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
taskYIELD()
taskENTER_CRITICAL()
taskEXIT_CRITICAL()
taskDISABLE_INTERRUPTS()
taskENABLE_INTERRUPTS()
vTaskStartScheduler()
vTaskEndScheduler()
vTaskSuspendAll()
xTaskResumeAll()
vTaskStepTick()
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!

vTaskStartScheduler

[RTOS Kernel Control]

task. h

```
void vTaskStartScheduler( void );
```

Starts the RTOS scheduler. After calling the RTOS kernel has control over which tasks are executed and when.

The idle task and optionally the timer daemon task are created automatically when the RTOS scheduler is started.

vTaskStartScheduler() will only return if there is insufficient RTOS heap available to create the idle or timer daemon tasks.

All the RTOS demo application projects contain examples of using vTaskStartScheduler(), normally in the main() function within main.c.

Example usage:

```
void vAFunction( void )
{
    // Tasks can be created before or after starting the RTOS
    scheduler
    xTaskCreate( vTaskCode,
                 "NAME",
                 STACK_SIZE,
                 NULL,
                 tskIDLE_PRIORITY,
                 NULL );

    // Start the real time scheduler.
    vTaskStartScheduler();

    // Will not get here unless there is insufficient RAM.
}
```

[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.

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

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! ⇓


Google™ Custom Search Search