

## FreeRTOS Software Timers

### Software Timer 개요

Software Timer는 설정된 시간에 Timer의 Callback Function을 실행 하도록 한다. Timer 가 Start 되고 설정된 시간 후(Period)에 Timer Callback Function이 실행 된다.

Timer Callback Function은 Timer Service Task로 실행 되기 때문에 Blocking 상태가 될 수 없다. 그렇기 때문에 Timer Callback Function에서는 vTaskDelay(), vTaskDelayUntil() API를 Call 할 수 없고, Queue 나 Semaphore를 Accessing 할 때 Block Time을 zero로 하여야 한다.

### Timer Command Queue

FreeRTOS는 여러 종류의 Timer 관련 API 함수를 갖고 있다. 이 들 함수는 Timer Command Queue를 사용 하여 Timer Service Task 에 명령을 전달 한다. Timer Command Queue는 Timer 가 구현 될 때 생성되고, Private 속성을 갖기 때문에 사용자가 직접 Access 할 수 없다.

### Software Timer를 사용 하기 위한 준비

FreeRTOS/Source/timers.c File를 Project에 포함 시킨다.

FreeRTOSConfig.h Header File에 다음 상수를 Define 한다.

Constant	Description
configUSE_TIMERS	1로 Set 하면 RTOS Scheduler 가 Start 할 때 Timer Service Task를 자동으로 Create 한다.
configTIMER_TASK_PRIORITY	Timer Service Task의 Priority를 설정 한다. 다른 Task와 마찬가지로 0 - (configMAX_PRIORITIES-1) 사이의 값으로 설정 할 수 있다.
configTIMER_QUEUE_LENGTH	Timer Command Queue에 미처 처리 되지 못한 Command를 Hold 할 수 있는 수
configTIMER_TASK_STACK_DEPTH	Timer Service Task에서 사용 할 수 있는 Stack Size(Word)

### One-shot Timers 와 Auto-reload Timers

#### One-shot Timers

One-shot Timer는 한번 Start 되면 Callback function을 한번 만 실행 한다. One-shot Timer를 다시 Start 시키려면 xTimerStart() 등의 API 명령을 사용 하여야 한다.

### Auto-reload Timers

Auto-reload Timer는 Callback function이 실행 된 다음에 자동으로 Restart 되어 일정한 주기로 반복 하여 실행 된다.

### Software Timer의 Resetting

이미 Start 된 Timer의 Resetting 이 가능 하다.

Timer를 Resetting 하는 경우 Timer의 시작 시간이 재 설정 되어 다시 설정 주기 가 만료된 후에 Callback function이 실행 된다.

### FreeRTOS Software Timer API Function

- xTimerCreate
- xTimerIsTimerActive
- pvTimerGetTimerID
- xTimerStart
- xTimerStop
- xTimerChangePeriod
- xTimerDelete
- xTimerReset
- xTimerStartFromISR
- xTimerStopFromISR
- xTimerChangePeriodFromISR
- xTimerResetFromISR
- xTimerGetTimerDaemonTaskHandle

참고자료: [www.freertos.org](http://www.freertos.org)에서

FreeRTOS -> API Reference -> Software timers 를 참고 할 것.