zu erkennen. Ein Beispiel soll dies aufzeigen:

```
: draw-box ( left toprightbottom -- )
  2over (l t) move-to
  over 2 pick (r t) draw-to
  2dup (r b) draw-to
  3 pick over (l b) draw-to
  2over (l t) draw-to
  2drop 2drop ;
```

Hier fällt insbesondere auf, daß die Parameter für jeden Teilschritt nur kopiert werden, am Ende aber explizit gedroppt werden müssen. Das scheint nicht gerade ideal zu sein. Eine bessere Lösung konsumiert alle Eingangswerte, und verzichtet hinterher auf ein Aufräumen des Stacks:

```
: draw-box ( left top right bottom -- )
  2over (l t) move-to
  >r 2dup swap (r t) draw-to
  (l t) (R: b)
  r@ (r b) draw-to
  (l t) (R: b)
  over > (l b) draw-to (l t)
  (l t) draw-to ;
```

Statt neun Stack-Wörtern haben wir hier nur sieben, insbesondere sind die zwei Picks verschwunden. Alle Eingangsparameter leben tatsächlich nur solange, wie sie gebraucht werden. Als Faustregel für diese Optimierung kann nur gelten: Wenn ein Wert kopiert wird, wird er wirklich später noch gebraucht oder einfach gedroppt? Wenn letzteres der Fall ist, sollte man den Wert nicht kopieren, sondern ihn konsumieren lassen.

## Schleifenoptimierungen

Erfahrungsgemäß wird die meiste Zeit in Schleifen verbracht. Hier lohnt es sich also besonders, Optimierungen anzusetzen. Insbesondere gibt es einige

zusätzliche Optimierungsmöglichkeiten, die nur bei Schleifen Sinn machen. Am meisten lohnt eine Optimierung der innersten Schleife.

## Invariant Code Motion

Viele Schleifen enthalten invarianten Code, also Code, der bei jedem Schleifendurchlauf einen konstanten Wert neu berechnet. Diesen Code sollte man auf alle Fälle außerhalb berechnen. Unsere Beispielschleife soll die Summe zweier Werte in ein Array schreiben:

```
( a b n )
0 DO
  2dup + array ! cells + !
LOOP 2drop ( 1.79ms )
```

Diese angegebene Zeit bezieht sich auf bigFORTH, ein 4096-elementiges Array und einen 486/50MHz.

Das  $2dup +$  zieht man aus der Schleife heraus:

```
( a b n ) -rot + swap
0 DO
  dup array ! cells + !
LOOP drop ( 1.51ms )
```

Daß hier im Endeffekt mehr Code steht als vorher, schadet nicht, der Code innerhalb der Schleife ist deutlich weniger geworden.

## Strength Reduction

Teilweise ist dieser konstante Code aber nicht so einfach zu entdecken. So sieht die Berechnung ARRAY I CELLS + irgendwie nach so etwas Konstantem aus. Leider ist sie nicht konstant, sie ist ja von I abhängig. Allerdings sind die Ergebnisse offenbar jeweils nur um 1 CELLS unterschiedlich. Also berechnet man einfach vor der Schleife den Startwert und addiert jedesmal 1 CELLS dazu:

```
( a b n ) -rot + array rot
0 DO
  2dup ! cell+
LOOP 2drop ( 1.18ms )
```

Strength reduction bedeutet im eigentlichen Sinn des Wortes, daß komplexere (stärkere) Befehle durch einfachere (schwächere) Befehle ersetzt werden - die Multiplikation oder der Shift in CELLS durch eine billige Addition.

## Induction Variable Elimination

Jetzt ist offensichtlich die Zählvariable I verschwunden. Die Schleife wird also nur noch zum Zählen der Anzahl Wiederholungen benötigt. Das scheint eine Ressourcenverschwendung zu sein. Versuchen wir also, die einzig wirklich hochgezählte Variable, die Speicherposition, als neue Zählvariable zu verwenden:

```
(a b n) -rot + array rot cells bounds
DO
  dup ! [ 1 cells ] Literal
+LOOP drop (      1,32ms)
```

Ob das jetzt wirklich 'was gebracht hat, hängt natürlich vom verwendeten FORTH-System ab. Ein gefädertes FORTH, bei dem NEXT einen wesentlichen Teil der Ausführungszeit benötigt, wird eher langsamer geworden sein. Schließlich haben wir jetzt 4 statt 3 Primitives pro Wort. Selbst bei einem Native-Code-FORTH wie bigFORTH muß das (wie im Beispiel) nicht unbedingt etwas gebracht haben, da die Berechnung von I in diesem Fall etwas

aufwendiger ist, als das Weiterschalten von Hand (auch +LOOP ist zeitaufwendiger als LOOP).

## Loop Unrolling

Wenn alles nichts hilft: Bei so kurzen Schleifen wie oben liegt der Verdacht nahe, daß ein nicht unwesentlicher Teil der Zeit allein in LOOP verbraucht wird, also reiner Schleifenoverhead ist. Dagegen kann man folgendes unternehmen: Man kopiert den Rumpf der Schleife einfach mehrmals hintereinander. Dabei muß man die Änderungen des Schleifenindizes natürlich berücksichtigen; da wir die indexfreie (und bisher schnellste) Schleife als Ausgangspunkt nehmen, brauchen wir das jetzt nicht. Allerdings müssen wir den Rest der Durchläufe extra behandeln:

```
(a b n) >r + array r@ 3 and
0 ?DO 2dup! cell+ LOOP
(a+b addr) r> 2 rshift
0 ?DO
  2dup! cell+ 2dup! cell+
  2dup! cell+ 2dup! cell+
LOOP 2drop (1.16ms)
```

Wie man sieht, hat das wenig gebracht. Unsere Annahme war also falsch. Der Schleifenoverhead war minimal; anscheinend ist die größte Bremse hier der vergleichsweise langsame Speicher. Auch sind die Optimierungstechniken von bigFORTH hier nicht so ausgefeilt.

## Resumé

Wir haben die klassischen Optimierungstechniken abgehandelt, die ein C-Compiler beherrschen muß. Ein FORTH-Compiler hat es einfacher und überläßt sie dem Programmierer. Dadurch leidet leicht die Übersichtlichkeit der Programme. Nichttriviale Optimierungsversuche führen aber auch in C zu unübersichtlichen Programmen.

Die nächste Folge wird sich mit Fädeloptimierung befassen und soll vor allem diejenigen ansprechen, die unbedingt ein 32-Bit-FORTH auf einem Intel implementieren wollen - und natürlich all jene, die derartige Diskussionen verfolgen und mehr verstehen möchten.

□

## ICR Intermediate Code for Robots

CNC-Maschinen und Roboter sind seit Jahrzehnten im Einsatz. Trotzdem hat sich bisher für Steuersoftware kein herstellerübergreifender Standard entwickelt. Sehr zum Leidwesen der Anwender, denn dieser Zustand führt zu Mehrkosten bei Beschaffung und Unterhalt der teureren Anlagen.

Das Steuerprogramm für einen Roboter wird entweder online mit einer teach-in-Box oder offline in einer Programmiersprache erstellt. Interaktives teach-in mit Joystick am Roboter funktioniert in der Praxis meist recht gut, weil man innerhalb der Grenzen des vom Hersteller angebotenen Systems bleibt. Teach-in erlaubt aber keine allzu komplexen Bewegungen und außerdem steht während der Einstellung die Fertigung.

Erstellt man das Programm offline am Computer anhand einer Simulation ist meist erforderlich Hardware und Software verschiedener Anbieter zu integrieren. Folgerichtig ergeben sich bisher Probleme bei der Übertragung auf den realen Roboter.

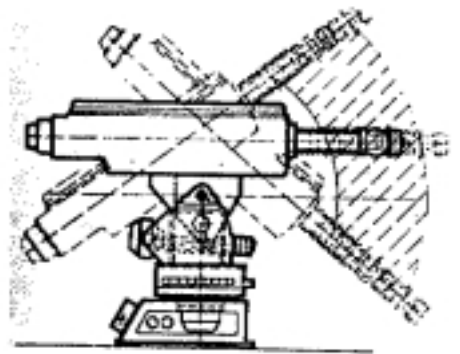
Dem abzuhelfen laufen in Deutschland seit 1980 Bemühungen um einen Standard für eine geeignete Programmiersprache und ihr Übergabeformat. Federführend sind dabei Forschungseinrichtungen und Universtitäten gewesen, insbesondere Aachen und Karlsruhe. Für die Programmiersprache ist IRL herausgekommen, für die Schnittstelle IRDATA. Beide sind jetzt nationale DIN-Norm, aber die Akzeptanz der Industrie war bisher eher verhalten. Sie sollen nun in internationale Normen überführt werden. Daß sich hierbei internationaler Widerstand regt, ist nur gut so, denn beide sind von Pascal und C verseucht. Von besonderem Interesse für FORTH ist dabei die Entwicklung bei ICR (Intermediate Code for Robots), dem ISO-Nachfolger von IRDATA. Er war bis 1992 zum ISO-Draft DP10562 gediehen. Dabei mußte ein Kompromiß aus IRDATA, französischen und japanischen Vorschlägen gefunden werden. Wobei etwa 75% des Drafts auf IRDATA beruhen, weil hierzulande schon die meiste Vorarbeit geleistet wurde. ICR ist maschinenunabhängiger Code, der die Schnittstelle zwischen dem offline-Programmiersystem und der Robotersteuerung bildet. Das Programmiersystem erzeugt dabei

ein Programm für eine virtuelle CPU. Die Robotersteuerung simuliert diese CPU.

Mit welcher Programmiersprache man am besten arbeitet, hängt sich davon ab, wie die Architektur der virtuellen CPU aussieht.

IRDATA will eine Drei-Adreßmaschine. Das ergibt sich daraus, daß IRDATA zu Zeiten konzipiert wurde, als man an CISC-Prozessoren und Pascal glaubte. Wobei sich echte Drei- und Vier-Adreßmaschinen auch bei Minicomputern nicht durchgesetzt haben, weil der Hardwareaufwand in keinem Verhältnis zur Rechenleistung steht. Damit liegen hier also die Anforderungen an die Robotersteuerung, die derartigen Code in Echtzeit verarbeiten soll, relativ hoch.

Der französische und der japanische Vorschlag sahen als anderes Extrem einen Stackprozessor vor, der offensichtlich einfach auf allen low-end-CPU's zu implementieren ist. Denn nicht alle Zielsysteme sind ausgewachsene



Roboter. Viele sind primitive Automaten und deshalb bisher mit wenig Rechenleistung ausgestattet. Wenn man wirklich Hochleistung braucht, kann simpler Code auch von modernen RISCs effizient verarbeitet werden.

Der für ICR vorgesehene Kompromiß sieht einen Stackprozessor mit zusätzlichen Adressierungsarten vor. Damit kann FORTH eine geeignete Basis sein um darauf ICR zu implementieren. Allerdings sind erhebliche Erweiterungen an Arithmetik und Multitasking nötig.

Literatur: Elektronik 20/92 S.116  
Elektronik 21/92 S.72

Rafael Deliano

## Brief aus der Diaspora

### PRIVATER BRIEF ENTHÜLT DIE UNTERGRUND- MACHENSCHAFTEN EINES GENIUS IN AMERIKA

Frank Rothkamm, Dadua@aol.com

**Im Moment schreibe ich den Soundtrack für eine Computeranimation die von der New York Film & Animation Co. hergestellt wird. Die Animation zeigt die neusten Möglichkeiten einer Software die auf Silicon Graphics' Chyron Rechner läuft. Das Stück wird auf der Siggraph Conference in Los Angeles uraufgeführt.**

**Hab vorgestern Nacht music & sounddesign für den Computeranimationfilm (SIGGRAPH) abgeschlossen, und bereite mich auf eine weitere Masteringsession heute im KINGSOUND studio vor. ("Mystery of the Leaping Fish" CD (flux volume two)).**

---

*Stichworte: Musik Formula CAMP fLux*

Ich habe in Moers Zeitungen ausgetragen, Mittel- und Oberstufe mit Abitur abgeschlossen und am Theater gelernt und gewirkt. Diese Zeit stellt die längste dar, welche ich an einem Ort zugebracht habe. Von dort bin ich zu meinen Schaffenswanderungen aufgebrochen, zunächst nach Köln, dann Kanada wo ich 4xForth kennengelernt habe, und nach Berlin, wo ich mit Dr. Rupert Nieberle die Computer Aided Processing Group (CAMP) an der TU gegründet habe. Wir haben dort u.a. mit Forth geforscht. Es wurden entwickelt: ein LISP Forth, Object Oriented Forth und verschiedene Algorithmen, um Musik mit Forth zu machen. Ich hab mit Markus den ersten "TANZMUSIKALGORITHMUS" hergestellt im H3001. Rupert ist jetzt sehr geschäftig mit seiner DSP firma... Wir hatten eine starke Verbindung nach Berkeley, Kalifornien. Ich habe auf der International Computer Music Conference ein Stück theatralisch aufgeführt.

#### **Die reinste Odissee.**

Besagtes Buch von Homer hat mich stark geprägt im zarten Jugendalter, die Realitaet sieht aber mehr wie James Joyce's Aufarbeitung des Mythos aus. Als mein Interesse an Academia im besagten Berlin beendet war, bin ich Kanadier geworden und habe in Vancouver für "Science World of British

Columbia" eine Interactive Software Installation hergestellt: (die jetzt in der permanenten Kollektion ist) PAINT MUSIC. Da konntest du in einen schallgesichterten Raum eintreten, und Bilder mit dem Trackball auf einen Bildschirm malen, während der Recher dein Bild liest und in Musik umwandelt. (alles in Forthmacs)

Wenn dich das nicht an XENAKIS's Upic system erinnert! (Danach habe ich eine Art von FM synthese implementiert für Videospiele in meinen einzigen "JOB, den ich je gemacht habe.)

Weiter gehts nach San Francisco. Dort wurde Forth eingesetzt für die Steuerung von Musik bei "Ritualen" in Clubs. Wie das ausgesehen hat, kanns man sich im PLAYBOY Feb91 angucken. Es wurde klar, daß Forth und FORMULA musikalisch geeignet sind, TRANCE (also undergrund tanzmusik) herzustellen, und "oh lord god" für die \_Death Rave 2000\_ compilation wurde auch ein milder "hit" (#19 USA-CMJ-chart). Nicht nur ist die Musik ganz und gar in Forth gemacht, sondern auch die Form und Details des Stückes wurden stochastisch bestimmt. (bedingte und begrenzte Wahrscheinlichkeiten u.s.w.). Jetzt verliert sich der Forth Faden ein wenig, weil ich ein Kommerzielles "production house" hatte (spots für die USA kampagne fuer "LEVIS 501 JEANS"). Dabei habe ich viel Geld gemacht und verloren und dann gings nach:

#### **New York**

Hier hab ich ein Label für meine Aktivitäten ins Leben gerufen. (Sonic, Symbolic, Scriptural) und arbeite an einem 4.56 Stunden CD Zyklus, wo viele Stücke in Forth geschrieben wurden. Das erste Volumen "PLANET GENIUS" ist veröffentlicht. Rupert hats. Alles in Forth auf dem!!! (und bin mit Laurie Anderson, u.a. auf P.Scherer's neuer Scheibe auf Capitol). Arbeite auch mit Forth/Scanner/Graphic/Algorithmen im Moment, und nach 5 Jahren im Untergrund mit verschiedenen Identitäten bin ich jetzt 'Amerikaner'. Dieses möge als kleiner Einblick in USA-Ereignisse genügen.

Habe neulich mit meinem Produzenten gesprochen: Heutzutage muß man fast jede Plattform haben. PC,MAC,-ATARI,AMIGA, und eine NeXt in der Garage sieht auch nicht schlecht. Hab hier einen portable IBM-PC von 1985(?) rumstehen, ein Freund in Idaho versucht gerade den Soundchip zu verunsichern, gib's da ein altes Forth für? (5 inch floppy, single-sided, 512k). Gibt es noch das erste Forth, das auf der Sternewart entwickelt wurde? Es wäre mal interessant, mit diesem Material der Gründungsstunde rumzuspielen.

Hättest Du Lust, uns da etwas unter die Arme zu greifen? Sicher, keep 'em coming. Bevor du jetzt einschläfst, schicke in den Brief mal ab.



## ... nix mit Forth zu tun?

Ein Track der PLANET GENIUS CD welcher komplett in Forth geschrieben wurde: Stochastisch erzeugte Ganztonleitern werden auf rhythmische Gitter gelegt und nudeln dann zyklisch immer weiterschreitend vor sich hin. Eine Opernsängerin singt immer mal etwas kurzes Schlaues, bevor der Zyklus fortschreitet. "Bliss in Circles opts for

the strange - Residential rhythms, a gentle pulsing and a distant opera singer that all make you feel like you're in a slow spiral" Alternative Press USA. □

### THE VOICES OF THE PRESS:

- *AMBIENT INTO TRANCE: dramatically different timbres, splendid morass of swirling rhythms..and the opts for the strange..that elevate it far above your average fare.* (Alternative Press).
- *lovely unique package...as much art object as recording* (Option)
- *REALLY weird,...dramatic, surprising, novel,...perhaps signifies the birth of a new genre of music.* (UK-Dance, IDM)
- *experimentation and artistry thrown into your dance mix...the work of true mastermind* (Independent Music Review)
- *It's amazing because while listening to it, everthing seems to be possible* (Spin Off, GER)

### THE FAQ:

- Q: Where do they live ?  
A: New York
- Q: Where do they come from ?  
A: The german theater
- Q: Who has released their music in the past?  
A: Capitol, Hardkiss, Fabulous, 21st Circuitry
- Q: Who has written about them ?  
A: Playboy, Streetsound, Option, Internet, Hypno, URB, Alternative Press, IMR
- Q: What other things did they do ?  
A: Classical composition; software engineering; body manipulation, S&M & ritual performance; raves & commercials.

Subj: Charts From CFRC-FM, Toronto  
Date: July 31st to August 6th, 1995  
From: SHANKSP@QUCDN.QUEENSU.CA

## TOP 20 DANCE / SOUL / AMBIENT

TC	LC	Artist/Record/Label
1	6	* ADSR/Universality EP/12/Cargo
2	13	Hardfloor/Respect/CD/Harthouse
3	--	Banco De Gaia/Last Train To Lhasa/CD/Planet Dog
4	2	VA/Positive Energy/CD/Moonshine
5	11	Planet Genius/Flux Vol. One/CD/Flux
6	4	VA/The Theory Of Evolution/CD/Warp
7	5	Richard H. Kirk/The Number Of Magic/CD/TVT
8	1	* Deepspace/The Alexia Beach Explorer/CD/Hypnotic
9	3	Underworld/Born Slippy/CX/Wax Trax!
10	7	Electric Skychurch/Knownoneness/CD/Moonshine
11	--	VA/Jazzmatazz Vol. II: The New Reality/CD/Chrysalis
12	--	Rockers Hi-Fi/Rockers To Rockers/CD/Island
13	--	VA/Virtual Energy/CD/Beatbox
14	R	VA/Digital Sects 2/CD/Matrix
15	14	VA/Jungle Warfare/CD/Moonshine
16	8	* The Fire This Time/Dancing On John Wayne's Head/CD/Extreme
17	16	VA/Throne Of Drones/CD/Sombient
18	15	Pentatonik/Anthology/CD/Deviant
19	10	Sven Vath/The Harlequin, The Robot & The Ballet Dancer/CD/Eye Q
20	9	F.S.O.L./ISDN/CD/Virgin

-----fLUX press release-----

## Forth Sounds on CD:

# PLANET GENIUS (fLUX volume one)

NEW YORK. Frank Rothkamm of the PLANET GENIUS project completed the music & sounddesign for a computeranimated short produced by the New York Film & Animation Co.. It was shown on a Silicon Graphics' CHYRON machine at the SIGGRAPH Conference in Los Angeles, CA. [August '95 /red.]. The dramatic soundtrack mixes analog & 40's style effects, orchestral episodes & operatic Intelligent Dance Music.

```

/}
/ G }
/ ___}
~~~~~fLUX_NYC~~~~~
~~~~~Box 2141~~~~~
~~~~~New York~~~~~
~~ NY 10163~~

```

### Bestellungen der CD bitte an:

F.Rothkamm  
P.O.Box 2141  
New York, NY 10163  
USA  
International Money Order  
\$ 15 + 3(S&H) = \$18

Ein separates Buch in Englisch ist ebenfalls erhältlich (Auflage: 150). Alle Stücke sind mit Algorithmen in Forth Musical Language hergestellt. Deborah Borchers ist Gastsängerin auf dem letzten Stück. PLANET GENIUS wurde in San Francisco & New York produziert und in Texas hergestellt. Ein besonderes Silberfolienspiegelmaterial und eine im Zentrum der Buchkunst in New York im Handdruck erstellte Rückseite sind in die CD eingelegt.

*"lovely unique package  
...as much art object as recording"*

(Option magazine, USA)

**fLUX/NYC is a label for sonic, symbolic & scriptural propaganda funded by the Ancient Creators on PLANET GENIUS.**

*This is the Electronic Mail transcription of the original limited edition book, letterpress cover, with hemp paper end sheets, 8 \* 8 inches. You may obtain the original for \$ 251.-(US) at the address below.*

/ }  
/ G }  
/ \_\_\_\_ }

frank genius presents:  
PLANET GENIUS ( fLUX volume one)

## I

*"Genius is the ability to reduce the complicated to the simple" C.W. Ceran*

## II

This booklet and Compact Disc are fLUX volume one, the first in an edition of four volumes:

PLANET GENIUS  
MYSTERY OF THE LEAPING FISH  
METAPHYSIQUE  
STAR STRUCK

## III

I get out of my Chevy and walk into the California desert  
After a while I lose sight of the car.  
I hear a voice and stop.  
"Go nowhere", it says.  
I want to give up but keep walking until I discover a well.  
I drink.  
This water is the music on PLANET GENIUS.

## IV

In San Francisco I dance at a rave.  
All of a sudden I find myself flying above the roofs of the city.  
"What just happened?" I wonder.  
"What would trigger such leaps of consciousness?"  
I come across the tape recordings of the Tuareg people in Mali, Africa.  
In order to exorcize the spirit from a tribe member,  
the Tuaregs chant and clap their hands in a seemingly simple rhythm without melody or harmony.  
They repeat these sounds for long periods of time.  
I am hypnotized.

## V

I generate all the music in Forth Musical Language (FORMULA).  
This esoteric computer language

originated at Kitt Observatory in the Arizona desert  
to synchronize the movement of a giant telescope  
to the orbits of celestial bodies.  
Rather than looking to the stars above me,  
I'm concerned with the dancing bodies that surround me.

## VI

A friend has a psychedelic dream.  
He wakes up and tells me about it:  
In the dream we sold an object we'd made together for \$251.00. This leads to an inspiration:  
251 will determine a variable in the computer program.  
I use it in the same manner I'd usually use a random number:  
to create an unpredictable music destination.  
The music I'm making has no definite length, so "Tribe 251" simply becomes 251 bars long.  
From then on I pay closer attention to numbers in dreams.

## VII

All night I channel surf through New York's cable TV stations. A spacecraft flies over America.  
I am sitting without motion,  
staring at a screen of constantly changing colors and sounds. Sometimes I write a number on a piece of paper  
and send it to the place where the colors and sounds are born. In the morning I wake up with a new concept: PLANET GENIUS.

## VIII

The design of the Compact Disc contains the future and the past simultaneously.  
Like a planet it has two sides:  
synthetic and organic, day and night, bright and dark.  
On the front side there is no cover of logo.  
The CD lies on silver mirrors within a clear case.  
At first glance there are no letters.  
Hold the CD to the light.  
Now look closely at it.  
Turn the case around the see the back has been handprinted on a 19th century press.

## IX

I was born in Gutersloh, Germany.  
I threw an axe into a fish tank in a German theater.  
I smashed bottles with a slegdehammer

and played violin to burning scores in Cologne's galleries. I cut my arm and inserted fish into the wound  
at Swiss and Canadian Festivals.  
I did not move while sitting for hours on a pedestal  
at the International Computer Music Conference.  
I wrote software to transform pictures into music  
for Science World' permanent installation and for video games. In San Francisco I organized raves  
and conducted national campaigns for Levi Strauss & Co.  
My S&M performances with Monique are documented in PLAYBOY. My music has been released by Hardkiss, Capitol, Fabulous and 21st Circuitry.  
I now reside in New York City.

## X

I am eternally grateful to:  
BERLIN:  
Dr. Rupert Nieberle of Computer Aided Music Processing e.V. BERKLEY,  
CALIFORNIA:  
Dave Anderson/Ron Kuivila for FORth MUsical LAnguage.  
SAN FRANCISCO:  
Deborah Borchers for vocals.  
Matthew Rogers (MRR) for disc design.  
LOS ANGELES:  
Joi and Katie of Way-to-Go for product coordination.  
PLANO, TEXAS:  
Disctronics for manufacturing.  
NEW YORK CITY:  
Nina Schneider at the Center for Book Arts for letterpress design and printing.  
Brian Hannon for text consultation.  
Larry Bercow and Bea Da Costa for photography.  
Peter Scherer at King Sound for mastering.  
PLANET GENIUS is made possible by the generous support from the Rothkamm Fund and N. Schneider.

## XI

frank genius is represented by  
fLUX  
Box 2141  
New York City 10163  
USA  
This is book No...in a limited edition of 150.



# The Freiburg-Zurich-Nashville-Story

## Wie man mit Forth und Brodie lernt, C zu programmieren ... von Philip Zembrod

Philip Zembrod; In den Muehlmatten 5; 79288 Gottenheim; zem@iwmm.fhg.de

Der ACM, das ist wohl DIE internationale nichtkommerzielle Informatikervereinigung (übrigens auch mit einer Special Interest Group für Forth). Seit 19 Jahren veranstaltet der ACM jährlich einen internationalen Programmierwettbewerb für Studenten. Dieses Jahr ging der erste Platz zum ersten Mal an eine europäische Hochschule, und zwar an die Uni Freiburg. Einer der Teammitglieder, Philip Zembrod, ist Mitglied der Forth-Gesellschaft und hat einen guten Teil seiner Programmierkenntnisse auf dem C64 mit Forth erworben. Daß im Wettbewerb als Sprachen nur Pascal, C und C++ zugelassen waren, war kein Problem. Man ist ja schließlich mehrsprachig aufgewachsen ...

Näheres zum Wettbewerb, Teilnahmebedingungen etc. finden sich auf dem WWW-Server des ACM unter <http://www.acm.org>

*Stichworte: ACM C Wettbewerb*

*About July 15th, in the year of the lord 1994:*

*Xbiff: Beep !*

*Philip: ?????*

*Xbiff: Beep ! I mean it. You've mail.*

*Philip: Ah. Hey, elm, what's up ?*

*Elm: Hm, let's see. There's someone looking for teammates for some kind of programming contest.*

*Philip: Oh. That's bound to be fun. Tell him, I'm interested.*

*Elm: Ok ...*

Let us now leave this rather metaphorical scene and briefly recall what happened next. Well, after the first meeting and then, in spite of Philip's first enthusiasm, a lot of doubt, like shall I do this when I should be studying for my finals, or I can't put so much time in training for this, I have my finals, after all that ("Yes, I'm doing it" "No, I'm not" "Yes, I am" "No, I'm not"), the team as we know it was finally formed. Which turned out to be one of Philip's best decisions in 1994, in the end. Good. Next, Matthias says, we

should get some practice. Programming quick is a matter of practice. So they sit, mostly on Sundays, and do old contests' problems. 5 hours. One Machine. Getting "Runtime error" and "Wrong answer" from their simulated jury. Swearing. "Who had this idea with this bloody contest ?" "I'm fed up, I'm going home". Debugging. "Print that out, I need the machine" "Did I ever mention that I hate C ?" "Who wants to scan this pathologic input format anyway" "How could they get this one to crash with legal data ?" "Pushing the specifications to the limit ?" "Great idea ! And where, please !?" "Okay, I'm submitting E" (wait 5 minutes) "Accepted !" "Aaah ! Now, who's great ?" Speculating. "If we do just a little bit better, we could win" "Aw, come on, get real. There's teams coming from all over Europe" and so on and so on.

### [Timewarp]

We're in Zurich, it's high noon. We see three weird-looking strangers walking up the main street. "Remember, boys, this is not about a quick draw, but about a good aim." Now what the hell's

that supposed to mean ? They enter the saloon, approach a terminal (help!! I'm in the wrong movie!) and calmly start to type lots of cryptic comments into it.

Warp back, adjust context] Please excuse my fall back into a bad Western. Zurich, evening after the contest. Motto: Free cheese for all. The whole bunch of contestants sitting around long tables, eating swiss chees fondue (great compliment to the organizers). I was pretty nervous; we allready had the idea that it could be Nashville, TN for us. While waiting for dinner I made friends with Jean-Paul, Modena's coach, spying out the prettiest among the (few, alas) girls, sang Crash Test Dummies with Hartmut from Erlangen, and startet to drink with determination as soon as the wine arrived. Then Big Bill climbs the stage along with the jury and staff and happily tells us "We know who won and You don't, and in a few minutes we'll tell You". Big Bill, that's Bill Poucher, International Contest Director. He calls up all the teams, and when the other team at our table get their envelope and open it, they take out a ranking list, and I glance over and see 'Albert-Ludwig' at the top of it, and I just couldn't believe it. The rest of the evening we're having a great time, exchanging email addresses, learn that we'll see Frank and Hans, the Regional Contest Director from Zurich and his secretary (me boss, You not) again in Nashville, and Martin from Ulm tells me to do well in Nashville and not to cut my hair, and on our way back to our quarters we make plans to find us a sponsor for the final and get a lot of money (nothing came out of those plans :)

### [Day dawns. Time passes. Location: back in Freiburg]

Philip receives his grade as a physicist. Our team books a flight to USA and a Greyhound ticket (ugh!). They take the train at 2.40 am and decide that the act of travelling itself is very tiresome and a rather bad idea altogether. Someone should at last implement a Star Trek type transporter. They see a lot of planes and spacecraft in the Air and Space Museum in Washington, DC.

They change a set of German coins for a three dollar bill with the hostel warden in the Great Smoky Mountains. They see the Jack Daniels Destillery and don't get a drop of whiskey because it's Sunday. They approach the Opryland Hotel, the location of the contest, and check in.

### [Zoom in]

"Jesus, what kind of a place is this?" "Looks rather like a jungle inside. Good thing there are paths through it, so we don't have to cut our way through." "This place is so big, a bike rental would be a good business here." We're absolutely amazed. We're in some kind of big greenhouse with the walls being the hotel. "We should have invested \$30 a day more to have one of those rooms with a balcony above the jungle." We even saw real bananas on a palm tree near a water fall.

### [Up pops a little window for parallel viewing]

Location: one of the larger hotel suites. We see Frank and Hans as they generously declare themselves responsible for disposing of a styrofoam box filled with ice which is some kind of leftover from a party. To make things worse, the box even contains some cans and bottles of beer.

### [Little window closes; we're single-task again]

Rest of the day we're hotel-exploring, writing some mails, buying postcards, trying Gentleman Jack in the Hotel bar (not bad, only the stuff came out of some funny device like a mixture of a shower and a telephone which is supposed to automatically produce all kinds of drinks with whiskey in it by pressing a few keys), and then meeting Hans and Frank who surprize us with well-cooled beer. Now where the hell did they get that? Anyway, the next day passes. Nothing happens.

### [Wednesday, the day of the CONTEST !]

Lots of excited programmers assembled in front of the governor's ballroom. Rules being announced. Cameras. Big Bill's pre-contest show, giving us mysterious hints how often exactly to touch the trophy he's showing around. Touch it right, and You win, touch it too often, and You loose, it seems. We don't understand, but are having fun anyway. Spectators enter the room. Contestants enter the room. "Hey, you're not supposed to be here yet". Contestants leave the room. Contestants enter the room again. We are VERY excited. Countdown homing down on zero. Confusion. "Hey, the contest has begun!" ... !!! Hastily open envelope. Start reading problems. Get sober and nervous. We don't know which problems to tackle. I look for my traditional parsing problem and don't find it. Panic! Ah, the last one has to do at least a little with scanning. That'll have to do then. Set to work. Things start to look good. Early success triggers expectations.

I'm thinking about that clock problem. Hm. Can't calculate it directly. So we'll have to simulate the damn thing. Going to take a lot of runtime. Might get us into problems. In comes Matthias with his typical "That one's trivial. There's just 101 different inputs possible. We can pre-calculate them". I grin. I like the idea. Though I'm not so sure if the jury will like it. Anyway, we change our program to calculate all possible cases and write an array initialization. We start it. Yes, NT \*is\* an operating system, and multi-tasking, too. Half an hour or so later we have our array. We make a program out of it and test it. Wow, it's really FAST. Will the jury be suspicious? We submit it. We laugh. This is really fun. And then the jury accepts it. Great. Who wants a biscuit? (Back at home I tell this my friend who's a mathematician, and he tells me right away how to calculate that much faster. I slap my forehead and find that brain cells don't \*always\* work at maximum efficiency when under stress.)

30 seconds more to go ... we relax and lean back. No way to get the seventh problem to run anymore. 10 .. 9 .. 8 .. 7 .. 6 .. 5 .. 4 .. 3 .. 2 .. 1 .. it's

over. Pew! Get up, breathe deep. We're happy. We did well.

The next hours are painful. Again we know we might have won. Tension builds up and becomes hard to bear. We go to the great evening banquet where all the ACM awards are given. The contest is last on the agenda. At last again the Bill Poucher Show. He makes us laugh, and we are thankful for it. The Microsoft representative tells us they've raised the prices. Our hearts jump up and sink low ... more tension. Then the winning teams are called, from sixth to first. With every called team that's not us our hopes rise. Then Berkeley is called as second, and we know, WE MADE IT !!! We don't believe it. We are called. We climb the stage. We receive the trophy. Lots of photos, cheers, I don't know, I'm like drugged for the rest of the evening. (Btw, the time we were ahead of Berkeley turned out to be just 4 minutes! Pew!)

The rest of the evening is party, meeting Ben, Sam, Mike and Trent from Nebraska and Shar and Amber from Waterloo, moving on to a hotel bar later, getting drunk and celebrating. In the morning we decide to stay in the hotel for another day, the Microsoft people make a video interview with us, we send mails to the folks back home.

Next day we set out, heading home. Turned out difficult because of somehow our luggage had grown considerably. Back home, our success caused some excitement, but that's a different store, to be told another time. □

Immer wieder werden nahmhafte Mitglieder der Forthgesellschaft dabei beobachtet, wie sie zum Erstellen von Programmen Systeme verwenden, die nicht wie Forth aussehen und auf denen noch nicht einmal 'Forth' draufsteht. Einige dieser Mitglieder scheuen auch nicht davor zurück mit derart sakralen Techniken Wettbewerbe zu gewinnen - teilweise sogar unter den Augen der Öffentlichkeit! Das Forth-Magazin berichtet wie immer schonungslos und aus erster Hand.



# Forth International

von Fred Behringer

Die holländische "Forth-  
Gebrauchergruppe"



Eigenartig: Es könnte Deutsch sein, wenn man die zweite Lautverschiebung rückgängig macht (was mir als gebürtigem Berliner nicht schwerfällt), ist es aber nicht. Trotzdem: Bayerisch ist schwerer ("abgehen" für "fehlen", "geh her" für "komm her", "geh weiter" für "ach, nein" usw., von der unmöglichen Lautverschiebung gar nicht erst zu reden.)

Ich kann nur jedem empfehlen, in das Feigenblatt (Het Vijgeblad) reinzusehen. Mir gefällt es. Die Artikel sind mindestens so lesenswert wie unsere. Und sie kümmern sich ausdrücklich auch um die Anfänger.

Das Feigenblatt (Het Vijgeblad) wird von der Forth-Benutzergruppe (Forth-

Gebruikersgroep) innerhalb des Hobby-Computer-Clubs (HCC) herausgegeben und erscheint viermal pro Jahr. Das Jahres-Abonnement, Mitglied oder nicht Mitglied, kostet 40,- Gulden.

Die Zeitschrift umfaßt durchschnitt-

## Ab und zu werden Dinge aus der Vierten Dimension berichtet ...

lich 50 Seiten in DIN-A4-Format. Feststehende Rubriken sind zur Zeit "Von der Redaktion", "Gelesen in ...", "Mitteilungen des Vorstands" und "Das Beste aus dem USENET".

Die meisten Artikel sind natürlich holländisch geschrieben, einige aber auch englisch. Das "ganz schnelle"

Inhaltsverzeichnis (mit jeweils zwei Zeilen Anmerkungen) ist englisch gehalten.

Vorsitzender ist Willem Ouwerkerk, Boulevard Heuvelink 126, NL-6828 KW Arnhem. Redakteur des Feigenblatts ist Lennart Benschop, Sibeliuslaan 35A, NL-5654 CZ Eindhoven.

Ab und zu werden Dinge aus der Vierten Dimension berichtet, genauso wie aus der FORTHWRITE, dem Organ der FIGUK, dem englischen Forth-Verein.

*Fred Behringer, München*

Ich möchte in dieser Rubrik gern in unregelmäßiger Folge über Forth-Aktivitäten in anderen Ländern berichten. Ich bin fleißig am Sammeln, würde mich aber natürlich auch über Zusendung von Material von anderer Seite freuen.

**Fred Behringer**  
**Planegger Str. 24**  
**81241 München.**

Verständlich sind für mich die Sprachen Englisch, Holländisch, Italienisch und Französisch. Bei Russisch würden meine vier Jahre Schulkenntnisse nicht ausreichen. Aber vielleicht könnte ich dann Unterstützung von den sehr aktiven Rostockern bekommen (?)

## Gehaltvolles

zusammengestellt und übertragen von Fred Behringer

## Het Vijgeblad von der HCC Forth-Gebruikersgroep, Holland

**Nr. 45, Drittes Quartal 1994**

6 Over optimalisering in Forth (Teil 1)  
Willem Ouwerkerk

Beschreibung eines universellen "Peep-hole-Optimizers" (er hält durch ein Schlüsselloch nach dem zu optimierenden Code Ausschau, wie der Autor erklärend hinzufügt) für Forth. Er ist sowohl für Compiler verwendbar, die in Maschinencode geschrieben sind, als auch für solche in gefädelttem Code. Er wird im neuen 8051-

ANS-Forth verwendet. Der zweite Teil folgt in der nächsten Ausgabe.

9 Enkele dubbel nummer primitieven  
(ohne Autor)

Doppelpunkt-Definitionen für UD\* D\* D< D> D0> D0< D= DMAX DMIN. Meinung des Rezensenten: Interessant. Das alles (und noch viel mehr) habe ich in mein Transputer-Forth-System F-TP schon vor Jahren

eingebaut. Aber natürlich als Code-Definitionen. Ist das nicht falsch verstandene Portabilität: Doppelpunkt-Definitionen sind für DUMP und SAVE-SYSTEM ausgezeichnet, weil durchsichtig, aber für D> ? - Ich meine, wenn ich das zum ernsthaften (z.B. wissenschaftlichen) Arbeiten wirklich brauche. Was spricht denn dagegen, sowas schnell mal als Code-Definitionen für alle möglichen Maschinencodes bereitzustellen? Das kostet auch nicht viel mehr Platz in den Zeitschriften und ist für den, der's braucht, von großem Wert. Forthens Stärke kann doch nicht in der Portierbarkeit um jeden Preis liegen (?) Forthens Stärke liegt unter anderem in der bequemen Handhabung von Bausteinen, die überall schnellstens eingefügt werden können.

10 CHForth  
Coos Haak

Der Autor gibt einen kurzen Abriss der Entwicklungsgeschichte seines CHForth-Systems: "Es begann alles mit FIG-Forth auf



## Das kleinste Forth

Liebe Redaktion,

wie sieht das kleinste Forth aus, das in der Lage ist, "sich selbst hochzuziehen"? Kein Tiny-Forth á la Ting. Ich meine einen Pre-Kernel mit kleinstmöglicher Wortzahl, ein Bootstrapped-Forth, in das man schalten kann und das dann den Rest des Gesamtsystems hinzulädt und compiliert. (In einem solchen Fall würde man nicht mehr von Metacompilation reden können. Es wäre eine einfache Compilation, so wie ein Anwenderprogramm nicht metacompiliert, sondern einfach nur compiliert wird.) Natürlich hängt jedes Forth an der Nabelschnur eines Betriebssystems und man müßte erst einmal eine genaue Begriffsbestimmung vornehmen, bevor man das Problem präzise formulieren kann. In welcher Form soll der hinzuzuladende und zu compilierende Teil des Restsystems vorliegen? Als Datei oder soll er sich schon im RAM befinden? Von mir aus darf er sich ruhig schon im RAM befinden. Die Datei-Organisation ist nicht unbedingt die Aufgabe Forthens. Sicher ist, daß es mehrere Lösungen gibt. Um anzudeuten, in welcher Richtung ich denke: Das Minimalsystem enthält natürlich kein , (Komma), sondern kommt mit C, aus - und compiliert selbstverständlich , (Komma) als allererstes beim Hinzuladen des eigentlichen KERNELS.

*Fred Behringer, München*

## FORTH und AutoCAD

Grüß euch alle!

Ich verdiene meine Brötchen (mit Belag :-)) durch das Entwickeln von Software unter AutoCAD. Ich stoße in LISP auf Geschwindigkeitsprobleme und in C auf Geldprobleme (Entwicklung viel teurer als in LISP).

Deshalb möchte ich jetzt ein Forth unter ADS implementieren, um wieder einen Interpreter zu haben, der aber flotte Programme liefert. Völlig FORTH - untypisch möchte ich das lieber kaufen als erfinden!

Falls ich das doch erfinden muß, wird es wohl LMI- 386 - URFORTH werden.

Wenn jemand mitliest, der ein eigenes FORTH hat, das als Pharlapkompatible Library existiert oder als als C-Quelltext vorliegt, bitte melden (Lizenzvereinbarung)!

Kommentare? Meinungen?  
Wünsche? Bitte reichlich!

*Gruß, Herbert  
in de/comp/lang/forth  
hkoenig@hkoenig.forth-ev.de*

Hi, there,

VD 2/95 am 7. problemlos erhalten! Der Cover war diesmal etwas trübselig nüchtern. Irgend ein Bildchen würde der Sache ein Gesicht geben. Der Artikel über Kreisalgorithmus wäre von der Sorte, die FORTH und der VD ein ernsthaftes Image geben könnte. Auch der Artikel Plewe/Allinger war angenehm seriös in der Feststellung, daß FORTH auf PCs nicht immer optimal ist.

Die Postscript-Grafiken in meinen Artikeln sind offensichtlich danebengegangen. Sie wurden nur als Outlines dargestellt. Wohl besser, wenn man in Zukunft alles scannt.

Die Adressenseite ist immer noch nicht überarbeitet. Einige fehlen. Z.B. die Münchner Gruppe. Viele Karteileichen sind dagegen aufgeführt.

*Rafael Deliano, 16.7.95*

Hallo Claus,

Deine Version der VD kann sich wirklich sehen lassen. Man merkt, daß es Dir Spass macht! Das ist wichtig!

Zu den Bildern: Wie kommen die Verzerrungen zustande? Soll ich lieber wieder per PapierFAX schicken? Unter der Qualität des Bildes zu Meßtechnik sollte es nicht sein, wenn mein Name dran steht.... :->

Ansoensten: weiter so.....

*rolf.kretz@rkaachen.forth-ev.de  
(Rolf Kretzschmar)*

## Zu 'Brief aus der Provinz' in VD 2/95, S.18

... Da fällt mir übrigens eine Korrektur zur letzten VD ein. Darin stand, daß ich der einzige Prinz in Moers bin - oder so ähnlich. Das ist völlig falsch. In unserer Forthgruppe gibt sich mein Bruder Ulrich gelegentlich die Ehre (und uns auch). Mitglied in der FG ist er genau so lange wie ich. Und in Berlin war er auch dabei. Der Ulrich ist sogar mit auf dem Photo.

# Leserforum

## *Glückauf, Friederich Prinz [die red. entschuldigt sich untertänigst bei den vergessenen Prinzen]*

Lieber Claus Vogt,

jedesmal freue ich mich, wenn die VD in meinem Briefkasten gelandet ist und neugierig schaue ich dann hinein. Selber habe ich noch nie einen Beitrag zur Vereinzeitschrift geleistet. Die Ausrede mit der Zeit will ich nicht anführen - wir haben ja alle gleich viel Zeit; so etwa 24 Stunden jeden Tag.

Anbei übersende ich Dir zwei Files, die vielleicht das Interesse der Einen oder des Anderen erwecken könnten. [s. DOER in diesem Heft / red]

*th@marie.graphikon.com  
(Thomas Hoehenleitner), 7/95*

## **Zu 'Das war die Forth-Tagung' in VD 2/95, S.15**

Hallo allerseits,

ich habe gerade die neue VD bekommen und durchstöbert. Ist gut geworden, finde ich (dickes Lob an Claus!!!).

Jörgs Artikel über die Forthtagung bietet auch eine gute Zusammenfassung der Geschehnisse. Meinen Vortrag über die automatische Generierung von On-Line-Glossaren betreffend, finde ich aber nicht unbedingt, daß Leute, die das comFORTH für Windows nicht haben "in die Röhre gucken müssen".

Eigentlich ist es ein einfaches Verfahren, bei dem zur Zeit der Compilation die Informationen über das gerade compilierte Wort (Name, Kommentar, Vokabular,...) in eine Datei geschrieben werden. Das sollte wohl mit den meisten Forth-System durch entsprechende Verbiegungen möglich sein. Um die Informationen über das Hilfesystem von Windows als On-Line-Glossar verfügbar zu machen, ist die bewußte Datei ledig-

lich im RTF-Format mit einigen speziellen Formatierungen angelegt worden. So kann sie anschließend mit dem MS-Hilfecompiler in eine On-Line-Hilfe übersetzt werden.

Aber Du hast Recht Jörg: im comFORTH für Windows Professional wird dieses Tool fertig ausprogrammiert zur Benutzung bereitstellen. Ausprobieren, und dazu sollte der Vortrag anregen, kann es allerdings jeder, der ein Forth-System hat (und ein wenig daran herumbiegen kann), sich ein bißchen mit dem RTF-Format auseinanderzusetzen und den Hilfecompiler von MS besitzt (gibt's glaube ich auch via FTP).

MfG Udo

*us@egd.igd.fhg.de (Udo Schütz), 7/95*

Ja, nur irgendwie sind in Jörgs Artikel einige Orthographieflaws(tm) 'reingerutscht. Ob das an Jörgs Pentium liegt? Und was ist aus dem Monospaced-Font geworden? Sieht immer noch nach Courier aus. [der Font wird fleißig für Listings verwendet. Besten Dank / red]

*paysan@informatik.tu-muenchen.de (Bernd Paysan) in de/comp/lang/forth 7/95*

## **Zu 'Hochsprachen in Silizium' in VD 2/95, S.31**

Liebe VD Redaktion,

ich bin kein großer Leserbriefschreiber, aber der Artikel von Rafael Deliano "Hochsprachen in Silizium" enthält einige gravierende Fehler über die PDP-11. In der Reihenfolge des Auftauchens im Artikel:

1.) Die VAX ist keine aufgebohrte PDP, sondern besitzt eine vollkommen andere Architektur. Das einzige,



## 28 CASE CONSTRUCTIE

Fred Burmester

Eakers CASE und kein Ende: Interessante kurze Darstellung (2 Seiten). Alles Doppelpunkt-Definitionen. Im dpANS-5-Standard-Implementierungsvorschlag für den Eaker-CASE benötigt jedes OF-ENDOF-Paar 7x2 = 14 Bytes im Compilat. In Fred Burmesters Implementierungsvorschlag werden nur 3x2 = 6 Bytes per OF-ENDOF benötigt. Meinung des Rezensenten: Der Autor hat recht. Da läppert sich ganz schön was zusammen. Ich arbeite mit einem selbstgebastelten Drucker-Einstellprogramm für den NEC-P6. Darin verwende ich einen CASE-Konstrukt mit 512 Eingängen. Da kann ich also 4 Kilobyte sparen. Naja (?) Aber immerhin! Das sind schließlich genau die 4 Kilo, um die mein Druckerprogramm beinahe für Turbo-Forth zu lang geworden wäre.

## 30 Het beste van USENET

Lennart Benschop

17 Seiten (!) Streifzug durch USENET: Bericht von euroForth '94. Pentium-Fehler-test (Jörg Plewe, uns allen aus der VD bekannt). Gedanken über die "unmögliche" Notation in Chuck Moores erstem Forth und über den Fortschritt, den ANS bedeutet. Assembler-Quelltext zu Assembler-Quelltext metacompilieren? Ein paar weitere Beiträge zum selben Thema. Und nochmal Jörg Plewe, auch zu diesem Thema.

## 47 Foutafhandeling in de 8051 Forth.

Willem Ouwerkerk

Kurze Abhandlung (1 Seite): In den meisten Forths wird die Compilation abgebrochen, sobald ein nicht definiertes Wort angetroffen wird. Einige Forths handhaben die Fehlerbehandlung anders: Im 8051-Forth wird das Konstrukt EVALUATE compiliert. Wird die Definition des Wortes später nachgeliefert, dann arbeitet das betreffende Wort schließlich richtig, wenn nicht, wird am Ende des Compiliervorgangs eine Fehlermeldung ausgegeben. (Frage des Rezensenten: Was macht denn DEFER bei Turbo-Forth usw. anders? Nur muß man DEFER eben explizit angeben, und es verbraucht eine Zelle Speicherplatz. Zumindest wäre ein Hinweis auf DEFER angebracht. DEFER (und EVALUATE ?) bricht schließlich ein Forth-Paradigma: Kein Wort ohne vorherige Definition.)

## 48 Uit de oude doos (1)

Coos Haak

In diesem kurzen Lückenfüller (?) gräbt der Autor die Worte @BITS und !BITS aus Forth-79 aus und stellt ein paar ungläubige Fragen.

## Forth-Systeme:

### FASTGRAF

#### ein Graphikpaket für Turbo-Forth aus Frankreich

Produkt: Grafikpaket für Turbo-Forth  
Preis: 250 FF  
Diskette 3 1/2" mit etwa 750kB

Handbuch 157 Seiten (französisch)  
Bezug:

**Marc Petremann**  
MP7  
17, allée de la Noiseraie  
F-93160 Noisy le Grand

Das von Marc Petremann, Michel Zupan und anderen Mitgliedern der (anscheinend nicht mehr bestehenden) französischen Forth-Gruppe JEDI in den Jahren 1989 und danach entwickelte Turbo-Forth (Forth 83 ohne Screens, mit DOS-File-Anbindung) ist um ein Graphikpaket erweitert worden. Nähere Informationen gibt es direkt von Marc Petremann. Er versteht Deutsch.

FASTGRAF ist ein residentes Programm, das mehr als 200 in hohem Maße optimierte Routinen bereitstellt, die von Turbo-Forth aus über Softinterrupts aufgerufen werden können. Der Quelltext zum Treiberprogramm, FGDRIVER.EXE, findet sich in FASTGRAF.FTH. (FTH-Dateien sind Dateien, die normalerweise Turbo-Forth-Anwendungsprogramme enthalten, die per INCLUDE XXX.FTH hinzugeladen und dann per SAVE-SYSTEM als COM- oder EXE-Dateien abgespeichert werden können.) Diese Routinen sichern die Graphikanbindung beim PC und dem PS/2. Sie gestatten auch den Zugriff auf Tastatur, Maus, Joystick und Soundkarte.

Hier ein paar Eigenschaften, die ich dem Handbuch entnommen habe: Hauptmerkmal von FASTGRAF ist, wie der Name andeutet, seine Schnelligkeit.

FGDRIVER ist weitgehend in 8086-Assembler geschrieben und die Routinen wurden auf maximale Leistung getrimmt. Von Forth aus geht der Aufruf sogar schneller als von C aus. Bei der Entwicklung der Forth-Primärworte, die FGDRIVER ausmachen, hat sich herausgestellt, daß ihre Entsprechungen aus der C-Unterprogramm-bibliothek zwei Unterprogramm-aufrufebenen (instruction call) durchlaufen müssen und ihre Werte indirekt indiziert über BX in die Register laden. Forth erledigt das mit einfachen Stackoperationen über PUSH und POP und ist damit schneller.

FASTGRAF unterstützt alle Text- und Graphik-Modi der Systeme vom Typ PC (PC, PC/XT, PC/AT) und PS/2, sowie alle kompatiblen Systeme. Darüberhinaus unterstützt FASTGRAF vier erweiterte VGA-Modi und den 16-Farben-Graphikmodus des Tandi 1000 und des PCjr. FASTGRAF unterstützt auch noch den HERCULES-Modus.

FASTGRAF kann in Verbindung mit Turbo-Forth 2.0, 2.1 oder 3.0 verwendet werden, Versionen, die wahlweise auf 8086 oder 80386 geschaltet werden können. Die in FASTGRAF.FTH enthaltenen Routinen lassen sich ohne große Schwierigkeit auch auf andere Forth-Systeme übertragen, vorausgesetzt, es handelt sich um Systeme für PCs oder PC-Kompatible.

*Fred Behringer, München*

## 27.-29.10.1995: EuroForth'95 auf Schloß Dagstuhl

Vom 27. bis 29. Oktober 1995 findet auf Schloß Dagstuhl bei Saarbrücken mit Unterstützung der GI Fachgruppen 2.1.4 'Alternative Konzepte für Sprachen und Rechner', 3.1.3 'Entwurf und Architektur hochintegrierter Schaltungen' und der ITG die EuroForth'95 Konferenz statt. Erwartet werden auch zahlreiche internationale Gäste aus dem europäischen und dem nichteuropäischen Ausland. Studenten und Teilnehmer aus Osteuropa erhalten Sonderkonditionen.

Anmeldungen werden gerne entgegengenommen von:

*euroFORTH office: Marina Kern; c/o Delta t;  
Adenaue Allee 54; 20097 Hamburg*

*(uho)*

## 5.-7.12.1995 RTSS'95: 16th IEEE Real-Time Systems Symposium, Pisa, Italy

The purpose of this symposium is to bring together researchers and developers from academia, industry, and government to advance the science and technology in real-time computing. Papers on all aspects of real-time computing are sought, including operating systems and scheduling, fault-tolerance, databases, programming languages, tools, communication networks, architectures, performance modeling, formal methods, case studies, and applications. Of particular interests are reports describing practical experiences and experimental results based on system building efforts, and real-time issues in applications such as avionics, multimedia, robotics, automated process control and manufacturing.

*RTSS'95 WWW home page is available at  
<http://cs-www.bu.edu/pub/ieee-rtss/rtss95/>*

*contact: Azer Bestavros, Computer Science Department;  
111 Cummington St; MCS-276; ; Boston University,  
Massachusetts 02215; Phone:(617) 353-9726*

*(/comp/lang/forth, best@cs.bu.edu -Azer Bestavros, Di 24.01.95)*

## 6.11.1995 WRTP'95: 20th IFAC/IFIP- Workshop on REAL TIME PROGRAMMING

The Workshop at Fort Lauderdale, Florida is to promote interaction among researchers and practitioners and to evaluate the maturity of new directions in real-time software. The conference is being held in conjunction with the first International Conference on Complex Computer System (ICECCS), which is sponsored by the IEEE TSC on Complex Computer Systems. Topics are: real-time programming languages, engineering methods, Software design methods, scheduling and resource management, communication architectures (incl. MAP and Fieldbus), Verification and validation (esp. of timing properties)

*contact & german chair: W.A. Halang, FernUniversitaet, Faculty of  
Electrical Engineering, D-58084 Hagen, Germany, Tel.:+49-2331-987-372,  
Fax: +49-2331-987-375, E-mail: wolfgang.halang@fernuni-hagen.de  
comp/realtime, bruce@cs.sunysb.edu - Bruce McMillin, Di 07.02.95*

## 7.-9. 12.1995 IDPT: Integrated Design & Process Technology

The First World Conference on INTEGRATED DESIGN & PROCESS TECHNOLOGY am 'IC2 Institute at the University of Texas, Austin'. The IDPT Conference is emerging as a major international forum for the integration of design and process technologies to create a symbolic relationship among scientists, engineers, decision makers, and other critical thinkers.

There are isolated islands of knowledge which are often treated as being

oceans apart. These islands of knowledge coexist disparately with little productive exchange. Thus the Society for Design and Process Science Call (SDPS) promotes efforts on facilitating integration among disciplines; consider the economical, environmental, and ethical aspects of human endeavors; to foster greater interdisciplinary cooperation and international collaboration.

*For more information about the SDPS phone (806) 742-0967 or e-mail to  
meae@coe3.coe.ttu.edu*

*Submit abstracts to: Atila Ertas; Mechanical Engineering Dept.; Texas  
Tech University; PO Box 1021 Lubbock, TX 79409-1021. Phone (806) 742-  
3563; FAX (806) 742-3540; e-mail meae@coe3.co3.ttu.edu.  
(from comp/realtime, 19.2.95 by gatch@aol.com - Gatch)*

## 14-18.4.96 ECCV '96: Fourth European Conference on Computer Vision

University of Cambridge, England. Following the highly successful conferences held in Antibes, Santa Margherita Ligure and Stockholm, the Fourth European Conference on Computer Vision will be held from 14-18 April 1996 in Cambridge, England. The Conference is to be held under the auspices of the European Vision Society (EVS) and the British Machine Vision Association (BMVA).

*ECCV'96 Conference Contact; 42 Devonshire Road;  
Cambridge CB1 2BL; England. tel: +44/0 1223 323437; fax: +44/0 1223  
460396; e-mail: cc@confcon.demon.co.uk  
(comp/robotics, Di 07.02.95, ahg@eng.cam.ac.uk -A.H.Gee)*

## 20-22.5.1996 ICEC'96: IEEE International Conference on Evolutionary Computation

Nagoya, Japan. Special Session on Evolutionary Robotics and Automation: More and more researchers are applying evolutionary computation techniques to challenging problems in robotics and automation, where classical methods fail to be effective. In addition to being vastly applicable to many hard problems, evolutionary concepts inspire many researchers as well as users to be fully creative in inventing their own versions of evolutionary algorithms for the specific needs of different domains of problems. This special session serves as a forum for exchanging research results in this growing interdisciplinary area and for encouraging further exploration of the fusion between evolutionary computation and intelligent robotics and automation.

ICEC'96 is co-sponsored by IEEE Neural Network Council (NNC) and Society of Instrumentation and Control Engineers (SICE). It will be organized in conjunction with the conference of Artificial Life (Kyoto, JAPAN, May 16-18, 1996).

*Contact (General Chair): Toshio Fukuda; Dept. of Micro System  
Engineering and Dept. of Mechano-Informatics and Systems Nagoya  
University; Furo-cho, Chikusa-ku, Nagoya 464-01, JAPAN; Phone:+81-  
52-789-4478; Fax: +81-52-789-3909; E-mail: fukuda@mein.nagoya-u.ac.jp  
/comp/realtime, xiao@uncc.edu (Jing Xiao), Di 21.02.95)*

## 14.2. - 16.2.96 Embedded Systems, Stuttgart

Am 14. bis 16.2.1996 veranstalten MagnaMedia AG und Drebingen in Sindelfingen die Messe "Embedded Systems '96". Ob sie die "Echtzeit" ergänzt oder ersetzt, ist unklar.

*jrd (Rafael Deliano), 9/95*



# ====Termine====

(zu den fett gedruckten Veranstaltungen findet sich eine Kurzbeschreibung auf diesen Seiten)

- 03.10. - 04.10.95 CAN Conference, London
- 16.10. - 20.10.95 SYSTEMS 95, München
- 27.10. - 29.10.95 EuroForth 95, Saarbrücken**
- 30.10. - 04.11.95 Interkama, Düsseldorf
- 07.12. - 09.12.95 IDPT, Austin (Texas)**
- 29.1. - 31.1.96 MULTIMEDIA & Net, California**
- 14.2. - 16.2.96 Embedded Systems '96. Stuttgart**
- 14.4. - 18.4.96 ECCV 96, Cambridge (GB)**
- 24.9. - 26.9.96 Opto 96, Leipzig

# ====Termine====

## Die weniger bekannten Programmiersprachen (44.Folge)

Nachdem wir uns in den bisherigen Folgen mit der einen wirklich wichtigen Programmiersprache und ihren Derivaten beschäftigt haben, wagen wir heute den Blick über den Tellerrand.

...

### THE LESSER-KNOWN PROGRAMMING LANGUAGES LITHP

This otherwise unremarkable language is distinguished by the absence of an "S" in its character set; users must substitute "TH". LITHP is said to be useful in protheththing liththh.

### THE LESSER-KNOWN PROGRAMMING LANGUAGES SLOBOL SLOBOL

is best known for the speed, or lack of it, of its compiler. Although many compilers allow you to take a coffee break while they compile, SLOBOL compilers allow you to travel to Bolivia to pick the coffee. Forty-three programmers are known to have died of boredom sitting at their terminals while waiting for a SLOBOL program to compile. Weary SLOBOL programmers often turn to a related (but infinitely faster) language, COCAINE.

*in der KBBS gefunden, gepostet von Juergen Schardt*

When someone says "I want a programming language in which I need only say what I wish done," give him a lollipop.

*(wird fortgesetzt)*

## Abstracts der EuroForth '94

"Following are the abstracts form the EuroForth '94 conference, entitled Exploiting Forth : Professionally, Commercially and Industrially. "

So beginnt ein Text von knapp 20 kB mit den Abstracts der EuroForth '94. Die kompletten Proceedings können als ISBN 0 952 5310 03 bei MPE bezogen werden kann:

**MicroProcessor Engineering Ltd**  
133 Hill Lane  
Southampton  
England  
SO15 5AF  
Email: mpe@mpeltd.demon.co.uk

### Hier nur die Titel:

- Stack Caching for Interpreters / M. Anton Ertl
- Combined addressing model for 8086 processor /M L Gassanenko
- The Effects Of Local Variable Optimisation On A C-Based Stack
- Processor Environment / C Bailey et al
- Forth and Register Transfer Language / Graeme Dunbar / John Boyce
- Automatic Scoping of Local Variables / M. Anton Ertl
- BacFORTH: An approach to new control structures / M L Gassanenko
- Forth and Formal Language Theory / Jaanus Poial
- How to make money out of FORTH / Ben Campbell
- Embedded System Debugging: The return of software / Stephen Pelc (MPE)
- Validating Forth Source Test -

- A Design Concept / Howerd Oakford & Inventio Software
- Rhyme, Reason and the Tao of Forth\* / Lawrence P. G. Forsley
- Going Forth with Echelon / Malcolm Bugler
- A Distributed Data Acquisition and Decision Making System / Tim Hendtlass
- A GUI Toolkit for polyFORTH / Elizabeth Rather (FORTH Inc )
- Tool interactivity for rapid GUI application development / Gary Ellis & Roy Goddard
- Commercial exploitation of the interactive advantage / Roy Goddard
- A Forth Based Process Control Software Product / Elizabeth Rather
- An ANS Heap / Gordon Charlton
- A Library versus A Kernel / Alan M Robertson
- Handling source code and images in natOOO / Markus Dahm
- Interactive Remote Target Compilation and the PIC 16CXX /Alan M Robertson BSc
- A Taste of Direct Programming / Wolf Wejgaard (Forth Engineering )
- Breakthrough In Knowledge Management / Bjorn J. Gruenwald

*clv nach einer Mail von cloan@mpeltd.demon.co.uk (Joan Perham) in comp/lang/forth 7/95*

## 29.-31.1.1996: Multimedia Computing and Networking 1996, San Jose, California

This conference is part of the IS&T/SPIE 1996 International Symposium on Electronic Imaging to be held on Jan 28 - Feb 2, 1996.

Advances in computer and networking technologies have fueled the rapid growth of research and development in multimedia computing and high-speed networking. The conference will serve as a forum for the dissemination of state-of-the-art research, development, and implementation of multimedia systems, technologies, and applications. Presenters will be encouraged to make multimedia presentations and demonstrate their solutions.

Papers are solicited in all areas of multimedia, including, but not limited to: Multimedia Computing, Networking, User Interfaces, Authoring Systems, Applications, Standards, Documents, and Data Interchange.

*contact: Professor Harrick M. Vin  
Department of Computer Sciences; Taylor Hall 2.124  
The University of Texas at Austin; Austin, TX 78712-1188  
Phone: (512) 471-9732; E-mail : mmcn96@cs.utexas.edu*

*(comp/realtime, 9.4.1995)*

---

---

## Vorschlag für eine Arbeitsgruppe

# MARC4

---

---

Der Prozessor MARC4 wurde Ende 1986 von dem in Eching bei München beheimateten Halbleiterhersteller EUROSIL vorgestellt. Es handelt sich um einen 4-Bit-Stackprozessor. Für die Programmierung wird vom Hersteller qFORTH angeboten. Assembler ist möglich, bringt aber kaum Vorteile. FORTH ist ähnlich nahtlos wie beim NOVIX eingebunden. Der Chip wurde in einem Artikel der EUROSIL-Mitarbeiter Göttle und Kobenter in der VD 4/88 genauer beschrieben. Da er eigentlich für Großstückzahlen gedacht ist und damals auch der ROM-less-Chip für praktische Experimente nicht existierte, hat sich anscheinend kaum jemand damit praktisch beschäftigt. EUROSIL wurde später an TEMIC (=Telefunken =Daimler-Benz) verkauft. Die Ferti-

gung in München wurde eingestellt und nach Fernost verlagert. Und damit sah es so aus als wäre wiederum ein FORTH-Prozessor gestorben.

Der Chip scheint aber schon zu EUROSIL-Zeiten profitable designs gehabt zu haben. Mit einer Geldspritze vom BMFT wurde die CPU-Standardzelle von der Firma GEMAC in Chemnitz auf Silicon-Compiler umgestellt. Damit ist auch Fertigung auf Strukturweiten im Bereich 0,6 - 0,8µ CMOS möglich. Womit der Chip wohl auch in Zukunft verfügbar sein dürfte.

TEMIC gibt umfangreiche, gut gemachte Handbücher zum MARC4 heraus, in denen auch eine Einführung in FORTH erfolgt und die Vorzüge von FORTH herausgestellt werden. qFORTH ist, abgesehen von den

hardwarebedingten Modifikationen, ein ziemlich normales FORTH. Ergo haben wir hier den seltenen Fall, daß ein renommiertes und nicht eben kleines deutsches Unternehmen in einem kommerziell erfolgreichen Produkt explizit FORTH unterstützt.

Man sollte die sich daraus ergebenden Möglichkeiten nutzen. Ich darf daran erinnern, daß die Vorstellung des RTX2000 von Harris auf der Elektronik 88 das Interesse an FORTH hierzulande beträchtlich gefördert hat. Und für den damaligen Mitgliederzuwachs der e.V. mit von Bedeutung war. Der MARC4 ist zwar kein "high-end-RISC"-Prozessor, aber das Microchip-PIC-Ende des Marktes ist fürs Publikum genauso von Interesse. Zumal es preiswerter ist.

Es gibt inzwischen vom MARC4 ROM-less-Chips und die Verfügbarkeit eines OTP steht unmittelbar bevor. Mit ihnen ist praktisches Arbeiten möglich. Man muß jetzt nur noch für eine low-cost Entwicklungsumgebung sorgen. Das kann durch eine Arbeitsgruppe "MARC4" der Forth e.V. geschehen. Vorausgesetzt das Direktorium stimmt zu, vorzugs-

weise als offizielles Projekt derForth e.V.. Da sich die Arbeitsgruppe besonders auch an Anwender außerhalb des Vereins wendet, verhilft das dem Verein zu Publizität und Mitgliedern.

Aufgabe der Arbeitsgruppe ist: die Schaffung der low-cost Entwicklungsumgebung, die Verbreitung des MARC4, der Informationsaustausch unter den Anwendern. Neben der Entwicklung von Hardware und Software ist die Herausgabe eines Newsletters a la "CreepingFORTH" sowie Öffentlichkeitsarbeit zum MARC4 in den Fachzeitschriften vorgesehen. Klaus Kohl wird den Vertrieb von Produkten und Literatur übernehmen.

Wer sich an der Entwicklung der Werkzeuge beteiligen will, sollte möglichst frühzeitig mit uns Kontakt aufnehmen. Die Anwender hingegen sollten sich bis Januar gedulden, bis wir die Vorbereitungen abgeschlossen haben.

*MfG Rafael Deliano*

---

Fortsetzung von Seite 8

daß ich in den vergangenen Jahren prinzipiell alles abgelehnt habe, womit sich die 'Mäuseschubser' intensiv auseinandergesetzt haben. Ich verstehe einfach nichts. Gerade noch an den Doppelpunkten kann ich erkennen, wo eine Definition anfängt. Aber was mit den 5380 (!) Worten in WIN32FOR's Vokabularen alles anzufangen ist, zeigt das System in beeindruckender Weise auf seiner Oberfläche. Schade, daß es nicht dazu zu bewegen ist, unter WINOS/2 zu arbeiten. Tom Zimmer's jüngstes Werk könnte mich dazu bewegen, mich endlich auch mit Klassen und Methoden zu befassen. Tom Zimmer, Andrew McKewan, Andy Korsak, Y.T. Lin, Robert Smith und Jim Schneider haben mit WIN32FOR ein Forth-Entwicklungssystem geschaffen, das zur Zeit ganz sicher noch viele Wünsche offen läßt, aber durchaus bereits eine Menge bietet. Das 32-Bit 'flat' Memory Modell wird voll unterstützt, ebenso wie die 32-Bit API von Windows95, bzw. WindowsNT schreibt Tom Zimmer zumindest über WIN32FOR. Objektorientierte Tools sollen ebenso zur Verfügung stehen, wie der Zugriff auf alle (!) Windows DLLs. Schnell soll WIN32FOR schon zur Compilezeit sein, und die meisten vom F-PC bekannten Tools seien ebenfalls integriert - heißt es in der Datei

FORML94\*. Die Entwicklungsumgebung nutzt den externen Hypertexteditor NEWZ (F-PC Usern wohl bekannt), der von der Windowsmanie abweicht und noch "oldfashioned" daher kommt - sprich: als zeichenorientiertes Programm im "Textbildschirm". Den integrierten 486er Assembler hätte ich zu gerne ausprobiert. Vielleicht kann man ihn in ruhigen Winterstunden für das ZF nutzbar machen.

Wenn noch irgendjemand freie Zeit in WIN32FOR investieren möchte, dann sende ich ihm (ihr) auf Anfrage gerne die Diskette zu, oder ich lege deren Inhalt in der MHB ab...

*F.Prinz*

*[Eine Anfrage bei AndrewMcKewan ergab: win32forth läuft nicht (!) unter OS/2. Unter Windows 3.1 läuft es nur mit win32s, das F.Prinz fehlte.]*

### **Forth-Systeme: win32forth doch unter OS/2?**

Run in Windows 3.1 enhanced mode - I use the following settings in addition to my default WINOS/2 session configuration:

XMS\_MEMORY\_LIMIT=8192

```
DPMI_MEMORY_LIMIT=64
DPMI_NETWORK_BUFFER_SIZE=1
AUDIO_ADAPTER_SHARING=Optional
VIDEO_SWITCH_NOTIFICATION=1
VIDEO_8514A_XGA_IOTRAP=0
```

Set the default directory to where you've put the WIN32FOR program files. The software should run in "seamless" mode, no problem.

If you still have problems getting it to run, email me , I'll email you a copy of my full WINOS/2 settings.

*lkaplan@BIX.com (Leonard Kaplan)*

### **Forth-Systeme: Mops 2.6 released**

To all Mops users (or potential users):

Version 2.6 is now released. It can be obtained from the usual places:

<ftp://taygeta.oc.nps.navy.mil/pub/Forth/Mops/Mops26s.sea>

<ftp://taygeta.oc.nps.navy.mil/pub/Forth/Mops/Mops26m.sea>

and also via Jay Farrell's Mops web page:

<URL:http://www.netaxs.com/~jayfar/mops.html>

Like 2.5, this is mainly a consolidati-

on release. Since 2.5, I added a few useful new features that people were asking for, and released them as the 2.5.1 update. Inevitably a few bugs crept in from somewhere, so 2.6 is intended to consolidate everything in a single release and fix the bugs.

Also Craig Treleven has completely reformatted the manual, including adding proper styles on everything, so that you can easily customize the way the manual looks to your own liking. Thanks to Craig for a magnificent job.

(One minor correction to the Release Notes: I state there that Jay Farrell was working on an application framework based on Mops - I was mixed up. Jayfar runs the web page. It's Johnathon McAllister who's doing the framework.)

-- Mike. mikeh@zeta.org.au  
(Michael Hore) in comp/lang/forth, Juni '95

## Forth-Gruppen regional

<b>Berlin</b>	Claus Vogt Tel.: +30 -782 81 79 p Treffen: nach Absprache
<b>Rhein-Ruhr</b>	Jörg Plewe Tel.: +208 -49 70 68 p Treffen: jeden 1. Samstag im Monat im S-Bahnhof Derendorf Münstererstr. 199,
<b>Moers</b>	Friederich Prinz Tel.: +2841 -5 83 98 p Treffen: jeden Samstag 14:00 Arbeitslosenzentrum, Donaustr. 1, Moers
<b>Darmstadt</b>	Andreas Soeder Tel.: +6257 -27 44
<b>Mannheim</b>	Thomas Prinz Tel.: +6271 -28 30 p Ewald Rieger Tel.: +6239 -86 32 p Treffen: jeden 1. Mittwoch im Monat, Vereinslokal Segelverein Mannheim e. V., Flugplatz Mannheim-Neuostheim

## µP-Controller Verleih

Thomas Prinz  
Tel.: +6271 -28 30 p

## Gruppengründungen, Kontakte

**Regional**  
Stuttgart  
Wolf-Helge Neumann  
Tel.: +711 -8 87 26 38 p

**Fachbezogen**  
8051 ... (Forth statt Basic,  
e-FORTH)  
Thomas Prinz  
Tel.: +6271 -28 30 p

## Forth-Hilfe für Ratsuchende

**Forth allgemein**  
Jörg Plewe  
Tel.: +208 -49 70 68 p  
plewe@mpi-dortmund.mpg.de  
  
Karl Schroer  
Tel.: +2845 -2 89 51 p  
  
Jörg Staben  
Tel.: +2103 -24 06 09 p

## Spezielle Fachgebiete

<b>Anfänger und Wiedereinsteiger</b>	Gerd Limbach Tel.: +2051 -25 51 12 p Mo.+Di. 20:00 - 22:00
<b>32FORTH (Atari)</b>	Rainer Aumiller Tel.: +89 -6 70 83 55 gp
<b>FORTHchips (FRP1600, RTX, Novix...)</b>	Klaus Schleisiek-Kern Tel.: +40 -330 674 p
<b>F-PC &amp; TCOM, ASYST (Meßtechnik), embedded controller(H8/5xx//TDS2020, 8051 ... eFORTH...), FUZZY</b>	Arndt Klingelberg Tel.: +2404 -6 16 48 agp
<b>Gleitkomma-Arithmetik</b>	Andreas Döring Tel.: +721 -59 39 35 p
<b>HS/Forth (Harvard Softworks)</b>	Wigand Gawenda Tel.: +30 -44 69 41 p
<b>KI (Künstliche Intelligenz), OOF (Object Oriented Forth)</b>	Ulrich Hoffmann Tel.: +431 -80 12 14 p  Birgit Steffenhagen Tel.: +38204 - 129 33 pa +381 - 498 35 52 g
<b>Unterricht mit Forth</b>	Rolf Kretzschmar Tel./Fax: +2401 -8 88 91 ap
<b>UUCP (FORTH ... per eMAIL)</b>	Andreas Jennen Tel.: +30 -3 96 52 27 ap
<b>Forth-Vertrieb</b>	<b>volksFORTH/ultraFORTH, RTX/FG/Super8/KK-FORTH</b> Ingenieurbüro Klaus Kohl Tel.: +8233 -3 05 24 p Fax: +8233 -99 71 f
<b>Forth-Mailbox (KBBS)</b>	+431-533 98 98 :8N1 Sysop: Holger Petersen Fax: +431-533 98 97 f Tel: +431-533 98 98 p <i>bis 22 Uhr</i> Mail: hp@kbbs.org

## Hinweise

### **Zu den Telefonnummern**

f == FAX  
a == Anrufbeantworter, hier können Sie Ihren Ansprechpartner eventuell vorinformieren,  
erwarten Sie bitte keinen (kostspieligen) Rückruf  
g == geschäftlich, zu erreichen innerhalb typischer Arbeitszeiten  
p == privat, zu erreichen außerhalb typischer Arbeitszeiten

Die Adressen des Forth e. V. (Forth Büro) und der Redaktion/ finden Sie im Impressum

# FORTECH Software

- Forth-Entwicklungsumgebung comFORTH unter DOS oder Windows
- interaktive Crossentwicklungssysteme für Mikroprozessoren von Intel, Motorola, Zilog, TI ...
- Softwareentwicklung für PC und Mikrocontroller
- System- und Anwendungsprogrammierung unter Windows

## comFORTH

- Forth-Entwicklungsumgebung für Windows
- interaktive Benutzbarkeit aller Windows-API-Funktionen und -Strukturen
- kombinierbar mit anderen Programmiersprachen
- Unterstützung von DDE, DLL, VBX, ...

## fieldFORTH

- Forth-Entwicklungssystem für eingebettete Systeme
- interaktive Programmierung off-line und on-line
- verfügbar für diverse 8-, 16- und 32-Bit Mikrocontroller und -Prozessoren (TMS320C40, M68332, M68HC11,...)
- **NEU!!!** Evaluation-Kit M68HC11 inclusive Board MINI-HC11 296,70 incl. MwSt.

FORTECH Software GmbH

J.-Jungius-Str. 9 • D-18059 Rostock • Tel: (03 81) 4 05 94 72 • Fax: (03 81) 4 05 94 71