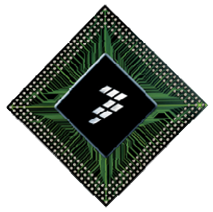
March 2009.

OSEK/RTOS & OSEK*turbo* Introduction



Christian Michel Sendis
Field Applications Engineer

OSEK/VDX Introduction

OSEK/VDXTM

- Began in French and German automotive industries
- Steering committee includes BMW, Daimler Chrysler, Robert Bosch, Siemens, Volkswagen and several other prominent automotive companies.
- Is now the standard for most Operating Systems used in the Automotive Industry.
- Standards-based: ISO Standard 17356

The name OSEK/VDX:
“Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug”
(“Open systems and corresponding interfaces for automotive electronics”)
“Vehicle Distributed eXecutive”

Different standards defined in OSEK/VDX

There are three main standards in OSEK/VDX:

- **OS (Operating System version 2.2) – Provides a common API**
- **COM (Communication version 2.2.2) – Provides the standard interfaces and protocols for data exchange**
- **NM (Network Management version 2.5.1) – Provides the standard functionality to ensure proper operation of in-vehicle networks**

OIL- (OSEK Implementation Language version 2.3) – Provides system configuration and object description (OS and internal COM)

ORTI (OSEK Real Time Interface) – provides debuggers with OS Aware information

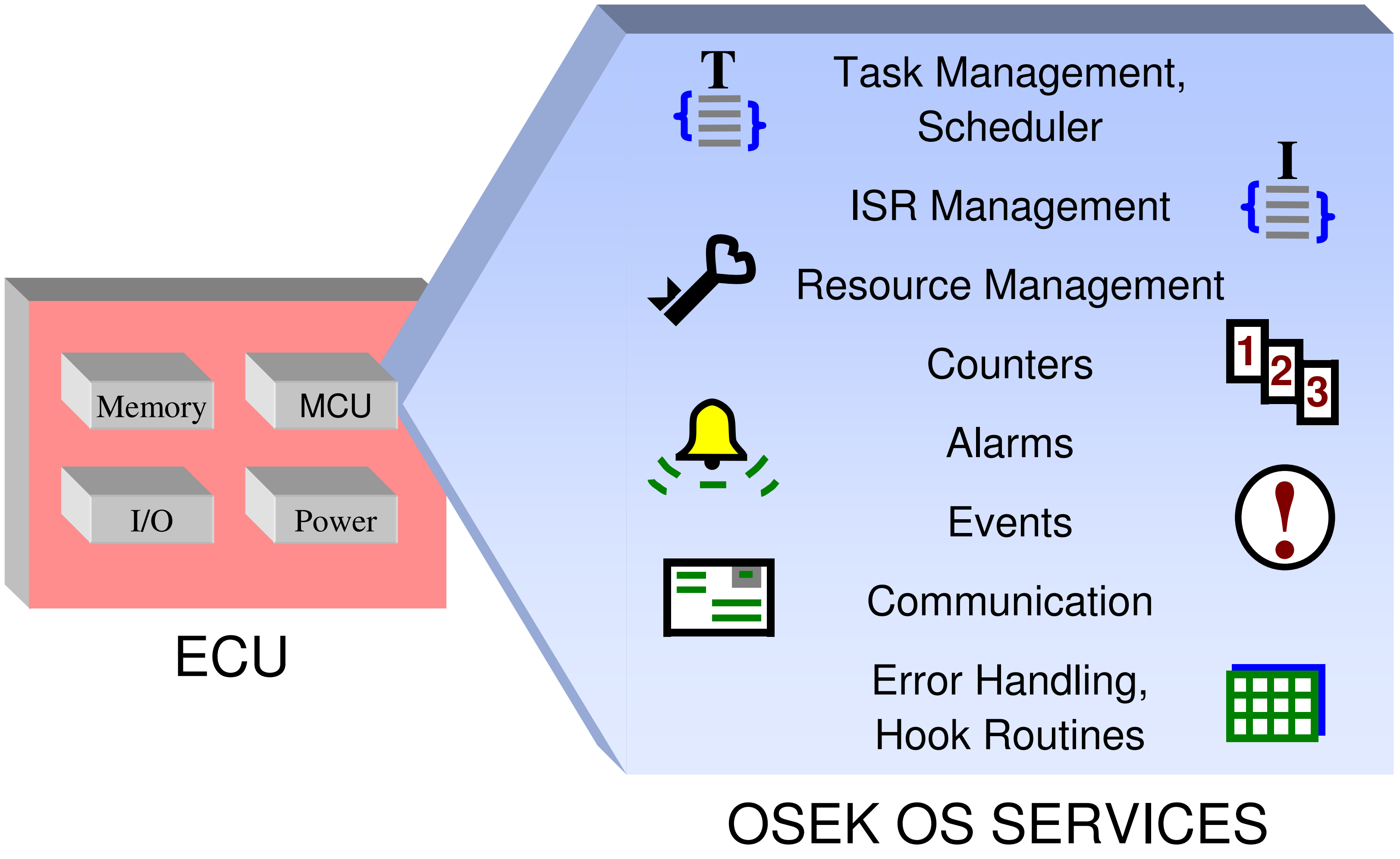
Typical OSEK implementation uses OS, OIL, and subset of the other components

OSEK/VDX OS Introduction

Features that influenced the architectural choices when designing the OSEK Standard:

Scalability
Portability of software
Configurability
Statically allocated OS

OSEK OS Executes in a Single ECU



Tasks in OSEK/VDX OS

Tasks in OSEK/VDX OS

A task provides a frame for executing functions.
Tasks will execute concurrently and asynchronously.

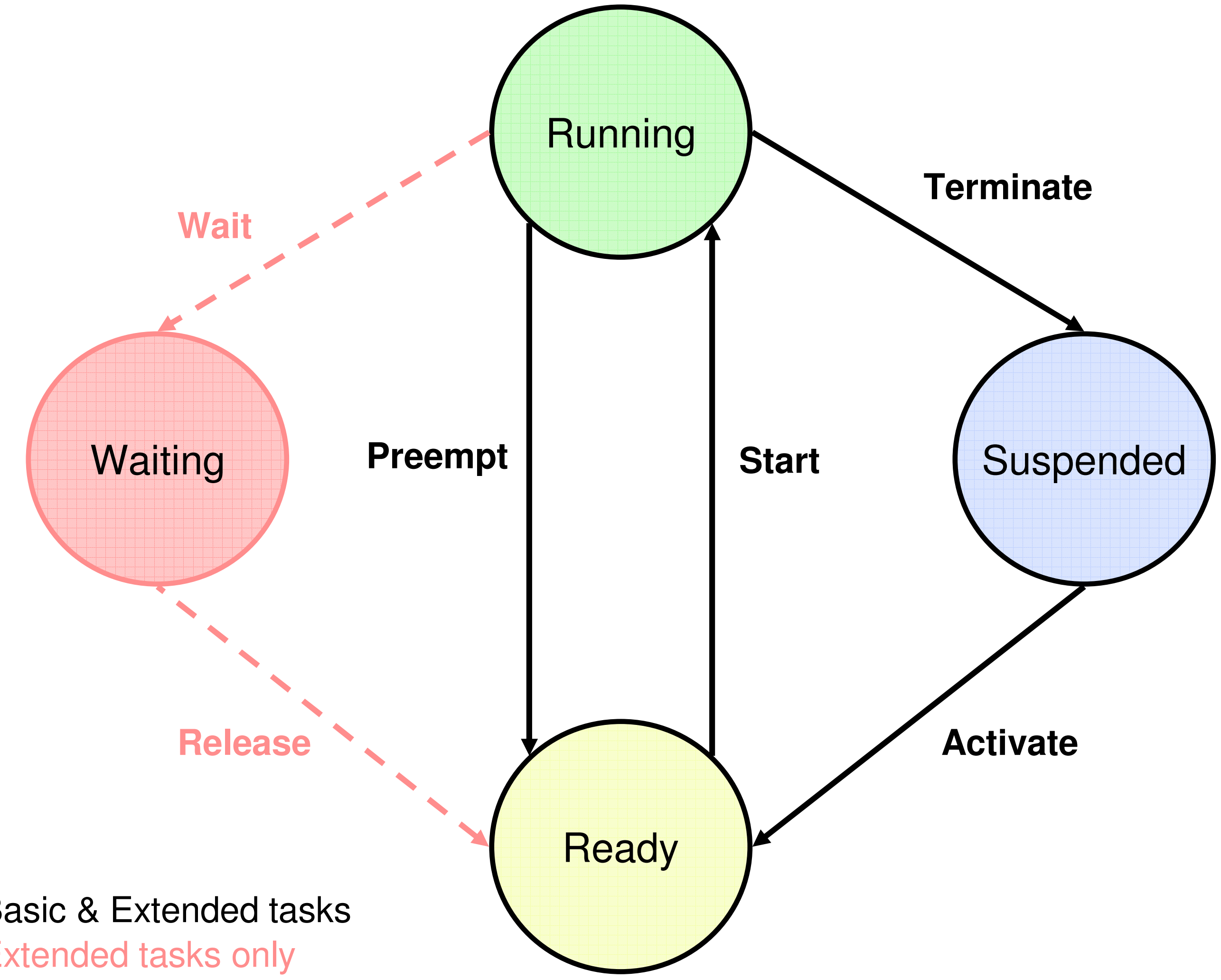
OSEK provides two kinds of tasks:

Basic tasks - no waiting state. Synchronization possible only at start and end of task

Extended Tasks – can use the call to WaitEvent(). Event Synchronization possible several times inside the task.

The termination of a task instance only happen when a task terminates itself (to simplify the OS, no explicit task kill primitives are provided).

Task state transitions



— Basic & Extended tasks

- - - Extended tasks only

Task activation

A task may be activated (transferred from suspended to ready state) by:

- the OS, at system start-up
- another task (service calls: ActivateTask, ChainTask)
- an ISR (service call: ActivateTask)
- an alarm expiration
- sending a message

An extended task is transferred from waiting to ready state (released) by setting an event

Tasks states changes are triggered by the application and the scheduler

Task Termination

- **Tasks must terminate with one of the following API calls**
  - Terminate Itself (with call to TerminateTask())
  - Terminate itself and activate a different task (with call to ChainTask())

  - No “Kill Task ” service is provided. This simplifies the OS.

Scheduling policies in OSEK/VDX OS

Reminder of terms related to scheduling

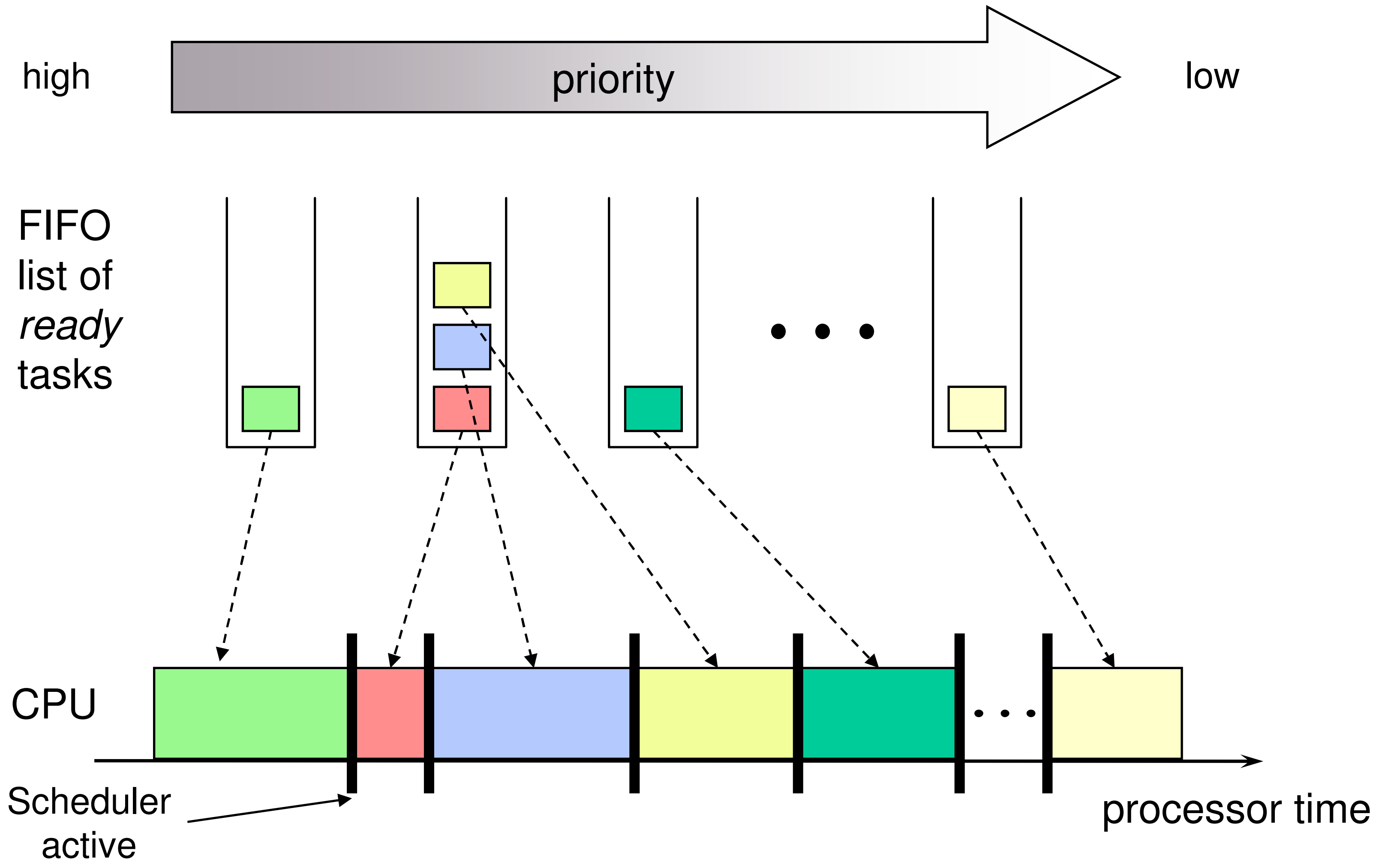
Scheduling
The processor assignment of the tasks is called scheduling. All tasks which havethe state **ready** may assign the processor (if tasks were activated or an event occurred).
Non preemptive scheduling (cooperative)
Tasks can not be preempted by other tasks during their execution.
With the preemptive scheduling tasks with high priority delay tasks with lower priority.
Preemptive scheduling
Tasks may preempted by other tasks during runtime.
A low priority task does not cause a delay for a higher priority task if preemptive scheduling is used. ⇒ fast reaction time of the more important/critical tasks
Static scheduling
The processor assignment of the tasks with static scheduling has a predefined sequence. The sequence of the task execution is determined at compile time.
Dynamic Scheduling
The decision which task is executed is determined at run time. The scheduler adapts to the current task situation.

OSEK uses priority-based dynamic scheduling

OSEK Scheduling Policy

OSEK scheduler will select the highest priority task from the list of ready tasks. It will select the oldest task if more than one of same high priority exists.

Scheduling Policy



OSEK can have 4 different preemption policies:

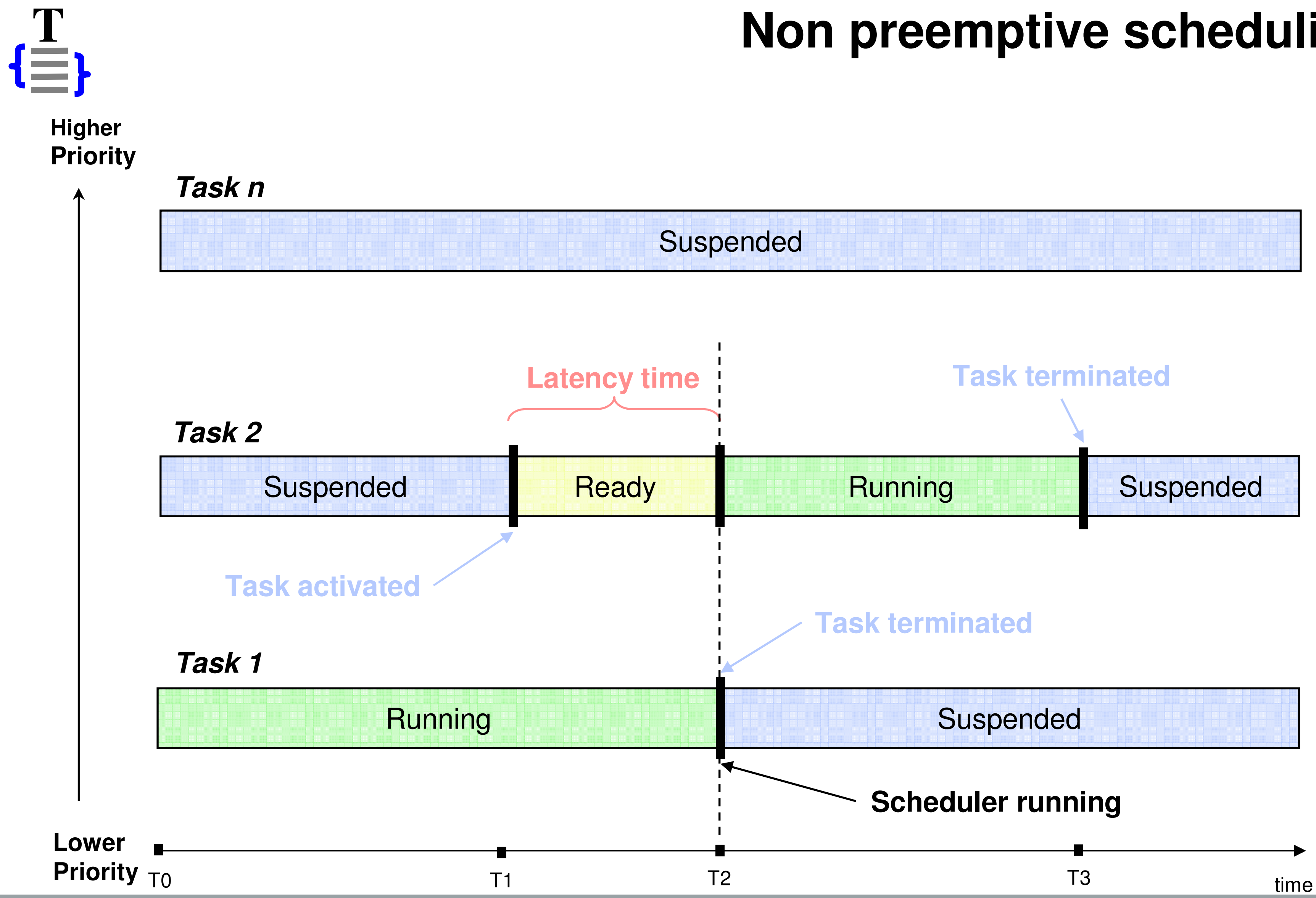
The preemption policy can be chosen by the user :

Four preemption policies:

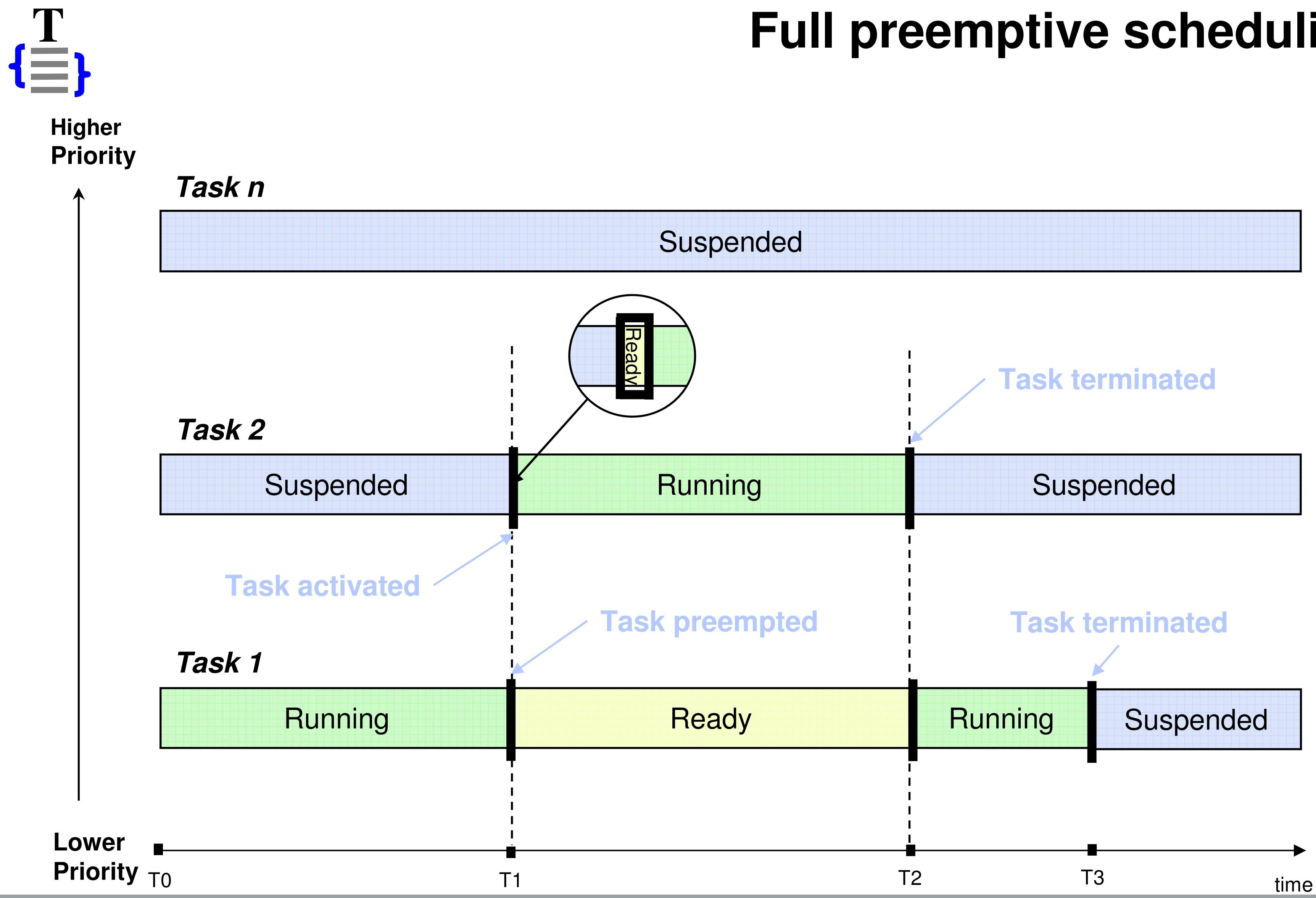
- Non preemptive
- Full preemptive
- Mixed preemptive
- Groups of tasks (cooperative)

Application behavior is specified by task priority and scheduling policy

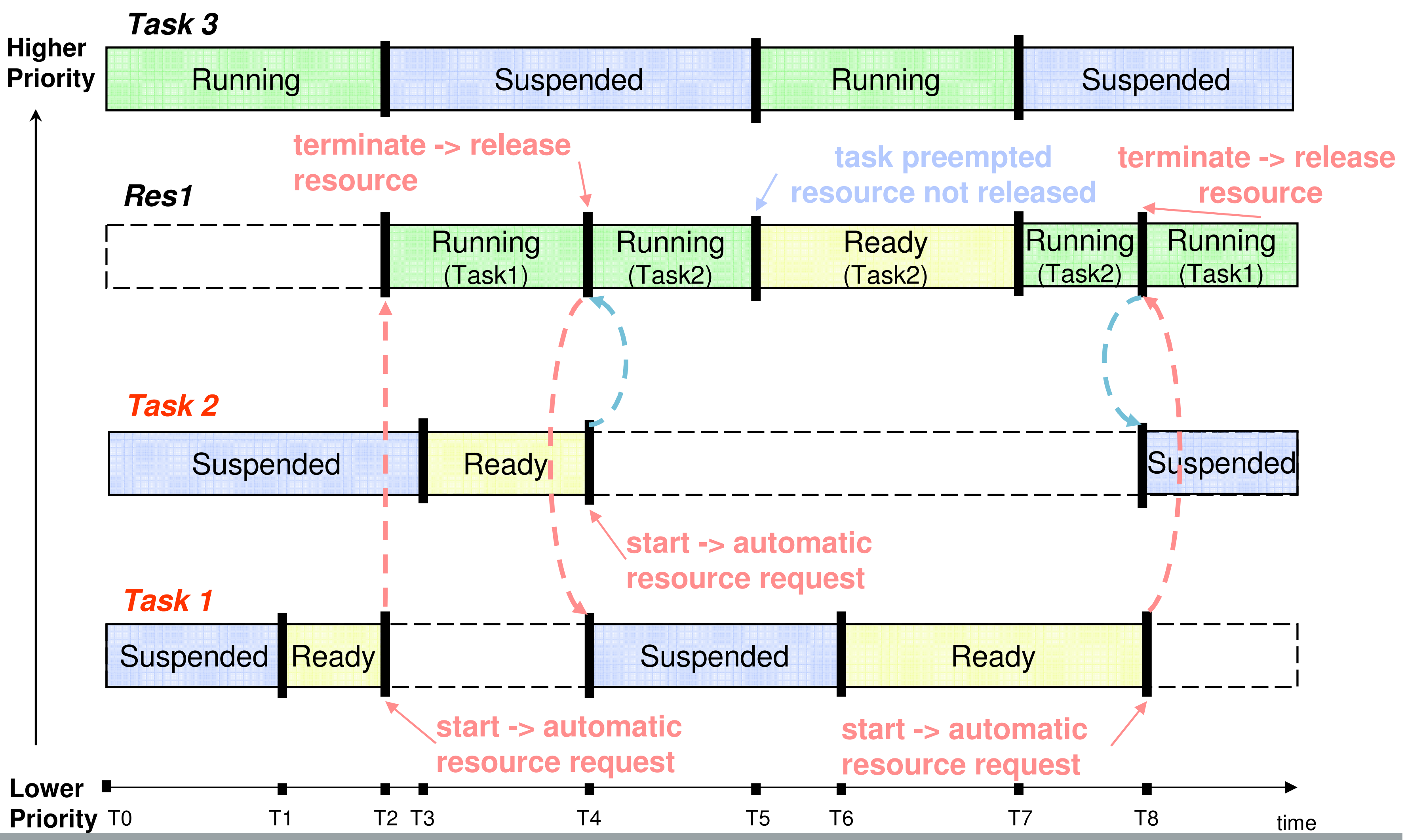
Non preemptive scheduling



Full preemptive scheduling



Groups of tasks / Cooperative scheduling



Conformance Classes

OSEK/VDX OS Conformance Classes

Conformance classes exist to allow partial implementations of the standard along pre-defined lines

The conformance classes specify different requirements for the following attributes:

Multiple requesting task activations (only one activation or more than one)

Task types (basic tasks only or basic and extended tasks)

Number of tasks per priority (one or more than one)

Scaleable through conformance classes

Task Characteristic	BCC1	BCC2	ECC1	ECC2
Basic tasks	✓	✓	✓	✓
Extended tasks			✓	✓
One task per priority	✓		✓	
Multiple tasks per priority		✓		✓
No multiple activations	✓		✓	
Multiple activations for basic tasks		✓		✓

Conformance Class

Interrupts in OSEK

Interrupt Processing

Nested interrupts are possible

Interrupt Service Routines have highest priority in the system

Interrupts are the key part of any real-time operating system

2 categories of ISRs

Category 1 *)

```
ISR (ISR_NAME)
{

  /* code without
     OS calls except
     interrupt */

}
```

Category 2

```
ISR (ISR_NAME)
{


/* code with
     OS calls */


}
```

*) Usage of ISR macro is OSEKturbo specific

Resource management OS services

Resource management

When using resource management

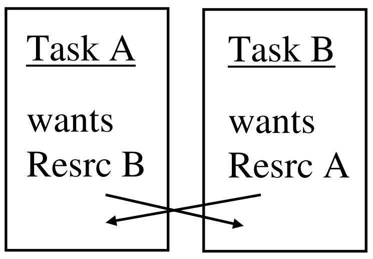
- Two tasks/ISRs cannot “own” the same resource at the same time
- Priority inversion cannot occur while resources are used
- Deadlocks do not occur as a result of using resources
- Tasks accessing resources never enter a Wait state

OSEK Priority ceiling protocol implemented as a resource management discipline

Predefined resource RES_SCHEDULER

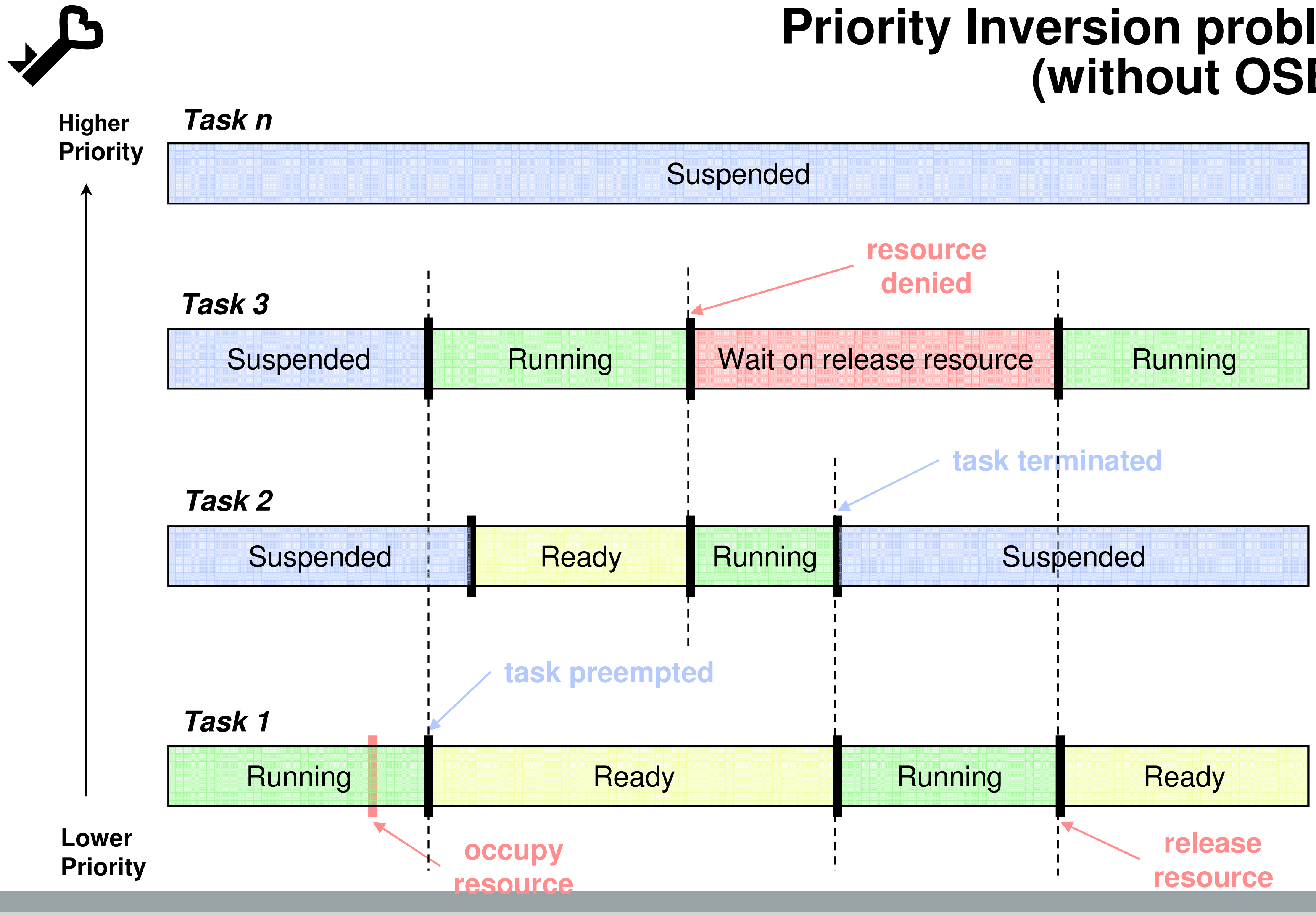
- Resource with ceiling priority higher than all tasks

Deadlock



Concurrent access coordination of shared resources

Priority Inversion problem (without OSEK)



OSEK resource types

Standard resource

- OSEK Priority Ceiling Protocol

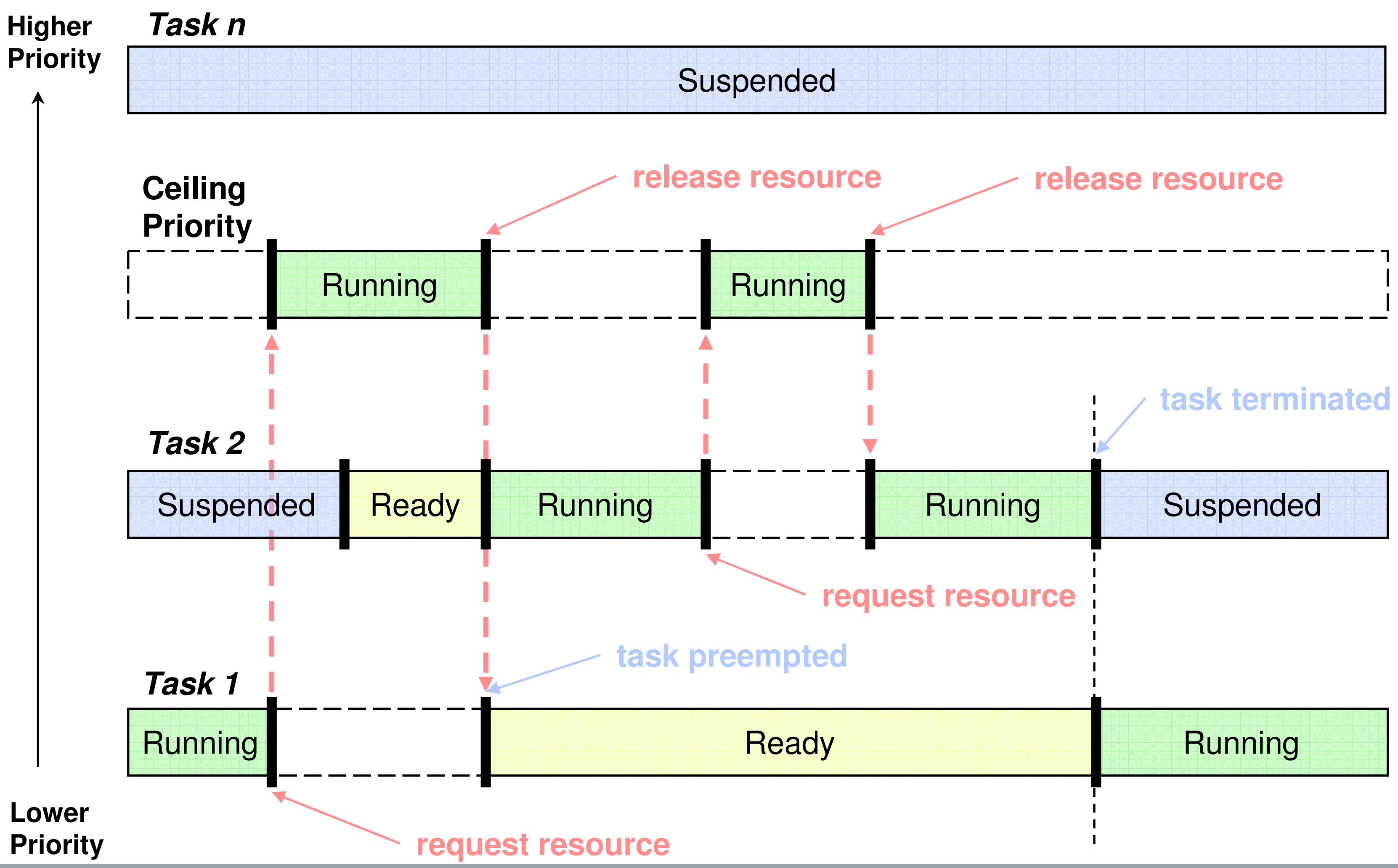
Linked resource

- Has the same properties like another existing resource (priority, type)

Internal resource

- The internal resource is not visible for the user.
- The internal resource has the same behavior as the standard resource (OSEK Priority Ceiling Protocol etc.).
- Groups of tasks is realized through the use of internal resources (Cooperative scheduling).

OSEK Priority Ceiling Protocol



OSEK Summary

- System elements are statically defined
  - All tasks execute out of Flash, EEPROM, or other non-volatile memory
  - No dynamic task allocation
  - No dynamic memory allocation or heap used
  - Fixed stack size
  - Benefits
    - The required resources are fixed quantities
    - System architecture is simplified
- Task priorities are fixed
  - Simplifies task scheduler
  - Scheduler *can* temporarily adjust task’s priority level to resolve deadlocks
- Provides set of OS APIs
  - Offers services for:
    - Task management and synchronization
    - Data exchange
    - Resource control
    - Interrupt handling
  - Uses C-style syntax

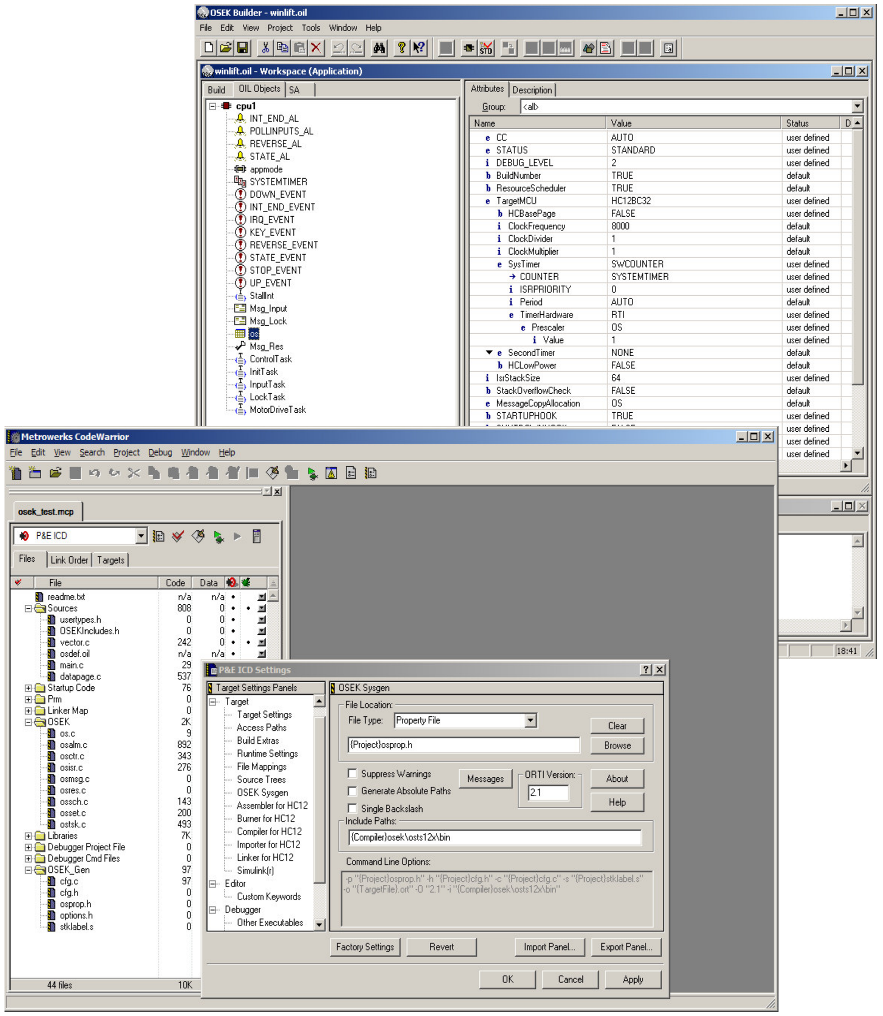
OSEKturbo

OSEKturbo Summary

- Freescale offers an efficient, scalable, reliable implementation of OSEK/VDX called OSEKturbo.
- High quality, efficient, reliable and scalable OSEK/VDX^{TM} compliant Operating System
- Can be used in any embedded application that needs a small footprint RTOS (occupies less than 2K of memory on most controllers)
- Supports deterministic scheduling (timing analysis) to uncover timing problems early in the design phase
- Works seamlessly with CodeWarrior Development Studio with RTOS Aware debugger functionality
- Supports industry leading compilers CodeWarrior, Diab, Cosmic, IAR and more
- Support for HC(S)08, HCS12(X), DSC and Automotive Power Architectures (MPC5xx, 55xx, MPC5200 and MAC7x00) from Freescale.
- Industry Leading Support, Training and Services are available to support your customers.

OSEKturbo Benefits

- The smallest and fastest, fully-certified OSEK implementation available
- Multiple MCU support
- High speed performance and low RAM usage
- Support for all conformance classes available
- ORTI for debugging (debugging standard interface)
- Integration to Codewarrior
- Builder tool for easy configuration of OSEK applications



cpu1
- INT_END_AL
- POLLINPUTS_AL
- REVERSE_AL
- STATE_AL
- appmode
- SYSTEMTIMER
- DOWN_EVENT
- INT_END_EVENT
- IRQ_EVENT
- KEY_EVENT
- REVERSE_EVENT
- STATE_EVENT
- STOP_EVENT
- UP_EVENT
- StallInt
- Msg_Input
- Msg_Lock
- os
- Msg_Res
- ControlTask
- InitTask
- InputTask
- LockTask
- MotorDriveTask

	Name	Value	Status
e	CC	AUTO	user defined
e	STATUS	STANDARD	user defined
i	DEBUG_LEVEL	2	user defined
b	BuildNumber	TRUE	default
b	ResourceScheduler	TRUE	default
e	TargetMCU	HC12BC32	user defined
b	HCBasePage	FALSE	user defined
i	ClockFrequency	8000	default
i	ClockDivider	1	default
i	ClockMultiplier	1	default
e	SysTimer	SWCOUNTER	user defined
→	COUNTER	SYSTEMTIMER	user defined
i	ISRPRIORITY	0	user defined
i	Period	AUTO	default
e	TimerHardware	RTI	user defined
e	Prescaler	OS	user defined
i	Value	1	user defined
e	SecondTimer	NONE	default
b	HCLowPower	FALSE	default
i	IsrStackSize	64	user defined
b	StackOverflowCheck	FALSE	default
e	MessageCopyAllocation	OS	default
b	STARTUPHOOK	TRUE	user defined
			user defined
			user defined
			user defined

File	Code	Data
readme.txt	n/a	n/a
Sources	808	0
usertypes.h	0	0
OSEKIncludes.h	0	0
vector.c	242	0
osdef.oil	n/a	n/a
main.c	29	
datapage.c	537	
Startup Code	76	
Prm	0	
Linker Map	0	
OSEK	2K	
os.c	9	
osalm.c	892	
osctr.c	343	
osisr.c	276	
osmsg.c	0	
osres.c	0	
ossch.c	143	
osset.c	200	
ostsk.c	493	
Libraries	7K	
Debugger Project File	0	
Debugger Cmd Files	0	
OSEK_Gen	97	
cfg.c	97	
cfg.h	0	
osprop.h	0	
options.h	0	
stklabel.s	0	
44 files	10K	

OSEK Sysgen

File Location:
File Type: Property File
{Project}osprop.h

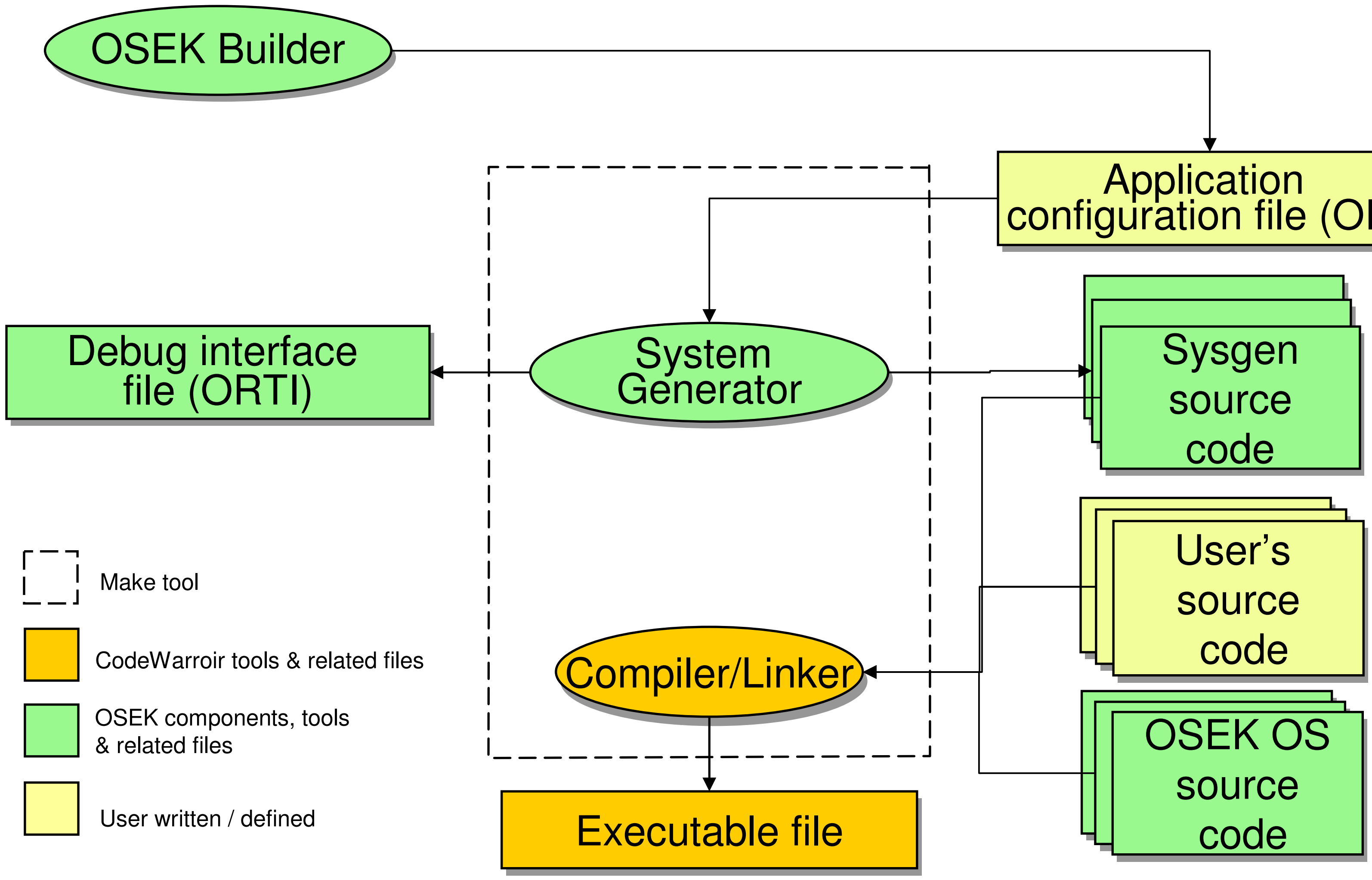
☐ Suppress Warnings
☐ Generate Absolute Paths
☐ Single Backslash

ORTI Version: 2.1

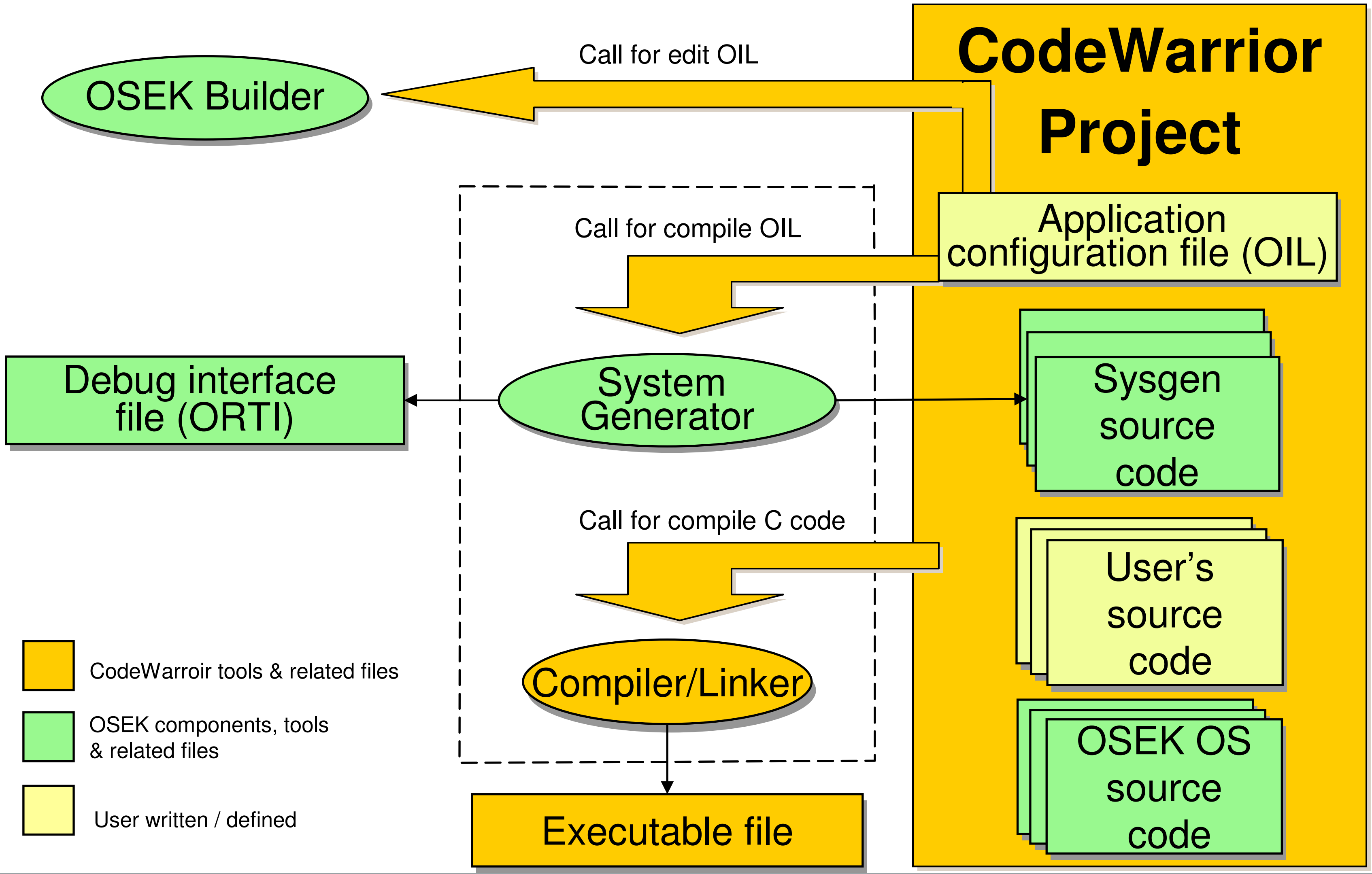
Include Paths:
{Compiler}osek\osts12x\bin

Command Line Options:
```
-p "{Project}osprop.h" -h "{Project}cfg.h" -c "{Project}cfg.c" -s "{Project}stklabel.s"
-o "{TargetFile}.ort" -O "2.1" -i "{Compiler}osek\osts12x\bin"
```

Building an OSEKturbo application (make)



Building an OSEK*turbo* application (IDE)



OSEK Implementation Language - OIL

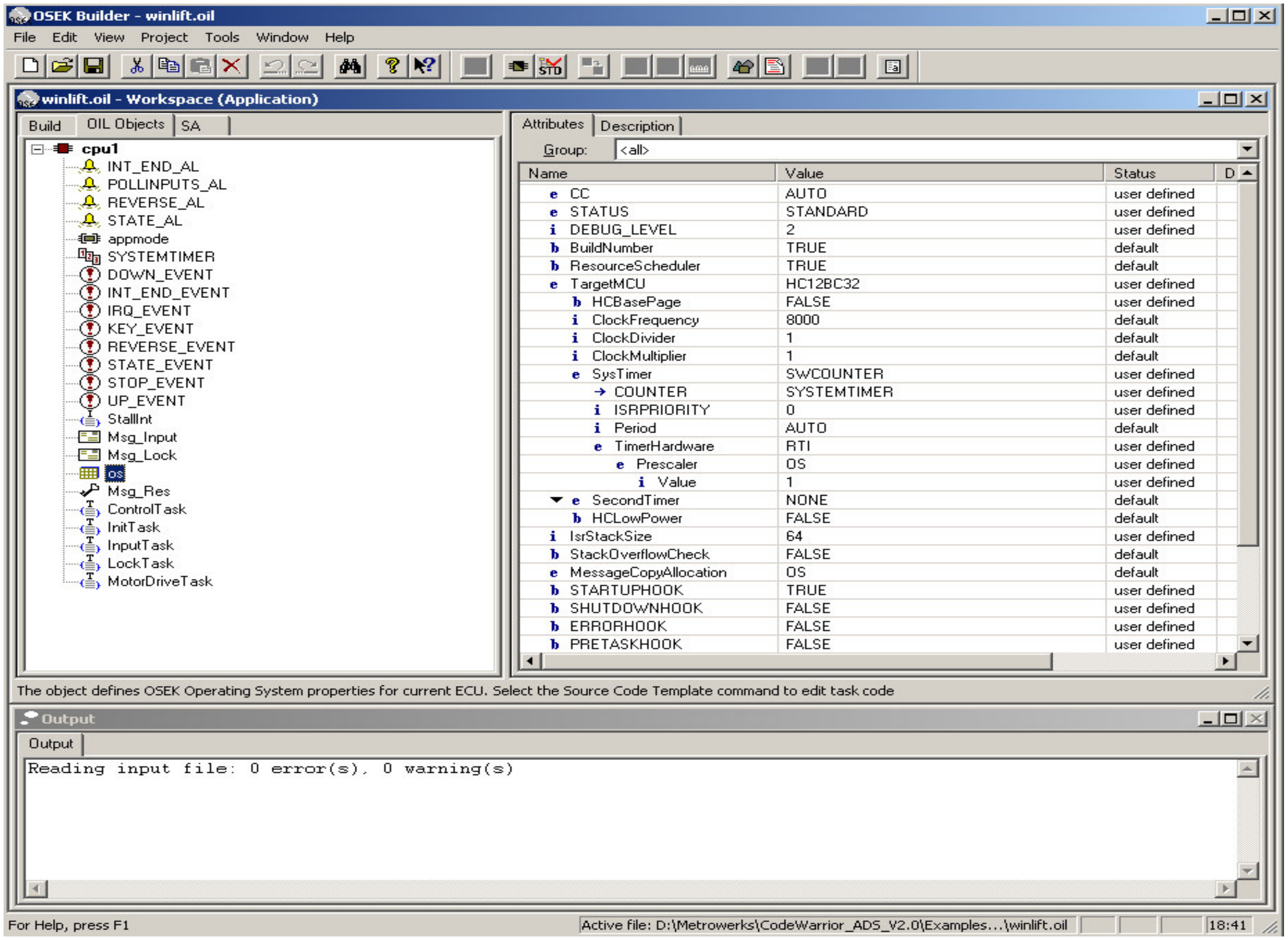
```
OIL_VERSION = "2.3";
IMPLEMENTATION OSEKturbo_OS12_2_2_1_59 {
   OS {
      ENUM WITH_AUTO [BCC1,ECC1] CC =
AUTO;
      ENUM [STANDARD,EXTENDED] STATUS;
      UINT32 [0,1,2,4] DEBUG_LEVEL = 0;
      BOOLEAN STARTUPHOOK;
      BOOLEAN ERRORHOOK;
   };
 COUNTER {
        UINT32 [ 0x0 .. 0xffff ]
MINCYCLE;
        UINT32 [ 0x1 .. 0xffff ]
MAXALLOWEDVALUE;
        UINT32 TICKSPERBASE;
    };

  • • •

};
```

```
CPU cpuname {
   OS osname {
      CC = ECC1;
              STATUS = EXTENDED;
      STARTUPHOOK = TRUE;
      ERRORHOOK = TRUE;
   };
 TASK InitTask {
        PRIORITY = 30;
        SCHEDULE = NON;
        AUTOSTART = TRUE{
            APPMODE = appmode;
        };
        ACTIVATION = 1;
    };
 COUNTER SYSTEMTIMER {
        MAXALLOWEDVALUE = 0xFFFF;
        TICKSPERBASE = 10;
        MINCYCLE = 0;
    };
   • • •
};
```

Builder V2.3, workspace



OSEK Builder - winlift.oil
File Edit View Project Tools Window Help
winlift.oil - Workspace (Application)
Build
OIL Objects
SA
Attributes
Description
Group: <all>
Output

cpu1
- INT_END_AL
- POLLINPUTS_AL
- REVERSE_AL
- STATE_AL
- appmode
- SYSTEMTIMER
- DOWN_EVENT
- INT_END_EVENT
- IRQ_EVENT
- KEY_EVENT
- REVERSE_EVENT
- STATE_EVENT
- STOP_EVENT
- UP_EVENT
- StallInt
- Msg_Input
- Msg_Lock
- OS
- Msg_Res
- ControlTask
- InitTask
- InputTask
- LockTask
- MotorDriveTask

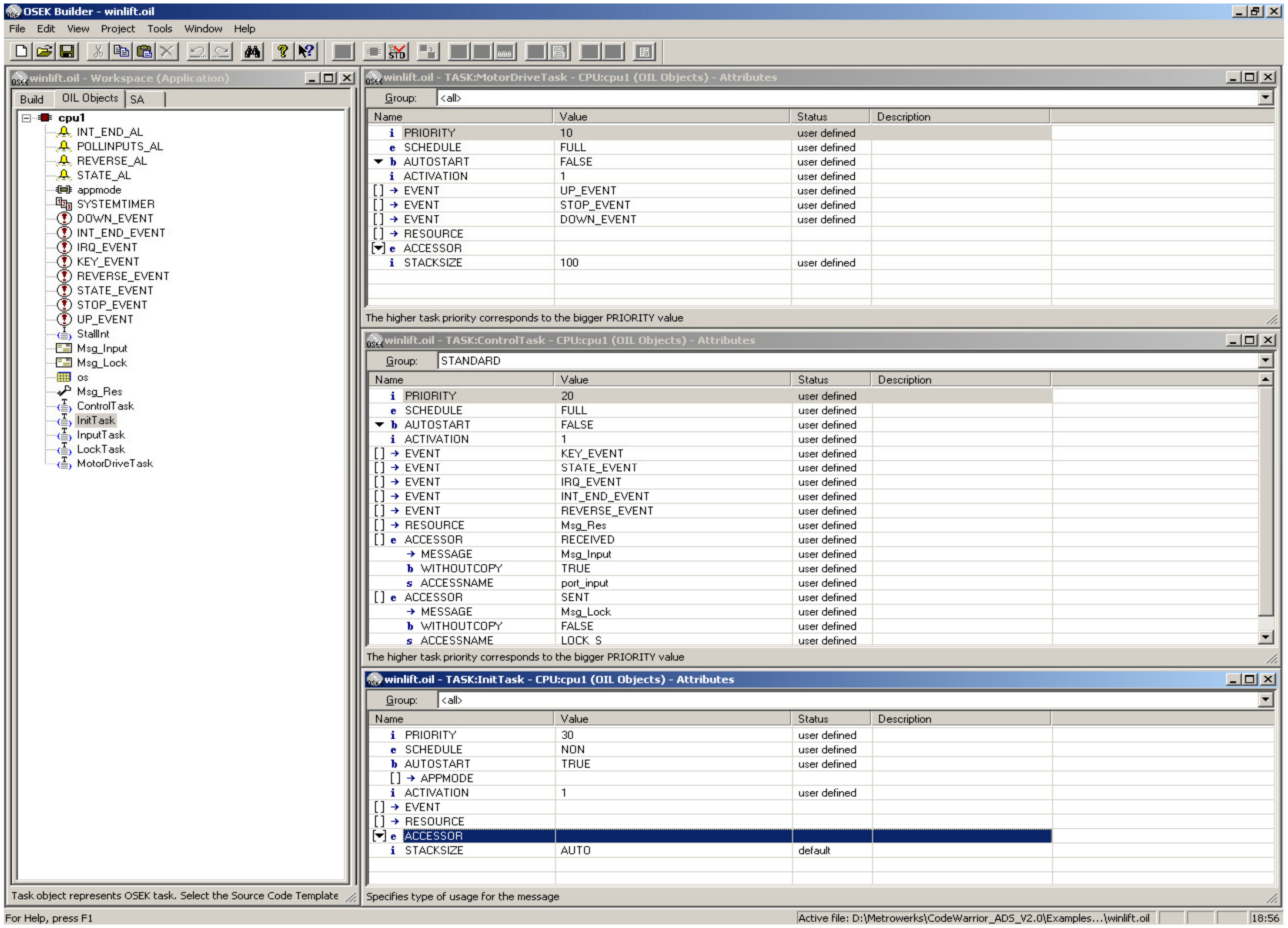
	Name	Value	Status
e	CC	AUTO	user defined
e	STATUS	STANDARD	user defined
i	DEBUG_LEVEL	2	user defined
b	BuildNumber	TRUE	default
b	ResourceScheduler	TRUE	default
e	TargetMCU	HC12BC32	user defined
b	HCBasePage	FALSE	user defined
i	ClockFrequency	8000	default
i	ClockDivider	1	default
i	ClockMultiplier	1	default
e	SysTimer	SWCOUNTER	user defined
→	COUNTER	SYSTEMTIMER	user defined
i	ISRPRIORITY	0	user defined
i	Period	AUTO	default
e	TimerHardware	RTI	user defined
e	Prescaler	OS	user defined
i	Value	1	user defined
e	SecondTimer	NONE	default
b	HCLowPower	FALSE	default
i	IsrStackSize	64	user defined
b	StackOverflowCheck	FALSE	default
e	MessageCopyAllocation	OS	default
b	STARTUPHOOK	TRUE	user defined
b	SHUTDOWNHOOK	FALSE	user defined
b	ERRORHOOK	FALSE	user defined
b	PRETASKHOOK	FALSE	user defined

The object defines OSEK Operating System properties for current ECU. Select the Source Code Template command to edit task code

```
Reading input file: 0 error(s), 0 warning(s)
```

For Help, press F1 | Active file: D:\Metrowerks\CodeWarrior_ADS_V2.0\Examples...\winlift.oil | 18:41

Builder 2.3, tasks


OSEK Builder - winlift.oil
File Edit View Project Tools Window Help
winlift.oil - Workspace (Application)
Build
OIL Objects
SA

- cpu1
  - INT_END_AL
  - POLLINPUTS_AL
  - REVERSE_AL
  - STATE_AL
  - appmode
  - SYSTEMTIMER
  - DOWN_EVENT
  - INT_END_EVENT
  - IRQ_EVENT
  - KEY_EVENT
  - REVERSE_EVENT
  - STATE_EVENT
  - STOP_EVENT
  - UP_EVENT
  - StallInt
  - Msg_Input
  - Msg_Lock
  - os
  - Msg_Res
  - ControlTask
  - InitTask
  - InputTask
  - LockTask
  - MotorDriveTask

Task object represents OSEK task. Select the Source Code Template

winlift.oil - TASK:MotorDriveTask - CPU:cpu1 (OIL Objects) - Attributes

Group: <all>

Name	Value	Status	Description
i PRIORITY	10	user defined	
e SCHEDULE	FULL	user defined	
b AUTOSTART	FALSE	user defined	
i ACTIVATION	1	user defined	
[] → EVENT	UP_EVENT	user defined	
[] → EVENT	STOP_EVENT	user defined	
[] → EVENT	DOWN_EVENT	user defined	
[] → RESOURCE			
e ACCESSOR			
i STACKSIZE	100	user defined	

The higher task priority corresponds to the bigger PRIORITY value

winlift.oil - TASK:ControlTask - CPU:cpu1 (OIL Objects) - Attributes

Group: STANDARD

Name	Value	Status	Description
i PRIORITY	20	user defined	
e SCHEDULE	FULL	user defined	
b AUTOSTART	FALSE	user defined	
i ACTIVATION	1	user defined	
[] → EVENT	KEY_EVENT	user defined	
[] → EVENT	STATE_EVENT	user defined	
[] → EVENT	IRQ_EVENT	user defined	
[] → EVENT	INT_END_EVENT	user defined	
[] → EVENT	REVERSE_EVENT	user defined	
[] → RESOURCE	Msg_Res	user defined	
[] e ACCESSOR	RECEIVED	user defined	
→ MESSAGE	Msg_Input	user defined	
b WITHOUTCOPY	TRUE	user defined	
s ACCESSNAME	port_input	user defined	
[] e ACCESSOR	SENT	user defined	
→ MESSAGE	Msg_Lock	user defined	
b WITHOUTCOPY	FALSE	user defined	
s ACCESSNAME	LOCK_S	user defined	

The higher task priority corresponds to the bigger PRIORITY value

winlift.oil - TASK:InitTask - CPU:cpu1 (OIL Objects) - Attributes

Group: <all>

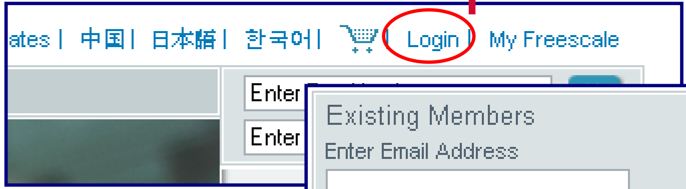
Name	Value	Status	Description
i PRIORITY	30	user defined	
e SCHEDULE	NON	user defined	
b AUTOSTART	TRUE	user defined	
[] → APPMODE			
i ACTIVATION	1	user defined	
[] → EVENT			
[] → RESOURCE			
e ACCESSOR			
i STACKSIZE	AUTO	default	

Specifies type of usage for the message

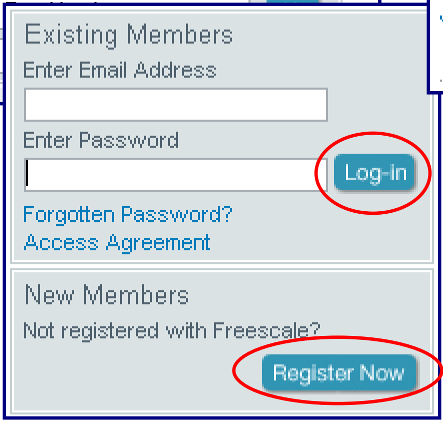
For Help, press F1 | Active file: D:\Metrowerks\CodeWarrior_ADS_V2.0\Examples...\winlift.oil | 18:56

At **www.freescale.com**

Most of your problems solved in less than 24 hours!



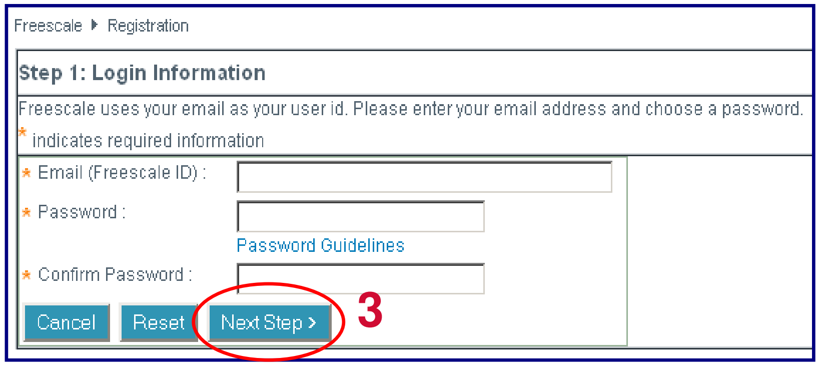
1



4

2

Use the **Freescale Web Technical Support!**


Freescale ▸ Registration

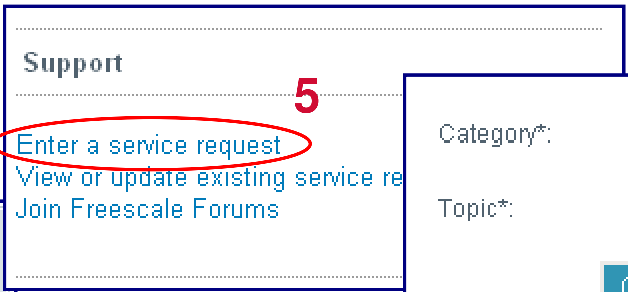
Step 1: Login Information

Freescale uses your email as your user id. Please enter your email address and choose a password.

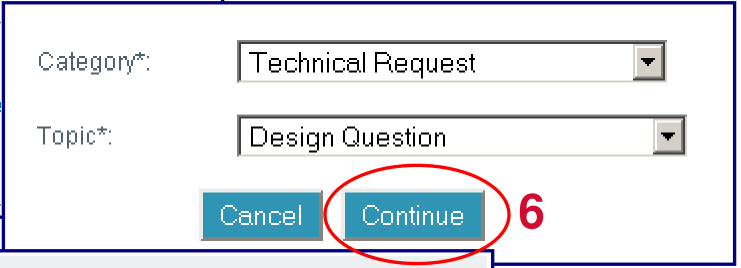
* indicates required information

* Email (Freescale ID) :	
* Password :	Password Guidelines
* Confirm Password :	

3

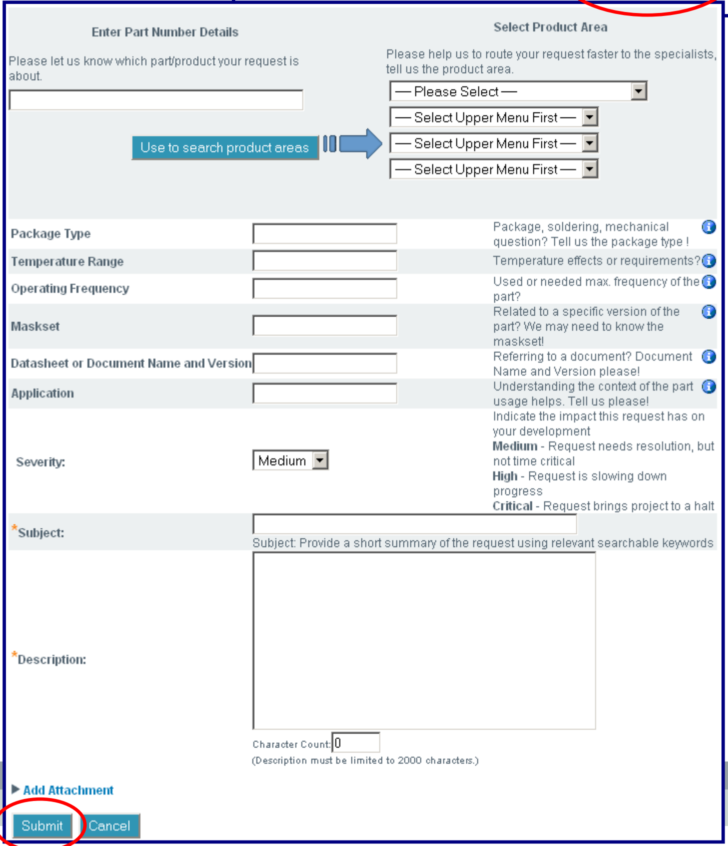


5



Category*:	Technical Request
Topic*:	Design Question

6


Enter Part Number Details

Please let us know which part/product your request is about.

Select Product Area

Please help us to route your request faster to the specialists, tell us the product area.

Use to search product areas
— Please Select —
— Select Upper Menu First —
— Select Upper Menu First —
— Select Upper Menu First —

Package Type		Package, soldering, mechanical question? Tell us the package type !
Temperature Range		Temperature effects or requirements?
Operating Frequency		Used or needed max. frequency of the part?
Maskset		Related to a specific version of the part? We may need to know the maskset!
Datasheet or Document Name and Version		Referring to a document? Document Name and Version please!
Application		Understanding the context of the part usage helps. Tell us please!
Severity:	Medium	Indicate the impact this request has on your development **Medium** - Request needs resolution, but not time critical **High** - Request is slowing down progress **Critical** - Request brings project to a halt
*Subject:	Subject: Provide a short summary of the request using relevant searchable keywords	
*Description:	Character Count: 0 (Description must be limited to 2000 characters.)	

7 Enter your request details here!

8

Email: support@freescale.com

Freescale Technology Forum

Collaboration. Innovation. Inspiration.

www.freescale.com/FTF

FTF Americas

June 22-25, 2009 | Orlando, FL

Registration Opens March 16

Q & A