

DSP RTOSs Come of Age, But Developers Need More

By Nick Lethaby

In recent years, the way that DSP software applications are developed has undergone a major transformation. With the number of functions performed by a DSP exploding, writing the majority of code in hand-optimized assembly language has become impractical, and most DSP code is now written in C.

As applications continue to grow in complexity, developers are turning increasingly to real-time operating systems as well. Indeed, DSP RTOSs are coming of age, but more is needed for complete solutions, in particular, device drivers, frameworks, and algorithms.

Three years ago, few DSP system developers used commercial RTOSs. Since then, their use has grown so rapidly that, for example, over half of Texas Instruments' customers now use a commercial RTOS.

Two trends are driving the adoption of DSP RTOSs. In some applications, DSPs perform all system control functions in addition to signal processing and so need an RTOS. Cost-sensitive consumer electronic applications, such as MP3 players, where eliminating the microcontroller reduces the overall system cost, are good examples.

Most DSP applications, however, use one or more DSPs communicating with a microprocessor. In those

cases, the DSP executes multiple signal processing functions, like speech compression and dial tone detection. For maintainable multifunctional systems, a multithreading DSP RTOS is ideal.

Given the size and performance constraints of most DSP applications, most developers today prefer to use a real-time

DSP kernel rather than more heavyweight RTOS kernels typical of microprocessor applications. Obviously, simply porting a standard microprocessor or microcontroller kernel to a DSP is inappropriate for most applications. Unlike their

cousins, DSP kernels must scale to provide basic multithreading support in just a few kilobytes. In addition, interrupt latency and interrupt-to-task time are critical, since unlike microprocessors, DSPs almost always process streams of real-time data.

Now, the market is starting to demand more complete solutions. To meet those demands, DSP RTOSs need to evolve beyond simple kernels to address three key areas—device drivers, frameworks, and algorithms.

Device drivers are notoriously difficult to debug and test. Moreover, no DSP application can be seriously tested on the embedded target without working drivers. Working drivers

for a DSP's on-chip peripherals and commonly used external peripherals enable developers to bring up an application on real hardware much faster. They also eliminate the need for a significant effort to develop software that doesn't differentiate the final product.

Frameworks are particularly critical for developing robust DSP applications quickly. They're the glue that integrates algorithms, I/O, and control code and are ultimately responsible for managing system resources efficiently. By nature, frameworks are application-specific. Successful DSP RTOSs must therefore offer a range of frameworks optimized for different types of applications.

Frameworks must also provide standardized interfaces for integrating proprietary as well as third-party algorithms. With DSP applications now supporting multiple functions, more development teams are buying algorithms from third-party providers. By simplifying the process of integrating algorithms from multiple sources, frameworks can help reduce the system integration time, which already consumes much of the product development cycle.



Nick Lethaby is product manager for the DSP/BIOS real-time kernel at Texas Instruments, Inc.'s Software Development System group in Santa Barbara, Calif.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265