

FreeRTOS for STM32

Step 3

FreeRTOS - Queue Management

Queue

fixed size data item의 한정된 수만큼 보유

'length'는 queue item의 최대 수

각 data item의 length와 size는 queue가 생성될때 설정

데이터를 queue에 보낼때 queue에 데이터를 복사(Queue by copy)

Queue의 Blocking

읽을때

- queue가 비어있다면 blocked state
- 다른 task나 interrupt에서 queue에 쓰거나, 정의한 block time이 다 지난 경우 자동으로 Ready state

쓸때

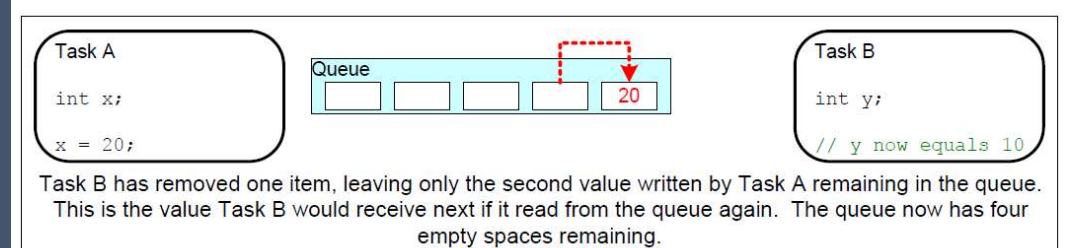
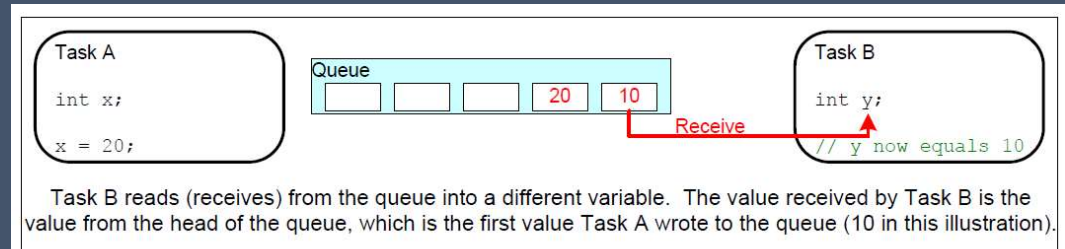
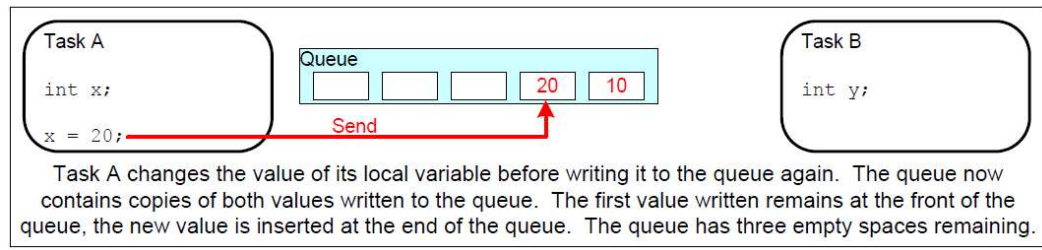
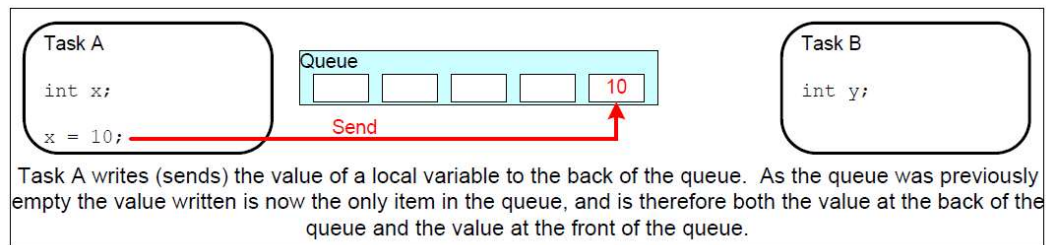
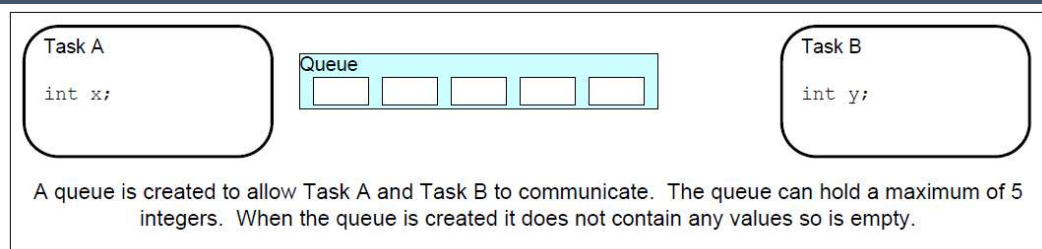
- queue가 full일때 block state
- 다른 task나 interrupt에서 queue를 읽거나, 정의한 block time이 다 지난 경우 자동으로 Ready state

Queue의 Unblocking 순서

특정 queue에 대해서 multiple reader/writer 있을때

- 우선순위가 높은 task 가 우선
- 우선순위가 같다면 기다린 시간이 가장 긴 task가 우선

FreeRTOS - Queue Management



FreeRTOS - Queue Management

Queue사용

xQueueCreate() : Queue 생성

QueueHandle_t xQueueCreate(UBaseType_t uxQueueLength, UBaseType_t uxItemSize);	
uxQueueLength	item의 최대수
uxItemSize	각 data item의 size(byte)
return value	NULL: 생성 실패 non-NULL: 생성 완료, 생성된 queue handle

xQueueSendToBack() : data를 tail에 write == **xQueueSend()** 함수

xQueueSendToFront() : data를 head에 write

BaseType_t xQueueSendToFront(QueueHandle_t xQueue, const void * pvlItemToQueue, TickType_t xTicksToWait);	
BaseType_t xQueueSendToBack(QueueHandle_t xQueue, const void * pvlItemToQueue, TickType_t xTicksToWait);	
xQueue	queue handle
pvlItemToQueue	queue 복사할 data의 pointer
xTicksToWait	task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정)
return value	pdPASS : write 성공 errQUEUE_FULL : write 실패 (queue full or xTicksToWait expired)

FreeRTOS - Queue Management

Queue사용

xQueueReceive() : data를 queue에서 읽음

BaseType_t xQueueReceive(QueueHandle_t xQueue, void * const pvBuffer, TickType_t xTicksToWait);	
xQueue	queue handle
pvBuffer	queue에서 data를 복사할 memory의 pointer
xTicksToWait	task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정)
return value	pdPASS : write 성공 errQUEUE_EMPTY : read 실패 (queue empty or xTicksToWait expired)

uxQueueMessagesWaiting() : queue에대한 item의 수를 쿼리

UBaseType_t uxQueueMessagesWaiting(QueueHandle_t xQueue);	
xQueue	queue handle
return value	queue의 item의 수

FreeRTOS - Queue Management

Example10 – Queue create, write, read

FreeRTOS - Queue Management

Queue사용

FreeRTOS는 일반적으로 하나이상의 소스로 부터 데이터를 수신하도록 설계된다.
따라서 수신 task는 전달된 데이터가 어디서 온것인지 어떻게 처리해야 하는지 알아야한다.
queue를 하나 만들어서 전달되는 데이터에 ID와 data를 전달

FreeRTOS - Queue Management

Example11 – Queue 구조체

FreeRTOS - Queue Management

Queue Set

하나이상의 queue로 부터 task가 데이터를 수신

효율이 좋은 편은 아니기 때문에 설계제약 조건상 절대적으로 필요한 경우에만 사용

configUSE_QUEUE_SETS ("FreeRTOSConfig.h") 설정 1

xQueueCreateSet() : Queue set 생성

QueueSetHandle_t xQueueCreateSet(const UBaseType_t uxEventQueueLength);	
uxEventQueueLength	queue set이 보유할 수 있는 queue handle의 최대수 queue set에는 semaphore도 포함할수 있다 .
return value	NULL : 실패 non-NULL: 성공, queue set handle 반환

FreeRTOS - Queue Management

Queue Set

xQueueAddToSet() : queue set에 queue나 semaphore를 추가

BaseType_t xQueueAddToSet(QueueSetMemberHandle_t xQueueOrSemaphore, QueueSetHandle_t xQueueSet);	
xQueueOrSemaphore	'queue set'에 더할 queue 또는 semaphore의 handle
xQueueSet	'queue set'의 handle
return value	pdPASS : 성공 pdFAIL : 실패

xQueueSelectFromSet() : queue set으로 부터 queue handle을 읽음

QueueSetMemberHandle_t xQueueSelectFromSet(QueueSetHandle_t xQueueSet, const TickType_t xTicksToWait);	
xQueueSet	'queue set'의 handle
xTicksToWait	task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정)
return value	NULL : 실패 non-NULL: 성공, 데이터를 포함하는 queue 또는 semaphore의 handle (casting)

FreeRTOS - Queue Management

Example12 – Queue Set

FreeRTOS - Queue Management

Mailbox 만들기 (길이가 1인 queue)

xQueueOverwrite() : queue안의 저장된 데이터를 변경

BaseType_t xQueueOverwrite(QueueHandle_t xQueue, const void * pvItemToQueue);	
xQueue	queue handle
pvItemToQueue	변경할 데이터
return value	pdPASS만 리턴

xQueuePeek() : queue의 head에서 데이터를 읽지만 읽은 데이터를 지우지 않는다.

BaseType_t xQueuePeek(QueueHandle_t xQueue, void * const pvBuffer, TickType_t xTicksToWait);	
xQueue	queue handle
pvBuffer	queue에서 data를 복사할 memory의 pointer
xTicksToWait	task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정)
return value	pdPASS : write 성공 errQUEUE_EMPTY : read 실패 (queue empty or xTicksToWait expired)

FreeRTOS - Queue Management

Example13 – Mailbox

FreeRTOS - Software Timer Management

Software Timer

미래에 설정된 시간 또는 주기적으로 기능 실행을 예약하는 데 사용

FreeRTOS 커널에의해서 실행 (not use hardware timer)

Software Timer는 콜백(callback)함수를 호출

FreeRTOS의 옵션으로 사용 설정은

- FreeRTOS/Source/timers.c 를 포함
- configUSE_TIMERS("FreeRTOSConfig.h") 설정 1

Software Timer Callback 함수

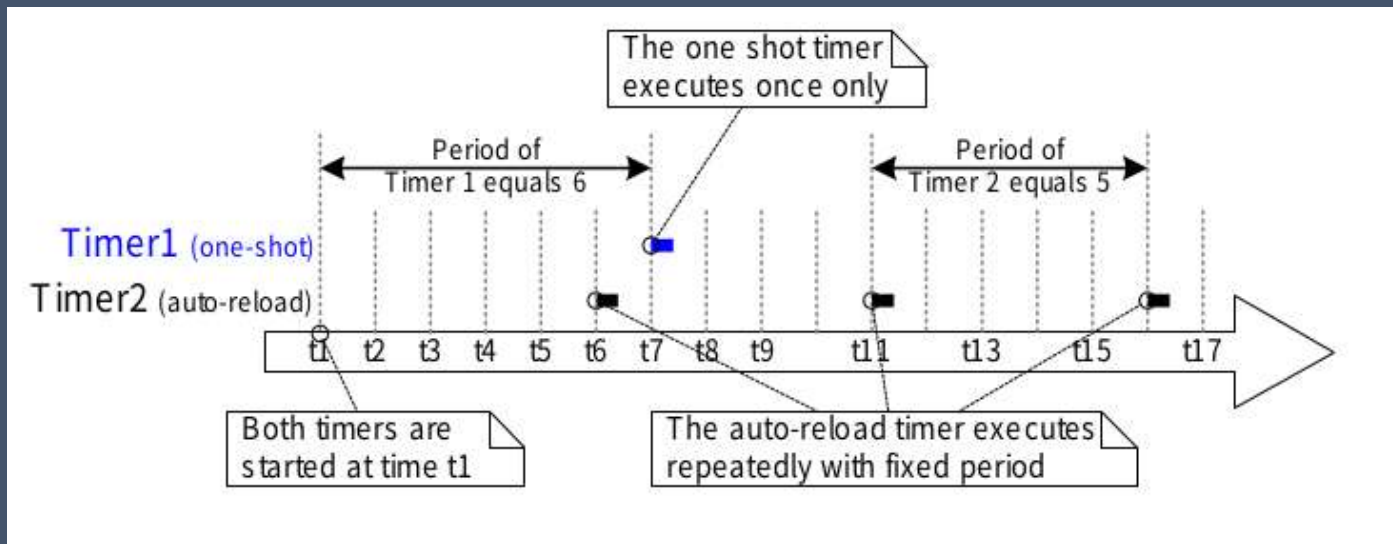
```
void ATimerCallback( TimerHandle_t xTimer );
```

FreeRTOS - Software Timer Management

Software Timer

One-shot, Auto-reload Timers

- one-shot timer : 콜백함수를 한번만 실행
- auto-reload timer: 타이머가 실행되면, 주어진 주기내에 반복적으로 실행



소프트웨어 타이머 상태

- Dormant(수면)
- Running(실행) :

FreeRTOS - Software Timer Management

Software Timer

RTOS Daemon Task

모든 Software timer callback은 RTOS daemon task의 context에서 실행
daemon task는 스케줄러가 시작되면 자동으로 생성

daemon task의 설정

- configTIMER_TASK_PRIORITY("FreeRTOSConfig.h") 우선순위
- configTIMER_TASK_STACK_DEPTH("FreeRTOSConfig.h") stack size

Timer Command Queue

Software timer 함수를 호출한 task로 부터 daemon task로 timer command queue에 명령을 보냄 (Start timer, stop timer, reset timer 등)
timer command queue는 스케줄러가 시작될때 자동으로 생성

'timer command queue의 설정

- configTIMER_QUEUE_LENGTH("FreeRTOSConfig.h") queue의 길이

FreeRTOS - Software Timer Management

Software Timer

xTimerCreate() : Software timer 생성

TimerHandle_t xTimerCreate(const char * const pcTimerName, TickType_t xTimerPeriodInTicks, UBaseType_t uxAutoReload, void * pvTimerID, TimerCallbackFunction_t pxCallbackFunction);	
pcTimerName	타이머의 이름
xTimerPeriodInTicks	tick 단위로 지정된 타이머의 기간, pdMS_TO_TICKS()로 ms 사용가능
uxAutoReload	pdTRUE : auto-reload timer pdFALSE : one-shot timer
pvTimerID	각 소프트웨어 타이머의 ID 값 이 ID는 하나이상의 소프트웨어 타이머에서 같은 콜백을 사용하는 경우 유용하게 사용된다.
pxCallbackFunction	소프트웨어 타이머의 콜백함수
return value	NULL : 실패 non-NULL : 소프트웨어 타이머의 handle

FreeRTOS - Software Timer Management

Software Timer

xTimerStart() : Software timer 시작

BaseType_t xTimerStart(TimerHandle_t xTimer, TickType_t xTicksToWait);	
xTimer	소프트웨어 타이머의 handle
xTicksToWait	<p>xTimerStart()함수는 'start a timer' 명령을 command queue를 사용해서 daemon task로 전달한다. 따라서 command queue가 full이면 block state가 된다.</p> <p>task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정)</p>
return value	<p>pdPASS : 성공 pdFALSE : 실패(시간초과)</p>

xTimerStop() : Running state안의 software timer 중지

BaseType_t xTimerStop(TimerHandle_t xTimer, TickType_t xTicksToWait);	
xTimer	소프트웨어 타이머의 handle
xTicksToWait	<p>task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정)</p>
return value	<p>pdPASS : 성공 pdFALSE : 실패(시간초과)</p>

FreeRTOS - Software Timer Management

Example14 – one-shot, auto-reload timer

FreeRTOS - Software Timer Management

Software Timer

Timer ID

각 소프트웨어 타이머는 ID를 갖는다.

ID 값은 void pointer값으로 사용자가 어떤 용도로도 활용할수 있다.

초기값은 타이머가 생성될때 할당

vTimerSetTimerID() : ID 값 변경

void vTimerSetTimerID(const TimerHandle_t xTimer, void *pvNewID);	
xTimer	소프트웨어 타이머 handle
pvNewID	변경할 timer id의 값

pvTimerGetTimerID() : ID 값 쿼리

void *pvTimerGetTimerID(TimerHandle_t xTimer);	
xTimer	소프트웨어 타이머 handle
return value	timer id의 값

FreeRTOS - Software Timer Management

Example15 – software timer ID

FreeRTOS - Software Timer Management

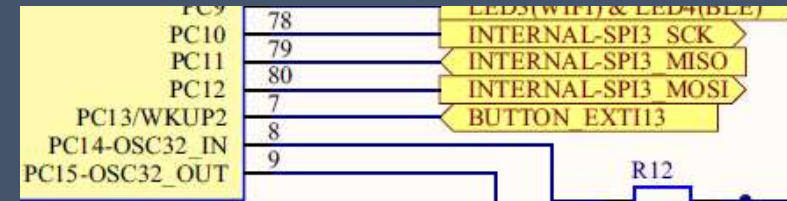
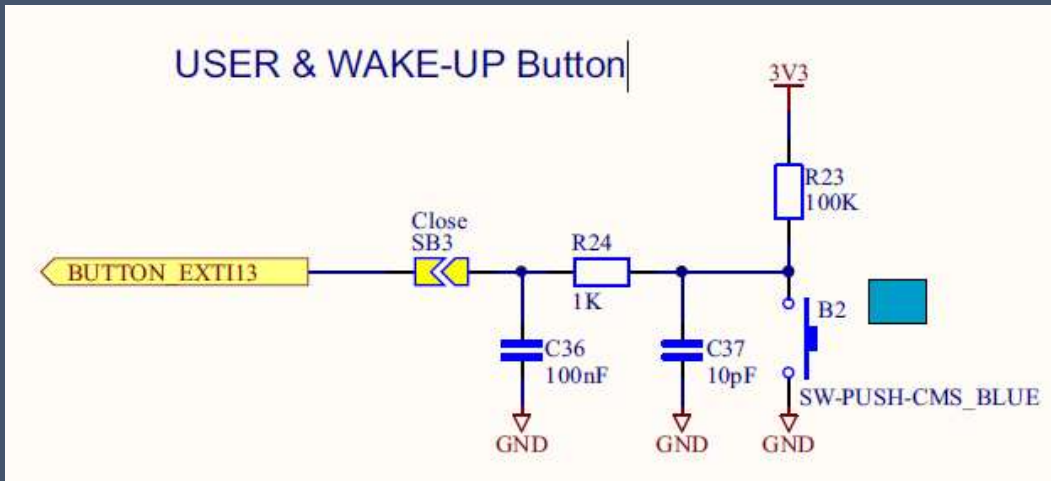
Software Timer

xTimerChangePeriod() : software timer 간격(period)을 변경

BaseType_t xTimerChangePeriod(TimerHandle_t xTimer, TickType_t xNewTimerPeriodInTicks, TickType_t xTicksToWait);	
xTimer	소프트웨어 타이머 handle
xNewTimerPeriodInTicks	tick단위의 새로운 소프트웨어 타이머의 간격, pdMS_TO_TICKS() // ms
xTicksToWait	task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정
return value	pdPASS : 성공 pdFALSE : 실패(시간초과)

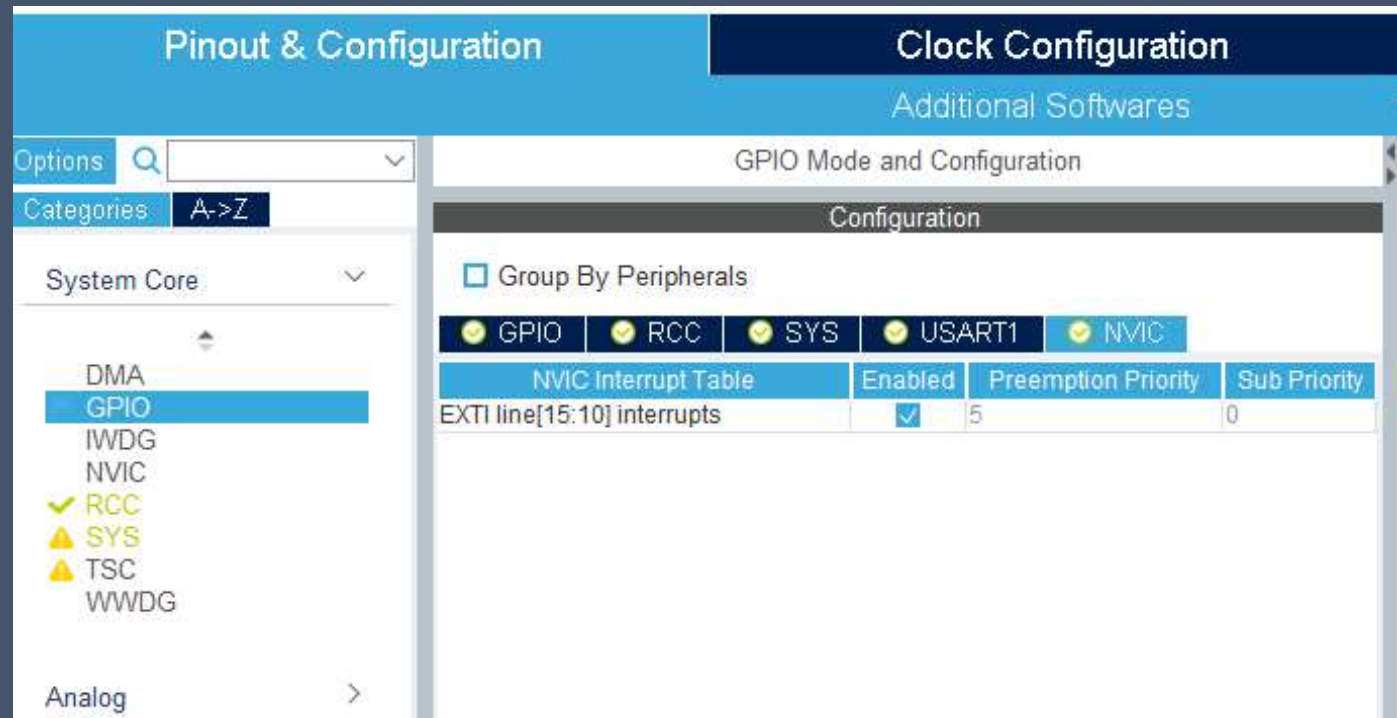
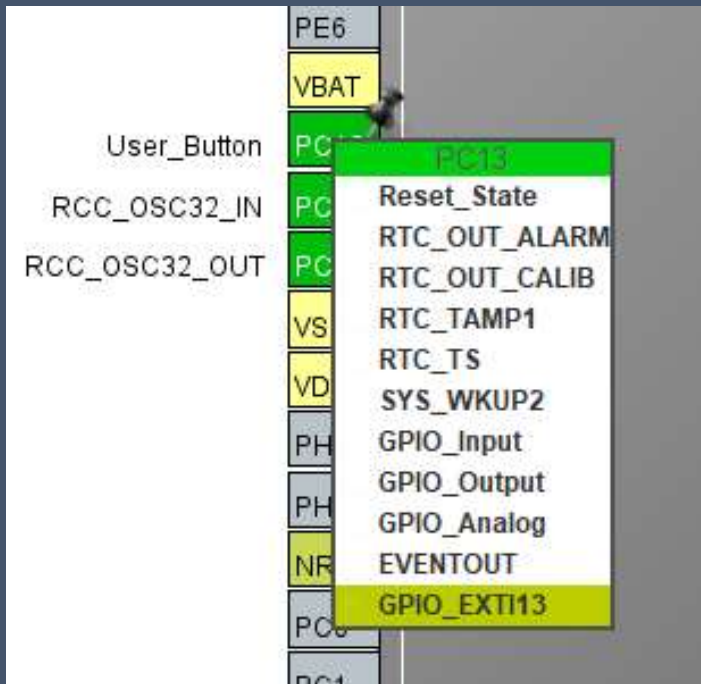
FreeRTOS - Software Timer Management

User Button Interrupt 생성



FreeRTOS - Software Timer Management

User Button Interrupt 생성



FreeRTOS - Software Timer Management

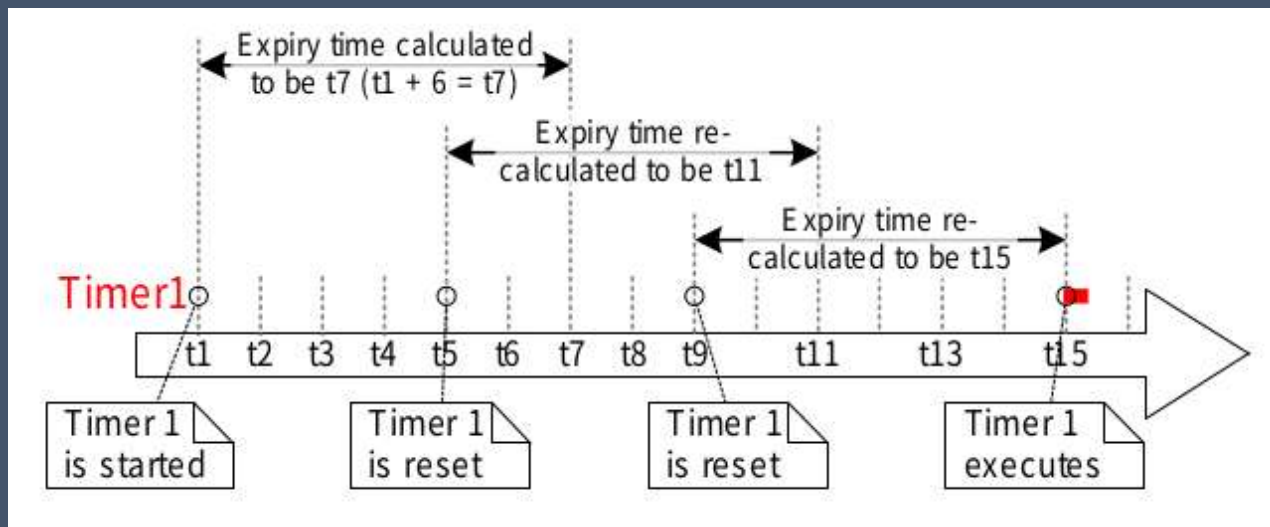
Example16 – software timer period 변경

FreeRTOS - Software Timer Management

Software Timer

xTimerReset() : Software timer reset

BaseType_t xTimerReset(TimerHandle_t xTimer, TickType_t xTicksToWait);	
xTimer	소프트웨어 타이머 handle
xTicksToWait	task가 Blocked state에 있을 시간의 최대값 0이면 바로 리턴 portMAX_DELAY은 무한대기 (INCLUDE_vTaskSuspend ("FreeRTOSConfig.h") 1로 설정
return value	pdPASS : 성공 pdFALSE : 실패(시간초과)



FreeRTOS - Software Timer Management

Example17 – software timer reset

감사합니다.