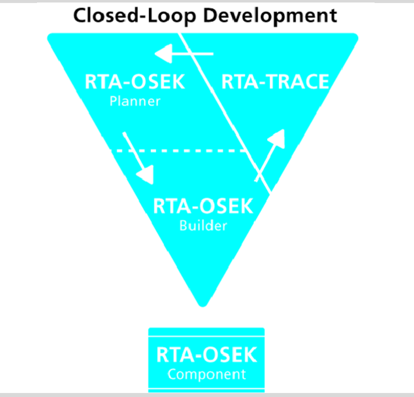
LiveDevices
ETAS Group

RTA-OSEK

Renesas SH2 with the Renesas Compiler



- OSEK/VDX OS version 2.2 certified OS
- RTOS overhead: 28 bytes RAM, 172 bytes ROM
- Category 2 interrupt latency: 70 CPU cycles
- Applications include: Engine Management, Transmission, 3-phase motor control, etc.

RTA-OSEK

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

This port data sheet discusses the Renesas SH2 family port of the RTA-OSEK kernel alone and should be read in conjunction with the Technical Product Overview "*Developing Embedded Real-Time Applications with RTA-OSEK*" available from LiveDevices.

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. Determining 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.

Software Environment

Compiler/Assembler/Linker

The libraries containing the code for the RTA-OSEK kernel have been built using the following tools:

- Renesas (Hitachi) SH Series C/C++ Compiler Version 7.1.02.001
- Renesas (Hitachi) SH Series Assembler Version 5.1
- Renesas (Hitachi) Hitachi Optimising Linkage Editor Version 7.1.07.001

These are components of Hitachi Embedded Workbenck Version

3 Release 21

Memory Model

The SH2 only supports a single memory model (flat 32-bit address space) which is used by RTA-OSEK.

ORTI Debugger Support

ORTI is the OSEK Run-Time Interface. Currently there are no ORTI compatible debuggers supported by RTA-OSEK for this target.

Hardware Environment

RTA-OSEK supports all variants of the Renesas SH2 Family, including the SH7055 and SH7058.

Interrupt Model

The SH2 supports 15 levels of interrupts. RTA-OSEK provides suitable initialisation values for the Interrupt Priority Registers. RTA-OSEK can generate a vector table if required

Floating Point Support

SH2e CPUs contain a hardware floating-point arithmetic unit that is not part of a standard SH2 CPU. The Renesas SH2 port of SSX5 supports the SH2e hardware floating-point. In order to ensure correct functionality of floating-point code in RTA-OSEK tasks and Category 2 ISRs "wrappers" are supplied to save and restore the additional context. To enable this functionality, any tasks and Category 2 ISRs as floating-point using the RTA-OSEK.

On a standard SH2 cpu, RTA-OSEK is designed to work with the fully re-entrant software floating-point libraries supplied by Renesas (Hitachi). This allows floating-point to be used in RTA-OSEK tasks and ISRs without the need to save and restore any additional context.

Evaluation Board Support

This port of RTA-OSEK can be used with any SH2 evaluation board. An example application is provided to run on the Hitachi MS7058CP01/100 evaluation board. This application can be adapted for other target boards by adjusting the linker command file (eg, to alter the allocation of program sections) and one source file (if alternative output pins are required)

Functionality

The below 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 arrivalpoints 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	4294967295			

Memory Usage

The memory overhead of RTA-OSEK is:

Memory type	Overhead (bytes)
RAM	28
ROM/Flash	172

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	52
ECC1 task	92	60
ECC2 task	94	68
Category 1 ISR	0	0
Category 2 ISR	0	48
Resource	0	20
Internal Resource	0	0
Event	0	4
Alarm	8	50
Counter	2	28
Taskset (RW)	4	4
Taskset (RO)	0	4
Schedule	12	36
Arrivalpoint (RW)	12	12
Arrivalpoint (RO)	0	12

In addition to these static memory requirements each task priority and Category 2 interrupt has a stack overhead (in addition to application stack usage). The single stack model means that this overhead applies to each priority level rather than to each task. Similarly, for Category 2 interrupts this overhead applies for each unique interrupt priority. The below table shows stack usage for these objects.

RTA-OSEK provides an optimization for task termination if the

Object	Stack Bytes
Task priority level	84
Category 2 interrupt	52

user can guarantee that tasks only terminate from their entry function. Tasks that terminate from elsewhere are not eligible for this optimization and duly require 84 more stack bytes per priority level than indicated in the table above.

Performance

The following table gives the key kernel timings for operating system behavior in CPU cycles.

Task Type	Basic	Extended	Ref
Category 1 ISR Latency	48	48	K
Category 2 ISR Latency	70	70	A
Normal Termination	33	87	D
ChainTask	76	180	J
Pre-emption	77	147	C
Triggered by alarm	119	188	F
Schedule	70	135	Q
ReleaseResource	78	144	M
SetEvent	n/a	239	S
Category 2 exit switch latency	67	131	E

All performance figures are for the non-optimized interface to RTA-OSEK. Using the optimized interface will result in shorter execution times for some operations. All tasks use lightweight termination and no pre or post task hooks were specified.

The execution time for every kernel API call is available on request from LiveDevices.

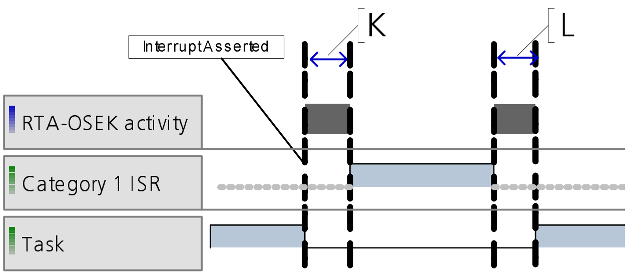


Figure 1 - Category 1 interrupt with return to interrupted task

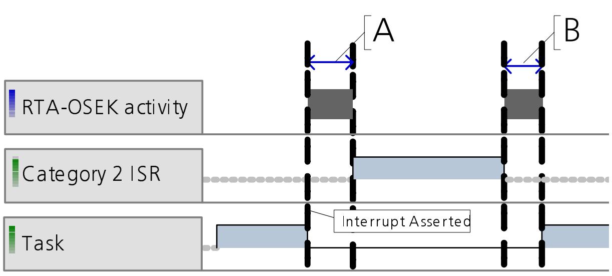


Figure 2 - Category 2 interrupt with return to interrupted task

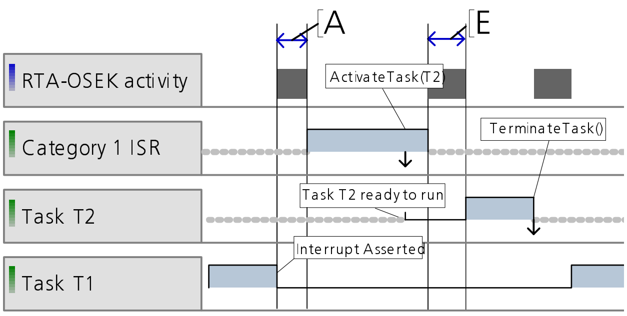


Figure 3 - Category 2 interrupt activates a higher priority task

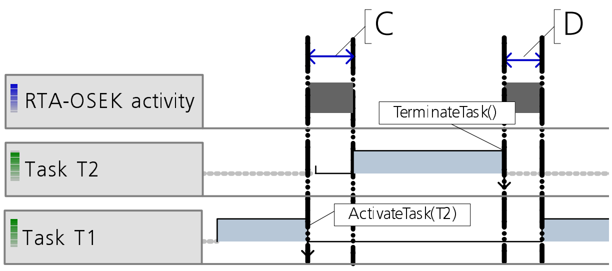


Figure 4 - Task activates a higher priority task

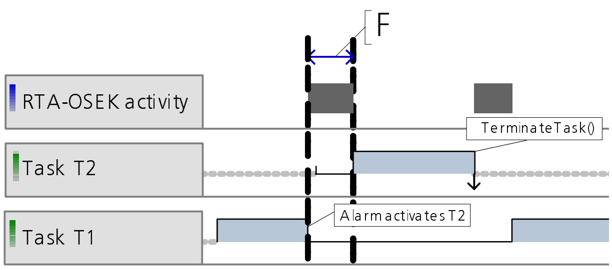


Figure 5 - Alarm activates task

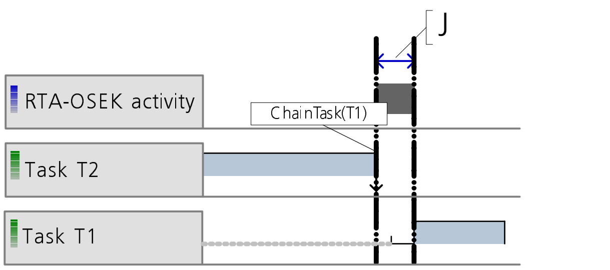


Figure 6 - Task chaining

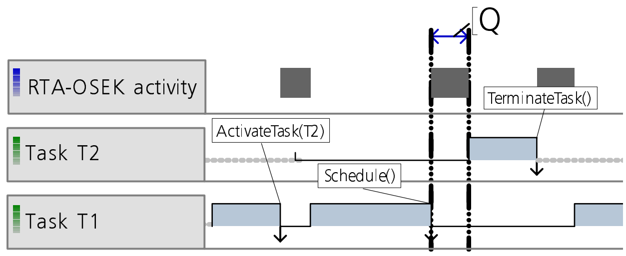


Figure 7 - Schedule() call

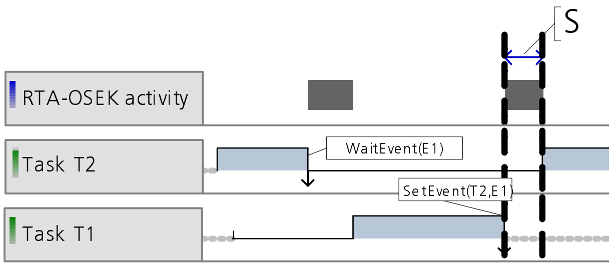


Figure 8 - Activation by SetEvent()

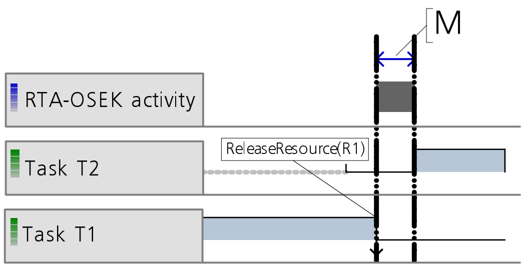


Figure 9 - ReleaseResource()

Benchmarks

The following sections shows benchmarks for RTA-OSEK memory usage for BCC1, BCC2, ECC1 and ECC2 conformant applications. The applications have the following framework:

- 8 tasks plus the idle task
- All basic tasks are lightweight tasks
- 1 Category 2 ISR with a 10ms minimum inter-arrival time
- 1 Counter
- 7 or 8 alarms, all attached to the same counter
- No resources or internal resources
- No hooks
- No schedules
- No tasksets
- Built using standard status

The following table shows the task priority configuration for each benchmark application:

Task/ISR	Stack (bytes)	Period (ms)	BCC1	BCC2	ECC1	ECC2
ISR1	10	10	IPL1	IPL1	IPL1	IPL1
A	10	10	8	8	8	8
B	20	20	7	7	7	7
C	30	20	6	6	6	6
D	40	30	5	5	5	5
E	50	50	4	4	4	4
F	60	80	3	3	3	3
G	70	100	2	2	2	2
H	80	150	1	1	1	2
Idle	10	-	idle	idle	idle	idle

The overhead figures give the ROM and RAM required for RTA-OSEK in addition to that required by the application. The RAM figure is shown split into RAM data and RAM stack.

BCC1

The BCC1 application uses 8 basic tasks with unique priorities.

This application has the following overheads:

Memory usage	Bytes
OS ROM	**1786**
OS RAM	**846**
comprising RAM data	94
comprising RAM stack	752

BCC2

The BCC2 application uses 8 basic tasks with unique priorities.

Tasks A-G are attached to 7 alarms. Task H is activated multiple times from Task A and has maximum queued activation count of 255.

This application has the following overheads:

Memory usage	Bytes
OS ROM	**2170**
OS RAM	**850**
comprising RAM data	94
comprising RAM stack	756

ECC1

The ECC1 application uses 7 basic tasks and 1 extended task with unique priorities. Task H is the extended task and it waits on a single event that is set by basic tasks A-G.

This application has the following overheads:

Memory usage	Bytes
OS ROM	**2700**
OS RAM	**1062**
comprising RAM data	186
comprising RAM stack	876

ECC2

The ECC2 application uses 6 basic tasks and 2 extended tasks. Tasks G and H are the extended tasks and share a priority. The extended tasks wait on a single event that is set by tasks A-F.

This application has the following overheads:

Memory usage	Bytes
OS ROM	**3336**
OS RAM	**1320**
comprising RAM data	288
comprising RAM stack	1032

Stack Optimization

Using stack optimization with the benchmark example identifies that the following tasks can share internal resources:

"Tasks A, B and C

"Tasks D, E and F

"Tasks G and H

The benefit of this optimization is shown in the following table:

Total Stack Space (bytes)	BCC1	BCC2	ECC1	ECC2
Non-optimized	**1132**	**1136**	**1256**	**1412**
OS Overhead	752	756	876	1032
Application Overhead	380	380	380	380
Optimized	**512**	**516**	**636**	**648**
OS Overhead	332	336	456	468
Application Overhead	180	180	180	180

LiveDevices
ETAS Group

Contact addresses:

LiveDevices Ltd.
Atlas House
Link Business Park
Osbaldwick Link Road
Osbaldwick
York YO10 3JB, Great Britain
Phone+44 (0) 1904-562580
Fax+44 (0) 1904-562581
info@livedevices.com
www.livedevices.com

ETAS GmbH
Borsigstraße 14
70469 Stuttgart, Germany
Phone+49 (711) 89661-102
Fax+49 (711) 89661-106
sales@etas.de
www.etas.de

ETAS Inc.
3021 Miller Road
Ann Arbor, MI 48103, USA
Phone+1 (888) ETAS INC
Fax+1 (734) 997-9449
sales@etas.us
www.etas.us

ETAS K.K.
9-1, Ushikubo 3-chome
Tsuzuki-ku
Yokohama 224-0012, Japan
Phone+81 (45) 912-9550
Fax+81 (45) 912-9552
sales@etas.co.jp
www.etas.co.jp

ETAS S.A.S.
1, place des Etats-Unis
SILIC 307
94588 Rungis Cedex, France
Phone+33 (1) 56700050
Fax+33 (1) 56700051
sales@etas.fr
www.etas.fr

ETAS Korea Co., Ltd.
3F, Samseung Bldg. 61-1
Yangjae-dong, Seocho-gu
Seoul, Korea
Phone+82 (2) 57 47-016
Fax+82 (2) 57 47-120
sales@etas.co.kr
www.etas.co.kr