



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

FreeRTOS Tutorial Books and Reference Manuals  
Become an expert, while supporting the FreeRTOS project

[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  
xSemaphoreCreateBinary()  
vSemaphoreCreateBinary()  
xSemaphoreCreateCounting()  
xSemaphoreCreateMutex()  
xSemaphoreCreateRecursiveMutex()  
vSemaphoreDelete()  
xSemaphoreGetMutexHolder()  
xSemaphoreTake()  
xSemaphoreTakeFromISR()  
xSemaphoreTakeRecursive()  
**xSemaphoreGive()**  
xSemaphoreGiveRecursive()  
xSemaphoreGiveFromISR()  
Software Timers  
Event Groups (or 'flags')  
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:  
3<sup>rd</sup> 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

## xSemaphoreGive

### [Semaphores]

semphr. h

```
xSemaphoreGive( SemaphoreHandle_t xSemaphore )
```

Macro to release a semaphore. The semaphore must have previously been created with a call to xSemaphoreCreateBinary(), xSemaphoreCreateMutex() or xSemaphoreCreateCounting(), and obtained using xSemaphoreTake().

This must not be used from an ISR. See xSemaphoreGiveFromISR() for an alternative which can be used from an ISR.

This macro must also not be used on semaphores created using xSemaphoreCreateRecursiveMutex().

**Parameters:**

*xSemaphore* A handle to the semaphore being released. This is the handle returned when the semaphore was created.

**Returns:**

pdTRUE if the semaphore was released. pdFALSE if an error occurred. Semaphores are implemented using queues. An error can occur if there is no space on the queue to post a message - indicating that the semaphore was not first obtained correctly.

**Example usage:**

```
SemaphoreHandle_t xSemaphore = NULL;
void vATask( void * pvParameters )
{
    // Create the semaphore to guard a shared resource. As we are using
    // the semaphore for mutual exclusion we create a mutex semaphore
    // rather than a binary semaphore.
    xSemaphore = xSemaphoreCreateMutex();

    if( xSemaphore != NULL )
    {
        if( xSemaphoreGive( xSemaphore ) != pdTRUE )
        {
            // We would expect this call to fail because we cannot give
            // a semaphore without first "taking" it!
        }

        // Obtain the semaphore - don't block if the semaphore is not
        // immediately available.
        if( xSemaphoreTake( xSemaphore, ( TickType_t ) 0 ) )
        {
            // We now have the semaphore and can access the shared resource.
            // ...
            // We have finished accessing the shared resource so can free the
            // semaphore.
            if( xSemaphoreGive( xSemaphore ) != pdTRUE )
            {
                // We would not expect this call to fail because we must have
                // obtained the semaphore to get here.
            }
        }
    }
}
```

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

Latest News:

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

Buildable Examples

FreeRTOS+TCP



FreeRTOS+FAT



Try Them Now

Sponsored Links

Now With No Code Size Limit!



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

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.

Google™ Custom Search Search

