

Zur Softwareproblematik in der Meßtechnik

Softwarepaketen werden sich weiterhin "maßgeschneiderte" Softwarelösungen behaupten müssen. Gerade in der Meßtechnik, wo der Vervielfachung der Programme aufgrund der Spezifika nahezu jeder Applikation oft enge Grenzen gesetzt sind, bietet sich die Nutzung einer Programmiersprache wie FortH vorteilhaft an.

dest zu diesem Zeitpunkt vorrangig. Für die Meßwert-
erfassung und -verarbeitung steht wiederum eine um-
angreiche Palette an Softwareprodukten zur Verfügung.
Die Entscheidung für ein solches Produkt ist abhängig
von einer Reihe applikationsbedingter Größen und setzt
die unbedingte Kenntnis der meßtechnischen Problema-
tik voraus. Als Entscheidungskriterien für ein Software-
produkt können Bedienbarkeit, Adaptierbarkeit und
Verarbeitungsgeschwindigkeit angesehen werden [3].

Leicht bedienbare Systeme sind heute interaktiv, füh-
ren einen Bedienerdialog über Windows, Menüs und
Icons und sind weitgehend gegen Fehleingaben gesi-
chert. Systeme, die auf einem festen, meist durch den
Nutzer selbst zu erstellenden Programm beruhen, sind

den zur Verfügung stehenden Tools geprägt. Der Adaptierbarkeit von Software zur Meßwertfassung und -verarbeitung sollte besonderes Augenmerk geschenkt werden. Im allgemeinen ist in der Meßtechnik nicht nur eine einzige konstante Applikation vorhanden, sondern diverse. Auch bei nur einer Applikation können sich die Anforderungen im Laufe der Zeit ändern. Einfache Systeme erlauben die dann erforderlichen Anpassungen kaum oder nur mit beträchtlichen Eingriffen in Schichten der Software, die dem Anwender nicht ohne weiteres zugänglich sind. Flexible Systeme erlauben anwenderspezifische Programmänderungen und sichern damit eine ständige Adaptierbarkeit an das betreffende Problem.

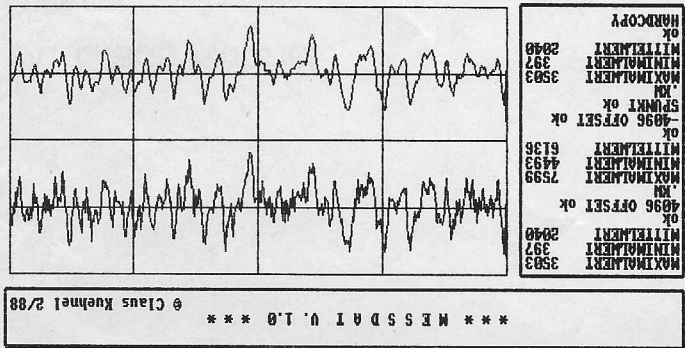


Bild 2. Darstellung von Meßwertverlauf und Bedien-Dialog

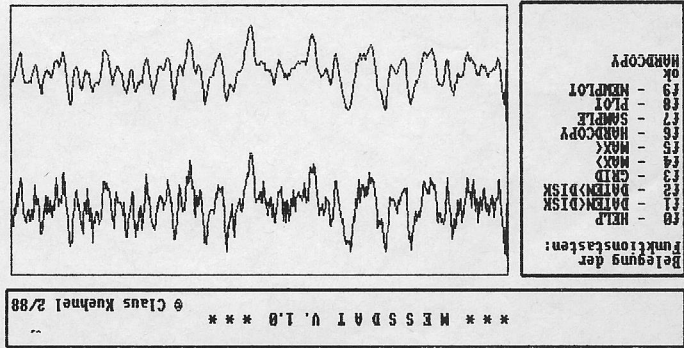


Bild 1. Darstellung von Meßwertverlauf und Funktionskassen-
belegung

Der Begriff Verarbeitungsschwindigkeit kann unter-
schiedliche Bedeutungen haben. Zum einen versteht
man darunter die Abtast- und Abspeicherungsrate
(Datendurchsatz) des Systems oder aber auch die Aus-
werte- (Kennwertermittlung) bzw. Darstellzeiten (Daten-
präsentation). Zum anderen kann damit die Reaktions-
zeit des Systems bei einer Echtzeitentscheidung
beschrieben werden.

Das gesamte Softwarespektrum erstreckt sich unter
Verwendung der drei Kriterien Bedienbarkeit, Adapter-
barkeit und Geschwindigkeit von einfach bedienbaren,
starren und langsamen bis hin zu kompliziert bedienba-
ren, flexiblen und schnellen Softwareprodukten.

Zur Auswahl eines die betreffende Applikation betrie-
bigen Softwareproduktes sind neben der Erfüllung
der meßtechnischen Aufgabenstellung die erforderli-
chen Programmierkenntnisse, die verfügbare Program-
entwicklungs- und -testzeit, die entstehenden Kosten
sowie zukünftige Applikationen zu betrachten. Da der
„Verschleiß“ eines verwendeten Personal Computers im
allgemeinen größer als der einer umfangreicheren Meß-
einrichtung ist, sollte die Applikation auch mit einem
neueren PC betriebsfähig sein. Portierbare Software ist
dafür eine unbedingte Voraussetzung.

Betrachtet man nun nach den beschriebenen Kriterien
die im Angebot befindliche Software, kommt man mit-
unter recht schnell zu dem Ergebnis, daß trotz eines
erheblichen Funktionsumfangs und dementsprechen-
der Kosten eine Applikation von einem Softwarepro-
dukt oft nicht in dem gewünschten Maße unterstützt
wird und demzufolge ein weiteres Produkt hinzugezo-
gen werden müßte. In diesen Fällen müssen dann noch
die erhobenen Meßdaten in das Format der verschiede-
nen Analyseprogramme konvertiert werden [4].

Vor diesem Hintergrund entsteht zwangsläufig die
Frage nach einer „maßgeschneiderten“ Meßtechnik-
Software, die dann natürlich alle Forderungen des Meß-
technikers in geeigneter Weise erfüllt. Nicht unter-
schätzt werden sollte jedoch bei einer solchen Entschei-
dung der resultierende Programmieraufwand, da nur
selten auf eine sog. „toolbox“, wie sie von Turbo-Pascal
oder „C“ bekannt ist, zurückgegriffen werden kann. Geht
man dennoch diesen Weg, sollte auf eine Programmier-

sprache zurückgegriffen werden, die interaktives Arbei-
ten und einen problemlosen Zugriff auf die Hard-
ware-Ressourcen des Personal Computers zuläßt, eine
einfache Bedienung der Schnittstellen zur Meßtech-
nik ermöglicht, Echtzeitanforderungen erfüllt und
eine weitgehende Standardisierung aufweist, die die
Portabilität der so geschaffenen Software zu sichern
hilft [5].

Die angegebenen Forderungen werden in hohem
Maße durch Forth sichergestellt, so daß mit diesem
Software-Werkzeug ein geeignetes Mittel für die Ent-
wicklung von Software für die Belange der Meßtechnik
zur Verfügung steht. Forth-Systeme sind jedoch bei wei-
tem nicht mehr auf Anlagen beschränkt, die mit wenig
Hardware auskommen müssen [6], sondern werden für
alle gängigen Personal Computer angeboten. Durch die
einfache Verknüpfung von Assembler- und Hochspra-
chenanteilen werden an zeitkritischen Stellen die Echt-
zeitanforderungen durch Assemblerroutinen (Primitive
Words) erfüllt. Desweiteren sichert die günstige Ver-
knüpfung von Assembler und Hochsprache auch den
einfachen Zugriff auf die Hardware-Ressourcen des Per-
sonal Computers. Der Nutzung von Assembler-Bibli-

otheken für den betreffenden Prozessor steht gleichfalls
nichts im Wege. Obwohl Forth selbst ausreichend
Betriebssystemeigenschaften aufweist, werden auch
Systeme angeboten, die das jeweils vorhandene
Betriebssystem (CP/M bzw. MS-DOS) und dessen
gesamte Fähigkeiten, wie Dateiverwaltung, Datum- und
Uhrzeit-Eingabe, nutzen.

Anhand des folgenden Beispiels sollen diese Aussa-
gen untermauert werden. Für die experimentelle Ana-
lyse langsam veränderlicher Signale, wie sie beispiels-
weise in der Biomedizin vorkommen, sollte ein
einfaches Instrumentarium zur Erfassung und Auswer-
tung von im allgemeinen als Spannungs-Zeit-Verlauf
vorliegenden Signalen geschaffen werden. Als Compu-
ter fand ein preiswerter, grafik- und windowfähiger Z80-
Microcomputer mit Diskettenlaufwerk sowie einem
achtkanaligen ADU Verwendung. Das Forth-System
wurde dabei so erweitert, daß der Bildschirm in drei
Fenster (2 Textfenster, 1 Grafikfenster) eingeteilt wurde.
Das eine Textfenster zeigt nur den Programmteil, wäh-

griff auf Hardware-Ressourcen und Analog/Digital-Umsetzer (ADU) verdeutlichen. Zum Einrichten des Grafikfensters wurden Firmware-Routinen des Rechners verwendet. Wie in Bild 3 zu sehen ist, wurden die Adressen der Routinen zum Setzen und Abfragen der Fensterbreite und -höhe als Konstanten definiert. Zur Parameterübergabe dienen die mittels CODE eingeleiteten Assembler-Routinen. Da die Grenzen des Grafikfensters intern auf Byte-Grenzen festgelegt werden, müssen die resultierenden Grenzen nach dem Setzen abgefragt werden. Die Umrandung wird schließlich als Colon-Definition ausgeführt. Wie Screen # 8 schließlich zeigt, ist die Definition des Grafikfensters mit den zwei Angaben 168 631 SETWIDTH 11 310 SETHEIGHT erledigt. Mit den vorstehenden Angaben erstreckt sich das Grafikfenster über eine Breite von 464 Pixel, beginn-

rend das zweite als Dialogfenster dient. Bei Zugriffen auf die Diskette erscheint im zweiten Fenster das komplette Directory. Der Aufruf häufig benötigter Programmfunktionen erfolgt über Funktionsstasten, deren Belegung im Sinne einer Help-Funktion im Dialogfenster ausgegeben werden kann. Alle weiteren Bedieneaktionen erscheinen ebenfalls in diesem Dialogfenster und werden bei korrekter Ausführung durch das Forth-typische „ok“ quittiert. Die grafische Darstellung des Meßwerteverlaufs geschieht in einem Grafikfenster mit einer Auflösung von 464 x 300 Bildpunkten. Die Bilder 1 und 2 sollen die Gestaltung des Bildschirmes verdeutlichen. In Bild 1 wird die Belegung der Funktionsstasten im Dialogfenster wiedergegeben. Im Grafikfenster ist im oberen Kurvenzug ein über den ADU erfaßtes, verrauschtes Signal wiedergegeben. Nachdem dieses Signal einem Glat-

Kurvenzug den resultierenden Signal-

verlauf. In Bild 2 hat sich nur der In-

halt des Dialogfensters verändert. Es

werden die vorgenommenen Manipu-

lationen am erfaßten Meßwertever-

lauf sichtbar. Das sind im einzelnen

die Veränderung des Offsets, der Auf-

ruf des Glatungsalgorithmus, die

grafische Ausgabe des so behandelten

Meßwerteverlaufes einschließlich der

Ermittlung von Maximal- und Mini-

malwert sowie das Einschalten eines

Gitters (Grid) über die Darstellung.

Mit einer ebenfalls implementier-

ten Hardcopy-Funktion kann der ge-

wünschte Bildschirminhalt über ei-

nen Epson-kompatiblen Matrixdruck-

ker „zu Papier“ gebracht werden.

Zur Zeit soll der Umfang der zur

Verfügung stehenden Algorithmen

um eine schnelle Fouriertansforma-

tion (FFT) ergänzt werden, die dann

Untersuchungen im Spektralbereich

zuläßt. Da Forth ursprünglich nur ei-

ne Integer-Arithmetik bereitstellt,

sind bestimmten Signalverarbeitungs-

Algorithmen gewisse Beschränkun-

gen auferlegt. Bei der softwaremäßi-

gen Realisierung einer Gleichkomma-

Arithmetik muß in jeder Program-

miersprache eine entsprechende Ver-

arbeitungszeit in Kauf genommen

werden. Wo dies unzumutbar er-

scheint, ist der Einsatz von Arithme-

tikprozessoren angezeigt. Für die 16-

Bit-Personal-Computer ist das durch

wobei es aber für die 8-Bit-Computer

mit einem AMD 9511 Lösungsmög-

lichkeiten gibt.

Zwei Listings (Bilder 3 und 4) sol-

len abschließend den einfachen Zu-

Bild 3. Listing zum Einrichten des Grafikfensters

```
Screen # 6
( WINDOWS: GRAFIK1
HEX
BB0F CONSTANT ADDRSETWIDTH
BB0E CONSTANT ADDRSETHEIGHT
BB0D CONSTANT ADDRGETWIDTH
BB0C CONSTANT ADDRGETHEIGHT
DECIMAL
CODE SETWIDTH
D POP H POP B PUSH ADDRSETWIDTH CALL B POP NEXT
CODE SETHEIGHT
D POP H POP B PUSH ADDRSETHEIGHT CALL B POP NEXT
Screen # 7
( WINDOWS: GRAFIK2
CODE SETWIDTH
D POP H POP B PUSH ADDRSETWIDTH CALL B POP NEXT
CODE SETHEIGHT
D POP H POP B PUSH ADDRSETHEIGHT CALL B POP NEXT
Screen # 8
( WINDOWS: DEMO
1 0 0 39 20 KOORD WINDOW1
2 0 21 79 24 KOORD WINDOW2
320 630 SETWIDTH
70 390 SETHEIGHT
: NOISE 641 320 DO I 350 RANDOM 50 + DRAW 2 +LOOP :
: NONSENS 1000 0 DO 256 RANDOM EMIT LOOP :
: DEMO
WINDOW1 FRAME SELECT CLS NONSENS CR ." WINDOW 1"
WINDOW2 FRAME SELECT CLS NONSENS CR ." WINDOW 2"
GRCLS GRFRAME NOISE 1 STREAM ;
```

CK 18/12/87)

CK 21/12/87)

CK 21/12/87)



Dr.-Ing. Claus Kühnel wurde in der Oberlausitz geboren. Nach dem Studium der Informationstechnik an der TU Dresden arbeitete er als Schaltungsentwickler der rechnergestützten Meßtechnik waren die Grundlage für eine freiberufliche Tätigkeit. Neben konventionellen PC-Anwendungen stehen derzeit die Vertiefung der FORTH-Philosophie sowie betreffende Meßtechnik-Applikationen an vorrangiger Stelle seiner Arbeit.

Literatur

- [1] Johannis, R.; Feger, O.: Peripheriebausteine integriert – Detailapplikation zum SAB 80515. ELEKTRONIK 1987, H. 1, S. 59...64.
- [2] Baum, E.: Neuer CMOS-Mikrocontroller. Elektroniker 1988, H. 3, S. 89...96.
- [3] Comparing Data Acquisition Software for Personal Computers. Application Note # 804, Keithley Instruments.
- [4] Formatting Instrumentation Data for LOTUS 1-2-3, ASYSTTM, and DADISP II. Application Note # 803, Keithley Instruments.
- [5] Kühnel, C.: Eignung verschiedener Hochsprachen für die Bearbeitung von Meßdaten. Elektroniker 1988, H. 3, S. 97...100.
- [6] Bergmann, E.; Gottschalk, K.: Einfaches Hochsprachen-Echtzeitsystem für Minimalssysteme. ELEKTRONIK 1986, H. 22, S. 121...130.

Ändert sich nun aus den unterschiedlichsten Gründen eine Komponente des Meßsystems, dann ist der Arbeitsumfang zur Adaptierung recht gering. Ist es beispielsweise erforderlich, einen ADU mit größerer Auflösung oder kürzerer Umsetzzeit im System einzusetzen, dann ist das Forth-Wort ADU (inkl. seiner Bestandteile ADUBLANK, ADUCONV und ADUREAD) zu verändern, und das System arbeitet schließlich mit der neuen ADU-Erweiterungskarte, so wie bislang mit der alten. Gleiches ist aufgrund der weitgehenden Standardisierung moderner Forth-Versionen (Standard Forth-83) und der damit prinzipiell gegebenen Portabilität der Software beim Einsatz neuer Personal Computer zu erhoffen.

Das gesamte hier vorgestellte System umfaßt das Forth-System selbst, einen Wordstar-kompatiblen Full-Screen-Editor, einen Z80-Assembler sowie den Applikations- und hat einen Umfang von insgesamt ca. 22-KByte, wobei auf den Applikationsteil nur ca. 4 KByte entfallen.

beginnend an der Position 168, und eine Höhe von 300 Pixel. In der gleichen Weise wird die Bedienung der ADU-Erweiterungskarte vorgenommen. Bild 4 zeigt in Screen # 10 das Ansprechen des ADU in einer Hochsprachendefinition (ADU), während in Screen # 11 das gleiche Problem durch eine kleine Assembleroutine (%ADU) gelöst wird. Die Hardware des ADU wird durch die niederwertigen vier Bits des Datenwortes gesteuert. Die Bits 0...3 geben die Kanaladresse des Multiplexers an, während Bit 4 = L den ADU löscht und Bit 4 = H den Umsetzvorgang startet. Die ADU-Erweiterungskarte wird am Port mit der Adresse F8F0_H betrieben. Diese Größen werden wiederum als Konstanten definiert. Die Steuerung des ADU erfolgt nun einfach durch die Bereitstellung des betreffenden Datenwortes (Kanaladresse OR ADU-Status) und der Portadresse. Das Einlesen des Ergebnisses der Umsetzung benötigt nur die Portadresse. Diese Bemerkungen sollen zur Charakterisierung des Problems ausreichen.

Bild 4. Listing zur Arbeit mit dem Analog/Digital-Umsetzer

```
( MESSAGE: MS ADU SAMPLE DVM
Screen # 10
HEX F8F0 CONSTANT PORT 0000 CONSTANT CLEAR
ADUBLANK KANAL CLEAR OR PORT PI ;
ADUCONV KANAL CONV OR PORT PI ;
PORT F8 ;
NORM 10000 255 * / ;
ZEIGE 1 17 GOT0XY 10 .R. " mV" ;
MS 0 DO 10 100US LOOP ;
ADUBLANK ADUCONV 1 100US ADUREAD ;
SAMPLE CLS LAENGE 1+ 1 DO ADU NORM I VERLAUF ; LOOP ;
DVM
Screen # 11
( MESSAGE: %ADU %SAMPLE
CK 6 / 3/88)
CODE %ADU
B D MOV C E MOV B POP
A XRA L ORA A OUT(C)
8 A MVI L ORA A OUT(C)
NOP NOP NOP NOP NOP NOP NOP
NOP NOP NOP NOP NOP NOP NOP
C IN(C) 0 B MVI B PUSH
D B MOV E C MOV NEXT
: %SAMPLE CLS KANAL PORT
LAENGE 1+ 1 DO 2DUP %ADU I VERLAUF ; LOOP 2DROF ;
CK 22 / 2/88)
```

Postkarte
genügt,

um uns Ihre Meinung
zu den Beiträgen dieses
Heftes mitzuteilen.
Nehmen Sie beispiels-
weise die am Ende
des Heftes beigefügte
Redaktions-Rücklaufkarte.