FreeRTOS - SysTick and Delay

SysTick is mainly used for delay function in non-RTOS firmware, and is used as the interrupt for RTOS scheduler. If STM32 HAL utilizes another timer as its time base, RTOS has its own right to initialize and handler SysTick. The Delay function also works differently when RTOS is being used.

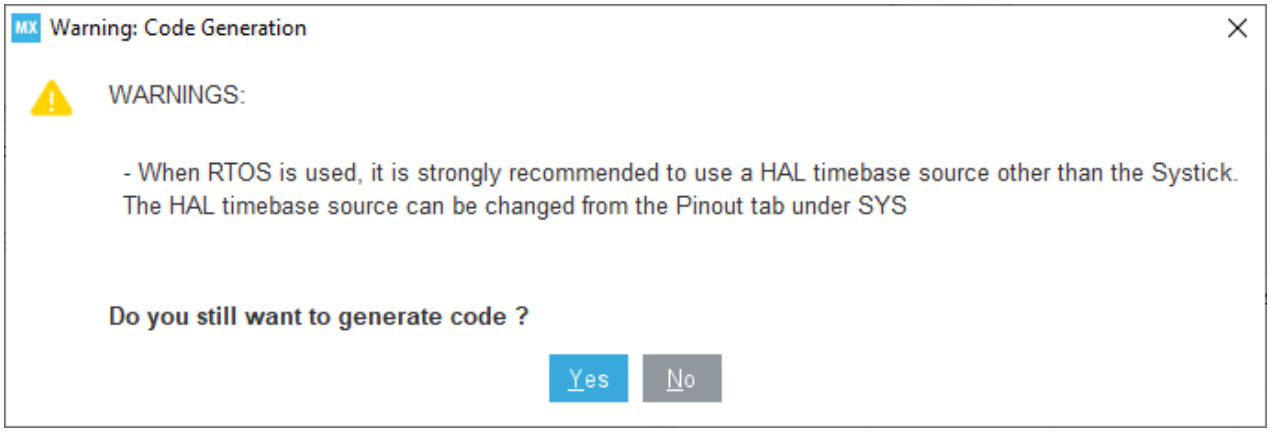
#arm #stm32 #rtos #systick

Last update: 2021-08-04 17:31:07

Table of Content

1. SysTick

When generating a new project from STM32CubeMX, there will be a popup saying that:



Prompt asking about set another time base source

What does this mean?

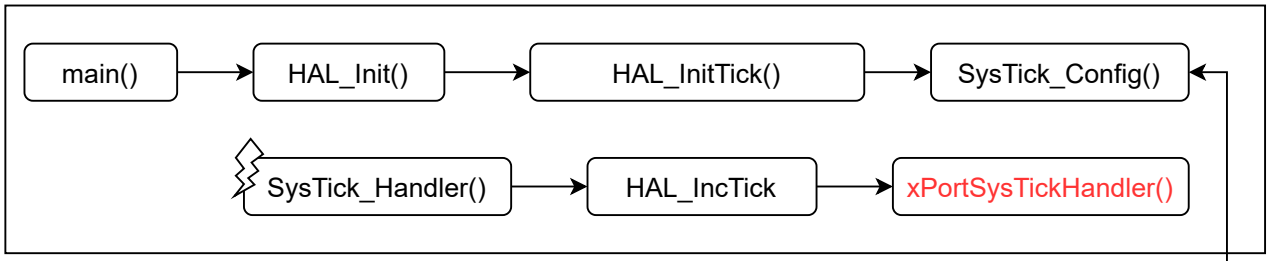
SysTick

SysTick is apart of the ARM Core, that counts down from the reload value to zero, and fire an interrupt to make a periodical event. SysTick is mainly used for delay function in non-RTOS firmware, and is used as the interrupt for RTOS scheduler.

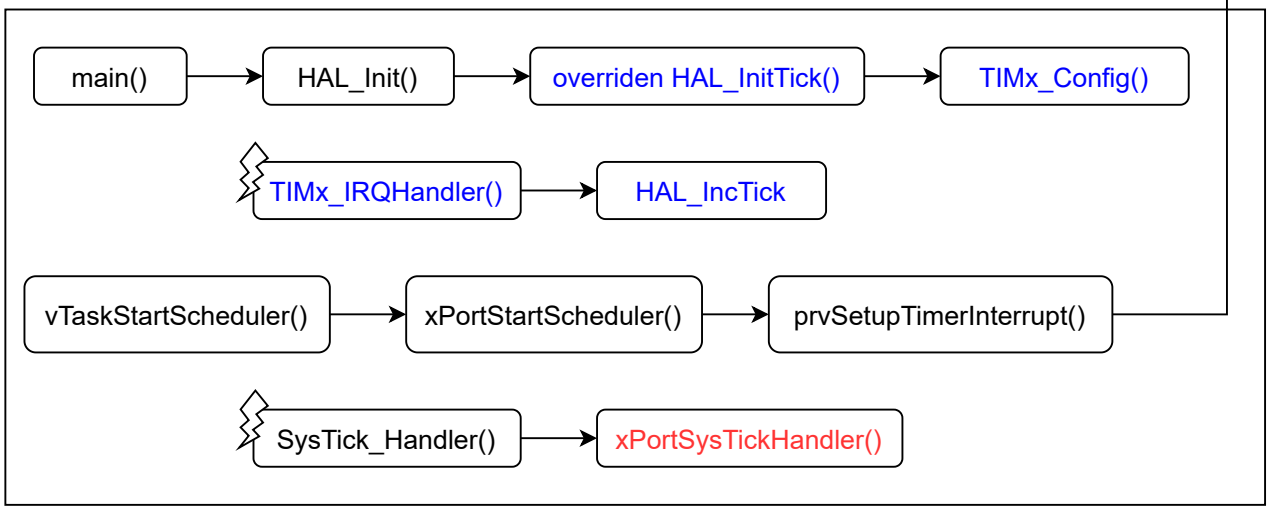
In case STM32 HAL code also uses SysTick as its time base, RTOS will be generated to use HAL's Handler.
If STM32 HAL utilizes another timer as its time base, RTOS has its own right to initialize and handler SysTick.

It is recommended to use SysTick for RTOS only, and set a basic timer as the time base for HAL.

Shared SysTick



Delicated SysTick for RTOS



SysTick in different use cases

2. Delay

- The function `osDelay()` is part of CMSIS Library and uses `vTaskDelay()` internally to introduce delay with the difference that input argument of `osDelay` is delay time in milliseconds while the input argument of `_vTaskDelay()` is number of Ticks to be delayed. Using this `osDelay()` function, OS will be notified about the delay and OS will change the status of task to blocked for that particular time period.
- The function `HAL_Delay()` is part of the hardware abstraction layer. It basically uses polling to introduce delay. Using this `HAL_Delay()` function, OS won't be notified about the delay, and the code is in polling mode.
- FreeRTOS delay functions: `vTaskDelay()` or `vTaskDelayUntil()` only take effect after the scheduler has started.

Always use `osDelay()` when using RTOS.