



Q·Kernel

The new generation RTOS

Dual mode RTOS ♦ Micro Kernel ♦ Hard Real-Time
Never disables Interrupts ♦ Threads and Fibers

Why is Q-Kernel a New Generation RTOS?

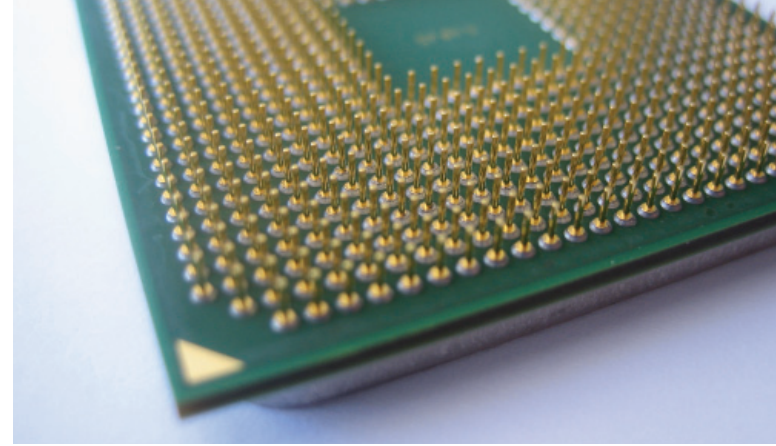
Q-Kernel represents a new generation in real-time operating systems, designed from the ground up to address the demands of today's sophisticated embedded systems. At the heart of Q-Kernel is a unique, segmented interrupt architecture that provides dual mode characteristics. The result is an RTOS that is equally suited for 8, 16 and 32-bit MCUs, high performance digital signal processors and the rapidly emerging convergent processors.

What is a Dual Mode RTOS?

A dual mode RTOS provides the developer with a two scheduling characteristics. It provides fibers for high data load applications, like digital signal processors, and threads for control applications. Fibers must be optimized for cooperative scheduling and very fast interrupt response to support the tight time window once the data is collected. To sustain the required very fast context switches the RTOS must support single stack stateless fibers. The threads must be optimized for pre-emptive scheduling and must support flexible mechanisms for interaction between the threads and between interrupts and threads.

How can Q-Kernel Solve Your Interrupt Jitter?

Q-Kernel never disables interrupts and does not limit interrupt handlers to communicate with the RTOS. This feature means that interrupts will never fail to run because the RTOS will never interfere. Conventional real-time operating systems disable interrupts during critical operations like context switches, which is the source of the jitter.



How Does Q-Kernel Solve Your Memory Issues?

Q-Kernel will reduce memory by minimizing the size of the thread stacks and the number of threads. By providing an interrupt stack the thread does not need to provide space for interrupt handlers. The interrupt stack is also used for fibers. Some of the functionality of an application can be moved from threads to fibers or lightweight threads limiting the number of threads. Fibers and lightweight threads do not require a stack reducing memory use.

How Does Q-Kernel Speed Up Development?

Developing applications with Q-Kernel goes faster because the developer can concentrate on the application while Q-Kernel keeps track of the timing aspects of the system. Q-Kernel makes debugging faster because the API is highly type-safe. This means that errors are found during compile time instead of run-time. Stack-overflow protection is available if supported by the MCU and the central error handling feature eliminates the requirement to write error handling code.

To make a purchase, or for additional information about Quasarsoft Ltd. or its products, please visit our website:

www.quasarsoft.com

SALES: sales@quasarsoft.com
INFO: info@quasarsoft.com
SUPPORT: support@quasarsoft.com

CONTACT INFO:

Tel: +1 (403) 450 3482

Quasarsoft Ltd.
POBox 687, Cochrane
Alberta, Canada
T4C 1A8