



PEARL-News

Ausgabe 2 November 2000

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. Birgit Vogel Bergische Universität Fachbereich 13 Lehrstuhl Automatisierungstechnik / Prozeßinformatik Fuhlrottstraße 10 D-42097 Wuppertal Telefon: 0202/439-3848 Telefax: 0202/429-2944 E-Mail: vogel@lfa.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 D-58084 Hagen Telefon: 02331/987-372 Telefax: 02331/987-375 E-Mail: wolfgang.halang@fernuni-hagen.de
ISSN	1437-5966

Redaktionell abgeschlossen am 10. November 2000

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 E-Mail in reinem ASCII und nicht codiert – eben ohne jeden Firlefanz – oder noch besser in LaTeX.

Inhalt

1	Editorial
2	PEARL mit RTOS-UH für PC unter Windows
3	Bericht des Arbeitskreises 5
5	GI-Seminar über Forschungs- und Entwicklungsperspektiven der Echtzeitinformatik
5	Der PEARL-Kurs der FernUniversität

1 Editorial

Endlich ist sie da: die lang ersehnte Portierung von PEARL mit RTOS-UH in die schnöde “Wintel-Welt”. Es handelt sich dabei um eine Emulation, die am Institut für Regelungstechnik der Universität Hannover entwickelt wurde und im folgenden Abschnitt beschrieben wird. Wir freuen uns, daß diese Ablaufumgebung als Freeware im Netz zur Verfügung steht.

Der Bericht des Arbeitskreises 5 “PEARL in der Ausbildung und Sprachpflege” kann eine weitere Erhöhung der Anzahl von Hochschulen vermelden, an denen PEARL gelehrt wird.

Die Schwesterfachgruppe 4.4.1 plant, ein GI-Seminar über Forschungs- und Entwicklungsperspektiven der Echtzeitinformatik zu veranstalten. Dieses Projekt wird kurz vorgestellt und es wird zu aktiver Mitarbeit daran aufgerufen.

Vom PEARL-Kurs der FernUniversität ist schließlich zu berichten, daß das Bestellverfahren vereinfacht und der Preis gesenkt wurden sowie daß eine verbesserte Programmierumgebung zur Verfügung steht.

2 PEARL mit RTOS-UH für PCs unter Windows

2.1 Einführung

Mit Hilfe des Atari-ST-Emulators “WinSTon” ist der Einsatz des Programmiersystems PEARL RTOS-UH jetzt auch auf Standard-PCs unter Windows möglich. Im von WinSTon emulierten Atari ST läuft dazu das Echtzeitbetriebssystem RTOS-UH.

Das RTOS-UH-System für WinSTon erreicht ab der Mindest-Hardware-Anforderung P MMX 166 MHz etwa die Leistungsfähigkeit eines originalen Atari ST mit 8 MHz unter RTOS-UH. Bei entsprechender Hardware wird sie sogar um ein vielfaches übertroffen. WinSTon emuliert nicht nur den 68000-Prozessor, sondern auch wesentliche Peripherie-Hardware eines Atari ST. Daher ist bereits ein originär für den Atari ST vorgesehenes RTOS-UH mit ihm lauffähig.

Das RTOS-UH-System für WinSTon ist lediglich um einen Treiber für den Dateitransfer zum PC-Dateisystem erweitert worden und entspricht somit in seinen Systemkomponenten dem aktuellen Stand. Es enthält zwar (wie das RTOS-UH für alle Ataris) keine graphische Bedienoberfläche, da aber Compiler, Assembler und Editor für PEARL 90 enthalten sind, sind Programmentwicklung und -ausführung uneingeschränkt möglich. Der PEARL 90-Compiler ist hinsichtlich seiner Betriebssystemaufrufe von der verwendeten Hardware unabhängig. Insofern können auch PEARL-Programme getestet werden, die für modernere Ziel-Hardware (z.B. PowerPC) entwickelt werden. Dies kann bei der Cross-Entwicklung eingebetteter Systeme sehr hilfreich sein, da jetzt viele Programmtests nicht nur auf dem Zielsystem ausgeführt werden können, sondern bereits vorher mit dem WinSTon-RTOS-UH-System.

2.2 Eigenschaften

WinSTon sollte mit allen Varianten von Microsoft Windows ab Windows 95 arbeiten. Der emulierte Atari ST läuft als Windows-Task wahlweise im Vollbildmodus oder in einem eigenständigen Fenster. Einen Screenshot der RTOS-UH-Startmeldung zeigt Abbildung 1.

Die Arbeitsgeschwindigkeit von WinSTon läßt sich in mehreren Stufen zwischen der Geschwindigkeit eines originalen Atari ST mit 8 MHz und einem Maximalwert einstellen. So erreicht z.B. ein PC mit einem 500 MHz schnellen K6II-Prozessor etwa die Leistung eines Atari ST mit 50 MHz. Auf diese Werte hat aber auch die Graphikkarte des PC großen Einfluß. Die in RTOS-UH geführte Zeit stimmt allerdings nur im Geschwindigkeitsmodus Atari ST mit der realen Zeit überein. Hierfür ist bereits ein Treiber in Arbeit, der die RTOS-UH-Zeit entsprechend anpaßt.

Ein wesentliches Merkmal von RTOS-UH, die hervorragende Echtzeitreaktivität, ist bedingt durch die Emulation und das darunter liegende Windows natürlich eingeschränkt. Die Reaktivität rührt hier vor allem vom Leistungsüberschuß der PC-Hardware her. Bei starker Hintergrundaktivität des Windows-Betriebssystems kommt es daher zu “Verzerrungen” der Zeit im emulierten Atari.

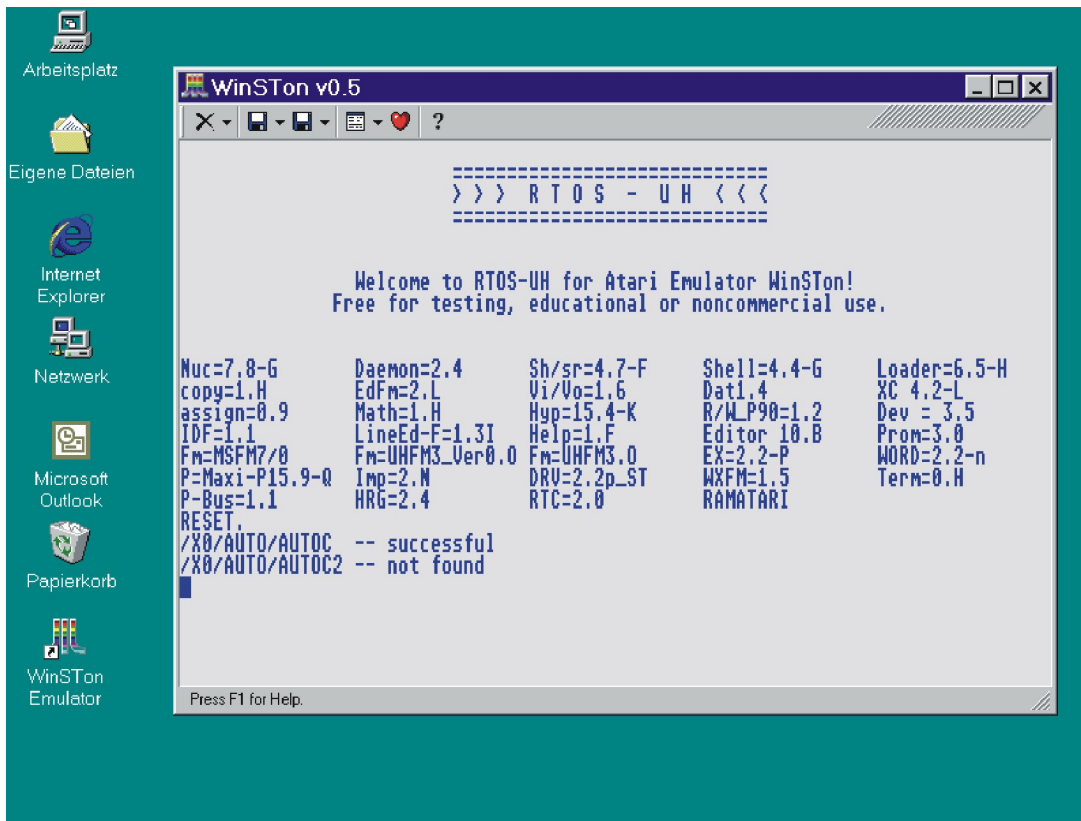


Abbildung 1: RTOS-UH Startmeldung

Auch Zugriffe auf das Windows-Dateisystem behindern das Echtzeitverhalten, da der hierfür vorgesehene Treiber diese in unteilbaren Sequenzen über Ressourcen des Emulators an Windows weiterleiten muß, d.h. bis zur Rückmeldung von Windows steht der emulierte Atari still.

Die Verwendung externer Unterbrechungen ist zur Zeit nicht möglich. Eine Lösung für digitale Ein-/Ausgabe über Parallelports des PCs ist aber bereits in Arbeit.

2.3 Installation

Zunächst sollte der Atari ST-Emulator WinSTon und anschließend RTOS-UH installiert werden. Aktuelle Download-Links und detailliertere Hinweise zu den Versionen sind unter

<http://www.irt.uni-hannover.de/rtos/freeware/winston-rtos.html>

auf dem WWW-Server des Instituts für Regelungstechnik zu finden.

Das RTOS-UH-System für WinSTon ist als ca. 300 kB großes Archiv gepackt. Es kann in ein beliebiges Verzeichnis im Windows-Dateisystem entpackt werden. Dieses Verzeichnis ist dann in WinSTon als Laufwerk C: des Atari einzustellen. Im Ordner AUTO liegt das Programm `rtoslod8.prg`, mit dem das RTOS-UH-System von TOS aus gestartet werden kann.

WinSTon selbst enthält bereits ein TOS-Image der Version 1.00, das für den Start von RTOS-UH ausreicht. Ansonsten benötigt RTOS-UH kein TOS und ist daher von der verwendeten TOS-Version unabhängig.

Im Ordner XD befinden sich einige nützliche Skripte. Durch die Autostart-Skripte des Systems wird u.a. dieser Ordner als Execution-Directory eingestellt und die Shell mit LineEdit (ähnlich Doskey) geladen.

Hilfe bei der Bedienung von RTOS-UH insbesondere des Editors, des PEARL 90-Compilers oder des Assemblers gibt das etwa 700 seitige Handbuch, das unter

<ftp://ftp.irt.uni-hannover.de/pub/rtos-uh/HANDBUCH/AKTUELL/>

als PDF- (2.4 MB), PS- (2.5 MB) oder DVI-Dokument (1.8 MB) herunterladbar ist. Für neue Anwender von RTOS-UH ist vor allem der Schnellkurs in Kapitel 2 zu empfehlen.

2.4 Spezielle Einstellungen von WinSTon

Für den Einsatz von RTOS-UH unter WinSTon sind einige Einstellungen sinnvoll, die von den Default-Optionen von WinSTon abweichen. WinSTon enthält "Help topics", die diese Optionen erläutern.

Hard Disks

Hier ist als Laufwerk C: das Verzeichnis einzustellen, in das das Archiv des RTOS-UH-Systems entpackt wurde. Mit eingeschaltetem 'Boot from Hard Drive' startet RTOS-UH beim Reset von WinSTon automatisch (falls keine Disketten-Images eingestellt sind).

Keyboard

Um mit WinSTon im Maximum Speed Modus zu arbeiten, sollte 'Disable key repeat on maximum speed' gesetzt werden. Anderenfalls ist die Tastaturwiederholrate auf schnellen Rechnern viel zu hoch. Weiterhin empfiehlt es sich, zu den Default Short Cuts

- F11: Full Screen (schaltet zwischen Full Screen und Window Modus um) und
- F12: Mouse Mode (schaltet zwischen Atari-Emulation und Windows um)

noch

- Shift+F11: Maximum Speed (schaltet zwischen Default Speed, die im Configure Menu gesetzt ist, und Full Speed um)

zu setzen.

Mit der Default-Tastaturbelegung sind unter RTOS-UH einige Tasten (z.T. Atari-spezifisch) nicht zugänglich. Mit der in der Datei `rtos.map` enthaltenen Belegung liegt zumindest '#' richtig und Atari-'HELP' auf 'Rollen'. Auf Grund der unterschiedlichen Behandlung der Shift-Kombinationen bei Atari und PC ist getrenntes Umsetzen einer Taste und ihrer Shift-Kombination nicht möglich. Mit der `rtos.map` liegt daher das zum Programmieren wichtige Hochkomma direkt rechts neben 'ß' (ohne Shift). Vorschläge für eine sinnvollere Belegung sind willkommen. WinSTon enthält zum Testen das Feld 'Test Key'. Das Prinzip der Scancode-Zuordnung in der `.map`-Datei ist selbsterklärend und die Dateien sind mit einem ASCII-Editor einfach zu bearbeiten.

Joysticks

Sofern kein Joystick angeschlossen ist, stellt WinSTon per Default für Joystick1 'Use cursor emulation' ein. Dann arbeiten die Pfeiltasten nur bei gleichzeitiger Betätigung der Shift-Taste wie unter RTOS-UH gewohnt. Daher wird die Abschaltung dieser Option empfohlen.

Screen

Die hier eingestellten Optionen Display Mode, Colour/Mono, Full Screen usw. haben großen Einfluß auf die Leistung von WinSTon. Nicht mit jeder Graphikkarte sind die als langsamste bzw. schnellste Einstellung angegebenen Optionen tatsächlich die langsamste bzw. schnellste. Daher sollten hier verschiedene Modi ausprobiert werden.

2.5 Start von RTOS-UH

RTOS-UH kann manuell über TOS von C:\AUTO durch das Programm `RTOSLOD8.PRG` gestartet werden. Automatisch wird dieses Programm bei jedem Reset von WinSTon aufgerufen, falls die Option 'Boot from Hard Drive' angewählt und kein Disketten-Image als Laufwerk eingestellt ist. Beim Start von RTOS-UH ist es sinnvoll, an der PC-Tastatur 'CapsLock' zu aktivieren, da das RTOS-UH-System intern beim Start von eingeschaltetem 'CapsLock' ausgeht. Anderenfalls sind die Groß- und Kleinbuchstaben unter RTOS-UH vertauscht und 'Shift' und 'CapsLock' schalten nicht richtig um.

Auf das Windows-Dateisystem, also die Atari-Harddisks C:, D:, E: oder F: wird von RTOS-UH aus über die Laufwerksbezeicher /X0, /X1, /X2 bzw. /X3 zugegriffen. Auf diese Weise können auch Netzlaufwerke des Windows-Dateisystems angesprochen werden. So ist es z.B. möglich, daß das WinSTon-RTOS-UH auf den Samba-Server eines Prozeßrechners (nativ laufendes RTOS-UH) zugreift.

Der Zugriff auf die unter RTOS-UH sonst üblichen Bezeichner für Festplatten, z.B. /H0, liefert lediglich die Fehlermeldung `DRIVE_NOT_PRESENT`.

2.6 Ausblick

Am Institut für Regelungstechnik der Universität Hannover wird das RTOS-UH-System für WinSTon weiterentwickelt. Daher freuen wir uns auf Anregungen, Kritik und auch Hinweise auf Fehler.

Zur Zeit konzentriert sich die Weiterentwicklung auf zwei bereits erwähnte Ziele. Die RTOS-UH-Systemuhr soll adaptiv an die reale Zeit angepaßt werden. Damit würde Echtzeitbetrieb auch bei maximaler Geschwindigkeit von WinSTon innerhalb von RTOS-UH möglich. Das zweite Ziel besteht darin, digitale E/A über Parallelports von PCs bereitzustellen. Erste Tests hierzu zeigten aber, daß auf Grund gelegentlicher Zeitverzerrungen ein Echtzeiteinsatz nur etwa im Anforderungsbereich einiger 100 ms realistisch ist.

2.7 Danksagung

Voraussetzung für den Einsatz von RTOS-UH auf PCs unter Windows ist ein funktionierender, hardwarenaher Atari-Emulator. Der Funktionsumfang von RTOS-UH unter WinSTon wurde mit keinem anderen uns bekannten Emulator erreicht. Unser Dank gilt daher Paul Bates, dem Autor von WinSTon, für seine unermüdliche Arbeit an diesem Projekt, für die gute Zusammenarbeit und vor allem dafür, daß er WinSTon als Freeware zur Verfügung stellt.

Prof. Dr.-Ing. W. Gerth und Dipl.-Ing. S. Bunzel
Universität Hannover
Institut für Regelungstechnik
Appelstraße 11
30167 Hannover

3 Bericht des Arbeitskreises 5

3.1 Plenarsitzung des ISO/IEC JTC1/SC22

Der Obmann des DIN NI-22, Herr Prof. Giere, hatte angeregt, die besonderen Eigenschaften von PEARL, z.B. die niedrige Komplexität, in Form eines Kurzbeitrages des NI-22 AK PEARL in den "German Activity Report" für die 13. Plenarsitzung des internationalen Normierungsgremiums ISO/IEC JTC1/SC22 aufzunehmen, die vom 12. – 15. September 2000 im japanischen Nara stattfand. Obwohl dies dann krankheitsbedingt leider doch nicht umgesetzt werden konnte, möchte ich meinen Vorschlag an dieser Stelle zum Zwecke der kritischen Diskussion und auch zukünftiger Aktivitäten mitteilen.

Report on the DIN NI-22 Project PEARL 90

The real time programming language PEARL 90 (DIN 66253-2) has become of outstanding interest for the support of certifiable safety related embedded systems. This is due to the inherent low complexity and the problem orientation of the real time constructs of this programming language, especially for high level exception handling and multitasking, preserving the real time software design architecture and, therefore, software understandability.

Owing to the intrinsic complexity of proving correctness of real time software with formal methods, there is a modern engineering trend towards proving software correctness "by inspection", which is, again, well supported by PEARL 90. More precisely, PEARL 90 is, for instance, a highly understandable basis for the development of safe and fault tolerant Programmable Logic Controllers (PLCs, cp. IEC 61131) for applications at highest safety integrity levels (IEC 61508) in a wide range of industries.

A corresponding PLC prototype is under development, called PEARL-PLC.

3.2 PEARL in der Ausbildung

Die Liste der Institutionen, die PEARL in der Ausbildung anbieten, konnte um zwei weitere Einrichtungen ergänzt werden. Es handelt sich dabei um die Berufsakademie Stuttgart, wo Herr Prof. Ghassemi-Tabrizi PEARL lehrt, und um die Hochschule für Technik und Wirtschaft Saarbrücken, in deren Fachbereich Maschinenbau Herr Prof. Dr.-Ing. Heinrich Warmers vor kurzem auf die Professur für Steuerungs- und Regelungstechnik berufen worden ist. Damit umfaßt die Liste der in PEARL ausbildenden Institutionen 21 Universitäten, 22 Fachhochschulen und 5 Berufsakademien, d.h. insgesamt 48 (uns bekannte) Einrichtungen im In- und Ausland.

In seiner Antwort auf meine entsprechende Recherche äußerte Herr Prof. Ghassemi-Tabrizi übrigens den Wunsch, zu seinem Buch, das bereits in der Ausgabe 1/2000 der PEARL-News von uns angekündigt wurde, auch eine inhaltliche Stellungnahme zu erhalten. Diesen Wunsch gebe ich hiermit gern an die Leserschaft der PEARL-News weiter.

An dieser Stelle möchte der Sprecher des AK 5 die Anregung geben, daß aktuelle und potentielle Buchautoren vielleicht neben den nunmehr üblich gewordenen "defensiven" Bemerkungen zum industriellen Einsatz von PEARL doch vielleicht auch einmal "offensive" Bemerkungen zum Ausbildungspotential von PEARL machen mögen.

3.3 Treffen des AK 5

Das nächste (zehnte) Treffen des AK 5 findet wieder – wie auch in der Einladung zum Workshop PEARL 2000 angekündigt – am Rande dieses Workshops am Donnerstag, dem 23. November 2000, um 11.00 Uhr in Boppard statt.

3.4 Mitglieder

Ideen und tatkräftige Mitarbeit jeder Art sind jederzeit sehr herzlich willkommen.

Prof. Dr.-Ing. G. Thiele
Universität Bremen, FB1
Kufsteiner Straße
28359 Bremen

4 GI-Seminar über Forschungs- und Entwicklungsperspektiven der Echtzeitinformatik

Echtzeitsysteme wurden bereits im Oktober 1972 in der Norm DIN 44300 unter der Nr. 161 wie folgt definiert:

Ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, daß die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorbestimmten Zeitpunkten anfallen.

Die erwähnte Zeitspanne ist natürlich nicht die Systemzeit eines Rechners, sondern ist durch Erfordernisse der (Anwendungs-) Umgebung des Rechensystems bestimmt. Daher sind Echtzeitsysteme immer als in eine größere Umgebung eingebettet zu betrachten und werden daher auch als *eingebettete Systeme* bezeichnet.

Diese schon 28 Jahre alte Sichtweise wird zwar partiell seit etlichen Jahren in verschiedenen Ansätzen angesprochen, aber verglichen mit dem immer noch sehr schwankenden Sprachgebrauch über eingebettete Systeme muß sie als geradezu rigoros gelten. (Dabei ist sie tatsächlich, wie alle Normen, in erster Linie aus Bedürfnissen der industriellen Anwendung erwachsen.)

Die Umgebungszeitabhängigkeit von Echtzeitprozessen wird aus der relevanten Anwendungspraxis heraus noch durch Anforderungen ergänzt, die sich aus

- Problemen der verteilten Kontrolle,
- der Vorhersehbarkeit des Verhaltens von Echtzeitprozessen und
- der Verlässlichkeit, insbesondere Fehlertoleranz,

ergeben. Die beiden letzten Punkte sind jedoch durch die Formulierung der Norm schon erfaßt. Das bedeutet aber in aller Regel, daß hier *Entwurfskonflikte* gelöst werden müssen, wie etwa zwischen möglichst guter Erfüllung der Echtzeitfähigkeit und Fehlertoleranz bzw. Verfügbarkeit. In vielen Fällen, und zwar gerade bei sicherheitskritischen Systemen (bei denen das Verpassen gewisser *harter* Zeitschranken katastrophale Konsequenzen hätte), ist sogar der Abgleich zwischen den *konfligierenden* Erfordernissen *dynamisch-adaptiv* zu gestalten, nämlich insofern, als die Umgebungssituation unvorhersehbar ist. Die Vorhersehbarkeitsforderung erzwingt, alle Systemfunktionen vollständig zu integrieren, und zwar *muß die geforderte Integration auch über alle Ebenen erfolgen*. Es ist auch leicht klar, daß die Modellierung von Systemfunktionen unter konfligierenden Erfordernissen nicht nur sehr komplex werden kann, sondern sogar immer *heuristisch* angegangen werden muß. Da ja schließlich auch Vorsehbarkeit garantiert werden soll, heißt das aber, daß *alle Lösungsmaßnahmen sehr einfach* gehalten werden müssen: sonst wäre ja die Vorhersehbarkeit praktisch nicht mehr evaluierbar.

Alle obigen Gesichtspunkte sind zwar inzwischen Zielvorstellungen für Forschungsprojekte, obwohl fast ausschließlich immer nur einige davon. Sie waren, und daran kann nicht oft genug erinnert werden, aber in ihrer Gesamtheit schon in der industriellen Entwicklung vor 30 Jahren zugrunde gelegt.

In der Vergangenheit gab es im deutschen Raum allerdings wertvolle vereinzelte Bemühungen wie etwa die Entwicklung von PEARL und, im Gefolge dieser Sprache, Betriebssystemprojekte mit für PEARL geeigneten Leistungsmerkmalen. Auch das Projekt PDV muß hier genannt werden.

Deshalb ist die Fachgruppe 4.4.1 der GI der Auffassung, daß es an der Zeit ist, auf allen Ebenen, von der Hardware über Betriebssysteme bis zu verschiedensten Anwendungen, die in der Vergangenheit erarbeiteten Lösungen zu dokumentieren und neue substantielle Arbeiten durchzuführen, die von der oben charakterisierten umfassenden, nur scheinbar neuen Sicht von Echtzeitsystemen ausgehen. Dies soll in Form eines GI-Seminars geschehen, dessen Ergebnisse immer in Buchform niedergelegt werden. Die Leser der PEARL-News werden hiermit um die Einsendung geeigneter Beiträge gebeten.

Prof. Dr. Horst F. Wedde
Universität Dortmund
Lehrstuhl Informatik III
44221 Dortmund
horst.wedde@udo.edu

5 Der PEARL-Kurs der FernUniversität

Über den Kurs *Realzeitprogrammiersprache PEARL* (Kursnummer 02417) der FernUniversität ist frohe Kunde zu vermelden: nach langwierigen Verhandlungen mit der Universitätsverwaltung konnte sein Preis deutlich gesenkt werden. Der Kurs ist multimedial und wird auf einer CD-ROM ausgeliefert. Zur Ausführung sind ein PC mit 8 MB Hauptspeicher, CD-ROM-Laufwerk, mindestens VGA-Graphikkarte und Sound-Karte sowie Windows erforderlich. Neben dem eigentlichen Lernprogramm umfaßt der Kurs eine komplette Programmentwicklungsumgebung mit integrierten vollständigen Dokumentationen. Da man Echtzeitprogrammierung nicht losgelöst von realer Hardware und zu automatisierenden Prozessen lernen kann, wird als Experimentierplattform ein Einplatinenrechner mitgeliefert, der mit dem PC über eine serielle Schnittstelle kommuniziert und an den Sensoren und Aktoren, und somit beliebige Prozesse angeschlossen werden können. Vom PC aus können Prozeßsteuerungsprogramme in den Einplatinenrechner geladen und dort unter Kontrolle des bewährten Echtzeitbetriebssystems RTOS-UH ausgeführt werden, das von Herrn Professor Gerth für diese Lernplattform freundlicherweise zur Verfügung gestellt wurde. In dieser Betriebsart übernimmt der PC die Funktion der Bedienkonsole.

Studierende und Gasthörer der FernUniversität können den PEARL-Kurs so wie jeden anderen Kurs belegen und so CD-ROM und Einplatinenrechner beziehen. Nicht an der FernUniversität eingeschriebenen PEARL-Interessierten werden CD-ROM und Einplatinenrechner zum Zwecke der persönlichen Weiterbildung zum Preise von DM 340 angeboten. Zum Bezug ist nur das umseitige Formular auszufüllen und einzusenden, das Sie auch als PDF-Datei von <http://www.fernuni-hagen.de/IT/lehre/pearl.htm> herunterladen können.

Herr Dipl.-Ing. Heinz Tatz hat die im Kurs enthaltene und auch zur Ansteuerung des Einplatinenrechners dienende Programmierumgebung deutlich verbessert und auf die Bedienung der DOS-Entwicklungsumgebung des Instituts für Regelungstechnik der Universität Hannover hin optimiert. Der verwendete Zeichensatz ist einstellbar. Es sind alle Grundfunktionen einfacher Editoren wie Drucken, Suchen, Ersetzen oder Kopieren verfügbar. In der Programmierumgebung lassen sich mehrere Fenster gleichzeitig öffnen bzw. erzeugen. Jedes im Arbeitsbereich geöffnete Fenster enthält einen Editierbereich für PEARL, Assembler- oder Linker-Quelltexte und darunter einen Anzeigebereich für eventuell aufgetretene Programmierfehler. Diese werden formatiert zusammen mit Angaben über ihren Typ und Ort aufgelistet. Ein Klick mit der Maus setzt den Cursor des Editierfensters auf die entsprechende fehlerhafte Textstelle. Eine kontextsensitive Hilfefunktion unterstützt den Anwender bei der Fehlersuche. So öffnet ein Mausklick auf ein PEARL-Schlüsselwort eine Hilfedatei an entsprechender Stelle. Zur Hervorhebung der Syntax werden die Schlüsselworte von Basic-PEARL und PEARL 90, die Mnemonics in Assembler-Quelltexten für die MC68000er Familie sowie Kommentare und Zeichenketten farblich abgesetzt dargestellt. Compiler, Assembler und Linker lassen sich über einen gemeinsamen Button starten. Häufig wiederkehrende Folgen von Übersetzerläufen können durch Stapelverarbeitung automatisiert werden. Zur Kommunikation mit einem an den PC angeschlossenen Einplatinenrechner dient ein integriertes VT100-kompatibles Terminal als Bedienkonsole. Hier stehen Funktionen zum direkten Laden von S-Records sowie zum Kopieren von Dateien vom und zum Einplatinenrechner zur Verfügung, die über Buttons aufgerufen werden. Eine kontextsensitive Hilfefunktion öffnet eine RTOS-Bedienbefehlsreferenz an zum zuletzt eingegebenen Bedienbefehl passender Stelle. Zur Wiederholung von Befehlssequenzen wurde eine konfigurierbare Makroverarbeitung bereitgestellt.

Für Fragen zu dieser neuen Programmierumgebung steht Herr Tatz unter Heinz.Tatz@fh-bochum.de zur Verfügung.