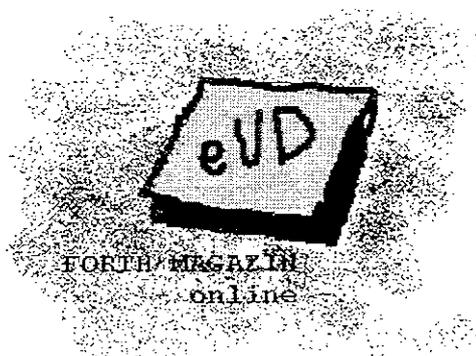


VIERTE DIMENSION

4/1996

12. Jahrgang 1996, 4. Quartal, DM 10.-



- Stackprozessor als FPGA
- Schöne Sachen selber saugen
- Serie PC-Meßtechnik, Teil VI - PC-Meßkarten
- An oder Aus - High oder Low
- Schachcomputer - Die historische Entwicklung
- Die 64 Constant-Falle im ZF
- Forth International
- Über Eakers CASE in Forth und IF in Occam2
- From the other side of The Big Teich
- Intelhex-Format
- Forth Online
- N'bißchen wat Praxis
- Freitexttabellenkalkulation

FORTH MAGAZIN

Organ der Forth Gesellschaft e.V.
<http://falbala.informatik.uni-kiel.de/~uho/VD/>

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

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 Einsatzbedin-
gungen in Walzwerken, KKW, Medizin,
Verkehr / >20 Jahre Erfahrung.

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)

**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.

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. Profession-
nelle Entwicklung für Steuerungs- und
Meßtechnik.

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.

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

FORTH MAGAZIN - VIERTE DIMENSION
Organ der Forth-Gesellschaft e.V.

Herausgeberin

FORTH-Gesellschaft e.V.
Postfach 1110
85701 Unterschleißheim
Tel./Fax: 089/3173784
Mail: secretary@admin.FORTH-eV.de

Redaktion & Layout

Claus Vogt
Katzbachstr. 23
Tel.: 030/786 84 60 (Fax & E
Mail: vd@FORTH-ev.de

*Neue Adresse
ab Juli '96*

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; 81241 München.

Zeichnungen: Rolf Kretzschmar;

Titelbild: Hans-Georg Schmid

Redaktionsschluß '96

Erste Januar-, April-, Juli-, und Oktoberwoche.

Erscheinungsweise

Viermal im Jahr.

Preis

Einzelpreis: DM 10,-

Manuskripte und Rechte

Berücksichtigt werden alle eingesandten Manuskripte. 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.



Multimedial.. (lat.:vielmittelig)

Wenn man dem Duden Glauben schenkt, bezeichnet als 'medial' der Mediziner jene Körperteile, die zur Mitte hin gelegen sind. Im Körper der Forth-Gesellschaft e.V. darf sich das Forth Magazin sicher als zentrales Organ wännen - es ist ja auch ein Medium. Daß es mehrere Mitten geben soll, wundert ein wenig. Aber die alljährliche Forth-Tagung ist seit nunmehr 12 Jahren etabliert. Auch das Usenet-Brett de.com.p.lang.forth bietet Forthlern - zumindest den vielen Modembesitzenden unter ihnen - ein zentrales Diskussionsforum.

Auch eine CD-ROM wie die Weihnachts-CD 1995 ist ein Medium. Ob sie allerdings eine Mitte bilden kann, sei dahingestellt. Selbst ist sie zwar schön rund, aber da, wo andere Dinge eine Mitte haben, ist ein Loch. Schön rund, werksseitig sorgfältig ausgestanzt, hoffentlich einer stofflichen Wiederverwertung zugeführt. Dieses Weihnachten wird es keine CD (-Ankündigung) geben. Klaus Kohl arbeitet fleißig daran und hofft, sie zum Osterfest vorlegen zu können.

Pünktlich zum Weihnachtsfest hat die Forth-Gesellschaft nun ein weiteres Medium bekommen. Die rastlose Tätigkeit von Michael Schröder hat es ermöglicht. Man sieht dem Produkt an, daß eine Menge Arbeit und Überlegung darin steckt. Ulrich Hoffmann ermöglichte die technische Seite des Pilotprojekts. Innendrin steckt das Forth-Magazin Vierte Dimension. Und wenn es sich denn zu einer wirklichen Mitte der deutschsprachigen Forth-Gemeinde entwickeln will, dann benötigt es außer dem FTP-Server die Pflege, Kritik und das Interesse möglichst vieler Mitmenschen. Unser neues Medium hat leider noch keinen sehr schönen Namen ...

- Claus Vogt, Editor des Forth Magazins Vierte Dimension -



In eigener Sache:

Aus formalen Gründen scheint eine Umstrukturierung der Finanzierung des Forth-Magazins sinnvoll. Da hierzu ein Direktoriumsbeschluß nötig erscheint, wird sich das Erscheinen des Forth-Magazins VD 1/97 um unbestimmte Zeit verzögern.

Wir danken für Ihr Verständnis /red.

<http://falbala.informatik.uni-kiel.de/~uho/>

MARC4

-Newsletter der FORTH e.V.



Diese Ausgabe

bringt eine kommentierte Darstellung des qFORTH Befehlssatzes nach dem Stand des TEMIC-Handbuchs von 1984. Eine Beschreibung des Assemblers findet sich in Ausgabe 1. Um Platz zu sparen wurde er hier nicht nochmal besprochen. Aus gleichem Grund ist die Darstellung zu knapp um als Einführung in FORTH zu dienen. Diese folgt in einer späteren Ausgabe. Einzelheiten sei dafür auf die einschlägige Literatur verwiesen.

qFORTH

Mit ca. 150 Befehlen ist qFORTH recht umfangreich geraten. Es ist deshalb sinnvoll sich erstmalig auf die häufig verwendeten Befehle zu konzentrieren. Hier werden die Exoten deshalb in die jeweiligen Kapitel "Erweiterungen" abgeschoben. Selbst der Kern von qFORTH ist noch deutlich größer als bei kompakten Controller-FORTHs, da viele Befehle mehrfach vorhanden sind, abgestimmt auf die unterschiedlichen Datenwortlängen Nibble, Byte, 12 Bit.

Als der MARC4 entworfen wurde, war FORTH-83 der gängige Standard. qFORTH adaptiert deshalb FORTH-83 auf den MARC4. Inzwischen wurde 1994 ein ANSI-Standard verabschiedet, ANSI-Forth, der wahrscheinlich auch ein ISO-Standard wird. Praktisch ist ANSI jedoch nur ein aufgekochtes FORTH-83. Bezo-

Inhalt

- 1 qFORTH
- 2 Arithmetik
- 4 Vergleichsbefehle
- 5 Programmfluß
- 8 Stackbefehle
- 10 Speicherzugriff
- 11 Befehle & Strukturen
- 14 Assembler
- 15 Index der qFORTH-Befehle
- 16 Impressum

Ausgabe ②

★ ★ ★

gen auf den für den MARC4 sinnvollen Befehlsumfang sind die Abweichungen von FORTH-83 sehr gering. In dieser Beschreibung werden die wichtigsten Unterschiede von qFORTH zu FORTH-83 und ANSI-Forth kurz angesprochen. Bedingt durch die Einschränkungen der Hardware entsprechen die FORTHs für Stackprozessoren nie völlig den Standards. Der MARC4 macht da keine Ausnahme. Der wichtigste Unterschied ist natürlich, daß qFORTH einen 4 Bit breiten Stack hat, während sonst ein 16 Bit Stack typisch ist. Portabilität zu anderen FORTHs ist damit praktisch ausgeschlossen.

Rubriken

- 1 Titelbild
- 3 Editorial
- 6 Leserforum
- 7 Kurz berichtet
- 39 Adressen und Ansprechpartner

Inserenten

- 2 ForthWORKS
- 40 FORTHeCH

Forth-Gesellschaft

- 4 Marc4 Newsletter 2 erschienen
- 7 Forthtreffen in Kiel

Termine

- 4 RTS'97 in Paris
- 7 POPL'97 Call for Papers

Forth online

- 7 Neue Adressen
- 9 Überall Forth drin
Weltraum
FAQs in comp.lang.forth

Forth inside

- 8 Europay

Forthsysteme

- 8 Holon4th 4.0 released
win32for 3.3 released
win32for 3.3 workaround
ANS macht Dynoof-0.1.5
Risc OS Forthmacs letter Sep'96
F6 für Sinclair QL

Bücher

- 10 Vack "Programmieren mit Forth"
Lieferbare deutsche Bücher
- 23 Steinwender "Schach am PC"

Zeitschriften

- 10 FLEX TA64x30
M&T:Compilerübersicht

Forth und Musik

- 10 fLux

Was noch ..

- 20 Programmierer Erwin K.
- 37 Simple Interrupt System

Was fehlt

- Die Rubriken Anfänger, Produktinfo, Prozessorgeflüster, Firmen, und die Kolumne ANS Forth.
Rio Reiser (Sänger, ex. Ton/Steine/Scherben) starb am 20.8.96 an einem Herz-Kreislauf-Kollaps. Seymour Cray (Mitbegründer Control Data, Gründer Cray bis zum Konkurs 1972) starb am 5.10.96 an den Folgen eines Verkehrsunfalls.

Etwas verspätet nun die 2. Ausgabe des Newsletters. Sie gibt eine Einführung in qFORTH und erklärt die Unterschiede zu anderen FORTHs. Derweil laufen die Arbeiten an der nächsten Ausgabe, die die Beschreibung für ein Programmiergerät für den M48C260 enthalten soll. Es hat sich bisher als harte Nuß erwiesen, aber steter Tropfen höhlt den Stein.

Erhältlich ist Heft 2 des Marc4-Newsletters bei:

Forth Vertrieb Klaus Kohl
PF 1173
86406 Mehring
Zeppelinstr. 10
85415 Mering
Tel.: 08233-30524
Fax.: 08233-9971 f

□

Stackprozessor als FPGA

von Rafael Deliano..... 11

Längst haben wir uns an 'Forth in Hardware' gewöhnt. Da schallt aus dem Süden der Republik der Ruf nach einem variabel programmierbaren Silikonforth. Angepaßt an das Problem, die Kaufkraft des Kunden und die Tageslaune des Programmierers.

Schöne Sachen selber saugen

von Michael Kalus 13

Die KBBS - die Mailbox der Forth-Gesellschaft - braucht längst keine Werbung mehr. Ein paar Tips und Tricks sind manchmal doch nützlich.

**Serie PC-Meßtechnik, Teil VI
PC-Meßkarten**

von Klaus Kohl..... 15

Wie kriege ich die gerade gebaute Karte in den ISA-Slot? Kein Problem, wenn man sich (vorher) die Slotbelegung anschaut.

An oder Aus - High oder Low



von Arndt Klingelberg .19

Das Ausdrucken einzelner Bits gehört zur edelsten Kunst des puristischen Forthlers. Natürlich in High-Performance!

Schachcomputer - Die historische Entwicklung

von Rafael Deliano.....21

Wer sich nach einem harten Arbeitstag mit Forth nach Entspannung sehnt, sollte sich noch rasch einen Spielkameraden programmieren.

Die 64 Constant-Falle im ZF

von Martin Bitter24

Hätten wir weiter Kerben in Hinkelsteine geritzt, statt uns von den Arabern die Zahlen andrehen zu lassen, dann wäre uns diese Zahl auch erspart geblieben...

Forth International

von Fred Behringer22

Rochester findet zur Abwechslung mal in Toronto statt. 'Gehaltvolles' gibt es aus den Forth Dimensions (USA) von Jan/Feb und Sep/Okt 1996

Über Eakers CASE in Forth und IF in Occam2

von Fred Behringer27

Ein beliebtes Thema von einem beliebten Autor.

From the other side of The Big Teich

von Henry Vinerts.....28

Mit dieser Ausgabe beginnt Henry Vinerts eine Kolumne über das Leben in den USA, speziell das Leben mit Forth, ganz speziell das Leben mit Forth im Silicon Valley. Und Chuck Moores OK im Fernseher der Zukunft.

Intelhex-Format



von Bernd Beuster29

Manchmal muß eine Binärdatei durch einen Ascii-Draht. Mit Intelhex geben wir's dem EPROM so richtig.

Forth Online

von Olaf Stoyke 38

Wie kommt das Buch ins Netz? Und wieso sind Stackarchitekturen plötzlich so beliebt?

N'bißchen wat Praxis



von Wolfgang Führer31

Der Titel inspiriert sicher nicht nur das Satzsystem zu falschen Schlüssen. Es geht um richtig Handfest-grundlegendes. Und die längste Quelle des Jahrgangs!

Das Letzte - Freitexttabellenkalkulation



von Claus Vogt 38

Keine Angst, der Titel ist länger als der Inhalt.

3.502 2.05.96 20:27 bytedis\bytedis.tds
2.613 1.10.96 23:21 freitext\freitext.seq
2.461 28.08.96 18:19 intelhex\intelhex.f
16.847 8.02.96 23:37 pid\pid.seq
4 Dateien in VDL964.ZIP



= Listing auf Diskette



Dear Claus,

It is good to hear from you. ... SVFIG did have a booth at the Embedded Systems Conference. I was there for the first day only. ... I missed the September SVFIG meeting, but attended the October one, on the 26th. I must confess though, that most of the talks were "over my head" or out of my areas of interest, so I finished reading my latest copy of Vierte Dimension, the 3/1996 issue.

The "Group" still puts quite a magazine together. My favorite articles this time were by Olaf Stoyke, because they are oriented toward "Einsteiger", and I have been an "Einsteiger" for a looong time. And Martin Kalus is always fun to read. ... I am curious, aren't there any other Forthers on this side of the "pond" that read (or try to read) VD? I don't see any others in our SVFIG (Silicon Valley Forth Interest Group), but when I show my copy, there is some interest about possibly getting translations of your magazine into English.

Thanks for asking me to write more for VD. I shall try to do so,



even if it may be via e-mail to Friederich, and he can sift out the better parts for you. You have to know that I am still too much of an "Einsteiger" or "Anfaenger" (novice) to give qualified and correct reports on Forth matters, but I intend to keep on learning!

The next meeting, on November 16th will be our "Forth Day", which combines the SVFIG with the North Bay Group (Dr. Morgenstern's), so more presentations are being planned. I intend to spend the day there, and see what I report I can fly over the "Teich" afterwards.

Alles Gute!

Henry (v.vinerts@genie.com)

[An dieser Stelle möchten wir uns noch einmal ganz herzlich bei Henry bedanken. Er beginnt mit dieser Ausgabe eine Kolumne über die SVFIG im Silicon Valley und anderes forthige von der anderen Seite des "großen Teichs". / Die Redaktionsgruppe]

Leserforum: zu "Forth und der Anfänger" in VD 2/96, Seite 31

Den Bedarf an Newcomern gibt es schon lang, und in regelmäßiger Folge kann man auch hier darüber Gejammer lesen. Trotz verschiedener Maßnahmen konnte die Zahl der Neuzugänge nicht erhöht werden. Warum, wenn man doch mit Forth alles machen kann?

Geben Sie mal unter F-PC Words ** ein. Bei mir sind es so ca 4000 Worte, die angezeigt werden. Es scheinen sehr viele Worte nötig zu sein, damit man alles machen kann. Mal ehrlich gefragt: Kennt einer alle? Eben nicht.

Aber das sei nicht nötig, wird man entgegenen. Es gebe eine Onlinehilfe und Kommentare. In der Tat, diese Dinge gibt es. Aber wo? Wieviel der Worte stehen denn so im Hilfesystem drin? Wieviele der Worte sind denn einigermaßen gescheit dokumentiert? Fast alle Worte haben eine Stackbilanz, aber weitergehende Erklärungen sind bei mir praktisch nicht vorhanden. Den routinierten Forthanwender stört das nicht. Aber der Blick über den Tellerrand zeigt, daß es andere durchaus stört. Und das sind eben Anfänger, die eben noch keine Experten sind. Sollte es ein Zufall sein, daß FEP (Friedrich Prinz) mit einem sorgfältiger dokumentiertem System und einer gescheiten Dokumentation dazu mehr Anfängern helfen kann als der gesamte Rest der Gemeinde?

Ich habe schon vor langer Zeit eine Art von Anfängerhilfe angeregt. Geschehen ist nichts. Ich rege also nun zum zweiten mal an, folgendes zu tun:

Nachdokumentation des F-PC!

Wenn ein Forthexperte ein Wort pro Monat sorgfältig dokumentiert, dann braucht es 10 davon, um in einem Monat 300 Worte für Anfänger durchschaubar zu machen. Wenn das dann noch Worte sind, die so eine Art Starterset darstellen, wäre schon vielen Leuten geholfen. Diese Nachdokumentation kann in Papierform vorgenommen werden, aber es ist auch eine Installation in der Hypertexthilfe möglich. Und wenn es daran scheitern sollte, dann koordiniere ich das eben auch noch.

Weiterhin könnte es eine brauchbare Idee sein, ein Forth System auf Maschinencontrollerbasis zum Selbstbau über den Verein verfügbar zu machen. Platine und Eprom sollten zentral hergestellt werden, der Rest von gängigen Bauelementen nicht. Forth kann dann zumindest die Leute

Leserforum



Leserforum: Manöverkritik 3/96

Hi, there,

Wie die vorhergehende Ausgabe habe ich kaum was zu bemängeln, sehe aber auch keine drastischen Verbesserungen. Scheint wohl dein Normalniveau zu werden. Die Adressenseite ist jetzt von den offensichtlichsten Karteileichen gesäubert. Die Rubrik Buchbesprechungen fehlt. Die Ausgabe wirkt auch deshalb angenehm, weil überall eine lesbare Schriftgröße eingehalten wurde. Illustrationen, wie z.B. die Briefmarken, waren, wenn auch sparsam, vorhanden. Allerdings wirken solche kleinen Bilder auflockernder, wenn man sie in der Mitte der Seite plaziert wie auf Seite 8, statt am Rand wie auf Seite 36/37. Auf dem Treffen der Münchner Gruppe wurden Detailfehler im Layout bemängelt. Z.B. daß auf Seite 3 im Impressum Text und Rahmen ineinander übergehen. Letztlich spielen diese Kleinigkeiten keine Rolle.

Wichtig am Layout ist "Lesbarkeit". Die scheint jetzt zu 100% gegeben zu sein. Sowie ein gefälliges Aussehen. Das ist erst zu 75% erreicht, und wird erst besser, wenn die Autoren mehr gute Illustrationen liefern.

Die andere Seite der Qualität einer Zeitschrift macht der Inhalt der Artikel aus. Darauf hat der Redakteur nur wenig Einfluß. Meiner Einschätzung

erreichen, die einen Lötkolben richtig bedienen können.

Ich würde mich freuen, wenn ich nach diesen Vorschlägen nicht "nichts" als Reaktion feststellen muß.

So long, Robert Freitag, Oct'96
Formesstr. 31 a; 51063 Köln;
0221-617932

zu "Ein heiliger Krieg - das THEN im Forth" in VD 3/96, Seite 33

In der VD 3/96, gut wie immer, hat mir der "Heilige Krieg" von Michael Kalus besonders gut gefallen. Für denjenigen, der einfach nur "programmiert", ist es sicher von wenig Belang, ob IF-ELSE-THEN oder IF-THEN-ELSE oder sonstwas. In OCCAM2, zum Beispiel, hat ja IF ein wiederum ganz anderes Aussehen, das mit der natürlichen Sprachkonstruktion nicht die geringste Ähnlichkeit hat, ohne dass das bisher jemanden gestört hätte. Hat man aber außer Forth noch andere Interessen und denkt über "Forth und die Welt" nach, dann ist das Thema höchst interessant. Michael Kalus hat es mit seinem eigenen Beitrag dazu glänzend geschafft, meine Aufmerksamkeit auf diese Forum-Diskussion zu lenken.

Fred Behringer, Okt'96



Leserbriefe:

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

Forth-Magazin 'Vierte Dimension'
Claus Vogt, Katzbachstr. 23,
D-10965 Berlin, vd@FORTH-ev.de

Forth. Kurz und Knapp. Das kommt

nach ist die VD über all die Jahre hinweg hier nie über 25% vom Soll gekommen. Das einzige, was derzeit am Inhalt verdäulich ist, sind Kurzmeldungen, Leserbriefe und nichttechnische Artikel. Wie z.B. der Artikel zur "Echtzeit". Und das ist auch das, was gelesen wird.

Die "Fachartikel" belegen den meisten Platz im Heft, werden aber nicht gelesen. Ein Grund ist sicherlich die unattraktive Aufmachung ohne Illustrationen. Die sich allerdings die Autoren selbst zuzuschreiben haben. Ein kommentiertes Listing ist eben kein Artikel. Das andere Problem ist die enge inhaltlich Ausrichtung. Dargestellt wird meist:

- a) eine fertige Lösung zu
- b) einem einzigartigen Problem (nur der Autor hat es)
- c) auf einem bestimmten FORTH-System (nur wenige Leser haben es).

Besser wäre es:

- a) Algorithmen darzustellen. Mit Grafiken und notfalls Pseudocode. Der Leser kann sie sich dann passend implementieren. Die "fertige Lösung" paßt für ihn nie, weil er andere Hardware, andere Compiler, andere Anforderungen hat.
- b) Wenn schon nicht der Autor, dann sollte sich wenigstens der Redakteur die Frage stellen: Wieviele Leser könnte das Thema interessieren? Praktische Nützlichkeit und allgemeine Anwendbarkeit sind gute Kriterien für die Auswahl.
- c) Da es kein einheitliches FORTH gibt, wird wohl jedes Listing modifiziert werden müssen. Der Leser hat es leichter, wenn der Autor die Source mit Blick auf Portabilität schreibt und im Text in einen Abschnitt gezielt Inkompatibilitäten anspricht.

Rafael Deliano (jrd), Okt '96

WIN32FORTH zu langsam?

Die Reichhaltigkeit an Werkzeugen und Zusatzeinrichtungen bei WIN32FORTH ist beeindruckend. Trotzdem:

WORDS braucht zur Anzeige der 1829 Worte im Vokabular FORTH 5 Minuten und 21 Sekunden. Mein Transputer-Forth-System braucht für 1011 Worte 5,7 Sekunden. Das 16-Bit-Turbo-Forth-System braucht für 582 Worte 2,3 Sekunden.

Gibt es eine Möglichkeit, WIN32FOR von der DOS-Eingabezeile aus laufen zu lassen? Die einzigen ausführbaren Dateien, die vom DOS-Fenster aus zur Auswahl stehen, scheinen WIN32FOR.EXE und FKERNEL.EXE zu sein. WIN32FOR.EXE liefert die "normalen" Verhältnisse, bringt also nichts. Ein



Aufruf von FKERNEL.EXE bringt genauso wenig: WORDS steht dann noch nicht zur Verfügung. INCLUDE WORDS.F zieht einen Haufen von Meldungen über noch nicht definierte Worte nach sich. Aber

```
:XXX 5000 0 DO 7777 .LOOP ;
```

führt zum Ergebnis. Zum negativen Ergebnis: Die Zahlenausgabe auf dem Bildschirm ist genauso langwierig wie bei WORDS im "Normalsystem". Wo liegt der eigentliche Sinn? Was sehe ich falsch? Im Übrigen (neue Schreibweise) setze ich immer noch große Hoffnungen in WIN32FORTH. Und den Tom Zimmer bewundere ich. Was der an Arbeit investiert hat, kann ich inzwischen abschätzen: Die Entwicklung meines Transputer-Forth-Systems war gleichfalls Hobby in Extremform.

Fred Behringer, Okt'96

Forth-Gesellschaft: Forthtreffen im Storchennest von Kiel

Seit September treffen sich Forthler aus der Region im Storchennest von Kiel. Das Lokal liegt im Uni-Bereich und dürfte dort leicht zu finden sein: Westring - Gutenbergstrasse. Treffen jeden 4. Mittwoch im Monat.

Kontakt über Michael Kalus
(michael@malente.forth-ev.de)
Okt'96

Termine: RTS'97 in Paris

RTS'97
Palais des Congrès
Paris, 15.-17. Januar 1997

Echtzeitsystem-Kongress. ((R)eal (T)ime (S)ystems?) "Call for papers" bis Ende September 1996. Preis für den besten Beitrag.

Kontakt: BIRP-RTS'97, 17, avenue Ledru Rollin, 75012 Paris, Frankreich. Tel.: 33 (1) 53 17 11 40; Fax: 33 (1) 53 17 11 45. beh, Sep'96

Termine: POPL'97 - Call for Papers

This was posted by kamin@cs.uiuc.edu (Sam Kamin) to a bunch of other newsgroups: comp.lang.c++, comp.lang.functional, comp.lang.lisp, and comp.lang.misc. There *must* be Forth-related papers that would fit in here...

Call For Papers
Workshop on
Domain-Specific Languages
Paris, January 18, 1997
(in association with POPL '97)

Domain-specific languages (DSL's) are those languages designed to accomplish tasks within a specialized domain when general-purpose languages do not support the necessary domain abstractions, notations, analyses, and optimizations. There has been increasing interest in these languages recently, for a variety of reasons, including the growing importance of "end-user programming" and encouraging results from using "project-specific" languages to increase code re-use. At the same time, programming language researchers are coming to see these languages as a great potential source of ideas and applications.

The most striking aspect of existing DSL's is their enormous variety, which poses a number of scientific

questions: Are differences among DSL's deep or superficial? Do the needs of end-users and casual programmers call for fundamentally new language design ideas, or should principles of design of general-purpose languages be applied to DSL's? Are the available tools for implementing computer languages well suited to the needs of DSL developers?

The purpose of this workshop is to bring together programming language researchers and DSL designers to discuss these topics. Papers on all aspects of domain-specific languages are solicited, including, but not limited to: descriptions of particular DSL's and experience in their use; language design concepts suited particularly to some problem domain; tools for designing and implementing DSL's; applications of DSL's throughout the software life cycle.

Important Dates

Submission deadline:
October 14, 1996
Acceptance decisions:
November 15, 1996
Final papers due:
December 13, 1996
POPL: January 15-17, 1997
DSL Workshop:
January 18, 1997

Important Addresses:

Sam Kamin (Program Committee chair):
s-kamin@uiuc.edu
Workshop web page:
[http://www-sal.cs.uiuc.edu/~kamin/dsl/POPL web pages](http://www-sal.cs.uiuc.edu/~kamin/dsl/POPL_web_pages)
(mirrored): <http://www.cs.umd.edu/~pugh/pop197/>
<http://www.diku.dk/pop197/>
clv nach comp/lang/forth Sep'96

Forth Online: Neue Adressen

Einige recht bekannte Internet-Adressen sind seit Mitte September im Sinne übersichtlicherer Namen geändert worden: Die folgenden Namen sind von nun an verbindlich:

- ftp.taygeta.com ist die Adresse zum FTP-Archiv von Taygeta
- <http://www.taygeta.com> ist die Adresse der Taygeta-Homepage

Die "Forth Interest Group (FIG)" ist unter <http://www.forth.org> zu finden.

Als Konsequenz dieser Neuordnung entfällt die Adresse taygeta.com sowohl für den FTP- als auch für den Web-Zugriff.

Olaf Stoyke (os@cs.tu-berlin.de)
Okt'96

Forth inside:

Europay

Sprechen vielleicht in ein paar Jahren alle Geldautomaten Forth? Auf der Rochester Forth Conference 1996 in Toronto präsentierten Elizabeth Rather, Jon Lee, Stephen Pelc und Peter Johannes die Arbeit an Europay (Forth Dimensions Sep/Oct'96, Seite 33).



In `comp.lang.forth` wird Stephen Pelc etwas konkreter:

"The MPE SENDIT system will be commercially available by the year end, and the Europay system developed by MPE with Forth Inc will be available for payment terminal use from Europay, MPE, and Forth Inc. ... There are both Forth and C to token compilers for this application, and interpreters written in both Forth and C. For more details get the papers, and the specifications are available to payment terminal vendors from Europay. The original SENDIT specification is available from MPE.

Potentially 7.5 million [terminals will be installed] in Europe over the next few years. The software was successfully demonstrated to Europay's member banks in Seville, Spain the first week of June. Eight different models of terminals were running (6 Point-of-Sale terminals with CPUs ranging from 8051s to 68000's, plus 6303-based vending machines and Windows-based banking terminals). Over 4,000 real transactions were processed in a 'model town' with real shops, restaurants, etc."

Papers were presented on this technology, called the "Open Terminal Architecture" at the Toronto/Rochester conference, by Europay, FORTH Inc. and MPE.

Why was Forth chosen: compactness (matters in the terminals, and even more for applications being transmitted over phone lines), flexibility, adaptability to token architectures."

clv nach comp/lang/forth Sep'96

Forth-System

Holon4th 4.0 released

Gegenüber der letzten Holonversion sind zahlreiche kleine Verbesserungen erfolgt. Dies betrifft den

Object-Type-Mechanismus, eine stringere Benennung einiger Worte, die neu eingeführten Arrays und eine Vereinfachung der Metaworte.

Quelltexte können jetzt auch als Textdateien (*.MTX) exportiert und importiert werden. Weitere Änderungen erleichtern die Kommunikation zwischen Host und Target.

Holon4th ist eine voll funktionsfähige Freeware Version der integrierten Entwicklungsumgebung HolonForth. Einzige Einschränkung: Die Applikationen sind auf 1000 Wörter begrenzt.

Holon4th.zip ist in der Kieler Box verfügbar, und in `ftp.taygeta.com/pub/forth/applications/holon4th.zip`

Die neue Profiversion 'Holon-86' wird demnächst erscheinen.

Kontakt und weitere Informationen:

*Wolf Wejgaard, Forth Engineering, CH-6045 Meggen
Fax: +41-41-377-4774;
wejgaard@access.ch
(clv)*

Forthsysteme:

win32for 3.3 released

Win32Forth version 3.3 has been released on you the unsuspecting public. You can download it from `taygeta` when skip moves it into his compilers page, or if you want it in a hurry, then it can be obtained from his `/incoming/forth` directory at [korrigiert nach zweiter News/clv]:

- `ftp://ftp.taygeta.com/incoming/forth/w32for33.exe`
- `ftp://ftp.taygeta.com/incoming/forth/w32for33.new`
- `ftp://ftp.taygeta.com/incoming/forth/w32for33.faq`
- `ftp://ftp.taygeta.com/incoming/forth/w32for33.txt`

This version includes some interesting new features as follows;

- Multi-Tasking: Contributed by Bill Muench from EFORTH
- Adjustable Dictionary: Easy dictionary resize from File menu
- Windows Constants: Removed 2000 windows constants out of Win32Forth, into WINCON.DLL. Donated by Andrew Mckewan. This increases the number of available Windows constants to over 8000 without burdening the dictionary.
- Improved Font Support: Added a Font class, to make font manipulation easier and more flexible.
- Better ANS compatibility: I claimed that Win32Forth was

ANS compatible, and I still claim that, but now it is better. I fixed a few small things related to upper casing and case insensitivity.

- Added Documentation: Added documentation to several parts of the source code and documentation files.

You can find a detailed report of the ongoing changes to Win32Forth in the file `Win32For.NEW` after it is installed.

As is the usual custom, now that I have released it, someone will most likely find a serious bug, but until then this is it. I hope some of you will find it interesting.

*Tom Zimmer (tzimmer@austin.finnigan.com), comp.lang.forth Okt'96
[PS: Die Dateien sind auch in der KBBS/mka]*

Forthsysteme:

win32for 3.3 workaround

I have discovered a minor but irritating bug in WinView. The Printer button in the tool bar will cause WinView to crash when it is used the second time. ... The simplest fix is to change one word in the following program segment from the file `WINVIEW.F` to use `PRINT-TEXT` instead of `AUTO-PRINT-TEXT`:

10 HSpace

3 PictureButton auto-print-text ;
ButtonInfo" Print "

10 HSpace

Tom Zimmer (tzimmer@austin.finnigan.com), comp.lang.forth, okt'96

Forthsysteme:

ANS macht Dynoof-0.1.5

Dynoof-0.1.5 ist auf `Taygeta` unter `pub/Forth/Linux/dynoof-0.1.5.tgz` zu finden. Und in der Box der Forthgesellschaft in Kiel auf Wunsch ebenso. Wendet Euch bitte an Holger Petersen. Es heißt nun nicht mehr "doof", denn das war wohl zu doof auf die Dauer. Nun heißt es dynamic object oriented forth, eben Dynoof. Bis auf einen bug fix und `<()>` als Alias von `{ }` ist es dasselbe geblieben. Damit geschweifte Klammern auch für Matrizen statt Objekte genommen werden können :-)

Einen ANS Forth Emulator für das (Dny)OOF gibts nun ebenso. Dieser ist gedacht für alle, die nicht das ganze Ding benutzen wollen, jedoch an OOP Interesse haben. (`taygeta:pub/incoming/forth/oofemu.4th`)

Michael Kalus (michael@malente.FORTH-eV.de), Okt'96

Forthsysteme:

RISC OS Forthmacs letter, September 1996

RISC OS Forthmacs 3.1/2.20 hat jetzt ANS Floating und das Floating Extended wordset.

HTML Documentation mit StrongED4 Forthmode Unterstützung. Eine automatische Suche nach allen Informationen zu dem Wort unter des Editors Cursor ist dabei, die ein HTML-Dokument liefert, wenn ein Browser zur Verfügung steht, sonst einen ASCII-Text.

CATCH und THROW sind implementiert, alle ABORT" Meldungen gebrauchen es zweisprachig: deutsch und englisch.

Die RISC OS Forthmacs Implementaton der Sprache Forth entwickelt sich weiter. Sie bietet (und nun weiter 'in english' da teuflisch zu übersetzen wegen der termini technici-teritis):

- 32-bit stacks and addresses
- structured source-code-ready decompilation
- macro assembler and disassembler
- Source-code-level debugging/tracing/single-stepping
- Assembly language debugging/tracing/single-stepping
- Command line editing with interactive history editor
- command completion
- callfinder
- Operating system independent file system interface
- operating system interface compiler
- strings
- crash resistance (also in multitasking)
- run-time search order specification
- fast and effective cooperative multitasking
- runtime profiling tool
- on-line documentation as ASCII and HTML
- complete source code (optional) with target/metacompiler
- xmodem and serial interface tool
- Floating point library

Die aktuelle Version des RISC OS Forthmacs ist hier zu finden: `ftp://ftp.uni-stuttgart.de/pub/systems/a-com/riscos/lang/forth/forthmacs`

Zuerst sollte die WorkDisc (self extracting dicottery archive) Application bestellt werden. Die anderen Toolarchive werden dann wie in der `docs.ascii.convention` beschrieben in das `risc_os.installing` Verzeichnis gelegt.

Den richtigen StrongED4 - ältere Versionen hatten Fehler - bekommt man bei: <http://login.eunet.no/~guttorvi/strong.html>

Hanno Schwalm (schwalm@bre.winnet.de) RISC OS Forthmacs Support, Sep'96

**Forthsysteme:
F6 für Sinclair QL**

Für alle Freunde des Sinclair QL liegt in Taygeta ein erweitertes System nach Art des f.i.g.- Forth. Das F6 System ist für QDOS/SMSQ OS's der 680xx Prozessoren und im Kern äußerst angepaßt an die Wirtsmaschine. Etliche Eigenschaften des Systems mögen darüber hinaus von allgemeinem Interesse sein. Interessenten sehen nach bei:

<ftp://taygeta.com/pub/Forth/Archive/68000/GF6...>

mka nach phpr@berlin.snafu.de in /comp/lang/forth Sep'96

**Forth Online:
Überall Forth drin!**

Unter <http://www.forth.com/AppNotes.html> findet sich neuerdings eine Liste von Beschreibungen zu Projekten, die mit FORTH realisiert wurden. Die Liste soll weiterhin wachsen und enthält derzeit folgende Projekte:

- Einen Simulator für den Roboterarm des Shuttles der NASA
- Eine Schnittstelle zu Shuttle-Navigationsinstrumenten
- Kontrollsoftware in einem Solarkraftwerk in Sacramento, Kalif.
- Eine Anwendung in FORTH implementierter neuronaler Netze
- Eine Anwendung zur präzisen Dosierung von Nahrungszusätzen
- Software für die mobilen Strichcodeleser von Federal Express

Olaf Stoyke (os@cs.tu-berlin.de) Okt'96

**Forth Online:
Weltraum**

Nicht nur die Trekkies unter den FORTHlern sind möglicherweise an den technologischen Entwicklungen im Bereich bemannter und unbemannter Raumfahrt interessiert. Wer beiden Interessen parallel nachgehen will, findet auf <http://groucho.gsfc.nasa.gov/forth/> eine Liste von Firmen oder Institutionen wie Universitäten, die mit FORTH Weltraumtechnologie implementieren. Die Liste selbst enthält je eine (sehr) kurze Beschreibung des Projekts und die Adresse

eines Ansprechpartners. Die Autoren der Liste haben leider wenige der eingebauten Verweise für den Web-Browser zugänglich gemacht, so daß mensch sich auf eine ausgedehnte Übung seiner Copy- und Paste-Fähigkeiten freuen darf.

Olaf Stoyke (os@cs.tu-berlin.de) nach comp.lang.forth Sep'96



**Forth Online:
FAQs in comp.lang.forth**

Im Juli erschienen die FAQs zu Forth aus dem Internet auch in comp.lang.forth. Sie wurden von J.D. Verne dort ins Nachrichtenbrett gehängt. Zwar kam es dabei zu einer Panne, indem gleich alles zweimal zu lesen war, was nicht alle auch doppelt erfreute. Schließlich kostet sowas Übertragungsgebühren. Doch wer über den Forth-ev inzwischen auch in der Mailbox zu Hause ist und auch die Forth Nachrichten verfolgt, konnte sehen, was sich da tut im Forthmarkt. Die ganze Nachricht ist zu groß, um hier abgedruckt zu werden. Dazu schaut man eben besser mal in der Mailbox nach. Aber was in den FAQs so zu finden ist sei doch hier einmal wiedergegeben.

Table of Contents:

1. Acknowledgements
2. comp.lang.forth FAQs
3. General Questions
 - 3.1. What is Forth?
 - 3.2. Why and where is Forth used?
 - 3.3. What language standards exist for Forth?
- 3.4. What is the Forth Interest Group?
4. Miscellaneous
 - 4.1. Where can I find a C-to-Forth compiler?

4.2. Where can I find a Forth-to-C compiler?

4.3. RECORDS in Forth?

1. Acknowledgements

This FAQ is based on previous work by Gregory Haverkamp, J. D. Verne, and Bradford J. Rodriguez.

2. comp.lang.forth FAQs

The comp.lang.forth FAQ is published in six parts, corresponding to these six sections. This part is the General/Misc FAQ, where the questions not covered in the other FAQs are answered. The six parts are:

- o General questions <<http://www.complang.tuwien.ac.at/forth/faq/faq-general.html>>
- o Online resources <<ftp://forth.org/pub/Forth/FAQ/online>>
- o Forth vendors <<ftp://forth.org/pub/Forth/FAQ/vendors>>
- o Forth systems <<ftp://forth.org/pub/Forth/FAQ/systems>>
- o Books, periodicals, tutorials <<ftp://forth.org/pub/Forth/FAQ/books>>
- o Forth groups & organizations <<ftp://forth.org/pub/Forth/FAQ/groups>>

These FAQs are intended to be a brief overview of the tools and information available for the new FORTHer. For a historical reference, programming paradigms, and deep technical information try some of the listed references. For general questions on the internet, or the methods used to get this information, try these other Usenet groups:

- o news.announce.newusers
- o news.newusers.questions
- o news.announce.important

(...so, und hier kamen nun die ganzen Kilobytes, aber das seht ihr ja dann selber in der Mailbox, gell? :-)

Und nur mal so um zu zeigen wie der "große Bruder" sich inzwischen in den Netzen so präsentiert - hier noch das Signum der Nachricht.

=====
current FIG address/contact info:
=====

Forth Interest Group
P.O. Box 2154
Oakland, California 94621
telephone: 510-893-6784
fax: 510-535-1295
e-mail: office@forth.org
Forth Interest Group home page:
<http://www.forth.org/fig.html>

Also mit Homepage wie sich das gehört.

Was kostet diese Weltenbummelei in den Netzen uns im Forth-ev? Nicht

viel wie ich meine. Gerade 1x Jahresbeitrag + Telekom bis zur Box in Kiel. Ein Programm in deutsch zum Erstellen, Versenden und Empfangen der E-Mail gibts gratis dazu. Damit werden auch die Nachrichtenbretter und Files bestellt und verwaltet. Der eigentliche Nachrichtenaustausch zwischen eurem Rechner und der Box geschieht beim Anruf vollautomatisch in kürzester Zeit, weil gepackte Files übertragen werden. Auspacken und Lesen geschieht dann anschließend "Offline" in aller Ruhe ohne Zahlungen an die Telekom. Ich brauche für so 2-3 Briefe und 20-30 Nachrichten pro Tag einen einzigen Anruf bei der Box und der Austausch ist in ca 2 (zwei) Minuten in aller Regel über die Bühne (14400 Modem). Briefe und News sind ja oft kurz, so um 1000 Bytes. So kostet ein Brief in die USA wohl nicht mal 30 Pfennige, ist noch am gleichen Tag da und die Antwort habe ich bereits am anderen Morgen. 2x täglich tauscht sich die Box mit dem Rest der Welt aus. Die eigene E-mailadresse darin ist vom Internet aus zu erreichen, man hat also selbst eine Internetadresse damit erworben. Nur Internet online geht nicht, ist aber aus Kostengründen in der BRD auch keinem zu empfehlen.

Also, langer Rede kurzer Sinn: Forthmitglieder, willkommen im Netz. Jetzt steht die Sache wirklich.

Dem Fritz Prinz danke ich für die Anregung zu der ganzen Sache.

Viele Grüße aus Malente, Euer Michael.

mka (Michael Kalus), michael@malente.forth-ev.de, Jui'96

**Bücher:
Vack "Programmieren mit Forth"**

Ich habe hier ca. 70 Exemplare Vack: Programmieren mit FORTH, 1. Auflage 1990, ISBN 3-341-00518-8, Verlag Technik Berlin, abzugeben zum Verlagspreis von 42 DM + 4 DM Versandkosten (Inland). Bitte einen Verrechnungsscheck über *46,00 DM* an meine Adresse. Absender nicht vergessen!

Auf Wunsch lege ich Disketten zum Preis von 2 DM pro Stück bei:

- F83 2.1 (PD) - für Nostalgiker
- F-PC 3.6 (PD) - sehr komfortables DOS-System mit allen Quellen + ANSI
- HOLON Lite 3.2 (PD) - für die Target-Entwicklung am laufenden Programm
- WinForth 1.01 (Shareware) - vom Laboratory Microsystems Inc.

- Win32Forth 3.2 (PD) - "Nachfolger" vom F-PC, in Entwicklung

Der Versandkostenanteil erhöht sich in diesem Fall auf 5,50 DM. Die volle Packung gibt es also für 57,50 DM.

Das Buch enthält 336 Seiten, 85 aussagekräftige Bilder, Glossarium von FIG-FORTH und FORTH-83. Das Bezugssystem für dieses Buch war FIG-FORTH, während sich FORTH-83 gerade durchzusetzen begann. Der Autor widmet deshalb einen Teil des Buches dem Vergleich und den Unterschieden der verschiedenen Standards. Konzepte und Interna von FORTH werden sehr kompetent erklärt. 38 schöne Programmbeispiele inklusive dem Quelltext einiger spezieller Programmierverfahren runden den guten Eindruck ab. Ein Buch, das im Bücherregal keines FORTHlers fehlen sollte.

PS: Ich würde auch ein oder zwei Exemplare gegen Zech: Forth 83 tauschen.

Das Buch ist auch für F-PC geeignet, denn F-PC ist FORTH-83 kompatibel. Zitat aus dem Vorwort des Buches: "Im Unterschied zu anderen Programmiersprachen, in denen man häufig das Problem der Sprache anpassen muß, paßt man in FORTH die Software dem Problem an." Im Grundwortschatz ist jedes FORTH gleich, und wenn nicht, wird es gleich gemacht. Zu F-PC gibt es inzwischen die ANSI-Erweiterung und wie sich zeigt, hat sich gegenüber FORTH-83 nicht so viel geändert. Das CORE Wordset besteht überwiegend aus gleichnamigen Aliasdefinitionen des FORTH-83. Hinweis: Vorsicht ist dann geboten, wenn spezielle Programmierverfahren (wie etwa DOER/MAKE) auf ein 32-Bit-System übertragen werden sollen, da sich die Adreßrechnung ändert (z.B. werden dann mit ! und @ statt 2 Byte 4 Byte geschrieben bzw. gelesen).

Helfried Schürer, Berliner Str. 105, 13189 Berlin (HSc@arson.forth-ev.de) Okt'96.

Presseschau: FLEX TA64x30

Die Produktbesprechung in "Byte" 9/96 ist etwas irreführend. Denn nicht "Telelink designed the FLEX TA64x30 ..." sondern es handelt sich um ein gemeinsames Produkt von JW Datentechnik (Systemsoftware), ikon (ISDN-Hardware/Firmware) und Telelink (Vertrieb). Wobei JW-Datentechnik bei der Entwicklung federführend ist. Bei dem Router handelt sich um eine VME Bus Rack

in der ein ISDN-Primärdaten-Adapter, Modemleiterplatten für bis zu 30 analoge Modems und ein embedded PC als Steuerrechner stecken. Der PC läuft unter Linux, die Programmierung erfolgt in Bernd Paysans big-Forth. Zu den Aufgaben der Software gehört u.a. die Realisierung von Firewalls, und diverse Protokollverarbeitung (TCP/IP, PPP).

Rafael Deliano (jrd), Okt'96

Bücher: Lieferbare deutsche Bücher

Man findet in BTX beim ABC-Bücherdienst zum Schlagwort FORTH bzw Forth:

- Leo Brodie "In FORTH denken" 1986 Hanser 54,00 DM. ISBN 3-446-14334-3
- Leo Brodie "Programmieren in FORTH" 1984 Hanser 49,00 DM. ISBN 3-446-14070-0
- Ekkehard Floegel "FORTH Anwendungsbeispiele" 1984 Hofacker 49,00 DM. ISDN 3-88963-200-9
- Ekkehard Floegel "FORTH Handbuch" 1982 Hofacker 49,00 DM. ISDN 3-921682-88-6
- Ekkehard Floegel "FORTH Schnellkurs" 1989 Hofacker 9,80 DM. ISDN 3-88963-264-5
- Konrad Koller "FORTH und FORTH-Prozessoren" 1993 Expert Verlag 34,00 DM. ISDN 3-8169-0785-7

Die beiden Bände von Brodie können vorbehaltlos empfohlen werden. "Programmieren in FORTH" ist der klassische Text für Leute die FORTH programmieren lernen wollen. "In FORTH denken" ist die Weiterführung dazu. Das Buch von Koller wendet sich weniger direkt an Programmierer, sondern soll industriellen Anwendern einen Überblick über das Phänomen FORTH geben. Die Hofacker-Bücher sind aus einer vergangenen Epoche übriggeblieben.

Rafael Deliano (jrd), Okt'96

Presseschau M&T: Compilerübersicht

In der Markt & Technik 5/96 war eine Übersicht über Compiler speziell auch für embedded Anwendungen aufgeführt. Es fanden sich auch drei Anbieter für FORTH.

Diessner Datentechnik hat einen Compiler für H8. Allerdings ist aus anderer Quelle zu erfahren, daß der nicht sonderlich lieferfähig ist.

FORTecH hat comFORTH für MS-DOS und Windows ab 400 DM.

Aber auch Crosscompiler für eine inzwischen beachtliche Liste von Targets zum Preis von 2200 DM. Die weiterentwickelten fieldFORTH-Crosscompiler kosten 4000 DM.

Noch teurer ist Swiss-Forth von Mikrap für 7500 DM mit einer ähnlich langen Liste von verfügbaren Targets. Der teuerste Compiler in der Liste war ein Prolog für 12k DM. Alle Preispositionen von 200 DM bis 10k DM waren aber erstaunlich gleichmäßig besetzt. Mit besonderen Schwerpunkten bei 400 DM und 2000 DM.

Rafael Deliano (jrd), Nov'96

Forth und Musik: fLux

Jüngst fragte ich Frank "Genius" Rothkamm: "Und sonst? Gibt es neue Gerüchte aus der Forth-Musik-Szene? Grade mache ich die Jahres-End-VD und eine Kurzmeldung könnte ich grade noch brauchen." Hier seine Antwort:]

Dieses Jahr hab icke NUR Auftragsarbeiten für andere geschrieben. Remixes, Sound tracks, Productions. Von Trip-Pop bis Hard-edge noise. (George Lucas, Raymond Chen, Elliott Sharp, Zeena Parkins, Alfred Harth, Wolfgang Muthspiel, DJ Glove). Ein paar Algorithmen eingesetzt. (meine Website hat den 2ask "Best of the Planet" Preis bekommen, was auch immer das sein soll - [http://www.thing.net/~flux.](http://www.thing.net/~flux))

David Tudor ist gestorben und Matt Rogalsky (<http://lo-cal.music.wesleyan.edu:8000/device.html>) ist aus England hier nach NYC gekommen, um den Nachlass zu verwalten und hat ein Tonband meiner Musik (die isolationism/mystery-of-the-leaping-fish Sachen) dort gefunden. Hat mich dann kontaktiert und war hier, hab ihm mein Forth/Formula System gezeigt, hat ihm gut gefallen, da kannst du so Wahrscheinlichkeitsprozesse der Veränderung von FM-Synth-klängen auf Sliders legen. Er hat einen Forth/Formulatrack auf der Songlines/Anthology CD, sehr Cage mäßig, ruhig, gutes Algo-Stück.

Paul Modler <100550.3017@CompuServe.COM> hat immer noch weiter auch an einer Übertragung des Berliner C.A.M.P. Atari Systems (von dem mein NYC System nur eine Weiterentwicklung darstellt) auf Mac's gearbeitet. Er macht auch gute Live-Sachen und hat die CD mit seiner Band raus. Hatte in Berlin nicht mehr die Zeit ihn zu besuchen, weil ich nach Frankfurt für die Harth/Sharp productions ins Studio mußte.

Werde, wenn ich Ende December nach den 4 Corners gehe (Utah, Colorado, New Mexico, Arizona -- wir machen da ein paar UFO photo-sessions), mal ein wenig ausspannen und den versprochenen Formula NYC Artikel andenken.

Die "Planet Genius" Platte wurde übrigens als "Planet Genius is to techno what John Cage was to Classical" hier neulich besprochen, und ist wieder in der Hitparade (Platz 18) aufgetaucht in A.P. (Was auch immer das alles aussagen soll.)

Wir bedanken uns und freuen uns auf den Artikel. (clv) Nov'96

Stackprozessor als FPGA

von Rafael Deliano
Steinbergstr. 37; D-82110 Germering

FPGAs, Field Programmable Gate Arrays, sind komplexe

Logikschaltungen die der Anwender genau wie ein EPROM selbst programmieren kann. Was mal mit PALs und GALs angefangen hat, erreicht inzwischen tatsächlich die Komplexität von GateArrays.

Stichworte: FPGA Stackprozessor Projekt

Der NC4000 von Charles Moore wurde bekanntlich als GateArray gefertigt und benötigte etwa 4000 Gates. Da die Gates in FPGAs weniger flexibel verwendbar sind als in GateArrays, wird man hier wahrscheinlich 10k Gates verbrauchen. Die Obergrenze von FPGAs liegt heute bei ca. 100k Gates, geeignete Typen sind also schon verfügbar.

Im Süden der Republik regt sich deshalb das Interesse an einem Stackprozessor als FPGA. Nicht als technische Spielerei, sondern als kommerzielles Produkt. Die folgenden Details stammen von mir und sind noch spekulativer Natur. Aber die Randbedingungen deuten auf eine Entwicklung in diese Richtung.

Vorteile gegenüber Controller

Bisher wurde bei der Entwicklung einer Applikation das Problem solange verbogen, bis es zur lieferbaren Hardware paßte. Im FPGA paßt man sich die Hardware ans Problem an. In handelsüblichen Controllern wird die I/O (Timer, Ports, UARTs usw.) als eierlegendes Wollmilchschwein ausgelegt. Sie ist komplex, kann mit vielen Registern eingestellt werden und stellt doch niemanden wirklich zufrieden. Im FPGA baut man sich aus sehr simplen Zellen nur genau das auf, was man braucht. Übliche Zellen kann man aus einer Library entnehmen. Exotische Schaltung kann man selber bauen. Bis aufs letzte Gate genau so, wie man es für richtig hält.

Auch von der CPU werden wohl 2 - 3 nach MIPS gestufte Varianten zur Auswahl stehen. Versionen, die aus weniger Gates bestehen, haben geringere Rechenleistung. Der FORTH-Befehlsatz für den Anwender kann über alle Varianten identisch sein, aber Opcodes, die in Software emuliert werden, bremsen natürlich den Durchsatz. Feintuning der CPU ist durch die Wahl zusätzlicher Opcodes (z.B. Fuzzy-Opcodes) möglich. Genauso kann man natürlich eigene Opcodes erzeugen, die direkt mit der I/O zusammenarbeiten.

Alles bisher Gesagte ist weniger ein Phantasiegebilde als es den Anschein hat. Es wurde von Delta t schonmal so gemacht. Der IX1 ist ein GateArray. Er hat relativ wenig Gates in der I/O verbaut und eigene Feldbusbefehle die eng mit der I/O zusammenarbeiten. Der IX1 löst jedoch nur die Feldbusapplikation. Das FPGA knackt beliebige Nüsse.

Es gibt noch weitere Vorteile. Ein Maschinenbauerhersteller der nach 10 Jahren noch Ersatzteile liefern will, dürfte es schwierig finden die (NMOS)-CPUs des Jahres 1986 heute noch im Handel zu finden. Ein kompatibles FPGA wird 2006 noch verfügbar sein.

SecondSource-Probleme und Motorola-Lieferzeiten entfallen, wenn FPGAs sich immer weiter durchsetzen.

Probleme

Nachteile gibt es natürlich auch. Erstens ist der Stückpreis eines komplexen FPGAs recht hoch (50 - 250DM). Wenn man die Applikationsschaltung jedoch nur in geringen Stückzahlen fertigt, beerdigen die hohen Entwicklungskosten meist ohnehin alle Materialkosten. Die Kosten des FPGA sind dann nicht so kritisch. Geht die Schaltung in hohe Stückzahlen über, kann man das FPGA nach dem Serienanlauf immer noch in ein GateArray konvertieren. GateArrays sind keine Konkurrenz, sondern eine Ergänzung. Sie haben Maskenkosten (>50kDM) und eine Mindestabnahmemenge (>5k Stück). Sowie eine Anlaufzeit von Monaten bis der Hersteller sie liefern kann. Sie sind jedoch dann wesentlich billiger als das FPGA.

Die wesentliche technische Einschränkung gegenüber einem GateArray ist, daß die Integrationsdichte nicht so hoch ist, daß man Speicher auf dem Chip unterbringen kann. Das rekordverdächtige 100k Array von Altera kann etwa 4k Byte RAM enthalten. Ist jedoch dementsprechend teuer. Man ist also auf den Entwicklungsstand des NC4000 und FRP1600 zurückgeworfen. Um alte Fehler nicht zu wiederholen, verzichtet man besser auf allzu viele parallele Busse und die dazu nötigen teureren PGA-Gehäuse. Für den Anwender zumutbar sind die Bauformen PLCC-68 und 84. In Bild 1 ist eine typische Beschaltung gezeigt. Adreß- und Datenbus werden gemultiplext. Das ist zwar nicht sehr schön, aber der 8051 tut das schon seit 15 Jahren. Der Datenbus ist 16

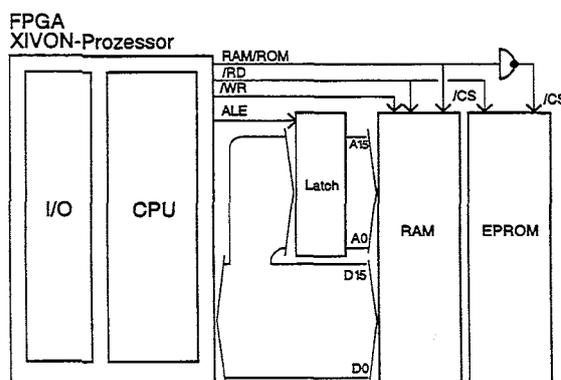


Bild 1: Schaltung

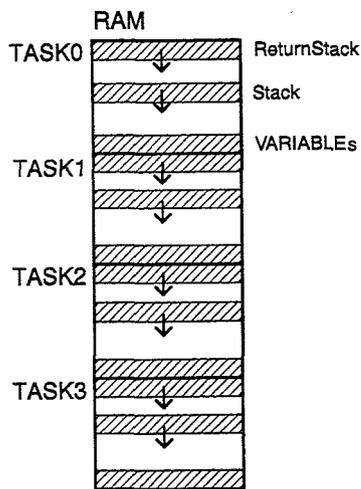


Bild 2: Speicherbelegung

Bit breit, der Adreßraum jedoch auf 64k Word beschränkt. Das genügt für einen Controller.

Ein externer Speicher hat auch Vorteile. Man kann im Gegensatz zum SingleChip-System billig viel RAM und EPROM bereitstellen. Redundante Codierung der Opcodes ist erwünscht, um die CPU zu vereinfachen. Niedrige Codedichte schlägt sich bei externem ROM bestenfalls auf den Durchsatz, nicht aber auf den Preis nieder. Soweit man das Datenwort breit genug wählt, hier 16 Bit, leidet auch der Durchsatz nicht allzusehr.

Multitasking

Hat man genügend RAM, kann man wie in Bild 2 gezeigt, einen multitaskenden Stackprozessor mit 4 oder 8 Tasks aufbauen. Dabei hat jede Task ihre eigenen Stacks. Das beschleunigt den Taskwechsel. Man kann durch simple RoundRobin-Umschaltung eine höhere Granularität der Zeitscheiben erreichen, als das mit konventionellen CPUs möglich ist. Diese herausragende Multitaskingfähigkeit dürfte das entscheidende Feature eines Stackprozessors mit externem RAM sein.

In einem ABS-System z.B. würde jede der 4 Tasks ein Rad überwachen. Es ist im entscheidenden Moment wichtig, daß alle 4 Räder gemeinsam richtig reagieren. Verwendet man einen konventionellen 16 Bit Controller mit langsamem Taskwechsel, kann man nicht jedes Rad dauernd so genau im Auge behalten, daß man Veränderungen sofort erkennt. Verknüpft man vier 8 Bit Controller über einen seriellen Bus, hat man im entscheidenden Augenblick Probleme mit der Kommunikation zwischen den Tasks. Der Stackprozessor löst harte Echtzeitanforderungen.

Technologie

Die Programmier-elemente in FPGAs können RAM, EPROM, EEPROM und Antifuse, eine neumodische Art von PROM, sein. Typen mit RAMs booten nach dem Reset aus einem seriellen EEPROM. Durch diesen langsamen Start scheiden sie für die meisten Anwendungen aus. OTP-EPROMs und Antifuse kann man nur einmal programmieren. Dann ist der teure Chip erledigt. Da man seine selbstgestrickte I/O in der Zielhardware austesten muß, ist mehrfache Programmierbarkeit erwünscht. FPGA-Varianten aus EEPROMs und Flash-EPROMs dürften für den Anwender am vorteilhaftesten sein.

Am liebsten würde man natürlich die I/O-Library und die Modelle der Stackprozessoren herstellerunabhängig in VHDL definieren. Ein Modell des Z80, das konventionell 6k-10k Gates benötigt, wächst in VHDL jedoch auf 20k - 25k Gates. Real muß

man den Stackprozessor also auf unterster Ebene direkt auf einen bestimmten FPGA-Typ ausgelegt zusammenbasteln.

Neben der mangelnden Herstellerunabhängigkeit, sind derzeit die hohen Kosten von 2k-10k DM für die Entwicklungssoftware des FPGAs ein ernstes Problem. Jeder Endanwender benötigt diese Software um seine CPU zusammenzusetzen. Da immer mehr Hersteller komplexer FPGAs auf den Markt drängen, wird der Preis für die Software jedoch in Bewegung kommen. Letztlich werden Tools, die auf Bausteine bestimmter Hersteller beschränkt sind, kostenlos verfügbar.

An Software für den Stackprozessor ist natürlich außer FORTH noch ein C-Compiler fällig, wenn man am Markt ernstgenommen werden will. Ein interessanter Gag dürfte die Unterstützung von Java-Bytecode sein. Die praktische Bedeutung von Java für embedded Controller wird jedoch gering bleiben.

Anwendungen

Während man mit technischen Features spielt, sollte man immer die Frage im Kopf behalten: wo ist der Markt dafür? Das System muß mit konventionellen Lösungen konkurrieren können. Dieser Stackprozessor kann von der Rechenleistung her nicht mit 16 und 32 Bit RISCs konkurrieren. Und vom Preis nicht mit 8 Bit Controllern.

Chancen ergeben sich, wenn ein konventioneller Controller durch ein externes I/O-IC oder ein FPGA erweitert werden muß. Der FPGA-Prozessor als prinzipielle Einchiplösung hat hier einen Platz - und wahrscheinlich auch schon einen Preisvorteil. Ein weiterer Ansatzpunkt ergibt sich bei Anwendungen mit extremen Anforderungen an Multitasking. Auch die anwendungsspezifische Erweiterbarkeit des Befehlssatzes kann entscheidend sein. Mit verbesserten Shiftbefehlen kann man Algorithmen wie CORDIC beschleunigen. Und damit DSPs oder FloatingPoint-Coprozessoren aus einfacheren Anwendungen kostengünstig verdrängen.

Es wird noch eine gute Weile dauern, bis die Entwicklungsprobleme gelöst sind. Vielleicht kommen dann auch schon die ersten Systeme aus den USA. Das wäre schade, denn bei FPGAs herrschen gleiche Chancen für alle. Bisher war dem nicht so. Mangels kompetenter Halbleiterhersteller und der nötigen Infrastruktur, ist es in Deutschland nicht so einfach wie in Kalifornien Prozessoren zu machen. FRP1600 und Hyperstone waren nur gegen erhebliche Widerstände durchzusetzen. Mit FPGAs entfallen einige Bremsklötze.

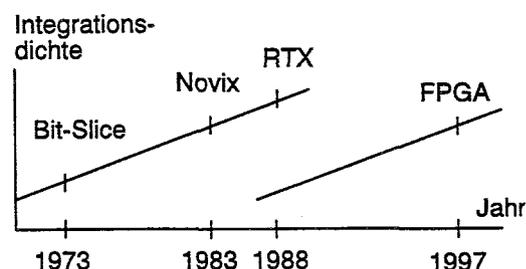


Bild 3: historische Entwicklung

Listings und Files aus der Box „saugen“ Schöne Sachen selber saugen.

Michael Kalus

Plönerstr. 24a; D-23715 Malente; michael@malente.forth-ev.de

Stichworte: KBBS Download Listings

In der Box der Forthgesellschaft liegen viele schöne Sachen bereit und warten darauf abgeholt zu werden. Mit CROSS POINT (XP) ist das sehr einfach. Die Files können in aller Ruhe zu Hause ausgesucht und dann per Knopfdruck bestellt werden, einzeln oder als Liste. Und Schwuppdiwupp, sind sie da.

Überhaupt keine Schwierigkeiten? Eigentlich nicht. Uneigentlich aber doch. Denn ich verstand anfangs noch nicht, wie die Sache organisiert ist - das mit dem FILESERVER, der Bestellungen macht und den Befehlen an die KBBS. Und die Ausdrucksweise im XP war mir nicht immer gleich verständlich. Außerdem ahnte ich anfangs garnicht, was da alles bereit liegt - um Files bestellen zu können, muß man schließlich erst einmal wissen, wo es sie gibt! Und darum schreib ich das jetzt hier auf (und weil der Claus das so wollte :-)

Erster Anruf im KBBS: INDEX bestellen.

Und so beginnt man die Entdeckungsreise zu den Schätzen in der KBBS. Zuerst einmal fordert man die Liste der Files an. Das geht per Knopfdruck. Die Bestellung kann in XP einfach über das Menü gemacht gemacht werden:

(N)achrichten

(F)ileserver

(L)iste

(a)nfordern

Das ist alles. Der Fileserver des XP erzeugt daraus eine passende Kommandodatei für die Box (1). Beim nächsten Anruf geht diese mit raus. Geliefert wird sogleich der INDEX aller Files. Warum die Liste der Files INDEX genannt wird, wissen die Götter, aber sie kommt als komprimierte Datei INDEX.GZ und wird von XP im Verzeichnis C:\CROSS\FILES abgelegt.

Nun geht es nicht mehr so ganz automatisch weiter, denn der INDEX.GZ muß zunächst dekomprimiert werden. Dazu muß man XP leider verlassen (Taste F9) und GZIP.EXE -d c:\cross\files\INDEX.GZ ausführen. GZIP ist im CROSS POINT Paket dabei. Hilfe bei der Syntax erhält man mit GZIP -h wie üblich. So, ausgepackt steht jetzt der INDEX im Textformat zur Verfügung und kann auch ohne spezielle Tools gelesen werden. Um damit auf komfortable Weise Files bestellen zu können, muß

man XP allerdings noch mitteilen, daß INDEX jetzt die neue FORTH-EV.FL ist. Das geht so: Man kehre zurück ins Hauptmenü des XP und wähle dort mit

(N)achrichten

(F)ileserver

(L)isten

(D)atei einlesen

den soeben erzeugten INDEX aus. XP macht daraus dann ohne weiteres Zutun seine neue FORTH-EV.FL Fileliste. Und ist dabei so freundlich, die alte Liste als FORTH-EV.BAK aufzuheben - für alle Fälle.

Zweiter Anruf im KBBS: FILE(S) bestellen.

Da nun alle Files jederzeit OFFLINE zur freien Auswahl bereit stehen, kann in aller Ruhe in der Liste gekramt werden. Im Listenleser des XP wird dann auch gleich bestellt. In der Liste *lesen* und dabei Files *markieren* bedeutet diese *bestellen*. Darum findet man im Menü auch nur diesen Pfad:

(N)achrichten

(F)ileserver

(B)estellen

So in der Fileliste alias INDEX alias FORTH-EV.FL gelandet, wird beim Anschauen mit der Leertaste markiert, was man haben will. Dann hat man die Wahl direkt die Box anzurufen oder die Bestellung später abzuschicken. XP macht dazu den passenden Brief (mail) fertig, der dann halt beim nächsten Anruf zugestellt wird.

3. Viele Wege führen nach Rom

Nun ist es so, daß Holger Petersen in etwa monatlich in den KBBS/general-Nachrichten die *neuen* Files veröffentlicht. Er startet diesen Vorgang derzeit manuell, wann immer er glaubt, daß genug Neues drin ist.

Diese Liste kann nach FORTH-EV.FL gebracht werden noch während man die Nachricht liest. Taste W drücken und damit die Nachricht in eine Datei kopieren, diese dann wie oben beschrieben mit NFLD einlesen, fertig. Hierbei muß XP also nicht verlassen werden. Allerdings verwaltet XP immer nur *eine* Fileliste! Bringt man also die neue Liste dorthin, wird die alte zwar wieder nach FORTH-EV.BAK gerettet. Der INDEX wird aber damit herausgedrängt. Das hat mir nicht gefallen, ist aber leider so im XP. Übrigens landen alle bestellten Files im Verzeichnis C:\CROSS\FILES, wobei das System so nett ist, diesen Vorgang mit einer Notiz bei den privaten Nachrichten anzuzeigen.

Und so gehts per Hand...

Und kann man denn nicht auch einzelne Dateien 'einfach so' bestellen? Man kann. Aber man muß das Format einhalten, sonst wirds natürlich nix. Also, hier bestelle ich mal 2 Files:

```
forth/INDEX
forth/vd/vd1962.zip
```

Die korrekte gRoSs- und kLeIn-Schreibung sowie die Verwendung von Slash / statt Backslash \ sind essentiell! Die Angabe des Pfades ist erforderlich. Sonst nichts reinschreiben. Absenden als PM an den USER, der auf den schönen Namen UUCP-Fileserver@forth-ev.de hört.

4. Und wer darf das?

Ist das ganze jetzt nur für Mitglieder der Forthgesellschaft oder können auch andere Menschen an die Files heran? Nun, auch Fremde könnten in der Box Files bestellen.

Wie? So: Wer keinen uucp-Login bei der Box des Forth-eV's KBBS hat, darf auch per 'anonymem uucp' bestellen. Im XP-Paket gibt's eine DOK dazu. Kurz gesagt geht das so: Man muß eine Mailbox (forth-ev/kbbs) mit dem Usernamen 'nuucp' und dem Passwort 'nuucp' und natürlich der Telefonnummer der Box neu ins XP eintragen. Damit geht's dann wie oben beschrieben.

Aber da wir ja alle in der Forthgesellschaft sind und unseren Jahresbeitrag bereits entrichtet haben, haben wir natürlich auch alle schon unser Passwort von Holger Petersen samt XP-Paket angefordert - ja, das alles ist im Jahresbeitrag bereits enthalten!

Und nun viel Spaß und viel Erfolg! Man trifft sich im KBBS im Forth-ev/Forum. Euer Michael.

Was noch...

Die Leserinnen und Leser dieser Zeitschrift wird sicher besonders freuen, daß sie außer der Datei:

forth/vd/vd1962.zip

auch diverse andere Listings der Jahrgänge 1995 und 1996 des Forth-Magazins direkt schnell und kostengünstig aus der Box saugen können:

8967	Sep	14	1991	forth/vd/vd-srcs.lzh
94452	Jul	21	1995	forth/vd/vd95-1.zip
4889	Jul	21	1995	forth/vd/vd95-2.zip
56301	Nov	11	1995	forth/vd/vd95-3.zip
5120	Oct	3	1991	forth/vd/vd_3_91.zip
37357	Aug	26	21:54	forth/vd/vd1954.zip
54143	Aug	25	04:06	forth/vd/vd1961.zip
86749	May	19	14:12	forth/vd/vd1962.zip
49254	Aug	25	04:06	forth/vd/vd1963.zip

und demnächst auch:

forth/vd/vd1964.zip

Die KBBS ist zu erreichen unter:

<u>Nummer Analog</u>	<u>ISDN</u>
5339898 TKR 14400	X75
5339899 Motorola 28800 AVM Fritz! (Spezialeinstellung!)	

Für ältere Listingdisketten sollte lieber auf den Vertrieb Klaus Kohl zurückgegriffen werden.

Forthvertrieb Klaus Kohl;
PF 1173;
D-86406 Mering;

Tel.: 08233 / 30524
eMail: kk@forthver.FORTH-eV.de

Hier kann unter anderem eine CD mit allen vorhandenen Listings vor 1995 bestellt werden.

Claus Vogt

(1) Anmerkung von Holger Petersen beim Korrekturlesen dieses Beitrages:
„Pur technisch gesehen werden bei jedem Anruf von XP bei der KBBS - wie zwischen zwei Unix-Systemen eben - zwei Dateien verschickt; eine mit Daten und eine mit Kommando's, was mit jenen Daten zu geschehen hat. Normalerweise sind das Kommando's an RMAIL und RNEWS, die entsprechenden Daten als Mail bzw. News aufzufassen und artgerecht weiterzuverarbeiten. Es gibt aber auch das Kommando UUCP (unix-to-unix-copy), wo befohlen wird, eine Datei zu kopieren. Wenn Du einen Upload machst, ist die Datei natürlich dabei; bei einem Download wird von Dir nur die o.a. Kommando-Datei übertragen, und die Daten (sowie eine Rück-Kommando) kommen zu Dir.“

Serie: PC-Meßtechnik, Teil VI

PC-Meßkarten

Autor: Klaus Kohl / BMC Systeme GmbH

Zeppelinstr. 10; 86406 Mering; Tel.: 08233 / 30524

Im vorerst letzten Teil der PC-Meßtechnikserie schließt sich der Kreis um die an einem PC verfügbaren Schnittstellen, und wir kehren zu dem zurück, was dem Rechner erst seine vielfältigen Anwendungen ermöglicht. Bei der Beschreibung der IBM-kompatiblen PC/AT-Slots und deren Verwendung für Meßkarten sollen auch die Interrupt- und DMA-Leitungen und deren Ansteuerung nicht zu kurz kommen.

Stichworte: Meßtechnik PC PC/AT-Slot Interrupt DMA

1. Der PC/AT-Slot

Da das Mainboard des ersten PC im August 1981 mit einem 8088-Prozessor, 64KByte RAM (Standardausstattung, erweiterbar auf 640K), 40KByte ROM und der notwendigen Verbindungslogik einschließlich Timer, DMA- und Tastaturcontroller schon sehr gefüllt war, wurde entschieden, die optionalen Erweiterungen wie Schnittstellen (seriell und parallel) und Grafikkarten nicht fest einzubauen, sondern als steckbare Ergänzungen vorzusehen. Um sich die Arbeit einfach zu machen, legte man einfach die notwendigen Prozessorsignale für Speicher- und I/O-Zugriff sowie die freien Interrupt- und DMA-Leitungen auf fünf sogenannte Slots, die jeweils eine Platine mit ebenen Kontaktflächen aufnehmen können. Die erste, meist mitgelieferte Karte war eine Grafikkarte.

Obwohl inzwischen schon die unterschiedlichsten PC-Busse wie VESA Local Bus, Microchannel und PCI eingesetzt wurden und werden, bleiben die hier beschriebenen Slotkarten auch in der Meßtechnik die verbreitesten PC-Erweiterungen.

1.1 Pinbelegung

Da zuerst nur ein Prozessor mit 8Bit-Datenbreite verwendet wurde, genügte auch ein 8Bit-Erweiterungsslot mit 2*31-poligen Klammern.

Erst nach Einführung des 80286-Prozessors und den anderen Erweiterungen des PC-AT wie zusätzliche Daten- und Adreßleitungen, weiteren Interrupts und den für 16Bit-Übertragungen gedachten DMA-Kanäle war eine Erweiterung nötig. Diese wurde glücklicherweise so gewählt, daß einfach eine zweite, 2*18polige Klammer den 8Bit-Slot fortsetzt.

8Bit-Slot			
GND	B1	A1	-I/O CHCK
RESET DRV	B2	A2	D7
+5V	B3	A3	D6
IRQ2	B4	A4	D5
-5V	B5	A5	D4
DRQ2	B6	A6	D3
-12V	B7	A7	D2
Reserviert	B8	A8	D1
+12V	B9	A9	D0
GND	B10	A10	I/O CH RDY
-MEMW	B11	A11	AEN
-MEMR	B12	A12	A19
-IOW	B13	A13	A18
-IOR	B14	A14	A17
-DACK3	B15	A15	A16
DRQ3	B16	A16	A15
-DACK1	B17	A17	A14
DRQ1	B18	A18	A13
-DACK0	B19	A19	A12
CLOCK	B20	A20	A11
IRQ7	B21	A21	A10
IRQ6	B22	A22	A9
IRQ5	B23	A23	A8
IRQ4	B24	A24	A7
IRQ3	B25	A25	A6
-DACK2	B26	A26	A5
T/C	B27	A27	A4
ALE	B28	A28	A3
+5V	B29	A29	A2
OSC	B30	A30	A1
GND	B31	A31	A0

16Bit-Slot			
- MEM CS16	D1	C1	SBHE
-I/O CS16	D2	C2	LA23
IRQ10	D3	C3	LA22
IRQ11	D4	C4	LA21
IRQ12	D5	C5	LA20
IRQ15	D6	C6	LA19
IRQ14	D7	C7	LA18
-DACK0	D8	C8	LA17
DRQ0	D9	C9	-MEMR
-DACK5	D10	C10	-MEMW
DRQ5	D11	C11	SD08
-DACK6	D12	C12	SD09
DRQ6	D13	C13	SD10
-DACK7	D14	C14	SD11
DRQ7	D15	C15	SD12
+5V	D16	C16	SD13
-MASTER	D17	C17	SD14
GND	D18	C18	SD15

1.2 Einsatzmöglichkeiten

Auf dem PC-Markt verfügbare Karten zeigen die Nutzung der PC/AT-Slots:

·Reine I/O-Karten:

Die parallelen und seriellen Schnittstellenkarten nutzen nur den I/O-Speicherbereich. Dazu genügt es, wenn neben den Adreßleitungen A0 bis A9 und den Datenleitungen D0 bis D7 nur noch -IOR (I/O-Read), -IOW (I/O-Write) und ALE (Address latch enable) berücksichtigt werden.

Oft werden dann noch Interruptleitungen IRQ3/4 (für RS232) bzw. IRQ5 oder 7 (LPT) verwendet.

·Grafikkarten:

Diese Karten verwenden außer dem I/O-Bereich auch noch den Speicher unterhalb der 1MByte-Grenze für die Betriebssystem-Erweiterung und den Videospeicher.

·DMA-Karten:

Festplattencontroller liefern Daten schneller, als früher die CPU sie speichern konnte. Deshalb entwickelte man einen Baustein, der die Daten direkt in das RAM bringt. Von diesem Baustein sind zwei freie Kanäle und zwei weitere Acknowledge-Leitungen am Slot verfügbar.

Eine später nicht mehr erläuterte Leitung ist I/O CHCK. Damit kann die Karte dem Prozessor mit einer 0 signalisieren, daß ein Speicherfehler aufgetreten ist.

Da der PC im Normalfall auch bei einem 16Bit-Zugriff auf I/O und RAM die Daten byteweise holt, muß eine Slotkarte rechtzeitig an -MEM CS16 bzw. -I/O CS16 seine 16Bit-Breite anzeigen.

1.3 Freie I/O- und RAM-Bereiche

Natürlich kann nicht jede Karte einen beliebigen Adreßbereich für sich in Anspruch nehmen, weil sonst die Funktion des gesamten Systems gestört werden könnte. Auf Grund der PC-Entwicklung sind auf dem Mainboard schon Bereiche reserviert und wegen entsprechender BIOS- und Windows(95)-Unterstützung sollte man andere Adressen nur für bestimmte Karten nutzen. Der Rest ist dann unser Schlachtfeld für Meßkarten.

RAM-Bereiche

\$000000 - \$09FFFF	640Kbyte RAM
\$0A0000 - \$0BFFFF	Speicher der Grafikkarte
\$0C0000 - \$0C3FFF	BIOS der Grafikkarte
\$0C4000 - \$0C7FFF	BIOS des Festplattencontrollers
\$0C8000 - \$0EFFFF	Weitere BIOS oder Upper Memory
\$0F0000 - \$0FFFFF	BIOS des Mainboards
\$100000 - \$10FFEF	Upper Memory für DOS
\$10FFFF -	weiteres RAM

Da der Speicher für Programme (hauptsächlich Spiele) unterhalb der 640K-Grenze sehr kostbar ist, wird der nicht benötigte Speicher oberhalb der Grafikkarte für ständig benötigte Programme wie Tastaturreiber oder Betriebssystem genutzt. Durch einen Trick kann der Prozessor selbst im 16Bit-Modus noch den Speicherbereich über der magischen 1MByte-Grenze nutzen. Dazu wird einfach zu einem Seitenregister des Prozessors (ist für die absolute Adresse mit 16 zu multiplizieren) der Offset des gewünschten Zeigers addiert. Dadurch kann er noch die Speicher-

adresse \$FFFF0 + \$FFFF = \$10FFEF erreichen, falls das sogenannte A20-Gate mitspielt. Dieses Gate wird über das Betriebssystem verwaltet.

I/O-Bereiche

\$000-\$00F	DMA-Controller 1 (8237)
\$010-\$01F	reserviert (DMA im PS/2)
\$020-\$03F	Interruptcontroller 1 (8259)
\$040-\$05F	Timer (8253), Watchdog (PS/2)
\$060-\$06F	Tastaturcontroller, 8255
\$070-\$07F	RTC und NMI-Maske
\$080-\$09F	DMA-Seitenregister
\$0A0-\$0BF	Interruptcontroller 2 (8259)
\$0C0-\$0DF	DMA-Controller 2 (8237)
\$0E0-\$0EF	reserviert für Systemplatine
\$0F0-\$0FF	Ansteuerung des Coprozessor 80287
\$100-\$1EF	frei für Erweiterungen
\$1F0-\$1FF	Festplattencontroller
\$200-\$20F	Gameport
\$210-\$21F	frei für Erweiterungen
\$220-\$22F	Soundblaster
\$230-\$26F	frei für Erweiterungen
\$270-\$27F	Parallelport 2
\$280-\$2AF	frei für Erweiterungen
\$2B0-\$2DF	EGA-Grafikkarte
\$2E0-\$2E7	frei für Erweiterungen
\$2E8-\$2EF	Serielle Schnittstelle 4
\$2F0-\$2FF	Serielle Schnittstelle 2
\$300-\$31F	Ethernetkarten (z.B. NE2000)
\$320-\$36F	frei für Erweiterungen
\$370-\$37F	Parallelport 1
\$380-\$3AF	Diverses
\$3B0-\$3BB	MDA-Grafikkarte (Herkules)
\$3BC-\$3BF	Parallelschnittstelle (z.B. im Compac)
\$3C0-\$3CF	EGA-Grafikkarte
\$3D0-\$3DF	CGA-Grafikkarte
\$3E0-\$3E7	frei für Erweiterungen
\$3E8-\$3EF	Serielle Schnittstelle 3
\$3F0-\$3F7	Floppy Controller
\$3F8-\$3FF	Serielle Schnittstelle 1

Hier muß man darauf achten, daß die meisten I/O-Karten nur die Adreßleitungen A0 bis A9 dekodieren und deshalb im restlichen Speicherbereich als Spiegelungen auftauchen. Auf der anderen Seite dekodieren einige Bausteine wie DMA-Seitenregister oder Joystick die Adreßleitungen auch nicht vollständig und sind deshalb mehrfach im angegebenen Bereich vorhanden. Man muß sich vor Verwendung einer Karte immer vergewissern, ob der gewünschte Bereich leer ist.

1.4 Timing

Timing wird immer irgendwie mit einem Takt verbunden. In einem PC wird hier zwischen dem Prozessortakt und dem Bustakt unterschieden. Beginnend mit 4,77MHz am ersten PC stieg sowohl der Prozessortakt (jetzt schon über 200MHz) als auch der Bustakt (standardmäßig ca. 8MHz für I/O) kontinuierlich. Für den Zugriff auf eine Karte ist aber die Zeit zwischen dem Anlegen aller relevanten Signale (z.B.: A0-A9, D0-D7, ALE und IOW) und der Übernahme der Daten von der I/O-Karte entscheidend. Diese Zeit kann meist im Setup des Mainboard eingestellt werden. Aber auch die Karte hat ein Mitspracherecht durch die Lei-

tung I/O CH RDY (I/O Channel Ready). Solange sie auf 0 gezogen wird, wartet der Prozessor. Wird sie nicht angesteuert, so bleibt sie auf 1 und der Prozessor arbeitet mit den im Setup eingestellten Werten.

2. Interruptleitungen

Wir haben im Laufe dieser Serie schon mehrfach mit Interrupts zu tun gehabt. Neben dem auf der Grundplatine verwendeten Timer-Interrupt 0 haben wir auch schon die herausgeführten Interruptleitungen 3 und 4 bei der seriellen Schnittstelle genutzt. Im Uralt-PC gab es nur einen Interruptcontroller, der über 8 Interrupteingänge verfügt und an I/O-Adresse \$20/\$21 angesprochen wird. Im AT leistete man sich einen zweiten Controller mit weiteren 8 Eingängen. Weil man aber keine eigene Prozessoreitung spendieren wollte, werden diese Interrupts über Eingang IRQ2 durch den ersten Interruptcontroller geführt.

2.1 Verwendung im AT

Wie man an folgender Liste sieht, sind die auf den 8bit-Slot geführten Interrupts meist schon in festen Händen. Deshalb sollte für eigene Karten eher die Verwendung der selten genutzten IRQ10-IRQ12 und IRQ15 angestrebt werden.

Leitung	Interruptvektor	Verwendung
IRQ0	\$08	Timer 0-Ausgang
IRQ1	\$09	Tastatur
IRQ2	—	Verkettung mit Interrupt 8-15
IRQ3	\$0B	Serielle Schnittstelle 2 / 4
IRQ4	\$0C	Serielle Schnittstelle 1 / 3
IRQ5	\$0D	Festplattenlaufwerk oder LPT2
IRQ6	\$0E	Diskettenlaufwerk
IRQ7	\$0F	LPT1
IRQ8	\$70	Echtzeituhr (CMOS-RTC)
IRQ9	\$0A/\$71	Bildschirmsteuerung (alte IRQ2)
IRQ10	\$72	frei (meist Ethernetkarten)
IRQ11	\$73	frei
IRQ12	\$74	frei (oder Mausschnittstelle)
IRQ13	\$75	Coprozessor
IRQ14	\$76	Festplattencontroller
IRQ15	\$77	frei

Jedem Eingang ist ein eigener Vektor in der Interrupttabelle des Prozessors zugeordnet. Jeweils 4 Byte ab Adresse \$0020 sind für IRQ0 bis 7 und ebenfalls jeweils 4 Byte ab Adresse \$01C0 sind für IRQ8 bis 15.

2.2 Ansteuerung

Der Interruptcontroller 8259 löst in der üblichen PC-Einstellung einen Interrupt aus, wenn eine 1 (bzw. 5V) an einem der Interrupteingänge anliegt. Deshalb muß das zugehörige Interruptprogramm vor der Freigabe weiterer Interrupts erst dafür sorgen, daß der entsprechende Baustein die Leitung wieder zurücksetzt. Wie im Teil 2 (1995/3-S.16) am Beispiel des Timers oder im Teil 4 (1995/2-S.29) mit RS232 geschildert, muß nach Bearbeitung der gewünschten Routine durch Einschreiben des Wertes \$20 (unspezifischer End Of Interrupt) in Portadresse \$20 (Command-Register) der nächste Interrupt zugelassen werden.

Die Interruptleitung selbst wird durch Löschen des zugehörigen Bits im Maskenregister auf I/O-Adresse \$21 aktiviert. Durch Setzen dieses Bits wird der Status für diese Leitung ignoriert und keine Interrupts mehr ausgelöst.

Bei den Interruptleitungen IRQ8 bis IRQ15 wird die Verkettung mit dem ersten Controller genutzt und über Eingang IRQ2 durchgeschleift. Es muß deshalb nach dem Register des zweiten Interruptcontrollers auf Adresse \$A0/\$A1 auch der erste Controller wie oben beschrieben zurückgesetzt werden. Das Aktivieren der Interruptleitung IRQ2 wird normalerweise schon durch das Betriebssystem vorgenommen.

3. DMA-Ansteuerung

Sollen Datenmengen im Bereich von mehreren 10000Hz durch einen PC bearbeitet werden, so ist ein Interruptsystem nicht das geeignete Mittel. Schon allein durch den EMM386 ist ein Jittern (der Abweichung zwischen minimaler und maximaler Zeit) der Interruptbearbeitung von bis zu 50µs zu beobachten. Außerdem kann jeder andere Interrupt das System beim Speichern eines Blockes auf Festplatte oder Diskette fast beliebig lange blockieren. Auch ein Maustreiber ist ein guter Kandidat für derartige Verzögerungen.

In solchen Fällen verwenden wir ähnlich wie z.B. der Festplattencontroller die am Steckplatz verfügbaren DMA-Leitungen. Diese sind mittels des eingebauten Controllers in der Lage, die Daten direkt von der Karte in einen angegebenen Speicher zu schreiben. Da dabei der Prozessor nicht eingreifen muß, braucht er nur für die Dauer der Übertragung angehalten werden.

Meist wird die DMA genutzt, um einen Datenblock schnell in den PC-Speicher zu bringen. Dazu wird bei jedem Signal von der Hardware ein Byte oder Wort (im AT) übertragen, Zieladresse erhöht und der DMA-Zähler erniedrigt. Bei 0 ist die Übertragung beendet.

In einem anderen Modus, der gut für die Meßdatenerfassung genutzt werden kann, verwendet man einen definierten Bereich im Speicher. Dieser wird wie zuvor mit Werten von der Meßkarte gefüllt. Hat der Zähler dann 0 erreicht, so springt der Adreßzeiger wieder auf den Anfang und der Zähler beginnt wieder mit dem Startwert. Dadurch kann ein Puffer von maximal 64KByte bzw. 128KByte bei 16Bit-DMA als Zwischenpuffer genutzt werden.

Hardwaremäßig hat man für jeden DMA-Kanal nur die beiden am Slot verfügbaren Leitungen DMA-Request (DRQ) und DMA-Acknowledge (-DACK). Um noch über Lesen und Schreiben zu entscheiden, werden dann noch -IOR bzw. -IOW benötigt. Will eine Slotkarte eine DMA-Übertragung auslösen, so setzt es die DRQ-Leitung. Sobald der DMA-Controller dann die Werte in den Speicher schreiben will oder Daten bereitstellt, zieht er die -DACK-Leitung auf 0 und steuert noch -IOR oder -IOW an. Die Adressen und das -MEMR bzw. -MEMW dürfen von der Karte nicht beachtet werden, weil damit gleichzeitig der Speicher angesteuert wird.

Tip: Bitte versuchen Sie niemals, ein Latch direkt über DMA einzulesen. Da sich der Wert noch in letzter "Sekunde" ändern kann, liefert der Paritycontroller nicht den richtigen Wert. Dies führt dann beim Lesen der Daten aus dem Speicher zu einem Parityfehler, der meist mit erneutem Booten des Rechners gehandelt wird.

3.1 Verwendung im PC

Im PC wurde nur ein DMA-Controller mit 4 Kanälen bestückt. Da es sich um einen 8Bit-Baustein handelte, kann er auch nur 8Bit-Daten und 16Bit-Adressen bearbeiten. Um die schon im ersten PC verfügbaren 20 Adreßleitungen nutzen zu können, wurde noch durch einen zusätzlichen Baustein die Adreßleitungen A16 bis A19 während der Übertragung auf vorgegebene Werte gesetzt. Leider hat diese geschichtliche Tatsache dazu geführt, daß in jedem PC der verwendete DMA-Puffer innerhalb eines Blockes von 64KByte begrenzt bleibt. Durch seine Beschaltung ist leider nur eine Übertragung von I/O-Bereich auf das RAM oder umgekehrt möglich. Dies ist aber kein Nachteil, da die heutigen Prozessoren sowieso schneller sind als die inzwischen zu langsame DMA.

Die netteste Möglichkeit, den PC in den Selbstmord zu treiben, war die Abschaltung der DMA0. Sie wurde früher aus Kostengründen für den Refresh des RAM's verwendet. Heute erledigen dies die großen namenlosen Chips auf dem Mainboard und die DMA0 kann frei genutzt werden. Die anderen Kanäle werden wie folgt verwendet:

<u>Leitung</u>	<u>Verwendung</u>
DMA0	früher RAM-Refresh, jetzt frei
DMA1	frei
DMA2	Diskette
DMA3	früher Harddisk, jetzt frei

3.2 Verwendung im AT

Natürlich wurde dann im AT das Seitenregister des ersten Controllers erweitert und kann jetzt A16 bis A23 ansteuern, was den Zugriff auf 16MByte erlaubt. Außerdem wurde ein zweiter, gleichartiger Baustein für DMA 4 bis 7 eingesetzt. Um aber für den erweiterten AT-Slot die 16Bit-Übertragung zu nutzen, wurde er speziell für diese Zweck anders betrieben. Durch Verschiebung der Adressen um ein Bit nach links (A0 ist bei 16Bit-Übertragung immer 0) kann der Baustein selbst 128KByte und mittels des ebenfalls verschobenen Seitenregister in manchen Rechnern bis zu 32MByte adressieren. Die Verwendung der zusätzlichen DMA-Kanäle ist wie folgt:

<u>Leitung</u>	<u>Verwendung</u>
DMA4	Verkettung mit ersten Baustein, nicht nutzbar
DMA5	PS/2-Harddisk, sonst frei
DMA6	frei
DMA7	frei

3.3 Programmierung

Der Ablauf der DMA-Programmierung geschieht wie folgt:

- Anforderung des DMA-Speichers (kein Seitenwechsel !!!)
- DMA abschalten
- Seitenregister setzen
- Anfangsadresse des DMA-Speichers im Controller setzen
- Größe des DMA-Speichers setzen
- Modus einstellen
- DMA aktivieren
- DMA auf der I/O-Karte initiieren
- Kontrolle, bis gewünschte Datenanzahl übertragen wurde

- DMA auf der I/O-Karte stoppen
- DMA-Kanal deaktivieren
- Daten verarbeiten
- Speicher freigeben

Dies nur theoretisch zu behandeln, ist sehr schwierig und wegen der vielen Register in den oft abgedruckten Listing manchmal falsch. Außerdem ist sowohl die Initialisierung der entsprechenden Hardware im PC-Slot als auch die Verarbeitung der Daten aus dem DMA-Puffer wichtig. Ich werde deshalb in einer späteren Folge eine konkrete Anwendung mit einer eigener Hardware schildern, weil es hier den Platz sprengen würde.

4. Wie geht's weiter

Hiermit findet die Serie zur PC-Meßtechnik seinen Abschluß. Jedoch ist deshalb keine Trauer angesagt, da die eigentliche Meßtechnik erst beginnt. Bei weitergehender Numerierung wird jetzt ohne notwendigen aber meist vorhandenen Bezug zum PC die Serie fortgesetzt. Dies geschieht, um mehr den Blick auf die eigentlichen Probleme (und natürlich auch deren Lösungen) bei Meß- und Steuerungsaufgaben freizuhaben. Neben dem PC werden deshalb auch vermehrt Beispiele zur Anwendung von Microcontroller aufgezeigt. Darüber hinaus sollen aber auch Hardware und Algorithmen nicht zu kurz kommen. Folgende Themen befinden sich dabei auf dem Stack:

- Informationen zu A/D und D/A-Wandler
- Preiswerte Sensoren für diverse physikalische Größen
- Integration und Differentiation
- Digitale Filter, Mittelungen und Glätten
- FFT für Signalanalyse und Synthese
- Beachtenswertes bei Regelkreisen
- Beispiele zum Einsatz von Meßtechnik in der Industrie

Bei allen Folgen wird dabei Wert auf die Praxis gelegt und meist eine entsprechende Anwendung geschildert oder sogar programmiert. Für den vertieften Einstieg in die Theorie muß deshalb auf die immer angegebene Literatur verwiesen werden.

Leider ist bei einem Erscheinungsintervall von 3 Monaten kaum an eine Beantwortung von Fragen über dieses Medium zu denken. Trotzdem würde ich mich über entsprechende Briefe oder Mails mit Fragen oder Anregungen freuen.

Literatur

- (1) Thom Hogan
Die PC-Referenz für Programmierer
Microsoft Press
- (2) Blank/Bernstein
PC-Schaltungstechnik in der Praxis
ISBN 3-89090-914-0
Markt-und-Technik-Verlag 1990
- (3) Frank Van Gulluwe
The undocumented PC
ISBN 0-201-62277-7
Addison Wesley
- (4) Tischler / Jennrich
PC INTERN (1-5)
ISBN 3-8158-1169-4
Data Becker GmbH&Co.KG (1995)

□

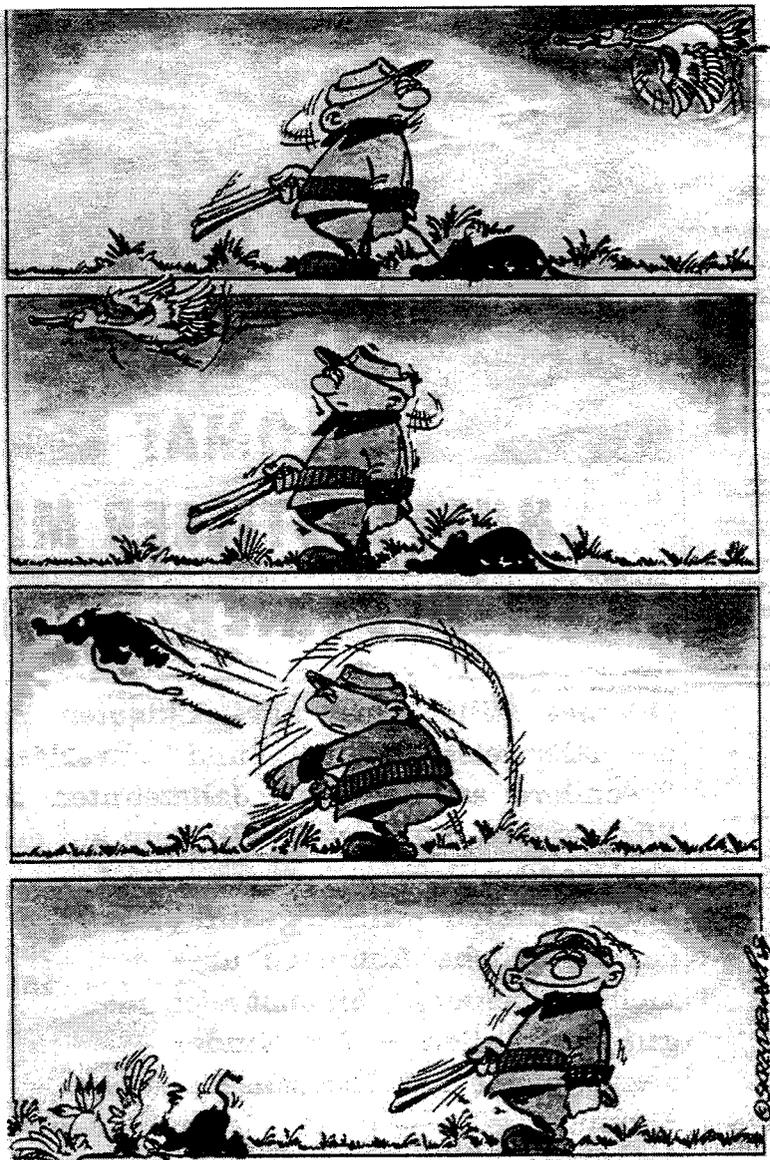

```

39 \ timing (external ROM)                U.B-BIN 10 ms
40 \ ( including XOR and OR ) ... XOR ... OR S" .....".BYTE 8 ms
41 \ ( with high level NOS-7TH-BIT? )    S" .....".BYTE 9 ms
42
43
44 TITLE: _BYTE_
45
46 DECIMAL
47
48 \ : NOS-7TH-BIT? ( x1 x2 - x1' x2 f ) \ tests NextOnStack for 7th
bit
49 \      >R 2* R> OVER $100 AND :
50
51 CODE NOS-7TH-BIT? ( x1 x2 - x1' x2 f ) \ tests NextOnStack for 7th
bit
52 \ f is FALSE if 7th bit was clear, 256 if set
53 \ x1' == x1 *2 ( shifted left )
54 2 @R7 R3   MOVI. \ get second cell on stack
55 R3        SHLL. \ shift left ( 2* )
56 2 @R7 R3   MOVO. \ write back modified
57 $100 ## R3 AND. \ 7th bit already shifted to
pos. 8

58 R3PUSH-END \ NEXT 2- JMP. SMUDGE
59
60 \ PAD is a USER variable. NO global usage
61
62 : (.BYTE) ( x ca 1 - ca 2 12 ) \ convert to (transient)
string
63 \ string at memory is counted ( countbyte at ca2-1 == PAD )
64 \ display or move the string soon, before it's overwritten
65 8 MIN \ UMIN not defined "no
ans
66 DUP PAD C! \ store countbyte
67 0
68 ?DO NOS-7TH-BIT?
69 IF DUP I + C@ \ transfer CHAR from mask
70 ELSE [CHAR] . \ transfer 0 NOT-SET
71 THEN PAD CHAR+ I + C! \ to string at PAD
72 LOOP 2DROP
73 PAD COUNT :
74
75 : .BYTE ( x ca 1 - ) \ display BYTE
76 (.BYTE) TYPE :
77
Langste Zeile hat 79 Zeichen

```

Die Geschichte von Programmierer Erwin K.



Heute sahen wir: Wie Programmierer Erwin K. einmal nach einer durchzechten Nacht dank seines treuen Forthes erfolgreich dem Kormoran nachstellte.

Nächstesmal: Wie Programmierer Erwin K. einmal fürchterlichen Ärger mit dem Tierschutzverband kriegte und sein Forth ihm dabei garnicht helfen konnte.

(kk & clv)



Schachcomputer

Die historische Entwicklung

von Rafael Deliano

Steinbergstr. 37; D-82110 Germering

Schachcomputer werden in der Öffentlichkeit häufig als praktisch erfolgreiches Resultat der AI-Forschung angepriesen. Ein Blick hinter die Kulissen ist eher ernüchternd.

Stichworte: Schachcomputer Geschichte

Schach erfreute sich bis in die 70er Jahre einer gewissen Beliebtheit in „intellektuellen“ Kreisen. Manch einer hoffte, daß der Anspruch des Spiels auf das Image des Spielers abfärbt. Damit ist es vorbei, seitdem billiges Spielzeug 'Made in Hongkong' jeden Amateur an Spielstärke übertrifft. Bisher gelingt es dem Computer jedoch noch nicht Spitzenspieler zu schlagen. Der Ausgang des Kampfes ist trotzdem unzweifelhaft und das Ende ist nahe.

Der menschliche Spieler

Unmittelbar vor dem 2. Weltkrieg hatte der holländische Psychologe Adrain de Groot Experimente am menschlichen Schachspieler durchgeführt, die Einblick in dessen Spielweise geben. Der Mensch spielt Schach durch Mustererkennung. Er blickt aufs Brett und erkennt dort zusammenhängende Teilstrukturen, „Schachwörter“. Er verknüpft diese Fragmente durch eine Syntax und reduziert sie dadurch auf eine abstrakte Bedeutung. Somit besteht eine Ähnlichkeit zum normalen Denken, das auch Sprache und Wörter braucht, um Gedanken formulieren zu können. Diese Abstraktion im Gehirn ermöglicht ihm eine drastische Reduzierung der Datenmenge, die er verarbeiten muß, selbst wenn das Brett noch voll mit Figuren belegt ist. Der gute Spieler erkennt intuitiv die wenigen sinnvollen Züge. Damit gelingt es ihm auf bis zu 14 Halbzüge vorauszublicken. Dies wiederum ermöglicht ihm, auf langfristige, strategische Ziele hinzuarbeiten.

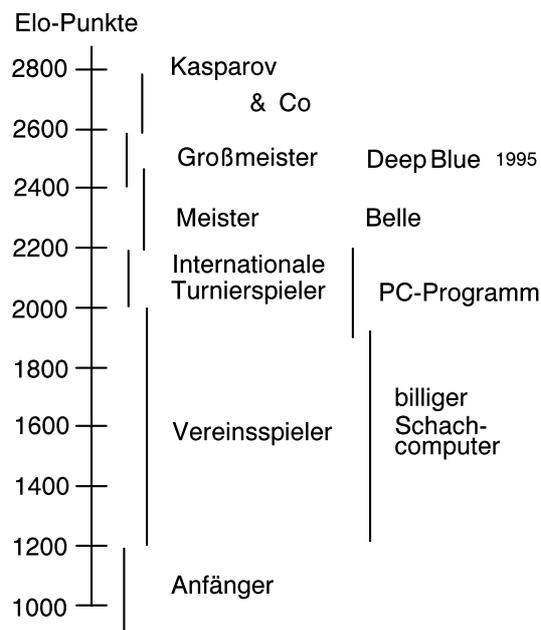
Inzwischen gibt es auch einen Schach-Benchmark, eine Zahl, die angibt, wie gut ein Spieler ist. Er wurde von dem amerikanischen Professor Arpad E. Elo entwickelt, die Kennzahl heißt deshalb auch Elo. Es ist ein statistisches

Verfahren, bei dem die Stärke der Spieler anhand des Ergebnisses der Turnierspiele ermittelt wird. Seit 1972 benutzt, hat sich dieses Verfahren als sehr, sehr zuverlässig erwiesen. Insbesondere kann man über diese neutrale Kennzahl Mensch und Maschine direkt vergleichen.

Die Anfänge

In den USA begann der bekannte Computerpionier Claude Shannon ab 1950 die Grundlagen für Schachcomputer auszuarbeiten. Er definierte die A-Strategie als das naheliegende brute-force-Verfahren: alle möglichen Varianten werden durchgerechnet. In jedem Zustand des Spiels sind 20 - 40 Züge möglich, typisch etwa 35. Um zwei Halbzüge tief zu rechnen sind damit 35^2 , also 1225 Stellungen zu bewerten. Für 4 Halbzüge sind es schon 1,5 Millionen. Die Zahl der Kombination eskaliert. Grafisch wird das Problem als Baumstruktur dargestellt. Sind in einem

Augenblick 35 neue Züge möglich, verzweigen von diesem Punkt 35 Äste. Will man 5 Halbzüge tief durchrechnen, hat der Baum 5 Knotenebenen. Da der Suchbaum sehr schnell in die Breite geht, kann man ihn nur schwer in die Tiefe durchsuchen. Shannon zog daraus den Schluß, daß die A-Strategie undurchführbar ist und verlegte sich auf die B-Strategie. Hier werden anhand von Bewertungskriterien die wenigen sinnvollen Züge bestimmt. Und



nur diese werden weiterverfolgt. Zumindest als Begriff gibt es auch noch die C-Strategie. Sie würde sich dem menschlichen Denken annähern. Das könnte z.B. ein Programm sein, das eine strategische Wunschsituation formuliert und dann auf sie hinarbeitet. Jedoch hat dieser Ansatz bisher keine praktische Bedeutung.

Das Dilemma

Die reine B-Strategie hat sich in der Praxis nicht bewährt. Die gründliche Bewertung eines Halbzugs kann bis zu 50 Kriterien benötigen. Kriterien können Materialbilanz, Positionsvorteil, Beweglichkeit, Sicherheit, Bauernstruktur, Feldbeherrschung u.a. sein. Derartige Berechnungen sind jedoch zeitraubend, wodurch auch hier nicht unmittelbar die gewünschte Suchtiefe im Baum erreicht wird. Zudem sind diese Faustformeln etwas willkürlich und schematisch. Sie sind nur eine grobe Annäherung an die Intuition des menschlichen Spielers.

1958 verbesserte man an der Carnegie-Mellon-Universität das Verfahren deshalb um den „Alpha-Beta-Algorithmus“. Dabei merkt man sich beim Suchen den bisher besten Zug. Sobald sich herausstellt, daß der Halbzug den man gerade durchrechnet, schlechter ist als dieser beste Zug, ignoriert man den ganzen Zweig („Alpha-Cut“). Beim Gegenzug, der Vorhersage der Antwort des Gegners, geht man genauso vor („Beta-Cut“). Der Gegner würde einen schlechten Zug ohnehin nicht ausführen. Die Zahl der untersuchten Züge reduziert sich dabei drastisch von n auf $2 \cdot (n^{0,5})$. Also z.B. von 1 Mio Züge auf 2000. Damit wird der Vorstoß in die Tiefe möglich.

Basierend auf B-Strategie entwickelten ab 1968 die Studenten Larry Atkin und David Slate ihr Schachprogramm CHESS. Die Version 3.6 war 1972 stark genug mehrere Computerschachwettbewerbe zu gewinnen. Allerdings noch zu schwach, um eine ernste Herausforderung für gute menschliche Spieler zu sein. 1973 beschlossen Atkin und Slate, daß der Weg vorwärts ein Schritt zurück zur A-Strategie war. Sie reduzierten die Bewertung drastisch im wesentlichen auf das Kriterium Materialbilanz. Also Gewinn und Verlust von Figuren. CHESS 4.5. gewann ab 1975 Computerschachwettbewerbe in Reihe. Version 4.7 verlor zwar 1978 das Turnier gegen den schottischen Meister Levy, gewann aber zumindest eine Partie. Worauf die Experten CHESS 4.7 etwas optimistisch eine Elo-Wertung von etwa 2200 zuerkannten. Einstweilen blieben Computer aber bei der Berechnung auf eine Tiefe von bestenfalls 7 Halbzügen beschränkt. Das genügt, 99% aller menschlichen Spieler zu schlagen, kann aber einen Großmeister nur selten beeindrucken.

Brute Force triumphiert

1980 wurde an den Bell Labs von Ken Thomson und Joe Condon der Schachcomputer BELLE fertiggestellt. Die Bezeichnung Schachcomputer ist hier treffend. Es wurde ein Spezialrechner gebaut, der Teile des Algorithmus direkt in Hardware realisiert. Der Rest wurde in C programmiert (Hätten Sie von Ken Thomson etwas anderes erwartet?). Ziel waren 1 Mio Stellungen pro Sekunde zu berechnen, was 10 Halbzüge bedeutet hätte. Erreicht wurden nur 150k. Das genügt für 8 Halbzüge, ausgewählte Teile können bis auf 10 Halbzüge durchgerechnet werden. Es war aber ein enormer Sprung nach vorn, weil CHESS 4.9 damals auf

konventionellen Mainframes nur 5k Stellungen/sec erreichte. Heute macht der handelsübliche Pentium etwa 120k und hat damit BELLE erreicht. Erwartungsgemäß gewann BELLE damals über Jahre hinweg Computerschachwettbewerbe und setzte internationalen Meistern hart zu.

Doch auch konventionelle Hardware wurde schneller. Robert Hyatt arbeitete seit 1975 an der Universität von Mississippi daran, der Cray 1 das Spielen beizubringen. 1981 wurde er als Gag zu den offenen Meisterschaften von Mississippi eingeladen. Er gewann alle Partien. Die Veranstalter waren schlechte Verlierer. Sie erklärten den Zweitplazierten, einen Menschen, zum Sieger. BELLE war ein härterer Gegner. Erst nachdem die Software auf Assembler optimiert wurde und man auf einen X-MP als neue Hardware überging, konnte man sie 1983 besiegen.

Es gibt auch weiterhin leistungsfähige Parallelrechner mit konventioneller Hardware, die Schach spielen. Der „Paragon“ von Intel enthält 1800 Knoten die mit i860-CPU's bestückt sind. Auf ihm erreicht das Schachprogramm *Socrates 2 Mio Stellungen/sec. Er ist sicherlich gut genug manchmal einem Großmeister ein Bein zu stellen, aber keine Gefahr für Kasparov.

Gegen ihn braucht man wieder einen Spezialrechner. An der Carnegie-Mellon-Universität wurde 1987 die nächste Generation gebaut. „Deep Thought“ basierte auf kundenspezifischen VLSI-Chips und erreichte bereits 0,5 Mio Stellungen/sec. Die Entwicklungsgruppe wechselte daraufhin zu IBM. „Deep Thought II“ erreichte 1994 mit 24 Prozessoren bereits 7 Mio. Die nächste Version „Deep Blue“ sollte mit 1000 Prozessoren 1 Milliarde erreichen. Was einer Tiefe von 14 - 15 Halbzügen entsprechen hätte. Real wurden damit bisher 200 Mio/sec erreicht. Gegen diese Maschine verlor Kasparov im Februar 1996 erstmals eine Partie unter Turnierbedingungen. Gewann das Turnier dann allerdings doch noch recht überzeugend. Er ist noch einmal davongekommen.

Der Gedanke, daß es noch einen menschlichen Spieler gibt, der nicht gegen den Computer verliert, ist für die Wissenschaftler unerträglich. Es handelt sich hier nicht um einige „mad scientists“, sondern um das gesamte wissenschaftliche Establishment. Die renommiertesten Konzerne und Universitäten stehen mit Ressourcen und vollem finanziellen Einsatz hinter der Jagd. 1991 gaben Experten Schätzungen ab, wann sie zuende ist. Sie reichen von 1992 bis 2040. Kasparov meinte 2010 „oder vielleicht nie“.

Man braucht jedoch weder „Deep Blue“ noch Pentium, um Amateure plattzumachen. Ein typischer Mephisto-Schachcomputer Baujahr 1985 hatte eine 8-bit 6301-CPU mit 2MHz-Busfrequenz. Der Speicher beschränkte sich auf 16k ROM und 256 Byte RAM. Gegen Anfänger genügt das völlig.

Zwischen der Hardware der kleinen und großen Maschinen liegen Welten. Die Software ist jedoch sehr ähnlich. Die Leistungssteigerung bei Schachcomputern ist weniger auf verbesserte Algorithmen, sondern hauptsächlich auf schnellere Hardware zurückzuführen.

Literatur:

[1] Steinweder, Friedel „Schach am PC“

[2] Obermann „Schachcomputer Report 84“

[3] Produktvergleich von Stiftung Warentest in „test“ 12/93

q



**Dieter Steinwender,
Frederic Friedel:
Schach am PC**

*Markt&Technik, 1995
ca. 500 Seiten, mit CD
ISBN 3-87791-522-1
69,00 DM*

Mit seinen 500 Seiten ist das Buch sozusagen der Zech für Computerschach. Im ersten Teil wird die historische Entwicklung geschildert. Im zweiten Teil werden die 20 wichtigsten kommerziellen Schachprogramme besprochen sowie 10 PD- und Sharewareprogramme, die sich dann auch auf der beiliegenden CD befinden. Teil 3 vermittelt dem Spieler Hintergrundwissen, damit er besser gegen die Programme ankommt. In Teil 4 sind die Programmierer gefordert: wie schreibt man ein Schachprogramm? Mit Source für ein simples Programm in BASIC. Auf der CD befindet sich zusätzlich Source für ein starkes Programm in C. Das Lexikon in Teil 5 erläutert die wichtigsten Begriffe.



Und was lernt man daraus?

Die Liebhaber von Schachcomputern sind eine Gruppe, die ähnlich klein und unbekannt wie FORTH ist. Aber offensichtlich etwas konstruktiver und kreativer wenn's ums Lösen ihrer Probleme geht.

Sie haben im M&T-Verlag ein informatives, gutes Buch herausgebracht. Es ist einfach genug geschrieben um viele Leute anzusprechen. Und hat genug Tiefgang, um seinem Besitzer langfristig als Nachschlagewerk zu dienen. Bei den in deutsch erschienenen FORTH-Büchern haben nur Zech und Brodie dieses Prädikat verdient. Der Rest ist recht schnell wieder vom Markt verschwunden.

Dadurch, daß das Buch in einem etablierten Verlag herauskommt, hat es schon einen Vertriebsweg eingebaut, mit dem die Öffentlichkeit tatsächlich erreicht wird. Im Gegensatz dazu spricht die FORTH-"Weihnachts"-CD ohne Buch kein sehr breites Publikum an, weil die Papierdokumentation zu ihrem Inhalt fehlt. Und sie hat nicht das Absatzsystem eines etablierten Verlags.

Es ist ein gutes Stück Arbeit ein Buch mit 500 Seiten zu schreiben. Die Autoren Steinweder und Friedel geben aber seit längerem die einschlägige Zeitschrift CSS heraus und konnten deshalb Material ansammeln. Nominell wäre es

analog dazu möglich aus einer konsequent geplanten Artikelserie in der VD ein Buch zusammenzusetzen.

Die Schach-Autoren haben einen klaren Schwerpunkt gesetzt: Schach als Software auf IBM-PCs. Schachhardware a la Mephisto wurde bewußt nicht mehr behandelt. Für FORTH wird jedoch in der VD munter behauptet, es wäre auf jeder CPU von MARC4 bis DEC's Alpha geeignet. Und nicht nur für embedded Controller, sondern auch für Tischcomputer ideal. Ein FORTH-Buch, das keinen klaren Schwerpunkt, z.B. embedded Controller, setzt, kann auch nicht auf ein bestimmtes Publikum ausgerichtet werden. Und paßt damit grundsätzlich nicht in das Vertriebssystem eines Verlags.

Letztlich braucht ein derartiges Unternehmen einen etwas ernsthafteren Ansporn als "die Liebe zu FORTH". Autorenhonorar ist uninteressant, weil niedrig. Die Schachcomputer-Autoren haben jedoch in Hamburg ein kleines Geschäft (mit Versandservice), das als Spezialvertrieb für derartige Produkte bekannt ist und auf das auch im Buch hingewiesen wird. Das Buch wird dort den Umsatz ankurbeln. Damit liegen hier besonders günstige Rahmenbedingungen vor: der Distributor kann mit dem Buch eine breite Palette von Produkten unterschiedlicher Hersteller fördern. Er wird als Händler immer verdienen, da durch ihn bestellt wird. Für FORTH existiert weder die Produktpalette, noch eine zentralisierte Distribution.

Die 64 Constant-Falle im ZF

von Martin Bitter

Möllenkampweg 1A; D-46499 Hamminkeln (Mehrhoog)

Stichworte: ZF, F-PC, HEX

Auf meinem Schreibtisch wurde es immer enger und das Herumhantieren mit zwei inkompatiblen Tastaturen für den PC und den ATARI ging mir langsam auf die Nerven. Eine Softwarelösung mußte her: Der PC emuliert nun bei mir die ATARI-Tastatur (ist nicht so einfach wie es sich anhört!).

Im Rahmen der Programmentwicklung wurde eine Tabelle mit dem Namen scan_tab angelegt ...

```
Create Scan_tab      256 allot
```

gefüllt ...

```
Scan_tab 256 fill_tab
```

fill_tab schreibt n fortlaufende Bytes. Einzelnen Scancodes der PC-Tastatur wurden die Scancodes der ATARI-Tastatur mit dem Wort

```
pc>Atari ( n1 n2 — )
```

zugewiesen. (PC>ATARI trägt unter anderem an die Stelle n1 der Tabelle SCAN_TAB den Wert n2 ein.)

Das funktionierte für alle Scancodes - bis auf zwei Fälle. Dort standen in der Tabelle urplötzlich falsche Werte. (Ganz schlimm war, daß dadurch die Tastenkombination zum Abbruch einer Endlosschleife nicht erkannt wurde.)

Wo lag der Fehler? Zum Selbstdenken hier der betreffende Quellcode:

```
base @ hex
...
...
64 01 pc>Atari
...
...
01 64 pc>Atari
...
base !
```

Des Rätsels Lösung: It's not a bug it's a feature!

Tom Zimmer hat bei seinem Forth im KERNEL86.SEQ folgendes geschrieben:

```
64 Constant 64
```

Nun ist alles klar! "64" wird vom Forth-Interpreter als Wort erkannt und ruft das dezimale "Literal 64" auf. Das bedeutet: unabhängig von der gewählten Basis hat die Ziffernfolge 64 immer den dezimalen Wert 64:

```
hex 64 ,
```

trägt den HEX-Wert 40 ein!

Ausweg: man verändert Kernel86.seq entsprechend und kompiliert neu.

oder: man schreibe

```
hex 064 ,
```

das wird nicht als Wort 64 erkannt und als Hex-Wert 64 (dezimal 100) eingetragen.

oder: man schaltet um in dezimale Darstellung

```
decimal 100 , hex
```

oder: man schreibt als Summe

```
hex 60 4 + ,
```

oder: man verwendet ein Number, das Präfixe verwendet

```
$64 ,
```

Ein solches Number, das die Präfixe %, &, \$, h für binäre, dezimale und hexadezimale Ziffern erkennt, gibt es in der MHB-Moers oder bei der VD [1].

Mir hat dieses kleine, süße, überraschende Feature einige Stunden voller Selbstzweifel und mit einer Menge bitterlicher und forthlicher Huh!s beschert. Im ZF finden sich im Kernel noch folgende Konstanten :

```
0 Constant 0
1 Constant 1
2 Constant 2
3 Constant 3
```

[1] NEW_NUMB.SEQ. Erhältlich z.B. auf der Listingdiskette zu VD 3/1996: VDL963.ZIP

q

F-PC krisensicher!

Ins F-PC hat Tom Zimmer das geschilderte 'Problemchen' freundlicherweise nicht eingebaut. Hier hat er genau den entgegengesetzten Weg gewählt. Wo andere Systeme zumindest die gängigen Konstanten '0', '1', '2' als vordefinierte Wörter im Wortschatz haben, besitzt das typische F-PC-System keine einzige vordefinierte Konstante, deren Name eine Zahl zwischen '0' und '9' enthält und die als

Decimalzahl (fehl)interpretiert werden könnte. Die Frage, ob diese Sicherheit auch für Hexzahlen gilt und ob nicht eines der zahlreichen zuladbaren Files eine Falle birgt, überlasse ich dem Forschungsdrang der LeserInnen. Ich werde jedenfalls Hexzahlen lieber weiter mit '\$' beginnen lassen, Zahlen wie 'FF' kollidieren doch allzuleicht mit einem Kurzwort für FormFeed in irgendeiner Druckutility.

clv



Forth International

von Fred Behringer



Forth in Amerika, Rochester woanders

Die Rochester Forth Conference fand bisher immer in Rochester, New York, statt. Diesmal nicht. Diesmal wurde sie zum ersten Mal woanders abgehalten. In Toronto, Canada. Nicholas Solntseff aus Toronto gibt in der September-Ausgabe der "Forth Dimensions" einen kurzen Bericht darüber. Ich versuche, über diesen Bericht zu berichten.

Drei wichtige Konferenzen werden im Zusammenhang mit Forth genannt: FORML in Asilomar, California, euro-Forth in verschiedenen Städten Europas und eben die Rochester Forth Conference. Der Autor ist der Meinung, dass Forth nun einen Entwicklungsstand erreicht hat, der die Welt nach Forth greifen lässt. "Die Welt braucht Forth" ... Mitch Bradleys Open Firmware, Suns Java-Development-System, Bernard Hodsons Software-Gene und Europay, ein Zusammenschluss von europäischen Geldtransaktionsdiensten, verwenden systemübergreifende Software, die im Endeffekt auf einer abstrakten Byte-Code-Maschine, einer virtuellen Maschine mit Befehlscode im Byteformat, läuft.

Das ist der rote Faden, der auf der diesjährigen Tagung in Toronto entdeckt wurde. Und zum Thema für die nächste Tagung, 1997, wurde folgerichtig "Abstrakte Maschinen" gewählt.

Schließlich erwähnt der Berichterstatter noch die enge Verwandtschaft von Java mit Forth und berichtet von einer Working-Group, die sich auf der Konferenz mit diesem Thema beschäftigt hat.

Apropos: Claus Vogt fragt mich, den Bericht-Berichterstatter, was es denn eigentlich mit Europay auf sich habe. Ich weiß nichts. Weiß jemand mehr?

Apropos 2: Mit der Aufgabe betraut, ein paar Zeilen zu dem Rochester-Tagungsbericht zu schreiben, und der Entdeckung, dass sich eine ganze "Working-Group" mit Java und Forth beschäftigte, kommt mir ein Artikel aus der iX in Erinnerung: Rainer Fischbach, "Java: Programmiersprache der Zukunft? Kalter Kaffee", iX 10/96, S.85 ff. Der Tenor des Artikels: Alles schon dagewesen. Java nichts Neues. Folgende Programmiersprachen werden als Beispiel angeführt: FORTRAN, UCSD-

Pascal, UCSD-Basic, Modula-2, Smalltalk-80, EUMEL, C, C++, Eiffel, Clu und Theta. --- Forth ist **nicht** dabei ! --- Überhaupt: Java von SUN? Waren die SUN-Workstations nicht jene, die mit Forth-Meldungen aus dem Kern auf dem Bildschirm überraschten, wenn das System zusammenbrach? Oder irre ich mich?

Fred Behringer, München

Ich möchte in dieser Rubrik 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.

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

Forth Dimensions der Forth Interest Group, USA

September/Oktober 1996

6 Development Aids for New Micros *Richard W. Fergus*

Der Autor verwendet die bekannten New-Micros-Produkte, wollte aber die mitgelieferten Zusatzeinrichtungen soweit ausbauen, dass sein System einen wesentlich erweiterten Eindruck macht. Er vermittelt uns, wie er das Download-Problem und die Terminal-Emulation anpackt, Variablen gesondert im RAM, Motorola-S19-Record-Ausgabe, Brodies MAKE-DOER, Multitasking und eine Logging-Funktion im Terminal-Emulationsmodus zum Kopieren der S19-Records.

11 The Elephant Who Refuses to Be Bagged *C.H. Ting*

Beim Versuch, einen Kompressionsalgorithmus in Forth zu verwenden, über den auf einer Konferenz berichtet worden war, ergaben sich aufgrund der neuen Daten neue Herausforderungen. Das bereitgestellte Programm gibt einen Einblick in den dornenreichen Weg von der Entdeckung über die Analyse zur Lösung.

18 4tH, an Experiment in C *Hans Bezemer*

4tH ist anders als andere Forth-Implementationen. Es enthält viel Forth, ist aber nach C übertragen. Wohl hat auch 4tH zwei Interpreter. Die aber arbeiten ganz anders. 4tH ist Token-gefädelt, aber ohne Dictionary. Es ist ein Forth, das die übliche Compiler-Technologie verwendet, jedoch keinen Standalone-Compiler - es ist eine Bibliothek. Ein wesentliches Ergebnis: 4tH erzeugt Byte-Code, ähnlich wie Java, der ohne jegliche Änderung zu jedem beliebigen

System portiert werden kann.

31 FIG Board Increases Member Benefits *Elizabeth Rather*

Der Vorstand der Forth Interest Group traf sich auf der diesjährigen Rochester Forth Conference. Es war die zweite offizielle Zusammenkunft des zur Zeit amtierenden Vorstands. Unter anderen wurden folgende Punkte behandelt: Zusätzliche Vergünstigungen für FIG-Mitglieder, Klärung von Fragen der Vergünstigung für angegliederte Körperschaften, Neufassung der Anzeigensätze für die "Forth Dimensions", Ankurbeln einer Werbekampagne und Besprechung von Plänen zur Neugestaltung des FIG-Forthbüros und seiner Aktivitäten.

33 Rochester Forth Conference *Nick Solntseff*

Auf Canadas erster Forth-Konferenz wurde ein breites Spektrum von Arbeiten präsentiert, die aktuelle Probleme widerspiegeln. "Rochester-in-Toronto" war das erste Mal, dass die weithin geschätzte Rochester Forth Conference außerhalb New Yorks veranstaltet wurde. Forth ist kraftvoll und lebensstüchtig, wofür, wie der Berichtersteller bemerkt, reichlich Zeichen zu sehen waren. □

Gehaltvolles

zusammengestellt und übertragen
von Fred Behringer

Forth Dimensions der Forth Interest Group, USA

Januar/Februar 1996

5 Scattering a Colon Definition *M.L. Gassanenko*

Der in Präfix-Notation gehaltene Forth-Assembler des Autors erledigt einige Initialisierungsarbeiten, bevor er einen neuen Befehl verarbeitet. Die Schalter des Assemblers werden entsprechend der Voreinstellungen und nach Maßgabe des jeweiligen Befehlsoperanden gesetzt und legen fest, was geschehen soll. Bestimmte Schalter sorgen dafür, dass ein Forth-Wort ausgeführt wird. Das Problem bestand darin, dass die Initialisierungsaktionen gleichzeitig zu zwei ver-

schiedenen Modulen gehören: zum Modul, das sie initialisieren, und zum allgemeinen Initialisierungsmodul. Der Autor wollte diese Aktionen so verteilen, dass sie in den Modulen, die sie initialisieren, zu liegen kommen, aber nur aus einer einzigen Definition bestehen.

8 Mobile Computing in Brazil *Klaus Blass*

Brasilien rühmt sich, das achtgrößte Wirtschaftsvolumen der Welt zu haben, und die Importe strömen nun ungehindert ins Land, darunter auch Computer-Hardware. Die ansässigen Firmen bemühen sich um eine Automatisierung der Verkaufsförderung, wobei sie immer mehr nicht-ortsgebundene Computer einsetzen, mit Programmen der Art, wie sie die Firma des Autors in Forth entwickelt. Andere Firmen bemühen sich um denselben Markt, mit Forth lassen sich aber beschränkte Ressourcen maximal ausnützen. Forth packt normalerweise viermal so viel an Funktionen in ein ausführbares Programm, dessen Länge nur die Hälfte bis zu einem Viertel des C-Programms eines Konkurrenten beträgt.

18 FORML 1995 *András Zsótér*

Und wieder einmal sind unerschrockene Forth-Praktiker zusammengeströmt, um ihre jüngsten Arbeiten zu repräsentieren, um zwanglos neue Ideen auszutauschen und um Übereinstimmung in technischen, organisatorischen und politischen Fragen zu suchen, die die Forth-Gemeinschaft betreffen. In Pacific Grove, Kalifornien. Unser Reporter aus Hong-Kong berichtet über seine ersten Erfahrungen auf der diesjährigen FORML-Konferenz.

19 Stepper Motors *Skip Carter*

Mit diesem Artikel wird die neue Kolumne "Forthware" eingeleitet, die sich mit Forth bei Echtzeitproblemen beschäftigen soll. Hier soll die Steuerung von Motoren der verschiedensten Art untersucht werden. Diesmal wird die Verwendung der parallelen Schnittstelle des PCs zur Steuerung von Schrittmotoren besprochen, wobei von der Vorstellung ausgegangen wird, man arbeite mit einer mikroprozessor-gesteuerten Regelungsschaltung, und wo die parallele Schnittstelle des PCs die digitalen E/A-Kanäle des Reglers vertritt. □

Über Eakers CASE in Forth und das IF-Konstrukt in OCCAM2

von Fred Behringer
Planegger Str. 24; D-81241 München

Stichworte: Transputer OCCAM Assembler

Überlegung

Ich möchte gleich zur Sache kommen. Eine Begründung für meine Beschäftigung mit dem im Titel angeführten Thema steht weiter unten.

Das CASE-Konstrukt von Eaker in Forth geht sämtliche OF..ENDOF-Zeilen durch. Sobald eine erfüllte OF-Bedingung angetroffen wird, werden die Anweisungen zwischen OF und ENDOF ausgeführt und das CASE-Konstrukt wird verlassen.

Das IF-Konstrukt in OCCAM2 hat, vereinfacht gesehen, eine ähnliche Wirkung. Sobald eine Bedingung (als solche an der Einrückung um zwei Zeichen zu erkennen) erfüllt ist, werden die an der betreffenden Stelle stehenden (wiederum um zwei Zeichen eingerückten) Anweisungen ausgeführt und das IF-Konstrukt wird verlassen.

Eine Ausnahme, eine wesentliche allerdings, besteht jedoch: Beim IF-Konstrukt in OCCAM2 muss mindestens eine Bedingung erfüllt werden. Sonst hängt sich das IF-Konstrukt auf. Das ist der Grund, weshalb alle (guten) OCCAM2-Programme in den IF-Konstrukten als abschließende Bedingung ein TRUE haben. Die TRUE-Bedingung ist immer erfüllt und klärt die eventuell übrig bleibende Restabfrage, wenn nämlich die davor stehenden eigentlich beabsichtigten Bedingungen aus irgendeinem Grunde nicht greifen.

Was beim CASE-Konstrukt von Eaker in Forth auf den ersten Blick als Vorteil (gegenüber dem IF bei OCCAM2) erscheint (kein abschließendes TRUE nötig), kann aber auch als Nachteil ausgelegt werden: Gelegentlich möchte man (beim CASE von Eaker) eine letzte Abfragezeile haben, die dann (aber auch wirklich nur dann) ausgeführt wird, wenn keine der zuvor abgefragten OF-Bedingungen greift. Das folgende Beispiel bewerkstelligt das:

```
CASE
  1   CF   ." 1"           ENDOF
  2   CF   ." 2"           ENDOF
  3   CF   ." 3"           ENDOF
  DUP CF   ." >3 oder <1" ENDOF
ENDCASE
```

Erklärung

Die 2 vor dem OF in der zweiten Abfragezeile, beispielsweise, bewirkt, dass eine 2 auf den Stack gelegt wird. Das OF prüft dann auf Gleichheit mit der in das CASE-Konstrukt eingegebenen Zahl. Das DUP in der letzten Abfragezeile, andererseits, bewirkt, dass eine Kopie der eingegebenen Zahl auf den Stack gelegt wird. Die anschließend über OF erfolgende Prüfung auf Gleichheit der Eingabezahl mit eben dieser Kopie endet natürlich immer mit einer Bejahung. Als Kuriosum ist noch anzumerken: Belegt man die Zeile, in der DUP vorkommt, mit der „leeren“ Anweisung, so ist die betreffende Zeile weiter nichts als einfach nur redundant.

Beweggrund

Seit dem Aufkommen von Computern wird über die automatische Übersetzung von Sprachen nachgedacht. Bei natürlichen Sprachen ist man immer noch nicht geneigt, Programmen zu trauen. Man übersetzt selbst. Bei Computersprachen ist das anders. Man ruft nach einem Compiler oder Cross-Compiler. Beim Einbau von Fließkomma-Ergänzungen nach dem ANS-Standard in mein Transputer-Forth-System wollte ich OCCAM2-Programmteile zur Verwendung in einer Code-Definition in Transputer-Forth-Assembler (eigene Création) übersetzen. Dafür gibt es noch keinen (mir zugänglichen) Cross-Compiler. Also entschloss ich mich, das per Hand zu machen. Der Aufwand ist gerechtfertigt, da das in meinem Leben (höchstwahrscheinlich) nur einmal geschieht und keinesfalls zur Dauereinrichtung wird. Mein Transputer-Forth-Assembler ist, wie üblich, strukturiert (IF, ELSE, THEN, BEGIN, UNTIL usw.). OCCAM2 ist sehr maschinennah, aber eben doch nicht so sehr, dass nicht noch Überlegungen nötig wären. Was ich eingangs beschrieben habe, ist ein Ergebnis solcher Überlegungen.

9



From the other side of The Big Teich



Henry Vinerts

v.vinerts@genie.com

Silicon Valley Chapter, 24.8.96.

Dear Friederich and Claus,

I know that one of you is on vacation and the other one may not be expecting a letter from me, but here it is anyway. (I did send a copy to Wolf yesterday.) I thought that you might be interested in what is going on in the Northern California Forth community. (Incidentally, one of the attendees flew in from Portland, Oregon, some 650 miles from here, as he has done for quite a few months now.) A report on Silicon Valley Forth Interest Group Meeting, August 24th, 1996.

This was the second time at Cogswell College in Sunnyvale, California, next-door to Lockheed-Martin. We are glad to have a place to meet. I'd say some 35 to 40 people came during the 6-hour period, averaging 25 to 30 at any one time. Dr. Ting started the meeting with a report on his progress of booting to Forth on 486 motherboards that are tethered to a 486 PC. He has found a way to do it through the video ROM. A very good explanation and lecture to those who understood and were interested.

Then came Jeff Fox with an update on the progress at iTV, the company that is making a box to sit on top of a TV, available for much less than a PC, making it possible with the addition of keyboard and modem to get on the world-wide Web and to do e-mail and other things that currently require a much larger investment in hardware. This is where Chuck Moore's latest microprocessors and OK language can do their thing.

The highlight of the day was Steve Robert's BEHEMOTH, a bicycle with over a million dollars of high-tech equipment aboard. This is something unique, and Steve certainly has a knack of entertaining the audience. He brought the bicycle and the trailer that is hooked up to it. It would take quite a few pages to describe the highlights, so I'll just mention a few. First of all, Steve has appeared on Phil Donahue show with the BEHEMOTH and his stories; I suppose it is possible that TV viewers outside the U.S.A. have seen it. The bike is pedaled by human (Steve's) power, but it has solar panels and batteries in the trailer for feeding the computer equipment. Steve wears a specially equipped police helmet when he pedals. The helmet has attachments which include miniature screens for a MacIntosh and, if I heard correctly, also a PC, so that, using the keyboard(s) attached to the bicycle handlebars,

Steve can do his writing, as he travels the countryside. There is stereo music, for relaxation and inspiration; there is a ham radio station with a collapsible 20-foot dipole. The security system on the bike surpasses that of the Denver Airport. The bike "talks" to GPS and before anyone can move it from the GPS coordinates that it is parked to, the BEHEMOTH will itself dial 911 to report that it is being stolen. There is a voice-synthesizer on board, which Steve can control remotely, as well as program to suit the needs. Needless to say, a talking bicycle can initiate some pretty interesting conversations and draw different reactions in the various cultural climates of the country.

After some 17000 miles (in 1700 hours, did he say?) Steve decided to move to a different conveyance--a boat--one that does not require pedaling effort. So at present, with help from Apple Computers in Cupertino, he is busy equipping a boat with the latest in computer technology. As far as I heard, the bicycle has three 63HC11s, all running New Micro Forths, the boat will have more Forth.

To finish the meeting, there was some time to discuss the forthcoming Embedded Systems conference and what FIG could present in the booth that has been made available to the group.

Skip Carter had brought his six-legged robot, which is still a baby, less than a year old, but is evolving faster than any Darwinian creature. Skip says that he will be entering it in a

*This is where Chuck Moore's latest
microprocessors and OK language can do*

robot labyrinth contest at the San Francisco Exploratorium on September 15th. I'd like to see that. The weekend before the SVFIG meeting my wife and I went to ROBOT WARS, also in San Francisco, which is in its third year now and features 4 weight classes of robots, that are out there to do what we call "blood and guts" battles to amuse aficionados of various forms of technological violence. Unfortunately, most of these "creatures" are radio-controlled; there were only 5 autonomous entries, and we did not get to see any of them. Actually, we went mainly because one of my SVFIG acquaintances had entered his "Terminal Frenzy" which is programmed with QED Forth, but his opposition was not ready at the scheduled time when we were there.

So long for now!

11.9.96

Our Forth group is going to have a booth at the San Jose Embedded Systems Conference. I hope that I'll be able to take a day off work to attend. I imagine that Klaus Flesch will show, as he has for the past two years. I'll try to give you more reports of what I see in the Forth world on this side of the big Teich.

*Cheers,
-- Henry*

Intelhex-Format

Bernd Beuster (B.Beuster@bbepoint.FORTH-ev.de)
Carl-Benz-Str. 1A; D-55131 Mainz

Erzeugung des INTELHEX-Formats für die Programmierung von EPROMs. Ausgabe entweder auf Bildschirm oder in eine Datei.

Stichworte: Intelhex Datenübertragung EPROM

Sinn und Zweck

Für einen Mikrocomputer soll ein Forthassembler verwendet werden. Dabei erfolgt die Codegenerierung im Speicher des PC. Die Programmierung der EPROMs macht es notwendig, die Daten in einem kompatiblen Format zu übergeben. Dabei wurde das Intelhex-Format verwendet, welches die meisten EPROM-Programmierprogramme verwenden können.

Beschreibung des Intelhex-Formates

Jeder Block beginnt auf einer neuen Zeile mit einem Doppelpunkt. Die nächsten zwei hexadezimalen Ziffern geben die Blocklänge an. Bei Ausgabe eines größeren Speicherbereiches haben die Blöcke eine Länge von 16 Bytes (es sind auch andere Längen möglich). Die folgenden vier Ziffern bezeichnen die Anfangsadresse des Datenblocks, gefolgt von zwei Nullziffern und den Datenbytes.

Nach den Datenbytes steht die Prüfsumme, die sich aus der Summe aller Bytes einer Zeile berechnet. Es wird das Zweierkomplement gebildet und das Ergebnis auf ein Byte abgeschnitten. Beim Einlesen werden alle Bytes addiert und es ergibt sich automatisch das 8-Bit-Ergebnis Null.

Programm

Das Programm wurde unter einem Eigenbau 32-Bit OS/2-Forth getestet. Bei 16-Bit Systemen ist zu beachten, daß maximal (64 K-1) Byte geschrieben werden können, was aber für den obigen Anwendungsfall ausreichend ist.

BASE-ADDRESS enthält die 16-Bit Basisadresse des zu programmierenden EPROMs, meistens 0000H.

INTELHEX erwartet auf dem Stack die Adresse des auszugebenen Codes sowie dessen Länge in Bytes.

WRITE-INTELHEX erwartet zusätzlich noch die Stringadresse der Ausgabedatei. In der verwendeten Forthversion gibt es zwei Worte, welche nicht ANS-konform sind:

?DOS-ERROR (flag -) kann durch ABORT"" ersetzt werden.

OUTPUT-HANDLE enthält das Dateihandle für die Ausgabe. Die Umleitung der Dateiausgabe in eine Datei muß dem jeweiligen System angepaßt werden.

q

File: intelhex.f vom 28.8.1996

```

1)~ INTEL-HEX-FILE Output
2
3 ONLY FORTH DEFINITIONS HEX
4
5 MARKER EMPTY-INTELHEX
6
7
8 VARIABLE BASE-ADDRESS |~ base address for EPROM
9 10 CONSTANT/LINE |~ bytes/line
10
11
12~ print exactly 2 or 4 digit
13: U.2 ( u—) 0<#####> TYPE;
14: U.4 ( u—) 0<#####> TYPE;
15
16~ write a line
17:(INTELHEX) ( addr1 u— addr2)
18~ header
19 CR [CHAR]: EMIT
20 DUP U.2
21 BASE-ADDRESS @
22 DUP 0 100 U/MOD + >R ( RS: checksum)
23 U.4
24 DUP BASE-ADDRESS +!
25 0 U.2
26~ body
27 DUP R+ -ROT ( checksum addr1 n)
28 0 DO COUNT DUP U.2 ROT + SWAP LOOP
29~ tail
30 ( addr2 checksum) SWAP NEGATE OFF AND U.2;
31
32

```



```

33
34
35~ print hex table to console
36: INTELHEX ( addr u—)
37 BASE-ADDRESS @ BASE @ >R HEX
|~ save address and base
38 0 /LINE U/MOD |~ calculate# lines
39 ROT SWAP 0?DO /LINE (INTELHEX) LOOP
|~ write full lines
40 SWAP ?DUP IF (INTELHEX) THEN DROP
|~ write last line
41 2R> BASE! BASE-ADDRESS!;
|~ restor address and base
42
43
44~ write hex table to file
45: WRITE-INTELHEX ( addr u1 c-addr u2—)
46 W/O CREATE-FILE ?DOS-ERROR DUP >R |~create file
47 OUTPUT-HANDLE @ >R OUTPUT-HANDLE! |~ save and
redirect output handle
48 INTELHEX |~ do it!
49 R> OUTPUT-HANDLE! |~ restore output handle
50 R> CLOSE-FILE ?DOS-ERROR; |~ close file
51
52 DECIMAL
53
54
55
56 EXIT
57
58 (***) Test (***)
59 Defines a table with Z80 opcodes at address 1400h.
60 Usage: TEST-TABLE INTELHEX
61 or
62 TEST-TABLE S" test.hex" WRITE-INTELHEX)

```

```

63
64: 2CONSTANT ( name( n1 n2—) CREATE,, DOES> (— n1
n2) 2@;
65
66 HEX
67 CREATE "TEST-TABLE ( checksum)
68 0F3 C, 0E5 C, 0D5 C, 0C5 C, 0F5 C, 0DD C, 0E5 C,
0FD C, 0E5 C, 0ED C, 57 C, 0F5 C, 08 C, 0D9 C, 0E5 C,
0D5 C, ( 0FD)
69 0C5 C, 0F5 C, 0D9 C, 08 C, 21 C, 18 C, 00 C,
39 C, 06 C, 0B C, 0EB C, 21 C, 0E4 C, 0F C, 2B C,
72 C, ( 12)
70 2B C, 73 C, 0D1 C, ( 5A)
71
72 "TEST-TABLE HERE OVER - 2CONSTANT TEST-TABLE
73 1400 BASE-ADDRESS!
74
75 DECIMAL
76
77 EXIT
78
79 Result should be:
80
81 :1014000F3E5D5C5F5DDE5FDE5ED57F508D9E5D5FD
82 :10141000C5F5D90821180039060BEB21E40F2B7212
83 :031420002B73D15A
84
85
86

```

Längste Zeile hat 127 Zeichen



Forth Online



Olaf Stoyke

os@cs.tu-berlin.de

In der letzten Ausgabe habe ich die Seite von Reuben Thomas vorgestellt und bereits auf seine Implementierung eines virtuellen Stackprozessors hingewiesen, die jedoch zu jenem Zeitpunkt nicht verfügbar war. Jetzt ist sie es - unter <http://www.cl.cam.ac.uk/users/rrt1001/>! (Dateiname beetle.zip, so um 110 kB lang.)

Wer sich mit der Programmiersprache C unter DOS und den zahlreichen Compiler-Implementierungen für diese Sprache und dieses System nicht so auskennt, wird es sicherlich zu schätzen wissen, daß gforth (von Anton Ertl und Bernd Paysan) als ausführbare DOS-Version nun in der Homepage von Bernd Paysan (<http://www.informatik.tu-muenchen.de/cgi-bin/nph-gateway/hphalle2/~paysan/>) zur Verfügung steht.

Immer wieder taucht in den News unter comp.lang.forth oder dem lokalen Pendant de.comp.lang.forth die Frage nach Literatur zum Thema „Forth“ auf - der präkären Literaturlage kann ich zunächst einmal nichts entgegensetzen, aber: Seit kurzem ist die Buchhändler-Vereinigung im Web vertreten und auch wenn man von denen in der Regel nur etwas hört, wenn es am Jahresende darum geht, die Lage der Branche zu erörtern, so sollte einem doch das „Verzeichnis lieferbarer Bücher (VLB)“ ein Begriff sein. Dieses ist seit kurzem auch im Web abrufbar und kann dort nach Belieben unter <http://www.buchhandel.de> komfortabel durchforstet werden. Zwar ist noch nicht der gesamte Bestand erfaßt, aber etwaige Neuerscheinungen (wie zum Beispiel die lang ersehnten im Bereich Forth?) kann man dort sicherlich bereits feststellen.

Hat mensch dann ein Buch gefunden, kann es auch im Netz bestellt werden: Unter <http://www.telebuch.de/> findet sich der ABC Bücherdienst, der eine eigene komfortable Suchfunktion anbietet. Mithilfe letzterer können deutsche, amerikanische, englische, spanische und holländische Datenbestände durchforstet werden. Zum Suchbegriff „Forth“ werden zum Beispiel sieben deutsche Titel ausgewiesen. Es ist dann möglich, gefundene Titel zunächst in den „Einkaufskorb“ zu legen und dessen Inhalt dann zu ordern, wobei die Bezahlung entweder über das Bankeinzugsverfahren oder über die Kreditkarte abgewickelt werden kann. Bestellungen für Bücher, die in der Datenbank nicht zu finden sind, werden ebenfalls entgegengenommen – einfach mal nachfragen. Wer zusätzlich zur Recherche am Bildschirm auch noch in Papier schmökern will,

kann dieses hybride Vergnügen einerseits unter <http://www.LOB.de> sowie in den in verschiedenen Städten zu findenden Filialen der Fachbuchhandlung JF Lehmanns genießen. Deren LOB-Adresse – „LOB“ steht für „JF Lehmanns Online Bookshop“ – bietet ebenfalls eine Möglichkeit zum Durchsuchen zahlreicher fachspezifischer Kataloge.

Da wir gerade bei der Literatur sind: Unter der zur FU-Berlin gehörenden Web-Adresse <http://ariadne.inf.fu-berlin.de:8000/> wartet (laut einem Artikel im Berliner Tagesspiegel vom 15. Juli erst seit kurzem) ein sehr interessanter Dienst am Menschen im Computer- und Informatik-Geschäft: Der ARIADNE-Server versucht das Problem anzugehen, das bei der Veröffentlichung von Artikeln, Reports und ähnlichem in den einschlägigen Informatik-Fachjournals auftritt - die zum Teil arge Zeitspanne zwischen Eingang des Papiers beim Verlag und der angestrebten Veröffentlichung. Viele dieser Artikel existieren jedoch auch in einer digitalen Form und können über den ARIADNE-Server lokalisiert werden. Ein

... der präkären Literaturlage kann ich nichts entgegensetzen, aber: Seit kurzem ist die

weiteres „Schmankerl“ bei diesem Dienst: Ein Benutzerprofil kann von dem Server eingerichtet und verwaltet werden, bei dem man die eigenen, ganz persönlichen Schwerpunktseiten im Web eintragen und sich, wenn sich auf jenen etwas ändert, über eben jene Änderungen eine Mitteilung zukommen lassen kann.

An der University of Teesside (<http://www.tees.ac.uk>) wird unter anderem auch an der „University of Teesside Stack Architecture (UTSA)“ gearbeitet, die in einer Reihe von verschiedenen Projekten eingesetzt wird. Die Computer Architecture Research Unit (CARU) hat dabei auch eine Reihe von Berichten veröffentlicht, die sich mit den Problemen der Programmierung mit Hochsprachen auf Stack-Prozessoren befassen. Einer gut informierten Quelle zufolge (-) wird ein wesentlicher Schwerpunkt künftiger Forschungsarbeit der CARU-Mitarbeiter im Bereich „Token Execution Architectures“ liegen.

Wer sich ein Bild machen möchte, was denn dran ist am Java-Bytecode, dem Objektcode eines übersetzten Java-Programms, welches dann weltweit (oder besser: web-weit?) ausführbar ist, und wie damit die zugrundeliegende Stack-Maschine programmiert wird, kann sich mit dem Angebot mit der Adresse <http://siesta.cs.wustl.edu/~djh4/> einen Einblick verschaffen. Zwar gibt es auf dieser Ebene noch keine Festlegung seitens der Macher bei Sun (Siehe auch <http://www.javasoft.com>), dennoch wird auf dieser Seite bereits ein Java-Bytecode-Assembler zum Studieren vorgestellt, so daß man „Learning by Doing“ praktizieren kann. □

“N’ bißchen wat Praxis”

von Wolfgang Führer
45663 Recklinghausen, Idastr. 19, Tel.02361/891344
E-mail: wfuehrer@uni-upn.forth-ev.de oder
100415.3227@compuserve.com

Ein Thema, das mein besonderes Interesse fand, war die Ausgabe 1/94 “Messen-Steuern-Regeln”. Eine Sache, mit der ich täglich konfrontiert werde. Ein besonderes Augenmerk habe ich dabei auf die Reglereinstellung gelegt. Eigentlich nicht meine Aufgabe (bin Chemiefacharbeiter), sondern die der MSR-Technik, aber die hat nicht die Zeit stundenlang vor einer Regelstrecke zu sitzen, um P/I/D-Parameter zu optimieren. Ich hingegen muß mich dann mit schaukelnden Regelkreisen rumärgeren. Außerdem ist es oftmals sehr störend oder sogar unmöglich einfach mal einige Kombinationen von P/I/D-Werten auszuprobieren, da dann ganze Anlagen(teile) unruhig laufen und Schwierigkeiten machen. So bin ich auf die Idee gekommen, eine Regelstrecke im PC zu simulieren. Man müßte das Streckenverhalten annähernd gut anpassen und kann dann im PC beliebige Einstellungen ausprobieren ohne den Anlagenlauf zu stören.

Stichworte: PID-Regler Regelung Simulation



Zunächst habe ich die Reglerfunktionen von Rafael Deliano (Seite 34) untersucht, und bemerkte, das sie nicht das tun, was “unsere” Regler tun. Ich stellte fest, daß die P- I- und D-Kerne jeweils selbständig eine Fehlerkorrektur errechnen, welche dann addiert werden und ein neues Ausgangssignal ergeben. Die Regler, mit denen wir arbeiten, sind alle P-abhängig. D.h. die Intensität der P-Korrektur gibt an, wie stark die I- und D-Anteile arbeiten. Sprachlich ausgedrückt: “Was der P-Anteil sofort macht (Sprung) macht der I-Anteil in der eingestellten Zeit.” Reduziere ich also den P-Wert, wird auch der I-Anteil weniger leisten - das gleiche gilt für den D-Anteil. Was bei den “Kernen” von Rafael Deliano noch auffällt ist, das der I- und der D-Kern einen Speicher haben, in dem das Fehlersignal addiert (subtrahiert) wird, bevor es weiter bearbeitet wird. Hat sich das Fehlersignal nach dem ersten Regelungsdurchlauf nicht geändert, wird im zweiten Durchlauf mit dem doppeltem Fehlersignal gearbeitet, usw.

In der Praxis läßt sich mit so etwas aus folgendem Grund nicht arbeiten:

- der Regler einer Mengenmessung erhält ein Fehlersignal und berechnet eine neue Regelventilstellung
- die neue RV-Stellung wird an einen A/D-Wandler übergeben - der Impuls wird an das Steuergerät vor Ort übermittelt (elektrisch oder pneumatisch - ja, gibt es noch)
- das Steuergerät stellt die neue RV-Stellung am Stellglied ein
- die Durchflußmenge ändert sich (mehr oder weniger langsam)
- eine Meßblende ermittelt einen neuen Mengenwert und gibt ihn an den Regler zurück

Das ganze kann zwischen wenigen Sekunden (Mengen), bis zu einigen Minuten (Temperaturen, Gasdrücke) dauern. In der Zwischenzeit würde der Regler pausenlos die eingehenden Fehlersignale (z.B. Plus-Abweichung) addieren. Ist die geforderte Fehlerkorrektur nach einiger Zeit dann durchgeführt, hört der Regler jedoch nicht auf zu arbeiten, weil sein Speicher noch voll aufaddierter Werte ist, die durch einen negativen Fehlereingang eliminiert werden müssen. Der Regler übersteuert dann. Sind die Werte im Speicher dann endlich abgearbeitet (durch anhaltenden Minus-Eingang), ist die Regelabweichung aber schon weit im Minus. Die Folge ist ein ständig schwingender Regler.

Nachdem ich also bei der Untersuchung des Reglers eine Menge gelernt hatte, habe ich selbst einen geschrieben. Da es mein erstes “nützliches” FORTH-Programm wurde, bin ich dann auch mit all den Dingen konfrontiert worden, um die ich mich nie richtig gekümmert hatte, weil ich dachte, sie würden mich nicht betreffen, wie z.B. Integerarithmetik oder Timerfunktionen.

Beginnen wir zuerst mit der Arbeitsweise des Reglers: Zunächst einmal subtrahiert man vom Sollwert den tatsächlichen Istwert und erhält so die Regelabweichung. Diese multipliziert man mit dem P-Wert und erhält so den Wert für die P-Korrektur. Die P-Korrektur multipliziert man nun mit dem I-Wert und erhält so die I-Korrektur. Da der D-Anteil so gut wie nie eingesetzt wird, lassen wir ihn hier zur Vereinfachung mal weg. P-Korrektur plus I-Korrektur ergeben nun den Wert, um die die momentane Regelventilstellung verändert wird. Dabei ist zu beachten, daß die I-Korrektur die RV-Grundstellung dauerhaft verändert, während die P-Korrektur bei Eliminierung der Regelabweichung auf Null geht. Die momentane RV-Stellung wird somit

in der Variablen I-RAM festgehalten. Damit hätten wir dann schon einen PI-Regler.

```
variable sollwert
variable istwert
variable p-wert
variable i-wert
variable i-ram
variable nv-stell
```

```
: pi-regler ( -- )
  sollwert istwert -      \ Regelabweichung
  p-wert @ *              \ p-korrektur
  dup i-wert @ *          \ i-korrektur
  i-ram @ +               \ RV-Grundstellung
                          \ + I-Korrektur
  dup i-ram !             \ neue RV-Grundstellung
                          \ sichern
  +                       \ + P-Korrektur
                          \ = neue RV-Stellung
  nv-stell !
;
```

Das sieht ja noch sehr übersichtlich aus, wenn da nicht noch ein paar Haken wären. Zunächst ist zu beachten, daß die Regelungen mit 12 Bit A/D-Wandlern arbeiten, also mit Werten zwischen 0 und 4000 (die fehlenden 96 Bits dienen zur Anzeige von Über- bzw. Unterschreitungen des Meßbereiches). Um aber eine höhere Genauigkeit während der Berechnung zu erzielen, arbeitet der Regler intern von 0 bis 32000. Eine Addition darf somit 0 nicht unter- und 32000 nicht überschreiten. Das “+” ersetzte ich deshalb durch “+bis”.

```
32000 constant n-max

: +bis ( n summand -- n )
  over n-max swap - min
  over 0 swap - max +
;
```

Achtung! N und der Summand dürfen nicht vertauscht auf den Stack gelegt werden. Wer meint, das ist zu umständlich und ein einfacher “0 max n-max min” Test wäre ausreichend, irrt. Wenn nämlich n schon 30000 ist und 3000 addiert werden sollen, erhält man hiermit nicht etwa 38000 als Ergebnis sondern 0, da 33000 als -32536 verarbeitet werden. Mit 16-Bit signed ist bei 32767 Schluß!

Ein größeres Problem stellt die Multiplikation dar, da ich es in Integerarithmetik lösen wollte. Faktoren über 1 stellen eine Multiplikation, Faktoren unter 1 eine Division dar. Außerdem sind da ja auch noch Faktoren wie 2.37 zu berücksichtigen. Um das Problem zu lösen, wandle ich alle Faktoren in einen Bruch um (z.B. 237/100). Die Floatingpoint-Multiplikation läßt sich dann durch die Skalierungsfunktion (*) ersetzen. Bei der Umwandlung ist noch darauf zu achten, daß der Faktor 0 durch 0/1 zu ersetzen ist. Ich fand folgende Wandlung:

```
n-max 100 / constant n-max%

: <para> ( n.n -- n1 n2 ) \ für zwei Nachkommastellen
```

```
dup 0 >
IF
  dup 100 >
  IF 100 swap
  ELSE n-max swap / n-max%
  THEN
ELSE 1
THEN
;
```

Die Variablen mit den P- und I-Werten werden nun 32-Bit Variablen und mit 2! geschrieben und 2@ gelesen. Die Floating-Zeile

```
p-wert @ *   wird zu   p-wert 2@ */
```

sieht immer noch menschlich aus.

Das nächste Problem ist dann die Umwandlung des Ist- und Sollwertes in den internen Wert und umgekehrt. Intern arbeitet die Regelung von 0 bis 32000, extern kann das aber eine Menge von 0 bis 200 m3/h oder sogar ein Druck von -20.00 bis +20.00 mbar darstellen. Die Umwandlung erledigen folgende Worte:

```
variable #ls              \ Meßbereich Anfang
                          \ Low Scale
variable #hs              \ Meßbereich Ende
                          \ High Scale

: i-w ( n -- n.n )        \ internen Wert in
                          \ Ist-/Sollwert
  #hs @ #ls @ -          \ Meßbereich
  n-max */               \ skalieren
  #ls @ +                 \ damit auch ein Anfang
                          \ von -20 richtig wird
;

: w-i ( n.n -- n )        \ Ist-/Sollwert in
                          \ internen Wert
  #ls @ -
  n-max #hs @ #ls @ - */
;
```

Da sind noch zwei Sachen, die ich mit in die “Werkzeugkiste” geschmissen habe. Ist eine Regelabweichung fast völlig eliminiert, kommt man an einen Punkt, an dem die P-/I-/D-Korrekturen den Wert 0 annehmen. An dieser Stelle hat man dann das Gefühl, als wenn die Regelung “hängen” würde, weil nichts mehr passiert, obwohl noch eine geringe Regelabweichung vorhanden ist. Deswegen habe ich darauf geachtet, das immer wenigstens +1 oder -1 als Korrektursignal ausgegeben wird. Für IMAX wird zunächst einmal das Fehlersignal dupliziert (für das Vorzeichen). Nach erfolgter Berechnung der Korrektur wird dann IMAX ausgeführt.

Um die Genauigkeit von Divisionen zu erhöhen, habe ich dann noch ein “geteilt mit aufrunden” geschrieben. Mit /UP wird “24 10 /up” zu “2” und “25 10 /up” zu “3”. Dieser Genauigkeitsgewinn ist nicht zu unterschätzen!

```
: 1max ( +-n n1 -- n1 )
  abs 1 max swap ?negate
;
```

```

: /up ( n1 n2 -- n1 )
  dup 2/ >r /mod swap r>
  >= IF 1+ THEN
;

```

Ein weiterer Punkt, der mir Kopfschmerzen bereitete, war die zeitliche Abstimmung der Regelung. Wir arbeiten mit einem System der Holländischen Firma Foxboro aus den 80iger Jahren, in der ein Regelungszyklus 0.5 Sekunden dauert.

Zwei Schleifen in der Sekunde sind doch kein Akt - höre ich da grade. Aber der Pferdefuß liegt in meinem Rechner. Ein Atari-Portfolio Palmtop mit 8086 CPU und 4.9 Mhz. Das Betriebssystem bietet mir nur eine Abfrage im Sekundenabstand an und sonst nichts (leider nur ein FUDGE von 18).

Ich habe dann eine Kalibrierungsfunktion in das Programm mit eingebunden, die auf Tastendruck aufgerufen werden kann. Das Programm arbeitet nämlich je nach Modus unterschiedlich schnell. Wenn der Regler auf "Manuell" steht, brauch immer nur ein neuer Istwert berechnet werden, da der Regler selbst nicht arbeitet. Dadurch ist der Berechnungsvorgang fast ein Drittel schneller. Es gibt im Programm deswegen drei verschiedene Werte für die Warteschleife: Automatik, Manuell und Anzeige aus. Die Kalibrierungsfunktion zählt für die Korrektur 100 Berechnungsschleifen ab und ermittelt aus der Differenz zu 50 Sekunden den neuen Wert für die Warteschleife. Ein zweiter Durchlauf kann die Genauigkeit bei großer Abweichung noch erhöhen.

Ein weiterer Punkt ist die Benutzereingabe. Um z.B. den Sollwert zu verändern, hält man die + oder - Taste fest, bis der gewünschte Wert erreicht ist. Nun läuft die Programmschleife nur alle 0.5 Sekunden und die Anzeige aktualisiert sogar nur 1 Mal pro Sekunde. Es sammeln sich somit etliche Tastendrucke im Tastaturinputbuffer an, die nach Entnahme der ersten Taste gelöscht werden müssen.

```

: eingabe ( key -- )
  key?           \ taste gedrückt ?
  IF key         \ taste auf stack
    key?         \ noch mehr tasten ?
    IF
      BEGINN
        key drop   \ tasten löschen
        key? not   \ bis buffer leer
      UNTIL
    THEN
    dup ascii x = IF tuwas1 THEN
    dup ascii y = IF tuwas2 THEN
    ascii z = IF tuwas3 THEN
  THEN
;

```

Vielleicht geht das auch eleganter - ich bin für jeden Vorschlag dankbar.

Eine andere Sache ist die Eingabe der Meßstrecken- und Regelparameter. Ein guter Programmierer sollte immer auf einen DAU gefaßt sein (Dümmster Anzunehmender User). Deshalb ist ein Syntaxcheck nach einer Eingabe fast das Wichtigste und Aufwendigste in einem Programm. Was macht die Eingabefunktion, wenn absolut nichts vernünftiges

ankommt und sich der DAU durch wildes ESC drücken versucht, aus seiner misslichen Lage zu befreien? Ich bin deshalb einen anderen Weg gegangen. Ich biete dem Benutzer den alten Wert an und sage "Mehr oder weniger?" - ist also fast DAU-sicher. Vereinfacht:

```

2variable xypos           \ Position auf dem schirm
2variable min-max        \ größter und kleinster Wert

: range ( n min max -- n ) \ Wert auf Bereich begrenzen
  rot min max
;

: zeige ( n -- n )
  xypos 2@ at dup .
;

: begrenz ( key n -- n key )
  min-max 2@ range zeige swap
;

: einstellen ( x y adr min max -- )
  min-max 2! rot rot xypos 2!
  dup @           \ Wrt aus adr holen
  BEGIN key?
  IF key
    dup ascii + = IF swap 1+ begrenz THEN
    dup ascii - = IF swap 1- begrenz THEN
    dup ascii * = IF swap 10 + begrenz THEN
    dup ascii / = IF swap 10 - begrenz THEN
    13 = IF true ELSE false THEN
  ELSE false
  THEN
  UNTIL
  swap !           \ neuen Wert in adr
                  \ speichern
;

: test ( -- )
  0 0 at ." P-Wert -> "
  10 0 at p-wert @ . \ alten Wert anzeigen
  10 0 p-wert 1 9999 einstellen
;

```

So, ich denke, das waren so die Sachen, die wohl auch die Allgemeinheit interessieren könnten. Wer mehr wissen will, kann in das Listing schauen. - bei mir melden. Bei Bedarf kann man sich bei mir zu Hause auch einloggen und PID.SEQ und diesen Text (PID.TXT) saugen.

Das Programm stellt bestimmt nicht der Weißheit letzten Schluß dar, aber es ist ziemlich komplett, führt den Benutzer gut und läuft stabil. Das einzige, was mir noch nicht so gut gefällt, ist die Anpassung der Meßstrecke - ist noch zu kompliziert, aber ich arbeite daran.

Wer dann doch noch tiefer in die Materie einsteigen und das Programm nutzen will, dem kann ich eine genauere Funktionsbeschreibung liefern. Oder weiß man auf Anhieb, was eine Meßstrecke mit und ohne Ausgleich ist?

Das Listing befindet sich auf der Listingdiskette, oder in der KBBS unter FORTH/MOERS/PID.SEQ



File: pid.seq vom 8.2.1996



```

1 EMP: clv@clvpoint.forth-ev.de
2 ABS: wfuehrer@uni-upn.forth-ev.de (Wolfgang Fuehrer)
3 WAB: uni-upn@kbbs.org
4 BET: PID.SEQ
5 ROT: uni-upn.forth-ev.de
6 MID: 62L5mb0VrbB@uni-upn.forth-ev.de
7 EDA: 19960206151900W+02
8 LEN: 16357
9 Mailer: XP v3.02 R/A12445
10 U-Sender: uni-upn@kbbs.org
11 U-Received: by uni-upn.forth-ev.de (CrossPoint v3.02 R/A12445); 06 Feb 1996 17:52:31 +0200
12 U-MIME-Version: 1.0
13 U-Content-Type: text/plain; charset=ISO-8859-1
14 U-Content-Transfer-Encoding: 8bit
15 X-XP-NTP: 40
16
17 Hallo Claus!
18 Hier ist die andere Hälfte meiner Arbeit. Ich hoffe sie ist
19 nicht zu lang um abgedruckt zu werden. Wenn doch - auch nicht
20 tragisch. Die breite des Textes und des Listings sind auf 64
21 beschränkt. Wenn ich das umformatieren soll, um dir arbeit zu
22 ersparen, sage ruhig bescheid.
23 Bekommst du den Text mit den eingeschobenen Programmblöcken
24 überhaupt hin? Vielleicht läßt du alles in Courier und druckst
25 es einfach so ab?
26
27 ----- schnipp -----
28
29 \ PID-Regler Simulation
30 \ 23/12/95 von Wolfgang Fuehrer
31 \ ZF-Forth (F83) auf Portfolio
32
33 18 fudge !           \ mehr bring der pofo nicht
34
35 \ ===== allgemeine constanten =====
36
37 32700    constant n-max      \ maximaler interner wert
38 n-max 100 / constant n-max%
39 n-max 2/  constant n-max/2
40 120      constant cycmin    \ cyclen pro minute
41 create   eingabe 0 , 0 ,
42
43 \ ===== allgemeine variablen =====
44
45 variable kalibri      \ anzahl der schleifen
46                       \ fuer 0.5 sec cyclyus
47 variable kal-on
48 variable lastkal     \ fuer kalibrierung
49 variable infoaut     \ kalibr. fuer auto
50 variable infoman     \ kalibr. fuer man
51 variable infooff     \ kalibr. fuer anzeige aus
52
53 variable cyclen      \ cyclen seit letzter taste
54 variable temp        \ anzeige 1 sec. um ?
55 variable info        \ true = anzeige ein
56
57 variable oldkey      \ letzte taste
58 2variable min-max    \ bereich fuer einstellen
59 2variable xypos      \ screenpos fuer einstellen
60 variable komma       \ anzahl dezimalstellen
61 variable asn         \ umschalter eingabe sollw./RV%
62 2variable xyanz     \ anzeige was eingestellt wird
63
64 \ ===== allgemeine werkzeuge =====
65
66 : i-% ( n stellen -- % )   \ internen wert in %
67   100 swap 0 ?DO 10 * LOOP
68   n-max */
69 ;
70
71 : .f ( n stellen -- )
72   swap s>d tuck dabs
73   <# 3 roll 0 ?DO # LOOP
74     46 hold #s rot sign #>
75   7 over - spaces type
76 ;
77
78 : range ( n min max -- n )
79   rot min max
80 ;
81
82 : %-i ( % stellen -- n )   \ %-angabe in internen wert
83   n-max 100
84   rot 0 ?DO 10 * LOOP

```

```

85 */
86 ;
87
88 : /up ( n1 n2 -- n1 )     \ geteilt mit aufrundung
89   dup 2/ >r /mod swap r> >
90   IF 1+ THEN
91 ;
92
93 : 1max ( +n n1 -- n1 )    \ mindestens +1 auf stack
94   abs 1 max swap ?negate
95 ;
96
97 : +bis ( n1 summand -- n1 ) \ plus nur bis n-max oder 0
98   over n-max swap - min
99   over 0 swap - max
100  +
101 ;
102
103 : <para> ( n.n -- n1 n2 ) \ regelparameter in
104   dup 0 >
105   IF
106     dup 100 >
107     IF 100 swap
108     ELSE n-max swap / n-max%
109     THEN
110   ELSE 1
111   THEN
112 ;
113
114 \ ===== regeldefinitionen =====
115
116 variable #pr           \ p-wert des reglers
117 2variable p-regler    \ p-wert in * und / anteil
118                       \ umgerechnet
119 variable #ir          \ i-wert des reglers
120 2variable i-regler    \ i-wert in * und / anteil
121                       \ umgerechnet
122 variable i-ram        \ die grundstellung des reglers
123 variable #dr         \ d-wert des reglers
124 2variable d-regler    \ d-wert in * und / anteil
125                       \ umgerechnet
126 2variable d-teiler    \ gibt die form der d-abnahme
127 variable d-add       \ gibt die form der d-abnahme
128 variable d-richt     \ zeigt +/- wechsel an
129 variable d-on        \ d-wert auf 0 ?
130
131 variable #wi          \ wiktungsrichtung des reglers
132 variable rv-wert     \ regelventilstellung
133 variable steuer      \ true = auto, false = man
134 variable sollwert
135 variable istwert
136 variable #ls        \ regler-bereich anfang
137 variable #hs        \ regler-bereich ende
138 variable p-sprung
139 variable i-sprung
140 variable d-sprung
141
142 \ ----- werkzeugkiste -----
143
144 : i-w ( n -- n.n )       \ internen wert in ist/sollwert
145   #hs @ #ls @ - n-max */ #ls @ +
146 ;
147
148 : w-i ( n.n -- n )      \ ist/sollwert in internen wert
149   #ls @ - n-max #hs @ #ls @ - */
150 ;
151
152 : <pr-para> ( n.n -- )
153   #pr @ <para> p-regler 2!
154 ;
155
156 : <ir-para> ( n.n -- )
157   #ir @ <para> i-regler 2!
158 ;
159
160 : <dr-para> ( n.n -- )
161   #dr @ <para> over 0>
162   IF true d-on !
163   d-regler 2!
164   100
165   40
166     cycmin d-regler 2@ */
167     */
168     d-add !
169     100 200 d-teiler 2!
170   ELSE false d-on ! drop drop
171   THEN
172 ;

```


PID

```
349 11 2 at ." | P - Hs - |"
350 11 3 at ." | | - Ls - |"
351 11 4 at ." | D - Wi - |"
352 11 5 at ." +-----+
353 17 2 at #pr @ 0 .f
354 17 3 at #ir @ 2 .f
355 17 4 at #dr @ 2 .f
356 30 2 at #hs @ 1 .f
357 30 3 at #is @ 1 .f
358 30 4 at #wi @ 0 .f
359 17 2 #pr 0 1 9999 einstellen
360 17 3 #ir 2 1 2500 einstellen
361 17 4 #dr 2 0 2500 einstellen
362 30 2 #hs 1 -32700 32700 einstellen
363 30 3 #is 1 -32700 32700 einstellen
364 30 4 #wi 0 -1 1 einstellen
365 hauptschirm
366 r/s-para
367 ;
368
369 : streckparamit ( -- )
370 true #mit/ohne !
371 11 0 at ." +---Streckeneinstellungen---"
372 11 1 at ." | P - Sprungwert|"
373 11 2 at ." | | - Steigwert |"
374 11 3 at ." | Totz. - sec. |"
375 11 4 at ." | Dauer - sec. |"
376 11 5 at ." +-----+
377 19 1 at #ps @ 0 .f
378 19 2 at #is @ 2 .f
379 19 3 at #to @ 0 .f
380 19 4 at #da @ 0 .f
381 19 1 #ps 0 1 9999 einstellen
382 19 2 #is 2 1 2500 einstellen
383 19 3 #to 0 1 3600 einstellen
384 19 4 #da 0 1 3600 einstellen
385 r/s-para
386 ;
387
388 : streckparaohne ( -- )
389 false #mit/ohne !
390 11 0 at ." +---Streckeneinstellungen---"
391 11 1 at ." | ± RV - % |"
392 11 2 at ." | ± Ist - % |"
393 11 3 at ." | Totz. - sec. |"
394 11 4 at ." | Dauer - sec. |"
395 11 5 at ." +-----+
396 21 1 at #drv @ 1 .f
397 21 2 at #dist @ 1 .f
398 21 3 at #to @ 0 .f
399 21 4 at #da @ 0 .f
400 21 1 #drv 1 -1000 1000 einstellen
401 21 2 #dist 1
402 #hs @ #is @ - dup negate swap
403 einstellen
404 21 3 #to 0 1 3600 einstellen
405 21 4 #da 0 1 3600 einstellen
406 r/s-para
407 #dist @ #drv @ * 0<
408 IF -1 ELSE 1 THEN #wi !
409 ;
410
411 : streckenpara ( -- )
412 11 0 at ." +-----+
413 11 1 at ." | Meßstrecke MIT oder OHNE |"
414 11 2 at ." | Ausgleich ? |"
415 11 3 at ." | |"
416 11 4 at ." | MIT (M) OHNE (O) |"
417 11 5 at ." +-----+
418 25 4 at
419 eingabe 1 expect eingabe @
420 dup ascii m = IF streckparamit THEN
421 ascii o = IF streckparaohne THEN
422 hauptschirm
423 time-reset 0 temp !
424 ;
425
426 \ ===== tastaturbedienung =====
427
428 : ein ( +1 -- )
429 * steuer @ asn @ =
430 IF sollwert ELSE rv-wert THEN
431 dup @ rot n-max swap / +bis swap !
432 ;
433
434 : auf-auto ( -- )
435 steuer @ false =
436 IF
437 true steuer ! true asn !
438 rv-wert @ \
439 istwert @ sollwert @ - \
440 p-kern \
441 #wi @ -1 = IF negate THEN
442 +bis i-ram ! \ p-sprung auschalten
443 3 1 at ." AUT "
444 #mit/ohne @ IF 169 ELSE 47 THEN emit
445 THEN
446 info @ true =
447 IF infoaut @ ELSE infooff THEN kalibri !
448 ;
449
450 : auf-man
451 steuer @
452 IF
453 false steuer ! true asn !
454 3 1 at ." MAN "
455 #mit/ohne @ IF 169 ELSE 47 THEN emit
456 THEN
457 info @ true =
458 IF infoman @ ELSE infooff @ THEN kalibri !
459 5 2 do 15 i at ." 0.00" loop
460 ;
461
462 : auf-asn ( -- )
463 steuer @ 0=
464 IF asn @ 0= asn ! THEN
465 ;
466
467 : regelend ( -- )
468 depth 1 do drop loop \ stack von totzeiten reinigen
469 dark
470 ;
471
472 : info-on
473 true info !
474 steuer @ true =
475 IF infoaut @ ELSE infoman @ THEN kalibri !
476 ;
477
478 : info-off
479 false info ! infooff @ kalibri !
480 ;
481
482 : kal-start ( -- )
483 true kal-on !
484 0 lastkal ! 0 temp ! time-reset
485 29 5 at ." ~" 33 5 at ." ~"
486 ;
487
488 : fast ( -- )
489 kalibri @ 2/ kalibri !
490 ;
491
492 : slow ( -- )
493 kalibri @ 2* kalibri !
494 ;
495
496 : halt-ist ( -- )
497 true haltist !
498 ;
499
500 : weiter-ist ( -- )
501 false haltist !
502 ;
503
504 \ ===== tastaturabfrage =====
505
506 : fuehr ( key -- )
507 key?
508 IF key 0 cyclen !
509 key?
510 IF
511 BEGIN key drop key? not UNTIL
512 THEN
513 dup ascii 3 = IF -1 1000 ein THEN
514 dup ascii 4 = IF 1 1000 ein THEN
515 dup ascii 5 = IF -1 100 ein THEN
516 dup ascii 6 = IF 1 100 ein THEN
517 dup ascii 7 = IF auf-man THEN
518 dup ascii 8 = IF auf-auto THEN
519 dup ascii 9 = IF auf-asn THEN
520 dup 27 = IF regelend exit THEN
521 dup ascii i = IF info-on THEN
522 dup ascii o = IF info-off THEN
523 dup ascii k = IF kal-start THEN
524 dup ascii r = IF reglerpara THEN
```

```

525 dup ascii s = IF streckenpara THEN
526 dup ascii f = IF fast THEN
527 dup ascii F = IF slow THEN
528 dup ascii h = IF halt-ist THEN
529 dup ascii H = IF weiter-ist THEN
530 oldkey !
531 ELSE 32 oldkey !
532 THEN false
533 ;
534
535 \ ===== werzeuge =====
536
537 : make-kal ( -- ) \ 0.5 sec-cyclus kalirieren
538 kal-on @
539 IF
540 lastkal @ 1+ dup
541 lastkal ! 100 = \ 100 schleifen ?
542 IF sec-elapsed 1 max \ in n sec
543 dup 29 5 at ." ** .
544 kalibri @ 50 rot */ \ soll 50 sec
545 dup kalibri ! \ neue kalibrierung
546 info @ true = steuer @ true = and
547 IF dup infoout !
548 ELSE dup infoman !
549 THEN
550 info @ false =
551 IF dup infooff ! THEN
552 33 5 at ." @ " .
553 false kal-on ! \ kalibrierung fertig
554 THEN
555 THEN
556 ;
557
558 : pid-init
559 150 #pr ! \ p-wert = 100
560 100 #ir ! \ i-wert = 0.40
561 00 #dr ! \ d-wert = 0.50
562 1 #wi ! \ wirkungsrichtung normal
563 0 #is ! \ regler beginn = 0
564 1000 #hs ! \ regler ende = 100.0
565 true steuer ! \ regler auf autoatik
566 n-max/2 i-ram ! \ regelventilstellung auf 50%
567 n-max/2 rv-wert ! \ " " " "
568 n-max/2 sollwert ! \ sollwert auf 50%
569 n-max/2 istwert ! \ istwert auf 50%
570 100 #ps ! \ form der meßstrecke

571 100 #is ! \ " " " "
572 5 #to ! \ 5 sec. totzeit
573 20 #da ! \ dauer der regelung
574 100 #drv !
575 -10 #dist !
576 true #mit/ohne !
577 0 temp ! \ fuer anzeige
578 2 3 xyanz 2!
579 true info ! \ anzeige an
580 true asn !
581 370 infoaut !
582 430 infoman !
583 500 infooff !
584 r/s-para
585 ;
586
587 \ ===== haupt-steuerstruktur =====
588
589 : p ( -- )
590 dark hauptschirm time-reset
591 r/s-para 0 cyclen !
592 n-max/2
593 #to @ 2* 1 DO dup LOOP \ totzeit auf stack bringen
594 400 kalibri ! \ grundeinstellung
595 BEGIN
596 pid-regler
597 istneu \ aus reglerantwort
598 \ neuen istwert berechnen
599 anzeige
600 fuehr
601 cyclen incr
602 make-kal
603 kalibri @ ms
604 UNTIL
605 ;
606
607 \ ===== compiler ende =====
608
609 pid-init
610 ." p <cr> " fuer start )
611 cr
612
613 \ ===== programm ende =====
614 ## CrossPoint v3.02 ##
        
```

SIMPLE INTERRUPT SYSTEM



1. Processor idling



2. Peripheral thinks it needs service



3. Interrupt pending



4. INTERRUPT, processor leaves idle state



5. Restart, INTERRUPT



6. Start of Interrupt service routine



7.



8.



9. Peripheral knowing it's interrupt was acknowledged

Das Letzte ...

Freitexttabellenkalkulation

von Claus Vogt
Katzbachstr.23; D-10965 Berlin; clv@FORTH-eV.de

Kürzlich rief mich jemand an, der sich für Forth interessierte. Er benötigte für einige kleine Programme eine geeignete Programmiersprache. Bisher hatten er und seine Kollegen mit HP-Taschenrechnern gearbeitet und mochten sich mit der Basic-Notation nicht anfreunden. Daher das Interesse an Forth.

Meine Telefonnummer hatte er aus einem alten Forth-Magazin. Da er nur unter DOS arbeitete, empfahl ich ihm das F-PC von Tom Zimmer und wir vereinbarten einen Termin für eine Vorführung.

Beim Treffen erklärte er mir seinen Anwendungsfall. Als Unternehmensberater erstellt er Texte für seine Kunden. Darin kommen aber so viele zu berechnende Zahlen vor, daß es mit obengenannten HP-Taschenrechnern mühselig und fehlerträchtig wird. Das Format der Texte und die zu verwendenden Berechnungen unterscheiden sich, sodaß eine Lösung mit einer Tabellenkalkulation unbefriedigend erscheint.

Nachdem mir ein solches Problem bisher nicht begegnet war, probierte ich mal, während unseres Gesprächs, eine passende Idee zu entwickeln. Es wuchs folgendes Listing:

```
\\ Heute am blabla ...
{
: abort(( f — )
  IF
    Ascii ( parse
      here place
      abort
  ELSE
    [compile] (
  THEN;
}
```

Der Kunde hat { 100 } Beschäftigte, die jeweils { 250 } DM pro Tag erhalten, sodaß sich eine Ausgabe von { 24000 } DM { -rot * <> abort(falsch!!!) } DM pro Tag ergibt.

Im Grunde handelt es sich hier um einen frei gestalteten Text. Einige Sequenzen sind in geschweifte Klammern gefaßt. Das F-PC interpretiert sie als Forth-Quelle (und ignoriert den restlichen Text):

```
100 250 24000 -rot * <> abort( falsch!!! )
```

Für meinen Interessenten stellten sich noch ein paar Fragen. Die Eignung des F-PC-Editors als Schreibmaschinenersatz kann ich guten Gewissens bejahen, seitdem KEYB.SEQ dem F-PC die Umlaute beigebracht hat. Mit deutschsprachigen Büchern zum Einstieg in F-PC sieht es da schon schlechter aus. Ich empfahl ihm das Buch von Jörg Staben. Völlig beruhigen konnte ich ihn in Hinsicht auf Fließkommaoperationen. Floating Point wird von Menschen, die mit Geldbeträgen zu rechnen haben, bestenfalls als notwendige Krücke akzeptiert.

Nun bin ich gespannt, ob diese Spontanidee sich im praktischen Einsatz so weit ausfeilen läßt, daß sie echte Vorteile gegenüber der Taschenrechnerlösung ausbildet

q

FREITEXTSEQ
2.613
01.10.96



Forth-Gruppen regional

Rhein-Ruhr	Jörg Plewe Tel.: +208 -49 70 68 p Treffen: jeden 1. Samstag im Monat im S-Bahnhof Derendorf Münstererstr. 199,
Moers	Friederich Prinz Tel.: +2841 -5 83 98 p Treffen: jeden Samstag 14:00 Arbeitslosenzentrum, Donaustr. 1, Moers
Darmstadt	Andreas Soeder Tel.: +6257 -27 44
Mannheim	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

Arbeitsgruppe Marc4	Rafael Deliano Tel./Fax.: +89 -841 83 17 pf
Anfänger und Wiedereinsteiger	Gerd Limbach Tel.: +2051 -25 51 12 p Mo.+Di. 20:00 - 22:00
32FORTH (Atari)	Rainer Aumiller Tel.: +89 -6 70 83 55 gp
FORTHchips (FRP1600, RTX, Novix...)	Klaus Schleisiek-Kern Tel.: +40 375 008-03 g
F-PC & TCOM, ASYST (Meßtechnik), embedded controller(H8/5xx//TDS2020, 8051 ... eFORTH...), FUZZY	Arndt Klingelberg Tel.: +2404 -6 16 48 agp
HS/Forth (Harvard Softworks)	Wigand Gawenda Tel.: +30 -44 69 41 p
KI (Künstliche Intelligenz), OOF (Object Oriented Forth)	Ulrich Hoffmann Tel.: +4351 - 712 217 pf Birgit Steffenhagen Tel.: +38204 - 129 33 pa +381 - 498 35 52 g
Forth-Vertrieb	volksFORTH/ultraFORTH, RTX/FG/Super8/KK-FORTH Ingenieurbüro Klaus Kohl Tel.: +8233 -3 05 24 p Fax: +8233 -99 71 f
Forth-Mailbox (KBBS)	+431-533 98 98 :8N1 Sysop: Holger Petersen Fax: +431-533 98 97 f Tel: +431-533 98 96 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

Anmerkung zum Heft 4/1996

Dieses Heft wurde zwar schon mit einem Desktop Publisher erzeugt, aber die Fassung, die dann letztendlich gedruckt worden ist, ging verloren. Es gab aber noch etliche der Beiträge einzeln und maschinenlesbar im PDF und PS-Format. Daraus wurde nun das Heft rekonstruiert.

Die Seiten 11,12,15-20 fehlten jedoch, und wurden daher von der Printversion eingescannt. Der Text darin wurde mittels paper capture (OCR) diesen Bildseiten hinterlegt.

Die fehlende erste und letzte Seite wurde ebenfalls als Scann eingefügt.

Somit gibt das Heft das Original wohl weitgehend wieder, wenn auch nicht alle Seiten in der Qualität der PDF-Originale erhalten geblieben sind.

01.05.2007, M.Kalus