

**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](#)  

**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!**

**TrueSTUDIO**  
  
The best **UNLIMITED**  
**FREE** ARM® development  
on the planet.  
  
**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  
**Now  
Reliable Things** 

- [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](#)
      - [uxQueueMessagesWaiting\(\)](#)
      - [uxQueueMessagesWaitingFromISR\(\)](#)
      - [uxQueueSpacesAvailable\(\)](#)
      - [xQueueCreate\(\)](#)
      - [vQueueDelete\(\)](#)
      - [xQueueReset\(\)](#)
      - [xQueueSend\(\)](#)
      - [xQueueSendToBack\(\)](#)
      - [xQueueSendToFront\(\)](#)
      - [xQueueReceive\(\)](#)
      - [xQueueOverwrite\(\)](#)
      - [xQueueOverwriteFromISR\(\)](#)
      - [xQueuePeek\(\)](#)
      - [xQueuePeekFromISR\(\)](#)
      - [xQueueSendFromISR\(\)](#)
      - [xQueueSendToBackFromISR\(\)](#)
      - [xQueueSendToFrontFromISR\(\)](#)
      - [xQueueReceiveFromISR\(\)](#)
      - [vQueueAddToRegistry\(\)](#)
      - [vQueueUnregisterQueue\(\)](#)
      - [xQueueIsQueueFullFromISR\(\)](#)
      - [xQueueIsQueueEmptyFromISR\(\)](#)
  - [Queue Sets](#)
  - [Semaphore / Mutexes](#)
  - [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:](#)
[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](#)

## xQueueReceive

### [Queue Management]

queue. h

```
BaseType_t xQueueReceive(
    QueueHandle_t xQueue,
    void *pvBuffer,
    TickType_t xTicksToWait
);
```

This is a macro that calls the xQueueGenericReceive() function.

Receive an item from a queue. The item is received by copy so a buffer of adequate size must be provided. The number of bytes copied into the buffer was defined when the queue was created.

This function must not be used in an interrupt service routine. See xQueueReceiveFromISR for an alternative that can.

#### Parameters:

- xQueue** The handle to the queue from which the item is to be received.
- pvBuffer** Pointer to the buffer into which the received item will be copied.
- xTicksToWait** The maximum amount of time the task should block waiting for an item to receive should the queue be empty at the time of the call. Setting xTicksToWait to 0 will cause the function to return immediately if the queue is empty. The time is defined in tick periods so the constant portTICK\_PERIOD\_MS should be used to convert to real time if this is required.

If `#INCLUDE_vTaskSuspend` is set to '1' then specifying the block time as portMAX\_DELAY will cause the task to block indefinitely (without a timeout).

#### Returns:

pdTRUE if an item was successfully received from the queue, otherwise pdFALSE.

#### Example usage:

```
struct AMessage
{
    char ucMessageID;
    char ucData[ 20 ];
} xMessage;

QueueHandle_t xQueue;

// Task to create a queue and post a value.
void vATask( void *pvParameters )
{
    struct AMessage *pxMessage;

    // Create a queue capable of containing 10 pointers to AMessage structures.
    // These should be passed by pointer as they contain a lot of data.
    xQueue = xQueueCreate( 10, sizeof( struct AMessage * ) );
    if( xQueue == 0 )
    {
        // Failed to create the queue.
    }

    // ...

    // Send a pointer to a struct AMessage object. Don't block if the
    // queue is already full.
    pxMessage = &xMessage;
    xQueueSend( xQueue, ( void * ) &pxMessage, ( TickType_t ) 0 );

    // ... Rest of task code.
}

// Task to receive from the queue.
void vADifferentTask( void *pvParameters )
{
    struct AMessage *pxRxdMessage;

    if( xQueue != 0 )
    {
        // Receive a message on the created queue. Block for 10 ticks if a
        // message is not immediately available.
        if( xQueueReceive( xQueue, &( pxRxdMessage ), ( TickType_t ) 10 ) )
        {
            // pxRxdMessage now points to the struct AMessage variable posted
            // by vATask.
        }
    }

    // ... Rest of task code.
}
```



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

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.