

HyNetOS 3.0 markiert wichtigen Meilenstein im Embedded-Communication-Bereich

Erstes Instanz-orientiertes Embedded-Betriebssystem speziell für echte System-on-a-Single-Chip-Lösungen

Jülich, Januar 2004 – Smart Network Devices (SND), einem der technologisch weltweit führenden Unternehmen im Bereich zukunftsorientierter Embedded Internet-Technologien, ist es jetzt in enger Zusammenarbeit mit dem Prozessorhersteller Hyperstone gelungen, einen weiteren wichtigen Meilenstein auf dem Weg hin zu echten System-on-Single-Chip-Lösungen zu setzen. Das von SND erstmals auf der Fachmesse Embedded World 2004 in Nürnberg (Halle 12/Stand 232) präsentierte asynchron arbeitende Betriebssystem HyNetOS 3.0 läuft komplett auf dem internen 128-KB-SRAM des neuen Hyperstone 32-Bit-RISC/DSP-Kommunikationsprozessors hyNet32XS ab und kann trotz des extrem geringen Speicherbedarfs selbst komplexe Applikationen wie z.B. eine Netzwerk-Kamera inkl. TCP/IP, HTTP-Server, Bilderfassung und Echtzeit-Bildkompression ohne jegliche Einschränkungen ausführen.

HyNetOS zählte aufgrund des asynchron-modularen Aufbau schon bisher zu den effizientesten Embedded-Betriebssystemen für komplexe Netzwerk- und Kommunikationsapplikationen. Dass der gegenüber Unix- oder Windows-basierenden Embedded-Betriebssystemen vergleichsweise geringe Speicherbedarf jetzt nochmals deutlich reduziert werden konnte, ist vor allem auf die Umstellung des Nachrichten-basierten Systems von einer reinen Task- hin zu einer Instanz-Orientierung zurückzuführen. Hierdurch entfallen bislang notwendige Prozesse, wie zum Beispiel der Message-Dispatcher innerhalb der Tasks.

Alle Nachrichten werden nun direkt in den entsprechenden Empfänger-Instanzen verarbeitet. Damit wird erreicht, dass in Tasks, in denen mehrere Verbindungen (Clients, Instanzen) gleichzeitig verwaltet werden, nach dem Eintreffen einer Nachricht nicht erst die Referenz zu dieser Nachricht in der zentralen Verbindungstabelle aufgesucht werden muss, sondern vielmehr die Nachricht selbst die Referenz bereits enthält. Dies spart Rechenleistung und erhöht die Effizienz der Nachrichtenverarbeitung deutlich.



Eine weitere wesentliche Neuerung sind die überarbeiteten Zustandsautomaten des HyNetOS 3.0. Sie sind Garanten für kompakte und gleichzeitig überaus stabile Funktionsblöcke, da keine Codewiederholungen auftreten und alle Events in allen möglichen Zuständen zwangsweise behandelt werden.

Services werden bei HyNetOS 3.0 zudem nicht mehr mit einer fixen Priorität starr in den Tasks abgewickelt. Stattdessen werden Instanzen je nach Anforderung in beliebigen Tasks kreiert, wodurch sich ein dynamisches Verhalten bei unterschiedlich priorisierten Nachrichten im System ergibt. Das führt zu schnelleren Reaktionszeiten bei wichtigen Ereignissen, da diese nun bevorzugt behandelt werden können. Nur so ist es möglich, auch Antwortzeiten auf wichtige Ereignisse präzise vorherzusagen. Da jede Nachricht auch immer ein Stück Speicher alloziert, muss zudem gewährleistet sein, dass die Speicherzuweisung in klar definierten Zeiten erfolgt. Ein optimierter Memory-Manager sorgt hier für deterministische Abläufe auch in der Speicherverwaltung.

Auch hinsichtlich der im Hardware Abstraction Layer (HAL) befindlichen Device Driver präsentiert sich das Embedded-Betriebssystem HyNetOS in der Version 3.0 noch flexibler als die Vorgängerversionen. Die Device Driver sind jetzt ebenfalls als Instanzen in Tasks eingebunden. Zudem wurde der Real Time Kernel von allen Aufgaben der Zeit- und Prioritätsverwaltung befreit. Somit können mehrere Schnittstellen mit unterschiedlichen Bandbreiten stets in Echtzeit bedient werden.

Bei Smart Network Devices ist man überzeugt, dass HyNetOS3.0 in vielen kostensensiblen Bereichen, zum Beispiel der Vernetzung oder Überwachung von Fabriken und Gebäuden, den endgültigen Durchbruch bringen könnte. Die meisten Anwendungen scheitern heute nicht mehr an der technischen Realisierbarkeit, sondern schlichtweg an den Kosten. Wer Millionen von Lichtschaltern, Haushaltsgeräten, Sensoren und Aktoren vernetzen will, wird an preisgünstigen System-on-Chip-Lösungen also nicht vorbei kommen.



„In Verbindung mit dem neuen Hyperstone 32-Bit-RISC/DSP-Kommunikationsprozessor hyNet32XS ist es jetzt möglich, typische Applikationen aus der Kommunikationswelt wie zum Beispiel eine vollständige Netzwerk-Kamera mit HTTP-Server inklusive TCP/IP Stack, Bilderfassung, Echtzeit-Bildkompression, Ethernet-Treibern, Real Time Kernel und weiteren Schnittstellen-Treibern erstmals auf einem einzigen Chip ablaufen zu lassen. Wir sprechen hier von echtem „System-on-a-Single-Chip“. Durch diese Optimierung werden sich die Kosten eines leistungsfähigen Netzwerkknotens erstmals auf eine für die Anwender wirtschaftlich vertretbare Größenordnung reduzieren“, unterstreicht Peter Duchemin, Geschäftsführer Technik der Smart Network Devices.

Smart Network Devices GmbH

Die 1999 in Neuss gegründete und heute in Jülich ansässige Smart Network Devices GmbH (SND) hat eine eigene innovative Technologie entwickelt, die es ermöglicht, die komplexen Kommunikationsaufgaben in drahtgebundenen und drahtlosen Netzen der Zukunft auf einem einzigen Chip zu bewältigen (System on Chip). Mit innovativen Produkten wie dem Embedded-Netzwerk-Betriebssystem HyNetOS™, dem Micro WebTarget™-Board, dem Bluetooth-Board Micro BlueTarget™ und dem Bluetooth-Protokoll-Stack für HyNetOS™ avancierte Smart Network Devices innerhalb kürzester Zeit zu einem der in diesem Bereich technologisch weltweit führenden Unternehmen. Die Lösungen von SND ermöglichen heute leistungsfähiges und plattform-unabhängiges „Embedded Networking“ in Anwendungsbereichen wie Industrial Automation, Facility Management, Bluetooth Infrastruktur usw.

Ansprechpartner für die Presse:

Peter Duchemin
Smart Network Devices GmbH
Technologiezentrum
Karl-Heinz-Beckurts-Str. 13
D-52428 Jülich
Tel. +49 (24 61) 690 620
Fax. +49 (24 61) 690 620
E-mail info@smartnd.com

Werner W. Wiesmeier
3W Media & Marketing Consulting
Preisingerlohweg 2
D-85368 Moosburg
Tel. +49 (87 61) 75 92 03
Fax. +49 (87 61) 75 92 01
E-mail werner.wiesmeier@online.de