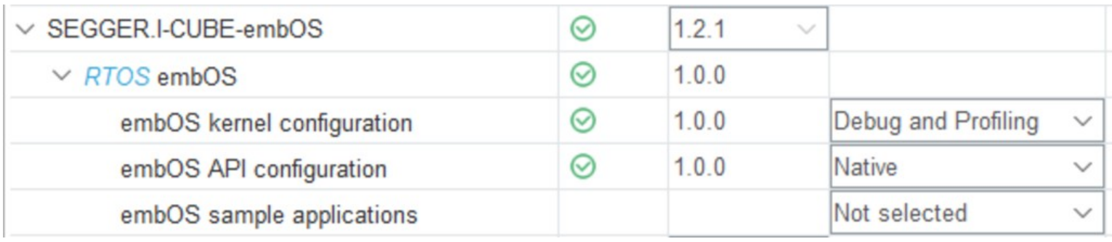
Getting started with the I-CUBE-embOS Expansion

How to use the I-CUBE-embOS Expansion in STM32CubeMX

1. Open STM32CubeMX and create a new or open an existing STM32CubeMX project for any STM32 Cortex-M device.

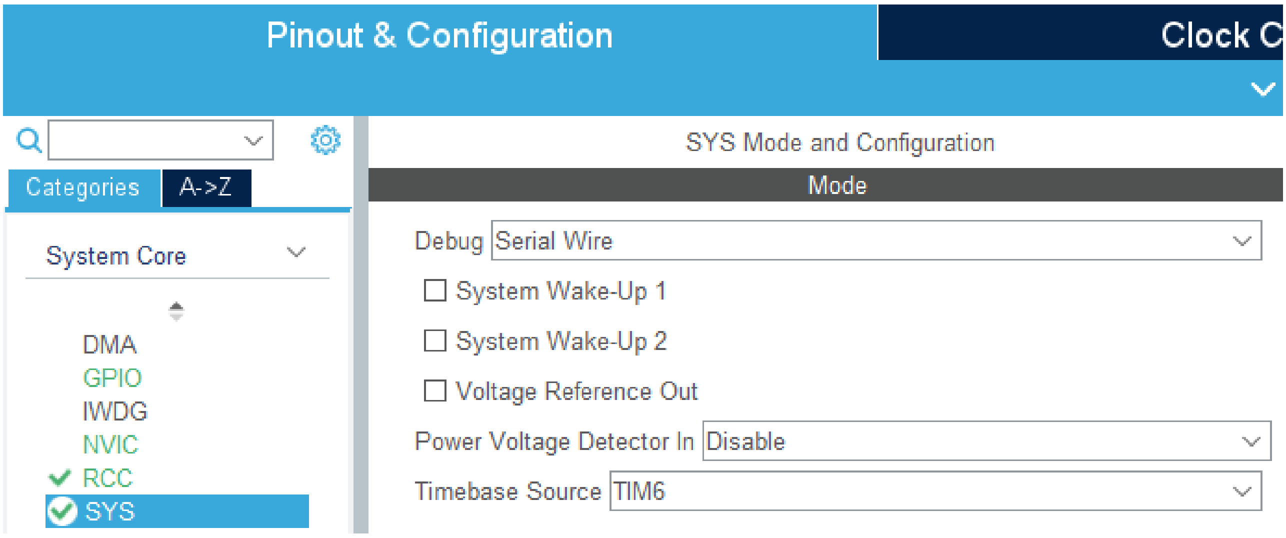
2. If not already installed, please install the I-CUBE-embOS Expansion pack.

3. Open the "Software Packs Component Selector" in STM32CubeMX and select the desired "embOS kernel configuration" and "embOS API configuration". We recommend starting with a Debug and Profiling configuration and the native API. Close the Component Selector with the "OK" button to confirm your selection.



SEGGER.I-CUBE-embOS	✓	1.2.1	
RTOS embOS	✓	1.0.0	
embOS kernel configuration	✓	1.0.0	Debug and Profiling
embOS API configuration	✓	1.0.0	Native
embOS sample applications			Not selected

4. In the "Pinout & Configuration" view, select "SYS" in the Categories on the left and choose another hardware timer as a timebase source for HAL to avoid any conflicts, since embOS is using the SysTick by default.



SYS Mode and Configuration

Mode

Debug Serial Wire

□ System Wake-Up 1

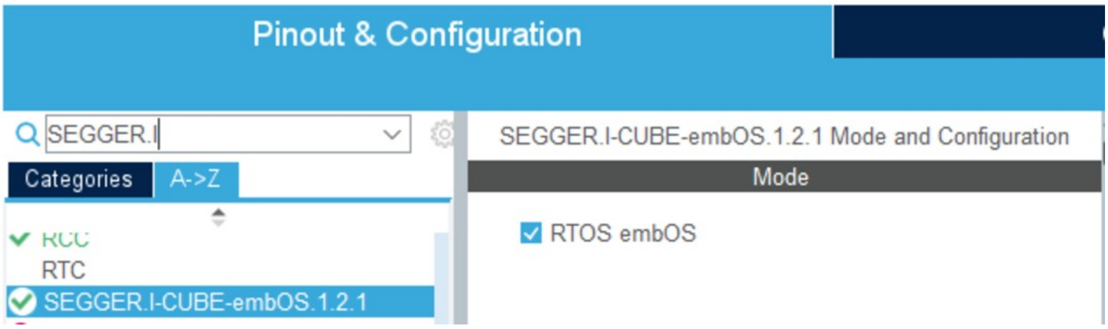
□ System Wake-Up 2

□ Voltage Reference Out

Power Voltage Detector In Disable

Timebase Source TIM6

5. Next, select the I-CUBE-embOS pack and select the checkbox to enable and include all required embOS files and libraries in your projects



SEGGER.I-CUBE-embOS.1.2.1 Mode and Configuration

Mode

☑ RTOS embOS

6. Now, you can generate your project. If your project was generated for Keil or STM32CubeIDE some modifications might be necessary for the project to build successfully.

Keil MDK

If the linker complains about the undefined symbol Stack_Mem, the Stack_Mem symbol needs to be exported by the start-up file. Open the startup assembler file and add following line:

```
EXPORT  Stack_Mem
```

Alternatively, if another symbol pointing to the end of the stack (lowest address of the stack) is exported, OS_InitSysStackInfo.c can be modified to use this symbol instead of Stack_Mem.

If the linker complains about the undefined symbol __initial_sp, the __initial_sp symbol needs to be exported by the start-up file. Open the startup assembler file and add following line:

```
EXPORT  __initial_sp
```

Please note, that there might be conflicts with the stack/heap set up methods and ARM compiler V6 when __initial_sp is exported and MicroLIB is not used. In this case, and if the start-up file supports the use of MicroLIB, MicroLIB can be enabled instead of exporting __initial_sp (Options for Target -> Target -> Code Generation -> Use MicroLIB).

Alternatively, if another symbol pointing to the beginning of the stack (highest address of the stack) is exported, OS_InitSysStackInfo.c can be modified to use this symbol instead of __initial_sp.

STM32CubeIDE

If the linker complains about undefined references to __stack_start__ and __stack_end__ open the linker description file for your project and replace:

```
. = . + _Min_Stack_Size;
```

with

```
__stack_start__ = .;
. = . + _Min_Stack_Size;
__stack_end__ = .;
```

Documentation

For more information on how to use embOS, please take a look into the embOS Manuals located in the I-CUBE-embOS pack at Middlewares\Third_Party\embOS\Documentation\.

Further information on embOS related topics can be also found in the SEGGER embOS Wiki:

https://wiki.segger.com/embOS

Example projects

This pack contains 3 example projects made for the NUCLEO-L152RE demonstrating how to create tasks, how to use mutexes and how to use mailboxes. Further examples can be found in the embOS manual and the embOS Shipments which can be downloaded from:

https://www.segger.com/downloads/embos/