

2/1995

VIERTE DIMENSION

11. Jahrgang 1995, 2. Quartal, DM 10.-

- Kreisalgorithmus
- Forth denkt - Windows lenkt
- Meßtechnik mit dem PC
- Multitaskingfähige Timerfunktion
- Hochsprachen in Silizium
- P 21 - Prozessor
- Forth Tagung '95

11. Jahrestagung
FORTH
in Berlin '95

FORTH MAGAZIN

Organ der Forth Gesellschaft e.V.

FORTH - Shirt



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

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

ForthWORKS

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

In-Circuit Emulation für die 68HC11E Serie

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

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

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

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

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

Ingenieurbüro Dipl.-Ing. Wolfgang Allinger

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

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

ETA Elektrotechnische Apparate GmbH

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

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

FORTech Software GmbH

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

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

Dipl.-Ing. Arndt Klingelberg

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

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

Ing.Büro Klaus Kohl

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

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

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

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

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

IMPRESSUM

Name der Zeitschrift

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

Herausgeberin

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

Redaktion & Layout

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

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

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

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

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

Redaktionsschluß

Jeweils 2 Monate vor Erscheinen des Heftes.

Erscheinungsweise

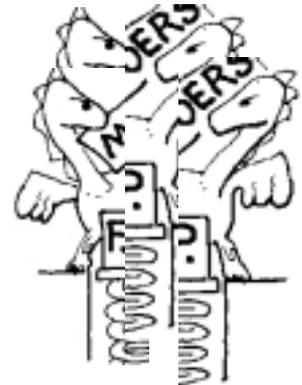
Viermal im Jahr.

Preis

Einzelpreis: DM 10,-

Manuskripte und Rechte

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



Zwischen den Tagungen..

Wie geht es denn der Forth-Gesellschaft? Die Forth-Tagung im April '95 in Berlin war sehr erfreulich. Auch wenn der eine oder die andere fehlte, so war das Programm doch rundum gelungen. Ohne Jörg Plewes Bericht in diesem Heft allzusehr vorzugreifen: Forth wird GUI-fähig werden und seine Portabilität über Betriebssystemgrenzen hinweg erhöhen. Die anderen 'Einrichtungen' der Forth-Gesellschaft, wie der Vertrieb und der Microcontrollerverleih, schlafen auch nicht. Auch der Sysop der Mailbox schrieb einen Brief für's Forth-Magazin. Die lokalen Gruppen sind wie immer vertreten. Erstaunlicherweise scheinen sie alle in und um Moers herum angesiedelt zu sein.

Das letzte Forth-Magazin kam zum ersten Mal aus Berlin. Die Leser und Leserinnen nutzten die Gelegenheit und schrieben durchweg positive Leserbriefe. Die Redaktion bedankt sich bei ihnen und veröffentlicht sie komplett. Die Redaktionsarbeit nimmt langsam Konturen an. Die Anzeigenverwaltung ist jetzt fest in der Hand von Ulrike Schnitter vom Forth-Büro. Rolf Kretzschmar bleibt 'seiner' Zeitschrift treu und erstellt Zeichnungen. Ulrich Hoffmann und Fred Behringer sind als Kolumnisten eingestiegen zum Thema ANS-Forth resp. 'Forth International'. Über diese vielfältige Unterstützung und die vielen von den LeserInnen eingesandten Kurzmeldungen freuen wir uns besonders. Wird das Heft doch erst durch die Beteiligung vieler Menschen bunt, anregend, interessant und umfassend. Weitere Ideen sind jederzeit willkommen.

Der 'Terminkalender' fehlt diesmal. Es mangelte sowohl an Einsendungen als auch an Platz. Aber das dürfte in der Urlaubszeit kaum auffallen.

Die Redaktionstechnik nähert sich langsam einem gewissen Standard. Das DTP-System wurde - hoffentlich erfolgreich - geupdated. Dank Leserunterstützung funktioniert auch Ghostscript, sodaß auch PS und EPS-Dateien eingelesen werden können. Die Redaktion bittet, von der Einsendung weiterer Ghostscriptversionen abzusehen. Netzwerk und Telefonserver stehen noch nicht so ganz, daher mußten die Faxfans diesmal andere Wege zur Redaktion einschlagen. Demnächst werden noch Freiwillige für das Pilotprojekt 'Autoren' layouts selbst gesucht. Wer etwas zusätzliche Arbeit investieren kann und die Artikel z.B. mit WinWord schreibt, rufe doch bitte kurz mal vorher durch.

Tja, und was der Redaktion wohl immer am meisten fehlt, zeigt ein Blick auf die Uhr. Es ist Montag, der 12.6.95, 23:54 und die letzte U-Bahn geht demnächst.

Eine schöne Urlaubszeit wünscht

Claus Vogt, Berlin, Editor der VD

Sie schenken uns einen Artikel..

Dafür möchten wir Ihnen auch ein kleines Präsent überreichen

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

Jupiter ACE - FORTH Programm-Handbuch

S. Vickers 1983, deutsch, 159 Seiten

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

- gestiftet von Rafael Deliano

Steve Jobs

J.S.Young 1988, deutsch, 495 Seiten

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

- gestiftet von Rafael Deliano

Vom Mythos des Mann-Monats

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

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

- gestiftet von Rafael Deliano

The Macintosh Way

Guy Kawasaki 1988 englisch 209 Seiten

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

- gestiftet von Rafael Deliano

Neu!

... aus Nebaj

Schal 2m lang, rot und gelb

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

- gestiftet von Holger Dya

Programmers at Work

S.Lammers 1986, englisch, 385 Seiten

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

- gestiftet von Rafael Deliano

Forth lernen mit dem ZF

Das ideale Geschenk für Forth-Anfänger

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

- gestiftet von Friederich Prinz

Neu!

Das kleine Zeitgeistpaket

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

- gestiftet von Claus Vogt

“Schlagzeuger”

Linolschnitt 21 x 17 cm

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

- gestiftet von Rolf Kretzschmar

“Fisch”

Linolschnitt 21 x 17 cm

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

- gestiftet von Rolf Kretzschmar

Forth-Magazin Vierte Dimension

von hohem archivarischen Wert

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

- gestiftet vom Forth-Büro & der Familie Schmitter

F-PC - Das Buch zum Forth

Für Ein-, Um- und Aufsteiger

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

- gestiftet vom Autor Jörg Staben

Sponsoren gesucht!

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

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

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

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



Kreisalgorithmus

Wie werden aus der kreisförmigen Linie bunte Pixel

von Ulrich Richter.....9

Forth denkt - Windows lenkt

Ein Embedded Controller leistet die Arbeit - VisualBasic sorgt für das GUI-Feeling.

J.Plewe / W.Allinger11

Das war die Forth Tagung '95

Direkt von der Jahrestagung - live berichtet!

von Jörg Plewe15

Brief aus der Provinz

Direkt vom Hüter des SWAP-Drachens!

von Friederich Prinz15

Meßtechnik mit dem PC

Im ersten Teil dieser Serie werden die verschiedenen Schnittstellen vorgestellt.

von Klaus Kohl.....20

Multitaskingfähige Timerfunktionen

Mit ein wenig Zeitsubstraktion braucht nicht jeder Task einen eigenen Timer.
Aber was passiert um Null Uhr Null?

von Bernd Beuster23

Hochsprachen in Silizium

Der Zusammenhang von Prozessordesign und Programmiersprachen läßt sich am besten anhand der historischen Entwicklung aufzeigen.

von Rafael Deliano.....31

P 21

Charles Moores neuester Prozessor ist geeignet seinen Ruf als 'enfant terrible' zu unterstreichen.

von Rafael Deliano.....33

An oder Aus

Eine schnelle Bitausgaberroutine

von Arndt Klingelberg .34

Kolumne:

Es geht - ANS Forth!

Diesmal geht es in der ANS-Kolumne um ANS-Kontrollstrukturen.

von Ulrich Hoffmann27

Forth International

Diese neue Kolumne beschäftigt sich heute mit Forth in Frankreich, Holland, UK.
Weiterhin gibt es 'Gehaltvolles' aus den 'Forth Dimensions' und 'Het Vijgeblad'.

von Fred Behringer28

Inserenten

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

Rubriken

- 3 Editorial
- 3 Impressum
- 4 Autorengeschenke
- 6 Leserforum
- 35 Adressen und Ansprechpartner

Forth-Gesellschaft

- 8 Die Spezialisten
- 26 Vertrieb
- 26 Jagen und Sammeln
- 26 Microcontrollerverleih
- 26 Forth-Magazin sucht...

Projekte

- 6 Braun Ludwig Junior
- 8 Das Forth der Zukunft

Forth-Systeme

- 24 Forth/2 für OS/2
- 19 pfe0911 uploaded

ANS

- 19 DPANS94 im WWW
- 19 Stand4th - Standardizer

Metasprachen

- 19 Microsoft C intern

Forth inside

- 19 Microsoft C intern
- 24 Textverarbeitung
- 30 IPS - Interpreter für Prozeßstrukturen

Prozessorgeflüster

- 19 Marc4:ASIC-Kern aus Chemnitz

Forth-Quellen

- 19 pfe0911 uploaded

Bücher

- 25 Hendtlass: Real Time Forth

Zeitschriften/Netz

- 25 M&T:
- NewMicros Inc & 68HC11
- Die Forth-Spezialisten
- 25 Design & Elektronik:
- Sprachen auf 8-bittern
- Experimente interaktiv steuern
- 25 Leo Brodie im WWW

Produktinfos

- 25 Forth auf CD

Was noch

- 24 Der Name 'C++'

Was fehlt:

- Die Rubrik 'Anfänger'
- Die Auflösung vom Tausche-Rätsel
- Der Artikel zum Gray Parser



Reaktionen auf das Forth-Magazin
vd 1/95

Hallo Claus,

ich habe heute die jüngste VD bekommen. Na, da ist Dir aber wirklich was gelungen ;-) Meine Hochachtung und meinen Dank - die Ausgabe 1/95 ist inhaltlich gut und erfreulich pünktlich ;-). Allerdings weiß ich nicht, wie Du auf den Gedanken kommst, daß diese Ausgabe - u.a. - ohne meine Mitwirkung (welche ?) gar nicht möglich gewesen wäre...

Mach weiter so...

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

Herzlichen Glückwunsch zu Deiner ersten VD. Man sieht (als Insider) dem Ding an, daß Du da tonnenweise Zeit reingesteckt hast. Außerdem beglückwünsche ich Dich zu Deiner Zusammenarbeit mit Rolf. Mit Illustrationen sieht so 'ne VD einfach besser aus.

Eine Meckerei kann ich mir aber doch nicht verkneifen. Ist die Druckerei dieser VD wirklich so konkurrenzlos billig, daß man diese scheiß Qualität akzeptieren muß? Das Deckblatt ist 'ne Katastrophe und mit 'nem Messer in der Hand mußte ich bis jetzt noch keine Zeitschrift lesen. Aber der Inhalt dieser VD macht es wieder wett.

Auf der Forth-Tagung wird die VD garantiert zu einem Thema. Dafür werde sicherlich nicht nur ich sorgen. Meine ganz persönliche Meinung ist nämlich, daß man

Leserforum



als Editor mit 1,2 kDM so herrlich unterbezahlt wird, daß man auf Dauer keine Lust zu diesem Job haben kann. Das ist aber meine PERSÖNLICHE Meinung. [die red. schließt sich dem vorbehaltlos an]. Eine Lösung zu diesem Dilemma müssen wir als Direktorium bzw. Forth-Gesellschaft finden und tragen.

GER.XDH0166@AppleLink.Apple.COM
(Ralf Neuthe)

Die VD ist heute bei mir eingetroffen. Glückwunsch. Sieht echt gut aus und ist ausserordentlich interessant.

- Joerg

joerg.plewe@jpsforth.forth-ev.de (Joerg Plewe)

Ich habe in der letzten VD die Adressen der lokalen Gruppen vermißt? War der Platz zu knapp??

Thomas Prinz, TOPEBAN@RNG.FORTH-ev.de

[Für lokale Gruppen ist jede Menge Platz vorhanden /clv]

Internet, Schaltjahre, ANS in a Box...

Ich habe mal die VD 1/1995 durchgelesen, und folgendes ist mir aufgefallen:

Das Titelblatt sagt: "Mit der Mailbox durch's Internet". Das Inhaltsverzeichnis reduziert das auf "Mit der Mailbox um die ganze Welt". Mit dem Begriff "Internet" wird (meist) die Online-Verbindung gemeint, die dann alle hübschen Möglichkeiten wie FTP, WWW, und was es alles sonst noch so gibt, bietet. Hier gibt es (derzeit[*]) 'nur'

Braun Ludwig Junior

Arbeitsgruppe: "60xxx Betriebssystem in Forth"

Fast immer ist die erste Sprache und das erste kleine Betriebssystem für neue Rechner ein Forth. Nur wird dann auf Basis von C ein anderes Betriebssystem hochgepöppelt, weil kein größeres Forth Betriebssystem (gegen Lizenzzahlungen) erhältlich ist. Diesem Umstande will ich abhelfen, indem ich auf Basis des F 68K von Jörg Plewe ein allgemein erhältliches Betriebssystem mit der Kommandosprache Forth zusammen mit interessierten Leuten erstelle, und dieses Betriebssystem dann als Shareware angepaßt auf viele 68008, 68000, 68001, 68020 ohne MMU und alle sonstigen 68xxx CPUs ohne MMU, die ja in diversen Rechnern vorhanden sind, als Shareware und als OEM Computer mit eingebautem ROM Forth verkauft werden. Geld ist damit wirklich zu verdienen und es gibt ja einige Programme von allgemeinem Interesse wie FIS, und diverse Sprachen, in Forth geschrieben, die angepaßt werden sollen.

Übrigens, zur Zeit (Okt.) besteht die Möglichkeit über die Firma Commodore Amigas mit einem eigenen Betriebssystem

(z.B. Forth) als OEM Produkte zu beziehen. Diese Möglichkeit will ich ausnutzen.

Braun Ludwig junior, 3.6.1995

Sehr geehrter Herr Vogt!

Ich möchte im Rahmen eines Vierte Dimension Sonderheftes Hardware Emufs mit einigen Besonderheiten veröffentlichen. Als Programmiersprache sollen ein Forth System, ein Basic System ein Modula2 System und ein C System als Software mitgeliefert werden. Der Vertrieb soll an Händler und Endkunden über Klaus Kohl von dem Forth Vertrieb erfolgen. An interessierte Ingenieure sollen Lizenzen und Blaupausen geliefert werden.

Braun Ludwig junior, 6.3.1995

Kleinanzeige: Suche Forth und Assembler Programmierer, die auf Basis von Jörg Plewes F68K und Assembler ein komfortables Forth Betriebssystem für verschiedene 68xxx Prozessoren wie 68008, 68000, 68001, 68ec020, (A 600, A1200, 68008 Einsteigercomputer, 68001 Slotcomputer und ähnliche) schreiben wollen.

Gestaffelte Stückzahl Bezahlung.

68008 und 68001 Verkauf in Mio. Stückzahlen vorgesehen und möglich. Durch PCMCIA ROM Card für Amiga 600 und 1200 wird das Commodore Betriebssystem ausgeschaltet und durch neues Forth Betriebssystem ersetzt.

Infos unter Ludwig Braun junior, Postfach 1236, 93328 Neustadt/Donau.

Kleinanzeige: Suche Forth Programmierer, die mit F68k verschiedene Forth Anwendungen wie FIS oder in Forth geschriebene Programmiersprachen neu coden wollen. Als Betriebssystem ist ein komfortables Forth BS auf Basis von F68k und Assembler für verschiedene Hardwareplattformen vorgesehen. Bezahlung erfolgt auf gestaffelter Stückzahlbasis.

Dokumentation muß mitgeliefert werden, bzw von Fremdautoren erstellt werden dürfen.

Infos unter Ludwig Braun junior, Postfach 1236, 93328 Neustadt/Donau.



Die Meinung unserer Leser und Leserinnen veröffentlichen wir gerne. Nur kurz sollte es sein, sonst trifft uns die Pflicht zur Kürzung. Die Redaktionsadresse lautet:

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

Forth. Kurz und Knapp. Das kommt

einen USENET-Anschluss. Das ist die Beschränkung auf die Offline-Dienste NEWS & MAIL, also eben öffentliche Gruppen/Bretter/Areas und private Briefe. Allerdings kann man (fast) alle Online-Dienste auch mühsam per Mail erreichen. Hat aber das Problem, das dabei leicht riesige Datenmengen übertragen werden (Kosten!).

Seite 6: Schaltjahre sind solche, die sich ohne Rest durch 4 Teilen lassen, nicht aber solche, die sich durch 100 ohne Rest teilen lassen, wohl aber wieder solche, die durch 400 teilbar sind.

Seite 6: Der ANS-Standard-Entwurf findet sich in der Box im Pfad /usr/spool/uucppublic [nur für Online-User interessant] im dortigen Unterverzeichnis forth/diverse/ mit dem Namen "ans4-doc.zip" und ist 213 Kilobyte lang. XP-User können sich ihn bestellen...

Seite 22: Im Artikel schreibt Michael Schröder: "Das DPMI-Paket habe ich ... in der Forthbox München und in der Box der Moerser Forthgruppe abgelegt" Ich hätte hier auch Platz dafür, ich muss es nur irgendwoher bekommen... [Du hast es schon im VD-Listing 'VD-9501.ZIP' / clv]. Genauso können natürlich andere Pakete hier abgelegt werden. Es gibt schon ein DOS (Ewald Rieger) und ein 68K (Jörg) -Paket.

Seite '35' (hintere Umschlagseite innen) KI: Ulli Hofmann hat eine neue Telefon-Nummer. Mailbox: Meine Privat-Nummer ist nicht mehr 333572, sondern 533 98 98. Bei allen Telefonnummern dort ist "0+VORWAHL" angegeben. Ich würde entweder das "+" oder die "0" weglassen...

Gruss, Holger
 hp@kbbs.org, April '95

[Die Redaktion lädt Dich hiermit ausdrücklich zu einer regelmäßigen Kurzmeldung/Rubrik 'Neu in der KBBS' ein. Vielleicht gemeinsam mit dem Forth-Vertrieb Klaus Kohl? /clv]

Achtung: Nicht schalten! in vd 1/95

Ja, die VD ist da. Pünktlich zu Beginn des zweiten Quartals. Die paar Tage holst Du noch auf, dann erscheint die VD sogar in dem Quartal, das auf der Titelseite steht :-)!

Ach ja, an Aprilscherzen ist mir beim ersten Durchlesen nur einer aufgefallen: Das

Jahr 2000 sei kein Schaltjahr. Das ist falsch. Das Jahr 2000 ist, wie alle durch 400 teilbaren Jahre, ein Schaltjahr. Nur das Jahr 1900 oder etwa 2100 sind (seit Papst Gregor VI.) keine Schaltjahre. Deshalb können wir alle noch getrost weiter "ewige" Cäsar-Kalender programmieren, bis das auffällt, sind wir längst tot ;-).

Bernd Paysan
 "Late answers are wrong answers!"
<http://www.informatik.tu-muenchen.de/cgi-bin/nph-gateway/hphalle2/~paysan/paysan@Informatik.TU-Muenchen.DE>
 (Bernd Paysan), 7.4.95

VD - Kommentar, "100 Tage neue Mailbox" in vd 1/95

Ein Kommentar zur letzten VD, prima! Loest in mir immer unangenehme Gefuehle aus -- eigentlich wollte ich doch schon seit langem.

Ich moechte lediglich den Artikel "100 Tage neue Mailbox" kommentieren. Auch ich habe Crosspoint eine Zeitlang benutzt, auf dem PC ist das Programm sicherlich konkurrenzlos guenstig/komfortabel/leistungsfahig.

Aber: Von anderen UUCP Umgebungen mit modernen News/Mailreadern trennen es doch noch Welten. Als Frontend fuer mail/news brauchbar .. aber mehr auch nicht.

z.B Taylor-UUCP (bei mir unter RiscOS aber auch unter Linux ...): Nach kurzer Einarbeitung lassen sich problemlos Zusatztools schreiben, bei mir laufen ein mail-server (als RiscOS-Forthmacs support) und eine mailing-list (alle in Forthmacs geschrieben) ohne Problem. Wie so viele DOS Loesungen ist dieses wieder ein abgeschlossenes Programm (mit zugegeben vielen Optionen), aber eben nur den eingebauten.

Hat eigentlich schon jemand versucht, XP-uucp von Forth aus zu nutzen? --

hs@4thware.winnet.de (Hanno Schwalm)

"Extending Forth" in vd 1/95

Zur Anmerkung (3) des Autors auf S.22: XZF läuft unter der OS/2 Warp Kompatibilitätsbox, wenn man bei den Einstellungen

DPMI_DOS_API_ENABLED

eingibt. Die Defaulteinstellung AUTO liefert zwar TRY-PM o.k., aber nach einem ALLOC ist der Selektor ungültig (kann man mit .SEL/DESC-INFO nachvollziehen). Übrigens - ein gut geschriebener Beitrag!

Gruß Bernd
 B.Beuster@bbepoint.forth-ev.de (Bernd Beuster)

Michael Schröder's 'XZF' ist uns hier in Moers schon etwas länger bekannt. Von der 'Mäuseschieberoberfläche' weiß ich, daß sie einen DPMI-Manager dazu nutzt, mehrere Anwendungsprogramme quasi parallel zu betreiben und sich dabei - mehr schlecht als recht - abmüht, diese Programme in Speicherbereichen zu halten, die vor- und gegeneinander geschützt sein sollen. OS/2 stellt spätestens seit der Version 1.3 ebenfalls einen DPMI- Manager zur Verfügung. Anders als Microsoft ist es IBM allerdings gelungen, diesen Part des Protected Mode laufsicher und stabil zu entwickeln. Das nutze ich seit vielen Jahren für mich dergestalt aus, daß ich ZF (und alle anderen DOS-Programme) in DOS-Sessions von OS/2 arbeiten lasse. Das

funktioniert reibungslos. Mehr hat mich bisher am DPMI nicht interessiert.

Das ZF hat etwas mehr als 30.000 Byte freien 'Codespace'. Wenn es wirklich einmal knapp wurde, war es nie ein Problem, ein oder zwei zusätzliche Segmente zu allokiieren. Ein bißchen aufwendig war es schon, über die Segmentgrenzen hinweg eine 'saubere Verzeigerung' zu organisieren.

Michael hatte von Anfang an neben den Sourcen eine sehr ordentliche Dokumentation seinem ZF-DPMI beigefügt. Da war das erste Ausprobieren kein Problem. Und ich war angenehm überrascht, als ich durch .MEM-INFO angezeigt bekam, daß mir 'irgendwo in den von OS/2 verwalteten Ressourcen' noch 4 MByte Speicher zur Verfügung standen !Damit mußte ich sofort experimentieren.

Mit einigen 'Dateiexperimenten' hatte ich in diesem Speicherbereich so beeindruckende Ergebnisse, daß ich mich für das 'ZF-DPMI' zu erwärmen begann. Vor allem das Sortieren einer großen Datei lief jetzt einfach

Das war es auch nicht. OS/2's Standardeinstellungen für DOS-Sessions stellen das DPMI auf ein LIMIT von 4 MByte ein. Das läßt sich natürlich ändern., z.B. auf das Maximum von 512 MByte. Weil das ZF ein reines 16-Bit FORTH ist, sind maximal 65 MByte nutzbar!

Durch Michael's Arbeit eröffnen sich für die Bereiche Dateiverwaltung, Bildbearbeitung und für alle 'Prozesse', in denen möglichst viel Informationen im Speicher eine möglichst hohe Verarbeitungsgeschwindigkeit mit sich bringen, ganz neue Möglichkeiten. Zudem entfallen die gelegentlich lästigen Überlegungen und Definitionen zu den bisherigen Segmentgrenzen der INTEL-Adressierungen.

Ich habe das XZF auf einem 80486/66 DX2 unter OS/2 3.0 WARP ausprobiert, wobei die 20 MByte meines PC bisher entweder von OS/2 oder von diversen Flugsimulatoren genutzt wurden. Dank XZF kann ich, sobald sich eine entsprechende Gelegenheit ergibt, das RAM auch einmal etwas 'sinnvoller' nutzen.

F.PRINZ@MHB.gun.de (Friedrich Prinz)
 Friedrich Prinz
 Forthgruppe Moers
 April 1995



**Worte
verkehrt rum
auf dem
Stack...**

Liebe Redaktion

Wer legt in F83 eigentlich fest, daß doppelgenaue Ganzzahlen in 2VARIABLEN und auf dem Stack verdreht abgelegt werden und warum? Beispielsweise erscheint 12345678. in einem DUMP-Auszug als 34 12 78 56 und nicht, wie es logisch wäre, als 78 56 34 12. - Der Transputer T800 ist in jeder Hinsicht "little-endian". Also habe ich in meinem Transputer-Forth-System F-TP diese Verdrehung von Anfang (1990) an "berichtigt". - Wie ist es bei dpANS-5? Wenn anderer-seits überhaupt keine Abspeichervorschrift besteht, dann sind so unschuldige Worte wie : U.R 0 SWAP D.R ; (für F83, F-PC und Turbo-Forth) systemabhängig: : U.R 0 -ROT D.R ; (bei mir seit 1990 in F-TP). Mir sind diese Dinge (wieder mal) beim Lesen des sehr interessanten Artikels "Omzetting van de 83-standaard naar dpANS-5" von Willem Ouwerkerk in der Nummer 44 (1994) der auch im Ganzen nicht minder interessanten Zeitschrift "Het Vijgeblad" der holländischen "Forth-Gebruikersgroep" in der HCC (Hobby Computer Club) aufgefallen.

Fred Behringer, München

Antwort:

Ein für die Portierung von Programmen nicht zu unterschätzendes Problem ist die Frage, in welcher Reichenfolge einzelne Teile der Zahlenrepräsentation im Speicher abgelegt werden (Endianess). Um effiziente Programme zu ermöglichen, erlaubt der ANS Forth Standard eine beliebige Anordnung der Daten innerhalb einer einzelnen Zelle (einfachgenaue Zahl), wodurch die Implementierung von @ ('Fetch') und ! ('Store') auf entsprechende Wort-Befehle der Hardware zurückgreifen kann. Diese Freiheit ist nur dann störend, wenn einfachgenaue Zahlen nicht als Einheit bearbeitet werden (z.B. Daten mit C! schreiben und mit @ lesen), was eher selten geschieht. Anders bei doppelgenauen Zahlen, die nicht nur als eine Einheit sondern eben oft auch als höher- und niederwertiger Teil behandelt werden. Für portable Programme muss also deren Reihenfolge festgelegt werden. ANS Forth legt hier die historisch übliche Reichenfolge: 'höherwertige Zelle oben auf dem Stack und an niedrigeren Adressen' fest. Dies führt zu einfacheren Umwandlungen zwischen einfachgenauen und doppelgenauen Zahlen. Entsprechendes legt auch der Forth-83 Standard fest.

Ulrich Hoffmann, Eckernförde



**Fragen
Sie sich
auch manchmal,
warum sie sich
selbst fragen,
statt mal
jemand anders
zu fragen?**

**Wir
auch!
Und zwar uns!**



**Das Forth der
Zukunft**

Hallo FORTHer,

F83 war lange genug Standard, um immer wieder einer ganzen Reihe Implementierer Optionen genug zu lassen, aus eben dieser Reihe zu 'tanzen'. ANSI, die Spaetfolge aller Versuche die gerade erwachten Implementierer wieder in die bewusste Reihe zu locken, ist kaum mehr als eine Totgeburt. Schliesslich 'macht' ANSI nichts Neues, sondern beschraenkt sich darauf, auf neue Art in alte Fehler zu stolpern. Insbesondere werden von ANSI die Moeglichkeiten moderner Prozessoren und Betriebssystemkonzepten nicht beruecksichtigt. Einen Trabbi rosa anzustreichen und mit Ralleystreifen zu bekleben, macht aber auch keinen Formel 1 Wagen aus diesem Automobil... Mir schwebt 'ein ganz neues FORTH' vor. Das sollte die Moeglichkeiten des Protected Mode der INTEL-CPU's nutzen, das preemptive Multitasking von OS/2, UNIX und Anderen und, wenn moeglich, auch die Optionen der graphischen Oberflaechen eben dieser Systeme fuer sich selbst und fuer alle Applikationen nutzbar machen.

Einen Ansatz dazu hat Ulf Stelbe auf der diesjaehrigen FORTH-Tagung bereits geliefert, der FORTH-Systeme (sehr vereinfacht) nur noch als indirekt gefadelten Fadencode implementieren und alle 'echten' Codes in DLLs auslagern moechte. Dieser Ansatz birgt meines Erachtens entscheidende Nachteile, ist aber ganz sicher mehr als nur 'diskussionswuerdig'.

Zu einer solchen Diskussion (natuerlich nicht nur ueber Ulf Stelbe's Ansatz) rege ich an, lade ich ein, fordere ich auf - wie auch immer. Ich moechte gerne ein 'utopisches' FORTH diskutieren, frei von F83, frei von ANSI und frei von allen bisherigen, bekannten Strukturen. Lasst und einfach einmal eine Weile ein wenig 'spinnen' (Das koennen wir FORTHer doch ohnehin am besten, sagt man...). Lasst uns versuchen, ein "FORTH der Zukunft" zu entwerfen, dem Jede(r) (vom Anfaenger bis zum Profi !!!) seine Ideen und Vorstellungen mitgeben sollte.

Ich biete mich an, alle dazu eingehenden Statements zu sammeln und, gegebenenfalls (das hoffe ich aber sehr) zusammenzufassen. Dazu werde ich diese Message gleichzeitig in Z-NETZ\SPRACHEN\FORTH und in DE\COMP\LANG\FORTH absetzen, sowie dem Claus Vogt fuer eine der naechsten VDs senden.

...und jetzt lasst mal ' hoeren' ...

F. Prinz, Forthgruppe Moers, 28.Mai1995

VIERTE DIMENSION
Die Spezialisten vom Forth-Magazin ...
FORTH MAGAZIN
Organ der Forth Gesellschaft e.V.

*ANS und Standard-Forth:Ulrich Hoffmann;
uho@informatik.uni-kiel.de; Sehestaeter Str. 26;
24340 Eckernförde.*

*Forth international: Fred Behringer; Planegger
Str. 24; 81241 München.*

*Aufbau Lokale Gruppen und globaler Utopien:
Friederich Prinz; f.prinz@mhb.gun.de; Homberger
Str. 335;*

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

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

*Noch Fragen?
Sonstiges:Claus Vogt; clv@forth-ev.de; Eberstr.
10 10829 Berlin.*



Kreisalgorithmus

von Ulrich Richter

Oberwallstraße 4, 47441 Moers

Problematik: Beim Zeichnen eines Kreisbogens mit einem beliebigen Radius um einen Mittelpunkt herum, lassen sich eine Reihe ganz unterschiedlicher, aus der Mathematik sattsam bekannter Lösungsansätze wählen, von denen in der Regel diejenigen bevorzugt werden, die mit Winkelfunktionen arbeiten. Solche Lösungsansätze sind mathematisch ausreichend genau - wenn man den Kreisbogen auf einem Blatt Papier darstellen will.

Auf dem Computerbildschirm steht der Routine die den Kreisbogen zeichnen soll aber kein Papier zur Verfügung, sondern eine Matrix aus Punkten. Hier beginnen die ersten Schwierigkeiten.

Stichworte: Grafik Kreis

Wenn der Kreisbogen an einem beliebigen, mathematischen Punkt 'P' exakt einen der physikalischen Punkte der Bildschirmmatrix 'schneidet', dann läßt sich dieser Punkt völlig problemlos auf dem Bildschirm darstellen. Allerdings ist die 'Inkongruenz' zwischen Mathematik und Technik wesentlich häufiger anzutreffen.



Der mathematisch exakte Punkt 'P' läßt sich mit eben dieser Exaktheit gar nicht in der relativ groben Matrix beschreiben. Statt dessen könnten drei der Bildpunkte für den mathematischen Punkt 'zuständig' sein. Es gilt also, denjenigen Punkt herauszurechnen, der 'am wenigsten falsch' ist.

Nun kommt als weitere Komplikation hinzu, daß Computer eigentlich gar nicht mit der mathematischen Exaktheit arbeiten können, die hier gefordert ist. Tatsächlich 'kennen' Computer keine Zahlen mit Nachkommastellen, sondern nur integrale (ganzzahlige) Zahlen. Die mittels Programmroutinen nachgebildeten 'Fließkommaeigenschaften' kosten den Computer sehr viel Rechenzeit, die bei der Berechnung eines einzelnen

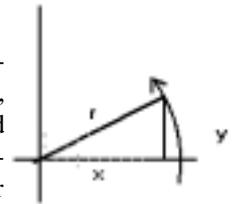
Bildschirmpunktes nur Bruchteile einer Sekunde ausmacht, sich aber bei der Berechnung eines kompletten Kreisbogens schnell zu deutlich sichtbar verstreicher Zeit aufsummiert. Gleiches gilt für transzendente Funktionen (wie die Winkelfunktionen). Auch diese Funktionen muß der Computer mittels spezieller ProgrammROUTINEN nachbilden - bei noch wesentlich größerem Rechenzeitverbrauch, als dies für Fließkommaoperationen notwendig ist.

Und auch der Gebrauch von Multiplikationen und Divisionen verbietet sich im Grunde immer dann, wenn ein Programm schnell arbeiten soll. Selbst die in der Prozessorhardware direkt 'verdrahteten' Multiplikationsbefehle benötigen noch ca. 65 Mal so viel Zeit wie eine Addition. Noch wesentlich ungünstiger ist das Verhältnis zwischen Division und Subtraktion.

Nun läßt sich (beinahe) jede mathematische Operation (wenn man einige Überlegungen investiert) auf eine Addition zurückführen [siehe FORTH-Kurs für Einsteiger, Grundlagen-Arithmetik, Forthgruppe Moers], was zumindest schon einmal den Verzicht auf Multiplikationen und Divisionen denkbar werden läßt. Es müßte halt zusätzlich einen Weg geben, den exakten Verlauf eines Kreisbogens so zu beschreiben, daß man auch auf Winkelfunktionen verzichten kann. Einen solchen Weg hat Ulrich Richter aus der Forthgruppe Moers aufgezeigt.

Lösung

Der Grundgedanke ist zunächst, daß sich, während der Kreisbogen gezeichnet wird, sehr viele rechtwinklige Dreiecke ergeben. Und für diese Dreiecke lassen sich eine Reihe von Aussagen treffen. Zum Beispiel ist die Hypotenuse $-r$ für jedes dieser Dreiecke gleich. Die Seite $-y$ jedes Dreiecks ist immer um -1 größer als es das $-y$ des vorhergegangenen Dreiecks war. Die Seite $-x$ wird allmählich kürzer, während $-y$ wächst, allerdings nicht stetig, sondern entsprechend dem Tangens, der sich bei jedem einzelnen $-y$ einstellt. Von $-x$ läßt sich außerdem sagen, daß es im 'allerersten Dreieck' gleich dem Radius $-r$ ist, wobei $-y$ dann -0 ist.

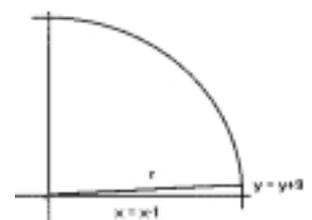


Die wichtigste Aussage ist aber sicherlich, daß für rechtwinklige Dreiecke neben der Trigonometrie auch noch Pythagoras' Lehrsatz gilt

Und nach Pythagoras gilt für jedes dieser Dreiecke $r^2 = x^2 + y^2$.

Wenn es, wie zu Anfang beschrieben, darum geht, aus mehreren möglichen Punkten denjenigen herauszufinden, der 'am wenigsten falsch' ist, dann kann es dazu eigentlich kein genaueres 'Meßinstrument' als den Radius geben. Es geht also darum, denjenigen Punkt herauszufinden, der dem geforderten Abstand zum Kreismittelpunkt am nächsten kommt. Das zwingt aber zu einer ständigen Fehlerkorrektur !

Man stelle sich einmal einen 'etwas größeren' Radius vor. Während $-y$ bereits 9-Mal verändert wurde, mußte zu Beginn der Steigung $-x$ erst 1-Mal verändert werden. Sehr wahrscheinlich ist diese Veränderung von $-x$ gar nicht genau genug. Und zuvor war $-x$ überhaupt nicht verändert worden,



Kreisalgorithmus

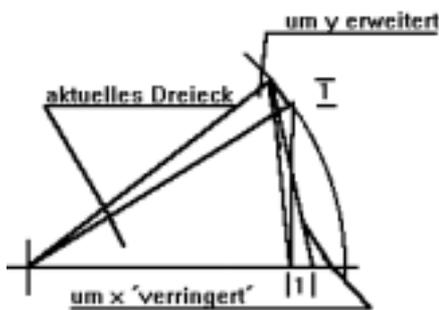
während -y- bereits 8 Schritte 'gewachsen' ist.

Ganz sicher stimmt Pythagoras hier nicht mehr!

Die Gleichung $r^2 = x^2 + y^2$ schleppt hier also einen Fehler mit sich herum, dessen gelegentliche Korrektur sich darin äußert, daß -x- um 1 verkleinert wird. Nun erfolgt diese gelegentliche Korrektur nicht in festgesetzten Intervallen. Wie schon aus der Skizze zu sehen ist, wird sie bei wachsendem -y- immer häufiger erfolgen müssen. Wenn -r- unter einem Winkel von 45° zur Abszisse steht, ist das Verhältnis von x:y exakt 1:1. Und wenn -y- weiter wächst, kehrt sich das Verhältnis sogar um. Dann wird stets -x- um 1 verändert, während die Korrekturen an -y- immer seltener erfolgen.

Mit anderen Worten: Nach der ersten Fehlerkorrektur wird (sehr wahrscheinlich) ein Restfehler übrig bleiben. Auf diesen Restfehler müssen sich die Fehleranteile der weiteren y-Operationen addieren, bis zur nächsten Korrektur, die dann 'etwas früher' erfolgen muß - usw.. Diese sich automatisch anpassende Fehlerkorrektur und -fortführung faßt Ulrich Richter in folgenden Gedanken zusammen:

Es müssen immer drei mögliche Punkte untersucht werden, wovon einer durch das aktuelle Dreieck beschrieben wird, ein anderer durch das aktuelle Dreieck plus der anstehenden y-Operation und der dritte durch das aktuelle



Dreieck plus der anstehenden y-Operationen und minus der letzten x-Operation.

Dabei läßt sich das aktuelle Dreieck einfach beschreiben mit:

$$r^2 - x^2 - y^2 = \text{Fehler}$$

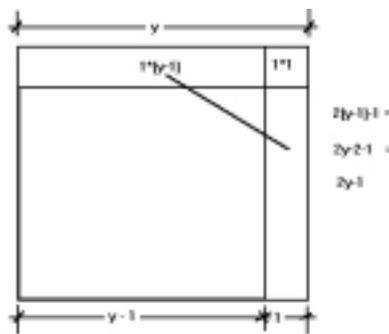
Der Fehler ist dabei ein Wert < 0 , weil x stets um seinen Fehlbetrag zu groß bleibt. Je länger x nicht korrigiert wird, umso größer wird der Fehler in seinem absoluten Betrag. Das zweite zu prüfende Dreieck läßt sich durch seine Veränderung zum aktuellen Dreieck beschreiben mit:

$$-2y - 1$$

-2y - 1 ist die Differenz zum vorherigen Dreieck, wobei hier sofort angenommen wird, daß y zwar relativ zur Abszisse wächst, in seinem tatsächlichen 'Wert' aber fällt, weil es sich auf dem Bildschirm 'nach oben' bewegt, also zu einer jeweils kleineren Zeilennummer hin. Der Ausdruck kommt durch einfaches Ausmultiplizieren zustande, wenn man annimmt:

$$(y-1)^2 = (y-1)(y-1) = y^2 - 2y + 1$$

Wenn also '-1' die Veränderung von einem y zum nächsten y ist, dann wird von einem y^2 zum nächsten y^2 die Veränderung -2y-1 betragen.



Das läßt sich auch 'visualisieren', wie die Skizze zeigt.

Nun läßt sich definieren:

$$r^2 - x^2 - y^2 - 2y - 1 = \text{Delta1}$$

und für den Fall, daß x und y 'nachgezogen' werden:

$$r^2 - x^2 - y^2 + 2x - 2y - 2 = \text{Delta2}$$

Dabei ist im letzten Ausdruck der x-Anteil positiv, weil das 'Verringern' von x numerisch ein 'Anwachsen' von

x zur Folge hat. Und $2x - 2y - 2$ ist natürlich nichts Anderes als $2x - 1 - 2y - 1$.

Jetzt läßt sich aber erkennen, daß -x- spätestens dann 'nachgezogen' werden muß, wenn Delta1 größer als Delta2 wird - weil dann der Fehler in -x- mehr als einen ganzen Punkt ausmacht.

Man könnte nun also einen Algorithmus festlegen, der natürlichsprachlich etwa so aussieht:

```
solange x nicht in seiner Endposition ist
(oder y)
  mache Folgendes
    setze den durch x und y beschriebenen Punkt
    r2-x2-y2-2y-1= Delta1
    r2-x2-y2+2x-2y-2= Delta2
    y = y-1
    wenn Delta1 > Delta2 dann x = x-1
  wiederhole
```

Hierzu läßt sich, wie später eine FORTH HighLevel Implementierung zeigen wird, durchaus noch etwas vereinfachen. Schließlich soll die Bildung der Quadrate vermieden werden. Tatsächlich kann noch auf mehrere Berechnungen verzichtet werden, wenn man noch zwei oder drei weitere Gedanken in den gerade beschriebenen Algorithmus investiert.

Aus der 'Herleitung' von -2y-1 läßt sich zum Beispiel sehen, daß in der Regel der numerische Abstand von $r^2 - x^2 - y^2$ zum nächsten $r^2 - x^2 - y^2$ ebenfalls -2y-1 sein muß. Wenn -x- noch nicht 'nachgezogen' wurde, -y- aber grundsätzlich immer um 1 dekrementiert wird, dann gilt für jeden dieser Ausdrücke

$$r^2 - x^2 - (y-1)(y-1) = r^2 - x^2 - y^2 - 2y - 1$$

Folglich läßt sich der gesamte 'pythagoräische Teil' zusammenfassen und in einer Variablen unterbringen, der sich zum Beispiel der Name PYTH geben läßt. Von dieser Variablen läßt sich aber wiederum sagen, daß sie zum Start der Routine den Wert 0 enthält und 'in der Regel' bei jedem Schritt um 2y-1 verringert wird. Die 0 zum Start der Routine erklärt sich natürlich einfach aus der Erkenntnis, daß -x- zu Anfang gleich -r- ist, während -y-, weil noch nicht verändert, gleich 0 ist. Von PYTH



läßt sich zudem sagen, daß dieser Ausdruck immer dann wenn $\Delta 1 > \Delta 2$ ist, zusätzlich um $2x-1$ verändert wird. Damit können die Multiplikationen bei der Bildung der Quadrate entfallen. PYTH läßt sich um 'feststehende' Ausdrücke per Subtraktion und Addition verändern !

Diese 'feststehenden Ausdrücke', z.B. $-2y-1$, ändern sich aber untereinander ebenfalls kontinuierlich. So gilt als Abstand von einem $2y-1$ zum nächsten $2y-1$, daß der jeweils nächste Ausdruck in seinem $-y$ - bereits ebenfalls wieder einen um 1 geringeren Wert enthält, als das vorhergegangene $-y$ -. Also läßt sich auch schreiben

$$-2y-1 = -2(y-1)-1 = y-1-y-1-1 = -2y-3$$

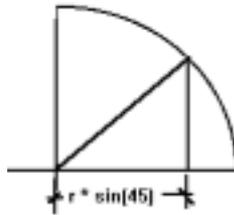
wodurch sichtbar wird, daß der Abstand zwischen

den Abständen der Quadrate (jetzt geht es natürlich verwirrend tief) stets 2 beträgt. Selbst-verständlich kann ein 'pfißiger' Algorithmus das ebenso nutzen wie die Zusammenfassung bei PYTH. Weitere Rechenarbeit kann eingespart werden, für das 'DeltaY' ebenso wie für das 'DeltaX'.

Mit den bisherigen Überlegungen ist es problemlos möglich, den Kreisbogen über Pythagoras zu beschreiben und gleichzeitig sowohl Rechenschritte einzusparen, als auch auf zeitintensive Multiplikationen und Divisionen (fast) vollständig zu verzichten. Lediglich zwei Multiplikationen sind zu Beginn der gesamten Kreisroutine noch notwendig. Zum Einen muß für das erste DeltaX, daß sich, entsprechend dem bisher Gesagten als $2*\text{Radius} - 1$ definiert, eine Multiplikation mit 2 durchgeführt werden, die aber in den meisten Forthsystemen als Verschiebeoperation nach links definiert und damit noch schneller als eine Addition ist.

Zum Anderen muß die Grenze für x (oder y) bestimmt werden. Hierzu ist noch ein weiterer, kurzer Gedanke notwendig.

Die Vorgehensweise $-y$ - bei jedem Schritt um 1 zu verändern, und nur dann $-x$ - 'nachzuziehen', wenn die beschriebenen Bedingungen erfüllt sind, läßt



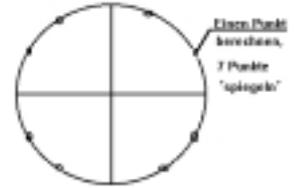
sich nur bis zu einem Winkel von 45° aufrecht erhalten. Danach kehrt sich das Verhältnis zwischen x und y um, d.h. nun müßte $-x$ - grundsätzlich um 1 weitersetzt

werden.

Das ist nicht weiter tragisch, denn ein wirklich schneller Kreisalgorithmus rechnet ohnehin nur einen 8-tel Kreisbogen aus und 'spiegelt' anschließend die einzelnen Punkte auf die anderen Achtel.

Die Frage ist also nur noch, wie sich die Grenze am einfachsten feststellen läßt. Auf der Skizze ist zu sehen, daß $-x$ - den Wert von $r * \sin(45)$ annimmt,

wenn $-y$ - unter einem Winkel von 45° zur Abszisse ansteigt. Deshalb benötigt der Algorithmus aber keine Winkel-



funktionen. Weil hier nur der Funktionswert eines einzigen Winkels gebraucht wird, läßt sich dieser einfach in der Routine 'fest verdrahten'. Nimmt man den Sinus von 45° mit 0,7071 an, erhält man, nach entsprechender Skalierung mit 10000, auch bei integeren Rechenoperationen eine mehr als ausreichende Genauigkeit bei der Bestimmung der Grenze. \square

```

\ Der KREIS - nach Ulrich Richter
\
\ möglicher Weise geht's schneller - aber ganz sicher nicht exakter !
\-----
0 CONSTANT Pyth          \ Radius^2 - X^2 - Y^2
0 CONSTANT dX             \ 2X - 1
0 CONSTANT dY             \ 2Y - 1

0 CONSTANT xEnde

: Kreis ( x y Radius )
\ Da steckt der 'Richter' drin ;-)
\ siehe Beschreibung...
\ x1 = Radius

    DUP =: x1
    0 =: y1
    2* 1- =: dX
    0 =: dY
    0 =: Pyth
    =: y2
    =: x2

    x1 7071 U*D
    10000 UM/MOD
    =: xEnde
    DROP

    BEGIN
    x1 xEnde > WHILE
\-----
        x2 x1 - y2 y1 -
        2DUP Pi xel
        y1 2* +
        2DUP Pi xel
        SWAP x1 2* + SWAP 2DUP Pi xel
        y1 2* -
        Pi xel
        x2 y1 - y2 x1 -
        2DUP Pi xel
        x1 2* +
        2DUP Pi xel
        SWAP y1 2* + SWAP 2DUP Pi xel
        x1 2* -
        Pi xel
\-----
        Pyth dY - ABS =: Delta0 \ Fehlerprüfung durchführen
        Pyth dX + ABS dY - ABS =: Delta2
\-----
        Pyth dY - =: Pyth      \ Diff. zum nächsten y^2 abziehen
        dY 2+ =: dY          \ aktuelles y^2 anpassen
        y1 1+ =: y1          \ y inkrementieren

        Delta0 Delta2 > IF Pyth dX + =: Pyth \ wenn der Y-Fehler größer
        dX 2- =: dX          \ als der X-Fehler wird, muß
        x1 1- =: x1          \ x auch angepaßt werden

    THEN
    REPEAT ;
    
```

Forth denkt Windows lenkt

Jörg Plewe, Wolfgang Allinger

Haarzopfer Str. 32; 45472 Mühlheim

Solche Projekte gibt es alle Tage: Ein Gerät ist mit einem Mikroprozessor und serieller Schnittstelle ausgerüstet und wird von einem PC überwacht und/oder gesteuert. Dabei wird das Gerät vom PC meist über ein mehr oder minder abstruses Protokoll angesprochen, das in der Regel vom Gerät bestimmt wird. Schön ist es, wenn dieses Protokoll Forth heißt und der Benutzer freut sich, wenn er neben seiner harten Arbeit mit dem Gerät nebenbei noch 'Tetris für Windows' genießen darf.

Stichworte: Windows Z80 Meßtechnik embedded

Projektbeschreibung

Sachlich geht es darum, das Langzeitverhalten bezüglich des pH-Wertes von Stoffen zu bestimmen, die auf einer Deponie eingelagert werden sollen. Dazu gibt es normierte Tests, in denen eine Lösung des betreffenden Stoffes über eine gewisse Zeit (24h) auf einen konstanten, vorgegebenen pH-Wert titriert wird. Dabei kann es sein, daß nur basisch, nur sauer oder aber beides titriert werden kann. Der pH-Verlauf über die Zeit sowie die zugegebenen

Base-/Säuremengen geben anschließend dem Fachmann Aufschluß über das Langzeitverhalten. Vorgesehen sind auch andere Tests, in denen z.B. neben dem pH-Wert auch Temperatur o.ä. erfaßt werden kann.

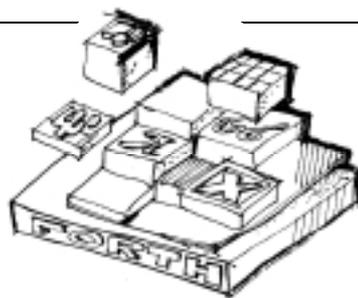
Bei dem Gerät handelt es sich um einen Aufbau, der relativ selbständig bei der Arbeit ist. Es hat die Aufgabe, die chemischen Versuche automatisch an bis zu acht unabhängigen Proben durchzuführen (auch asynchron) und dabei den Versuchsverlauf durch ständige Messungen zu protokollieren.

Der Aufbau besteht aus acht motorisierten Büretten, die, gesteuert von jeweils einem eigenem Microcontroller, ihren Inhalt schluckweise preisgeben

und sich aus einem Vorratsbehälter Nachschub besorgen können. Des weiteren gibt es acht Meßeingänge, die das Signal eines Sensors über einen A/

D-Wandler verfügbar machen. Jeweils eine Bürette mit einem Meßeingang ist auf einem Einbauelement untergebracht,

Die Büretten geben ihren Inhalt in eine Probenflasche ab, wobei aber auch zwei Büretten eine einzige Probe bedienen können. Damit kann das Gerät minimal vier, maximal acht Versuche gleichzeitig fahren.



das als fertiges Bauteil geliefert wird und über eine serielle Leitung verfügt.

Die Büretten geben nun ihren Inhalt in eine Probenflasche ab, wobei aber auch zwei Büretten (eine mit Säure, eine mit Base) eine einzige Probe bedienen können. Damit kann das Gerät minimal vier, maximal acht Versuche gleichzeitig fahren.

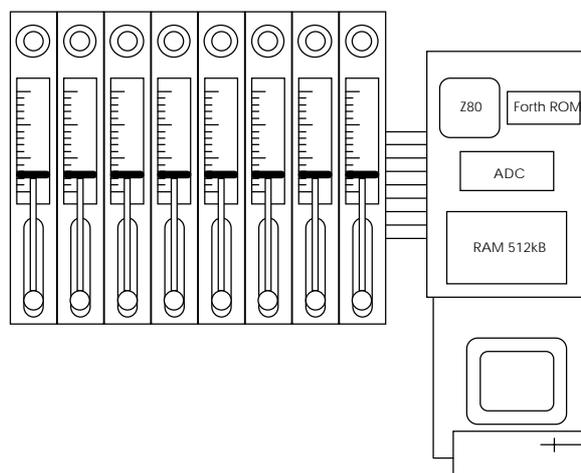
Ihre Kommandos erhalten die Büretten von einem Z80-Microcontroller, der auf einem kommerziell verfügbaren Board untergebracht ist. Dieser hat gleichzeitig die Aufgabe, die Meßergebnisse vom A/D-Wandler zu besorgen.

Die vollständige Regelung sowie das Loggen der Meßwerte werden vom Z80 übernommen. Dieser kommuniziert mit dem PC über eine schlichte serielle Leitung. Versuche werden vom PC aus parametrisiert und getriggert (starten-beenden). Der PC übernimmt die Versuchsergebnisse und stellt sie dar.

Die Konfiguration des Gerätes ist vollkommen wahlfrei. So können beispielsweise die Büretten der Einschübe 3 und 7 in eine Probe titrieren, während der pH-Wert am Meßeingang von Einschub 4 aufgenommen wird.

Vorgeschichte

Das Projekt begann damit, daß das Ingenieurbüro Allinger den Auftrag erhielt, die PC-Software unter Windows (natürlich) zu erstellen. Die Microcontrollersteuerung sollte von einer anderen Firma kommen, die bereits in ein Vorgängergerät wie auch in die aktuelle Steuerung einige Mannjahre investiert





hatte. Nur die bisherige PC-Software, die noch in C++ unter DOS erstellt worden war, sollte restrukturiert und verfenstert werden.

Auch der Microcontroller wurde in C++ programmiert. Da aber 8 Regelprozesse mit asynchroner Kommunikation zu den Büretten und dem PC durchzuführen waren, scheiterten die drei Ingenieure, und zwar vor aller Augen bei den ersten Kommunikationstests mit dem Prototypen unserer neuen PC-Software. Nachdem alle möglichen Fristen abgelaufen waren, erhielt das Ingenieurbüro Allinger den zusätzlichen Auftrag, auch die Steuerung des Gerätes zu übernehmen.

Wahl der Waffen

Allinger ist ein alter Forth-Hase. Damit war klar, womit der Z80 wohl programmiert werden würde. Nach weiteren drei Monaten (Monate! nicht Jahre) waren die neue Hardware samt Software dann soweit, daß man mit ihr etwas anfangen konnte.

Blieb die Frage nach dem PC-Werkzeug. Die Wahl eines Windows-Forthsystems konnte ich so gerade noch abwehren. Warum? Diese Systeme sind für die schnelle Implementation von Dialoganwendungen viel zu dürftig ausgestattet und schirmen den Anwendungsentwickler nicht wirkungsvoll vom Windows-Systemkram ab.

Nach einigen Tests mit Paradox fiel die Wahl schließlich auf Visual Basic 3.0. Diese Wahl hat sich bis zum Ende des Projektes als beinahe ideal erwiesen, denn:

- schnelles Prototyping durch schlichtes 'Malen' der Anwendungsoberfläche
- 'unendlicher' Umfang, der auch sehr einfache Behandlung der seriellen Kommunikation einschließt
- Zugriff auf Access-Datenbanken
- kein Windows-Spezialwissen notwendig
- umfangreiche Stringbehandlung
- alles funktioniert und alles ist gut dokumentiert

Implementation

Die Implementation des Forth-Programms auf dem Microcontroller wird denjenigen, die schon einmal mit so etwas zu tun hatten, nicht unbekannt vorkommen. Es laufen drei Tasks:

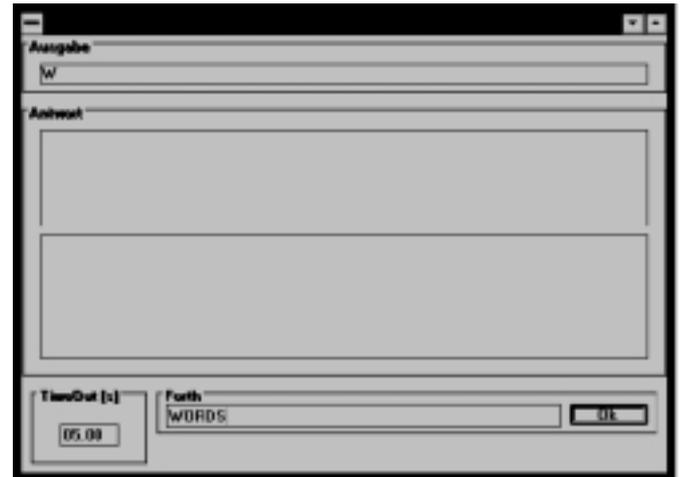
- Kommunikation mit den Büretten
- Datenverwaltung und Regelung
- Kommunikation mit dem PC

Diese drei Tasks verständigen sich untereinander durch Variablenfelder.

Die Kommunikationstask mit dem PC führt nun im Prinzip ein einfaches QUIT, den äußeren Interpreter, aus. Das bedeutet, daß man sich mit einem gewöhnlichen Terminalprogramm an das Gerät anklippen kann und ein interaktives Forthsystem vorfindet. Dies ist insbesondere für Diagnose- und Testzwecke von unschätzbarem Wert (man frage das C-Team, s.o.). Durch die Abteilerung der PC-Kommunikation in eine separate Task kann es auch nicht passieren, daß der PC, der ja irgendwann an den Microcontroller herantritt, diesen unpässlich vorfindet.

Wenn man soweit ist, bedeutet die Implementation des PC-Programms nur noch, eine grafische Oberfläche für ein Forthsystem zu entwerfen.

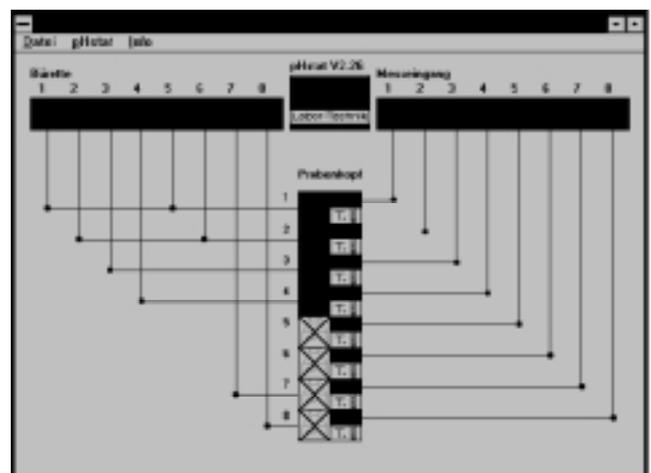
Dabei kann man voll aus dem Potential der Windows/Visual Basic-Möglichkeiten schöpfen. So gibt es denn z.B. anstelle von Schleifen, die alles mögliche pollen, schlicht für jeden Einschub einen Timer, der asynchron zum Rest der Anwendung die Daten für diesen Einschub abfragt. Damit



das keinen Salat gibt, definiert man abgeschlossene Kommunikationseinheiten (Kommunikationstransaktionen), die von einer zentralen Prozedur abgearbeitet werden. Wie schön, wenn diese Prozedur ihre Arbeit in einem Fenster protokolliert. Wenn dieses Fenster dann auch noch interaktiv ist..

Man hat hier mit der Anwendung beim Test die Möglichkeit, interaktiv in den Ablauf des Forth-Programms einzugreifen, Variablen zu erfragen und zu setzen, usw. Wenn das das C-Team gesehen hätte...

Da nun alles in Forth-Hochsprache geschieht, stehen auf der Seite der Anwendung Tür und Tor offen. Wenn man etwas implementieren möchte, auf das das Microcontrollerprogramm nicht vorbereitet ist, baut man einen entsprechenden Forth-String zusammen und schickt ihn zum Controller. Somit kann man besondere und plötzliche Kundenwünsche 'on the fly' erfüllen, ohne ein neues Eeprom brennen zu



müssen. Selbst spätere Updates der Firmware lassen sich z.T. so durchführen, ohne ein neues Eprom liefern zu müssen.

Auf der anderen Seite stehen für den Anwender ohne großen Implementationsaufwand viele Windows-Funktionalitäten wie gewohnt zur Verfügung, da solche Standarddinge von VisBas hervorragend abgedeckt werden:

- Konfiguration des Gerätes mittels Drag&Drop
- Fließender Übergang zu z.B. Excel oder Access mittels DDE
- Drucken auf beliebigen (Farb-) Druckern
- Kontextsensitive Hilfe

Für den Anwendungsentwickler, der den Wert einer relationalen Datenbank zu schätzen weiß, ergeben sich einfache und weitreichende Möglichkeiten. So werden in unserem Projekt sämtliche Versuchsparameter sowie die gesamte Gerätekonfiguration in Access-Tabellen abgelegt. Damit ist automatisch die Gewähr gegeben, daß auch der Netzwerkbetrieb in einer Folgeversion funktionieren wird.

Außerdem bieten die großen Pakete:

- raffiniertes Source-Management
- einfache Grafikerstellung für Bildschirm und Drucker
- viel Schnick-Schnack wie 3D-Effekte, Spin-Buttons, skalierbare Bilder, ...
- vollständige On-Line-Hilfen, die einen wirklich nicht im Stich lassen
- Installationshilfen für die fertige Anwendung

Emulation

Da so ein Gerät in der Entwicklung meist nur einfach vorhanden und zudem auch noch groß und schwer ist, mit Allinger und mir aber schon zwei Leute damit befaßt waren, stellte sich die Frage nach einer Emulationsmöglichkeit.

Hier zeigte sich dann der enorme Wert eines Standards für Forthsysteme. Die Applikation für den Microcontroller besteht aus ein paar hundert Screens LMI-Forth. Hier war also im wesentlichen nichts anderes zu tun, als die Anwendung schön ordentlich vom Kern zu separieren und den ganzen Rest mit

ein wenigbedingter Kompilation unter PC-Forth ablaufen zu lassen. Schon hat man die Emulation fertig. Besser noch. Wenn man das PC-Forth dazu bringt, seine Ausgabe auf die serielle Schnittstelle und gleichzeitig auf den Bildschirm zu machen, kann man sogar wunderbar ein Protokoll dessen, was da so zwischem dem PC und vermeintlichen Controller abläuft, auf dem Schirm mitlesen.

Fazit

Jedem das seine. Auf dem Microcontroller war Forth ganz klar das Mittel der Wahl. Als interaktives Multitasking-system ist Forth gegenüber C einfach übermächtig.

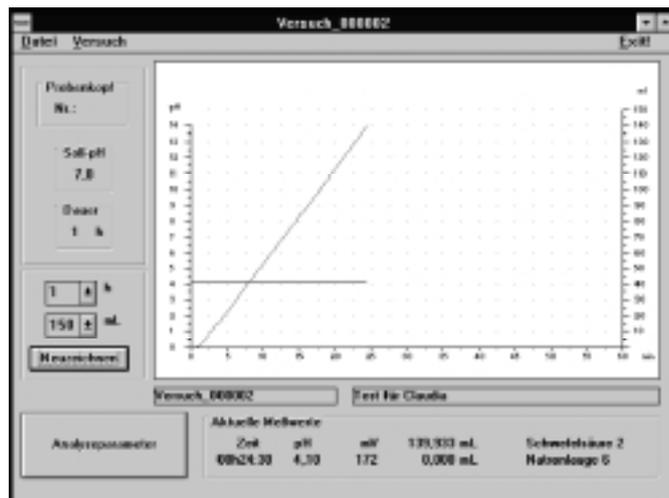
Auch Windows-Anwendungen ohne zwingenden Grund (z.B. weil man den Interpreter WIRKLICH braucht) in Forth zu erstellen, erscheint mir mit Sicht auf den ungeheuren Leistungsumfang moderner Entwicklungswerkzeuge von großen Firmen wie ein Schuß ins eigene Knie.

Um es klar zu sagen: gegen ein VisBas für 400,- sehen professionelle Forth-Umgebungen für 1000,- wie nicht registrierte Sharewarespielzeuge eines Garagenhackers aus. Angeblich hat Microsoft auch mehr Manpower zur Verfügung als die meisten Forth-Anbieter.

Man kann an dieser Stelle einwenden: ... wenn ich aber dies und jenes machen will dann geht das in VisBas aber auch nicht mehr so einfach und in Forth schon ...

Mag sein. Aber 99% der Praxisprobleme sind nicht 'dies und jenes' sondern nur 'normales' Zeug.

Es soll auch nicht verschwiegen werden, daß ein schneller 486 oder besser Pentium das geeignete Gerät für die Entwicklung ist. Ein 386SX macht jede Produktivität des Werkzeuges wieder zunichte.



Auf dem Controller gibt es keine Alternative. Hier übersteigt der Wert seines Nutzens den Kaufpreis des Metacompilers um ein vielfaches, besonders in Relation zu anderen Methoden.

Es hat sich gezeigt, daß die Kombination der beiden Werkzeuge, die jeweils auf ihrem Gebiet optimal erscheinen, durch die Unkompliziertheit und Stabilität eines ROM-Forth-Systems erst ihre gemeinsame Leistungsfähigkeit entwickeln. Das stärkste Windows-Entwicklungssystem wird nichts nutzen, wenn man sich auf die Gegenseite nicht verlassen kann.

Bei diesem Projekt übernimmt Forth die eigentliche Arbeit und führt sie autark und autonom durch. Das System funktioniert ganz prima auch ohne den PC. Um den PC anzuschließen, wird ein normaler Interpreter als eigene Task gestartet, was für den eigentlichen Arbeitsablauf der Steuerung aber ohne Bedeutung ist. Die Visualisierung der Ergebnisse und die ganze damit verbundene, asynchrone Kommunikation, die das ursprüngliche C-Programm zum Scheitern verurteilt haben, tangieren die Forth-Software nur peripher. Auf der PC-Seite kann man mit einem gewöhnlichen Forthsystem sprechen, ohne sich hier um die ganzen Regelabläufe kümmern zu müssen. Damit ist hier eine ganz strenge Trennung zwischen dem operativen Teil und der Oberfläche vollzogen: Forth denkt - Windows lenkt





Das war die Forth Tagung '95

von Jörg Plewe

Haarzopfer Str. 32; 45472 Mühlheim

Für viele Mitglieder der Forth Gesellschaft immer wieder der Höhepunkt des Jahres: Die Forth-Tagung. Mitglieder und Nichtmitglieder treffen sich und lauschen interessanten Vorträgen. Davor, dazwischen und danach wird auf Teufel-komm-raus kommuniziert, wobei so manche Zunge auch mit bewährten Hilfsmitteln gelöst wird. So war es jedes Jahr und so war es auch in diesem Jahr. Am 21. und 22. April in Berlin. Die Mitgliederversammlung der Gesellschaft hatte sich 1994 diesen Tagungsort gewünscht. Bernd Hinze und Helfried Schürer haben es dann möglich gemacht: Sie fanden ein Tagungshotel im Südosten Berlins. In dem Ortsteil, der einst durch einen schelmischen Hauptmann verunsichert wurde, konnte nun die Gesellschaft idyllisch am Dämmeritzsee tagen und die ersten Mücken des Jahres genießen.

Stichworte: Forth-Tagung Tagung Berlin

Mit nur geringer Verspätung ging es am Freitag Nachmittag los.

Dr. Möller präsentierte eine praxisnahe Forth-Applikation, die in Arztpraxen unter Windows zu Werke geht. Neben dem Aufzeigen einiger typischer Probleme des Aufgabengebietes, wie etwa der Zertifizierung des Abrechnungs- und Chipkarten-Systems, behandelte der Vortrag die objektorientierten Entwurfstechniken sowie die allgemeinen Probleme der Entwicklung mit Forth unter Windows. Das verwendete proForth für Windows des englischen Anbieters MPE, erst kürzlich in der Vierten Dimension vorgestellt, scheint noch etwas unter großen Projekten und unzureichender Windowseinbettung zu lei-

den. Letztendlich war es aber interessant zu sehen, wie Forth uns allen unvermittelt im Alltag begegnen kann.

Ebenfalls aus dem Medizinischen berichtete anschließend Claus Vogt in seinem von einem Video eingeleitetem Vortrag. Diesmal allerdings weniger von der organisatorischen, als vielmehr von der technischen Seite. Ein winziger Sender in Hüftgelenksprothesen übermittelt Daten von Dehnungsmeßstreifen und eines Temperatursensors aus dem Inneren der Patientenhüfte. Aufgenommen, ausgewertet und visualisiert werden diese Daten von einer umfangreichen Forth-Software, die erstaunlicherweise zum einen Teil in PC-FORTH und zum

anderen in F-PC erstellt wurde. Beide Forthsysteme kommunizieren über eine festgelegte Schnittstelle. Tests haben ergeben, daß die dabei wichtige Grafikperformance beim PC-FORTH besser ist als in einem QBASIC-Programm, aber von F-PC noch einmal um einen Faktor 5-10 überboten wird.

Wichtig für alle Mitglieder, die in Zukunft besonders kommunikativ sein wollen war der Beitrag von Holger Petersen und Ulrich Hoffmann. Holger ist der Sysop unserer Mailbox und die beiden haben versucht, einen Überblick zu geben, was man wie über die Mailbox erreichen kann, was nicht geht und wie man es noch machen könnte. Ein wichtiger Punkt dabei: Durch eine von Holger Petersen auf Anfrage verschickte Starter-Diskette wird der Einstieg in die elektronische Kommunikation sehr leicht gemacht. Da das Netzwerk eine sehr schöne Plattform für Diskussionen aller Art darstellt und hier auch echtes Vereinsleben stattfindet, erging der Aufruf an alle Anwesenden, es doch einmal zu versuchen. Ca. 10 spontane Interessenten wurden gefunden, so daß wir im Netz wohl bald neue 'Gesichter' sehen werden. Nicht ganz unwichtig war auch der Hinweis, daß der Netzzugang, der im Mitgliedsbeitrag ja enthalten ist, einer der günstigsten ist, die man überhaupt bekommen kann.

Die erste kontroverse Diskussion der Tagung löste Klaus Kohl aus, als er in seinem Vortrag die Erstellung einer Forth-CD anregte. Die vielfältigen Meinungsäußerungen zu diesem Thema beleuchteten viele Aspekte vom Sinn eines solchen Projektes über mögliche Vertriebs- und Finanzierungsformen bis zur technischen Machbarkeit. Klar wurde dabei nur eines: Ohne eine Person, die sich dieses Projektes ernsthaft annimmt, wird es nicht gehen.

Anschließend wurde es wieder etwas technischer, als Hr. Stelbe über CPU-unabhängigen Forthcode referierte. Aus der gestellten Anforderung konnte er einige grundsätzliche Punkte ableiten, die von einem solchen System zu erfüllen sind. So kann es sich nur um eine ITC- oder TTC-Implementierung handeln, da echte Maschineninstruktionen um jeden Preis zu meiden sind. Der Instruktionsteil wird jeweils maschinenabhängig dem eigentlichen

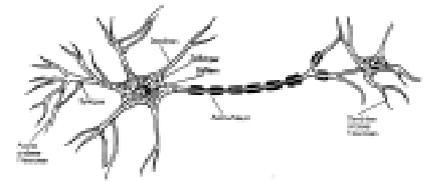
Fadencode zur Seite gestellt. Über dieses Thema wurde dann auch heftig diskutiert, wobei nun endlich auch die Gurus so richtig vom Leder ziehen konnten und den einen oder anderen sicher abgehängt haben (deshalb kann ich hier die Diskussion auch nicht so richtig darstellen).

Doch zum Glück war nun der Abend bereits erreicht und in kleineren Gesprächsrunden konnten nun die Wissenslücken aufgefüllt werden. Nach dem vorzüglichen Abendessen war deshalb noch lange nicht Schluß und der letzte Diskussionsdrang erlosch erst gegen 3:30h am Morgen, wobei man um diese Zeit noch unseren ortsansässigen Editor über ein Baugerüst aus dem rundum abgeschlossenen Hotel entkommen sah.

Damit die Pausen nicht zu lang wurden, ging es pünktlich um 9:00h schon wieder weiter. Dr. Egmont Woitzel stellte das fieldForth der Firma FORTECH vor. Dieser fast einstündige Vortrag führte den Zuhörer zunächst einmal gründlich in die allgemeinen Probleme der Entwicklung für embedded Controller und der Crosscompilation ein. In dem von ihm dann beschriebenen Lösungsansatz erfolgt eine möglichst weitgehende Trennung des Systems in Target- und Hostteile, wobei jedes an

seinem Platz verbleibt und trotzdem für den Entwickler der Eindruck eines monolithischen Entwicklungssystems entsteht. Durch den modularen Aufbau des Systems lassen sich auch zentrale Komponenten wie die CPU austauschen. fieldForth ist zur Zeit für die Targets TMS320C40 und M68332 implementiert. Andere Targets sollen jeweils innerhalb einiger Wochen hinzuzufügen sein.

Performanceprobleme von stackbasierten Architekturen, zu denen ja auch Forth gehört, haben Bernd Paysan von der TU München keine Ruhe gelassen. Der Stack erweist sich als Flaschenhals und behindert die Parallelisierung von Instruktionen auf der CPU. Seine Lösung: 4 Stacks. Bernd stellte einen möglichen Chip auf Basis dieser Architektur vor und erläuterte dessen Instruktionssatz. Am Beispiel einer FFT erklärte er die Wirkungsweise der Maschine und machte klar, daß ein solcher Chip in bestimmten Anwendungsbereichen tatsächlich sehr schnell sein würde. Wer sich näher für das Modell interessierte, konnte sich auch einen Assembler und einen Simulator vorführen lassen. Bernd schätzte den Realisierungsaufwand mit etwa 300.000 Gattern ab und sieht die Stärken der Architektur im



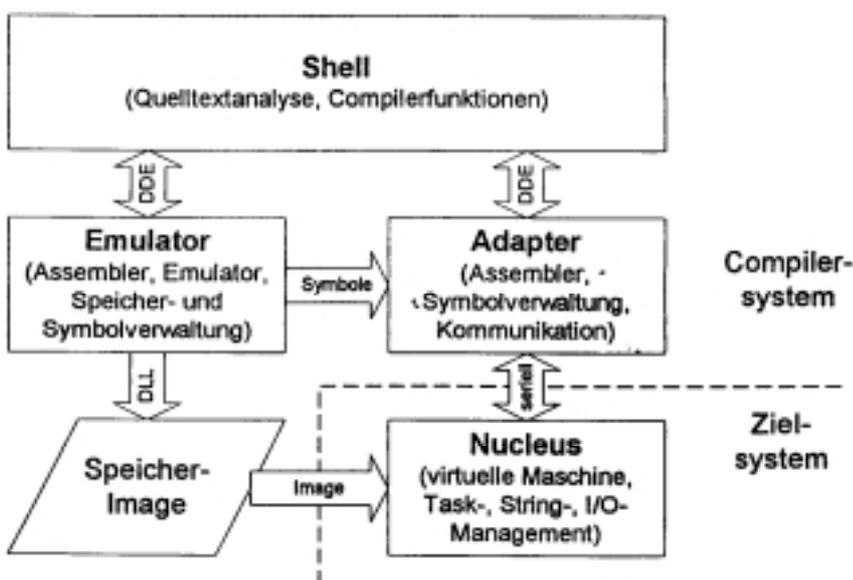
Eine Nervenzelle (Neuron)

DSP-Bereich oder etwa beim Z-Buffering in der 3D-Grafik.

Allgemeines zu Controllern, Memory und Peripherie konnten wir von Prof. Thomas Beierlein lernen. Nach seinem geschichtlichen Überblick zu diesem Thema leitete er zur Vorstellung eines modernen Entwicklungssystems für den Controllerbereich über. Holon, die Entwicklung des Schweizer Gesellschaftsmitgliedes Wolf Wejgaard stellt eine integrierte Umgebung dar, die sich insbesondere der Quelltextorganisation widmet. Menügeführt läßt Holon den Entwickler durch eine Vielzahl von Modulen navigieren. Und das nicht nur in der Theorie, wovon man sich in einer der Kaffeepausen selbst überzeugen konnte.

Mit den ganz modernen Softwaretechniken hat sich Helfried Schürer befaßt und uns auf der Tagung an seinen Erkenntnissen teilhaben lassen. Neuronale Netze sind immer wieder ein Stichwort in fast allen Zeitschriften und Magazinen. Nachdem Helfried die Begriffe in ihrer biologischen Bedeutung klar gemacht hat, zeigte er ein System, mit dem die Entwicklung neuronaler Netze komfortabel und flexibel möglich ist. Auch dieser Vortrag löste eine rege Diskussion aus, die sich im wesentlichen um die Vor- und Nachteile des Einsatzes von Floatingpoint-Arithmetik rankte. Nicht nur damit hat dieser Vortrag sicher zur Allgemeinbildung des Publikums beigetragen.

Um die Lücke bis zum Mittagessen zu füllen, durfte ich selbst etwas über Forth und Datenbanken im allgemeinen und später über eine angefangene Implementation im speziellen berichten. Da typische Forth-Anwendungen den Da-



Das fieldFORTH der FORTECH-GmbH trennt strikt zwischen Host- und Targetcode



tenbankbereich meist nicht tangieren, war auch hier eine gewisse allgemeinbildende Komponente vorhanden.

Das Mittagessen dürfte manchen knapp vor dem Kollaps gerettet haben... Auch das sollte erwähnt werden: Der nahe See und das erstklassige Wetter machten es etwas schwer, die Versammlung nach dem Essen wieder einzufangen!

Von vielen wird es benutzt, kaum

und damit offenzulegen. Damit sollen Fallen vermieden werden und der Programmierer soll schnell entscheiden können, ob er ein Modul von der Stange verwenden kann.

Rainer Hansen von der Universität Duisburg stellte das avisierte Thema seiner Diplomarbeit vor: Eine portable Textverarbeitung in ANSI-Forth. Damit sollen die Möglichkeiten portabler Forth-Programmierung u.U. mit grafi-

Mischung von Forth und C/C++-Programmen besteht, die sich kreuz und quer gegenseitig aufrufen können. Dies wird durch Namenslisten für offene und exportierte Referenzen der einzelnen Programmodule ermöglicht, mit deren Hilfe beim Booten das Programm erst zusammengelinkt wird. Auch dieses System konnte man in der Praxis bestaunen. Interessanterweise wird 'Forth' gegenüber den Käufern des Gerätes nicht genannt, sondern als 'residente Instruktionsbasis RIB' bezeichnet.

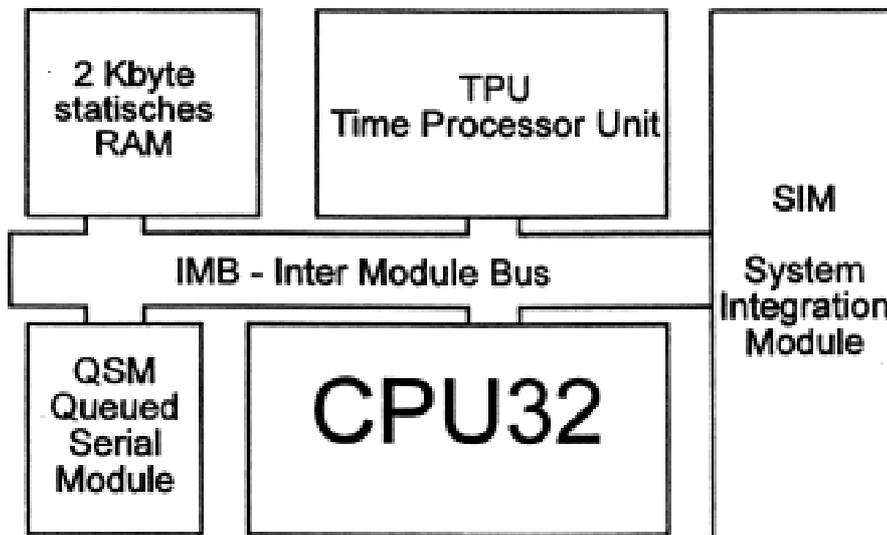
Als einen Teil des fieldForth-Systems von FORTECH ging Malte Köller speziell auf die Verwirklichung auf dem M68332-Controller ein. Dabei gab es besonders vieles über den Controller selbst zu lernen. Dieser Beitrag löste dann eine tiefgreifende Diskussion über die Frage "Wieviele Stackeinträge sollten in Registern gehalten werden?" aus.

In seinem abschließenden Doppelvortrag berichtete Ulrich Hofmann kurz über seine Portierung des F68KANS-Systems auf den Mac bevor er sich der Problematik eines Terminplaners zuwandte. Etwas die auf dem Markt befindlichen, sehr aufwendigen Windows-Terminplaner verhöhrend zeigte er eine verblüffend einfache Implementation eines Terminplaners in Forth. Wir ließen uns überzeugen, daß ein paar einfache Definitionen, die das Forth-BLOCK-Interface nutzen, wohl viele aufwendigere Systeme an Funktionalität übertreffen. Ullis Fazit allerdings: Auch seine Lösung kommt an die Nützlichkeit eines kleinen Terminheftchens aus der Apotheke nicht heran.

Damit hatten wir alle Vorträge hinter uns und nach dem Abendessen stand Fete auf dem Programm. Da Forthler scheinbar auch gerne mit Maschinen feiern, wurde es ein sehr fröhlicher Abend, an dem noch viel über die gehaltenen Vorträge geredet wurde. Wieviel es zu Reden gab mag man daran ermesen, daß einige die Mitgliederversammlung am Sonntag erreichten, ohne zwischenzeitlich ihren Zimmerschlüssel gesucht zu haben.

Alles in allem: Die Tagung war super und wer nicht dabei war, hat etwas verpaßt!

Der MC68332 besteht aus folgenden Modulen:



Blockdiagramm des MC68332

jemand hat es durchschaut: Das Hypertextsystem des F-PC. Doch das hat sich jetzt geändert, denn Claus Vogt hat es uns erklärt! Und nicht nur das. Er hat gezeigt, wie man das System über seine ursprüngliche Hilfe-Funktion hinaus in eigenen Applikationen gewinnbringend einsetzen kann und praktische Beispiele dafür vorgestellt: Aus der Online-Dokumentation eines von ihm verfaßten Anwendungsprogramms konnte er einige Teile seiner Diplomarbeit generieren! Für die Zukunft schwebt ihm die Organisation eines Forth-Archives mittels des Hypertextsystems vor.

Real programmers don't read manuals! Auch wenn das stimmt, der ANSI-Standard verlangt gewisse Formen der Dokumentation. Ulrich Hoffmann hat sie uns dargestellt. Der Sinn besteht vor allen Dingen darin, Abweichungen und Schwächen wirklich zu dokumentieren

schers Oberfläche untersucht werden.

Ein weiterer Vortrag aus der FORTECH-Reihe von Udo Schütz beschäftigte sich der automatischen Generierung von Glossaren aus Forthprogrammen. Und damit das nicht zu einfach wird, soll ein solches Glossar mit der Windows-Hilfefunktion zugänglich sein. Udo hat den Weg beschrieben über die Erzeugung von RTF-Dateien und Help-Compiler. Besitzer des comForth-Paketes für Windows können es einfach ausprobieren, die anderen gucken in die Röhre.

Schnelle Spezialhardware zur Echtzeit-Farbbildanalyse hat Hr. Höhenleitner vorgestellt. Es basiert auf dem DSP TM320C40 auf dem Forth für die Inbetriebnahme und als eine Art Basisbetriebssystem eingesetzt wird. Das besondere an diesem System ist, daß die Applikationssoftware aus einer bunten

Brief aus der Provinz

von Friederich Prinz

f.prinz@mhb.gun.de; Homberger Str. 335; 47433 Moers



Ich habe anlässlich der Forthtagung den SWAP-Drachen der FG verliehen bekommen, was ich wohl als Auszeichnung akzeptieren soll - und auch versuchen will. Das fällt mir allerdings so schwer, wie es mich getroffen hat, als Jörg Plewe in diesem Zusammenhang nach einer langen Aufzählung vermeintlicher Vorzüge und Verdienste meiner Person meinen Namen genannt hat. Ich möchte an dieser Stelle noch einmal versuchen, meine Haltung deutlich zu machen. Zunächst bin ich grundsätzlich gegen jede Art von Ehrung im Sinne einer Hervorhebung. Eine meiner Lebenserfahrungen - besser: immer wieder bestätigte Beobachtung - sagt mir, daß die Aussicht auf Auszeichnungen jeglicher Art den größten Teil unserer Mitmenschen immer wieder aufs Neue zu Handlungen und Verhaltensweisen anstachelt, die ich für mich selbst, aber auch für die Menschen meiner 'näheren Umgebung' kompromißlos ablehne. Gleichzeitig kann ich nur wiederholen, was ich schon in unzähligen Gesprächen vorgetragen habe: Ich bin nur einer von mehreren FORTHern der Moerser Grup-

pe. Ich 'führe' diese Gruppe nicht. Die Moerser Gruppe benötigt keine Führung. Wenn ich unter den Moerser FORTHern die Stellung eines "Ersten unter Gleichen" einnehme, dann höchstens deshalb, weil ich in der zeitlichen Abfolge der Moerser Forthgeschichte einer der Ersten war. Die zugegebenermaßen recht große Menge an Arbeit, die ich in den ersten Jahren in die Moerser Gruppe investiert habe, kam zuallererst mir selbst zu gute. Heute, da ich mich aus beruflichen Gründen ein wenig aus der allwöchentlichen Arbeit zurückziehen mußte, leistet Michael Major diese Arbeit weiter. Und in ein oder zwei Jahren werden andere diese Arbeit übernehmen. Ich habe den SWAP-Drachen aber gerne stellvertretend für die ganze Moerser Gruppe angenommen. Ich sehe darin eine Anerkennung der 'Art', in der wir hier am Niederrhein FORTH auffassen, FORTH betreiben und versuchen, so viel Spaß wie möglich durch FORTH zu gewinnen und an FORTH zu vermitteln.

Einen Bericht über die FORTH-Tagung hat Jörg Plewe in die VD gegeben. Da kann ich mir einen solchen Bericht sparen. Vermutlich fallen Michael's und meine Beschreibung, die wir unseren Moerser Freunden geben werden, ohnehin ganz anders aus. Wir sind voll von den Eindrücken einer großen Stadt, die sich lebendig und locker - fast fröhlich - daran gemacht hat, die Spuren einer wenig geliebten Vergangenheit zu beseitigen, und gleichzeitig zu bewahren, was erhaltenswert ist. Wir sind auch voll mit Bildern einer überaus forthigen Heimreise. Diese war nämlich gekennzeichnet durch den Versuch, einmal mehr 'in das Kleinste' zu sehen. Das heißt, daß wir auf Landstraßen aus

Vielleicht liegt das 'Besondere' unserer Gruppe einfach darin, daß wir uns öfter

Berlin herausgefahren sind. Wir haben jede Straße gemieden, die auch nur nach einer Bundesstraße aussah. Dabei sind wir 'übers Land' gekommen, durch eine wunderschöne Landschaft und durch Dörfer Brandenburgs und Sachsen-Anhalts. "Just do it" - eines der schlagkräftigsten Argumente für FORTH selbst - hat uns dabei als Motto gedient, und uns bis Wittenberg geführt. Erst danach haben wir mangels Tageslicht wieder die Autobahn aufgesucht. Die Forthtagung war schon 'nicht schlecht'. Die Bilder aus Berlin und von der Heimfahrt waren mehr als nur eine Reise wert.

Trotzdem noch einige Worte zur Tagung selbst: Mir haben einige Menschen dort gefehlt! Mit dem Hannes Teich, der mir in meiner Anfängerphase mit vielen wertvollen Tips und Anregungen immer wieder 'auf die Sprünge' geholfen hat, hätte ich ebenso gerne ein Bier getrunken, wie mit dem Jörg Staben, der uns in Moers in fröhlicher Erinnerung ist. Letzteres gilt auch für den 'bärgen' Klaus Schleisiek (den aus Aachen), der unseren Tag der offenen Tür mit seinem 1·1 m² LED-Feld bereichert hatte. Dafür habe ich andere Menschen endlich persönlich kennengelernt. Mit dem Claus Vogt stehe ich seit einigen Monaten in recht intensiver DFÜ-Verbindung. Und ich bin froh, daß



Jörg Plewe links im Bild, einer der drei DirektorInnen der Forth-Gesellschaft
verleiht Friederich Prinz rechts im Bild, einziger Prinz der Forth-Gruppe Moers
den Swap-Drachen ganz klein im Bild, höchste Auszeichnung der Forth-Gesellschaft



sich der sympathische Eindruck auch vis a vis bestätigt hat. Leider war insgesamt viel zu wenig Zeit, auch wenigstens einige der anderen, interessanten Menschen kennen zu lernen.

Ich wurde während der Tagung mehrfach darauf angesprochen, daß 'die Moerser' offenbar mehr Phantasie hätten, als die FORTHer in anderen Städten. Das kann ich nicht glauben. Ich kann auch nicht glauben, daß anderenorts Gruppen nicht 'funktionieren' könnten, wenn sie so unorganisiert und 'anarchistisch' wären wie die Moerser. Hinter dem Funktionieren der Moerser Gruppe steckt auch kein Geheimnis. Vielleicht liegt das 'Besondere' unserer Gruppe einfach darin, daß wir uns öfter treffen - eben wöchentlich. Da kommen wir, wenn wir die Ferien ausnehmen, leicht auf 45 Treffen im Jahr. Da wächst man natürlich schneller und intensiver zusammen, als wenn die Anzahl der Treffen auf 10 - 11 Mal pro Jahr begrenzt bleibt. Das allwöchentliche

Treffen ist für uns längst zu einem festen Bestandteil des wöchentlichen 'Alltags' geworden. Das heißt natürlich nicht, daß Jeder immer dabei ist. Aber Irgendjemand ist eigentlich fast immer da ... Dazu mag kommen, daß wir unsere Gruppe nicht als 'Freizeit-Dienstleistungs-Betrieb' ansehen. Niemand kommt zu uns um nur zu konsumieren. Jeder bringt sich selbst ein, mit seinen Fähigkeiten, mit seinem jeweiligen Können und mit seinen Ideen. Dabei sind diese Ideen keineswegs auf FORTH beschränkt. Wenn ich einen interessanten Vortrag ankündige, dann kommen Viele. Wenn Irgendjemand für den nächsten Samstag selbstgebackenen Kuchen verspricht, dann fehlt Niemand. Natürlich steht FORTH bei uns 'ziemlich vorne'. Aber noch vor FORTH kommt der Spaß - der Spaß an FORTH, Spaß an Technik allgemein, Spaß am Entdecken und - vor allem - Spaß an uns selbst. Dabei dokumentiert sich der 'Spaß an uns selbst' nicht zuletzt an den

vielfältigen Aktivitäten, die wir gemeinsam oder in kleineren Gruppen außerhalb der Forthgruppe entwickeln.

In der letzten VD habe ich berichtet, daß wir uns mit dem Gedanken tragen, ein 'Niederrheinforum FORTH' ins Leben zu rufen. Ich hatte dazu aufgefordert, sich bei mir zu melden, wenn Interesse an einem solchen Forum besteht. Bis jetzt haben nur diejenigen 'Nichtmoerser' Interesse bekundet, die uns ohnehin zumindest sporadisch besuchen. Das ist nicht weiter schlimm. Wir können das Forum ohnehin erst im kommenden Jahr realisieren. Die beruflichen Einbindungen unserer Freunde lassen die Organisation für dieses Jahr nicht mehr zu. Aber es wird stattfinden - vermutlich im Spätsommer '96.

Das war's für dieses Mal aus der Provinz.

Glückauf
Friederich Prinz



Forth inside:

Microsoft C intern

Aus einer Produktbesprechung zu Microsoft C/C++ 7.0:

'Improvement in code size ... continue with C7's use of p-code... P-code is an old technique for reducing the size of executables. ... the compiler can optionally generate object code for a virtual machine, together with an interpreter to emulate the virtual machine. C7 packs code into a stackbased language that abstracts many of the details of 80x86 assembly language...p-code can reduce code-size to 50 percent in some cases... Such p-code has apparently been used in th user-interface code of Microsoft Word, Excel and Project...P-code is ...new technology in C7...'

jrd (Rafael Deliano) nach BYTE 9/92 S.274

Forth-Prozessoren:

Marc4: ASIC-Kern aus Chemnitz

In der heutigen Diskussion in c.l.f. fand ich einen Hinweis auf einen Forth-Controller MARC 4, der so 1978 herum in der VD schon einmal vorgestellt wurde. Hier in Chemnitz hat man im letzten Jahr daraus einen ASIC-Kern gemacht. Ich habe die Unterlagen da, vielleicht hat jemand Interesse.

Thomas Beierlein, tb@kbbs.org, März 1995

ANS:

DPANS94 im WWW

I have uploaded a new HTML version of the DPANS94 document to Taygeta.oc.nps.navy-

.mil. The contents are identical to the original posting. The new version, V2.0, has many more hyperlinks. Specifically, all (well, at least most :-)) of the See: references are now linked. Anchors have been added for each section in my working copy. This will facilitate filling in many additional links in the next version of the document.

I plan to start the next round of adding new hyperlinks after the next release of Until. You can access the new HTML document from either the FIG home page or the Forth page at <http://taygeta.oc.nps.navy.mil/forth.html>.

[dpANS94 ist der letzte Proposal zum ANS und frei erhältlich, z.B. bei Klaus Kohl /c/v]

Enjoy!

Norm Smith

(comp/lang/forth, 5.4.95)

ANS:

Stand4th

I have uploaded my Standardizer to Taygeta, in the ANS subdirectory, under the name Stand4th.ZIP. I'd welcome comments about how it should be changed.

This is an ANS standard program with some environmental dependencies. It adds a variety of error-checking to standard compilers, including type-checking, arithmetic-overflow checking, and array-bounds-overflow checking, all at run-time. It offers more comprehensivetesting than Joerg Plewe's elegant debugger (and is also much less flexible).

It warns about many things that might be

perfectly OK on your standard compiler, but which would fail on a minimal standard compiler, or a 1's-complement one, etc. However, some warnings are dependent on your own compiler, particularly things that depend on the cell-size:char-sizeratio.

comp/lang/forth, 14.4.95
Forth-Systeme

Forth-Systeme:

pfe0911 uploaded

Halloechen, ich denke Ihr versteht mein pidgin english :-)

I just uploaded a new version of the portable forth environment to several sites:

roxi.rz.fht-mannheim.de:/pub/languages/forth
taygeta.oc.nps.navy.mil:/pub/incoming/forth
ftp.cygnus.com:/incoming
sunsite.unc.edu:/pub/Linux/Incoming/pfe-0.9.11.tar.gz

Except on sunsite there are 4 files:

pfe-0.9.11.tar.gz-- sources only
pfe-0.9.11.bin.tar.gz
-- DOS and OS/2 binaries
· pfe0911.zip -- sources only for unzip
· pfe0911b.zip
-- DOS and OS/2 binaries for unzip

Changes vs. 0.9.9 include: Better configuration, bug fixes and additions, better file access, support for Watcom C added.

Hope everything works. Dirk
dusz@roxi.rz.fht-mannheim.de 16.4.95
de/comp/lang/forth

Meßtechnik mit dem PC

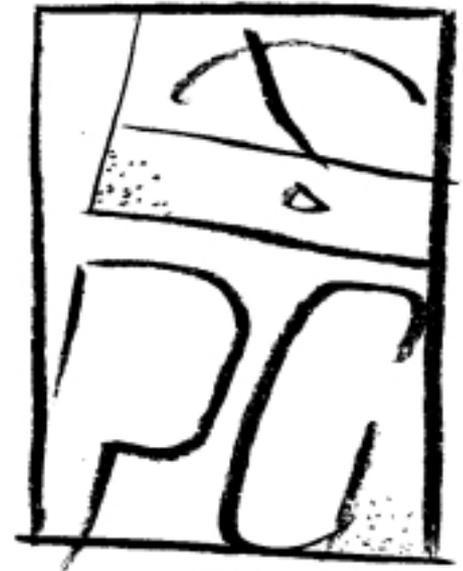
Autor: Klaus Kohl / BMC Systeme GmbH

Zeppelinstr. 10; 86406 Mering; Tel.: 08233 / 30 524

Der Autor entwickelt für die Firma BMC Systeme GmbH unter anderem PC-Meßkarten mit entsprechender Software. Einige der dabei ermittelten Möglichkeiten und Probleme soll in der hiermit begonnenen Serie erläutert werden. Ziel dabei ist, einen vorhandenen PC mit möglichst geringen Aufwand zu einem Meß- und Steuercomputer auszubauen. Der erste Artikel dient dabei als Übersicht und behandelt in einem Rundumschlag die Vor- und Nachteile der später verwendeten Schnittstellen. Außerdem werden noch einige bei Erstellung von Meßprogrammen erkannten Probleme aufgeführt. In den folgenden VD's wird das Thema Meßtechnik und PC mit folgenden Inhalten noch ausführlicher behandelt:

- * Beschreibung der PC-Schnittstellen und dessen Programmierung
- * Bau einfacher Hardware zur Nutzung der Schnittstellen
- * Informationen zur Verwendung verfügbarer Meßsysteme
- * Tools für eigene FORTH-Programme (Hardwareansteuerung und Auswerteprogramme)

Stichworte: Messen Hardwareansteuerung PC



1. PC-Schnittstellen

Schon in der Grundausstattung wird ein PC mit vielen Schnittstellen geliefert. Darunter fallen vor allem der Druckerport oder die serielle Schnittstelle. Weitere Ein-/Ausgabekanäle erhält man gegen geringen Aufpreis (z.B. Joystick) oder wurden im Rahmen der Multimedia-Anwendung integriert (z.B. Soundblaster). In diesem Kapitel liegt der Schwerpunkt der Betrachtungsweise hauptsächlich auf die unmittelbare Verwendbarkeit solcher Schnittstellen für Ein- und Ausgabe.

1.1. Vor- und Nachteile

Die Tabelle führt die bei üblichen PC's verfügbaren Schnittstellen und dessen Vor- und Nachteile auf.

1.2. Einsatzmöglichkeiten

Jedes der vier oben angegebenen Schnittstellen hat für die Meßtechnik ihre Hauptanwendungen auf verschiedenen Gebieten. Hier sind nur einige der Anwendungen aufgelistet:

Druckerport:

- * Direkte Ansteuerung von Schrittmotoren über entsprechende Treiberbausteine
- * Ansteuerung von seriell ladbaren Bausteinen wie:
- * D/A-Wandler

- * Sinusgeneratoren
- * Filter
- * Abfragen von seriell auslesbaren A/D-Wandler
- * Schnelles Übertragungsmedium zwischen PC's

	Vorteile	Nachteile
Druckerport	Normierte Steckerbelegung Normierte (TTL-)Pegel 8+3Bit als Ausgang 5 Bit als Eingang Hohe Ausgabefrequenz möglich eingeschränkte Interruptfähigkeit Auch bei Laptop's verfügbar	Sinnvolle Ausgabebreite: 8 Bit Als Eingang oft nur seriell nutzbar Portumschaltung nicht standardisiert Blockiert Druckerausgabe Nur gering belastbar
Serielle Schnittstelle	Normiertes Datenformat Interruptfähig Auch bei Laptop's verfügbar Vom DOS unterstützt	Geringe Geschwindigkeit Aufwendiges Interface notwendig Spannungspegel $\pm 3.. \pm 12V$ Nur gering belastbar
Gameport	4 Analogkanäle und 4 Digitalkanäle Meist schon eingebaut Standardisierte Steckerbelegung Teilweise vom DOS unterstützt	nicht kalibriert Analogwerte sind in Assembler-schleifen selbst zu ermitteln.
Buskarte	Interruptfähigkeit möglich (Busmaster)DMA-Fähigkeit möglich Sehr schnell Für alle Anwendungen erhältlich	Belegt internen Steckplatz Belegt I/O-Adressen Teuer



Serielle Schnittstelle

- * Kommunikation mit externen Meßsystemen
- * Abfrage und Steuerung seriell programmierbarer A/D-D/A-Wandler

Gameport

- * Abfrage eines Joysticks
- * Realisierung eines einfachen A/D-Wandlers
- * Abfrage von bis zu 4 Digitalsignale

Buskarten

Die Steckkarten für PC's sind in fast allen Ausführungen verfügbar. Ohne externe Schaltungen sind die einzigen Limitierungen der Stromverbrauch und der verfügbare Platz. Leider können nur selten diese Steckkarten in Laptops integriert werden. Dies begrenzt die Portabilität dieser "Meßsysteme".

2. Probleme beim Messen mit PC's

2.1. Stromversorgung

Normalerweise ist an Einsatzstellen von PC's ausreichende Stromversorgung vorhanden. Wenn aber auf externe Beschaltungen verzichtet wird, muß mit den bereitgestellten Leistungen des PC's ausgekommen werden. Diese liegen bei den angeführten Schnittstellen wie folgt:

Druckerport: Frühere Druckerports waren mit 8255 realisiert, die auf TTL-Basis bis zu 20mA auf Masse ziehen konnten. Mit der jetzigen CMOS-Logik zeigt sich in der Praxis, daß noch 5..10mA aus einer Druckerleitung gezogen werden können.

Serielle Schnittstelle: Die Versorgung von verfügbaren seriellen Mäusen oder Optokoppler geschieht über Statusleitung des seriellen Ports. Durch Spannungsinverter werden die meist gelieferten -12V in positive 5V gewandelt. Es werden dabei Ströme von 10-20mA gezogen.

Gameport: Anders als Druckerport und serielle Schnittstelle stellt dieser Port die Versorgungsspannung direkt zur Verfügung. Die Belastbarkeit ist

wegen mangelnden Einsatzes nicht bekannt.

Buskarten: Die Belastbarkeit der PC-Versorgung ist hier sehr groß und nur durch das Netzteil begrenzt. Oft muß aber für genaue Meßdatenerfassung eine galvanische Trennung eingebaut werden. Der mechanische Einbau der Karte zwischen Anderen wie Grafikkarte und Harddiskinterface erhöhen die Gefahr des Übersprechens von Störsignalen.

2.2. Betriebssysteme

Der Einfluß des Betriebssystems auf die Meßdatenerfassung soll hier hinterleuchtet werden. Dabei werden Probleme angesprochen, die nur selten oder erst nach vollständigem Aufbau einer Applikation erkannt werden.

2.2.1. (MS)DOS

Das MSDOS-Betriebssystem ist im Normalfall nur auf Anforderung aktiv und blockiert deshalb auch keine Quellen des Systems. Es gibt aber einige Ausnahmen:

Timerinterrupt:

Alle ca. 54.9255ms (Bustakt $1193180\text{Hz}/65536 = 18.2\text{Hz}$) wird ein Timerinterrupt durch Timer0 ausgelöst. Dieser Interrupt veranlaßt das Weiterzählen der internen Softwareuhr und der Aufruf evtl. eingebundener (BASIC-) Routinen. Wer in eigenen Routinen genaues Timing benötigt, muß alle Interrupts abschalten und danach die interne Uhr gemäß der meistens vorhandenen Hardwareuhr nachstellen.

Tastaturinterrupts:

Die Tastatur liefert bei Wechsel der Taste (Drücken und Loslassen) einen Interrupt, der das DOS zur Abfrage und Auswertung des Tastenports veranlaßt. Im Prinzip gilt das gleiche wie beim Timerinterrupt.

Speicherrefresh:

Dieser eigentlich nicht dem DOS zuzuordnenden Punkt bewirkt ebenfalls eine nicht vom Programm steuerbare Verzögerung. Durch diesen Refresh ist selbst bei abgeschalteten Interrupts ein

geringes Wobbeln in der Schleifendauer bemerkbar.

SMARTDRIVE und Harddisk-/Floppyzugriffe:

Dieser Software-Harddiskpuffer ist normalerweise sehr nützlich für Programme, weil dadurch die I/O-Zugriffe minimiert werden. Leider haben sie auch den Nachteil, daß dann bei Diskzugriffe größere Datenmengen übertragen werden und dadurch der Rechner für mehrere Millisekunden stillgelegt werden kann. Bei direkter Übertragung von Meßdaten auf Festplatte sollte deshalb auf diese Option verzichtet werden.

EMM386:

Auch der Speichermanager für erweitertes RAM ab 386-Rechner ist ein Wolf im Schafspelz. Obwohl er normalerweise nicht bemerkt wird, überwacht er I/O-Zugriffe und die Speicheraufteilung. Da nur er weiß, wo tatsächlich der angeforderte Speicher physikalisch vorhanden ist, ändert er bei Programmierung der DMA Parameter. Der einzige für den Anwender "sichtbare" Nachteil ist die Blockierung von Interrupts von bis zu 50us (386SX-16). Dies macht leider die Interrupt-Datenabstimmung ab 20kHz unmöglich.

2.2.2. Windows

Das Windows ist ein großer Resource-fresser und blockiert meist alle Systemfeatures. Da es nicht auf Echtzeit getrimmt ist und auch kein echtes, interruptgesteuertes Multitasking unterstützt, ist eine zeitsynchrone Datenerfassung eigentlich nicht mehr möglich. Falls man trotzdem eine DOS-Anwendung in einem Windows-Fenster laufen läßt, so hat man viele Probleme:

- * I/O-Adressen sind nicht beliebig verfügbar (z.B. Druckerport)
- * Ein 4kHz-Timerinterrupt blockiert auch einen 486AT-33 vollständig
- * Jedoch auch mit Windows-Programme bleiben viele Schwierigkeiten
- * I/O's, Interrupts und DMA's sind nur mit erheblichen Programmieraufwand verfügbar
- * Exaktes Timing unter Windows problematisch

Neuere Windows-Versionen (Windows NT und Windows 95) machen die Hardwareansteuerung noch schwieriger, da nur über geeignete VxD's oder DLL's darauf zugegriffen wird.

2.3. Hardwareschnittstellen

2.3.1. I/O-Adressen

Der PC hat normalerweise 1024 I/O-Adressen verfügbar. Davon werden aber sehr viele schon durch die vorhandene Hardware und den vielen Schnittstellen belegt. Zusätzliche Grafik-, Netzwerk- Soundkarten begrenzen die Auswahl noch erheblich. Übrig bleiben dann eigentlich nur noch die Bereiche \$110-\$1EF, \$208-\$21F, \$230-\$26F, \$280-\$2AF, \$300-\$31F und \$340-\$35F.

2.3.2. Interrupt

Durch das System und durch die PC-Steckkarten sind viele Interruptquellen angebar, die alle auf zwei Interruptcontroller des AT's laufen.

TIMER0-Interrupt:

Wie schon oben angesprochen, wird dieser Interrupt sowieso schon durch das Betriebssystem verwendet. Es hat sich in der Praxis gezeigt, daß er sehr gut für ständige Aktivitäten wie Messungen im (Interrupt-)Hintergrund verwendbar ist. Im BMC-Meßprogramm NextView wird z.B. der Interrupt von 18.2 auf 4000Hz geändert und bei jedem Auslösen eine Messung an einem Eingang durchgeführt. Man muß nur darauf achten, daß das Betriebssystem weiterhin seinen 18.2Hz-Interrupt erhält. Ein weiterer Vorteil ist, daß der Interruptcontroller nicht umprogrammiert werden muß.

Tastaturinterrupt:

Dies ist der einzige Interrupt, der durch Einfluß des Bedieners über Standardhardware ausgelöst werden kann. Ein sinnvoller Einsatz ist z.B. das Abbrechen des Abastprogrammen. Der Vorteil ist, daß während des Programmlaufes keine zusätzliche Programmzeit belegt wird.

Interrupt der serielle Schnittstelle:

Ebenfalls angenehm ist die Möglichkeit

eines Interrupts bei der seriellen Schnittstelle. Da aber das Betriebssystem ab 9600Baud nicht mehr verwendet werden kann, bleiben hier nur eigene Interruptprogramme. Terminalprogramm (auch in FORTH) haben gezeigt, daß 115200Baud keine Probleme bereiten, wenn der Interrupt nicht zu lange gesperrt wird.

Die einzige Konkurrenz sind noch die Mäuse bei PC's. Sie belegen ebenfalls eine Schnittstelle und eine Interrupt. Man muß bei eigenen Programmen darauf achten, daß nur die vorgesehene Schnittstelle verwendet und die Einstellung der Mausschnittstelle nicht verändert wird.

Druckerinterrupt:

Über eine Statusleitung kann ein Latch gesetzt werden, daß einen Interrupt bei bestimmten Pegel an Druckerport-Eingängen auslöst. Damit kann der PC Fehler wie "PAPER-END" erkennen. Die gleichen Statusleitungen lassen sich natürlich auch zu Handshake-Zwecke bei eigener Hardware mißbrauchen.

2.3.3. DMA

Insgesamt sind in einem AT bis zu 4 8Bit- und 3 16Bit-DMA's vorhanden. Bis auf die 8Bit-DMA für die Floppy sind alle für den Anwender verfügbar. Leider gibt es auch einige Nachteile:

- * DMA's sind nur über eine Buskarte anzusteuern
- * Zieladressen einer 8Bit-DMA müssen in einem 64K-Segment sein
- * Zieladressen einer 16Bit-DMA müssen in einem 128K-Segment sein
- * Einzelübertragung bei-DMA's dauern sehr lang (bis zu 7us)

In der Praxis hat sich gezeigt, daß bei Übertragung von Einzelwerten per DMA von Meßkarte in Speicher das Limit mit heutiger Hardware bei ca. 100kHz liegt. Darüber hinaus ist mit entsprechenden FIFO-Buffer zu arbeiten.

Ein weiteres Problem sind die verfügbaren SCSI-Controller mit Busmasterbetrieb. Diese können bei gleichzeitigem Zugriff auf Harddisk das gesamte AT-Bussystem für mehrere 10us blockieren und dadurch das rechtzeitige Abholen von Daten verhindern.

3. Anmerkung zum Kauf von Meßkarten

Da sich diese Serie durch das nur viermal im Jahr erscheinende Medium auf einen großen Zeitrahmen verteilt, soll schon in dem ersten Artikel ein Tip zum Kauf von Meßkarten erfolgen. Dies vor allem vor dem Hintergrund des ab nächsten Jahres zwingend notwendigen CE-Zeichens, daß viele Vertriebe zur Aufgabe billiger Meßkarten zwingt. Da der Autor aber nicht die Meß- bzw. Steueraufgabe kennt, kann er hier nur in Stichpunkten die wichtigsten Fragen aufführen. Man sollte an Hand dieser Liste die verfügbaren Meßkarten oder -Systeme beurteilen. Manchmal muß man den Vorteil eines schnellen und genauen Systems gegen den hohen Preis aufwägen. Billige System erfordern dagegen meist großen Programmieraufwand oder sind sehr langsam.

- * Einsatzort
 - freier Platz im PC (Laptop ?) und verfügbare Versorgung
 - Umgebung (Temperatur, Staub, Erschütterungen)
 - ausreichende Rechengeschwindigkeit (evtl. Karte mit eigener "Intelligenz" notwendig)
- * Form (Länge: halb, dreiviertel oder voll; Bauhöhe; Abschirmung, Kontaktschutz)
- * Veränderbare I/O-Adressen, Interrupts und DMA's (bei Laptops oft Einschränkungen)
- * Umsetzung des zu erfassenden Signals in die meßbare Spannungen (Verstärkermodule)
- * Programmtools
 - Sofort einsetzbare Meß- und Auswerteprogramme
 - Grundroutinen für eigene Programme
- * Beschreibung
 - Ausreichende Dokumentation der physikalischen Parameter (Stromverbrauch, Genauigkeit)
 - Ausreichende Programmierbeschreibung (Port, Abfrage von Meßwert)
- * Preis und Verfügbarkeit

□



Multitaskingfähige Timerfunktionen

von Bernd Beuster

B.Beuster@bbepoint.forth-ev.de

Der Schwerpunkt dieses Artikels liegt bei einer Timerroutine, die nicht unbedingt bis auf die letzte Mikrosekunde genau sein muß, dafür aber im Multitaskingbetrieb den Echtzeiterfordernissen genügen soll.

Stichworte: Timer Übertrag PowerForth

Bekanntlich ist "Echtzeit" Definitionssache, deshalb sollen bei einem kooperativen Multitaskingbetrieb hier keine allzu hohen Anforderungen an die Timerauflösung gestellt werden.

Als Basis für den gesamten Timer dient in diesen Beispielen das Wort CENTISECONDS. Mit *PowerForth+* (16 Bit) und *PowerForth/386* (32 Bit) wird das mit Hilfe des Wortes (TIME) programmiert, welches auf dem Stack die Stunden, Minuten, Sekunden und Hundertstel Sekunden zurückgibt. Da der Tag 8.640.000 Hundertstel Sekunden lang ist, erfolgt die Berechnung am besten mit 32-Bit-Zahlen. Der Einfachheit halber werden die Beispiele deshalb mit dem 32-Bit-Forth dargestellt. Eine Anpassung an 16-Bit-Systeme ist aber nicht schwierig.

Zeitdifferenzmessung

Zur Zeitdifferenzmessung soll das Wort ELAPSED dienen. Eine kleine Hürde bereitet eine Zeitdifferenzmessung um 0:00 Uhr herum, aber diese kann durch eine Korrekturaddition überwunden werden.

```
Screen 1
1 \ Multitasking Timer
2
3 \ get time in centiseconds
4 : CENTISECONDS ( -- u ) (TIME)
5 SWAP 100 * + SWAP 6000 * + SWAP 360000 * + ;
6
7 \ centiseconds per day
8 24 360000 * CONSTANT CENTISECONDS/DAY
9
10 \ calculate elapsed time in centiseconds
11 : ELAPSED ( n1 n2 -- n3 ) SWAP -
12 DUP 0< CENTISECONDS/DAY AND + ;
13
14
15
16
```

Verzögerungsschleifen

Eine einfache Möglichkeit, eine Verzögerungsschleife zu programmieren, wäre folgende:

```
: CENTISECONDS-WAIT ( # -- )
```

```
CENTISECONDS + \ calculate end time
CENTISECONDS/DAY MOD \ modulo a day
BEGIN PAUSE DUP CENTISECONDS = UNTIL
DROP ; \ wait for end time
```

Leider hat die obige Routine einen Haken. Es wird, speziell in einer Multitaskingumgebung, selten möglich sein, genau den Zeitpunkt der Endzeit zu erreichen. Meistens wird man wohl ein paar Hundertstel Sekunden zu spät sein. Deshalb muß anstelle des Gleichheitszeichens ein anderer Vergleich her, der auch bei Überschreitung des Zeitlimits die Schleife beendet, z.B. dieser:

```
: CENTISECONDS-WAIT ( # -- )
CENTISECONDS + \ calculate end time
CENTISECONDS/DAY MOD \ modulo a day
BEGIN PAUSE DUP CENTISECONDS > WHILE
REPEAT DROP ;
\ wait for end time
```

Diesmal wird die Schleife korrekt verlassen, wenn die aktuelle Zeit gleich oder größer der Endzeit ist. Leider funktioniert dieses Konstrukt nicht bei Zeiten um 0:00 Uhr herum, die Schleife würde dann im ersten Durchgang verlassen werden.

Hier bietet sich eine Berechnung mit Overflow-Arithmetik an. Ab FORTH-83 wurden die DO...LOOP-Schleifen mit Hilfe dieser Technik dahingehend geändert, daß mit ihnen nun auch 65536 Schleifendurchläufe (bei 16 Bit-Systemen) möglich waren. Das alte fig-FORTH konnte nur maximal 32768 Durchläufe. Dieser Vorteil wurde mit einem kleinen Laufzeitoverhead bei der Berechnung des Schleifenindexes erkauft.

Betrachten wir einmal 3-Bit-Zahlen im Zweierkomplement, welche unseren laufenden Timer darstellen sollen:

```
000 0
001 1
010 2
011 3
100 -4
101 -3
110 -2
111 -1
000 0
```

usw. ...

Die Betrachtung der Uhrzeit im Zweierkomplement scheint auf den ersten Blick etwas ungewöhnlich zu sein, aber letztendlich ist dem Rechner unsere Betrachtungsweise einerlei. Additionen und Subtraktionen werden sowohl im Zweierkomplement als auch in vorzeichenloser Darstellung gleich behandelt.

Jetzt stellen wir einen vorzeichenbehafteten Vergleich mit jeweils zwei aufeinander folgenden Zahlen an:

```
0 > 1 falsch
1 > 2 falsch
2 > 3 falsch
3 > -4 wahr
-4 > -3 falsch
```

Multi-Timer

```
-3 > -2 falsch
-2 > -1 falsch
-1 > 0 falsch
0 > 1 falsch
```

usw. ...

Wie man sieht, gibt es genau eine Stelle, wo ein Überlauf (Overflow) stattfindet, den man als Abbruchbedingung nutzen kann. Man muß also eine Verschiebung der Abbruchbedingung in den Bereich des Überlaufs bewerkstelligen. Bei einer Programmierung auf Assemblerebene stellt jede CPU, die etwas auf sich hält, ein Overflow-Flag für genau diesen Fall zur Verfügung. In Forth sieht das dann so aus:

```
Screen 2
1 \ Multitasking Timer
2 HEX
3
4 \ wait for # centiseconds
5 \ works with overflow arithmetic
6 : CENTISECONDS-WAIT ( # -- )
  CENTISECONDS TUCK + \ calculate end time
7 CENTISECONDS/DAY MOD \ modulo a day
8 80000000 + TUCK - SWAP \ prepare overflow math
9 BEGIN PAUSE \ do the next task
10 CENTISECONDS OVER - \ calculate time difference
11 >R SWAP R@ > R> -ROT \ check for overflow
12 UNTIL 2DROP ;
13
14 DECIMAL
15
16
```

Applikation

Der Name CENTISECONDS-WAIT ist zwar ein wenig lang, aber CS (als Äquivalent zu MS=Millisekunden) könnte bei i80x86 Prozessoren mit dem Codesegment verwechselt werden. Eine Alternative wäre vielleicht noch:

```
: MS ( # -- ) 10 / CENTISECONDS-WAIT ;
```

wobei man sich aber im klaren sein sollte, daß eine Millisekundenauflösung in diesem Fall völlig illusorisch ist.

Und daß das Ganze auch funktioniert, zeigt folgendes kleine Multitaskingprogramm. Es werden zwei Tasks generiert, der erste zeigt die aktuelle Uhrzeit im Sekundenrhythmus in der obersten Zeile des Bildschirms an und der zweite Task eine Zeile tiefer die Uhrzeit im Dreisekundenrhythmus.

```
Screen 3
1 \ Test
2 : SHOW-TIME ( x y -- )
3 CURSOR-OFF GETXY 2SWAP GOTOXY .TIME GOTOXY CURSOR-ON ;
4
5 : T1
6 BEGIN 100 CENTISECONDS-WAIT 35 0 SHOW-TIME AGAIN ;
7
8 : T2
9 BEGIN 300 CENTISECONDS-WAIT 35 1 SHOW-TIME AGAIN ;
10
11 : START-TASKS
12 INIT-MULTI \ init multitasker
13 ASSIGN T1 1 TO-TASK \ init tasks
14 ASSIGN T2 2 TO-TASK
15 1 ACTIVATE 2 ACTIVATE ; \ start tasks
16
```

Zusätzlich können natürlich auch im Vordergrundprogramm (Task 0 = Forth) Timerfunktionen genutzt werden. □

Literaturverzeichnis

Thomas Beierlein, "Was tickt denn da...", VD 4/92, S. 21-22.
Arndt Klingelnberg, "Pudding mit Sahne", VD 1/94, S. 18-20.
Bernd Paysan, "Zeitmessung auf dem PC", VD 2/94, S. 10-11.

Forth inside Textverarbeitung

Wer das Usenet-Brett 'comp/lang/forth' regelmäßig liest, konnte im April einiges über Textverarbeitungssysteme erfahren. Hier berichtet z.B. Wil Baden :
"The original Word Perfect was written in Forth. Word Perfect Corporation's original name had "Forth" in it. It is said to have been done on a Summer vacation by two Brigham Young University students. Forth's portability let them port it to many platforms. Now it is written in a profane language.
The word-processor that was delivered with the first IBM PC's was written in Forth by John Draper, AKA Cap'n Crunch. The story is told that IBM was set to

release the PC in three months when they suddenly realized that they didn't have a word-processor. Draper wasn't doing much doing time so they were able to furlough him. So the story goes. Maybe he will pop up here to confirm or deny."!
Martel Firing berichtet über diverse integrierte Programme, an denen er mitwirkte, insbesondere das von Noumenon 1984 herausgebrachte 'Intuit', ebenfalls in Forth.
Roedy Green glaubt, daß der frühe Apple II Word Processor und ein Lotus Clone, VP-Planner, ebenfalls in Forth geschrieben sind.

clv nach comp/lang/forth

Forth-Systeme

Geschichte: Woher kommt der Name 'C++'

In diesem Zusammenhang eine kleine Anekdote: Der Ulrich Richter hat, bevor er zur Moerser Forthgruppe kam, ausschließlich in FORTRAN und C programmiert. Die Christel hat das 'aufgeschnappt' und wollte wissen, woher überhaupt dieser merkwürdige Name 'C' kommt. Ich habe das, zugegeben mit böser Absicht, so erklärt, daß dieser Name von Leuten bevorzugt wird, die nur bis Drei zählen und nur bis 'C' buchstabieren können. Der Martin Bitter hat dann noch eins drauf gegeben und gesagt : "Exakt, und wenn die 'F' meinen, sagen die C++ !"

F.Prinz, April 95

Forth/2 für OS/2

Wer immer noch kein Forth für OS/2 gefunden hat, sollte vielleicht einen Blick auf Forth/2 werfen. es ist in assembler geschrieben, die Quelle ist dabei und es unterstützt 'multiple threads'.
Erhältlich bei: <http://www.interaccess.com/users/ka9dgx/forth2.html>

clv nach comp/lang/forth
10.4.95)



Presseschau: M&T Februar '95
New Micros Inc auf 68HC11

Die Einplatinencomputer des amerikanischen Herstellers basieren auf 68HC11-CPU's. Während der Hersteller als Softwareunterstützung bisher nur Forth im Angebot hatte, liegen auf Diskette nun auch Basic und Small-C als Alternative bei. Vertrieb durch:

Microscan
 Überseering 23
 22297 Hamburg

jrd/Rafael Deliano
 Presseschau: M&T Februar '95
Die Forth-Spezialisten

Wer Die Spezialisten für Forth-Software sind, verrät uns die M&T zwar nicht. Unter der Rubrik Geschäftsverbindungen erfahren wir aber deren Fax-Nummer, die 08131 / 78500 lautet. Recherchen der Redaktion ergaben, daß es sich dabei um einen Anschluß im schönen Ort Karlsfeld im Süden der Republik handelt. Eine weitere Befriedigung der Neugier unserer Leserinnen und Leser durch direktes Anfaxen scheiterte

bisher leider am beschränkten VD-Etat. Schade ...
 jrd & clv

**Design & Elektronik 10/95
 Sprachen auf 8-Bitern**

In der Design & Elektronik 10/95 war auf drei Seiten eine umfassende Tabelle mit derzeit angebotenen 8-Bit-Einplatinencomputern zu finden. An Software fanden sich reichlich C und Basic, aber auch zweimal Forth. Dies war zum einen das nanoFORTH für Mitsubishi-Controller bei Glyn (vgl. VD 1/95, S.6) und zum anderen Forth für den PIC 17C42 von Microchip. Letzteres liefert z.B.:

pd Computer
 Schloßstr.16
 61184 Karcken
 Tel.: 06039 / 7061

jrd/Rafael Deliano

Everett (Skip) Carter
 skip@taygeta.oc.nps.navy.mil
 comp/lang/forth, 12.4.95

Forth auf CD

Der tewi Verlag bietet für 69,80 DM unter der ISBN 3-89362-629-8 eine Doppel-CD mit dem Titel "POWER-PROGRAMMIERUNG" an. Laut Titelblatt des Päckchens und seiner Rückseite befinden sich auf den beiden Scheiben mehr als 1 GByte an Entwicklungssystemen, Sourcen und Werkzeugen. Das sind z.B. Vollversionen von: Turbo Pascal 5.5, Turbo Pascal für Windows, Borland's Resource Workshop 1.0, Turbo C++ 1.0, Turbo Assembler, inkl. Debugger für DOS und Windows. Weiterhin GNU C++ für DOS und OS/2, GNU ADA, Pascal für OS/2. Es folgen Compiler für Modula, Oberon, Cobol, Fortran; Interpreter für Lisp, Prolog, Logo, Smalltalk, Snobol; ein Transpiler (nie gehört), Assembler, Crossassembler; Upgrade Windows 3.1 --> Windows 3.11, Win32S 1.15; Emulatoren für C64, CP/M, Spectrum und Apple II...und "viele mehr".

Unter anderem werden auch 19 (!!!) FORTH-Systeme angekündigt, von denen es im kleinen, beigegepackten Heftchen heißt:

"Forth ist eine Sprache für Enthusiasten. Eigentlich kann man mit Forth auch alles schreiben. Dementsprechend viele Public-Domain- und Shareware-Forth-Systeme gibt es und dementsprechend viele Forth-Compiler und -Interpreter befinden sich auf CD1 in den Unterverzeichnissen von >|forth|compiler< und >|forth|interpreter<, nämlich insgesamt 19 Stück. Außer DOS-Versionen sind auch OS/2-Compiler dabei."

Bei näherem Hinsehen relativiert sich die Angabe des Verlages allerdings auf 14 Systeme. Eines der angepriesenen 'FORTHs' ist FIFTH. Eines der 'Systeme' ist die 'Razorback Da-

Bücher

**Hendtlass:
 Real Time Forth**

250 Seiten
 Shareware per Diskette, englisch
 Bezug einer Papierkopie (s.u.)

Dieses Buch ist bemerkenswert, weil es

- nach dem Sharewareprinzip vertrieben wird,
- auf F-PC 3.56 aufbaut,
- weil es für Einsteiger geschrieben wurde und trotzdem sehr tief geht.

Der Autor gibt zunächst eine Einführung in das Stackprinzip, erläutert dann den Unterschied zwischen Daten- und Returnstack und gibt Hinweise zum Manipulieren des Stacks. Besonders wichtig ist es, daß reichlich Beispiele vorhanden sind, die auch gut kommentiert sind. Auch Trainingsaufgaben (mit Lösungen) sind genug da.

Das dritte Kapitel behandelt Arithmetik-Logik-Vergleichsoperationen, auch dieses schließt mit Übungen ab. Im vierten Kapitel werden dann die wichtigsten Kontrollstrukturen (IF .. THEN .. ELSE ; DO .. LOOP usw.) besprochen. Die nächsten Kapitel behandeln Datentransport und Stringverarbeitung. Besonders als Hinweis für Anfänger sei gesagt, daß das Inbetriebnehmen von F-PC sehr ausführlich erläutert wird.

Das daran anschließende Kapitel behandelt den Fall, daß etwas nicht tut, was es soll. Es geht auf Debugging ein und behandelt die Technik genauso wie die einzel-

nen Worte. Kapitel 9 bis 11 und 13 bis 16 sind eher dem Aufsteiger vorbehalten und beinhalten Textverarbeitung, Mathevokabular 'Maths', Deferred words, Interfacing DOS, Vokabulare, CREATE DOES>, und Timing, Multitasking. Das Kapitel 12 ist ein einziges Beispielkapitel.

Abgerundet wird das Buch durch ein Kapitel zu PASM, die richtige Mischung aus Assembler und Forth, die Behandlung von Interrupts, Input/Output, Turnkey, Metacompiling, Targetcompiling und Übungen.

Im Anhang finden sich einige nützliche Listen über die Innereien von F-PC und Lösungen der Übungen.

Fazit: Das Buch hat mich bis jetzt einen großen Schritt nach vorne gebracht. Es werden wohl noch weitere folgen.

Einziger Nachteil: Das Buch ist in Englisch geschrieben, es ist aber leicht verständlich.

Kopien dieses Buches können bei mir gegen einen V-Scheck von 32 DM bezogen werden (bitte Adreßaufkleber beilegen!)

Robert Freitag
 Formesstr. 31 A; 51063
 Köln; Tel: 0221 / 617 93

**Design & Elektronik April '95
 Experimente interaktiv steuern**

Dieser Beitrag von J. Pisarz von der Firma Port auf Seite 58 beschäftigt sich mit einer erweiterbaren Experimentsteuerung für die D2-Mission. Als Erweiterungs- und Steuerungssprache wird dabei Forth auf Motorola/OS9-Systemen eingesetzt.

Jörg Plewe

**Literatur
 Leo Brodies 'Forth Style Guide' im WWW**

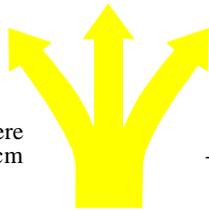
I have made an HTML version of Leo Brodies Forth Style Guide that appeared in "THINKING FORTH". The guide can be reached from the Forth Interest Group home page: http://taygeta.oc.nps.navy.mil/fig_home.html. Because the style guide was written some 10 years ago, parts of it are clearly dated.

I would like to get well considered commentary on the guideline and add them as hypertext links in an "updates" section for it.. Please take a look and give me your comments.

Rezensionen gesucht!

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

Bitte bei der Redaktion melden / Claus



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

*Geschichte
wird gemacht
- es geht voran!*

Forth-Gesellschaft: Jagen und Sammeln

Auf der Forth-Tagung wurden verschiedene Ideen geäußert, die in ähnliche Richtungen gehen:

- Klaus Kohl vom Forth-Vertrieb schlägt vor, eine CD herauszugeben.
- Skip Carter (USA) plant ebenfalls ein CD-Projekt
- Claus Vogt vom Forth-Magazin hätte Interesse, verschiedene Archive aufzubauen
- Holger Petersen von der Mailbox möchte interessante Programme zur Verfügung stellen
- Egmont Woitzel hätte gerne einen FTP-Server für Forth in Deutschland. FTP-Server wie Taygeta (Silicon Valley) und ein Server in Portugal spielen eine wichtige Rolle für Forth.

Die Lage stellt sich etwa so dar:

- Von weltweiten Netzen bis zu private Sparstrümpfenist hinreichend Material vorhanden.
- Der Bedarf ist vorhanden.
- Die technischen Möglichkeiten reichen aus, um jeden Bedarf zu jedem beliebigen Preis zu befriedigen

Aber irgendetwas fehlt. Wer sagt mir, was?

Die VD-Redaktion veröffentlicht gerne kurze Beiträge zum Thema.

(clv)

Fortsetzung von der letzten Seite

tenbank' - natürlich mit UR/FORTH definiert, aber ohne UR/FORTH nicht lauffähig. Ein System ist ein DADA-Compiler (nie von gehört...), dessen einziges, äußerliches Merkmal einer Verwandtschaft mit FORTH ein beliebiges File *.SCR im Blockformat zu sein scheint. Mike Warot's FORTH/2 ist gleich zweimal vertreten. Zu F83 existiert ein zusätzliches Verzeichnis F88.

Aber von den immer noch 14 Systemen verbleiben zwei OS/2-Forth, eben Mike's FORTH2 und Alan Pratt's OS2FTH. Dazu gesellen sich BBL, EFORTH, F83, FIG, FORTHCOMP, FORTHED, FPC, LOVE, MIN4ZH, PYGMY, UNIFORTH und ZEN. Das ist immer noch eine beachtliche Sammlung. Insgesamt erscheinen mir die beiden CDs als eine riesengroße 'Wundertüte', deren Ecken ich kaum jemals bis ins Detail 'erforsten' werde. 1 GByte ist mir einfach zuviel. Was die Zusammenstellung der FORTH-Systeme anbelangt, wird der eine oder Andere sicher etwas zu bemängeln haben, vor allem vermutlich die Auswahl der meist wenig bis gar nicht kommentierten Quellen. Aber es ist ein Anfang! Kaufen wird die CDs wohl Niemand (naja, ich schon)

wegen der FORTH-Systeme. Da werden Pascal und C - oder die anderen Exoten - vermutlich eher 'ziehen'. Immerhin ist FORTH dabei. Freuen wir uns auf Jeden, der dadurch neugierig wird ;-)

Erwähnt werden sollte an dieser Stelle unbedingt, daß wir, die Leser dieses Beitrages, den Hinweis auf die CD einem Forth-Freund aus Leipzig verdanken. Der Friedhold Birnkammerer, mit dem ich sowohl brieflich als auch (gelegentlich) telephonisch in Kontakt stehe, hat mir von tewi's CD erstmals berichtet.

Zusätzlich läßt sich noch die aktuellste Version von Mike Warot's FORTH/2 auf der HOBBS CD finden. Diese CD, die (eigentlich) ebenso

erscheint wie die VD - nämlich quartalsmäßig - ist allen OS/2-Lern ein Begriff. Auf ihr findet sich wieder, was sich im Laufe eines Vierteljahres auf dem HOBBS ftp.server zu OS/2 ansammelt. Für 29,- erstet der willige Käufer neben einer Unmenge an Treibern, Tools, Dokumentationen, Patches und Demos auch 'Programmiersprachen' aus der PD und aus dem Sharewarebereich - eben auch FORTH/2 in der Version Beta 0.40. Allerdings schreibt Mike in seinen Dokus, daß technisch kein Unterschied zur Version 0.39 besteht. Lediglich an der

Forth-Gesellschaft: Vertrieb

Neu im Vertrieb:

- VD95-1 - Diskette zum Forthmagazin
- VD92-2 "
- Dokumentation ANS X3.215 - 1994 im Word 2.0-Format
- Crossassembler für Z8 und PIC von Bernd Hinze
- EFORTH für PIC 1742

je Diskette: 10.-DM

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



Forth-Gesellschaft: Microcontrollerverleih

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

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

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

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

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

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

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

Forth-Gesellschaft: Forth-Magazin

Das Forth-Magazin sucht ...

An Sachmitteln fehlen noch: **An Beratern fehlen noch:**

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

- Spezialisten aus dem Satz-, Grafik-, Layoutbereich
- DTP-Erfahrene zur Beratung bei der Neuanschaffung
- Wer kann mir Hinweise zum Belichten von Diskette geben?

An Hardware fehlt noch:

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

An Software fehlt noch:

- Ein DTP-System, teurer als 50 DM
- Ein Archivsystem, vielleicht auf Basis des F-PC-Hypertextes?

und natürlich Beiträge...

.. zum Beispiel für das geplante Schwerpunktheft 'Embedded Systems'. Aber auch in allen anderen Bereichen, die Euch interessieren. Kurzmeldungen aus dem Internet oder von aktuellen Projekten, Rezensionen, der Phantasie sind keine Grenzen gesetzt...

Claus 030 / 782 81 79



Ja wo sind sie, die Kritiker? Nun hatte ich doch gedacht, auf der Forth-Tagung ein paar richtig kritische Stimmen zu hören. Aber - im Großen und Ganzen scheinen alle recht zufrieden mit ANS Forth zu sein. Nicht, daß es überhaupt keine Kritikpunkte gäbe und

(Dies ist ein Wink mit dem Zaunpfahl diesen Dienst der Forth Gesellschaft zu nutzen).

```
BEGIN ANS FORTH mysterious?
  WHILE learn REPEAT
```

Die ANS-Forth Kontrollstrukturen sind schon ein wenig erklärungsbedürftig, weil sie einfach das erlauben, was gefällt. Ich will einen kleinen Einblick geben, was dem einen oder anderem da alles gefallen könnte. Ich würde mich freuen, wenn das Thema in

einem getrennten Artikel einmal gebührend erschöpfend behandelt werden würde.

Die ANS Forth Kontrollstruktur-Worte müssen gar nicht ausschließlich in bekannten Kombinationen (BEGIN WHILE REPEAT, BEGIN UNTIL, IF THEN) auftreten, son-

BEGIN markiert das Sprungziel für den Rückwärtssprung der Schleife und legt dazu eine entsprechende Identifizierung (jedes System darf selbst entscheiden, was das ist) 'begin-sys' auf den Kontrollfluß-Stack. WHILE erzeugt einen Vorwärtssprung, dessen Sprungziel noch aufgelöst werden muß. ANS Forth gemäß vergräbt WHILE die entsprechende Identifizierung 'while-sys' unter der von BEGIN. Der Kontrollfluß-Stack hat dann die Form (while-sys begin-sys). AGAIN erzeugt einen Rückwärtssprung zu der Adresse, die durch das oberste Kontrollfluß-Element identifiziert wird (dies muß ein 'begin-sys' sein); in unserem Fall ist das die Identifizierung von BEGIN. Schließlich löst THEN den noch offenen Vorwärtssprung von WHILE auf.

BEGIN markiert also Rückwärtssprungziele, WHILE springt an einem Rückwärtssprung vorbei nach vorn, AGAIN springt unbedingt rückwärts und THEN markiert Vorwärtssprungziele. Ja!

Nun ist die WHILE-Schleife mit mehreren Ausgängen nur noch eine Fingerübung:

```
BEGIN ANS FORTH mysterious?
  WHILE concentrated?
    WHILE learn
    AGAIN THEN THEN
```

*Jetzt
geht's
ANS
FORTH!*

von Ulrich Hoffmann



das ist ja auch gut so.

Das technische ANS Forth Komitee hat jedenfalls gemäß der ANSI-Richtlinien eine Sperre, sich zu treffen und weiterzustandardisieren. Damit herrscht von offizieller Seite also verordnete Ruhe. Motivation ist, daß sich der aktuelle Standard auch über einen längeren Zeitraum etablieren kann.

Nichtsdestotrotz ist eine (damit inoffizielle) Auseinandersetzung mit den Schwachpunkten des aktuellen Standards und Überlegungen in Teilen, wo er gar nichts aussagt, wichtig, sollen mit der nächsten (wohl in fünf Jahren zu erwartenden) offiziellen Runde substantielle Fortschritte gemacht werden. Das heißt aber, daß sich auch um den aktuellen Standard eine Folklore ausbilden wird, welche Konstrukte verwendet und unterstützt werden sollen, und welche links liegen bleiben. Einen Vorstoß in dieser Richtung hat Anton Ertl mit den lokalen Variablen gemacht. Als grundsätzlicher Verfechter lokaler Variablen möchte er dennoch das Wort LOCALS| wegen der Reihenfolge, in der die lokalen Variablen definiert werden, geächtet sehen und schlägt statt dessen vor, die Syntax von John Hayes einzusetzen [1]. Auch die Scientific Forth Library, von der auch schon das letzte mal die Rede war, hat ANS Forth noch weitere Konventionen (die Scientific Forth Folklore) aufgesetzt, damit die Bibliothek mehr aus einem Guß wird. Der aktuelle Stand der Bibliothek wird regelmäßig im Usenet in comp.lang.forth veröffentlicht und unlängst wurde sogar auf Anfrage der gesamte bisherige Inhalt dort verschickt

Die ANS-Kontrollstrukturen sind

dem dürfen nach gewissen Regeln beliebig kombiniert werden.

Um ihre Bedeutung zu erklären, definiert der Standard einen Kontrollfluß-Stack samt zugehöriger Operationen, den ein System durch den Daten-Stack aber auch durch einen separaten Stack implementieren darf. Ein Programm, das eigene neue Kontrollstrukturen definieren will, muß als Konsequenz den schlimmsten Fall annehmen, nämlich, daß der Kontrollfluß-Stack durch den Datenstack implementiert ist.

Kombinieren darf man Kontrollstruktur-Worte nun, wenn ihre Kontrollfluß-Stack-Effekte zusammenpassen und letztlich einen blanchierten Kontrollfluß-Stack ergeben. Als Beispiel soll die WHILE-Schleife mit mehreren Ausgängen herhalten. Zum Verständnis ist es einfacher die obige Schleife einmal statt mit REPEAT mit AGAIN und THEN zu schreiben:

```
BEGIN ANS FORTH mysterious?
  WHILE learn AGAIN THEN
```

Alles klar?

Schlimm wird es erst, wenn nun der Kontrollfluß-Stack explizit während einer Definition umgeordnet wird. Dann ist der Kontrollfluß in der Regel nur noch dem Programmierer klar, und dem auch nur während des Hinschreibens. Die vernünftige Herangehensweise ist also, die für den exotischen Kontrollfluß notwendigen Kontrollfluß-Stack-Manipulationen in neuen Compiling-Words zu verstecken und auf das explizite Verändern des Kontrollfluß-Stacks zu verzichten. Ich hoffe, das wird Folklore...

Im nächsten Heft werde ich in meiner Wortkiste kramen und einige der ANS Forth-Implementierungen mir liebgewordener Worte vorstellen. Vielleicht findet der ein oder andere dabei ja etwas für seinen eigenen Fundus.

[1] "User-Defined Local Variable Syntax with ANS Forth", John R. Hayes, ACM SIG-Forth Newsletter Vol.4 No.2, S.19f





Forth International

von Fred Behringer



*Das französische
Turbo-Forth habe ich
mir zum Vorbild für
mein Transputer-
Forth-System F-TP*

Turbo-Forth und die Franzosen

Turbo-Forth wurde von Marc Petremann, Michel Zupan und anderen aus der französischen Gruppe JEDI seit 1987 (also vor F-PC !) entwickelt. Es basiert auf F83 von Laxen and Perry und ersetzt die Blockstruktur (Screens) durch eine konsequente Dateianbindung im Sinne von DOS (für IBM-Kompatible).

Die Gruppe JEDI scheint es nicht mehr zu geben. Marc Petremann gibt es noch:

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

Er betreibt eine Firma, die hauptsächlich Turbo-Forth vertreibt. Es gibt inzwischen eine ganze Reihe von Disketten mit Turbo-Forth und Erweiterungen. Ich habe vor Jahren solche Disketten zu DM 10,- pro Stück erstanden.

1988/89 habe ich eine deutsche Version des Grundsystems (Fehlermeldungen, HELP-Erklärungen, SEE-Decompilate usw.) erstellt. Marc Petremann hat mir seinerzeit die kostenlose Weitergabe erlaubt.

Turbo-Forth habe ich mir zum Vorbild für mein Transputer-Forth-System F-TP genommen, dessen Fertigstellung demnächst bevorsteht.

Forth in Holland

In Arnheim gibt es die Forth-Gruppe HCC-Forth-Gebruikersgroep. In Kontakt stehe ich mit dem Redakteur der Clubzeitschrift dieser Gruppe. Seine Anschrift:

**Willem Ouwerkerk,
Boulevard Heuvelink 126,
NL-6828 KW Arnheim .**

Die Zeitschrift kommt ein paar Mal pro Jahr heraus und heißt "Het Vijgeblad", also "Das Feigenblatt". Das ist ein Spiel mit dem englischen Wort "Fig" (Forth Interest Group), zu deutsch "Feige", und holländisch eben "vijg".

Und im Vereinigten Königreich

Von Willem Ouwerkerk habe ich einen Hinweis auf die Forth-Gruppe "FORTHWRITE FIGUK" in England erhalten. Adresse des Vorsitzenden:

**Paul Benett
25 Court Road
Horsfield Common
Bristol BS7 0BU
Great Britain.**

Ich habe geschrieben und bleibe am Ball.

Fred Behringer, München

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

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

Verständlich sind für mich die Sprachen Englisch, Holländisch, Italienisch und Französisch. Bei Russisch würden meine 4 Jahre Schulkenntnisse nicht ausreichen. Aber vielleicht könnte ich dann Unterstützung von den sehr aktiven Rostockern bekommen (?)

Gehaltvolles

zusammengestellt und übertragen von Fred Behringer

Het Vijgeblad von der HCC Forth-Gebruikersgroep, Holland

Nr. 44, Erstes Quartal 1994

3 Forth-83 naar dpANS5/6, Teil 2
Willem Ouwerkerk

Reichlich dokumentierte Doppelpunkt-Definitionen der Core-Erweiterungen: `.R 0<> 0> 2>R 2R> 2R@ :NONAME <> ?DO AGAIN C'' CASE ENDCASE ENDOF ERASE FALSE MARKER NIP OF PARSE REFILL RESTORE-INPUT SAVE-INPUT SOURCE-ID TO TRUE TUCK U.R U> UNUSED VALUE WITHIN \` Anmerkung des Rezensenten: Eigenartig, fast nichts ist dabei, was nicht in Turbo-Forth schon seit 1988 (und in meinem Transputer-Forth F-TP seit 1991) enthalten wäre.

11 Interview met Charles Moore
Johnny Hacker

Jux-Interview. Dem Autor ("Autor" ?) sitzt der Schalk im Nacken. ... ich hatte nicht viel zu tun. Alles war in COBOL programmiert und funktionierte prima. Ich hatte Zugang zu einem alten IBM-Großrechner und war der erste, der versuchte, seine Zeit am Computer mit Spielen totzuschlagen. TETRIS gab es noch nicht, und keine "256

leicht zu programmierende Spiele in TRS80-BASIC". Also fing ich an, meine eigenen Spielzeuge zu entwickeln. ... soll Charles Moore dem "Interviewer" erklärt haben. Das habe ich, der Rezensent, doch schon mal irgendwo gelesen ? 4 Seiten in englischer Sprache. Heiter bis tiefsinnig.

15 Beginnerservaringen met F-PC
J.M.C. Juursema

Anfänger am F-PC-Paket: ... "2,5 M HD nötig, Installation ("kaum") 45 Minuten, einfaches Kalenderprogramm von FORTH-79 nach F-PC, Debug arbeitet gut, viel gelernt" ... (holländisch).

25 GLOGEN: automatische generator voor programma documentatie
Lennart Benschop

Das Anlegen eines Glossars, also einer Liste von Erklärungen der im System enthalten Forth-Worte, ist mühsam und fehlerträchtig - sagt der Autor. Sein Programm, 5 Seiten in ANS-Forth, soll die Arbeit erleichtern und weitgehend automati-

sieren. 20 Zeilen Einleitung in holländischer Sprache, das Programm selbst mit allen Kommentaren englisch.

33 TISC II (a Threaded Instruction Set Computer)
Russ Webb / Coos Haak

Der (kurze) erste Teil des Artikels ist in holländischer Sprache geschrieben. Coos Haak hat ihn aus dem Englischen (Autor Russ Webb) übersetzt. Er beschreibt den Aufbau des TISCII. Das ist ein CPU-Baustein, der nach den Prinzipien einer gefädelten Sprache, so wie sie Forth oft darstellt, entwickelt wurde.

36 Invoer van getallen in Forth
Albert Nijhof

Zahlenumwandlung nach ANS-Forth-Mannier, basierend auf `>NUMBER`. Der Autor konstruiert einen Satz von Zahlenumwandlungsoperatoren mit verschiedenen Eigenschaften. Etwa 18 Doppelpunkt-Definitionen, einige davon Anwendungsbeispiele. Alles prima kommentiert. Holländisch.

42 Het beste van USENET
Lennart Benschop

Elf Seiten Fragen und Antworten über Forth, aufgegriffen in USENET, Newsgroups: `comp.lang.forth`. Natürlich in englischer Sprache. ?

Forth Dimensions der Forth Interest Group, USA

7, Vol.16 Januar/Februar 1995

8 Forth in 32-bit Protected Mode
Richard Astle

Der Autor erklärt seine direkte Methode, Forth in den 32-Bit-Protected-Mode-Raum zu bekommen - ganz ohne Assembler, Linker oder Protected-Mode-Lader. Die Methode erzeugt ein Mischsystem, das eigentlich aus zwei Forths besteht, einem 16-Bit- und einem 32-Bit-System. Zum Ausgleich erhält der Benutzer die Möglichkeit, zwischen den beiden Systemen hin- und herzuschalten und die ihm vertrauten Werkzeuge weiterhin zu verwenden, ohne sie einer neuen Umgebung anpassen zu müssen.

21 A Forth-Oriented Compiler Compiler and its Applications
Mati Tombak, Viljo Soo, Jaanus Pöial

Stack-orientierte Sprachen (Forth, PostScript usw.) werden in Software-Systemen oft als Zwischen- oder Zielsprachen verwendet, weil sie portierbar, flexibel, kompakt und einfach sind. Auf dem Gebiet

der Compiler-Compiler wird oft der Begriff einer "virtuellen Stackmaschine" verwendet, um die Semantik der Quellsprache und das Verhalten des Programms zu beschreiben. Leider verwendet jede/r Autor/in seine/ihre eigene Stackmaschine. Die Grundidee des vorliegenden Artikels ist die Verwendung einer "real-existierenden", weithin bekannten Standardsprache als Zwischensprache im Compiler. Forth ist die Sprache, in der das System programmiert wurde, und Forth ist auch die Sprache der Compiler, die es erzeugt.

23 Forth2LaTeX - a Pretty-Printer
Ronald T. Kneusel

Forthens Schönheit sollte leuchten - selbst aus einem Druckauszug heraus. In diesem Sinne schrieb der Autor sein Programm, mit dessen Hilfe man ein Forth-Programm nach LaTeX übersetzen kann, einem Derivat von Donald Knuths berühmten Drucksatzsystem. Forth2LaTeX ist für jene gedacht, die ansprechende Quelltexte produzieren wollen.

Es läßt normalen Text innerhalb eines Forth-Programms zu und der Programmierer kann ein Programm schreiben, das bei Ausführung seine eigene formale Beschreibung mitliefert.

28 Using Zeller's Congruence
Walter J. Rottenkolber

Das ist ein Musterbeispiel für die Verwendung von Forth, um herauszubekommen, auf welchen Wochentag ein bestimmtes Datum fällt. An welchem Tag sind Sie geboren ? Wann ist das nächste Mal Freitag, der 13. ?

30 Reports from euroForth '94
Gordon Charlton and Tim Hendtlass

Teilnehmer aus allen Ländern besuchten die euroForth-Konferenz 1994, um sich Vorträge der verschiedensten Gruppen von Forth-Programmierern, Produzenten und Wissenschaftlern anzuhören. Wie bei Europas bestangesehener Forth-Zusammenkunft nicht anders zu erwarten, waren Vorträge und Organisation ein voller Erfolg.

**Forth inside
IPS**

Die Sprache "Interpreter für Prozeßstrukturen" [1] wurde ursprünglich von Dr. Karl Meinzer für die CPU RCA 1802 zum Einsatz auf dem Amateurfunksatellit OSCAR-10 entwickelt und orientiert sich weitgehend an FORTH. Für den Einsatz auf den Satelliten OSCAR-13 [2] und RUDAK-II [3] folgte die Portierung auf die CPUs 65SC02 und RTX 2000.

Parallel zur Entwicklung für die Flugrechner ist auch laufende Portierung auf neue Entwicklungssysteme nötig. Der letzte Schritt war von einem Atari XL (6502, Kassettenlaufwerk) auf einen ATARI ST. Womit IPS nun auch für den 68000 existiert. Der Bordrechner des Satelliten, die IHU "Internal Housekeeping Unit", hat IPS nicht fest installiert. Beim OSCAR-13 erfolgt nach dem Reset ein kurzer Selbsttest. Dann werden die 7,2kByte IPS von der Bodenstati-



on geladen. Es folgt die Flugsoftware mit 36kByte die während des Ladens kompiliert wird. Die Flugsoftware ist in bis zu 8 Tasks aufgeteilt, die durch einen simplen round-robin-Multitasker, "Kette" genannt, von einem 20msec-Timer geschaltet werden. Weite Teile der Flugsoftware sind in Assembler geschrieben. Zu ihren Aufgaben gehören Kommunikation (AX.25 Protokoll), Bahnberechnung (z.B. mit CORDIC), Sicherheitsfunktionen (Batterieüberwachung, Speichertest). Die Betriebserfahrungen mit IPS ist so positiv, daß es auch weiterhin von der deutschen AMSAT für Flugrechner favorisiert wird.

Literatur:

[1] Byte 1/79 K.Meinzer "IPS, an Unorthodox High Level Language"

[2] AMSAT-DL Journal 2/94 J. Miller "Die Steuerung des OSCAR 13"

[3] AMSAT-DL Journal 1/90 Stefan Eckart "Die Software des RUDAK-II-Experiments"

jrd (Rafael Deliano)
nach BYTE 9/92 S.274

Het Vijgeblad von der HCC Forth-Gebruikersgroep, Holland**Nr 39, Drittes Quartal 1992**

Ein Heft, das sich ausschließlich mit Hardware-Fragen beschäftigt. Interessant für diejenigen unter uns, die ab und zu auch gern mal zum LötKolben greifen, ohne gleich ausgebildete Elektronik-Entwickler zu sein.

2 Van de redactie
Lennart Benschop

auch der Einsatz von Assembler von Forth aus ist ein Genuß, wenn man es mit den herkömmlichen Assemblern vergleicht Hardware ansprechen Forth Microcontroller unmittelbar. ... Turbo-C GNU-C ... EMACS ... unmöglich ...

3 Beginnervseravingen
Joop Kleinsma

F-PC Forth-Paket deprimierend groß und kompliziert ganz anders, als es in den Forth-Büchern steht und doch auch wieder überraschend einfach Hypertext per Knopfdruck

5 Forth cursus en Cursus-Forth
Albert Nijhof

Die HCC hat einen Forth-Kurs mit dem zugehörigen Kurs-Forth (eine ANS-antizipierende Version) entwickelt. 12 Seiten Kostprobe. Das Ganze stützt sich auf einen 6501-Single-Chip-Computer mit Forth im ROM (Hardware-Beschreibung im Verlaufe des Kurses). Auf Seite 6 lese ich (der Rezensent): Forth besteht aus Worten. Werden ein oder mehrere Worte hintereinander eingetippt und durch ein RETURN abgeschlossen, so führt Forth diese Worte in der eingegebenen Reihenfolge aus. Ein Wort, das eine Zahl benötigt, greift sich (dagegen) immer diejenige Zahl, die als letzte eingegeben wurde.

18 Het opslaan van een gedigitaliseerd audio signaal op floppy disk
Marcel Hendrix

Umfangreicher Artikel über den RTX2001A als Anrufbeantworter (28 Seiten: 18 Seiten Listing, 1 Seite Schaltbild). Offensichtlich eine Übersetzung (ins Holländische) und Beschreibung der Beschreibung (aus dem "Amerikanischen") eines industriell gefertigten Apparates (der Rezensent). Der RTX2001A ist eine sehr schnelle Forth-Maschine. Es wird gezeigt, daß er ohne komplizierte Zusatzausteile, wie Floppy-Controller oder AD/DA-Wandler, digitalisierte Audio-Signale aufnehmen und wiedergeben kann. Deltamodulation vereinfacht die AD/DA-Wandlung. Die mögliche Länge der Nachricht liegt derzeit noch bei nur 16 Sekunden. Aber das läßt

sich über Datenkompression und Extra-RAM auf 8 Minuten steigern. Das Programm paßt bequem in das 3K-Benutzer-RAM.

46 BANBOE
Willem Ouwerkerk

Beschreibung (mit Schaltbild) einer E/A-Karte, die an den parallelen Druckerport der meisten Computer und an den Userport des C64 gehängt werden kann. Erleichtert den Zugang zu den Hardware-Projekten des HCC-ANS-Forth-Kurses auf so ziemlich allen Computern. Ein paar Schieberegister sorgen dafür, daß man zur E/A-Ansteuerung mit einigen wenigen Bits am Druckerport auskommt.

56 Het inlezen van
BASICODE files in F-PC
Lennart Benschop

Bereits seit 10 Jahren überschwemmt der niederländische Rundfunk den Äther mit Computerprogrammen, die in BASICODE ausgestrahlt werden. Dies geschieht mit 1200 bit/sec, und die Programme können leicht softwaremäßig dekodiert werden. Für alle möglichen Computer, vom VC20 angefangen, wurden Programme entwickelt, um BASICODE von Kassette zu laden und ins RAM zu bringen. Will man BASICODE direkt aufnehmen, so muß man (mußte man bisher) Assembler einsetzen. Der Autor zeigt, daß man diese Aufgabe mit einem "schnellen" 80386/25Mhz auch in High-Level-Forth erledigen kann. (Anmerkung des Rezensenten: 80386 ? 25 Mhz ? Schnell ? Wie schnelllebig doch unsere Zeit ist !)

60 Het beste uit USENET
ohne Autor

18 Seiten Forth-Blockstruktur, konfrontiert mit dem DOS-Dateisystem. (Anmerkung des Rezensenten: Das war's ja gerade, was mich sofort auf Turbo-Forth von Marc Petremann und Co. zustürzen ließ - lange bevor Tom Zimmer bei uns (bei uns VDLern) aktuell wurde. Interessant, wie häufig und wie aufgeschlossen der Frage "Block oder nicht Block" gegenüber Bernd Paysan von der TU-München auftritt. Spricht mir aus dem Herzen. Muß doch mal die HP-Halle bei uns in der TU aufsuchen.)

□

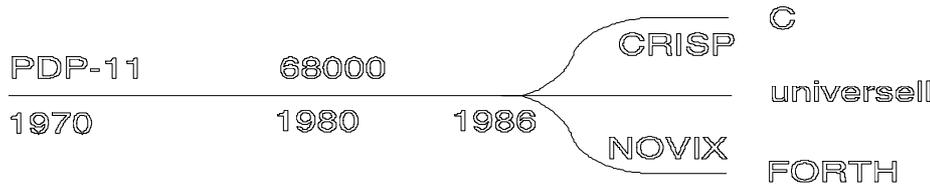
Hochsprachen in Silizium

von Rafael Deliano

Steinbergstr. 37; 82110 Germering; 089 / 841 83 17

Die Struktur problemorientierter Sprachen wie Fortran, Basic oder Cobol leitet sich aus der Anwendung ab, für die sie gedacht sind. Niedere Implementierungssprachen wie FORTH und C hingegen werden auf die Hardware abgestimmt, auf der sie laufen sollen. Sie sind in mancher Hinsicht nur hardwareunabhängiger Assembler.

Stichworte: Prozessoren Geschichte Forth-Prozessoren



Nikolaus Wirth hat es 1989 so formuliert:

“C ist keine ‘high level programming language’. C ist ein mit Syntax verzuckerter Assembler. Und wenn man die kryptische Notation ansieht, wird einem fast übel.”

Nämliches gilt für FORTH. Eine ursprüngliche Hardwarearchitektur spiegelt sich also in der Programmiersprache wider. Hat sich die Programmiersprache einmal etabliert, versucht man jedoch, für sie maßgeschneiderte Hardware zu entwickeln. Was bei der Rücktransformation in Hardware herauskommt, liegt allerdings meist weit neben dem Original.

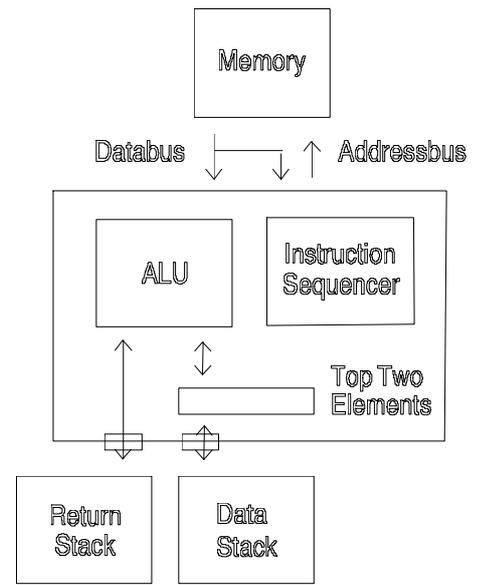
PDP-11

Die Architektur des PDP-11 Mini-computer von DEC scheint sowohl Charles Moore als auch Dennis Ritchie beeinflusst zu haben. Der Typ PDP-11 kam 1970 auf den Markt, speziell auch mit dem Anspruch, die Programmierung mit HLLs zu ermöglichen. FORTH und C entstanden zeitgleich 1971 bzw. 72. Der Entwurf der PDP-11 verwertete viele Erfahrungen aus früheren Mini-computern und Mainframes, nicht nur von DEC sondern auch von Konkurrenzunternehmen. Er war so erfolgreich, daß er sich - später bekannt als VAX - bis Mitte der 80er Jahre am Markt behaupten konnte.

Die PDP-11 hatte 5 universelle 16-Bit-Register. Sie konnten beliebig als Akkus und für die Addressierung verwendet werden. Die Register 6 und 7 waren Stackpointer und Programcounter. Die Maschine war mikroprogrammiert und hatte eine Unzahl möglicher Befehle. Z.B. 64 Varianten von ADD.

Die CPU war schwerpunktmäßig als 1-Adreß/GeneralRegister-Maschine ausgelegt. Die Speicherbewegung war Memory-to-Register. Für Bearbeitung mußten die Daten also in ein Register geladen werden. Einige 2-Adreß-Befehle (MOVE, ADD) waren jedoch vorhanden. Die Funktion war dann Memory-to-Memory. Register wurden dabei nicht verändert. Die Analyse der bis zu diesem Zeitpunkt auf DEC-Computern entwickelten Software ergab jedoch, daß sehr viel existierende Software stackorientiert codiert war. Der Grund waren die begrenzten Möglichkeiten der Vorgängermaschinen. Der Befehlssatz wurde deshalb so ausgelegt, daß er auch als 0-Adreß/Stackmaschine effektiv verwendbar war. @ ! + SWAP sind mit einem Assemblerbefehl realisierbar. Die

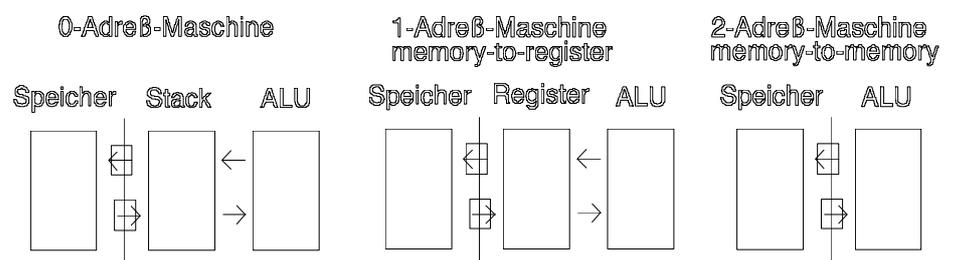
NOVIX NC4000A

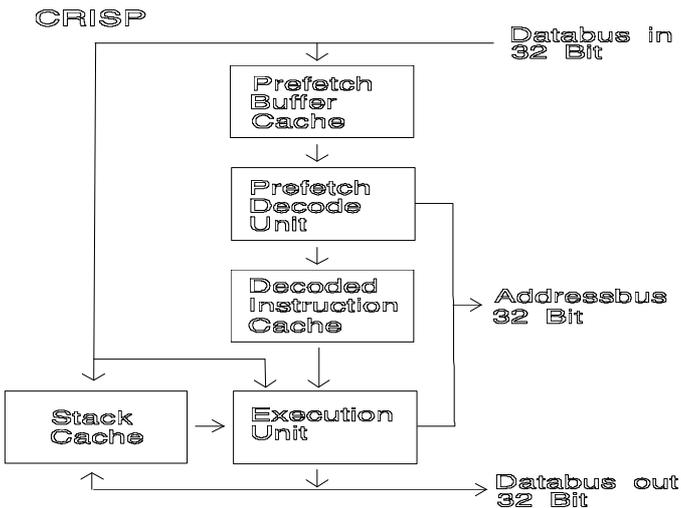


PDP-11 ist das, was man heute eine CISC-CPU nennen würde. Sie arbeitet registerorientiert. Die typische Mikroprozessorvariante dieses Konzepts ist der 68000 von Motorola (ca. 1980). Nur zur Verbesserung der Rückwärtskompatibilität wurde der Befehlssatz mit Stackbefehlen angereichert. Meist ist der Befehlssatz einer 0-Adreß-Maschine ohnehin in einer 1-Adreß-Maschine vorhanden.

NOVIX NC4000A

Dieser 16-Bit-Stackprozessors ist eine simple 0-Adreß-Maschine. Der Datenstack ist der Ersatz für Register. Die Variante wurde auf maximale Parallelisierung getrimmt: der Zugriff auf den Speicher und die beiden Stacks erfolgt gleichzeitig. Dem Geschwindigkeitsvorteil steht der Nachteil der hohen Kosten entgegen. Im Vergleich zu Registern ist der Zugriff auf einen Stack natürlich eingeschränkt. Deshalb sind Befehle wie SWAP OVER ROT DROP nötig. Für Register-Architektur entfallen diese redundanten Umladevorgänge. Daraus ergibt sich theoretisch bessere Code-





CRISP Instructions	Addr.	Function	Addr. mode	Data types
2, 2, 8 ADD	add	addition	immediate	unsigned byte
	sub	subtraction	absolute	signed byte
	mul	multiplication	stack offset	unsigned halfword
	div	division	STACK OFFSET, ind.	signed halfword
	and	bitwise		word
	or			
	shr	arithmetic shift right		
	lshr	logical shift right		
	lshl	logical shift left		
	lshr	logical shift right		
3 addr	umul	unsigned mult.		
	udiv	unsigned division		
	uror	unsigned remainder		
	ORP, =	OR operation		
	andp, =	AND operation		
	move	move effective addr.		
	move	move effective addr.		
	andl	interlocked bit, and		
	orl	interlocked bit, or		
3 addr	jmp	jump if true	absolute	
	ifp	jump if false	absolute (rel.,	
	ifcp	CALL	STACK OFFSET, ind.,	
	ifcpl	OverFlow	PC relative	
	call			
	call	return call	immediate	
	swap	swap stack space		
	swap	swap, and return		
	swap	swap Stack Cache		
3 addr	rop	reset register access		
	rdp	reset return		

dichte und höhere Geschwindigkeit. Simple Beispiel. Berechnet werden soll:

$$C = A + B$$

$$D = B * 2$$

2 Befehle auf Registerprozessor:

(r0 = A, r1 = B)

ADD r0, r1

MUL r1, #, 2

(C = r0, D = r1)

4 Befehle auf Stackprozessor:

(B, A --- C, D)

OVER + SWAP 2 *

Praktisch kann man primitive Stackprozessoren aber meist mit höherer Taktrate betreiben als komplexe CPUs. Da jene sehr wenige Befehle haben, kann man sie mit wenigen Bits codieren und das optimiert die Codedichte. Wenn man die realen technischen Gegebenheiten mitbewertet, liegt die reine Rechenleistung einer Registermaschine also nicht weit vor der eines Stackprozessors. Allerdings wird man Sprachen wie C hier nicht leicht effektiv implementieren können, was am Markt tatsächlich ein Nachteil ist.

CRISP

Etwa zum gleichen Zeitpunkt als der NOVIX auf den Markt kam, 1986, wurden bei AT&T auch die ersten Prototypen eines 32-Bit-C-Prozessors gefertigt. Die Definition der Architektur reicht bis 1982 zurück. Mit dem für Großfirmen üblichen gemächlichen Tempo dauerte es bis 1992, als ein

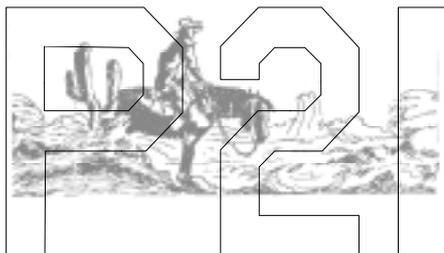
Serienchip unter dem Namen Hobbit verfügbar wurde. Erster Anwender war die Firma EO mit ihrem PDA, die allerdings inzwischen wieder das zeitliche gesegnet hat. AT&T will die Vermarktung trotzdem vorantreiben und hat weitere Varianten des Chipsatzes angekündigt. Schon der Ur-CRISP war mit 180k Transistoren nicht eben klein. Hobbit mit Cash-Speicher liegt bei >400k Transistoren. Im Gegensatz dazu benötigt die Stack-CPU SC-32, die sich jetzt sinnigerweise FRISC (=Forth-Risc) nennt, 18k Transistoren.

Der Befehlssatz des CRISP ist sogar für einen RISC-Prozessor ziemlich mager. Es wurde kein Versuch gemacht, komplexe C-Funktionen direkt in Silizium zu realisieren. Die Befehlsdekodierung erfolgt in einer 3-stufigen Pipeline. Die Häufigkeit der Branch-Befehle kann in C, wie Untersuchungen ergaben, bis zu 30% betragen. Damit die gewünschte Leistungssteigerung durch die Pipeline unter diesen Bedingungen noch gegeben ist, wurde das Sprung-Problem durch Parallelisierung (Branch-Folding) gelöst. Die Pipeline wurde dadurch natürlich dementsprechend komplexer. Das Problem ist aber gängig für RISC-CPU's und die Lösung nicht ungewöhnlich. Wirklich neu ist nur der Verzicht auf Register. Nach Untersuchung der CRISP-Entwickler machen Aufruf oder Rückkehr von einer Prozedur etwa 5% der Befehle aus, benötigen aber bei registerorientierten CPUs bis zu 50% der Laufzeit. Push & Pop der Register, die auf den Stack gerettet werden müssen sind dafür verantwortlich. Als Referenzmaschine für die statistischen Untersuchungen diente AT&T eine

VAX. Der CRISP verzichtet deshalb auf Register und hat eine Memory-to-Memory-Architektur. Für den Programmierer sieht es so aus, als ob er immer auf externen Speicher zugreift, ein angenehm einfaches Modell also. Externer Zugriff ist natürlich langsam und deshalb ist intern ein 64 Worte tiefer Stack-Cache vorhanden, durch den der tatsächliche Zugriff auf ein Minimum reduziert wird. Optimiert wurden am CRISP ferner Taskwechsel und Speicherverbrauch (20% besser als 68000). Er wurde also explizit als embedded Controller entwickelt. Zudem wurden kleinere Features für ADA eingebaut. Untersuchungen haben gezeigt, daß seine simple, klare Architektur nicht nur C, sondern alle konventionellen Sprachen von COBOL über ADA bis Modula-2 gut unterstützt. Letztlich ist der CRISP wohl kein "C-Prozessor" sondern eine innovative RISC-CPU. Die "semantische Lücke" zwischen Hardware und C wurde nicht geschlossen, eine echte C-CPU erwies sich als nicht kosteneffektiv.

Die Abkehr von der Registerarchitektur erfolgt hier analog zum Stackprozessor. Register wurden "erfunden", weil der Zugriff auf externen Speicher langsam ist. Sie sind eine reine Hardwarekrücke und für die Software eine Anomalität. Trotzdem beherrschen einstweilen registerorientierte CPUs das Feld. Auch Neuentwicklungen wie der PowerPC haben wieder 16 Register. Innovation bleibt somit Randerscheinung. Auch im Jahr 2000 wird man sich voraussichtlich noch nicht von der PDP-11 gelöst haben.

□



von Rafael Deliano

Steinbergstr. 37; 82110 Germering; 089 / 841 83 17

Charles Moores neuester Prozessor, P21 genannt, ist geeignet seinen Status als enfant terrible zu unterstreichen.

Mit 20-Bit Wortbreite, extremem Minimalismus der CPU, extern reichlich 1M-Word Speicher und einem programmierbarem Videointerface ist es nicht ganz klar, auf welche Anwendung (Videospiele?) er zielt. Andererseits ist er mit 60 Mips extrem schnell und zudem preiswert. \$25 einzeln, für Mitglieder der FIG \$20.

Stichworte: P21 Charles-Moore Forth-Prozessoren

Der Chip

Um den Preis niedrigzuhalten wurde ein 40-Pin DIL-Gehäuse gewählt. Leider ergab der erste Fertigungslauf durch die Plastikgehäuse Probleme, sodaß derzeit nur die Keramik-Typen funktionieren. PLCC44 wird möglicherweise bald verfügbar.

Die 20 Bit des Datenbus werden direkt herausgeführt, während der Adreßbus nur 10 Pins hat und auf 20 Bit gemultiplext werden muß.

Der Befehlssatz ist mit 24 Befehlen kompakt und von FORTH inspiriert. Er reicht aber nicht zur direkten Implementierung von Standard-FORTH-Systemen aus.

Die Tiefe des Parameterstacks auf dem Chip beträgt 6 Worte, die des Returnstack 4 Worte. Das ist die eigentliche Achillesverse der CPU. Man muß beim Programmieren aufpassen, daß kein Overflow eintritt. Moores Entwicklungssoftware zeigt deshalb auf dem Bildschirm immer die belegte Stacktiefe an, damit der Programmierer gewarnt ist.

Besonderer Gag ist ein programmierbarer Videogenerator für composite NTSC (384x482 Pixel). Nominell 15 Farben, die aus 8 Grundfarben und zwei Intensitätsstufen zusammengesetzt werden. Das Triggersignal wird ins Videosignal gemischt, damit diese Funktion nur einen Pin belegt. Man kann zwischen 4 Frames im RAM umschalten, wovon die Software aber derzeit noch nicht Gebrauch macht. Laut Moore ist auch PAL möglich.

Für konventionelle I/O steht ein Selectpin zur Verfügung. Zusammen mit den 10 Adreßleitungen können über ihn

1024 Ports aktiviert werden, die 8 oder 20 bit breit sein können.

Es können verschiedene Typen von Speicher verwendet werden. Typisch sind 1Mx20Bit DRAM. Man erreicht damit 50nsec Zugriff auf die selektierte Page, 150nsec auf nichtselektierte Page.

1Mx8Bit breiter statischer Speicher, EPROMs oder SRAM, ist auch möglich, aber mit 250nsec ziemlich langsam.

Die dritte Variante ist 1kx20Bit schnelles SRAM mit 25nsec für maximale Geschwindigkeit.

Eine typische Minimalversion enthält also 5 DRAMs 1Mx4Bit und ein EPROM. 256kx4-DRAMs eignen sich weniger, da der Speicherbereich nicht zusammenhängend ist. Praktisch muß man das EPROM nach dem Reset ins DRAM booten um akzeptable Geschwindigkeit zu erreichen. Aber bei der Speicherdichte heutiger DRAMs sollte das kein Nachteil sein und verändert die Kostenstruktur nicht wesentlich. Ein ultraschnelles System braucht zusätzlich die externe 1k-Page mit den teureren SRAMs.

Geeigneter Speicher vorausgesetzt, kann die Rechenleistung Spitzenwerte von 80-100 Mips erreichen. D.h. 10nsec pro Befehl, ohne Video 60 Mips, wenn Video mitläuft sind es allerdings nur 20Mips. Hauptbremse ist das Umschalten der Banks. CPU und Video greifen auf unterschiedliche Banks zu. Der Chip wird derzeit auf einer 1,2µm Fertigungslinie hergestellt. Skalierung auf kleinere Strukturbreite wird die Geschwindigkeit weiter erhöhen. Die Stromaufnahme der CPU beträgt bei 5V typisch 0,1Watt. Da die Komplexität nur bei 7k Transistoren liegt ist auch die Chipgröße mit 6 mm²

gering. Deshalb erwartet man eine hohe Fertigungsausbeute durch die der Preis der Serienchips auf \$10/10k fallen kann.

Das Board

Das ursprüngliche Demoboard hatte neben NTSC-Video eine Tastatur mit 8 Tasten. Das sind zwar fünf mehr als bei Moore üblich, aber immer noch weniger als bei einem Telefon. Als Massenspeicher war eine 128k PCMCIA vorgesehen. Diese sollte OK, Assembler, Videotreiber, Dokumentation, eFORTH usw. enthalten. Die Software ist sicherlich gewöhnungsbedürftig, da der P21 kein Standard-FORTH verträgt. Die Länge der Namen wurde auf 6 Zeichen beschränkt.

Man sah bald ein, daß PCMCIA ein teurer Spaß ist, weshalb die neue Version der Leiterplatte als Alternative konventionelles EPROM enthalten soll. Zusätzlich ist eine Erweiterung um schnelles SRAM vorgesehen. Preis derzeit \$100 für einen Bausatz dem einige Teile fehlen oder \$350 für das montierte Board

Der Vertrieb

Die Finanzierung der Entwicklung erfolgte durch Dr. Ting bei dem auch die Rechte für Herstellung und Vertrieb des P21 liegen. Foundry für die Muster war ORBIT, Beginn der Serie (10k Stück) September 94. Die Produktion soll später nach Taiwan verlagert werden, um die Herstellkosten zu senken.

Mit den nur teilweise funktionsfähigen Chips der Vorserie sind einige Boards nach Europa gegangen. Die Distribution hier allerdings noch unklar. In USA ist Dr. Ting zuständig:

Dr. C.H. Ting; Offete Enterprises, Inc.
1306 South B. Street San Mateo
CA 94402 (415) 574-8250

Vom nächsten Prozessor, dem F21, werden derzeit Muster gefertigt. Er soll eine Nummer größer werden und RGB als Video haben. Die Rechte dafür werden bei Jeff Fox liegen.

Jeff Fox; Ultra Technology
2510 10th St. Berkeley
CA 94710 (510) 848-0565
jfox@netcom.com

Er ist in comp.lang.forth erreichbar und auf seinen EMAILs beruht dieser Artikel.

□

Forth-Gruppen regional

Berlin	Claus Vogt Tel.: +30 -782 81 79 p Treffen: nach Absprache
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

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 -330 674 p
F-PC & TCOM, ASYST (Meßtechnik), embedded controller(H8/5xx//TDS2020, 8051 ... eFORTH...), FUZZY	Arndt Klingelberg Tel.: +2404 -6 16 48 agp
Gleitkomma-Arithmetik	Andreas Döring Tel.: +721 -59 39 35 p
HS/Forth (Harvard Softworks)	Wigand Gawenda Tel.: +30 -44 69 41 p
KI (Künstliche Intelligenz), OOF (Object Oriented Forth)	Ulrich Hoffmann Tel.: +431 -80 12 14 p
Unterricht mit Forth	Rolf Kretzschmar Tel./Fax: +2401 -8 88 91 ap
UUCP (FORTH ... per eMAIL)	Andreas Jennen Tel.: +30 -3 96 52 27 ap
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 98 p bis 22 Uhr 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