In My Opinion: Improving Interoperability with a Common Software Architecture

by Barry G. Brown, Curtiss-Wright Controls Embedded Computing

Interoperability, while presenting a key challenge to military system integrators, also offers great advantages. As COTS vendors increase the level of hardware and software interoperability, they reduce customer risk. Today, while COTS suppliers offer many products with good feature sets, the products are typically produced in competitive isolation. A more cohesive vision for the COTS industry calls for a comprehensive approach to compatibility and interoperability. This philosophy has been adopted by Curtiss-Wright Controls Embedded Computing's COTS Continuum initiative. Under this initiative the company defines common software and hardware architectures spanning multiple product families (**Figure 1, see page 20**). The architecture standardizes APIs to all hardware functionality, I/O routing and pin-outs, electrical interfaces, a HAL (Hardware Abstraction Layer), RTOS support, and user documentation. The result is a commonality between product families that enables technology reuse, migration, and insertion.

We believe that adoption of this type of Common Software Architecture (CSA) will increase the productivity of software engineers and system integrators, as they will need to learn and use fewer APIs. Customers utilizing CSAs will experience reduced application software complexity, particularly with infrastructure software that interacts with multiple modules in a system. Designers will be able to reuse more of their application code in more product generations because forward and backward compatibility are fundamental to a CSA API. Improved software interoperability lowers software development costs and speeds

continued on page 20

About the Author

Barry G. Brown has over 18 years of experience in leading architecture, development and integration teams for Real Time Embedded systems in harsh industrial and Mil/Aero applications. He has been with Curtiss-Wright Controls Embedded Computing (formerly Dy 4 Systems) for six years and is a Software Development Manager for the Single Board Computer group and an advocate coordinator for Common Software Architectures.

continued from page 3

software porting, which provides economic and time-to-market advantages for customers.

The CSA employs several strategies for software and hardware development. Software interoperability can be enhanced with the use of a common API layer, and the software interfaces addressed by the CSA include:

• PCI devices and switched fabric nodes including enumeration, memory mapping, and interrupt processing

• Built-in test (BIT) framework and routines to extend BIT in new or modified tests including application-specific system level BIT functions

• Firmware features such as board initialization, Ethernet download, flash programming, memory scrub (declassification), and access to recovery mode boot images

• Device driver programming interfaces that are highly portable and efficient across differing multiple product lines and operating systems

• Board support packages (BSPs) with standard and enhanced RTOS features.

Hardware interoperability can be boosted through the definition of a common hardware abstraction layer and prudent design (such as consistent I/O cabling). The HAL will contain libraries of these functions for various chips at the chip code layer and software for any particular board will simply link in the HAL for particular device/chip combinations.

The use of a HAL can shorten the development cycle for device drivers and interface functions for multiple operating systems. The common HAL approach enables future RTOS version upgrades or even porting between commercial RTOSs such as VxWorks or INTEGRITY. Hardware interfaces addressed by the CSA are included in **Table 1**.

The CSA Software Architecture

The prime goals of a CSA approach are to provide the binding elements to facilitate easy integration of many embedded products and to provide a common, comprehensive set of APIs to enable full and efficient usage of all available devices. This approach supports configurable initialization options, and provides quality BSPs and device driver software. To realize these goals requires adherence to a specific set of design principles and guidelines.

The CSA software architecture takes a layered and modular approach to the design framework and facilitates co-development and customer update management. The architecture's guidelines are contained in **Table 2**. To turn these principles into a true CSA requires several practices currently being developed as industry-wide initiatives. On the software front, layers must be defined, with each layer receiving a set of definitions on how they interact with the others. These definitions produce file and interface templates. Developers get the design rules and then produce a template or a sample module that would be in a particular layer, and that module is copied and used to produce all similar modules in that same layer.

Curtiss-Wright Controls has already implemented this approach in current products. The first two rugged VME-64x products to leverage CSA are the VME-184 single board computer and the CHAMP AV-V quad processing DSP card. VITA-46/48 products VPX6-185 and CHAMP AV-VI are also CSA empowered. While the CSA initiative is focused on optimizing future products, it provides additional shim layers that allow backward compatibility with many old legacy APIs used on products currently in service.

One of the primary design goals for the CSA initiative was development of a uniform and shared design framework, both from a user application and internal developer perspective. The use of shared hardware and software for common blocks among the designs is key to achieving high degrees of integration and coordinated development. A goal shared by the entire COTS industry is to minimize product development cost by amortizing software development over a larger number of products containing similar hardware.

Another initiative requires that COTS vendors contribute to and share a common code base to be used by the system integrator or the COTS customer. This implies that all compatible vendors and suppliers must use similar configuration management (CM) practices and development tool sets to ensure that code developed at one vendor compiles, integrates, and tests properly at all sites. To support this initiative, a central software repository, a common installation procedure, and directory structure with good network connectivity is essential. Supporting a central software repository is preferred and should be expected at all COTS customer sites.

The CSA approach will help move the COTS industry to the next level of sophistication and efficiency. To a COTS customer, all boards would look and behave the same no matter which vendor designed or supplied them. The CSA approach is the binding element that the COTS industry needs to move forward with further efficiency and growth.

Curtiss-Wright Controls Embedded Computing

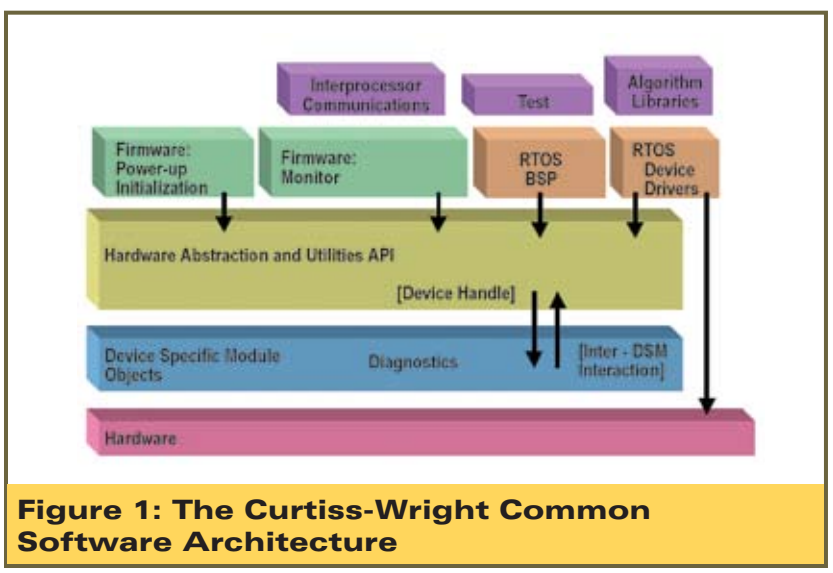

Figure 1: The Curtiss-Wright Common Software Architecture

Table 1. Hardware Interfaces Addressed by the CSA

• Real-time clocks, watchdog and high-resolution timers
• Configuration jumpers and various board reset functions
• Asynchronous and synchronous serial ports
• Discrete, differential, and peripheral I/O
• Bus technologies (VME, PCI, I2C)
• Full memory maps including NVRAM and flash memory
• Dynamic frequency and power configurations
• FPGA configurations
• Mass storage device interfaces
• PMC/XMC standard electrical interfaces

Table 2. CSA Software Architecture Guidelines

• There is no chip-specific code at the user API level
• Abstract hardware specific code is used wherever reuse is desired
• Build monitor, diagnostics, device drivers, and BSPs utilizing abstraction layers
• A single user API is provided for common devices
• A single, long-term roadmap outlines supported tools
• Maximize the technology insertion capabilities, reducing the risk of future enhancement while preserving investments in legacy software
• A broad portfolio of full-featured products