

ADD UNISON RTOS TO STM32CubeMX

Description

This manual explains how to add the UNISON RTOS to STM32CubeMX

Setup in STM32CubeMX

- 1 Open STM32CubeMX.
- 2 In the "Manage software installations" press "INSTALL/REMOVE" (see figure 1)

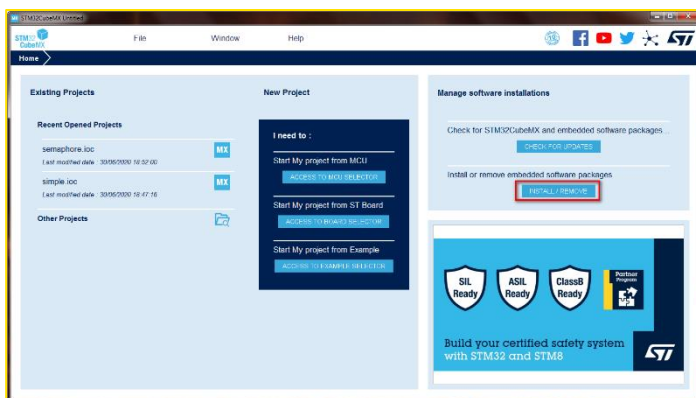


Figure 1

- 3 Press "From Local". Navigate to the directory with UNISON RTOS package, select it and press "Open" (see figure 2)

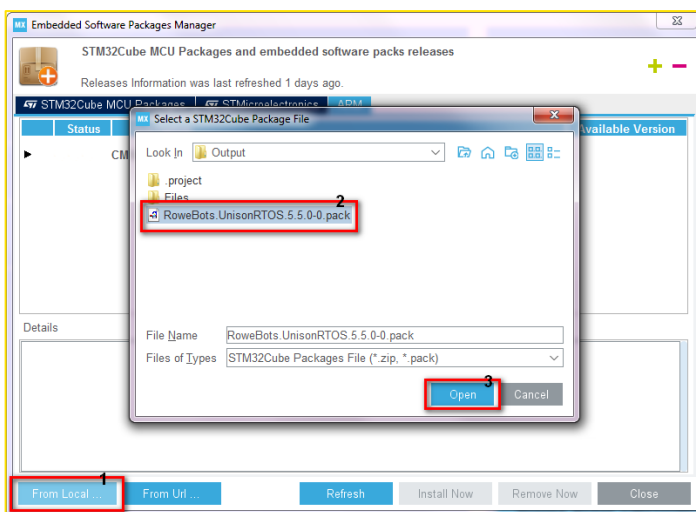


Figure 2



4. Read the License Agreement and choose "I accept the Terms of this License Agreement" checkbox to proceed. This is a mandatory step. (see figure 3)

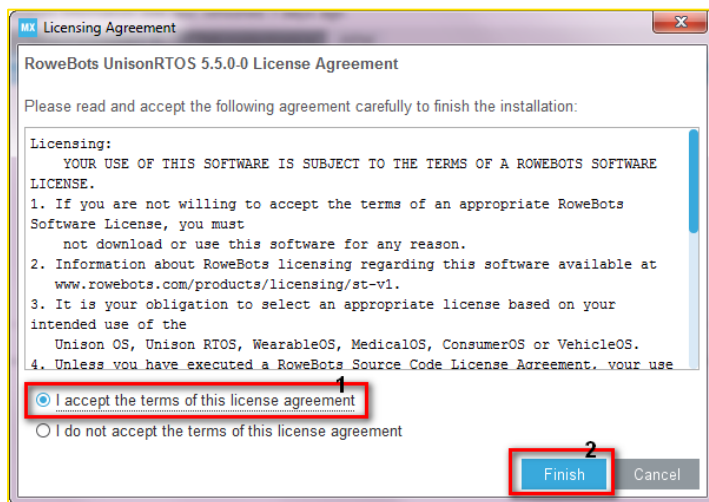


Figure 3

5. After successful setup UNISON RTOS will appear under the RoweBots Tab (see figures 4 and 5)

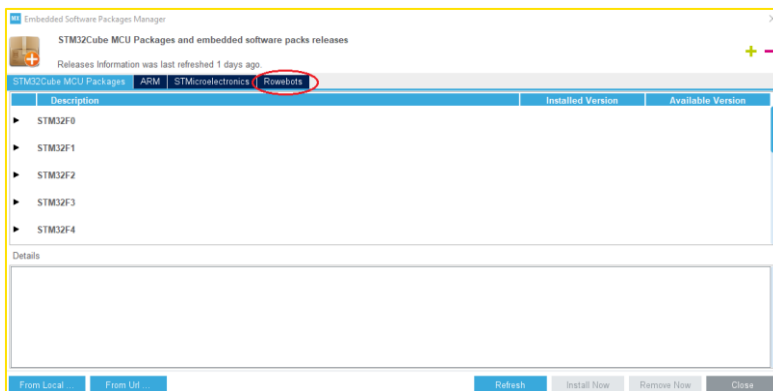


Figure 4

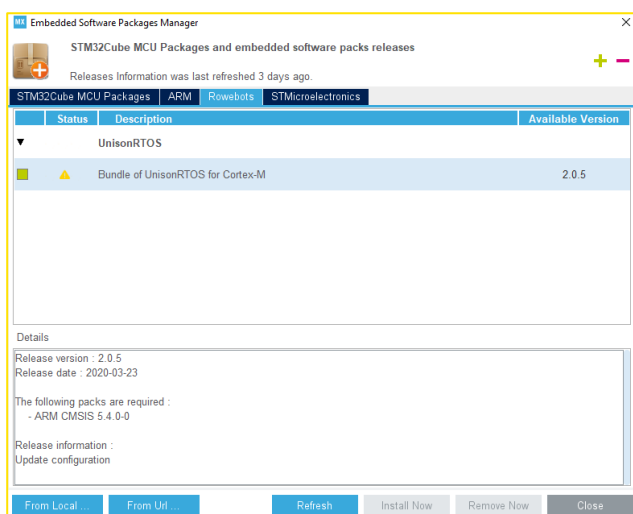


Figure 5

6. Click "Close"



Configure UNISON RTOS parameters

1. Create a new project in the STM32CubeMX tool. You should click "File" button and select "New Project" option from the drop-down menu (see figure 6)

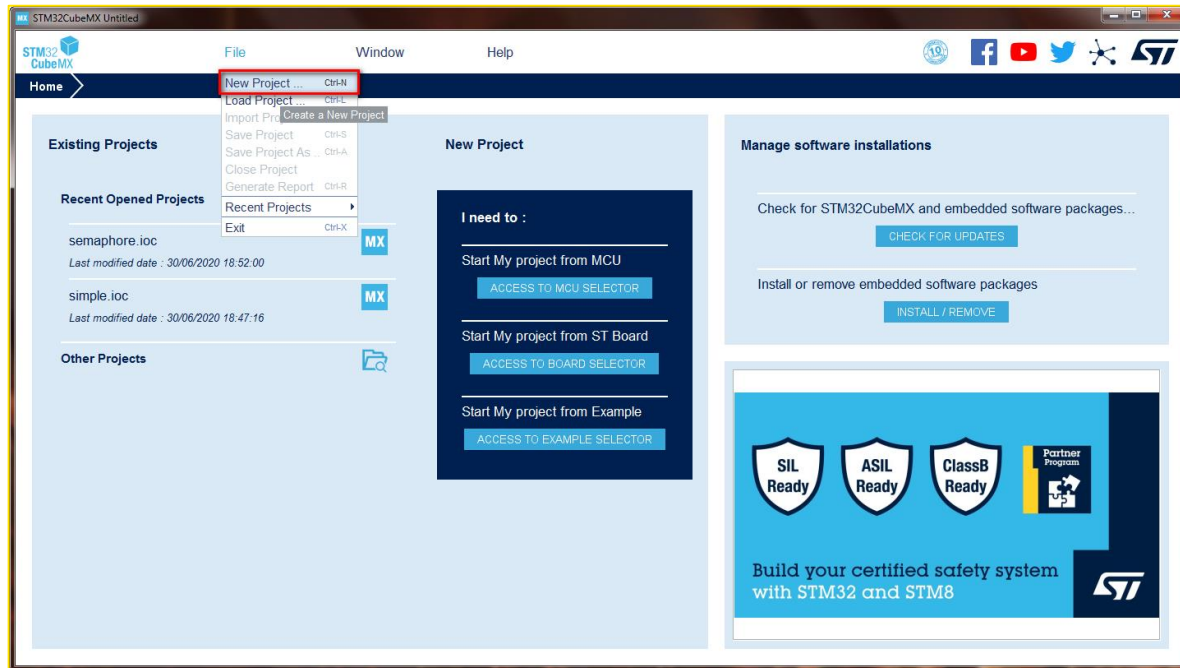


Figure 6

2. Select a required chip or a board from the selector menu and click "Start Project" to proceed. (see figure 7)

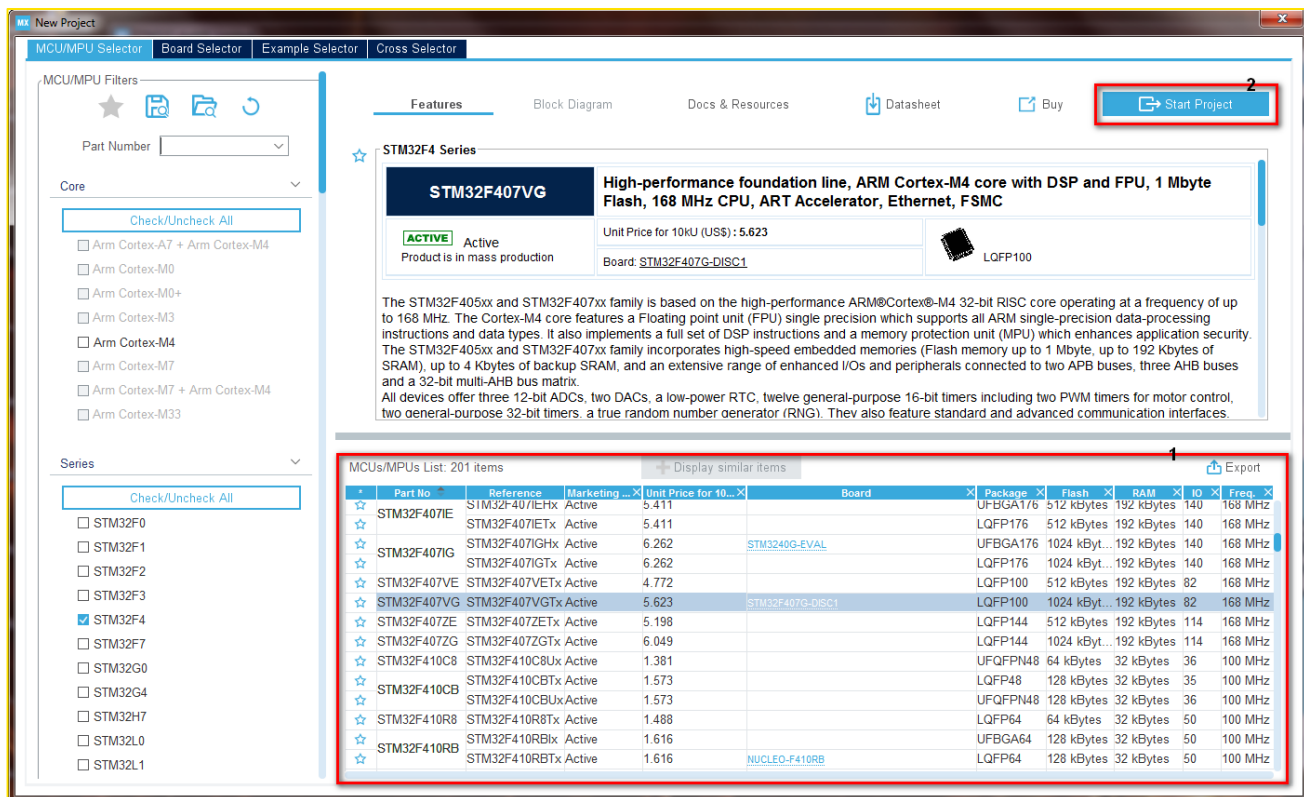


Figure 7



3. Click on the "Software Packs" drop-down menu and choose the "Select Components" option. (see figure 8)

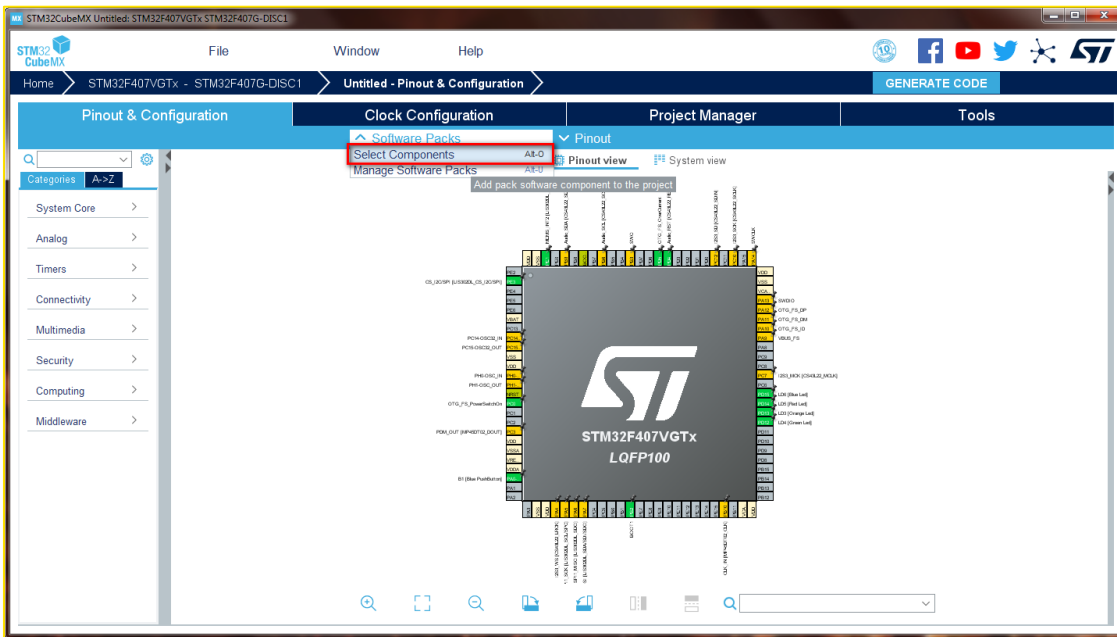


Figure 8

4. Find the RoweBots UnisonRTOS package. Set checkboxes for the "KERNEL" and "CMSIS" components from the RTOS UnisonRTOS bundle. You can select an application from the Application list if you prefer to generate some example off-the-shelf. Press "Ok" button to apply your settings. (see figure 9)

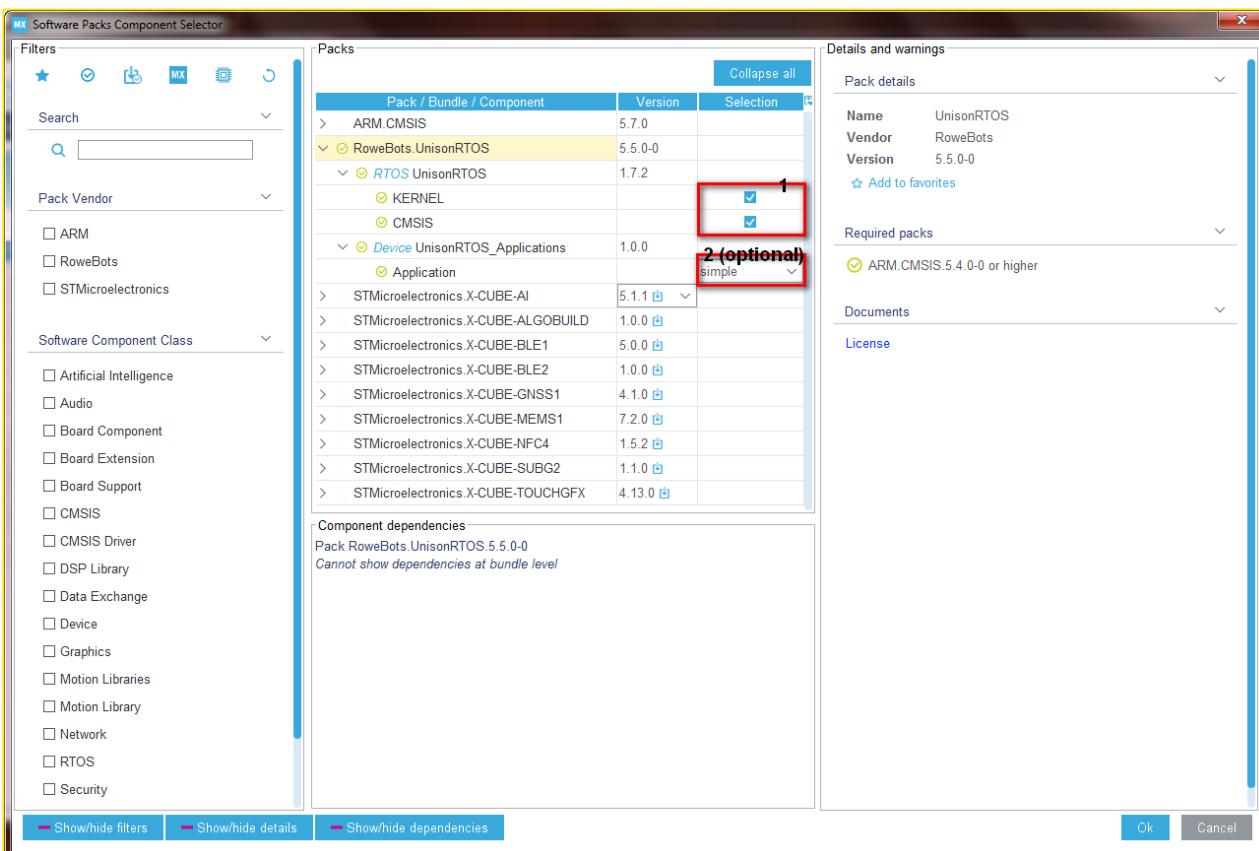


Figure 9



- 5 Find and select "RoweBots UnisonRTOS" package from the "Software Packs" list. Set checkbox for RTOS UnisonRTOS option. You can also set a checkbox for "Device UnisonRTOS Application" in case if some "Application" was selected in the previous step.
- 6 Check "Parameter Settings" tab to customize UnisonRTOS settings depending on your requirements. Feel free to customize some other CubeMX project's settings like "Clock Configuration" or others if necessary. (see figure 10)

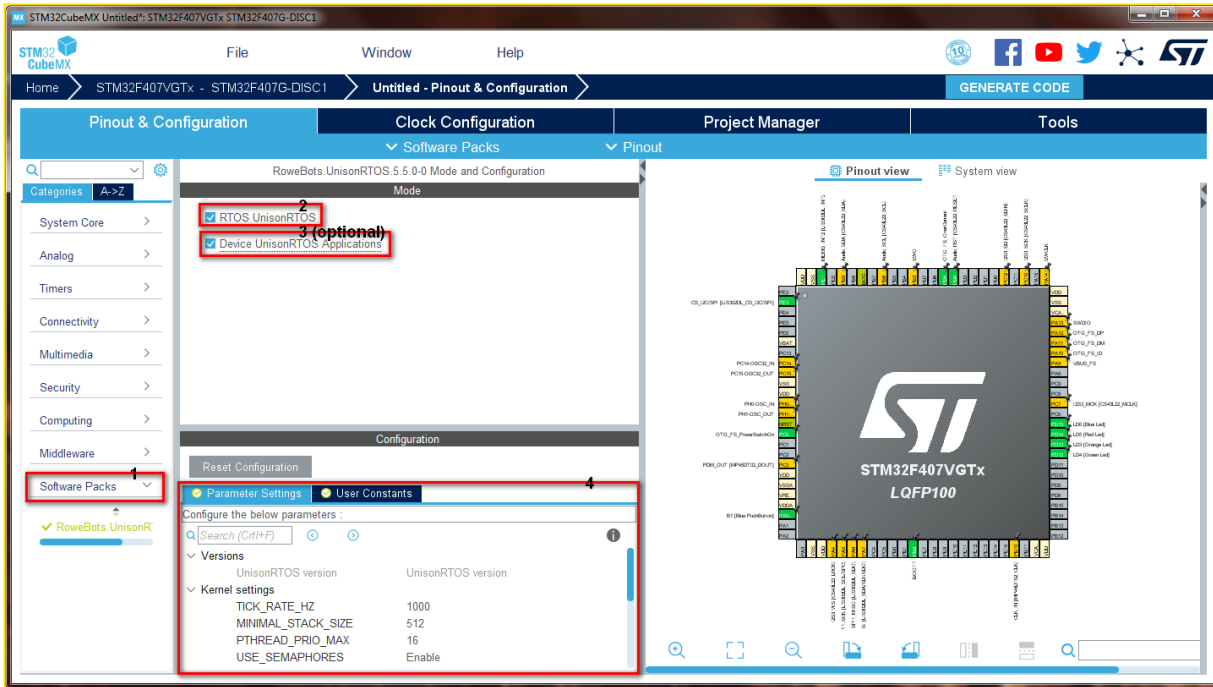


Figure 10

- 7 Don't forget to set checkboxes to generate and call SysTick ISR under the NVIC section. (see figure 11)

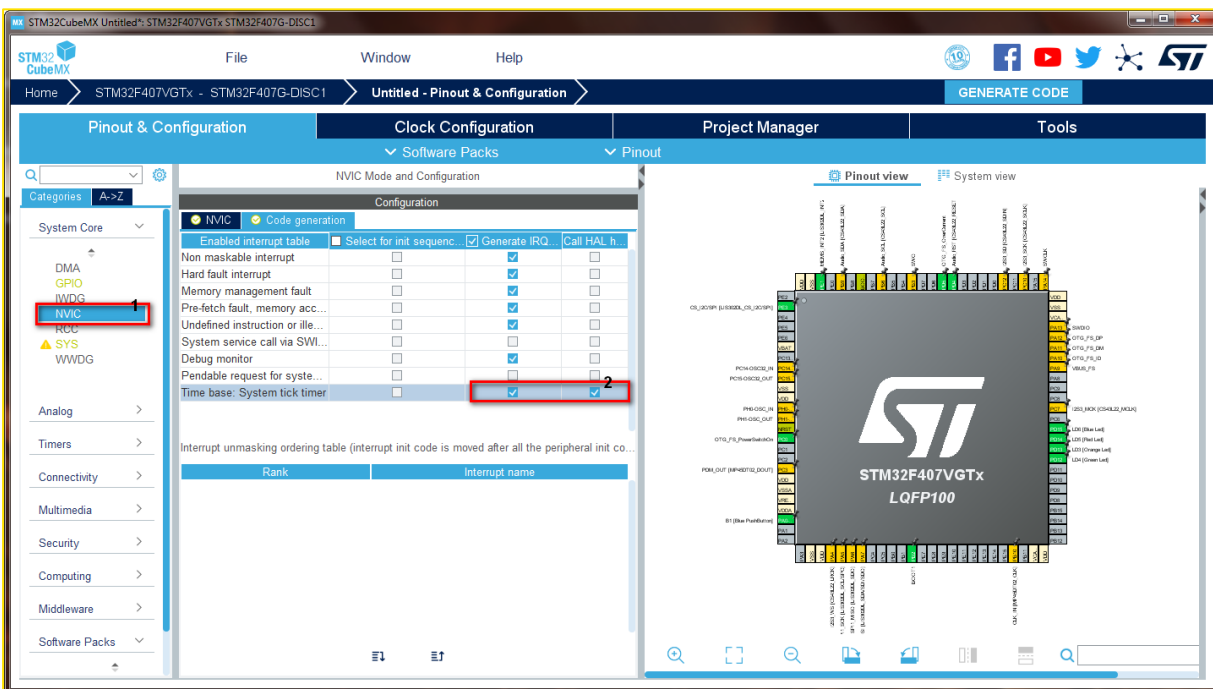


Figure 11



8. Configure project's name and other options under the "Project Manager" tab. Screenshot which is posted below is for reference only, user settings may be different, depending on requirements.
9. Select a required "Toolchain / IDE" from the listbox.
10. Press "GENERATE CODE" button when you are ready to proceed. (see figure 12)

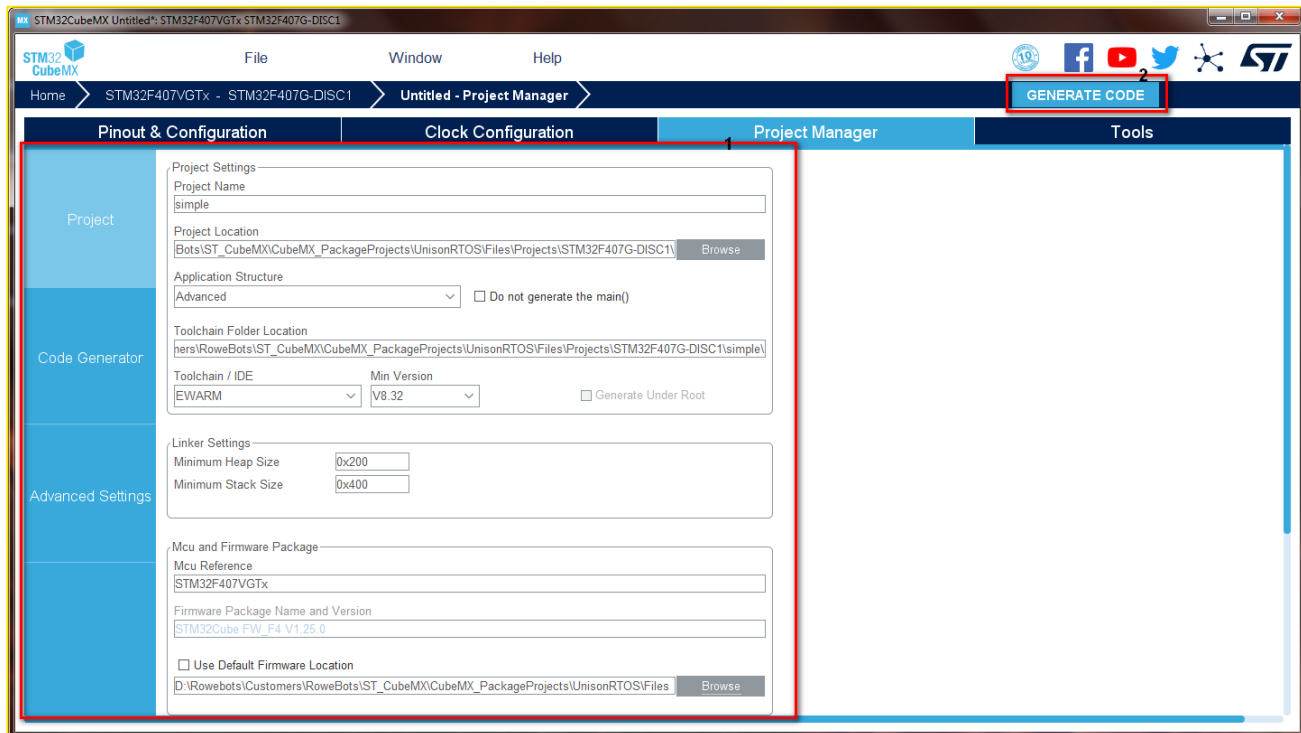


Figure 12

11. Use a SysTick as a timebase source. (see figure 13)

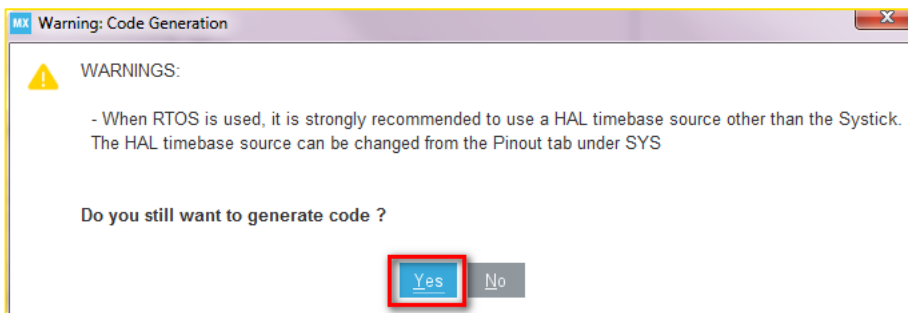


Figure 13

12. At this point, your first project with UnisonRTOS inside should be generated successfully. (see figure 14)

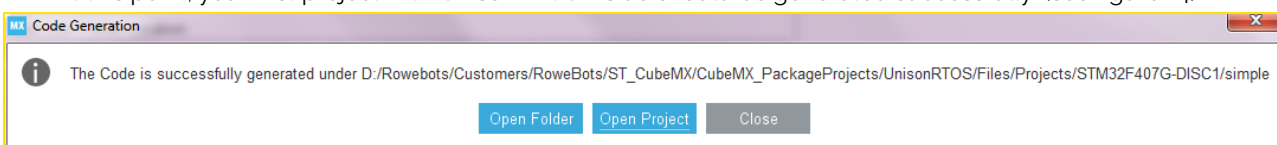


Figure 14



13. You can generate projects for different IDEs if required. Here is a list of IDEs which could be chosen. (see figure 15)

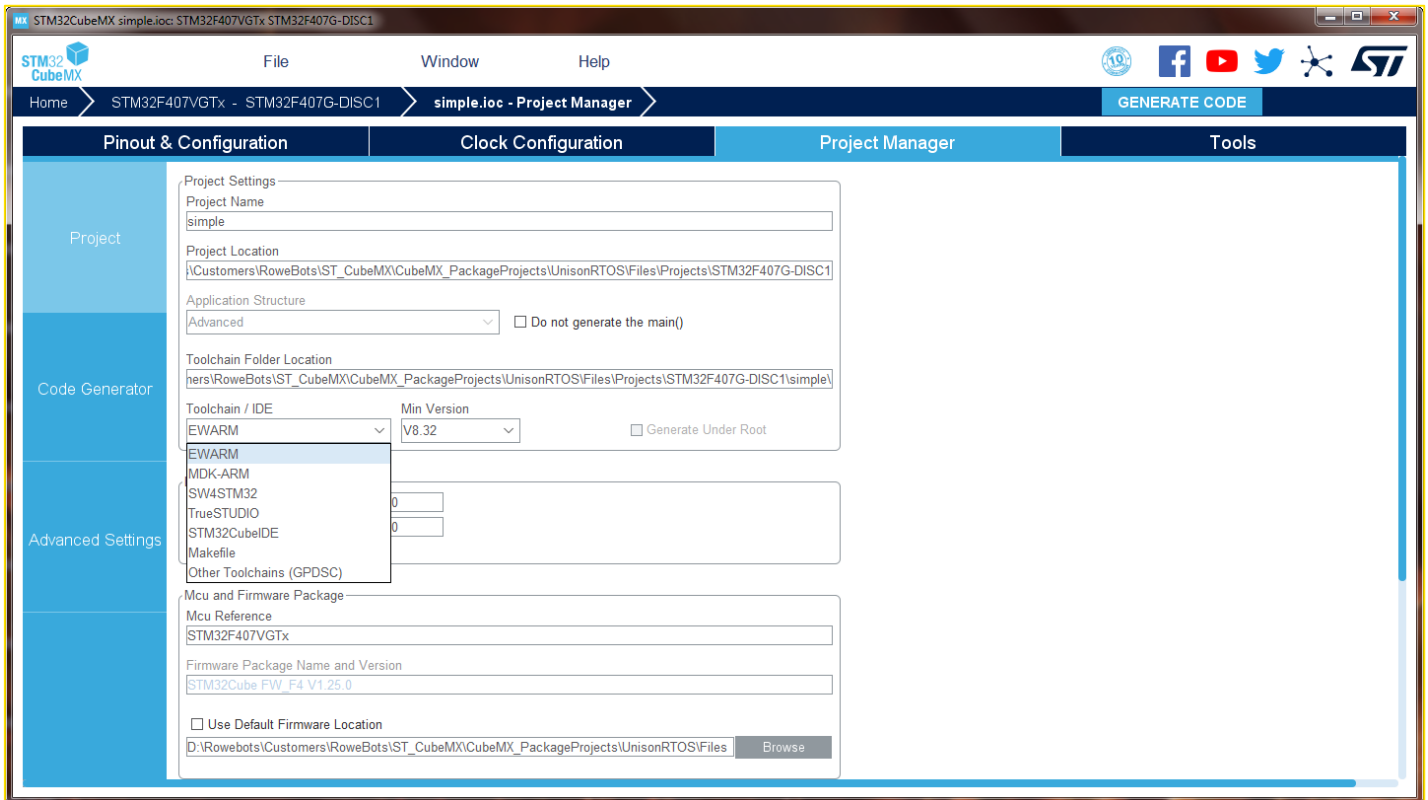


Figure 15

14. Here is how a workspace in the IAR Embedded Workbench IDE looks (see figure 16)

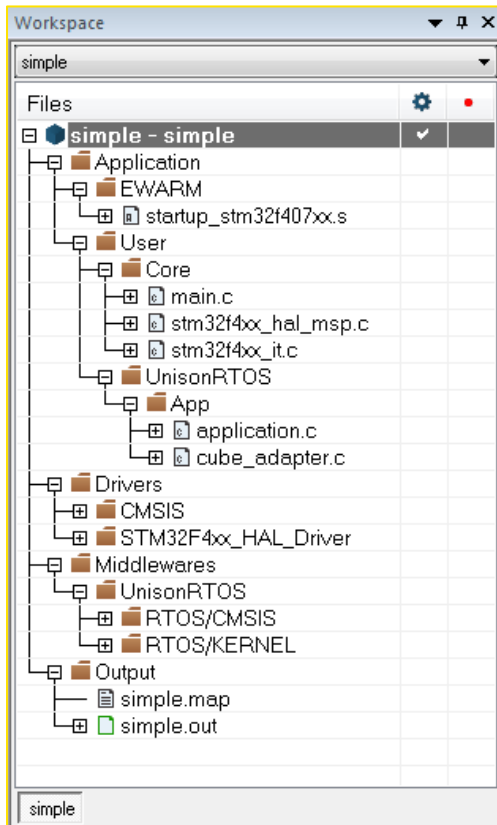


Figure 16



IMPORTANT



For KEIL

1. In the Project settings, in the tab "C/C++" add in the "Misc Controls" field:

-J"../Middlewares/Third_Party/Rowebots_RTOS_UnisonRTOS/include"

-J"C:/Install/Keil_v5/ARM/ARMCC/include"

Path 1 - is the path to our includes

Path 2 - is the path to the Keil library.

IMPORTANT! Path 2 is a local folder where the files of installed Keil are located. So users should input this manually anyway.

2. In the Project settings in the tab "Asm" add in the "Misc Controls" field the following: --cpreproc

3. In the low_level_asm.s file settings change the filetype to Assembly file

IMPORTANT



For CubeIDE

1. In the Project settings in the tab "C/C++ Build -> Settings -> Tool Settings -> MCU GCC Compiler -> General" select "ISO C99 (-std=c99)" encoding.

2. Delete Syscalls.c and systemem.c generated by CubeIDE.



15. 5 Note, that UNISON RTOS Main task could be found into the "application.c" file. You can customize it depending on your requirements. (see figure 17):

```

37  /*
38   * Main thread
39   */
40  THREAD Main(void *args)
41  {
42      struct timespec time;
43
44      // Setup system time
45      time.tv_sec = 0; time.tv_nsec = 0;
46      clock_settime(CLOCK_REALTIME, &time);
47
48      /*
49       * Infinite loop.
50       */
51      while(1)
52      {
53
54          sleep(5);
55      }
56  }

```

Figure 17

16. Please, verify that "SysTick_Handler" was generated successfully and "xPortSysTickHandler" routine is called inside. (see figure 18)

```

153  /**
154   * @brief This function handles System tick timer.
155   */
156  void SysTick_Handler(void)
157  {
158      /* USER CODE BEGIN SysTick_IRQn 0 */
159
160      /* USER CODE END SysTick_IRQn 0 */
161      HAL_IncTick();
162      #if (INCLUDE_xTaskGetSchedulerState == 1 )
163          if (xTaskGetSchedulerState() != taskSCHEDULER_NOT_STARTED)
164          {
165              #endif /* INCLUDE_xTaskGetSchedulerState */
166              xPortSysTickHandler();
167              #if (INCLUDE_xTaskGetSchedulerState == 1 )
168              }
169              #endif /* INCLUDE_xTaskGetSchedulerState */
170      /* USER CODE BEGIN SysTick_IRQn 1 */
171
172      /* USER CODE END SysTick_IRQn 1 */
173  }

```

Figure 18



IMPORTANT



Please, verify that "MX_UnisonRTOS_Init" routine is called inside the "main.c" file. If you didn't choose an "Application" option before, you should add "MX_UnisonRTOS_Init" call manually. In case if you choose an "Application" off the shelf, this call should be already here.. (see figure 19)

```

64 int main(void)
65 {
66     /* USER CODE BEGIN 1 */
67
68     /* USER CODE END 1 */
69
70     /* MCU Configuration-----*/
71
72     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
73     HAL_Init();
74
75     /* USER CODE BEGIN Init */
76
77     /* USER CODE END Init */
78
79     /* Configure the system clock */
80     SystemClock_Config();
81
82     /* USER CODE BEGIN SysInit */
83
84     /* USER CODE END SysInit */
85
86     /* Initialize all configured peripherals */
87     MX_GPIO_Init();
88     MX_UnisonRTOS_Init();
89     /* USER CODE BEGIN 2 */
90
91     /* USER CODE END 2 */
92
93     /* Infinite loop */
94     /* USER CODE BEGIN WHILE */
95     while (1)
96     {
97         /* USER CODE END WHILE */
98
99         MX_UnisonRTOS_Process();
100         /* USER CODE BEGIN 3 */
101     }
102     /* USER CODE END 3 */
103 }

```

Figure 19

Now you are ready to go.

Thank you for choosing UNISON RTOS.



RoweBots

+ 1(519) 279 4600

support@rowebots.com

rowebots.com