



From Deep Space to Deep Sea

Push the button high reliability

Altreonic Press Release

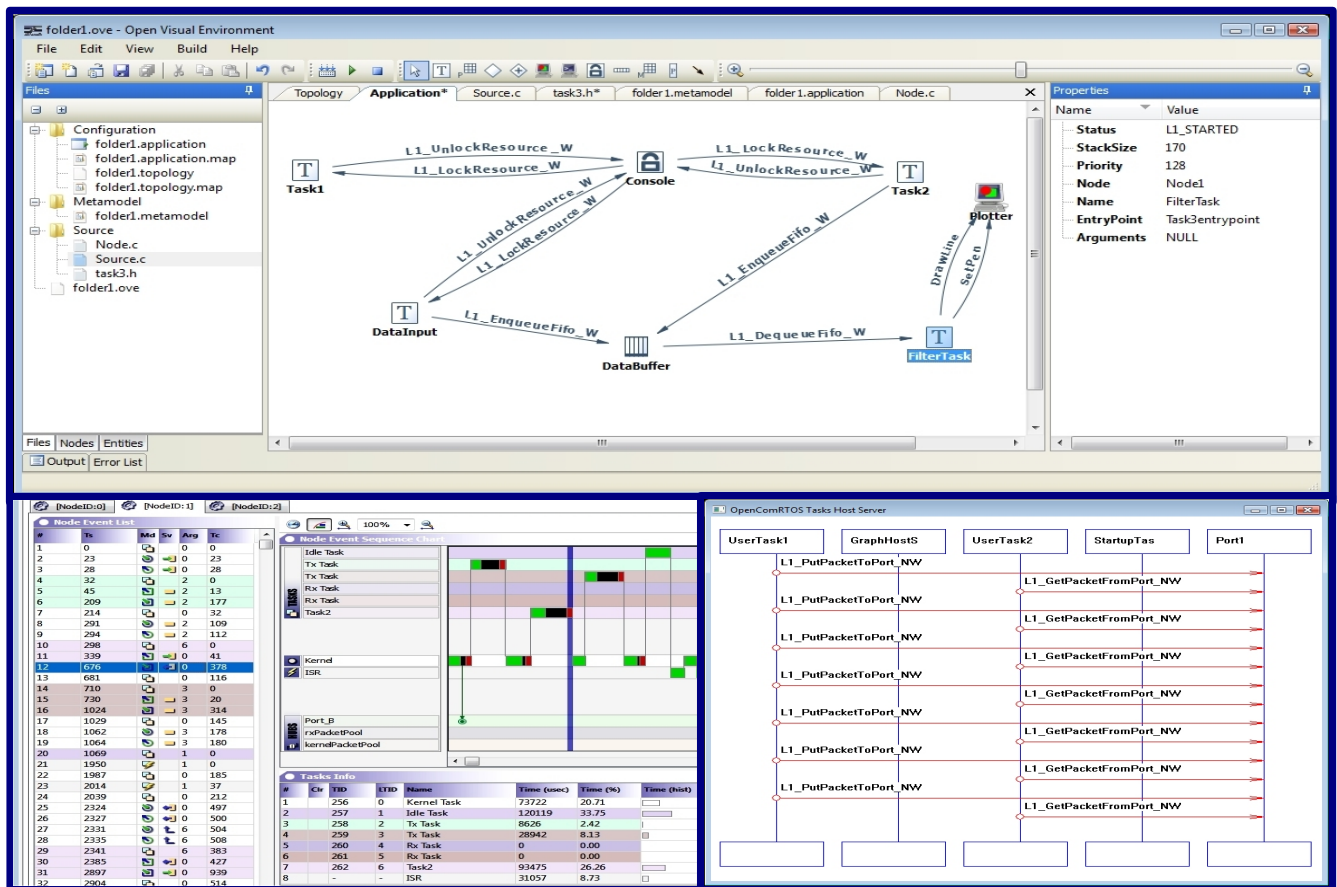
26th February 2009

For immediate release

Altreonic has announced the release of **OpenComRTOS with support for transparent programming of heterogeneous target systems**. The product will be demonstrated at Embedded World in Nuernberg next week. **Nominated for the Embedded Award at Embedded World**, OpenComRTOS is the result of a four year effort to develop a **network-centric** RTOS (Real-Time Operating System).

The result is a **break-through architecture**, basically reinventing the RTOS core concept and what concurrent programming is all about. Programming is essentially modeling the real-world on a processor. As the real world is concurrent, it is natural to express the program as a set of interacting entities. In OpenComRTOS active entities are called tasks and the interaction entities are called hubs. While the tasks are essentially programmable functions with their own priority and workspace, hubs come in different variants. Traditionally, these variants are called Events, Semaphores, Queues, Resources, etc. Furthermore, a hub was developed as a **Guarded Atomic Action** (a fundamental concept in formal techniques), therefore it has generic properties. Users can create their own RTOS services where the resulting code size is also much smaller, typical codesizes range from 1 Kbyte for an optimised single processor version to about 5 to 10 KBytes for a distributed version supporting all features. The result is not only a very scalable and modular architecture, but the small code size also increases the performance and lowers the power consumption while the effort of verification is reduced.

Another innovation in OpenComRTOS is the use of **Packet Switching** techniques at all levels. Actually, OpenComRTOS was first developed as a distributed communication layer, adding a scheduler allowed it to operate like a distributed RTOS. Using a similar mechanism for the internal working of the kernel provides safety properties like the provable absence of buffer overflow failures and auto-throttling. Nevertheless, RTOS developers are familiar with hub entities like events and semaphores and hence using OpenComRTOS is natural for embedded developers. This is facilitated by **OpenVE, Altreonic's Open Visual Development Environment**. The developer designs the nodes topology of the target processor system independently from his or her application. While doing this, the designer maps tasks and hub entities to the different nodes. It is possible to generate a simulation model that runs natively on his host system (typically a PC running Windows or Linux). The next step is then to remap the nodes to the network nodes, recompile and run the application on the distributed, heterogeneous target with no changes to the source code. The target can include host nodes which grant memory constrained target nodes access to host to file systems, graphic displays and other host services. A node can even be placed somewhere on the internet, e.g. on a network server. At the other extreme, a node can also be part of a multi-core CPU. Because **OpenComRTOS works distributed and independently of the communication mechanism** used (shared memory, TCP/IP, or direct point-to-point communication mechanisms like FIFOs). The real-time performance is reflected in the communication by using **system wide priorities** and



improves when low latency communication hardware is used with a bandwidth matching the processor performance.

To further support the developer, OpenVE is bundled with **OpenTracer**, a graphical debug tool. It displays task scheduling and hub usage in an oscilloscope-type graph which is augmented with statistical profiling data. As a novelty, the tool also combines the different traces from the different nodes to show the node interaction.

Finally, Altreonic also innovated the licensing model. While the software is made available under a traditional Binary License, Altreonic makes it also available under a so-called **“Open” License**. Under such a license, typically aimed at customers who need full control, such an Open License provides source code as well all design documents, formal models, test suites, etc. that document the software. An Open License also carries the right to create binary licenses of the tools. OpenComRTOS itself is royalty free and is complemented by a RTOS porting kit and a RTOS extension kit.

Supported targets: Win32, Linux, LEON3, MicroBlaze

In progress: ARM, XMOS, PowerPC.

For more information, please contact:

Eric Verhulst at eric.verhulst (@) altreonic.com Tel.:+32 -477 -608 -339

and visit the website at:

www.altreonic.com

Embedded World Hall 10, 401