RTA-OSEK
Infineon TriCore with the HighTec Compiler
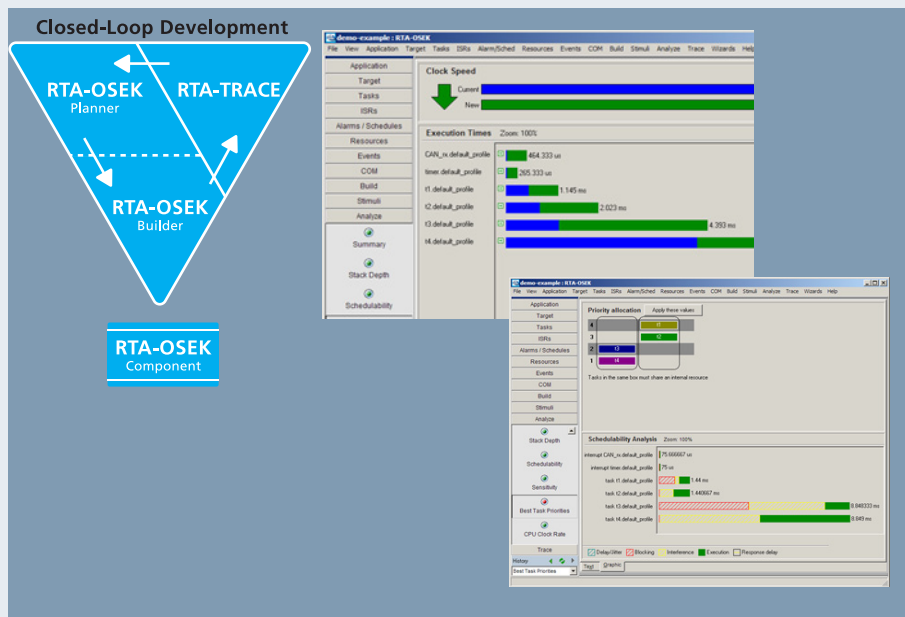
Closed-Loop Development

RTA-OSEK

RTA-OSEK provides an application design environment that combines the smallest and fastest OSEK RTOS with an unique timing analysis tool.

The kernel element of RTA-OSEK is a fixed priority, pre-emptive real-time operating system that is compliant to the OSEK/VDX OS standard version 2.2 for all four conformance classes (BCC1, BCC2, ECC1 and ECC2) and intra processor communication using OSEK COM Conformance Classes A and B (CCCA and CCCB).

All CPU overheads of the kernel have low worst case bounds and little variability in execution time. The kernel is particularly suited to systems with very tight constraints on hardware costs and where run-time performance must be guaranteed.

The kernel is configured using an offline tool provided with RTA-OSEK. Determi-

ning in advance which features are used allows memory requirements to be minimized and API calls to be optimized for greatest efficiency.

All tasks and ISRs in RTA-OSEK run on a single stack – even extended tasks. This allows dramatic reductions in application stack space requirements.

The RTA-OSEK kernel is designed to be scalable. When a task uses queued activation or waits on events, the additional RTOS overhead required to support these features is paid by the task rather than by the system. This means that a basic single activation task uses the same resources in a BCC1 system as it does in an ECC2 system.

Compiler Toolchain

RTA-OSEK supports the HighTec v3.4.5.9 compiler

Memory Model

RTA-OSEK supports a flat 32 bit memory model provided by the HighTec compiler.

ORTI Debugger Support

ORTI is the OSEK Run-Time Interface that is supported by RTA-OSEK for the following debuggers:
Lauterbach TRACE32
Further information about ORTI for RTA-OSEK can be found in the RTA-OSEK ORTI Guide.

Hardware Environment

RTA-OSEK supports all variants of the Infineon TriCore TriCore v1.3 and v1.3.1 CPU family TC1796, TC1792, TC1766, TC1764, TC1762, TC1797, TC1767 and TC1736.

Interrupt Model

RTA-OSEK supports 255 nested levels of interrupt processing.

Floating Point Support

Support for the floating-point hardware of the Infineon TriCore is provided by RTA-OSEK. Only those Category 2 interrupts and tasks that are marked as using floating-point will save the floating-point context. This means that memory overheads are reduced because the floating-point context is not saved for tasks and ISRs that do not use floating-point. Both software and hardware floating-point are supported through correct use of the C run-time libraries.

Functionality

The following table outlines the restrictions on the maximum number of operating system objects allowed by RTA-OSEK

Note that OSEK specifies that queued activations in an ECC2 system are only possible for basic tasks. Where tasks share a priority level, the maximum number of queued activations per priority level is 255.

The number of alarms, tasksets, schedules and schedule arrival-points is only limited by available hardware resources.

	BCC1	BCC2	ECC1	ECC2
Max. no. of tasks		32 plus an idle task		
Max. tasks per priority	1	32	1	32
Max. queued activations	1	255	1	255
Max. events per task	N/A	N/A	32	32
Max. nested resources		255		
Max. alarms		Not limited by RTA-OSEK		
Max. standard resources		255		
Max. internal resources		Not limited by RTA-OSEK		
Max. application modes		2^{32}-1		

Memory Usage

The memory overhead of the core RTA-OSEK kernel is as follows:

Memory Type	Overhead (bytes)
RAM	30
ROM/Flash	150

In addition to the RTOS overhead, each object used by an application has the following memory requirements

Object	RAM (bytes)	ROM (bytes)
BCC1 task	0	36
BCC2 task	10	56
ECC1 task	28	60
ECC2 task	30	68
Category 1 ISR	0	0
Category 2 ISR	0	64
Resource	0	20
Internal Resource	0	0
Event	0	4
Alarm	12	40
Counter	4	108
Schedule Table	16	48
Schedule Table Expiry Point	0	12
Taskset (RW)	4	4
Taskset (RO)	0	4
Schedule	16	36
Arrivalpoint (RW)	12	12
Arrivalpoint (RO)	0	12

Performance

The following table gives the key RTA-OSEK kernel timings in CPU cycles

Task Type	Basic	Extended	Ref
Category 1 ISR Latency	28	28	K
Category 2 ISR Entry Latency	29	29	A
Category 2 ISR Exit Latency	123	314	E
Normal Terminations	69	193	D
ChainTask	222	438	J
Pre-emption	164	356	C
Triggered by Alarm	201	393	F
Schedule	155	346	Q
ReleaseResource	176	366	M
SetEvent	N/A	499	S
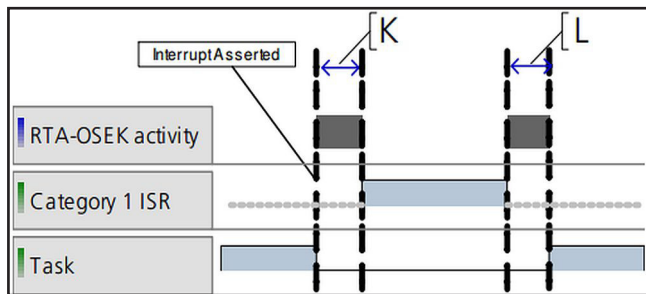
Figure 1 - Category 1 iSR with return to interrupted task
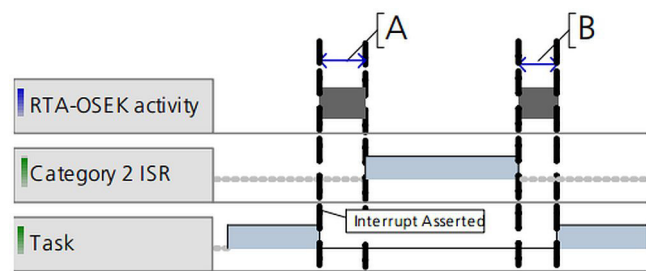
Figure 2 - Category 2 iSR with return to interrupted task
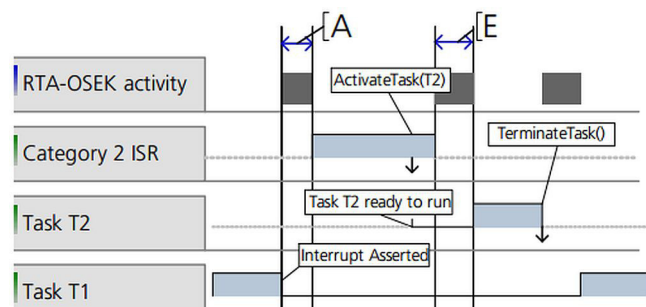
Figure 3 - Category 2 iSR activates a higher priority task
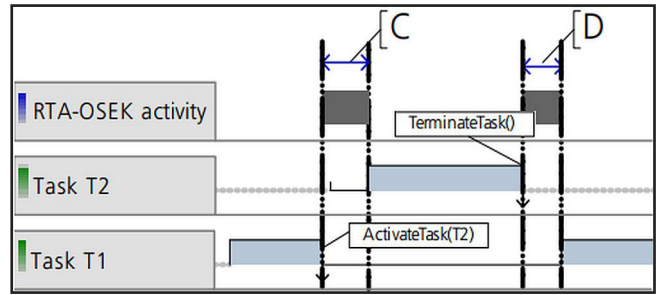
Figure 4 - Task activates a higher priority task
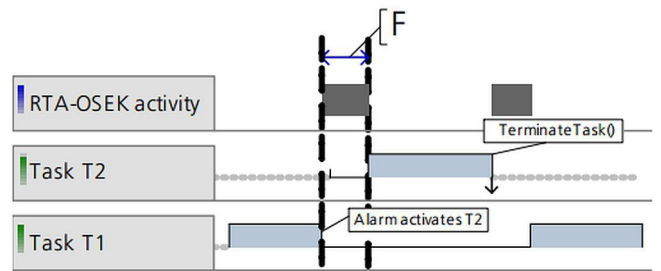
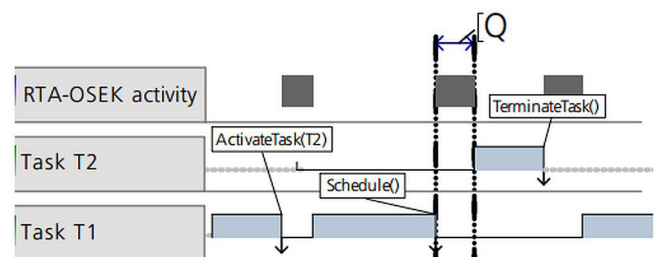**Figure 5 - Alarm activates task**
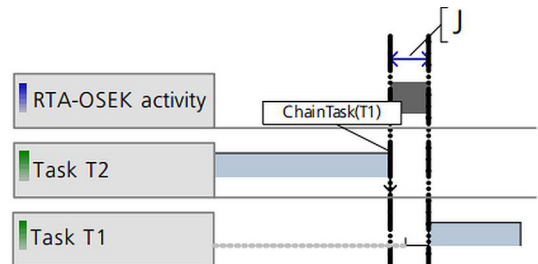
Figure 6 - Task chaining

Figure 7 - Schedule() call

The execution time for every RTA-OSEK API call is available on request from ETAS.
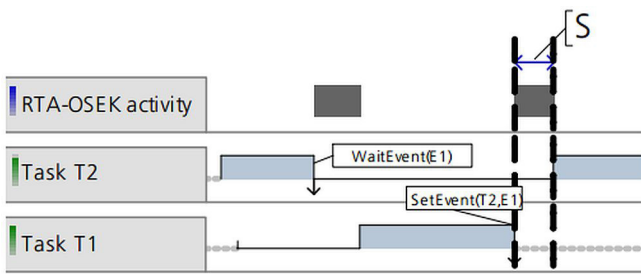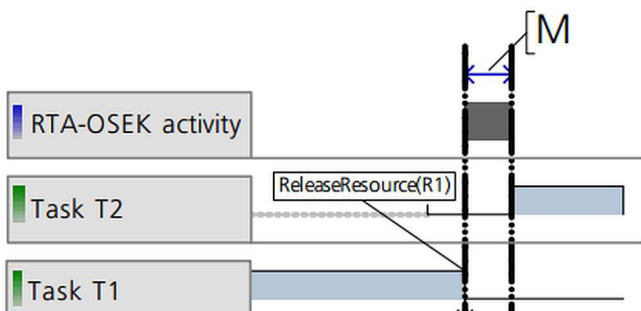
Figure 8 - Activation by SetEvent()

Figure 9 - ReleaseResource()

Order Information

To use RTA-OSEK it is necessary to purchase a development license for the tools together with an add-on license for the Infineon TriCore port. Node-locked and floating licenses are avaialble.

To use any operating system code from the RTA-OSEK libraries in ECU applications, a valid production license is required. Please contact your ETAS Sales Office for details of production licenses for RTA-OSEK.

RTA-OSEK Tools Order Information

Item	Characteristics	Object
F 00K 104 189	Node-locked license for RTA-OSEK Tools	LD_RTA-OSEK
F 00K 104 189	Floating license add-on for RTA-OSEK	LD_RTA-OSEK_FLOAT
F 00K 104 189	Product CD for RTA-OSEK Tools	LD_RTA-OSEK_PROD

RTA-OSEK Infineon TriCore Port Order Information

Item	Characteristics	Object
F 00K 106 733	Node-locked license for RTA-OSEK Infineon TriCore Port	LD_RTA-OSEK_P_5_TRIHIGH
F 00K 106 734	Product CD for RTA-OSEK Infineon TriCore Port	LD_RTA-OSEK_P_5_TRIHIGH_PROD