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]


ALTERA
freescale
Alliance Member
infineon
Texas Instruments
MCU
Developer Network
CYPRESS
FUJITSU
Microsemi
atollic
IAR
SYSTEMS
KEIL
Tools by ARM
Embedded
Artists