



Quality RTOS & Embedded Software
[About](#) [Contact](#) [Support](#) [FAQ](#)



SAFERTOS®
Safety certified RTOS from a safety systems company.



[Quick Start](#) | [Supported MCUs](#) | [Books & Kits](#) | [Trace Tools](#) | [Ecosystem](#) | [TCP & FAT](#) | [Training](#) | [Email List](#)  

Home

[FreeRTOS Books and Manuals](#)

FreeRTOS

- About FreeRTOS
- Features / Getting Started...
- More Advanced...
- Demo Projects
- Supported Devices & Demos

API Reference

- PDF Reference Manual
- Task Creation
- Task Control
- Task Utilities
- RTOS Kernel Control
- Direct To Task Notifications
- FreeRTOS-MPU Specific
- Queues
- Queue Sets
- Semaphore / Mutexes
- Software Timers
- Event Groups (or 'flags')
 - [xEventGroupCreate\(\)](#)
 - [vEventGroupDelete\(\)](#)
 - [xEventGroupWaitBits\(\)](#)
 - [xEventGroupSetBits\(\)](#)
 - [xEventGroupSetBitsFromISR\(\)](#)
 - [xEventGroupClearBits\(\)](#)
 - [xEventGroupClearBitsFromISR\(\)](#)
 - [xEventGroupGetBits\(\)](#)
 - [xEventGroupGetBitsFromISR\(\)](#)
 - [xEventGroupSync\(\)](#)
- Co-routines
- Contact, Support, Advertising

FreeRTOS Interactive!

Quick Start Guide

Download Source

FreeRTOS+ Lab Projects

FreeRTOS+TCP:
Thread safe TCP/IP stack

FreeRTOS+FAT:
Thread aware file system

FreeRTOS+ Ecosystem

Internet of Things:
Innovative complete solution

Fail Safe File System:
Ensures data integrity

InterNiche TCP/IP:
Low cost pre-ported libraries

FreeRTOS BSPs:
3rd party driver packages

FAT SL File System:
Super lean FAT FS

UDP/IP:
Thread aware UDP stack

Trace & Visualisation:
Tracealyzer for FreeRTOS

CLI:
Command line interface

WolfSSL SSL / TLS:
Networking security protocols

Safety:
TUV certified RTOS

RTOS Training:
Delivered online or on-site

IO:
[read\(\)](#), [write\(\)](#), [ioctl\(\)](#) interface

xEventGroupSetBitsFromISR()

[Event Group API]

Available From FreeRTOS V8.0.0

event_groups.h

```
 BaseType_t xEventGroupSetBitsFromISR(
     EventGroupHandle_t xEventGroup,
     const EventBits_t uxBitsToSet,
     BaseType_t *pxHigherPriorityTaskWoken );
```

Set bits (flags) within an RTOS [event group](#). A version of [xEventGroupSetBits\(\)](#) that can be called from an interrupt service routine (ISR).

Setting bits in an event group will automatically unblock tasks that are blocked waiting for the bits.

Setting bits in an event group is not a deterministic operation because there are an unknown number of tasks that may be waiting for the bit or bits being set. FreeRTOS does not allow non-deterministic operations to be performed in interrupts or from critical sections. Therefore [xEventGroupSetBitsFromISR\(\)](#) sends a message to the RTOS daemon task to have the set operation performed in the context of the daemon task - where a scheduler lock is used in place of a critical section.

NOTE: As mentioned in the paragraph above, setting bits from an ISR will defer the set operation to the RTOS daemon task (also known as the timer service task). The RTOS daemon task is scheduled according to its priority, just like any other RTOS task. Therefore, if it is essential the set operation completes immediately (before a task created by the application executes) then the priority of the RTOS daemon task must be higher than the priority of any application task that uses the event group. The priority of the RTOS daemon task is set by the [configTIMER_TASK_PRIORITY](#) definition in [FreeRTOSConfig.h](#).

`INCLUDE_xEventGroupSetBitsFromISR`, `configUSE_TIMERS` and `INCLUDE_xTimerPendFunctionCall` must all be set to 1 in [FreeRTOSConfig.h](#) for the [xEventGroupSetBitsFromISR\(\)](#) function to be available.

The RTOS source file `FreeRTOS/source/event_groups.c` must be included in the build for the [xEventGroupSetBitsFromISR\(\)](#) function to be available.

Parameters:

<i>xEventGroup</i>	The event group in which the bits are to be set. The event group must have previously been created using a call to xEventGroupCreate() .
<i>uxBitsToSet</i>	A bitwise value that indicates the bit or bits to set. For example, set <i>uxBitsToSet</i> to 0x08 to set only bit 3. Set <i>uxBitsToSet</i> to 0x09 to set bit 3 and bit 0.
<i>pxHigherPriorityTaskWoken</i>	As mentioned above, calling this function will result in a message being sent to the RTOS daemon task. If the priority of the daemon task is higher than the priority of the currently running task (the task the interrupt interrupted) then * <i>pxHigherPriorityTaskWoken</i> will be set to pdTRUE by xEventGroupSetBitsFromISR() , indicating that a context switch should be requested before the interrupt exits. For that reason * <i>pxHigherPriorityTaskWoken</i> must be initialised to pdFALSE. See the example code below.

Returns:

If the message was sent to the RTOS daemon task then pdPASS is returned, otherwise pdFAIL is returned. pdFAIL will be returned if the [timer service queue](#) was full.

Example usage:

```
#define BIT_0    ( 1 << 0 )
#define BIT_4    ( 1 << 4 )

/* An event group which it is assumed has already been created by a call to
xEventGroupCreate(). */
EventGroupHandle_t xEventGroup;

void anInterruptHandler( void )
{
    BaseType_t xHigherPriorityTaskWoken, xResult;

    /* xHigherPriorityTaskWoken must be initialised to pdFALSE. */
    xHigherPriorityTaskWoken = pdFALSE;
```

Latest News:

FreeRTOS V9.0.0rc1 is now available for [download](#) and comment.

Buildable Examples

FreeRTOS+TCP



2GB

FreeRTOS+FAT

Try Them Now

Sponsored Links

↓ Now With No Code Size Limit! ↓

TrueSTUDIO



The best **UNLIMITED** **FREE** ARM® development on the planet.

DOWNLOAD NOW WITHOUT REGISTRATION

NO CODE-SIZE LIMITATION



↑ Free Download Without Registering ↑

USB TCP/IP File Systems



Supplied as **integrated** and **functioning FreeRTOS projects**

from the **Official FreeRTOS Partner**

 **Reliance Edge™**

The Tiny, Open Source, Power Fail-safe File System for FreeRTOS

Download FreeRTOS Project

```
/* Set bit 0 and bit 4 in xEventGroup. */
xResult = xEventGroupSetBitsFromISR(
    xEventGroup, /* The event group being updated. */
    BIT_0 | BIT_4 /* The bits being set. */
    &xHigherPriorityTaskWoken );

/* Was the message posted successfully? */
if( xResult != pdFAIL )
{
    /* If xHigherPriorityTaskWoken is now set to pdTRUE then a context
    switch should be requested. The macro used is port specific and will
    be either portYIELD_FROM_ISR() or portEND_SWITCHING_ISR() - refer to
    the documentation page for the port being used. */
    portYIELD_FROM_ISR( xHigherPriorityTaskWoken );
}
```

[\[Back to the top \]](#) [\[About FreeRTOS \]](#) [\[Sitemap \]](#) [\[Report an error on this page \]](#)

Copyright (C) 2004-2010 Richard Barry. Copyright (C) 2010-2016 Real Time Engineers Ltd.
Any and all data, files, source code, html content and documentation included in the FreeRTOS™ distribution or available on this site are the
exclusive property of Real Time Engineers Ltd.. See the files license.txt (included in the distribution) and this [copyright notice](#) for more information.
FreeRTOS™ and FreeRTOS.org™ are trade marks of Real Time Engineers Ltd.

