NOTES

Not callable from ISR.

Will not cause a preempt.

3.3.2 SM_IDENT

NAME

sm_ident – "Obtain the id of a Semaphore"

SYNOPSIS

```
#include <semaphore.h>
uint sm_ident ( name, node, &smid )

        uint name;   /* semaphore name  */
        uint node;   /* node identifier */
                     /* 0 indicates any node */
        uint smid;   /* semaphore id - returned by this call */
```

DESCRIPTION

The *sm_ident* directive allows a task to identify a previously created semaphore by name and receive the *smid* to use in *sm_p*, *sm_v* and *sm_delete* directives for this semaphore.

If the semaphore name is not unique, the *smid* returned may not correspond to the semaphore named in this call.

The semaphore may exist on the local processor or any remote processor in a multiprocessor configuration, as long as the semaphore was created with the GLOBAL flags value set (see *sm_create*). If the semaphore name is not unique within the multiprocessor configuration, a non-sero node identifier must be specified in the *node* field.

RETURN VALUE

If *sm_ident* succeeds, the *smid* will be filled in, and 0 is returned.

If *sm_ident* does not succeed, an error code is returned.

ERROR CONDITIONS

Named semaphore does not exist.

Invalid node identifier.

NOTES

Can be called from within an ISR.

3.3.2 SM_DELETE

NAME

sm_delete – "Delete Semaphore"

SYNOPSIS

```
#include <semaphore.h>
uint sm_delete ( smid )

        uint smid;    /* semaphore id as returned by sm_create or sm_ident */
```

DESCRIPTION

The semaphore identified by the *smid* is deleted from the system.

If tasks are waiting for the semaphore when the semaphore is deleted, each is made ready and given a return code indicating a deleted semaphore.

The semaphore must exist on the local processor. If the semaphore was created with the GLOBAL flags value set in a multiprocessor configuration, a notification will be sent to all processors in the system, so the *smid* can be deleted from the global resource table.

The requester does not have to be the creator of the semaphore. Any task knowing the *smid* can delete it.

RETURN VALUE

If *sm_delete* successfully deleted the semaphore, 0 is returned.

If the semaphore was not successfully deleted, an error code is returned.

ERROR CONDITIONS

Invalid *smid*.

Semaphore not created from local node.

NOTES

Not callable from ISR.

May cause a preempt if a task waiting for the semaphore has a higher priority than the running task, and the preempt mode is in effect. A preempt will not occur if all tasks waiting for the semaphore exist on a remote processor in a multiprocessor configuration.

3.3.4 SM_P

NAME

sm_p -- "Access Semaphore"

SYNOPSIS

```
#include <semaphore.h>
uint sm_p ( smid, flags, timeout )


        uint smid;      /* semaphore id as returned by sm_create or sm_ident  */
        uint flags;     /* wait option */
        uint timeout;   /* number of ticks to wait */
                        /* 0 indicates wait forever */
```

The flags field values are:

NOWAIT	set	return immediately with error if semaphore count is negative
	clear	wait for resource

DESCRIPTION

If the **NOWAIT** flags value is clear, the current semaphore count of the semaphore identified by the *smid* is decremented by one. If the count is zero or positive, the requesting task continues execution, returning without error. If the count is negative, the requesting task must wait for access to the resource, and is put on a waiting list.

If the **NOWAIT** flags value is set, and the count is negative, an error is returned. If the count is zero or positive, zero is returned.

The semaphore identified by the *smid* may exist on the local processor or any remote processor in a multiprocessor configuration, as long as the semaphore was created with the **GLOBAL** flags value set (see *sm_create*).

When *sm_p* is called from an ISR, the no-wait option is forced by the executive.

RETURN VALUE

If *sm_p* succeeded, then 0 is returned.

If the call was not successful, an error code is returned.

ERROR CONDITIONS

Invalid *smid*.

Timeout (if wait and timeout is selected).

The semaphore count is negative (if no wait is selected).

Semaphore deleted.

ISR cannot reference remote node.

NOTES

Can be called from within an ISR, except when the semaphore was not created on the local node. The no-wait option is forced by the executive.

The running task will be blocked if the count is negative.