WHITE PAPER

Nick Lethaby
OS Product Manager
Texas Instruments Incorporated

TEXAS INSTRUMENTS

Why Use a Real-Time Operating System in MCU Applications

Introduction

Are you adding more features to each new generation of your microcontroller application? And are internet connectivity and touchscreen UIs becoming mandatory? If so then it's time to switch to a real-time operating system (RTOS). Taking advantage of an off-the-shelf RTOS environment frees you up from focusing too much on low-level peripheral control software and allows you more resources for differentiation.

Traditionally, MCU developers implement software applications around sequential processing loops and state machines. While adequate when the MCU is performing a rather limited number of functions, a different approach is required as MCUs integrate more memory and peripherals. A growing number of MCU applications are now based on a Real-Time Operating Systems (RTOS). The key driver of RTOS adoption is application complexity. An RTOS will often be used when there are more interrupt sources, more functions, and more standard communications interfaces that need to be supported. If the application is <64KB in size, an RTOS is not necessary. Conversely if, the applications is 1 MB, an RTOS will likely be used.

Internal debates within development teams considering use of an RTOS for the first time are passionate. Many MCU developers are accustomed to having intimate control over 100% of the code base and highly optimizing every byte of code and data. It can be challenging to give up this level of control and rely on code from a third-party. While it is unlikely that third-party software will completely match the efficiency of a highly customized run-time, using an RTOS counterbalances this by increasing development team productivity through greater code reuse.

This debate is analogous to "C vs. assembly language". An experienced assembly language programmer will typically generate faster and smaller code than a C compiler. However, using C is almost universally considered a superior approach because the amount of effort to develop an application is much less than when using assembly language. While it is quite possible to do a custom RTOS implementation, it is questionable whether such an approach is optimal. The availability of no-cost off-the-shelf RTOS solutions such as TI-RTOS and FreeRTOS™ remove any barriers associated with product cost. Furthermore, the cost of commercial RTOS products such as µC/OS III®, ThreadX®, SMX®, and Nucleus® is typically much lower than having in-house engineers develop and maintain a custom RTOS implementation.

The below are five productivity advantages and top reasons to condsider using an RTOS:

1. **Preemptive multitasking design paradigm:** For more complex real-time applications, especially those with a code base that is progressively enhanced in each release, the preemptive multitasking design paradigm is superior. This design paradigm makes response-times to each real-time event relatively independent of each other. As a result, new functions can be added without disrupting existing hard real-time ones. In contrast, in a sequential processing loop where each event is checked by polling, the addition of a new event will affect the response times for all events. Although real-time response for critical events can be maintained by use of background/foreground loops or other mechanisms to more frequently poll critical events, more complex real-time applications designed in this way often become very difficult to maintain and modify. Unlike a custom run-time, RTOSs are typically designed to support a wide range of application scenarios and do not need to be modified to support a new application.

2. **Pre-tested and pre-integrated communications stacks and drivers:** An RTOS will typically offer communications stacks such as TCP/IP and USB along with drivers for these and other peripherals. Having such software available working "out of box", eliminates the need for developers to implement it from scratch or spend time integrating third-party software into their run-time environment. In addition most MCU developers are not expert in communications stacks. An RTOS that provides these allows a developer to focus on their area of application expertise and not spend time on undifferentiated system capabilities such as internet connectivity.

3. **Application portability:** By providing a standardized set of stack and driver APIs that abstracts the specifics of the underlying hardware, an RTOS makes applications much more portable. There is much less risk that application software will get interleaved with code performing low-level accesses to peripheral control registers specific to a device. Highly portable code increases ease-of-software reuse.

4. **System-level debug and analysis tools:** As an application becomes more complex, it becomes more likely that it can behave in unanticipated ways. Excessive memory usage or leaks, delayed response to real-time events, or greater than expected CPU loads are all problems that can occur. Problems of this nature are often difficult to diagnose since they require a high-level understanding of system behavior and resource usage. Most RTOSs have associated system-level debugging tools that enable application behavior and resource usage to be examined. For example, TI-RTOS has associated tools that enable developers to look at stack usage and compare it against how much stack was assigned to a task. This makes it straightforward to detect stack overflows or to optimize task stack sizes to free up RAM for other parts of the application. For a custom run-time, a development team will need to create and maintain custom tools.

5. **More efficient use of CPU resources:** Simple loop-based run-times typically do a lot of polling to check if interrupts have occurred. As a result a great deal of processor time is occupied doing nothing. Because multitasking RTOS-based applications are interrupt-driven, it is possible to largely eliminate polling from the application. This frees up processor resources for useful work and enables power-saving modes to be invoked during idle periods.

If you are finding that each generation of your MCU application is adding more and more features and that internet connectivity and touchscreen UIs are becoming mandatory, it may be time to consider switching to an RTOS. Taking advantage of an off-the-shelf software environment like TI RTOS frees you up from focusing too much on low-level peripheral control SW and allows more resources for differentiating your application.

E010208

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions of TI 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265