Flexible Embedded Systems Architectures for a Future of Change

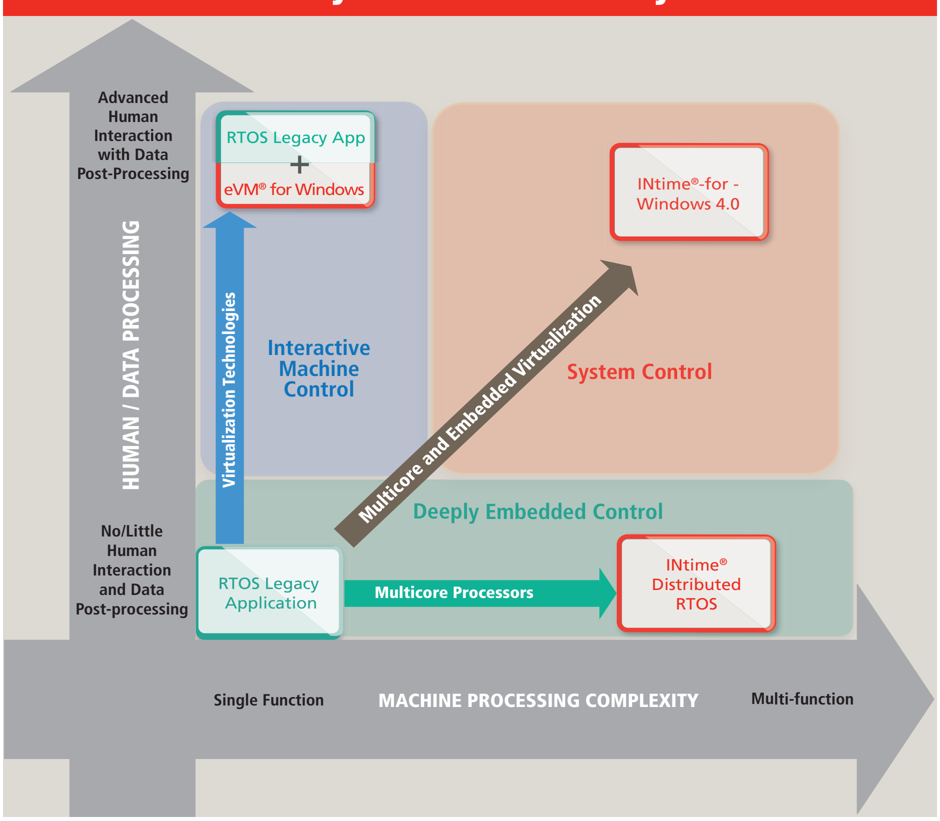
Two major trends will drive the software architecture of embedded systems in the future. One is the partitioning of applications to make the best use of multi-core processors, and the other is the growing use of networked computing to distribute intelligence among multiple computing platforms.

By Chris Grujon, TenAsys Corp.

In an interesting way, these trends almost oppose one another. With the advent of multi-core processors with 2, 4, 8, or more processing units comes the prospect of consolidating embedded systems that were once comprised of multiple discrete processor platforms onto a single platform with different tasks running on different cores.

This consolidation has the promise of decreasing system costs by eliminating the need for redundant computing support hardware. Additionally, improvements in network architectures and speeds allow for increased use of distributed computing in the future. For example, putting real-time processing nodes close to where the system's physical work is being done improves performance for specific tasks. Whether the integrated platform approach, or the distributed computing approach, or the combination of both approaches is best for a particular embedded system depends on the application. Embedded system software architectures need to be flexible enough to accommodate all of these system implementation models.

Real-time systems evolutionary matrix



One way of profiling embedded system software requirements is to consider the evolution of embedded system applications:

In the beginning, real-time operating systems (RTOS) were used for single-function/purpose applications and were run on dedicated processors. An example is iRMX, an RTOS with rudimentary interfaces like all computers in those days. As personal computers evolved, people saw the value of adopting its technology for advanced graphical user interfaces. By providing more meaningful information to the user about the devices and/or system that is being controlled, he/she may perform more sophisticated functions, thereby enhancing the value of the device/system that is being controlled. Two approaches were taken:

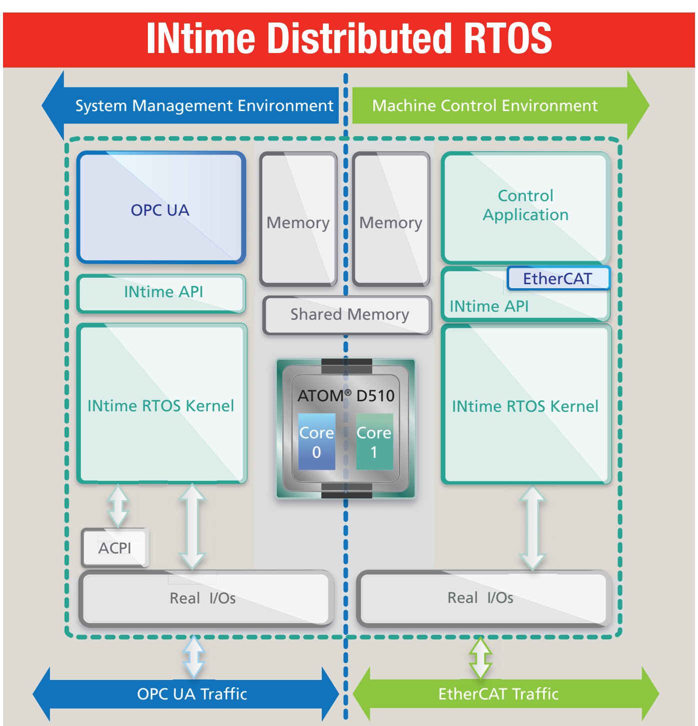
Some RTOS vendors added functionality to enhance the interface. These often didn't meet the expectation and users eventually evolved to a two-box solution, one running a Windows-based user interface and the other running the controlling software on the RTOS.

Others, for example TenAsys with INtime for Windows, took the approach of coupling their RTOS to Windows, providing the advanced user interface and the controlling software onto one platform.

Multi-OS multi-core systems are enabled

With the advent of multi-core platforms, integrated Windows® / RTOS solutions such as INtime® for Windows can run on the same platform with some processor cores dedicated to Windows and others dedicated to the RTOS. Segregating processing capacity in this manner ensures that both domains work independently without affecting each other's performance.

Multi-OS embedded systems will become commonplace in the future, and the secret to getting multiple operating environments to work together reliably on the same platform is embedded virtualization. To support this, processor hardware vendors such as Intel® have developed hardware-assisted virtualization support in their multicore processors. Now legacy single-function RTOS solutions can be upgraded with the addition of a Windows-based user interface. Likewise, systems that were using a two-box processor solution can be combined onto one platform. For example, TenAsys' eVM® for Windows embedded virtualization manager allows guest operating systems such as legacy RTOSes to run alongside Windows without requiring any modifications to the guest OS or its application software, or to Windows and the applications running on it.



Networked embedded systems growing in popularity

The multi-OS on multi-core trend is only one direction for the evolution of embedded systems. The other trend is increased use of networking and distributed computing. Embedded system designers want to build scalable applications with the flexibility to leverage multi-core processing by concentrating applications onto one platform, when possible, or by distributing applications across a network in order to be closer to the control point using an industry-standard network like Ethernet.

TenAsys' INtime Distributed RTOS is the first RTOS designed to leverage state-of-the-art multicore processing and computing across a network with a built-in application capability called GOBSnet (Global Objects Network). This allows multiple instances of the INtime RTOS and applications running on them to intercommunicate via semaphores and mail boxes in a deterministic way. GOBSnet's architecture is transportable across environments, allowing the same applications to be run on adjacent cores of a multicore processor or a nonadjacenet core on another platform, with no program changes.

INtime Distributed RTOS is based on a tried and proven RTOS kernel that has been used across hundreds of mission critical applications.

Features

- **Distributed AMP Architecture** enables predictable, deterministic inter-core communications across multiple platforms without special hardware for large real-time applications using multi-core processors. Enables segmentation to functional blocks, easier to control and debug.
- **Visual Studio Development Environment**
- **INtime for Windows Compatibility** for easy addition of an advanced HMI with the integration of Windows. Allows current INtime for Windows applications to run without modification.

TenAsys Corporation
1400 NW Compton Drive, #301
Beaverton, OR 97006 USA
+1 (503) 748-4720
fax +1 (503) 748-4730
info@tenasys.com
www.tenasys.com

Real-time Virtualization Experts