

real time programming tools (not only) for TI 24xx DSPs

Why a real time operating system?

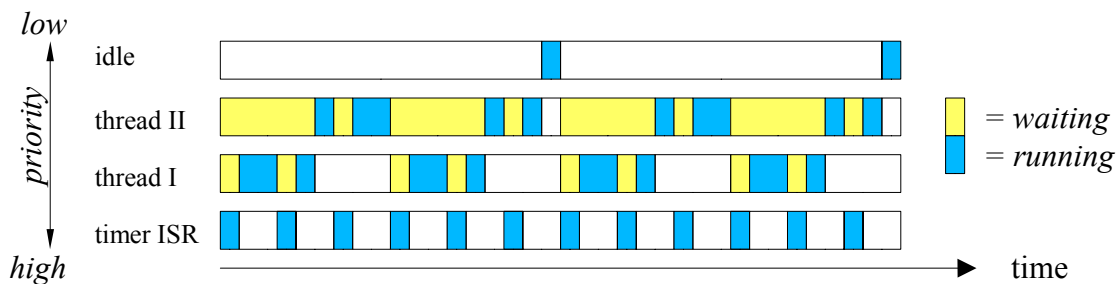
In many DSP applications, several tasks (controller algorithms, communication via different interfaces, inter-task message exchange etc.) must be executed with several priorities. All these tasks have to meet certain real-time requirements (i.e., they must finish within a well defined time). This may lead to a very complex software structure.

The LPKF - RTOS real time kernel was designed to simplify the development of our highly dynamic motor control applications. It was used to realize our SMCU II, a 4 DSP motion controller unit for 2 synchronous axes.

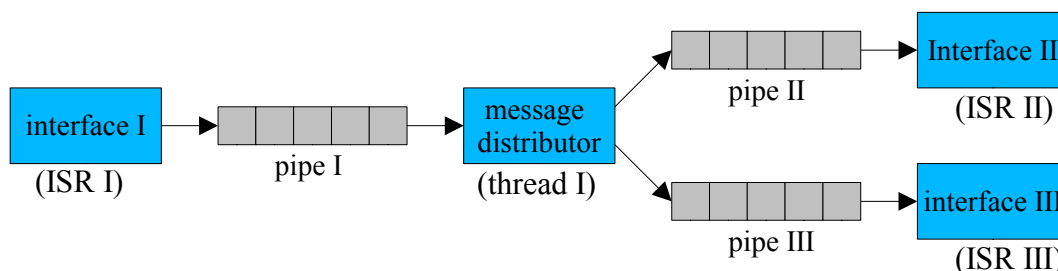
LPKF RTOS real time properties

- priority controlled preemptive scheduling of program threads (hardware / software interrupts)
- deterministic schedule times for high priority tasks
- unidirectional buffered communication channels (pipes) for inter-thread communication / message exchange, designed for message distribution in multiprocessor systems
- timer controlled execution of threads (periodic / one shot execution)
- dynamic memory allocation functions
- single stack system, user program threads must finish within a certain time, i.e. must not wait for resources / events
- object oriented static configuration by the user at compile time (number and properties of software interrupts, pipes, period objects)
- written in C with minimal hardware dependencies, < 1.5 kWord ROM, < 0.5 kWord RAM

Example 1: Program thread scheduling (timer controlled periodic execution)



Example 2: Communication structure (buffered message distribution with pipes)



Implementation / target hardware

LPKF - RTOS was developed for the Texas Instruments C24xx DSP series. It is nearly completely written in C with minimal hardware dependencies and it does not provide any processor specific support for hardware interrupt routines. The LPKF - RTOS may be ported with low effort to other processors.

User configurable modules

hwi	RTOS support routines to be called by hardware interrupts
swi	software interrupts: program threads with priority controlled execution
mem	dynamic memory management of blocks with fixed size (no heap fragmentation)
pip	pipes: unidirectional communication channels with message buffering
prd	timer controlled (periodic / one shot) execution of threads

Performance (measured on Texas Instruments C24xx)

<i>action</i>	<i>24xx cpu cycles</i>	<i>µs @ 40 MHz</i>
max. interrupt latency	< 80	< 2.0
HCI (= hardware C interrupt) prolog / epilog	< 56 / < 55	< 1.4 / < 1.375
HCI preempts swi to higher priority swi (incl. HCI)	< 230	< 5.75
fall back to swi with lower priority (incl. HCI epilog)	< 132	< 3.3
complete pipe message life cycle (incl. swi schedule)	about 600 - 750	about 15 - 18.75

Memory requirements (for Texas Instruments C24xx in 16 bit words)

<i>module</i>	<i>ROM</i>	<i>RAM (rtos data)</i>	<i>RAM (user data)</i>
hwi	0 (inline)	-	-
swi	318 + 4 per object	6 + 2 * 4 per object	-
mem	95 + 5 per object	2 + 2 * 5 per object	n * (framesize + 1)
pip	694 + 14 per object	2 + 2 * 14 per object	n * (2 + framesize + 1)
prd	217 + 8 per object	8 + 2 * 8 per object	-
total	1324 + x	18 + x	x

stack usage (RAM): < 32 Words per priority level + stack used by user application functions

License Conditions

full source code / examples / user manual on CD
no royalty fee for redistribution of object code

support via e-mail
1 year free update

Relation to DSP-BIOS: The LPKF - RTOS uses some of the ideas and concepts of the DSP-BIOS by Texas Instruments, that is available for other DSPs, but not for the 24xx series. However, LPKF - RTOS is not DSP - BIOS, some of API function calls look similar, other are different in the interface and the behavior.