RTA-OS
VRTA-ux with the GCC Compiler

Port Data Sheet

RTA-OS is the ETAS Classic AUTOSAR OS implementation. RTA-OS supports a wide variety of microcontroller/compiler combinations (RTA-OS ports). This port data sheet describes the support for a virtual RTA-OS environment on a generic Linux installation.

Supported Devices
RTA-OS supports the virtual cores on Linux operating systems.

Toolchain support
This port supports the GCC compiler.

Interrupt model
On the VRTA-ux port, RTA-OS supports 32 levels of nested interrupts.

Memory model
On the VRTA-ux port, RTA-OS uses the native 32 bit memory model of Linux.

Memory overhead of RTA-OS

Object	RAM (bytes)	ROM (bytes)
Task	0	16
Cat 2 ISR	0	32
Resource	4	8
Alarm	12	2
Counter	4	20
Schedule Table	16	16
Expiry Point	0	4

Performance
The following gives the key RTA-OS kernel performance data measured in CPU cycles.

Action	Exec time	Ref
Pre-emption	2682	A
Normal Termination	2646	B
Task Switch	2667	C
ChainTask	4089	D
WaitEvent	4533	E
SetEvent	5724	F
Schedule	2652	G
ReleaseResource	2715	H
Cat 2 ISR Entry Latency	1644	I
Cat 2 ISR Exit Latency – interrupted task	4170	J
Cat 2 ISR Exit Latency – task switch	2646	K
Cat 1 ISR Latency	1515	L

Public | ERS-PRM | 2020-08-17

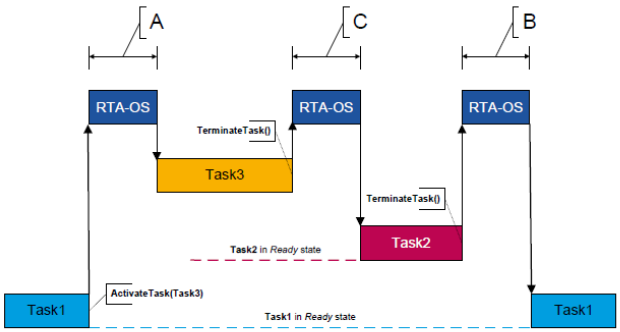

Figure 1 - Task1 is preempted by Task3, followed by a task switch and then normal termination of Task2

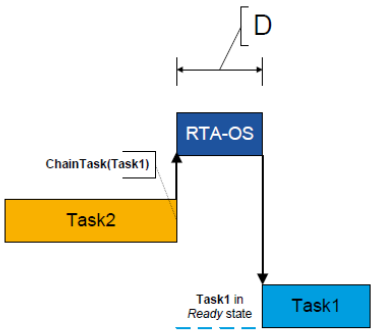

Figure 2 - Task2 chains Task1

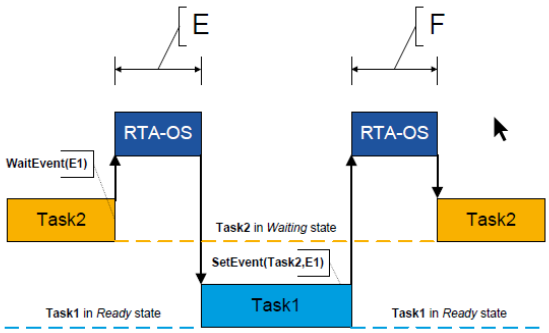

Figure 3 - Task2 waits for an event set by Task1

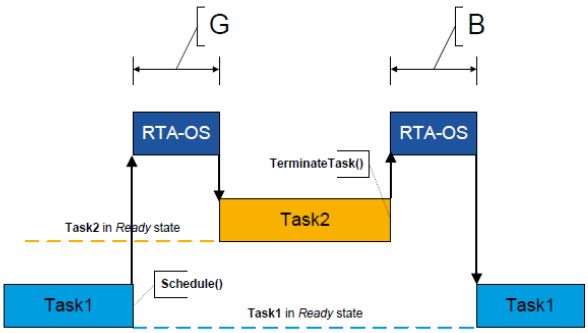

Figure 4 - Task1 allows cooperative scheduling by Task2

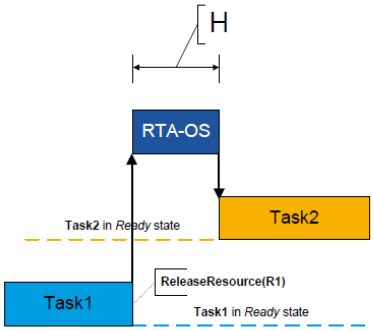

Figure 5 - Task1 releases a resource

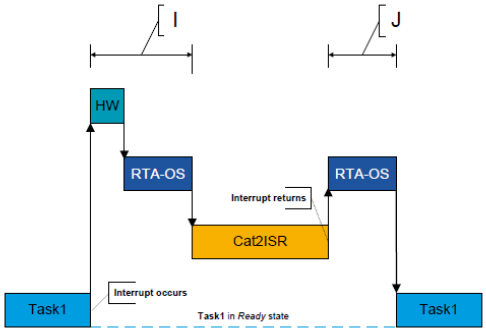

Figure 6 - Category2 ISR entry and exit latency

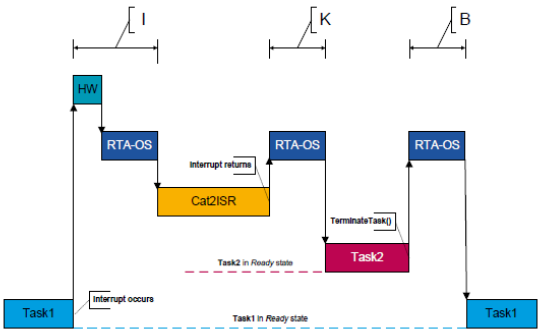

Figure 7 - Category2 ISR switches to Task2

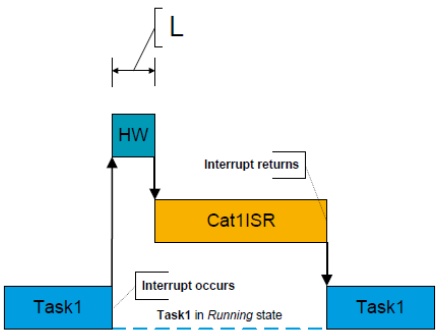

Figure 8 - Category1 ISR entry latency