

osal-headers

User Manual



MICROEJ[®]

Reference:	TLT-XXX-MAN-osal-headers-osal-headers
Version:	1.0.0
Revision:	XXX

Confidentiality & Intellectual Property

All rights reserved. Information, technical data and tutorials contained in this document are confidential and proprietary under copyright Law of Industrial Smart Software Technology (IS2T S.A.) operating under the brand name MicroEJ®. Without written permission from IS2T S.A., *copying or sending parts of the document or the entire document by any means to third parties is not permitted*. Granted authorizations for using parts of the document or the entire document do not mean IS2T S.A. gives public full access rights.

The information contained herein is not warranted to be error-free. IS2T® and MicroEJ® and all relative logos are trademarks or registered trademarks of IS2T S.A. in France and other Countries.

Java™ is Sun Microsystems' trademark for a technology for developing application software and deploying it in cross-platform, networked environments. When it is used in this documentation without adding the ™ symbol, it includes implementations of the technology by companies other than Sun.

Java™, all Java-based marks and all related logos are trademarks or registered trademarks of Sun Microsystems Inc, in the United States and other Countries.

Other trademarks are proprietary of their authors.

Table of Contents

1. File Documentation	1
1.1. osal.h File Reference	1
1.1.1. Macros	1
1.1.2. Enumerations	1
1.1.3. Typedefs	1
1.1.4. Functions	2

Chapter 1. File Documentation

1.1. osal.h File Reference

```
#include <stdint.h>
```

```
#include "osal_portmacro.h"
```

1.1.1. Macros

- `#define OSAL_INFINITE_TIME 0xFFFFFFFF`

define an infinite time

1.1.2. Enumerations

- `enum OSAL_status_t {
 OSAL_OK,
 OSAL_ERROR,
 OSAL_NOMEM,
 OSAL_WRONG_ARGS,
 OSAL_NOT_IMPLEMENTED,
 OSAL_NOT_SUPPORTED
}`

return code list

1.1.3. Typedefs

- `typedef void *(* OSAL_task_entry_point_t`
task function entry point
- `typedef void * OSAL_task_handle_t`
OS task handle.
- `typedef void * OSAL_queue_handle_t`
OS queue handle.
- `typedef void * OSAL_counter_semaphore_handle_t`
OS counter semaphore handle.

- typedef void * OSAL_binary_semaphore_handle_t

OS binary semaphore handle.

- typedef void * OSAL_mutex_handle_t

OS mutex handle.

1.1.4. Functions

- OSAL_status_t OSAL_task_create (OSAL_task_entry_point_t entry_point, uint8_t * name, OSAL_task_stack_t stack, int32_t priority, void * parameters, OSAL_task_handle_t * handle)

Create an OS task and start it.

- OSAL_status_t OSAL_task_delete (OSAL_task_handle_t * handle)

Delete an OS task and start it.

- OSAL_status_t OSAL_task_get_current (OSAL_task_handle_t * handle)

Get the handle of the current OS task.

- OSAL_status_t OSAL_queue_create (uint8_t * name, OSAL_queue_handle_t * handle, OSAL_queue_t queue)

Create an OS queue with a predefined queue size.

- OSAL_status_t OSAL_queue_delete (OSAL_queue_handle_t * handle)

Delete an OS queue.

- OSAL_status_t OSAL_queue_post (OSAL_queue_handle_t * handle, void * msg)

Post a message in an OS queue.

- OSAL_status_t OSAL_queue_fetch (OSAL_queue_handle_t * handle, void ** msg, uint32_t timeout)

Fetch a message from an OS queue. Blocks until a message arrived or a timeout occurred.

- OSAL_status_t OSAL_counter_semaphore_create (uint8_t * name, uint32_t initial_count, uint32_t max_count, OSAL_counter_semaphore_handle_t * handle)

Create an OS counter semaphore with a semaphore count initial value.

- OSAL_status_t OSAL_counter_semaphore_delete (OSAL_counter_semaphore_handle_t * handle)

Delete an OS counter semaphore.

- OSAL_status_t OSAL_counter_semaphore_take (OSAL_counter_semaphore_handle_t * handle, uint32_t timeout)

Take operation on OS counter semaphore. Block the current task until counter semaphore become available or timeout occurred. Decrease the counter semaphore count value by 1 and block the current task if count value equals to 0.

- OSAL_status_t OSAL_counter_semaphore_give (OSAL_counter_semaphore_handle_t * handle)

Give operation on OS counter semaphore. Increase the counter semaphore count value by 1 and unblock the current task if count value. equals to 0.

- OSAL_status_t OSAL_binary_semaphore_create (uint8_t * name, uint32_t initial_count, OSAL_binary_semaphore_handle_t * handle)

Create an OS binary semaphore with a semaphore count initial value (0 or 1).

- OSAL_status_t OSAL_binary_semaphore_delete (OSAL_binary_semaphore_handle_t * handle)

Delete an OS binary semaphore.

- OSAL_status_t OSAL_binary_semaphore_take (OSAL_binary_semaphore_handle_t * handle, uint32_t timeout)

Take operation on OS binary semaphore. Block the current task until binary semaphore become available or timeout occurred. Decrease the binary semaphore count value by 1 and block the current task if count value equals to 0.

- OSAL_status_t OSAL_binary_semaphore_give (OSAL_binary_semaphore_handle_t * handle)

Give operation on OS binary semaphore. Increase the binary semaphore count value by 1 and unblock the current task if count value. equals to 0.

- OSAL_status_t OSAL_mutex_create (uint8_t * name, OSAL_mutex_handle_t * handle)

Create an OS mutex.

- OSAL_status_t OSAL_mutex_delete (OSAL_mutex_handle_t * handle)

Delete an OS mutex.

- OSAL_status_t OSAL_mutex_take (OSAL_mutex_handle_t * handle, uint32_t timeout)

Take operation on OS mutex.

- OSAL_status_t OSAL_mutex_give (OSAL_mutex_handle_t * handle)

Give operation on OS mutex.

- OSAL_status_t OSAL_disable_context_switching (void)

Disable the OS scheduler context switching. Prevent the OS from scheduling the current thread calling #OSAL_disable_context_switching while the OS scheduling is already disable has an undefined behavior. This method may be called from an interrupt.

- OSAL_status_t OSAL_enable_context_switching (void)

Reenable the OS scheduling that was disabled by #OSAL_disable_context_switching. This method may be called from an interrupt.

- OSAL_status_t OSAL_sleep (uint32_t milliseconds)

Asleep the current task during specified number of milliseconds.

Detailed Description

OS Abstraction Layer API.

Author: . MicroEJ Developer Team

Version: . 1.0.0

Date: . 12 November 2020

Definition in file /home/is2t/workspace/M0124_CCO-OSAL/bsp-osal/target~/ccomponentWorking/bsp/util/inc/osal.h