osal-FreeRTOS

User Manual

Reference:	TLT-XXX-MAN-osal-FreeRTOS-osal-FreeRTOS
Version:	0.2.2
Revision:	XXX

Table of Contents

Chapter 1. File Documentation

1.1. osal_portmacro.h File Reference

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

1.1.1. Macros

- #define OSAL_task_stack_declare OSAL_task_stack_t _name = _size

1.1.2. Typedefs

- typedef int32_t OSAL_task_stack_t

  OS task stack.

Detailed Description

OS Abstraction Layer FreeRTOS port macro.

Author: . MicroEJ Developer Team

Version: . 0.2.2

Date: . 8 December 2020

Definition in file /home/is2t/workspace/M0124_CCO-OSAL_maintenance_M0124BSPF-169_osal_FreeRTOS_0.2.1/bsp-osal-FreeRTOS/target~/ccomponentWorking/bsp/util/inc/osal_portmacro.h

1.2. osal_FreeRTOS.c File Reference

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

```
#include <string.h>
```

```
#include "osal.h"
```

```
#include "FreeRTOS.h"
```

```
#include "task.h"
```

```
#include "semphr.h"
```

1.2.1. Functions

- static TickType_t OSAL_FreeRTOS_convert_time_to_tick (uint32_t milliseconds)
- 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_queue_create (uint8_t * name, uint32_t size, OSAL_queue_handle_t * handle)

  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 FreeRTOS implementation.

Author: . MicroEJ Developer Team

Version: . 0.2.2

Date: . 8 December 2020

Definition in file /home/is2t/workspace/M0124_CCO-OSAL_maintenance_M0124BSPF-169_osal_FreeRTOS_0.2.1/bsp-osal-FreeRTOS/target~/ccomponentWorking/bsp/util/src/osal_FreeRTOS.c