

# Entwicklungs-Kit für UML und Echtzeit-Linux



**Zukunftssicher entwickeln.** Steigende Komplexität der Entwicklungsprojekte und die Einführung neuer Betriebssysteme erfordern mehr und mehr den Einstieg in die 32-bit-Welt. Dies stellt für die Entwicklungsabteilungen einen Paradigmenwechsel dar und erfordert grundlegende Planungsarbeiten. Die Investitionskosten in die Software-Entwicklung und vor allem die Kosten der Software-Fortschreibung bestimmen maßgeblich die Gesamtkosten eines Projekts. Ein Entwicklungskit von Phyttec bietet einen möglichen Lösungsansatz, der ganzheitlich alle Phasen einer durchgängigen Projektplanung vom Hardware-Design bis zur Software-Pflege behandelt.

MARCUS LICKES

■ Mit Unterstützung der Partnerfirmen iLogix, ITK Engineering und Pengutronix wurde in Fortführung der Rapid-Development-Kits eine Zusammenstellung von Hard- und Software geschaffen, die ein mögliches Entwicklungsszenario vom UML-Einsatz über die Verwendung von Echtzeit-Linux bis zur direkten Unterstützung aller Hardware-Elemente des Entwicklungs-Boards für jeden leicht handhabbar und verständlich macht. Ziel ist es, Entwicklern innerhalb kürzester Zeit einen praxisnahen Einblick in ein Entwicklungsszenario unter dem Gedanken der Durchgängigkeit eines Lösungsweges zu vermitteln. Dieser Artikel beschäftigt sich mit der

Frage, warum sich Phyttec gerade für die gewählte Zusammenstellung von Komponenten entschieden hat, und bietet Hilfen zur eigenen Meinungsbildung.

## Die Hardware-Basis – und die Frage der Austauschbarkeit

Als Basis für das Kit wurde das phyCORE-PXA270CE-Modul mit dem 520 MHz schnellen Intel XScale-Prozessor gewählt. Das Modul bietet neben der Rechenleistung geringe Stromaufnahme, schnellen Speicher, Ethernet zur Einbindung in ein komplexeres Umfeld und eine Reihe von Prozess-E/A-Schnittstellen. Mit USB und Ethernet stehen Schnittstellen zur Verfügung, die ohne Betriebssystem nicht mehr angesteuert werden könnten. Neben dem eigentlichen Prozessormodul und der Basisplatine zur Anbindung an ein Host-System wird im Kit eine General-Purpose-E/A-Platine geliefert, die neben Tastern und LEDs auch mit einem kleinen Motor bestückt ist.

Das Board stellt exemplarisch eine Entwicklungsumgebung zur Verfügung, die die Anwendung ausreichend abstahiert,

um auch auf andere Architekturen portiert werden zu können. In der praktischen Anwendung wird die Wahl des Prozessors sicherlich von dessen technischen Eigenschaften und den verfügbaren Schnittstellen abhängen. Demnächst wird auch der PowerPC MPC5200 unterstützt. Eine Reihe von Schnittstellen, die das XScale-Modul bietet, finden sich auch auf dem PowerPC wieder und werden, trotz unterschiedlicher Architektur und Endianess, durch das Betriebssystem so weit abstrahiert, dass sie für den Anwender identisch angesprochen werden.

Die Verwendung eines Moduls – nicht nur zur Evaluierung, sondern auch in der Serie – ist ein wichtiger Schritt in der Planungsphase eines neuen Serienproduktes. Durch die Verwendung eines Moduls wird der komplexe Rechenkern von der Hardware für die Prozess-E/A getrennt. Das Hardware-Design ist schneller; die Software-Komponenten sind alle bereits für den Rechenkern (das Prozessormodul) angepasst. Der Weg zur Serie wird kürzer und preiswerter. Erst bei sehr großen Jahresstückzahlen ist ein Einplatinen-Design günstiger als die Modullösung. Da-

## KONTAKT

Phyttec Messtechnik GmbH,  
55129 Mainz,  
Tel. 0 61 31/92 21 -0,  
Fax 0 61 21/92 21 -33,  
[www.phyttec.de](http://www.phyttec.de)  
Halle 10, Stand 215

rüber hinaus ergeben sich weitere Vorteile des modularen Designs aus der Austauschbarkeit eines Moduls, zum Beispiel gegen ein leistungsstärkeres oder durch die Reduzierung des Pflegeaufwands bei Bauteilabkündigungen.

### Echtzeit-Linux – Standardisierung und Zertifizierung

Linux ist eine Alternative zu proprietären Betriebssystemen. Linux basiert vollständig auf bewährten Unix-Standards, deren Wurzeln bis in die 70er Jahre reichen und die Langzeitstabilität bereits bewiesen haben. Dank weltweit anerkannter API-Schnittstellen wie POSIX und der Verfügbarkeit des kompletten Source Codes lassen sich Software-Infrastrukturen für Embedded-Anwendungen schaffen, die

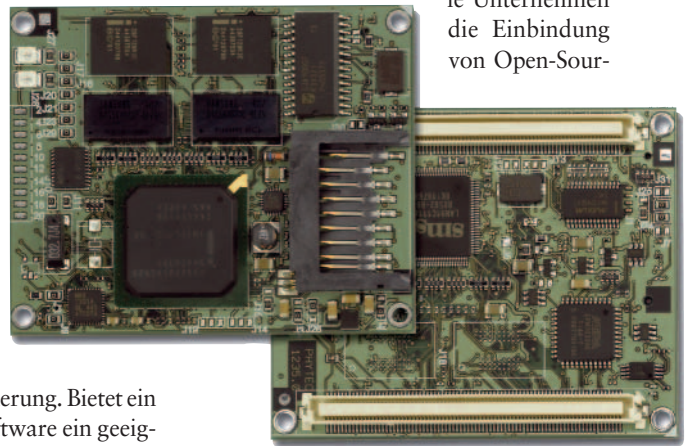


### TECHNISCHE DATEN

#### Modul mit Intel XScale PXA270CE / 520 MHz

- Größe: 57 mm x 71,5 mm mit zwei 160-poligen 0,635-mm-Molex-Verbindern
- pinkompatibel zu phyCORE-PXA255
- bis 64 MB Flash on board
- bis 256 MB SDRAM on board
- MMC/SD-Interface (Connector)
- Spannungsmanagement on board
- drei serielle Schnittstellen (auch nutzbar für Bluetooth und IrDa. FF-UART mit RS232-Treiber (Rest controllerintern))
- 10/100-Mbit-Ethernet
- Full-CAN-SJA1000-Interface
- USB-Host/Client (controllerintern)
- SPI-GPIO-Expander mit 28 I/O-Ports
- JTAG-Interface
- I<sup>2</sup>C-EEPROM
- zwei User-LEDs
- Temperaturbereich: - 20 °C...+ 85 °C

unabhängig von einer Hardware-Architektur sind. Mehr als je zuvor rückt damit das Know-how in den Mittelpunkt: Welche Open-Source-Werkzeuge sind heute einsetzbar, wo sind Potenziale, welche Software-Konzepte sind tragfähig? Der Open-Source-Gedanke ersetzt einen Hersteller und dessen Produktverantwortung durch die Leistungen einer weltweiten Entwickler-Community. Die Entscheidung für einen dieser Wege ist in erster Linie weniger eine Frage von ›richtig‹ oder ›falsch‹ als vielmehr eine Frage der eigenen Positionierung. Bietet ein Hersteller proprietärer Software ein geeig-



nen Fortentwicklung einen Entwicklungsaufwand, den ein einzelnes Unternehmen nur schwer stemmen kann. Typisches Beispiel hierfür ist die Entwicklung im Bereich der Compiler. Hier sehen schon heute viele Unternehmen die Einbindung von Open-Sour-

ce-basierten Compilern in eigene, proprietäre Entwicklungsumgebungen als Grundlage ihres Geschäftsmodells an. Auch bei den Entwicklungsumgebungen bündeln sich die Aktivitäten der professionellen Hersteller bei Open-Source-Plattformlösungen wie Eclipse. Der monetäre Aspekt spielt dabei oftmals eine untergeordnete Rolle. Die Zahlung von Lizenzgebühren für proprietäre Software wird ersetzt durch Aufwendungen für die Produktpflege von Open-Source-Lösungen. Ob diese Pflege im eigenen Unternehmen oder zusammen mit Drittanbietern (die im Gegensatz zum Hersteller proprietärer einfacher Lösungen austauschbar sind) gewährleistet wird, ist eine Frage der eigenen Positionierung. Was sich sicherlich ändert, ist der Umgang mit der Produktpflege. Diese Änderung ist gravierender, als es zunächst aussieht, und sollte bei der strategischen Unternehmensplanung wohl bedacht werden. Wichtig bei der individuellen Beantwortung dieser Frage ist allerdings die Langzeitbetrachtung der Gesamtkosten. Die Verbindung aus Echtzeit und Linux wird von verschiedenen Entwicklergruppen der Open-Source-Gemeinde mit unterschiedlichen Ansätzen vorangetrieben. Ein Aspekt der Open-Source-Philosophie ist der offene technologische Wettstreit konkurrierender Ideen und Lösungen, der einerseits Motor für schnelle Weiterentwicklung ist, andererseits aber auch zu divergenten Lösungen und zur Aufteilung von Energie führt. Dieser Charakter der Open Source kann nicht durch irgendeine Unternehmensentscheidung beeinflusst werden. Im Echtzeit-Linux-Bereich hat dies zur Ausbildung unterschiedlicher >

## INFO

**Entwicklungsumgebung**

Als Entwicklungs-Host dient ein PC unter Linux. Phytec schreibt den Einsatz der SUSE Distribution (v10) vor, um einen reibungslosen Ablauf der Quickstarts zu gewährleisten. Die Wahl von Linux als Entwicklungs-Host entspricht der konsequenten Fortschreibung des Linux-Gedankens des Kits und sorgt für eine insgesamt homogene Umgebung. Rhapsody ist sowohl mit Linux als auch mit Windows als Host sehr flexibel einsetzbar. Daher ist die Durchgängigkeit auch bis zur UML-Ebene garantiert.

**Quellen**

Übersicht über die Model Driven Architecture der OMG:

[www.omg.org/mda/](http://www.omg.org/mda/)

Rapid Development Kit von Phytec;

[www.phytec.de](http://www.phytec.de)

Entwicklungsunterstützung für UML und Embedded/Realtime-Software von ITK Engineering;

[www.itk-engineering.de](http://www.itk-engineering.de)

Embedded-Linux-Schulung, Support & Consulting von Pengutronix;

[www.pengutronix.de](http://www.pengutronix.de)

- Lösungsansätze geführt. Die wichtigsten Vertreter sind die Dual-Kernel-Ansätze (RTAI, RT-Linux) und die Varianten, die den Linux-Kernel direkt echtzeitfähig machen.

Bislang waren alle Entwicklungen mehr oder weniger entkoppelt von der Weiterentwicklung des Linux-Kerns. Ziel für industrielle Anwender muss es sein, eine funktionale Kompatibilität von Lösungen und vor allem die Annäherung der Lösungsansätze an den Mainstream der Kernentwicklung zu schaffen. Technologisch bieten heutige Echtzeit-Linux-Ansätze wie Preempt-RT hervorragende Grundlagen zur Eingliederung in die Kernel-Entwicklung. Auch hier stellt sich wiederum die Frage der kritischen Masse, allerdings stehen die Chancen derzeit so gut wie nie, dass der Linux-Kernel in absehbarer Zeit hart echtzeitfähig wird.

Echtzeit ist ohnehin mehr eine Frage des Determinismus als der absoluten Zeitanlagen.

Eine Zertifizierung von Lösungen gegen vorher definierte Standards, die allgemein anerkannt werden, stellt einen Meilenstein in der Akzeptanz von Echtzeit-Linux dar. Diese Zertifizierung und die vorausgehende Einigung auf Standards haben sich eine Reihe von Automatisierungsanbietern und -anwendern auf ihre Fahnen geschrieben und haben beschlossen, ihre Interessen auf Förderung des Open-Source-Gedankens in der Industrie zu bündeln. Phytec ist eines der Unternehmen der ersten Stunde dieser Bewegung. Das Linux im neuesten Entwicklungskit von Phytec wurde von Pengutronix implementiert, ebenfalls einem Gründungsmitglied dieser Bewegung. Das Unternehmen sorgt mit dem Dienstleistungsprodukt OSELAS (Open Source Embedded Linux Automation Suite) für professionellen Support, Schulung und projektbegleitendes Consulting für industrielle Linux-Projekte. Dabei wird besonderer Wert darauf gelegt, dass die eingesetzten Verfahren der ›best practise‹ aus IT- und Industriestandards entsprechen und keine Sonderwege beschränkt werden.

Linux wird sich entlang der gemeinsam im Kreise vieler Anbieter und Entwickler zu erarbeitenden Standards selbst weiterentwickeln. Technologisch ist der jetzige Stand bereits so tragfähig, dass die erfolgreiche Steuerung komplexer Maschinen die Praxistauglichkeit der Konzepte und Lösungen bereits bewiesen hat.

**UML – mehr als automatische Code-Generierung**

Das Meistern der Komplexität heutiger Softwareentwicklungen erfordert einen wohl strukturierten Lösungsansatz. Eine Aufgabenstellung muss in einzelne Teilgebiete zerlegt werden, Schnittstellen untereinander und zum Betriebssystem beziehungsweise zu den E/A müssen exakt spezifiziert werden, und die gesamte Projektentwicklung muss in jeder Pha-

se so dokumentiert sein, dass die Projektpflege auch bei sich verändernder Entwicklerteam gewährt ist. Die Abstraktion der Aufgabenstellung von der tatsächlichen Implementierung auf einer speziellen Hardware schafft die Grundlage für eine einfache Portierung auf sich fortentwickelnde Hardware-Umgebungen von morgen. Je höher die Abstraktionsebene, desto einfacher verständlich und wartbar wird eine Anwendung. Durch die Wahl der Entwicklungsumgebung wird

der zur Verfügung stehende ›Werkzeugkasten‹ festgelegt. Im Phytec-Kit bietet eines der marktführenden UML-Werkzeuge (Rhapsody von I-Logix in einer zeitbegrenzten Testversion) einen Einblick in modernste Entwicklungsmethoden – von Analyse und Design bis hin zur automatischen Code-Generierung aus grafikgestützten Modellen.

Die Vorteile von UML beschränken sich allerdings nicht nur auf diesen Aspekt. Der ganzheitliche Ansatz insbesondere in Bezug auf projektbegleitende Dokumentation schafft Transparenz und sichert die effiziente Fortschreibung komplexer Anwendungen. Auch die Fehlersuche wird auf eine höhere Abstraktionsebene gehoben: Neben Code-Level-Debugging ist nun

**FAZIT****Zukunftssicher durch Module**

Das vorgestellte Kit erleichtert den Einstieg in neue Technologien und die effiziente Entwicklung von Serienprodukten. Die Modularisierung bereits auf Hardware-Ebene durch den Einsatz eines Prozessormoduls bildet die Grundlage für die Verwendung fertiger Elemente bis zur Betriebssystemebene. Das Kit bietet mit Linux für Echtzeitanwendungen einen wichtigen Mehrwert, der gleichzeitig eine konkrete Antwort auf den Einsatz von Open-Source-Software bietet. Mit der Durchgängigkeit bis hin zu UML ist eine weit in die Zukunft reichende Perspektive aufgezeichnet, die die Diskussion um strategische Lösungsszenarien auf eine praktisch einsetzbare Basis stellt. Das geordnete Zusammenspiel aller Komponenten und die direkte Einsetzbarkeit in Projekte sind die Stärken des Kits. Phytec bietet neben einer Inbetriebnahmegarantie für das Kit Schulungen und Projektunterstützung gemeinsam mit den beteiligten Partnern an.

auch Design-Level-Debugging auf Modellebene möglich. Die Möglichkeit zur Einbindung eigener Code-Anteile ist ein wichtiges Element in der praktischen Realisierung von Aufgabenstellungen. Eine durchgängige Kette der Entwicklungswerkzeuge ermöglicht das frühzeitige Testen des Modells auf der Ziel-Hardware; das UML-Werkzeug generiert auf Knopfdruck Code, der auf dem Zielsystem mit Echtzeit-Linux sofort lauffähig ist.

Mit ITK Engineering hat Phytec einen Partner an den Entwicklungsarbeiten beteiligt, der einen großen Erfahrungsschatz in der Umsetzung komplexer Projekte – mit und ohne UML – vorweisen kann. Diese Praxiserfahrung ist in die Definition der Schnittstellen, die Anpassung des UML-Tools an das Echtzeitsystem und vor allem in die Realisierung von praxisnahen Beispielen eingeflossen, die als Be-

standteil des Kits im Quellcode mitgeliefert werden. Eine Besonderheit des Kits besteht in der direkten Anbindung einer E/A-Platine (Bestandteil des Kits), deren Hardware von diesen Beispielen angesprochen wird. Damit sind die Beispiele sehr nahe an der Realisierungspraxis von industriellen Lösungen, die auf der Basis von Phytec-Komponenten geschaffen werden können. ■

**Autor**

Dipl.-Ing. (FH) MARCUS LICKES ist Entwicklungsleiter Hardware bei Phytec Messtechnik in Mainz.



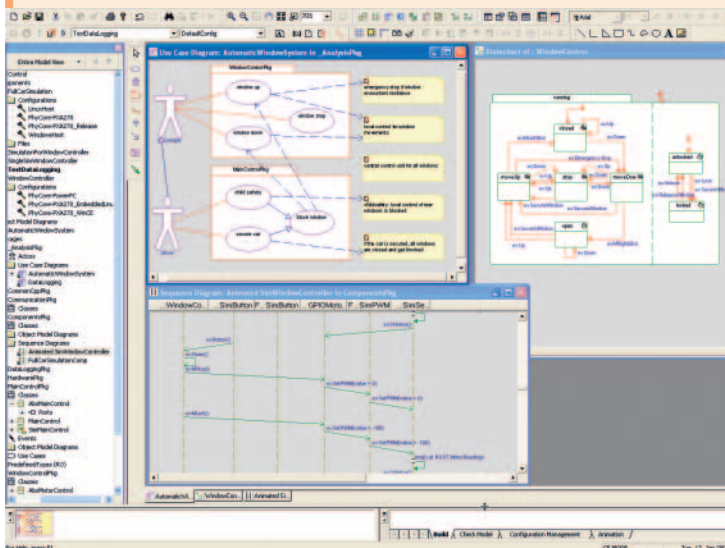
**EINHEITSSPRACHE UML**

**Abstrakte Modelle**

UML, die Unified Modelling Language, ist Teil der Meta Object Facility (MOF) der Standardisierungsorganisation OMG. Im ISO/IEC-Standard 19502:2005 ist das Konzept der sich selbst beschreibenden Metamodelle festgelegt. Der Anwender hat mit UML die Möglichkeit, auf standardisierte Weise Modelle zu definieren, die das Verhalten seines Systems auf einer formal-inhaltlichen Ebene beschreiben. In der Mess- und Automatisierungstechnik können so zum Beispiel Zu-

standsautomaten oder auch die Protokollflüsse von industriellen Feldbus-Protokollen modelliert werden. Die Implementierung der Modelle, die im XML-Dialekt XMI (XML Metadata Interchange) gespeichert werden, erfolgt entweder wie bei Rhapsody durch direkt generierten C++-Code für die Zielplattform oder durch Transformation auf einfache Applikationsmodelle mittels aus der IT bekannter Technologien wie XSLT; aus diesen können wiederum

durch überschaubare Code-Engines Runtime-Umgebungen generiert werden. Ebenfalls Teil der MOF ist die Verteilung von Objekten über Systemgrenzen hinweg durch den Kommunikationsstandard CORBA.



**UML-Diagramme definieren das Systemverhalten**

**Online sucht sich schneller!**

Das Archiv von **Mechatronik F&M** finden Sie bequem unter **www.mechatronik.info**. Alle Artikel stehen Ihnen als PDF-Dateien zur Verfügung.

**Archiv-Navigation**

- [Aktuelle Ausgabe](#)
- [Jahrgänge](#)
- [Suche](#)
- [Kostenlose Artikel](#)

**Ihre Vorteile:**

- Alle Artikel voll durchsuchbar
- Im Handumdrehen als PDF auf Ihrem Rechner
- Für Abonnenten kostenlos

ARCHIV Mechatronik F&M

Bitte geben Sie Ihren Suchbegriff ein

Suchbegriff(e) (inkl. Autoren)

alle Wörter (AND)  mindestens ein Wort (OR)

Suche:

über Alles

nur in Artikelüberschriften

nur in Zusammenfassungen (Abstracts)

Zeitraum: von  bis

**Schlagwort**

Schlagwort:

www.mechatronik.info