



PEARL-News

Ausgabe 1 Juni 2001

Mitteilungen
der GI-Fachgruppe 4.4.2
Echtzeitprogrammierung
PEARL

ISSN 1437-5966

Impressum

Herausgeber	GI-Fachgruppe 4.4.2 Echtzeitprogrammierung PEARL URL: http://www.real-time.de
Sprecherin	Prof. Dr.-Ing. B. Vogel Bergische Universität Fachbereich 13 Lehrstuhl Automatisierungstechnik / Prozeßinformatik Fuhlrottstraße 10 D-42097 Wuppertal Telefon: 0202/439-3848 Telefax: 0202/429-2944 E-Mail: bvogel@uni-wuppertal.de
Stellvertreter	Dr. P. Holleczeck Universität Erlangen-Nürnberg Regionales Rechenzentrum Martensstraße 1 D-91058 Erlangen Telefon: 09131/85-7817 Telefax: 09131/30 29 41 E-Mail: holleczeck@rrze.uni-erlangen.de
Redaktion	Prof. Dr. Dr. W. A. Halang FernUniversität Fachbereich Elektrotechnik und Informationstechnik Lehrstuhl für Informationstechnik, insb. Realzeitsysteme D-58084 Hagen Telefon: 02331/987-372 Telefax: 02331/987-375 E-Mail: wolfgang.halang@fernuni-hagen.de
ISSN	1437-5966

Redaktionell abgeschlossen am 12. Juni 2001

Einreichung von Beiträgen: Beiträge zu dieser Zeitschrift sind jederzeit hoch willkommen. Je früher und „rechtzeitiger“ sie eintreffen und je humorvoller sie sind, umso glücklicher ist der Redakteur. Einreichungen „auf den letzten Drücker“ und das Textverarbeitungsprogramm Word verleiten ihn hingegen zu Wutausbrüchen. Darum bittet er höflich um Übersendung der Beiträge per elektronischer Post in reinem ASCII und nicht codiert – eben ohne jeden Firlefanz – oder noch besser in LaTeX.

Inhalt

- 1 Programm des Workshops PEARL 2001
- 2 Bericht über die 18. Sitzung der Fachgruppenleitung
- 3 Protokoll der letzten Sitzung des Arbeitskreises 1
- 4 Bericht des Arbeitskreises 5
- 5 Verwendung von MATLAB-Routinen in PEARL-Programmen
- 6 Neues Lehrbuch für PEARL
- 7 PEARL 90 für Linux und Beispiele
- 8 Aufruf zur Mitarbeit in einem neuen Arbeitskreis

1 Programm des Workshops PEARL 2001

Auf seiner Sitzung am 16. Mai 2001 in Frankfurt/Main hat das Programmkomitee des Workshops PEARL 2001 das unten aufgeführte Programm zusammengestellt. In diesem Jahr steht die Veranstaltung unter dem Leitthema „Echtzeit-Kommunikation und Ethernet/Internet“. Wie gewohnt wurden natürlich auch wieder einige Beiträge zu aktuellen Echtzeitanwendungen ausgewählt.

Der Workshop wird im üblichen Rahmen im Hotel Ebertor zu Boppard am Rhein stattfinden. Der Tagungsband erscheint ebenfalls wieder in der Reihe „Informatik aktuell“ des renommierten Springer-Verlages. In Anbetracht des sehr interessanten und hochaktuellen Tagungsprogrammes, der immer anregend-freundlichen Atmosphäre in Boppard und nicht zuletzt unveränderter Tagungsgebühren sollten Sie sich noch heute zur Teilnahme anmelden, und zwar ganz bequem über die Leitseite der Fachgruppe mit dem URL www.real-time.de.

Erster Workshop-Tag: Donnerstag, der 22. November 2001

11:00 Treffen der Arbeitskreise

12:15 Imbiß

13:00 Begrüßung (Holleczek)

13:15 Sitzung *Automatisierung über Internet/Ethernet* (Holleczek)

Anforderungen an Ethernet-basierte Automatisierungssysteme und Web-basiertes Engineering aus der Verfahrens- und Fertigungstechnik
(Vogel, Bergische Universität Wuppertal)

Das Clausthaler Labor für „Plant Design and Virtual Manufacturing“
(Elzer und Sauermann, Technische Universität Clausthal)

Schutz von Automatisierungssystemen durch gerätetechnisch unterstützte Sicherheitsmaßnahmen
(Fitz und Halang, FernUniversität Hagen)

14:45 Pause

15:15 Sitzung *Echtzeit und Internet/Ethernet* (Heitmann)

Echtzeitdatenverkehr über IP-basierte Datennetze
(Hilgers, Regionales Rechenzentrum Erlangen)

Echtzeitkommunikation in der Automatisierungstechnik mit switched Ethernet
(Jasperneite, Phoenix Contact GmbH & Co. KG)

Ein/Ausgabe in der Automatisierungstechnik mit Internet-Protokollen
(Süßmann, Schneider Elektronik)

16:45 Pause

17:15 Sitzung *Harte Echtzeitkommunikation* (Gerth)

Funkdatenübertragung nach dem Zeitschlitzverfahren unter RTOS/PEARL
(Cseke, Cseke & Cie.)

SDI over ATM – Höchstqualitative Videoübertragung in Echtzeit
(Metz, Institut für Rundfunktechnik GmbH München)

18:30 Abendessen

20:00 Fachgruppensitzung

Zweiter Workshop-Tag: Freitag, der 23. November 2001

9:00 Sitzung *Programmiermethoden* (Vogel)

Zur Entwicklung zertifizierbarer fehlertoleranter Realzeit-Software mit Funktionsblockdiagrammen für die Automatisierung

(Thiele, Neimeier, Renner und Wendland, Universität Bremen)

Die Integration von UML-RealTime Komponenten in eine IEC 61131-3-Umgebung

(Heverhagen, Universität Essen)

Objektorientiertes Programmieren unter PEARL 90 bei einem Roboterprojekt

(Hofschulte, Universität Hannover)

10:30 Pause

11:00 Sitzung *Aktuelle Echtzeitanwendungen* (Halang)

Integriertes Daten-Management von Meß- und Videodaten

(Kneuer und Pornak, Werum GmbH)

Programmierung von vernetzten Prüfplätzen mit PEARL 90 am Beispiel der Lebensdauerprüfung von Waschautomaten

(Hein, Miele und Jensch, esd)

Embedded Systems – GATES oder Geht's nicht: Erfahrungen mit Windows NT Embedded

(Mangold, ATM)

12:30 Verabschiedung (Holleczek)

12:45 Imbiß

2 Bericht über die 18. Sitzung der Fachgruppenleitung

Vor der Sitzung des Programmkomitees des Workshops PEARL 2000 tagte am 16. Mai 2001 in Frankfurt/Main auch das Leitungsgremium der GI-Fachgruppe 4.4.2. Daran nahmen Frau Prof. Vogel sowie die Herren Arlt, Gerth, Halang, Heitmann, Holleczeck, Kaltenhäuser, Rzehak, Thiele und Windauer teil. Entschuldigt hatten sich Frau Prof. Danziger und die Herren Mangold und Müller. Dabei wurden u.a. folgende Tagungsordnungspunkte behandelt.

Haushalt: Die finanzielle Situation der Fachgruppe ist gut. Die Wirtschaftspläne für die Jahre 2001 und 2002 wurden bestätigt. Entsprechend wurde Frau Prof. Danziger für ihre Tätigkeit als Kassiererin entlastet. Die FGL dankte ihr für ihr Engagement. Der Vorschlag der GI, den Mitgliedsbeitrag in Euro auf die Hälfte des bisherigen DM-Betrages festzusetzen, wurde akzeptiert. Es wurde beschlossen, daß die Tagungsgebühren in Zukunft grundsätzlich im voraus zu entrichten sind.

Workshop PEARL 2001: Das Schwerpunktthema ist offensichtlich gut angekommen, so daß auf seiner Basis ein ausgewogenes Programm aufgestellt werden konnte. Die Ankündigung des Workshops im Internet ist aktuell. Deshalb wird für den Workshop 2001 keine Anzeige in der Fachpresse geschaltet. Statt dessen sollen die Möglichkeiten zur Ankündigung der Tagung in der Zeitschrift „Informatik Spektrum“ der GI sowie in den Veranstaltungslisten auf den Homepages der GI und des VDI KfIT genutzt werden.

Arbeitskreise: Frau Vogel regte die Gründung eines neuen Arbeitskreises zum Thema systematische Software-Entwicklung und Modellierung im Hinblick auf Echtzeitanforderungen an, was die FGL einhellig begrüßte. Für Näheres dazu wird auf Abschnitt 6 dieser Ausgabe verwiesen.

PEARL-Lehrbuch: Herr Prof. Rzehak hat freundlicherweise die Aufgabe übernommen, eine aktualisierte Neuauflage des Lehrbuches von Herrn Prof. Frevert zu erstellen (vgl. dazu auch den folgenden Abschnitt in dieser Ausgabe). Um die Herausgabe dieses Buches zu ermöglichen, faßte die FGL den Beschluß, die Abnahme eines Teiles der Neuauflage zu garantieren. Die Mitglieder der FGL wurden aufgefordert, Anregungen zum Inhalt an Herrn Rzehak zu leiten. Insbesondere ist es notwendig, Beschreibungen und Beispiele zum Umgang mit Zeigern zur Verfügung zu stellen.

3 Protokoll der Sitzung des Arbeitskreises 1

Das Treffen der PEARL-User-Group bzw. des Arbeitskreises 1 (Eingebettete Systeme, RTOS-UH/PEARL) der GI-Fachgruppe 4.4.2 fand am 7. Juni 2001 von 14:00 bis 16:30 Uhr mit 18 Teilnehmern am Institut für Regelungstechnik der Universität Hannover statt.

3.1 TOP 1: Weiterentwicklungen von PEARL 90

- Ein PEARL 90-Buch auf der Grundlage des Buches von Prof. Frevert befindet sich in Planung. Es wird von Prof. Rzehak überarbeitet bzw. ergänzt werden und im Teubner-Verlag erscheinen. Anregungen und Hinweise sind willkommen.
- Der PEARL-Workshop 2001 findet wie gewohnt statt. Nähere Informationen finden sich auf den WWW-Seiten der Fachgruppe 4.4.2: <http://www.real-time.de>. Der Workshop 2000 war mit 50 Teilnehmern gut besucht.
- Die Weiterentwicklungen des PEARL 90-Compilers wurden an Hand der nachfolgend aufgeführten History erläutert:

```
* 15.9-K 03/00 Extension: #INCLUDE + SETLINE *
* 15.9-L 03/00 Extension: BASIC+comment      *
*                                     Constant overflow msg*
* 15.9-M 03/00 Correction: Local Dations to even*
*                                     addresses      *
*                                     Extension: READ/WRITE*
*                                     with S-Format  *
* 15.9-N 04/00 Extension: No constants filed  *
*                                     for DCL-inits  *
*                                     Security SXB INPLT *
* 15.9-P 05/00 Extension: Allow constants as  *
*                                     proc-arguments at *
*                                     REF INV STRUCT[] *
* 15.9-Q 06/00 Correction: 68K only: S9 + end-lbl *
*                                     always even4      *
* 15.9-R 09/00 Correction: void formal with  *
*                                     naked proc as actual *
* 15.9-S 11/00 Extension: DEBUG-Mode only:   *
*                                     local array descr. add
* 15.9-T 11/00 Correction: WRITE BY S() Reg A1 *
*                                     sometimes not o.k. *
* 15.9-U 12/00 Extension: Only PowerPC:      *
*                                     C-Calls added  *
*-----*
* 16.1-A 03/01 Extension: Surpassing 64 kByte *
*                                     with LIDLT      *
* 16.1-B 03/01 Extension: Force CON-Purge     *
*                                     Correction: Stack setup C-Call *
*                                     PowerPC only  *
* 16.1-C 05/01 Extension: local structs in   *
*                                     DEBUG-T-Records *
* 16.1-D 05/01 Correction: PPC-C Call r12 saved *
```

TOP 2: Compiler und Laufzeitsysteme – Statusbericht und Ausblick

- Die Weiterentwicklungen des Nukleus wurden ebenfalls an Hand der History erläutert:

```
* 03/00 7.8-E Mess.passing of 7.8-A elim. (Ge)*7/8E
*                                     rebuild to normal I/O-Demon fac.*7/8E
```

```

* 08/00 7.8-F Error in TOQWA (did not clear *7/8F
*          CACT on empty queue)           *7/8F
* 08/00 7.8-G 2x buffered PEARL-Events (Ge/Wo)*7/8G
* 11/00 7.8-H No D1-Change in TOQ         (Wo)*7/8H
* 01/01 7.8-I PPC4xx: 700-ExcSlot angep. (Wo)*7/8I
* 01/01 7.8-J PPC: Traps mit Outer EE-Bit (Wo)*7/8J
*          DBGTRA-Trap (DebugTrace) reserv.*7/8J
* 03/01 7.8-K System on a Chip (555)     (A1/Wo)*7/8K
* 03/01 7.8-L 68k: No Trace in Superv-Mode(Wo)*7/8L
* 03/01 7.8-M RAM below EXCORG allowed (A1/Wo)*7/8M
*
* 03/01 7.9-A SYS-Option in DCDERR added .....*7/9A
* 04/01 7.9-B PPC:F1-P. Assist Exc. added (Wo)*7/9B

```

Besonders zu erwähnen sind hier die Versionen 7.8-G und 7.8-K.

Am Institut für Regelungstechnik wurde ein Verfahren zur automatischen Messung der Leistung von Echtzeitsystemen entwickelt. Als Kriterium dient die Phasenreinheit der Antwort auf ein zyklisch auftretendes Unterbrechungssignal in Abhängigkeit der Unterbrechungsfrequenz. Die Ergebnisse lassen sich übersichtlich als eine Art Frequenzgang darstellen. Bei der Vermessung von RTOS-UH wurde festgestellt, daß Ereignisse doppelt gepuffert werden sollten. Dieses wurde in der Version 7.8-G realisiert. Die Version 7.8-K enthält Erweiterungen, die es gestatten, RTOS-UH lediglich mit dem On-Chip-Flash und RAM des MPC555 zu betreiben. Externe Komponenten sind dann nicht mehr erforderlich.

- Die neue Einbaufunktion `CDL: PROC(par1 CHAR(1) IDENT, par2 REF STRUCT[] IDENT)` ermöglicht dynamisches Einrichten eines Links auf ein globales Symbol. Dabei ist `par1` eine Stringadresse und `par2` ein Zeiger auf Rückgabedatenstruktur.
- Der ATARI-Emulator WinSTon läuft mittlerweile recht robust. Herr Bunzel hat bereits eine rudimentäre Ausgabe auf dem Parallelport realisiert. Der Wunsch nach Prozeßein- und -ausgabe wurde bereits mehrfach geäußert. Am Institut wird zwar über mögliche Realisierungen nachgedacht, aber die Arbeiten am WinSTon haben geringe Priorität.
- Die Firma IEP hat erste Versuche mit dem HTTP-Server durchgeführt. Einige Sicherheitskomponenten wurden mit eingebaut. Neu ist ein PNG-Builder des IRT, der ein Bild komprimiert, um es via HTTP zu übertragen. Das Problem der Authentifizierung ist noch zu lösen.
- Das beim Verstellen der Uhrzeit aufgetretene Problem wurde ausführlich diskutiert. Ein `CLOCKSET` kann zum sofortigen Nachholen zahlreicher zyklischer Aktivierungen führen. Dieses soll zukünftig verhindert werden, wozu verschiedene Realisierungsmöglichkeiten diskutiert wurden. Gewünscht wurde, daß bei einer Einplanung der Form `AT 15:00 ALL 1 MIN ...` und einer Verstellung der Uhrzeit von z.B. 14:50:00 auf 15:02:30 die erste zyklische Aktivierung nicht sofort, sondern erst wie ursprünglich geplant um 15:03:00 erfolgt.

Generell gilt: Ist die neue Uhrzeit nach dem Verstellen kleiner als die alte, wird die Uhr immer zurückgestellt.

TOP 3+4: Neue RTOS-UH-Implementierungen, Berichte aus den Ingenieurbüros, Entwicklungsabteilungen und Forschungsinstituten

- Embedded Controllern auf Basis der MPC405 bzw. MPC555 werden bei den Ingenieurbüros immer öfter nachgefragt.
- Die Firma esd hat zwischen den X-Bus- und den VME-Bus-Systemen Compact PCI als neue Linie etabliert, für die bereits diverse Karten verfügbar sind.
- Die Firma IEP bietet einen PEARL-Hochsprachen-Debugger unter Windows mit einem Debug-Kernel auf dem RTOS-Zielsystem an.
- Die Firma Cseke, Jahns und Domeier arbeitet an einem Terminal/Graphik-Emulator für das Paket P-Graf von esd. Damit soll ein über das D-Netz vernetzter PC die gleichen Bilder wie ein Prozeßrechner zeigen.

- Die anwesenden Anwender von RTOS-UH berichten über den Einsatz in Medizintechnik, Mikrowellentechnik, Produktion und Qualitätssicherung. Hier kommen u.a. VME-Bus-Systeme zum Einsatz, die sich im Gegensatz zu manchen speicherprogrammierbaren Steuerungen als sehr robust erwiesen haben. Auch mikrocontrollerbasierte Systeme werden vermehrt als Ersatz für SPSen eingesetzt. Beim Einsatz von RTOS-UH in der Produktion ist Programmaktualisierung im laufenden Betrieb von großem Interesse, wie es z.B. mit dem CODESYS-SPS-System möglich ist.

Protokollführer:
 Dr.-Ing. T. Lilge
 Universität Hannover
 Institut für Regelungstechnik
 Appelstraße 11, 30167 Hannover
 lilge@irt.uni-hannover.de

4 Bericht des Arbeitskreises 5

Auf der 32. Sitzung des Ausschusses DIN NI-22 „Programmiersprachen“ am 30. November 2000 in Frankfurt/Main wurden Herr Prof. Giere (J.W. von Goethe-Universität Frankfurt/Main) auf eigenen Wunsch hin als Obmann verabschiedet und die Herren Breidhardt (IBM Stuttgart) und Küster (Rechenzentrum der Universität Tübingen) zum neuen Obmann bzw. seinem Stellvertreter gewählt. Aus Sicht der Fachgruppe soll die Tätigkeit von Herrn Prof. Giere an dieser Stelle ausdrücklich gewürdigt werden. Als Obmann des NI-22 hatte er entscheidenden Anteil am Erscheinen der Norm der Sprache PEARL 90. Weiterhin hat er sich intensiv dafür eingesetzt, PEARL im Rahmen des Komitees ISO SC 22 international bekannt zu machen, worin in den Ausgaben 1 und 2/2000 der PEARL-News berichtet wurde. Die Fachgruppe dankt Herrn Prof. Giere deshalb ganz herzlich für seinen Einsatz für die Sprache PEARL. Auf der Homepage des NI-22 mit dem URL <http://www.ni.din.de> ist PEARL mittlerweile auch präsent.

Im Rahmen seiner regulären Aktivitäten hat der Arbeitskreis damit begonnen, eine Liste von Projekten zusammenzustellen, die ganz oder teilweise in PEARL ausgeführt werden. Die Liste umfaßt zur Zeit 23 Einträge und ist als Vorlage für weitere Einträge vorläufig über den URL <http://www.iat.uni-bremen.de/thiele/PEARL-Projekte.html> erreichbar. Weiterhin setzt der AK die Initiative „PEARL-orientierte fehlertoleranzgerichtete Software für Realzeitanwendungen“ sowie die Erfassung der Hochschulen fort, die PEARL in der Ausbildung einsetzen. In diese Liste konnten die Hochschule für Technik und Wirtschaft Saarbrücken, die Fachhochschule Wilhelmshaven und die Berufsakademie Stuttgart neu aufgenommen werden. Die Liste mit weiteren Informationen ist unter <http://www.real-time.de> zu finden und umfaßt derzeit 21 Universitäten, (gleichbleibend) 23 Fachhochschulen unter 5 Berufsakademien.

Im Arbeitskreis wird auch die Einführung eines explizites Klassenkonzeptes in die Sprache PEARL diskutiert, um so die objektorientierte Programmierung zu unterstützen, da bisher im Hinblick auf harte Realzeitanwendungen „nur“ implizite Klassen als Realzeitobjekte verfügbar waren (vgl. dazu auch die OO-Checkliste des DIN NI-22). In der Diskussion sind die in der Ausgabe 2/1999 der PEARL-News und im Buch „Software-Entwurf in PEARL-orientierter Form“ von G. Thiele (Teubner-Verlag 1993) gemachten Vorschläge. Das Institut für Regelungstechnik der Universität Hannover hat seinerseits auch geeignete Vorschläge angekündigt.

Im WWW sind folgende Bücher mit signifikanten PEARL-Anteilen dokumentiert:

- B. Heimann, W. Gerth und K. Popp: *Mechatronik*. Fachbuchverlag Leipzig, 1998.
- U. Schneider und D. Werner: *Taschenbuch der Informatik*. 3. Auflage. Fachbuchverlag Leipzig, 2000.
- A. Ghassemi-Tabrizi: *Realzeit-Programmierung*. Springer-Verlag, 2000. (vgl. PEARL-News 2/2000)
- J. Heidepriem: *Prozeßinformatik 2*. R. Oldenbourg Verlag, geplant für 2001.

Die nächste, 11. Sitzung des Arbeitskreises findet am Rande des nächsten Workshops über Realzeitsysteme (PEARL 2001) am 22. November 2001 in Boppard statt. Teilnehmer und neue Mitglieder des Arbeitskreises sind dazu sehr herzlich willkommen.

5 Verwendung von MATLAB-Routinen in PEARL-Programmen

5.1 Einführung

Es ist bekanntlich nicht lohnend, das Rad nochmals zu erfinden. In diesem Sinne muß man auch bei der Software-Entwicklung i.a. danach trachten, geeignete vorhandene und bewährte Software-Bausteine in das zu entwickelnde Programm zu übernehmen und damit Zeit, Geld und Ärger zu sparen. In der Regel erfordert das gewisse Anpassungsarbeiten bzw. das Entwickeln geeigneter Schnittstellen. Der dafür erforderliche Aufwand ist jedoch häufig im Vergleich zum erreichbaren Nutzen gering, wie die in diesem Artikel beschriebene Anwendung der numerischen und graphischen Funktionen von MATLAB in PEARL-Programmen zeigt.

MATLAB ist ein Programmsystem für numerische Berechnungen und Visualisierungen mit einem sehr breiten Leistungsspektrum, effektiven Algorithmen und einer benutzerfreundlichen Arbeitsumgebung. Eine große Zahl von Toolboxen erweitert das Leistungsspektrum durch Bereitstellung zusätzlicher, oft sehr komplexer Funktionen für Aufgaben spezieller Gebiete, wie z.B. Signalverarbeitung, Optimierung oder Identifikation. Wegen dieser Vorteile wird MATLAB von sehr vielen Firmen, Hochschulen usw. eingesetzt und dabei meist als autonomes System verwendet. Es stellt aber auch Routinen zur Verfügung, mit deren Hilfe es durch andere Programme als Berechnungshilfsmittel nutzbar ist. Diese Routinen liegen in der „MATLAB Engine Library“ als Funktionen bzw. Prozeduren in den Sprachen C und FORTRAN vor.

5.2 MATLAB Engine Library

Diese Bibliothek umfaßt folgende 6 Funktionen:

engOpen startet MATLAB bzw. die MATLAB-Engine als eigenständigen Prozeß und stellt die Verbindung zur MATLAB-Engine her. Gegebenenfalls kann MATLAB auch auf einem separaten Rechner arbeiten.

engClose schließt die Verbindung zum MATLAB-Prozeß und beendet MATLAB.

engEvalString führt das als Parameter (String) übergebenen MATLAB-Kommando aus, kann aber auch die Ausführung kompletter MATLAB-Programme (M-Files) veranlassen.

engGetArray liest eine Variable (Skalar, Vektor, Matrix) der MATLAB-Arbeitsumgebung.

engPutArray überträgt eine Variable zur MATLAB-Engine bzw. in die MATLAB-Arbeitsumgebung.

engOutputBuffer erzeugt für **engEvalString** einen Zeichenpuffer zur Rückgabe der Zeichen, die bei der Ausführung von MATLAB-Kommandos im Dialogbetrieb auf dem Bildschirm angezeigt werden.

Die Funktionen und Einzelheiten zu ihrer Benutzung sind in den MATLAB-Handbüchern [1] und [2] beschrieben. Ihre Anwendung, insbesondere die von **engGetArray** und **engPutArray**, ist nicht ganz problemlos, weil MATLAB Variablen in einem speziellen Format ablegt und daher außerdem noch auf andere API- bzw. C-Funktionen zurückgegriffen werden muß, um Variablen von der MATLAB-Ebene zu übernehmen oder an diese zu übergeben. Das MATLAB-API arbeitet nur mit einem einzigen Datentyp, dem MATLAB-Array bzw. `mxArray`. Eine `mxArray`-Struktur enthält u.a. den Variablennamen, die Dimension der Variablen und ihren Typ sowie einen Identifikator für reelle bzw. komplexe Daten. Die Daten werden spaltenweise abgelegt. Ein Beispiel zur Anwendung der Funktionen des MATLAB-API für die Ausgabe eines Vektors von Gleitkommazahlen zeigt ein Programmausschnitt im nächsten Abschnitt.

Bei der Anwendung der Funktionen sind außerdem spezielle Maßnahmen zu berücksichtigen, beispielsweise um Laufzeitfehler beim beabsichtigten Lesen einer Variablen, die in der MATLAB-Arbeitsumgebung nicht bekannt ist, zu vermeiden.

5.3 Routinen zur einfachen Nutzung von MATLAB-Funktionen in PEARL-Programmen

Um die Nutzung der MATLAB-Engine zu vereinfachen, ist es sinnvoll, spezielle Unterprogramme als Schnittstellen bzw. Zugriffsfunktionen zu schreiben, denen bestimmte Aufgaben – z.B. Lesen einer skalaren Variablen oder Ausgabe eines Vektors – komplett übertragen werden. Für Demonstrationszwecke wurde daher die Bibliothek `engserv` entwickelt. Sie umfaßt vorerst nur C-Funktionen zur Ein- und Ausgabe von Skalaren und Vektoren, zum Öffnen und Schließen der MATLAB-Engine sowie zum Ausführen von MATLAB-Kommandos. Beispielhaft verdeutlicht der folgende Programmausschnitt von `engserv.c` die Realisierung der Ausgabe eines Vektors mit Hilfe der Funktionen des MATLAB-API und der C-Funktion `memcpy`.

```
/*  engserv.c
 *  C-Routinen zur Unterstuetzung der Anwendung von
 *  MATLAB-Funktionen in PEARL-Programmen
 * */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "engine.h"

Engine *ep;
mxArray *p = NULL; /* mxArray ist der Typ eines MATLAB-Arrays in C */

/** Ausgabe eines Vektors *****
 * x Adresse des Vektors
 * s Name des Vektors
 * n Zahl der Komponenten des Vektors
 * b Dummy; bedingt durch PEARL-Compiler
 */
void VectorOut(double x[],char b[], int n, char s[])
{
    p = mxCreateDoubleMatrix(1, n, mxREAL);
    mxSetName(p, s);
    memcpy((void *)mxGetPr(p), (void *) x, n*sizeof(double));
    engPutArray(ep, p); /* Senden MATLAB Array zur MATLAB Engine */
}
....
```

Das folgende PEARL-Programm demonstriert die Anwendung einiger Funktionen der Bibliothek `engserv` bei der Berechnung eines Approximationspolynoms, dessen graphischer Darstellung (Abb. 1) und der Übernahme der unter MATLAB berechneten Polynomkoeffizienten in das PEARL-Programm. Die von MATLAB auszuführenden Kommandos sind im Programm als Parameter der Aufrufe der `engserv`-Funktion `EvalString` angegeben.

```
MODULE (engdemo);
/* Beispiel fuer die Nutzung der MATLAB-Engine in einem PEARL-Programm
 *
 * Zwei Vektoren Temp und R1 beschreiben den messtechnisch erfassten
 * Zusammenhang zwischen dem Wert R1 eines Widerstands und der Temperatur
 * Temp. Das Programm nutzt MATLAB zur Loesung folgender Aufgaben:
 * - Ermittlung eines Approximationspolynoms fuer R1=f(Temp),
 * - graphische Darstellung der Funktion R1=f(Temp).
 * Abschliessend uebernimmt das PEARL-Programm die auf der MATLAB-Ebene
 * berechneten Koeffizienten k1(i) des Polynoms und zeigt das Polynom an.
```

```

SYSTEM;
  Bildschirm: STDOUT;
  CTRLC:      CTRL_C;

PROBLEM;
  DCL Anzeige DATION OUT ALPHIC CREATED(Bildschirm);
  SPC CTRLC   INTERRUPT;
  DCL Temp(6) FLOAT INIT (19, 25, 30, 36, 41, 47);
  DCL R1(6)   FLOAT INIT (76.3, 77.8, 79.75, 80.8, 81.6, 82.3);
  DCL k1(2)   FLOAT;

  /* Funktionen der Datei engserv.c */
  SPC EngineOpen  PROC GLOBAL;
  SPC EngineClose PROC GLOBAL;
  SPC EvalString  PROC(REF INV CHAR()) GLOBAL;
  SPC VectorOut   PROC(a() FLOAT IDENT, m FIXED, REF INV CHAR()) GLOBAL;
  SPC VectorIn    PROC(a() FLOAT IDENT, REF INV CHAR()) GLOBAL;

Start: TASK MAIN;
  OPEN Anzeige;
  EngineOpen;    /* Oeffnen der MATLAB-Engine */

  /* Uebergabe der Vektoren Temp und R1 an MATLAB */
  VectorOut(Temp, 6, 'Temp');
  VectorOut(R1, 6, 'R1');

  /* Berechnung der Koeffizienten des Approximationspolynoms 1. Grades */
  EvalString('k1=polyfit(Temp, R1, 1);');

  /* Berechnung der Polynomwerte an den Stuetzstellen mit MATLAB */
  EvalString('polwert=polyval(k1, Temp);');

  /* Graphische Darstellung des Approximationspolynoms und der Stuetzpunkte */
  EvalString('plot(Temp, polwert, '-','-', Temp,R1, ''+'');');
  EvalString('xlabel(''Temperatur [Grad C]');');
  EvalString('ylabel(''Widerstand [Ohm]');');
  EvalString('grid on;');

  /* Uebernahme des Vektors k1 der Polynomkoeffizienten von MATLAB und
  Anzeige des Polynoms */
  VectorIn(k1, 'k1');
  PUT 'R(Temp)=(', k1(2), ') + (', k1(1), ')*Temp'
      TO Anzeige BY SKIP, A, E(10), A, E(10), A, SKIP;
  PUT 'Eingabe von CTRL-C fuer Programmende' TO Anzeige BY A, SKIP;
  WHEN CTRLC RESUME;
  EngineClose;
  CLOSE Anzeige;
END;
MODEND;

```

Unter praktischen Gesichtspunkten wäre es natürlich sinnvoller, die im obigen Programm auf mehrere Aufrufe von `EvalString` verteilten MATLAB-Kommandos zum MATLAB-Programm `approx.m` zusammenzufassen und dieses mittels `EvalString('approx;')` zu starten. Dabei muß man nur sicherstellen, daß MATLAB die Datei `approx.m` auch findet, d.h. man muß das betreffende Verzeichnis in den MATLAB-Suchpfad aufnehmen. Das kann durch eine entsprechende Eintragung in der MATLAB-Arbeitsumgebung oder durch Ausführung der Anweisung `EvalString('path(path, ''verzeichnispfad'');')`; im PEARL-Programm erfolgen.

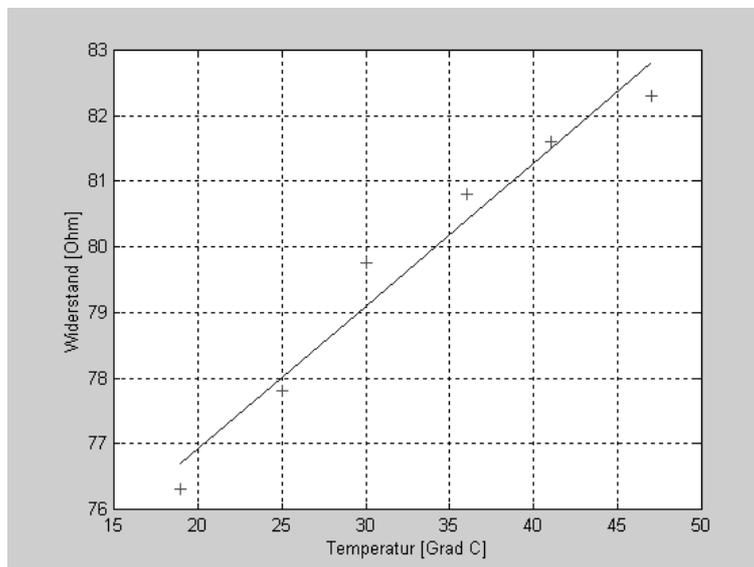


Abbildung 1: Vom Programm `engdemo` ausgegebenes Diagramm

5.4 Einbinden der MATLAB-Bibliotheken

Zum Erzeugen lauffähiger MATLAB-Engine-Applikationen werden aus den MATLAB-Dateien `libmx.dll` (array access library) und `libeng.dll` (engine library) erzeugte Importbibliotheken benötigt. Eine Analyse des zum MATLAB-Paket gehörigen Perl-Skripts `mex.bat` sowie der dafür bereitgestellten Optionendateien, z.B. `msvc50engmatopts.bat`, liefert die für das weitere Vorgehen nötigen Informationen. Mit `mex.bat` wird Übersetzen und Binden bei der Entwicklung von Engine-Applikationen mit C- oder FORTRAN-Programmen realisiert. Im Laufe der Ausführung dieses Programms werden die benötigten Importbibliotheken dynamisch erzeugt, zum Binden verwendet und am Ende wieder gelöscht. Löschen wird verhindert, wenn beim Ausführen von `mex.bat` – z.B. bei Anwendung auf das MATLAB-Demonstrationsprogramm `engdemo.c` im Verzeichnis `\extern\examples\eng\mat` (siehe [1, S. 6–5 ff.]) – die am Ende des Programms befindliche Löschanweisung `&expire("normally");` unwirksam ist (z.B. durch ein vorangestelltes Kommentarsymbol `#`). Die beiden Dateien mit der Namensendung `.lib`, deren Namen mit den Ziffern 1 und 2 enden und die beim Ausführen von `mex.bat` im Arbeitsverzeichnis (z.B. Verzeichnis von `engdemo.c`) erzeugt wurden, sind die benötigten Importbibliotheken. Man kopiert sie zweckmäßigerweise in das aktuelle PEARL-Arbeitsverzeichnis.

Nun müssen noch die entsprechenden Eintragungen in der Kommandodatei des Binders vorgenommen und einige Umgebungsvariablen aktualisiert werden. Die folgende Beschreibung dieser Schritte bezieht sich auf die Verwendung des PEARL-Systems für Windows NT der Firma Werum.

1. Modifikation der Datei `P90bind.cmd`

Damit die oben genannten Dateien beim Binden berücksichtigt werden, müssen sie an die Datei `command.link` angehängt werden. Der folgende Programmausschnitt verdeutlicht die dafür notwendige Änderung der Datei `P90bind.cmd`.

```

.....
REM Default CONFIGURATION-Modul
echo %PEARL90PATH%\defmodul.obj >> command.link
REM Libraries
echo pearl90i.lib msvcrt.lib oldnames.lib kernel32.lib advapi32.lib >> command.link
REM Anhaengen der MATLAB-Importbibliotheken
echo _lib71781.lib _lib71782.lib >> command.link
link @command.link
.....

```

Die Namen der beiden anzuhängenden Dateien im dargestellten Programmausschnitt stehen nur als Beispiele. Die tatsächlichen Namen vergibt `mex.bat` automatisch. Sie müssen daher vom Anwender,

der das System konfiguriert und sich zu diesem Zweck die beiden Dateien in der oben beschriebenen Weise erzeugt hat, aktualisiert werden.

2. Modifikation der Datei P90conf.cmd

Für fehlerfreies Funktionieren sind außerdem die Umgebungsvariablen PATH und INCLUDE zu aktualisieren. Zu diesem Zweck wurde für die Tests mit dem oben beschriebenen Beispiel die Datei P90conf.cmd wie folgt ergänzt:

```
set MATLAB=G:\MATLAB52
set MSVC_ROOT=C:\Programme\DevStudio
set MSVCDir=%MSVC_ROOT%\VC
set MSDevDir=%MSVC_ROOT%\sharedIDE
set PATH=%MSVCDir%\BIN;%MSDevDir%\bin;%MATLAB%\extern\include;%MATLAB%\bin;%PATH%
set INCLUDE=%MSVCDir%\INCLUDE;%MSVCDir%\MFC\INCLUDE;%MSVCDir%\ATL\INCLUDE;
    %MATLAB%\extern\include;%INCLUDE%
set LIB=%MSVCDir%\LIB;%MSVCDir%\MFC\LIB;%LIB%
cmd.exe
```

Dieser Programmausschnitt umfaßt nicht nur die Aktualisierungen für das Zusammenwirken mit MATLAB, sondern auch die Einstellungen für den C-Übersetzer.

5.5 Schlußbemerkungen

Es ist geplant, die Bibliothek **engserv** weiter auszubauen und beispielsweise Funktionen zur Ein- und Ausgabe von Matrizen zu integrieren. Außerdem muß das Zeitverhalten der Programme bzw. Tasks mit MATLAB-Unterstützung noch genauer untersucht werden.

Interessenten können die Bibliothek **engserv** über die unten genannte Adresse anfordern. Für Hinweise auf eigene Erfahrungen bei der Nutzung der MATLAB-Engine sowie Anregungen zur Verbesserung bzw. zum Ausbau der Bibliothek wäre der Autor dankbar.

5.6 Literatur

1. MATLAB Application Program Interface Guide. Version 5. The MathWorks, Inc.
2. MATLAB Application Program Interface Reference. Version 5. The MathWorks, Inc.

Prof. Dr.-Ing. R. Müller
HTWK Leipzig
Fachbereich Elektrotechnik und Informationstechnik
Wächterstraße 13, 04107 Leipzig
mueller@et.htwk-leipzig.de

6 Neues Lehrbuch für PEARL

Bei den Hörern der verschiedenen PEARL-Kurse und anderen Interessenten besteht großes Interesse an einem eingängig geschriebenen Lehrbuch für PEARL, da der Sprachreport eher die Qualität eines Nachschlagewerkes für Zweifelsfragen hat. Daher wurde auf der PEARL-Tagung 1999 von verschiedenen Teilnehmern angeregt, das PEARL-Lehrbuch von Herrn Frevert in einer Neubearbeitung auf der Basis von PEARL 90 neu aufzulegen, um wieder ein aktuelles Lehrbuch zur Verfügung zu haben. An Herrn Freverts Buch wurde insbesondere der gute didaktische Aufbau sehr gelobt. Der damaligen Diskussionsrunde war dabei klar, daß dies erhebliche Arbeit bedeutet, die von Herrn Frevert aus objektiven Gründen nicht geleistet werden kann. Etwas spontan habe ich mich damals bereiterklärt, diese Aufgabe zu übernehmen, wofür mir hinsichtlich verschiedener Teilaufgaben Unterstützung zugesagt wurde. Die Korrespondenz mit dem Verlag zog sich in die Länge, wobei klar wurde, daß das Buch zu einem erschwinglichen Ladenpreis nur aufgelegt werden kann, wenn der Verlag gewisse Abnahmegarantien erhält. Nachdem nun solche von

unserer Fachgruppe und einigen Firmen abgegeben worden sind, kann ich hiermit ankündigen, daß das Buch baldmöglichst verfügbar sein wird. Natürlich haben die Götter vor den Erfolg den Schweiß gesetzt, so daß ich den Begriff „baldmöglichst“ noch nicht näher interpretieren kann. So viel steht jedoch schon fest:

- das Buch wird wieder im Teubner-Verlag erscheinen, der jetzt zur Bertelsmann-Gruppe gehört,
- sein Seitenumfang bleibt unverändert,
- der Ladenpreis wird etwa DM 50 betragen und
- bei neuem Layout werden die inhaltlichen Aspekte (z.B. Syntaxdiagramme) erhalten bleiben.

An dieser Stelle möchte ich allen, die sich zur Mithilfe bereit erklärt haben, ausdrücklich danken. Und Hilfe werde ich mit Sicherheit benötigen.

Prof. Dr. H. Rzehak
rz@informatik.unibw-muenchen.de

7 PEARL 90 für Linux und Beispiele

Seit dem 1. Juni 2001 steht das PEARL 90-Programmiersystem für Linux mit Beispielen auch auf dem ftp-Server der Firma Werum GmbH in Lüneburg zur Verfügung:

`ftp://ftp.werum.de/pub/pearl90`

Herrn Dr. Hans Windauer sei dafür sehr herzlich gedankt.