MESSAGE ⊗

Gives the messages used by the ISR, usable only for ISR 2.

COUNTER objects attributes

MINCYCLE

Set the minimum number of ticks which separate two alarm expirations. Default to 1.

TICKSPERBASE

Set the prescaler of the counter. Default to 1.

MAXALLOWEDVALUE

Set the maximum value of the counter. Default to 32767.

ALARM objects attributes

COUNTER

Set the counter used by the alarm.

AUTOSTART

Specifies if an alarm is started or not when the OS is started.

FALSE	the alarm is not started
TRUE	the alarm is started

TRUE ▷ APPMODE ⊗

≥ 1 APPMODE sub-attribute shall be set. The alarm is AUTOSTART if the OS is started in one of these APPMODE.

ACTION

Specifies which action to do when the alarm expire.

ACTIVATETASK	activate a task.
SETEVENT	set an event to a task
ALARMCALLBACK	call a function

ACTIVATETASK ▷ TASK

Specifies which task to activate.

SETEVENT ▷ TASK

Specifies which task is the target of the event.

SETEVENT ▷ EVENT

Specifies the event.

ALARMCALLBACK ▷ ALARMCALLBACKNAME

Specifies a string with the name of the function to call.

EVENT objects attribute

MASK

Set the event mask of the event.

AUTO	the actual value is computed by the OIL compiler
$[0..2^{32}-1]$	the actual value is set to this value

RESOURCE objects attribute

RESOURCEPROPERTY

Set the kind of resource.

STANDARD	the resource is a standard one te be used with `GetResource` and `ReleaseResource`
INTERNAL	the resource is an internal one

Basic Oil QRDC

Jean-Luc Béchennec – LS2N

v1.0 – September 2018

The OIL standard defines standard attributes but allows implementation-specific attributes. The latter are suffixed by a †. Attribute values in a $\boxed{\text{box}}$ are the default values. If none of the attribute values are boxed no default value exists. Attributes suffixed by a ⊗ may appear 0, 1 or many times.

OS object attributes

STATUS

Set the error processing level of services. See the Osek QRDC. Possible values are:

STANDARD	does not check object ids, caller identity and state, alarm bounds and resources ownership.
EXTENDED	checks all possible errors.

STARTUPHOOK

Switch the call to the startup hook in StartOS.

$\boxed{\text{FALSE}}$	the startup hook is not called
TRUE	the startup hook is called

SHUTDOWNHOOK

Switch the call to the shutdown hook in ShutdownOS.

$\boxed{\text{FALSE}}$	the shutdown hook is not called
TRUE	the shutdown hook is called

ERRORHOOK

Switch the call to the error hook when an error occurs in a service or while processing an alarm.

$\boxed{\text{FALSE}}$	the error hook is not called
TRUE	the error hook is called

PRETASKHOOK

Switch the call to the pre task hook when a context switch is about to occur.

$\boxed{\text{FALSE}}$	the pre task hook is not called
TRUE	the pre task hook is called

POSTTASKHOOK

Switch the call to the post task hook when a context switch is about to occur.

$\boxed{\text{FALSE}}$	the post task hook is not called
TRUE	the post task hook is called

USEGETSERVICEID

Enable the macro allowing to access in the error hook the identifier of the service in which the error occurred.

$\boxed{\text{FALSE}}$	the macro to get the identifier of the service is disabled
TRUE	the macro to get the identifier of the service is enabled

USEPARAMETERACCESS

Enable the macros allowing to access in the error hook the parameters of the service in which the error occurred.

$\boxed{\text{FALSE}}$	the macro to get the parameters of the service is disabled
TRUE	the macro to get the parameters of the service is enabled

USERESSCHEDULER

Enable the `res_scheduler` resource. This resource has a priority higher than the highest priority task and lower than the lowest priority ISR2.

FALSE	the `res_scheduler` resource is disabled
$\boxed{\text{TRUE}}$	the `res_scheduler` resource is enabled

BUILD †

Enable the generation of a build script.

$\boxed{\text{FALSE}}$	no build script is generated
TRUE	a build script is generated

TRUE ▷ APP_SRC ⊗

Gives as a string a C source file name where the application code is located.

TRUE ▷ APP_CPPSRC ⊗

Gives as a string a C++ source file name where the application code is located.

TRUE ▷ APP_NAME

The output binary executable file name.

TRUE ▷ TRAMPOLINE_BASE_PATH

The path to the trampoline directory.

APPMODE objects attribute

DEFAULT

If > 1 APPMODE are defined, exactly one of the APPMODE shall have its DEFAULT attribute set to TRUE

$\boxed{\text{FALSE}}$	the APPMODE is not the default one
TRUE	the APPMODE is the default one

TASK objects attributes

AUTOSTART

Specifies if a task is in the `READY` state or not when the OS is started.

FALSE	the task is in the `SUSPENDED` state
TRUE	the task is in the `READY` state

TRUE ▷ APPMODE ⊗

≥ 1 APPMODE sub-attribute shall be set. The task is AUTOSTART if the OS is started in one of these APPMODE.

PRIORITY

Specifies the priority of the task. PRIORITY range from 0 (the lowest) to $2^{32} - 1$ (the highest).

ACTIVATION

Specifies the number of jobs that can be recorded. ACTIVATION ranges from 1 to $2^{32} - 1$.

SCHEDULE

Specifies if the task is preemptable or not.

FULL	the task is preemptable
NONE	the task is not preemptable

EVENT ⊗

Gives the events used by the task. A task having no EVENT attribute is a basic task. A task having ≥ 1 EVENT attribut is an extended task.

RESOURCE ⊗

Gives the resources used by the task.

MESSAGE ⊗

Gives the messages used by the task.

ISR objects attributes

CATEGORY

Set the category of the ISR.

1	the ISR is an ISR 1
2	the ISR is an ISR 2

PRIORITY

Specifies the priority of the ISR. PRIORITY range from 0 (the lowest) to $2^{32} - 1$ (the highest). Even a priority lower than the highest priority task is given, the OIL compiler compute an actual priority greater than the highest priority task.

SOURCE

Specifies the hardware interrupt source. Possible values depend on the hardware platform.

RESOURCE ⊗

Gives the resources used by the ISR, usable only for ISR 2.