

# Hook Functions

## INTRODUCTION

Hook functions in general are function calls that are placed in existing source code, to execute custom code that can be added later. This is often done to keep the base code extensible to the user without exposing the base code. In an RTOS, this is used to allow the user application to extend the kernel functionality. The calls to the hook functions are placed at predefined points of the kernel code.

Hook functions can be mandatory, that means that the application must define a definition for the hook function, or optional. Optional hook functions are predefined as weak functions which may be overridden by a strong application function at linktime.

Hook functions are a commonly used feature in software components but are not standard in an RTOS. If and how they are implemented is different for every RTOS.

The following hook functions are defined in the SAFERTOS® kernel code.

## MANDATORY HOOK FUNCTIONS

### vApplicationErrorHook

The vApplicationErrorHook function is the only mandatory hook function of SAFERTOS®. It is used to report non-recoverable errors to the application and should not return. This function is provided so that the application can place the system in a safe state. The safe failure state is application dependent on the system and may include procedures like shutting down motors. Reasons for vApplicationErrorHook to be called are: corrupted TCB, stack overflow etc. The function has two parameters: xHandleOfTaskWithError and xErrorCode which help to identify the cause of the error.

## OPTIONAL HOOK FUNCTIONS

### vApplicationSetupTickInterruptHook

The vApplicationSetupTickInterruptHook is used to set up the timer for the system tick. The predefined weak function initializes a predefined timer to be used for this, but the application may override this function to use another timer with different settings. The function has two parameters: ulClockHz which is the processors clock frequency in Hz and ulRateHz which is the desired tick frequency in Hz.

### vApplicationTaskDeleteHook

The vApplicationTaskDeleteHook is called from the kernel whenever a task gets deleted. The predefined weak function is empty, but the application may use this hook to react to a deleted task. The function has a single parameter xTaskBeingDeleted.

### vApplicationTickHook

The vApplicationTickHook is called from the tick Interrupt Service Routine (ISR). The predefined weak function is empty. It is called from an ISR context and should therefore be short and non-blocking.

### vApplicationIdleHook

The vApplicationIdleHook is called from the kernel whenever the idle task is active. The idle task runs at the lowest priority and is only active if no other task of higher priority is ready to run. Therefore, the vApplicationIdleHook enables the application to do background processing or put the processor into a low-power state as long as no processing is to be performed.

## PORT-SPECIFIC HOOK FUNCTIONS

Some ports of SAFERTOS® define further hook functions which are called from the portable layer code.

### vApplicationSVCHook

The vApplicationSVCHook is only available on some ports with an SVC (supervisor call) handler, like ARM Cortex M7. Usually, the SVC is used by the kernel to implement functions like yield, enter and exit critical sections, change privilege levels or start the first task. The function is selected by supplying a number as a parameter to the SVC handler. If that number does not match a predefined function of the kernel, the vApplicationSVCHook function with the number as parameter is called.

