



PEARL-News

Ausgabe 1/2008

Mitteilungen
der GI-Fachgruppe Echtzeitsysteme

ISSN 1437-5966

Impressum

Herausgeber	GI-Fachgruppe Echtzeitsysteme (RT) URL: http://www.real-time.de
Sprecher	Dr. P. Holleczeck Universität Erlangen-Nürnberg, Regionales Rechenzentrum Martensstraße 1, D-91058 Erlangen Telefon: 09131/85-27817 Telefax: 09131/30 29 41 E-Mail: holleczeck@rrze.uni-erlangen.de
Stellvertreter	Prof. Dr. Dr. W. Halang FernUniversität in Hagen Universitätsstraße 27 - PRG D-58084 Hagen Telefon: 02331/987-372 Telefax: 02331/987-375 E-Mail: wolfgang.halang@fernuni-hagen.de
Redaktion	Prof. Dr. R. Müller FH Furtwangen, Fachbereich Computer- & Electrical Engineering Robert-Gerwig-Platz 1, 78120 Furtwangen Telefon: 07723/920-2416 Telefax: 07723/920-2610 E-Mail: mueller@hs-furtwangen.de
ISSN	1437-5966

Redaktionell abgeschlossen am 24. Juni 2008

Einreichung von Beiträgen

Diese Zeitschrift soll nicht nur Mitteilungsblatt sein, sondern auch eine Plattform für den Informations- und Meinungsaustausch zwischen allen an den Fragen der Echtzeitprogrammierung Interessierten bilden. Diskussionsstoff bzw. offene Fragen gibt es auf unserem Gebiet reichlich.

Wir möchten Sie, liebe Leserinnen und Leser, daher ausdrücklich ermuntern, auch in Zukunft die PEARL-News durch Ihre Beiträge mit zu gestalten. Für ein ausgewogenes Bild der News sollten Beiträge nicht länger als 5 Seiten sein.

Rainer Müller (Furtwangen)

Inhaltsverzeichnis

1 Neues aus der Fachgruppenleitung	3
2 Programm des Workshops PEARL 2008	3
3 Arbeitskreis: Echtzeit in der Ausbildung — PEARL-Sprachpflege	4
4 Buch „Echtzeit-Projekte“ in Arbeit	5
5 PEARL-User-Group / AK1 GI-FG REAL-TIME – Treffen am 31.05.2007 –	5
6 Vorstellung einer Plattform für die studentische Ausbildung im Echtzeit- und Feldbusbereich	9

1 Neues aus der Fachgruppenleitung

Am 6.12.2007 wurde eine neue Fachgruppenleitung gewählt. Die bisherige Fachgruppenleitung wurde einstimmig entlastet. Der neuen Fachgruppenleitung gehören an: Prof. Benra, Prof. Gerth, Prof. Halang, Prof. Heitmann, Dr. Holleczeck, Prof. Müller

Kurzinformationen von der Sitzung am 14.5.2008:

Teilnehmer: Prof. Benra (Wilhelmshaven), Prof. Gerth (Hannover), Prof. Halang (Hagen), Prof. Heitmann (Hamburg), Dr. Holleczeck (Erlangen), Prof. Müller (Furtwangen), Prof. Schiedermeier (Landshut), Prof. Wedde (als Gast aus Dortmund); Entschuldigt hatten sich: H. Arlt (esd Hannover), Dr. Windauer (werum Lüneburg), Prof. Zöbel (Koblenz)

Fachgruppenleitung und Fachexperten: Die Leitung der Fachgruppe liegt bei Herrn Dr. Holleczeck (Leitung) und Prof. Halang (Stellvertreter). Es wurden nur 5 von 7 möglichen Fachexperten benannt. Diese sind: H. Arlt, Dr. Sauter, Prof. Schiedermeier, Dr. Windauer, Prof. Zöbel

Organisation innerhalb der GI: Die Fachgruppe residiert nun als Fachausschuß (FA) *Echtzeitsysteme* im Fachbereich *Technische Informatik*. Da Fachausschüsse keine persönlichen Mitglieder haben, lautet die offizielle Bezeichnung Echtzeitsysteme (FA-RT), (FG-RT)

PEARL-Archiv: eine großer Bestand wurde zwischenzeitlich eingescannt; ein weiterer Bestand aus Zeiten des PEARL-Vereins wird Zug um Zug eingescannt.

neue AK-Leitungen: Sprachpflege: Prof. Gumzej (Maribor), Netze: Prof. Heitmann (Hamburg)

2 Programm des Workshops PEARL 2008

Der diesjährige Workshop steht unter dem Leitthema „Echtzeit 2008 - Aktuelle Anwendungen in Technik und Wirtschaft“. Er wird wieder im Hotel Ebertor in Boppard am Rhein stattfinden. Das Programmkomitee hat auf seiner Sitzung am 14. Mai 2008 in Frankfurt/Main das nachstehende Programm zusammengestellt. Der Tagungsband erscheint - wie in den vergangenen Jahren - in der Reihe „Informatik aktuell“ des Springer-Verlages.

Erster Workshop-Tag: Donnerstag, der 27. November 2008

11:00 Treffen der Arbeitskreise

13:00 Begrüßung

13:15 Sitzung 1: *Logistik und Verkehr* (Sitzungsleitung: Prof. Halang)

Echtzeiterkennung von befahrbaren Bereichen in urbanen Szenarien
(K. Berger, TU Braunschweig)

Berührungslose Winkelbestimmung zwischen Zugfahrzeug und Anhänger
(J. Schikora, Universität Koblenz-Landau)

From Embedded to Cyber-Physical Systems: A Conceptual Leap in Distributed Embedded Real-Time Systems
(S. Lehnhoff, TU Dortmund)

14:45 Pause

15:15 Sitzung 2: *Ausbildung* (Sitzungsleitung: Prof. Benra)

Eine Plattform für die studentische Ausbildung im Echtzeit- und Feldbusbereich
(K.H. Niemann, FH Hannover)

HighTecBot- ein Roboter-Baukastensystem zur Unterstützung der Informatik-Lehre an Hochschulen
(M. Lehnser, HTW Saarland)

16:15 Pause

16:45 Sitzung 3: *Kommunikation* (Sitzungsleitung: Prof. Heitmann)

Echtzeit-Publish/Subscribe-Kommunikation für mobile Roboter
(T. Lindhorst, Universität Magdeburg)

Latency Analysis and Improvement of IEEE 802.15.4 in Industrial Real-Time Applications
(F. Chen, Universität Erlangen)

18:00 Abendessen

19:30 Mitgliederversammlung der Fachgruppe

Zweiter Workshop-Tag: Freitag, der 28. November 2008

9:00 Sitzung 4: *Studentische Beiträge* (Sitzungsleitung: Prof. Schiedermeier)

Real-Time Virtualization Solutions for x86 Multi-Core Platforms
(A. Hollmann, Fachhochschule Landshut)

Fault Tolerance Analysis of the FlexRay Startup Procedure Using Model Checking
(S. Bunte, Universität Saarland)

Konzeption und Entwicklung eines echtzeitfähigen Lastgenerators für Multimedia-Verkehrsströme in IP-basierten Rechnernetzen
(A. Kolesnikow, Universität Hamburg)

10:30 Pause

11:00 Sitzung 5: *Forschung* (Sitzungsleitung: Prof. Wedde)

Dynamic Virtual Memory Management for Real-Time Systems
(T. Zitterell, Universität Freiburg)

WCET-Analyseverfahren in der automobilen Softwareentwicklung
(M. Däumler, Technische Universität Chemnitz)

Echtzeitrechnerarchitektur mit exakt vorhersehbarer Befehlsverarbeitungszeit
(H. Stieger, FernUniversität Hagen)

12:30 Verabschiedung

12:45 Imbiss

3 Arbeitskreis: Echtzeit in der Ausbildung — PEARL-Sprachpflege

DIN NI-22 Sitzung am 12.10.2007 in Stuttgart: Die regelmäßigen Workshops in Boppard und Hannover bestätigen die Lebendigkeit von PEARL. Die Präsenz von PEARL im industriellen Bereich mit ca 10000 neuen Lizenzen im Jahr 2007 für RTOS-UH hat die Notwendigkeit des Fortbestands der Norm dokumentiert.

Arbeitskreissitzung 2007: Der letzte Treffen fand am Rande des PEARL-Workshops am 6.12.2007 in Boppard statt.

- Für die Außendarstellung z.B. beim DIN ist es wichtig einen Überblick über die aktuelle Verbreitung und Einsatzzahlen von PEARL zu verfügen. Aktuelle Informationen auf den Industrieunternehmen und Hochschulen sind willkommen.
- Es wurde festgestellt, dass die Webseiten des Arbeitskreises dringend aktualisiert werden müssten. Die Liste der Projekte mit PEARL-Anteilen ist veraltet – in den letzten Jahren gab es keine Neueinträge. Dies dürfte auch die Liste der Institute betreffen, an denen PEARL in der Ausbildung eingesetzt wird. Der AK-Leiter wird diese Aktualisierungen nicht alleine erledigen können. Es wird um Unterstützung gebeten.

- Buchprojekt: Frau Prof. Benra organisiert die Entstehung eines Buches zur Entwicklung von Echtzeitsystemen. Mehr hierzu in Abschnitt 4 in dieser Ausgabe.
- Ausbildungshardware: An der FH Hannover wird eine Hardware auf Basis eines PPC entwickelt. Details zum Stand werden in Abschnitt 6 dieser Ausgabe dargelegt. Ein Fleyer mit den technischen Daten ist seit kurzem auf der Homepage des Arbeitskreises verfügbar.

DIN NI-22 Sitzung am 3. und 4.4.2008 in Dresden: Herr Prof. Halang hat uns bei der DIN-Sitzung vertreten. Es wurde eine Überarbeitung der PEARL-Norm angeregt.

Leitung: Leider überfordert die Leitung des Arbeitskreises und die Herausgabe der PEARL-News meine Zeitreserve. Freundlicherweise hat Herr Prof. Gumzej (Universität Maribor) die Leitung des Arbeitskreises übernommen. Ich freue mich eine auf gute Zusammenarbeit.

Rainer Müller
Hochschule Furtwangen

4 Buch „Echtzeit-Projekte“ in Arbeit

Eine ganze Reihe von Mitgliedern der Fachgruppe sind derzeit in einem Buchprojekt engagiert. Es soll ein Buch „Echtzeit-Projekte“ voraussichtlich im Sommer 2009 im Springer – Verlag erscheinen. Themen, die in dem Buch behandelt werden sollen, sind im Folgenden aufgelistet:

- Analyse und Entwurf von Echtzeitsysteme
- Synchronisation und Konsistenz in Echtzeitsystemen
- Vernetzte Echtzeit
- Echtzeitbetriebssysteme
- Programmierung
- Qualität und Sicherheit
- Worst Case Execution Time
- Software Architekturen und Entwurfsmuster

Momentan bearbeiten die einzelnen Autorenteam die Kapitel. Danach ist eine fachliche Begutachtung geplant, bevor das Buch im Frühjahr 2009 an den Verlag weitergeleitet wird. Zielgruppe sind sowohl Praktiker, als auch Wissenschaftler und interessierte Studierende, die mehr über den aktuellen Stand in der Entwicklung von Echtzeit-Projekten wissen möchten.

Juliane Benra
Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven

5 PEARL-User-Group / AK1 GI-FG REAL-TIME – Treffen am 31.05.2007 –

Anmerkung der Redaktion: Leider ist letztes Jahr dieser Bericht nur auf der Homepage des Arbeitskreises veröffentlicht worden. Die Darstellungen zum Exceptionhandling sind sicher auch jetzt noch von Bedeutung.

Das Treffen der PEARL-User-Group bzw. des Arbeitskreises *Embedded Systems, RTOS-UH/PEARL* der GI-Fachgruppe *Echtzeitsysteme und PEARL (REAL-TIME)* fand am Donnerstag, 31.05.2007, 14:00-17:00 Uhr, mit 8 Teilnehmern am [Institut für Regelungstechnik](#) der [Leibniz Universität Hannover](#) statt.

TOP 1: PEARL 90: Weiterentwicklungen

- Innerhalb der Fachgruppe zeichnet sich durch das Ausscheiden einiger aktiver Mitglieder eine Art Generationswechsel ab.
- Der Workshop *PEARL 2007* findet vom 6.-7.12.2007 in Boppard mit dem Thema *Mobilität und Echtzeit* statt. Wegen der hohen Resonanz mussten einige Beiträge abgewiesen werden.
- In 2006 wurden ca. 7.000 angemeldete Kopien von RTOS-UH ausgeliefert.

TOP 2: Compiler und Laufzeitsysteme: Statusbericht und Ausblick

Behobene Fehler

- Unter Umständen hat der Compiler die neuen CHAR(1000)-Variablen in dem 32-Bit statt in den 16-Bit Adressraum der Task abgelegt.
- Ein CONVERT in eine CHAR(1000)-Variable erfolgte modulo 256.
- Mit dem neuen Lader 6.6-D wurde ein Fehler bei der Behandlung von CRC-Fehlern der S-Records behoben.

Ausnahmebehandlung und Signale

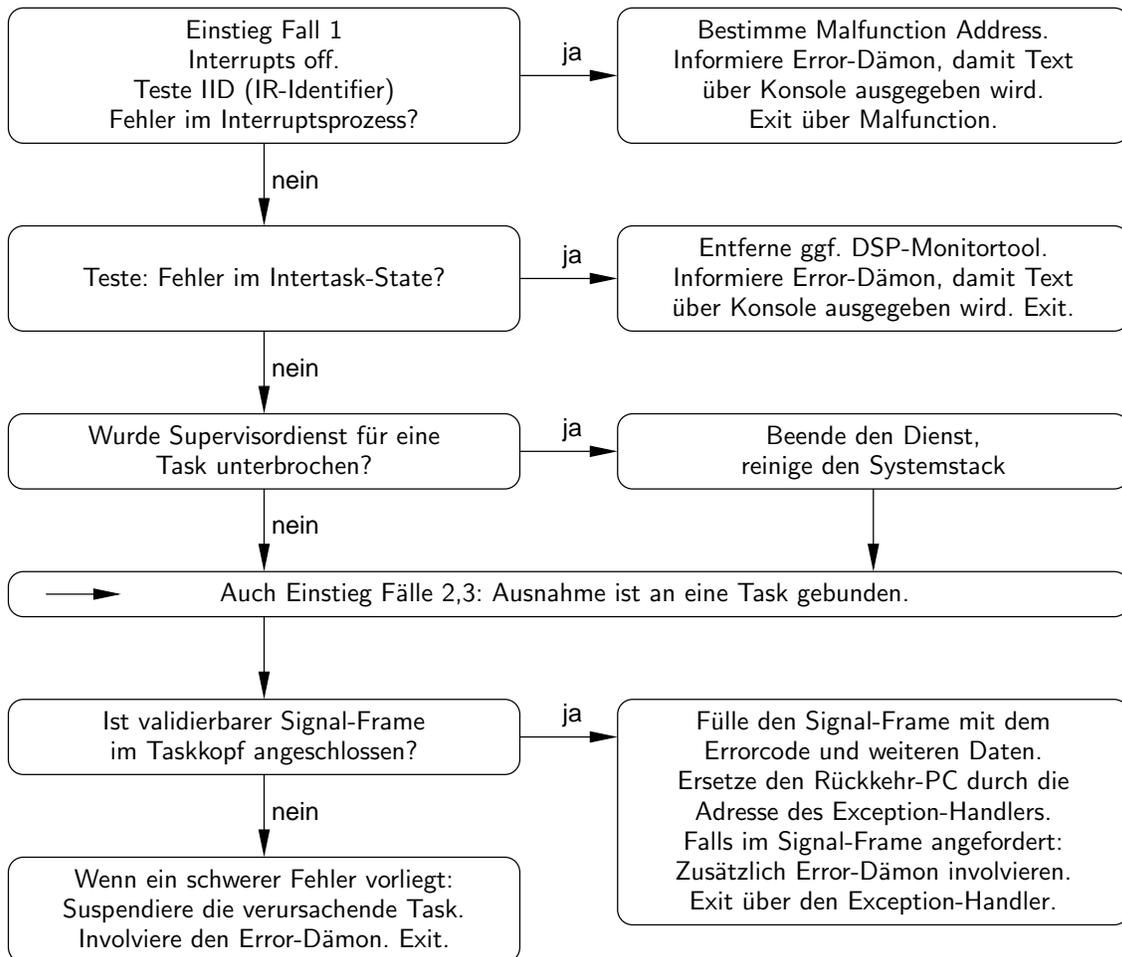
Vorgänge im Systemkern Zunächst soll die betriebssystemseitige Bearbeitung von Ausnahmesituationen betrachtet werden. Dabei wird von einem normalen System mit dem Standard Error-Dämon #ERRDM ausgegangen. Es sind 3 unterschiedliche Ausnahmesituationen zu unterscheiden:

- 1. Hardwareinduzierte Ausnahmen:** Die Rechnerhardware wurde mit einer nicht korrekt lösbaren Aufgabe betraut. Klassisches Beispiel: Es soll von einer nicht existenten Speicherzelle gelesen werden oder der gelesene Maschinenbefehl ist illegal kodiert. Aus der Sicht der gerade laufenden Software kommt das Ereignis unvorbereitet, wenngleich es an einen Maschinenbefehl gekoppelt ist - im Gegensatz zu von der Außenwelt getriebenen Interrupts. Der Prozessor wird bei dieser Ausnahmesituation wie bei einem Interrupt der höchsten Priorität nun zu einem speziellen Stück Code (im Supervisor mode) geführt, welches den individuellen Errorcode erstellt und darin vorgibt, ob der verursachende Prozess anzuhalten ist. Der Verursacher kann hier sowohl eine Task als auch ein Supervisorprozess (kernel mode) sein. Supervisorprozesse dürfen aber keinesfalls angehalten werden sondern müssen zu einem funktionserhaltenden Ende geführt werden.
- 2. Softwareinduzierte Ausnahmen:** Der Rechner funktioniert hardwareseitig korrekt, aber es ist eine (typischerweise datenabhängige) Fehlersituation aufgetreten. Klassisches Beispiel: die Software soll die Wurzel aus einer negativen Zahl bestimmen. Die Ausnahmesituation wird von der Software durch einen Betriebssystemaufruf (ERROR-Trap) eingeleitet. Durch die Kodierung des dem System mitgegebenen Error-Codes wird ein informeller Text erstellt. Auch hier wird dem System damit mitgeteilt, ob der laufende Prozess angehalten werden muss: Während man die Wurzel aus einer negativen Zahl ersatzweise mit Null beantworten kann (und die Task weiterlaufen kann), ist es nicht möglich, einer Prozedur übergebene (falsche) Zeiger (wrong parameterlist) zu reparieren. Im letzteren Fall muss die Task angehalten werden und sollte nicht entblockiert sondern beendet werden – eventuell durch einen Bedienerprozess oder manuell. Der Fehler kann stets einer Task angelastet werden.
- 3. Mischformen:** Es gibt in fast allen Prozessoren Maschinenbefehle, die abhängig vom inspizierten Datum eine Ausnahmesituation auslösen oder auch nicht. Klassisches Beispiel: Der Indextester des PEARL-Compilers überprüft im Test-Mode mithilfe des CHECK-Traps ob der errechnete lineare Feldindex eine Zelle innerhalb des Feldes adressiert. Liegt das Element außerhalb, feuert der Maschinenbefehl eine Ausnahme. (Nebenbei: Beim PEARL-Indextester repariert das Ausnahmebehandlungsprogramm den Fehler und adressiert das erste Element des Feldes - Fortsetzung möglich!). Der Fehler kann typischerweise immer einer Task angelastet werden, wird daher wie eine softwaregetriggerte Ausnahme behandelt.

Im Nukleus ist für alle drei Ausnahmesituationen eine Behandlungsroutine integriert. Diese kann man – für alle Fälle, in denen der Fehler einer Task zugeordnet werden kann – so parametrieren, dass die verursachende Task gezwungen wird, selbst auf den Fehler zu reagieren. Dabei kann das zentrale Fehlermeldesystem (#ERRDM = Error-Dämon) ausgeschaltet werden. So arbeitet z.B. die Shell mit einem eigenen Handler und verwendet das Fehlermeldesystem nicht.

Das Involvieren des Error-Dämonen erfolgt über einen Ringpuffer, der in der Shell des verantwortlichen Users angelegt ist. Bei den Ausnahmesituationen, die nicht einer Task zugeordnet werden können, erfolgt die Fehlermeldung auf der Konsole – der Anschluss eines eigenen Exception-Handlers ist für diese Situation nicht möglich und auch nicht sinnvoll. Aber: alle Interruptprozesse bringen zwangsweise (über den Malfuction-Exit) einen eigenen Handler mit. Dies ist eine markante Eigenschaft von RTOS-UH, die die Überlebenswahrscheinlichkeit erhöht.

Im folgenden Flussbild wird der Ablauf mit einer Ausnahme vom Typ 1 (Hardwarefehler) begonnen. Die Fälle 1,2 und 3 treffen sich an einem Sammelpunkt – immer dann, wenn eine Task als Verursacher feststeht.



Exception-Handler in PEARL Die Signalbehandlung ist über den Anschluss sogenannter ON-Blöcke möglich. Bei der Definition des Bearbeitungsmodos kann auf Wunsch der normale Error-Dämon neben dem eigenen Handler seine Meldung machen, allerdings wird die Task dann in keinem Fall mehr suspendiert.

Neben dem Bearbeitungsmodos muss eine Variable (die RST-Struktur) angegeben werden, in die das Betriebssystem im Fehlerfall Daten einschreiben kann. Dazu gehört die letzte überlaufene Zeilennummer, die letzte registrierte Modulnummer, der Errorcode, der Program Counter und ggf. der Induceparameter.

Es gibt zwei Stufen der Auslösung, die angewählt werden können.

Nur schwere Fehler: Der PEARL-Code im ON-Block wird nur angesprungen, wenn es sich um eine

Ausnahme mit gesetztem Suspend-Bit handelt. In diesem Fall kann die Task nicht an der Fehlerstelle fortgesetzt werden. Es ist aber möglich, per EXIT-Anweisung auf den Task-Grundlevel zurückzukehren oder Prozeduren abzubrechen. Über die Shell per INDUCE-Befehl gefeuerte Ausnahmen werden wie schwere Fehler behandelt, wenn der Induceparameter negativ ist. Auch wenn der Exception Handler bei einem leichten Fehler nicht angesprungen wird, so gibt es dennoch alle üblichen Einträge in die RST-Struktur. Nach Aufruf einer Prozedur kann man also abfragen, ob in ihr Fehler aufgetreten sind – auch wenn diese nicht zum Abbruch geführt haben.

Alle Fehler: Jede Auslösung wird wie ein schwerer Fehler behandelt.

Sinnvollerweise wird die RST-Struktur als statisch alloziertes Objekt definiert. Damit kann sie am einfachsten auch von Prozeduren aus erreicht werden.

```
#DEFINE X_ERRDM 1; ! Invoke #ERRDM
#DEFINE X_SEO 2; ! Invoke Exception handler Severe Errors Only
#DEFINE X_ALL 4; ! Catch all errors

MODULE EXCDemo; ! A simple demo prog
SYSTEM; A1;
PROBLEM;
SPC A1 DATION OUT ALPHIC;
DCL RS STRUCT(/Lino FIXED, ! Last registered line number
              Mono FIXED, ! Last registered module number
              Ecount FIXED, ! Exception counter (all)
              Errcode BIT(16), ! 16-Bit error code
              PC BIT(32), ! Program counter to exception
              Indpar FIXED, ! Induce parameter
              Buffer(100) FIXED ! Reserve for later extensions
              /);

/* +P */
TA:TASK;
DCL I FIXED;

ON E_(X_ALL+X_ERRDM) RST(RS); ! Catch all, invoke #ERRDM too
! Task will go here if exception handler is executed
PUT 'Exception fired at PC:', RS.PC TO A1 BY SKIP,A,B4(8);
SUSPEND;
! Do whatever may be necessary
EXIT -1; ! Step one procedure level down if not on task level
END;

!... Task may be here for the first time or after exception

IF RS.Ecount > 0 THEN
  PUT 'Restarted', RS.Ecount TO A1 BY SKIP,A,F(4);
ELSE
  PUT 'Initial Start' TO A1 BY SKIP,A;
FIN;

I=2//0; ! Diese Barriere wird nicht ueberwunden

END; ! TASK

MODEND;
```

TOP 3: Neue RTOS-UH Implementierungen

- Am IRT ist eine RTOS-UH Implementierung für die PPMC280/PPMC2800-Karte (MPC7447 PowerPC G4 mit 1 GHz, eine 128-Bit AltiVec-Einheit, 2 Ethernet-Ports und PCI) von Motorola entstanden. Das System beinhaltet einen Flash-Filemanager, der sich auch für den MPC555 eignet.
- Als Entwicklungstool kommt am IRT die freie IDE *Code::Blocks* zum Einsatz, die um eine Sprachunterstützung (Syntax-Highlighting und Code-Folding) für PEARL90 und den RTOS-UH Assembler ergänzt wurde. Für aufwändige mathematische Berechnungen kommen mit Hilfe des GNU-C-Compilers der Fa. *esd* für RTOS-UH lauffähig gemachte Bibliotheken aus dem Internet, wie z. B. die *GNU-Scientific-Library* zum Einsatz.

- Herr Seebode berichtete des Weiteren über die Entwicklung eines Laders für Objektdateien im Executable-and-Linking-Format (ELF) mit dem u.a. die Ausgaben neuerer GNU-C-Compiler direkt unter RTOS-UH geladen werden können. Inzwischen wurde eine erste Version eines solchen ELF-Laders fertig gestellt und erfolgreich zum Testen von AltiVec-Code des Compilers gcc 4.2 eingesetzt.
- Eine RTOS-Implementierung für den MPC5554 steht zur Verfügung, die abweichende SPE-Einheit für Vectorbefehle und speziellen Floatingpoint Operationen wird zur Zeit jedoch noch nicht unterstützt.

TOP 4: Berichte aus den Ingenieurbüros, Entwicklungsabteilungen und Forschungsinstituten

- In einer Diplomarbeit der FH Hannover soll in Kooperation mit der Fa. [IEP](#) ein preiswerter Einplatinencomputer auf Basis des 552-Prozessors aufgebaut werden. Dieser Prozessor hat im Vergleich zum 555 kein internes Flash und die resultierende Performance beträgt ca. 50 % des 555. Das Board ist mit einem geplanten Preis von max. 100 € insbesondere für den Einsatz in der Lehre sowie für Studierende und Heimanwender gedacht. Aus dem Hochschulbereich wurde bereits Interesse an diesem preiswerten Einplatinencomputer für RTOS-UH geäußert.
- Neue RTOS-Tools der Fa. IEP sind: NTP Client/Server, DNS Auflösung sowie E-Mail Versand über den IEP TCP/IP-Stack. Die mit Java realisierte grafische Ausgabe eines RTOS-Rechners auf einem PC ist jetzt auch über ein Java-Applet im Browser möglich. Der P-Monitor wurde um Prozesslaufzeiten erweitert.

Protokollführer

Dr.-Ing. T. Lilge
 Institut für Regelungstechnik
 Appelstr. 11
 30167 Hannover
lilge@irt.uni-hannover.de

6 Vorstellung einer Plattform für die studentische Ausbildung im Echtzeit- und Feldbusbereich

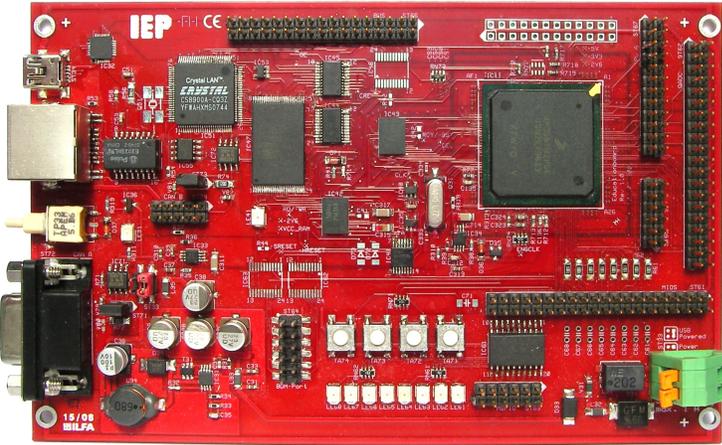
Mit dem Realtime Training Board (RTTB) steht eine kostengünstige und dennoch umfassend ausgerüstete Hard- und Softwareplattform für studentische Projekte im Bereich der Echtzeitprogrammierung bereit. Das RTTB entstand im Rahmen einer Diplomarbeit auf Initiative der IEP GmbH in Zusammenarbeit mit der Fachhochschule Hannover und ist ein sinnvolles Element des modernen und damit stärker praxis- und projektorientierten Hochschulstudiums. Die technische Ausstattung ermöglicht neben der Echtzeitausbildung auch den Einsatz in der Mikroprozessortechnik, der PEARL- und C-Programmierung, sowie beim Verständnis von Feldbussystemen. Außerdem ist die Programmierung in den typischen Sprachen der Speicherprogrammierbaren Steuerungen (SPS), welche in der IEC 61131-3 genormt sind, dank der Programmiersoftware CoDeSys möglich.

Bei der Hardwareauswahl stand neben dem Preis vor allem die Vielfalt der Nutzungsmöglichkeiten im Vordergrund. So bietet das RTTB mit dem Microcontroller MPC561 von Freescale einen mit 56 MHz getakteten PowerPC-Kern, der auch für anspruchsvolle Anwendungen genügend Rechenleistung bietet. Der 16 MByte große und mit 32 Bit angeschlossene Arbeitsspeicher dient sowohl als Ramdisk, als auch zur schnellen Ausführung des im RAM gespeicherten Betriebssystems und der zugehörigen Programme. Eine persistente Speicherung von System und Anwenderdaten wird durch das eingebaute 4 MByte Flash gewährleistet.

Die zahlreichen I/O-Möglichkeiten des RTTB laden zum Experimentieren ein. Mit den zwei Time Processing Units (TPU) stehen 32 vom Benutzer frei konfigurierbare Pins zur Verfügung. Zusätzlich sind 32 Analogeingänge, 8 pulswidenmodulierte (PWM) Ausgänge, diverse binäre I/O-Pins, ein SPI-Port mit

vier Chipselects und ein paralleler 16 Bit Bus für Erweiterungen vorgesehen. Für einfache Programmieraufgaben im Rahmen des Studiums sind außerdem vier interruptfähige Taster und acht LEDs auf dem RTTB vorhanden.

Dank seiner USB-Schnittstelle kann das Board ohne zusätzliche Hardware an jeden handelsüblichen PC angeschlossen werden, ein wichtiger Faktor beim Einsatz in Projekten, die oftmals außerhalb der Hochschule stattfinden. Zusätzliche Kommunikationsmöglichkeiten wie das LAN-Interface oder die zwei bis zu 1 MBit/s schnellen CAN-Schnittstellen bieten die Möglichkeit des Datenaustauschs zwischen mehreren Baugruppen oder im Netzwerk.



Das RTTB ist mit dem Echtzeitbetriebssystem RTOS-UH der Universität Hannover ausgerüstet, welches auf leicht erlernbare Weise Echtzeitfunktionen, wie zum Beispiel Multitasking und (zyklische) Einplanungen, unterstützt. Ein umfangreiches und bereits vorinstalliertes Softwarepaket sorgt für die Anbindung der diversen Schnittstellen des RTTB. Außerdem stehen Beispielprogramme mit Quelltext zur Verfügung, so dass die Einarbeitung und das Erstellen eigener Programme nur geringen Aufwand erfordert.

Mit Hilfe des Telnetserver ist eine problemlose Kommunikation über ein Netzwerk möglich und mit dem FTP-Server können von jedem eingebundenen Computer Dateien auf die Ramdisk übertragen werden. Um neben der reinen Technik die Praxis umfassend abzubilden, beispielsweise die graphische Darstellung von Messwerten, steht dem Studierenden der baugruppeninterne Webserver zur Verfügung.

Zur Zeit entsteht für das RTTB eine umfangreiche Dokumentation, die sowohl die Inbetriebnahme, als auch die Installation der Entwicklungssoftware beschreibt. Zusätzlich gibt es eine Aufgabensammlung mit verschiedenen Schwierigkeitsgraden, einschließlich der Musterlösungen. Dadurch kann sich der Benutzer auch ohne spezielle Einweisung mit dem RTTB vertraut machen und gelangt schnell an den Punkt die Möglichkeiten des Boards zum praxisnahen Lernen nutzen zu können.

Das Realtime Training Board (RTTB) ist also ein Baustein im Rahmen einer modernen, zukunftsfähigen Lehre und bietet vielfache Möglichkeiten die Echtzeitausbildung in Zukunft noch „echter“ zu gestalten.

Die Baugruppe wird vom Ingenieurbüro für Echtzeitprogrammierung (IEP) produziert und vertrieben. IEP fertigt auftragsgebunden, so dass das Lieferdatum und der Preis noch nicht endgültig feststehen. Ein RTTB wird zwischen 150 EUR und 180 EUR kosten und ab Ende 2008 lieferbar sein. Für Bestellungen und weitere Anfragen erreichen Sie die Firma IEP unter:

IEP GmbH
Am Pferdemarkt 9c
30853 Langenhagen

Tel.: 0511 70832 0
Fax.: 0511 70832 99
eMail: rttb@iep.de

Markus Hilmer
Fachhochschule Hannover, IEP GmbH
hilmer@iep.de