

関数名	機能と書式
xTaskNotify xTaskNotifyIndexed	<p>ノーティフィケーション待ちのタスクの待ちを解除する イベント・フラグをセットする</p> <p>《書式》</p> <pre> BaseType_t xTaskNotify(TaskHandle_t xTaskToNotify, uint32_t ulValue, eNotifyAction eAction); または BaseType_t xTaskNotify(TaskHandle_t xTaskToNotify, UBaseType_t uxIndexToNotify, uint32_t ulValue, eNotifyAction eAction); </pre> <p>xTaskToNotify: 解除するタスクのハンドル値 uxIndexToNotify: 解除するNotification Arrayのインデックス値 ulValue : Notification Arrayの更新値 (イベント用) eAction : ノーティフィケーション後の動作指定 (本文参照) eNoAction, eSetBits, eIncrement, eSetValueWithOverwrite, eSetValueWithoutOverwrite</p> <p>戻り値: 常にpdPASS</p> <p>《使用例》</p> <pre> xTaskNotify(task3, 0, eNoaction); </pre>
xTaskNotifyGive xTaskNotifyGiveIndexed	<p>ノーティフィケーションのGive (待ち合わせの解除)</p> <p>《書式》</p> <pre> BaseType_t xTaskNotifyGive(TaskHandle_t xTaskNotify); または BaseType_t xTaskNotifyGiveIndexed(TaskHandle_t xTaskNotify, UBaseType_t uxIndexToNotify); </pre> <p>xTaskNotify: Giveするタスクのハンドル値 uxIndexToNotify: Notification Arrayのインデックス値</p> <p>戻り値: 常にpdPASS</p> <p>《使用例》</p> <pre> xTaskNotifyGive(xTask2); //インデックス値は0固定 xTaskNotifyGiveIndexed(xTask2, 0); </pre>
ulTaskNotifyTake ulTaskNotifyTakeIndexed	<p>ノーティフィケーションをTakeする (待ち合わせる)</p> <p>《書式》</p> <pre> uint32_t ulTaskNotifyTake(BaseType_t xClearCountOnExit, TickType_t xTicksToWait); または uint32_t ulTaskNotifyTakeIndexed(UBaseType_t uxIndexToWaitOn, BaseType_t xClearCountOnExit, TickType_t xTicksToWait); </pre> <p>uxIndexToWaitOn: 待つNotification Arrayのインデックス値 xClearCountOnExit: pdFALSEにするとNotify値を-1する pdTRUEにすると0にする</p> <p>xTicksToWait: 待ち時間の指定 portMAX_DELAYで永久待ち</p> <p>戻り値: -1された値か0</p> <p>《使用例》</p> <pre> ulTaskNotifyTake(pdTRUE, portMAX_DELAY) //アレイ値は0固定 ulTaskNotifyTakeIndexed(0, pdTRUE, portMAX_DELAY); </pre>
xTaskNotifyWait xTaskNotifyWaitIndexed	<p>イベント・フラグがセットされるのを待つ</p> <p>《書式》</p>

	<pre> BaseType_t xTaskNotifyWait(uint32_t ulBitsToClearOnEntry, uint32_t ulBitsToClearOnExit, uint32_t *pulNotificationValue, TickType_t xTicksToWait); または BaseType_t xTaskNotifyWait(UBaseType_t uxIndexToWaitOn, uint32_t ulBitsToClearOnEntry, uint32_t ulBitsToClearOnExit, uint32_t *pulNotificationValue, TickType_t xTicksToWait); uxIndexToWaitOn : Notification Arrayのインデックス値 ulBitsToClearOnEntry : Waitする前にNotification Valueをクリア ulBitsToClearOnExit : Wait終了時にNotification Valueをクリア pulNotificationValue : Wait終了時にValueをコピーする xTicksToWait : Waitする最大時間 portMAX_DELAYで永久待ち 戻り値 : pdTRUE正常にNotifyされた pdFALSE : タイムアウト 《使用例》 xTaskNotifyWaitIndexed(0, 0, ULONG_MAX, &ulNotifiedValue, portMAX_DELAY); </pre>
<pre> xTaskNotifyFromISR xTaskNotifyIndexedFromISR </pre>	<pre> 割り込み処理内からタスク・ノーティフィケーションを解除する 《書式》 BaseType_t xTaskNotifyFromISR(TaskHandle_t xTaskToNotify, uint32_t ulValue, eNotifyAction eAction, BaseType_t *pxHigerPriorityTaskWoken); または BaseType_t xTaskNotifyIndexedFromISR(TaskHandle_t xTaskToNotify, UBaseType_t uxIndexToNotify, uint32_t ulValue, eNotifyAction eAction, BaseType_t *pxHigerPriorityTaskWoken); xTaskNotify : ノーティフィケーションを解除する相手タスク・ハンドル uxIndexToNotify : 相手のNotification Arrayのインデックス値 ulValue : Notification Arrayの値 eAction : Notifyされたときの動作指定 (本文参照) eNoAction, eSetBits, eIncrement, eSetValueWithOverwrite, eSetValueWithotOverwrite pxHigerPriorityTaskWoken : 常にpdFALSEとする 戻り値 : 常にpdPASS 《使用例》 BaseType_t pxHigerPriorityTaskWoken = pdFALSE; xTaskNotifyIndexedFromISR(xHandlingTask, 0, 0, eSetBits, &pxHigerPriorityTaskWoken); </pre>
<pre> vTaskNotifyGiveFromISR vTaskNotifyGiveIndexedFromISR </pre>	<pre> 割り込み処理からタスクのノーティフィケーション待ちを解除する 《書式》 void vTaskNotifyGiveFromISR(TaskHandle_t xTaskNotify, BaseType_t *pxHigerPriorityTaskWoken); または void vTaskNotifyGiveIndexedFromISR(TaskHandle_t xTaskNotify, UBaseType_t uxIndexToNotify, BaseType_t *pxHigerPriorityTaskWoken); xTaskNotify : 相手タスクのハンドル値 uxIndexToNotify: 相手タスクのNotification Arrayのインデックス値 pxHigerPriorityTaskWoken : pdTRUEにするとレディにした高位タスクを実行する </pre>

pdFALSEにすると自タスクに戻る

《使用例》

```
BaseType_t pxHigerPriorityTaskWoken = pdFALSE;
```

```
vTaskNotifyGiveIndexedFromISR(task1, 0, *pxHigerPriorityTaskWoken);
```