



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



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
 - vTaskDelay()**
 - vTaskDelayUntil()
 - uxTaskPriorityGet()
 - vTaskPrioritySet()
 - vTaskSuspend()
 - vTaskResume()
 - xTaskResumeFromISR()
 - Task Utilities
 - RTOS Kernel Control
 - Direct To Task Notifications
 - FreeRTOS-MPU Specific
 - Queues
 - 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

vTaskDelay

[Task Control]

task. h

```
void vTaskDelay( const TickType_t xTicksToDelay );
```

INCLUDE_vTaskDelay must be defined as 1 for this function to be available. See the configuration section for more information.

Delay a task for a given number of ticks. The actual time that the task remains blocked depends on the tick rate. The constant portTICK_PERIOD_MS can be used to calculate real time from the tick rate - with the resolution of one tick period.

vTaskDelay() specifies a time at which the task wishes to unblock **relative to** the time at which vTaskDelay() is called. For example, specifying a block period of 100 ticks will cause the task to unblock 100 ticks after vTaskDelay() is called. vTaskDelay() does not therefore provide a good method of controlling the frequency of a periodic task as the path taken through the code, as well as other task and interrupt activity, will effect the frequency at which vTaskDelay() gets called and therefore the time at which the task next executes. See vTaskDelayUntil() for an alternative API function designed to facilitate fixed frequency execution. It does this by specifying an absolute time (rather than a relative time) at which the calling task should unblock.

Parameters:

xTicksToDelay The amount of time, in tick periods, that the calling task should block.

Example usage:

```
void vTaskFunction( void * pvParameters )
{
    /* Block for 500ms. */
    const TickType_t xDelay = 500 / portTICK_PERIOD_MS;

    for( ;; )
    {
        /* Simply toggle the LED every 500ms, blocking between each toggle. */
        vToggleLED();
        vTaskDelay( xDelay );
    }
}
```

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

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.

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

atollic

Free Download Without Registering

USB TCP/IP File Systems



Supplied as **integrated** and **functioning** FreeRTOS projects from the **Official FreeRTOS Partner**

Google™ Custom Search

Search

1 von 2

01.03.2016 09:28

