Q·Kernel
The new generation RTOS

White Paper
Power consumption

Low power consumption is a deciding factor in many embedded applications. More and more applications like medical devices, wireless communications and personal devices demand low power consumption for a better battery life. Most hardware vendors, like Microchip, have equipped their processors with power saving features.

Many embedded applications spend most of their time waiting for an event to happen – a touch on a panel, incoming communication or wait for a time delay. In many applications the processor is only active for a small amount of time and battery life can be extended significantly by placing the processor in an idle or sleep mode.

The key is to minimize the active time and therefore maximize the idle or sleep time. This can be accomplished by structuring the code in a way that identifies when the system is not performing any tasks. A good approach is to have a thread oriented design and to use an RTOS. In thread oriented designs the RTOS can detect when there is no activity. When that is the case, the RTOS activates the idle thread that can implement power management. In practice, every time the idle task is activated, it puts the processor into a low power mode.

Power management

Most processors, including the 16 and 32 bit chips from Microchip, provide several power saving modes. ***Q·Kernel*** contains integrated power management that makes it simple to lower power consumption. When the RTOS is idle, it signals the application and provides the best power saving mode. The application has the ability to disable additional hardware and instruct the RTOS to select the required power mode. The power management module will select this mode without suffering from race conditions. ***Q·Kernel*** handles all race conditions and makes the implementation simple and flexible.

Power consumption can only be limited if the processor can be placed into idle or sleep mode. This means that state machines with looping and polling don't work well, but interrupt driven applications based on an RTOS do. While an RTOS is a much better solution, it has one disadvantage. Most RTOS require a tick for time management. This causes the processor to be frequently activated which consumes additional power. The shorter the tick-time, the more power is consumed. Applications often require a short tick-time for finer timing. A Tick-Less RTOS solves this problem.

The Tick-Less RTOS

Most RTOS use a tick for time management. This tick is mostly between 50 µSec and 1 mSec and is a type of polling that limits power saving significantly.

Q·Kernel uses a more advanced architecture that eliminates polling completely. It also optimizes power saving by splitting the timing into a human time scale (1 second to >30 years) and a processor time scale (1 µSec to 10 Sec). The human time scale uses the RTCC or the 32 KHz timer that is available in sleep mode and provides more power saving. The processor time scale provides a wait time with a granularity of 1 µSec. When there is an outstanding short time request, the processor can be switched to idle mode. If there is no outstanding short time request, the system will stop the timer and switch to sleep mode. This means that power consumption is always minimized while the system is waiting for user activity.

Quasarsoft
www.QuasarSoft.com