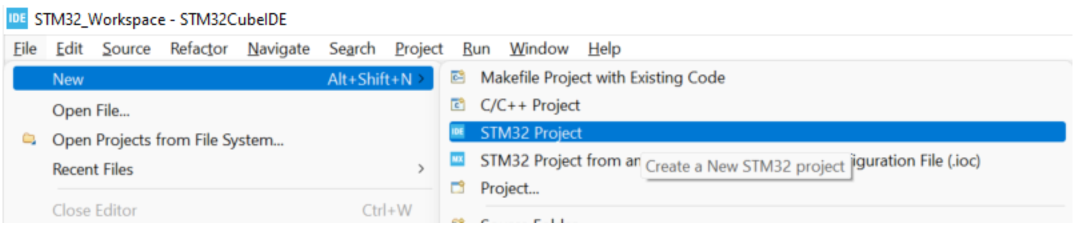
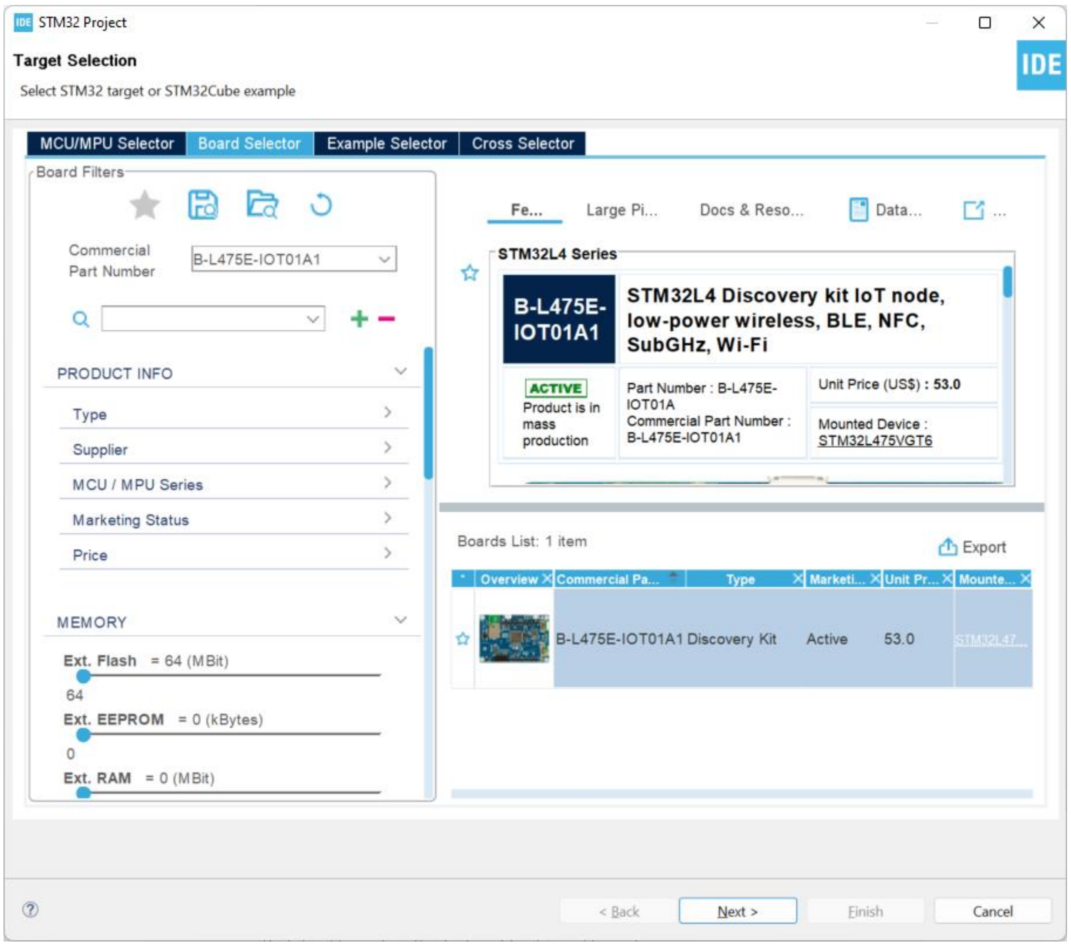
Lab Objectives

1. Create an STM32 project that uses FreeRTOS
2. Configure FreeRTOS parameters
3. Write code to create tasks and task functions
4. Configure STMCubeIDE debugger for FreeRTOS
5. Debug the project and observe FreeRTOS functionality

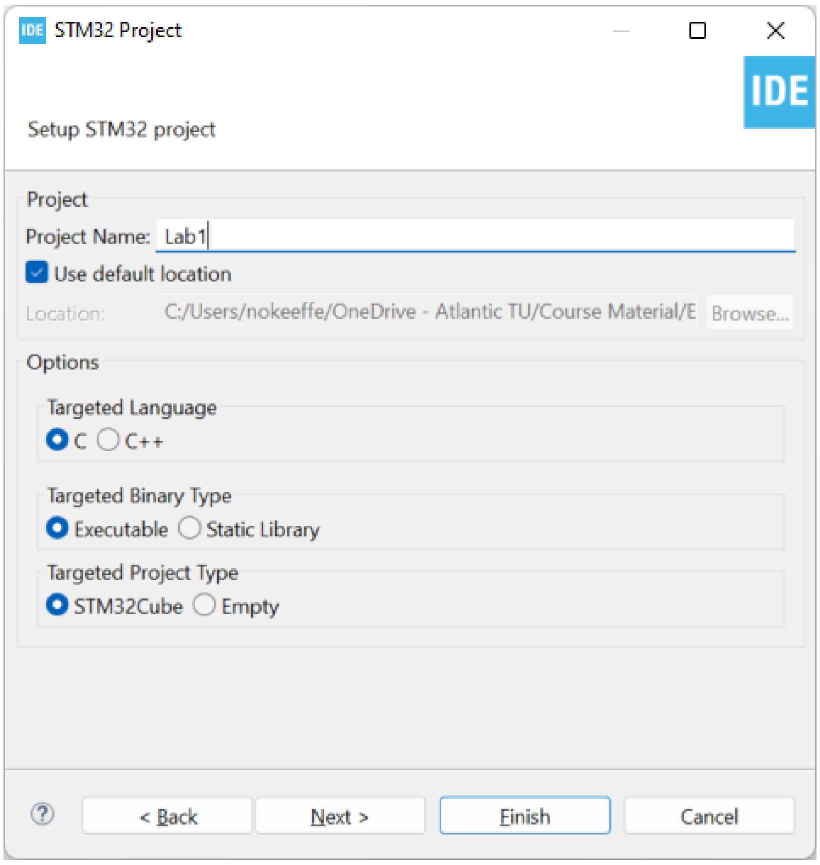
Step 1: Create a new STM32CubeIDE project.


STM32_Workspace - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor Ctrl+W
Makefile Project with Existing Code
C/C++ Project
STM32 Project
STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)
Create a New STM32 project
Project...

Click the Board Selector tab and type the board name B-L475E-IOT01A1. Select the board and click Next.



Enter a project name and click Finish.


STM32 Project

Setup STM32 project

Project

Project Name: Lab1

[✓] Use default location

Location: C:/Users/nokeeffe/OneDrive - Atlantic TU/Course Material/E

Options

Targeted Language

[✓ C] C++

Targeted Binary Type

[✓ Executable] Static Library

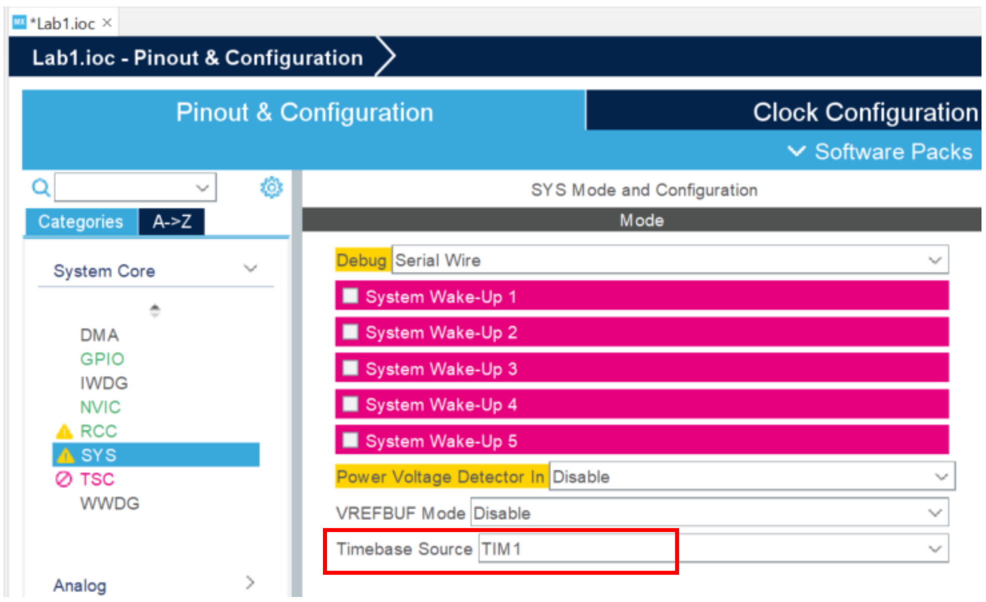
Targeted Project Type

[✓ STM32Cube] Empty

Click Yes to initialise all peripherals to default mode.

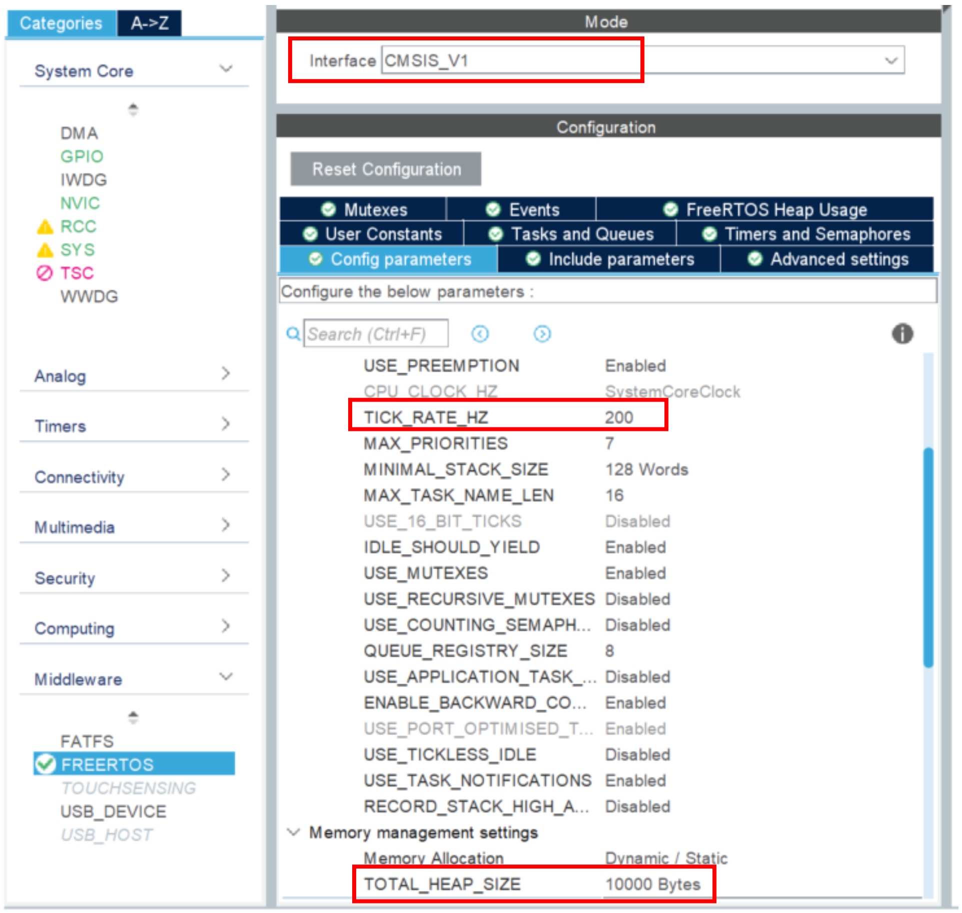
Step 2: FreeRTOS Configuration

The timebase needs to be changed from the SysTick timer to TIM1. Select the System Core tab and then the SYS tab.



Debug Serial Wire
Power Voltage Detector In Disable
VREFBUF Mode Disable
Timebase Source TIM1

Select the FreeRTOS tab in the Middleware category. Select the Config Parameters tab. Set the interface to CMSIS_V1, the tick rate to 200Hz and the Heap size to 10,000 Bytes.

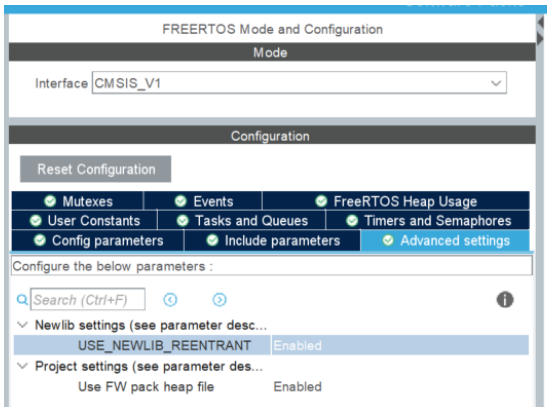


Interface CMSIS_V1

Configure the below parameters :

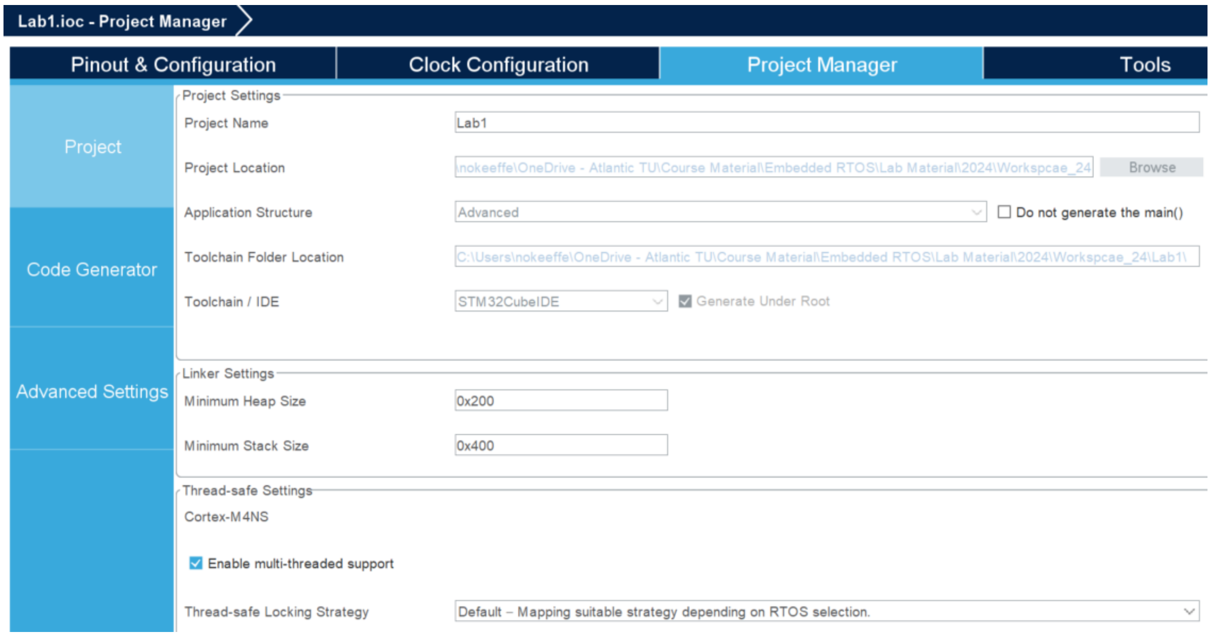
USE_PREEMPTION	Enabled
CPU_CLOCK_HZ	SystemCoreClock
TICK_RATE_HZ	200
MAX_PRIORITIES	7
MINIMAL_STACK_SIZE	128 Words
MAX_TASK_NAME_LEN	16
USE_16_BIT_TICKS	Disabled
IDLE_SHOULD_YIELD	Enabled
USE_MUTEXES	Enabled
USE_RECURSIVE_MUTEXES	Disabled
USE_COUNTING_SEMAPH...	Disabled
QUEUE_REGISTRY_SIZE	8
USE_APPLICATION_TASK_...	Disabled
ENABLE_BACKWARD_CO...	Enabled
USE_PORT_OPTIMISED_T...	Enabled
USE_TICKLESS_IDLE	Disabled
USE_TASK_NOTIFICATIONS	Enabled
RECORD_STACK_HIGH_A...	Disabled
Memory management settings	
Memory Allocation	Dynamic / Static
TOTAL_HEAP_SIZE	10000 Bytes

Select the Advanced Setting tab and set USE_NEWLIB_REENTRANT to Enabled.



Newlib settings (see parameter desc...
USE_NEWLIB_REENTRANT Enabled
Project settings (see parameter des...
Use FW pack heap file Enabled

Enable multi-threaded support in the Project Manager tab.

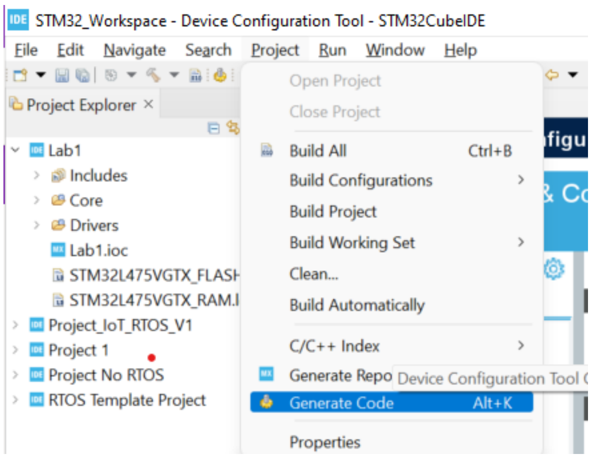


Project Settings
Project Name Lab1
Project Location \nokeeffe\OneDrive - Atlantic TU\Course Material\Embedded RTOS\Lab Material\2024\Workspcae_24
Application Structure Advanced ☐ Do not generate the main()
Toolchain Folder Location C:\Users\nokeeffe\OneDrive - Atlantic TU\Course Material\Embedded RTOS\Lab Material\2024\Workspcae_24\Lab1\
Toolchain / IDE STM32CubeIDE ☑ Generate Under Root

Linker Settings
Minimum Heap Size 0x200
Minimum Stack Size 0x400

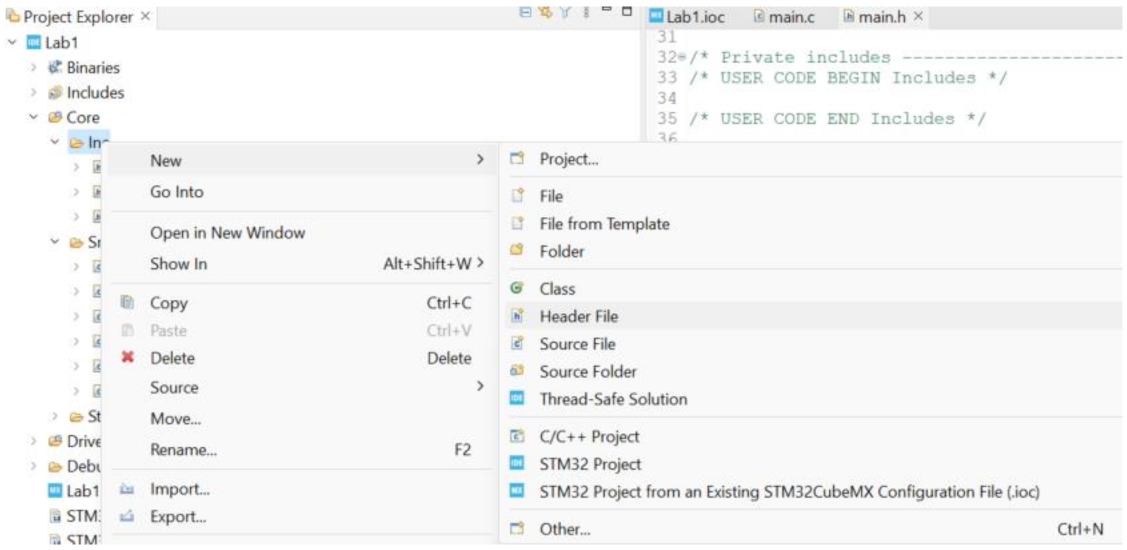
Thread-safe Settings
Cortex-M4NS
☑ Enable multi-threaded support
Thread-safe Locking Strategy Default – Mapping suitable strategy depending on RTOS selection.

Generate the configuration code.

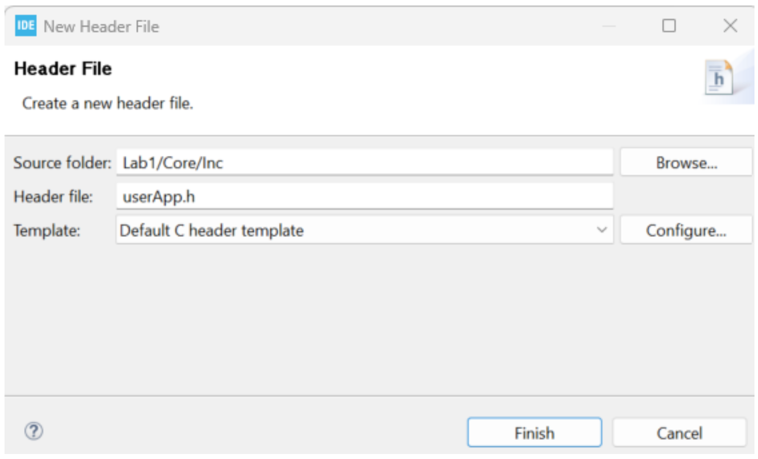


Step 3: Creating a User Header and C File

Add a header file called userApp.h to the Inc folder.



```
/* Private includes ----------------------
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */
```



Header File

Create a new header file.

Source folder: Lab1/Core/Inc

Header file: userApp.h

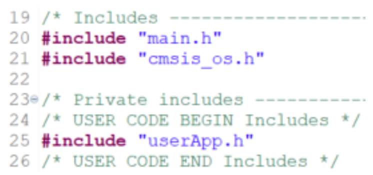
Template: Default C header template

Add the file name and select Finish.

Add the lines of code as shown below.

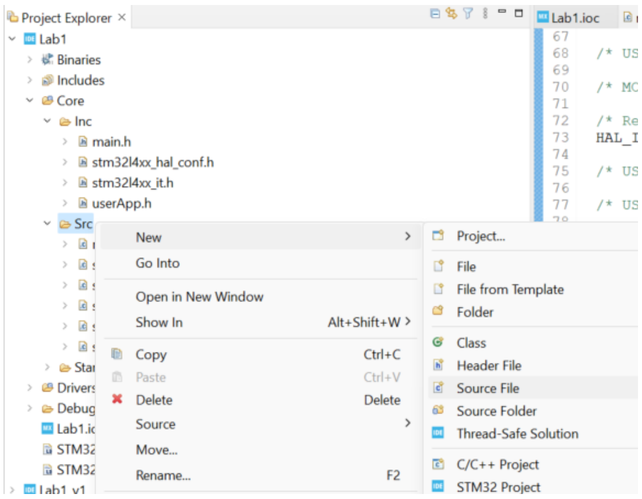
```
#ifndef INC_USERAPP_H_
#define INC_USERAPP_H_

#include "main.h"

void userApp();

#endif /* INC_USERAPP_H_ */
```

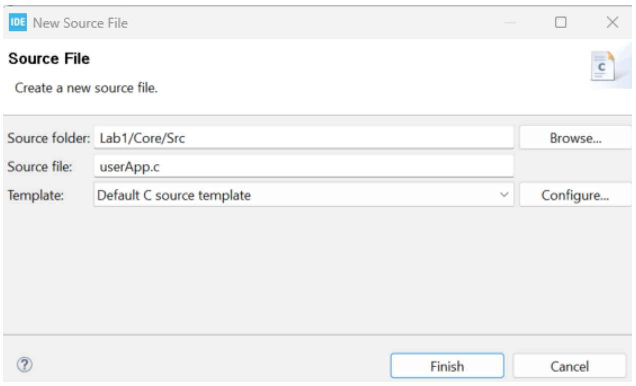
Open main.c and include the file userApp.h as shown below.



```
/* Includes ------------------
#include "main.h"
#include "cmsis_os.h"

/* Private includes ----------
/* USER CODE BEGIN Includes */
#include "userApp.h"
/* USER CODE END Includes */
```

Add a source file called userApp.c to the Src folder.




New Source File

Source File

Create a new source file.

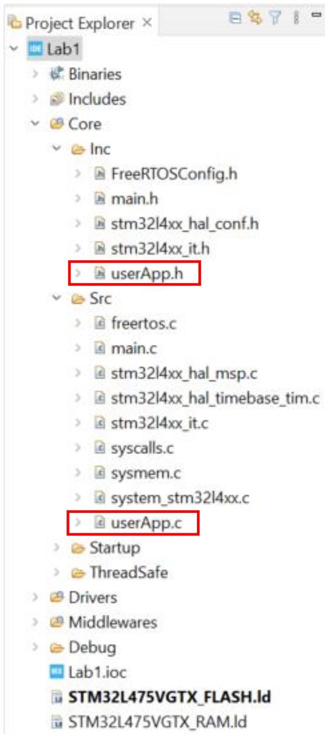
Source folder: Lab1/Core/Src

Source file: userApp.c

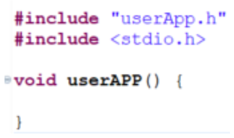
Template: Default C source template

Enter the file name and select Finish.

Your file structure should now look as follows: -

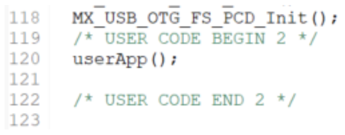


Add the following code to the file userApp.c.



```
#include "userApp.h"
#include <stdio.h>

void userAPP() {

}
```

We now need to call the userApp() function in main.c.



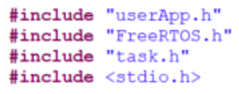
```
MX_USB_OTG_FS_PCD_Init();
/* USER CODE BEGIN 2 */
userApp();

/* USER CODE END 2 */
```

Any user application code will now be added to the userApp.c file.

Step 4: Writing FreeRTOS Code

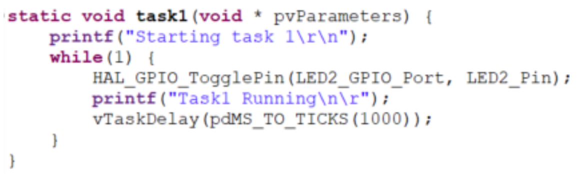
We can now start to add some FreeRTOS code to the file userApp.c.

Include the FreeRTOS header files.



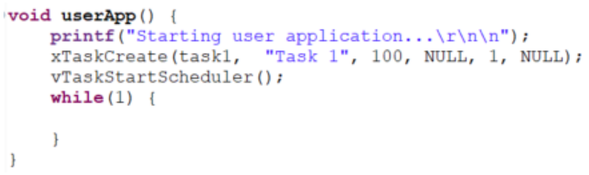
```
#include "userApp.h"
#include "FreeRTOS.h"
#include "task.h"
#include <stdio.h>
```

Add a task function.



```
static void task1(void * pvParameters) {
    printf("Starting task 1\r\n");
    while(1) {
        HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
        printf("Task1 Running\n\r");
        vTaskDelay(pdMS_TO_TICKS(1000));
    }
}
```

Create the task and start the scheduler in the userApp() function.



```
void userApp() {
    printf("Starting user application...\r\n\n");
    xTaskCreate(task1,  "Task 1", 100, NULL, 1, NULL);
    vTaskStartScheduler();
    while(1) {

    }
}
```

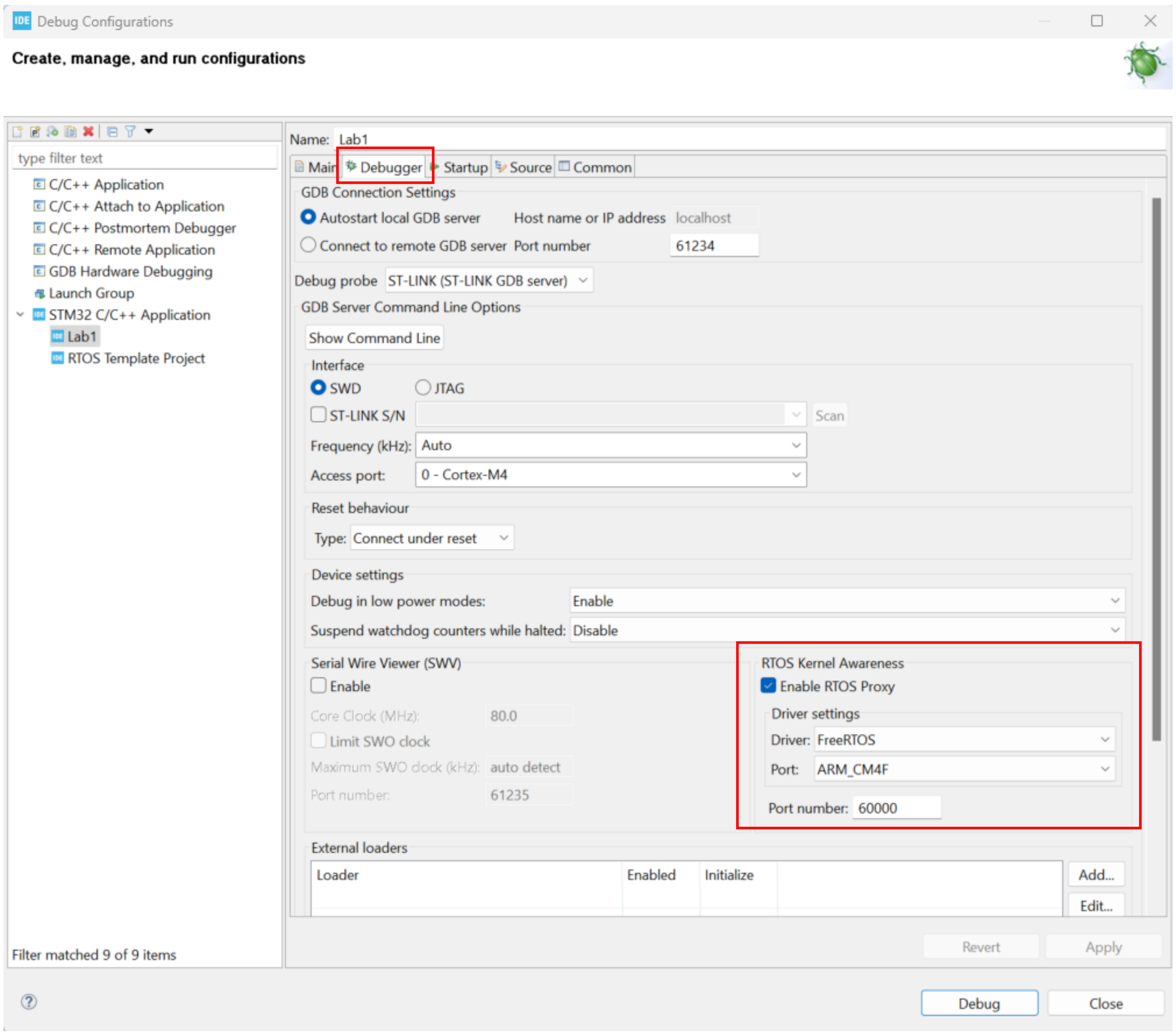
Build the project and fix any errors.

Debug the Project

Before running your code, the debugger should be configured for thread aware debugging.

Select Debug configurations from the Run menu and select the Lab1 project.

Configure the settings as shown in the Debugger tab and select Debug.


Debug Configurations

Create, manage, and run configurations

type filter text
- C/C++ Application
- C/C++ Attach to Application
- C/C++ Postmortem Debugger
- C/C++ Remote Application
- GDB Hardware Debugging
- Launch Group
- STM32 C/C++ Application
  - Lab1
  - RTOS Template Project

Filter matched 9 of 9 items

Name: Lab1

Main | Debugger | Startup | Source | Common

GDB Connection Settings
- Autostart local GDB server (selected) — Host name or IP address: localhost
- Connect to remote GDB server — Port number: 61234

Debug probe: ST-LINK (ST-LINK GDB server)

GDB Server Command Line Options

Show Command Line

Interface: SWD (selected) / JTAG

ST-LINK S/N: (unchecked) — Scan

Frequency (kHz): Auto

Access port: 0 - Cortex-M4

Reset behaviour — Type: Connect under reset

Device settings
- Debug in low power modes: Enable
- Suspend watchdog counters while halted: Disable

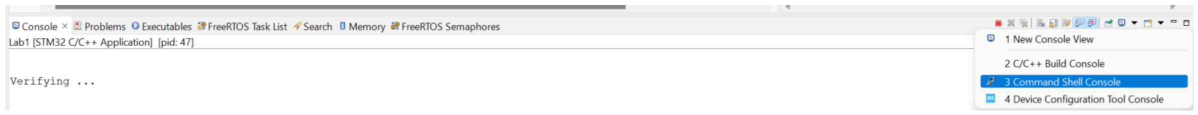
Serial Wire Viewer (SWV)
- Enable (unchecked)
- Core Clock (MHz): 80.0
- Limit SWO clock (unchecked)
- Maximum SWO clock (kHz): auto detect
- Port number: 61235

RTOS Kernel Awareness
- Enable RTOS Proxy (checked)
- Driver settings
  - Driver: FreeRTOS
  - Port: ARM_CM4F
- Port number: 60000

External loaders

Loader	Enabled	Initialize

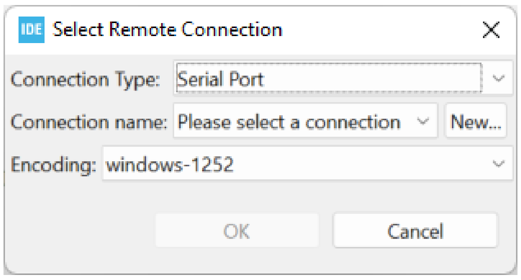
Start a debug session and add a serial terminal console. This will be used to display printf messages.



Console × Problems Executables FreeRTOS Task List Search Memory FreeRTOS Semaphores

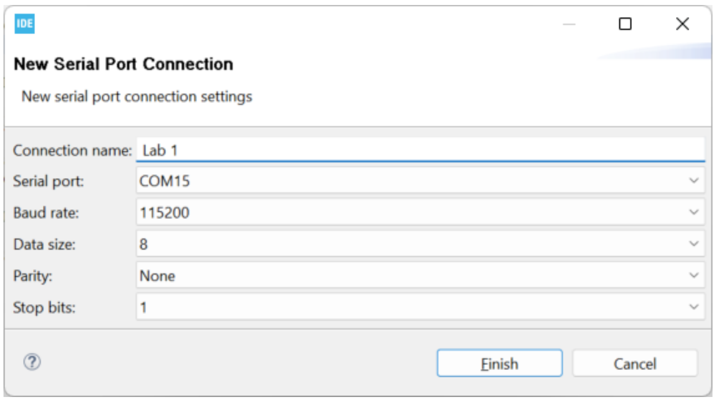
Lab1 [STM32 C/C++ Application] [pid: 47]

```
Verifying ...
```



Select Remote Connection

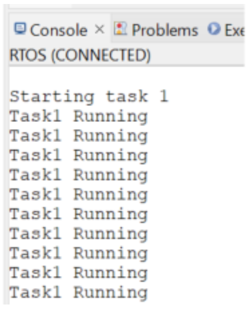
Connection Type:	Serial Port
Connection name:	Please select a connection
Encoding:	windows-1252



New Serial Port Connection

New serial port connection settings

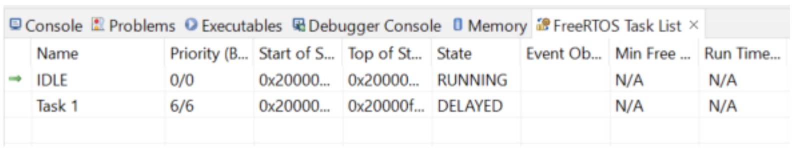
Connection name:	Lab 1
Serial port:	COM15
Baud rate:	115200
Data size:	8
Parity:	None
Stop bits:	1



Console × Problems Exe

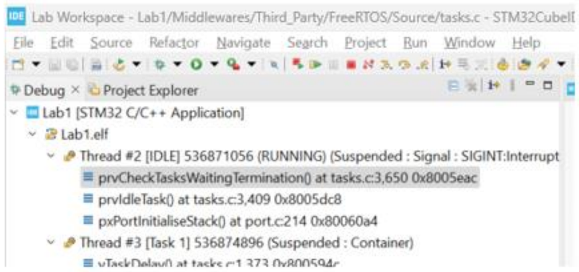
RTOS (CONNECTED)

```
Starting task 1
Task1 Running
Task1 Running
Task1 Running
Task1 Running
Task1 Running
Task1 Running
Task1 Running
Task1 Running
Task1 Running
Task1 Running
```



Console Problems Executables Debugger Console Memory FreeRTOS Task List ×

	Name	Priority (B...	Start of S...	Top of St...	State	Event Ob...	Min Free ...	Run Time...
➡	IDLE	0/0	0x20000...	0x20000...	RUNNING		N/A	N/A
	Task 1	6/6	0x20000...	0x20000f...	DELAYED		N/A	N/A



Lab Workspace - Lab1/Middlewares/Third_Party/FreeRTOS/Source/tasks.c - STM32CubeI

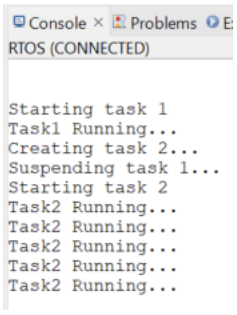
File Edit Source Refactor Navigate Search Project Run Window Help
Debug × Project Explorer

```
Lab1 [STM32 C/C++ Application]
  Lab1.elf
    Thread #2 [IDLE] 536871056 (RUNNING) (Suspended : Signal : SIGINT:Interrupt
      prvCheckTasksWaitingTermination() at tasks.c:3,650 0x8005eac
      prvIdleTask() at tasks.c:3,409 0x8005dc8
      pxPortInitialiseStack() at port.c:214 0x80060a4
    Thread #3 [Task 1] 536874896 (Suspended : Container)
      vTaskDelay() at tasks.c:1,373 0x800594c
```

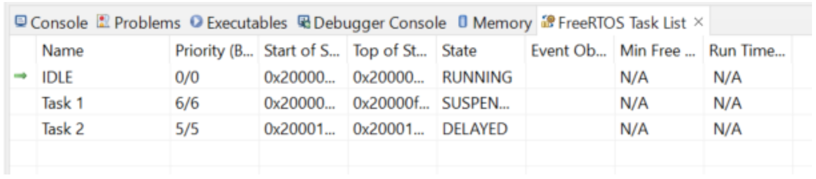
Part B: Adding a Second Task

Task 1 should toggle the LED 10 times at 500ms intervals. Task 1 should then create task 2 before suspending itself. The task2 priority should be lower than task 1.

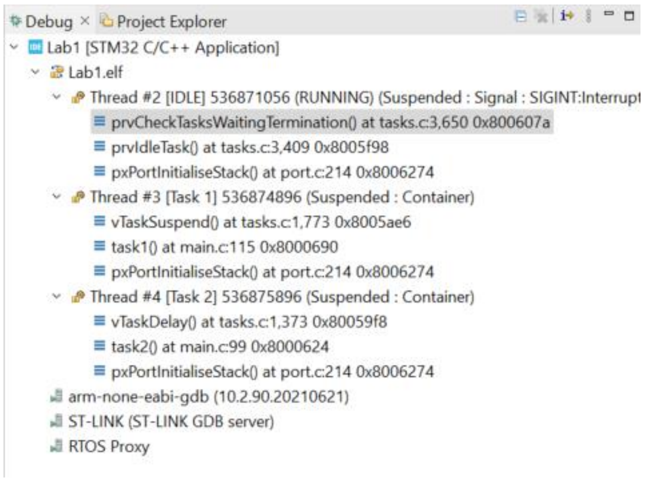
Task 2 should continually toggle the LED at 200ms intervals.



```
Starting task 1
Task1 Running...
Creating task 2...
Suspending task 1...
Starting task 2
Task2 Running...
Task2 Running...
Task2 Running...
Task2 Running...
Task2 Running...
```



Name	Priority (B...	Start of S...	Top of St...	State	Event Ob...	Min Free ...	Run Time...
IDLE	0/0	0x20000...	0x20000...	RUNNING		N/A	N/A
Task 1	6/6	0x20000...	0x20000f...	SUSPEN...		N/A	N/A
Task 2	5/5	0x20001...	0x20001...	DELAYED		N/A	N/A



```
Lab1 [STM32 C/C++ Application]
  Lab1.elf
    Thread #2 [IDLE] 536871056 (RUNNING) (Suspended : Signal : SIGINT:Interrupt
      prvCheckTasksWaitingTermination() at tasks.c:3,650 0x800607a
      prvIdleTask() at tasks.c:3,409 0x8005f98
      pxPortInitialiseStack() at port.c:214 0x8006274
    Thread #3 [Task 1] 536874896 (Suspended : Container)
      vTaskSuspend() at tasks.c:1,773 0x8005ae6
      task1() at main.c:115 0x8000690
      pxPortInitialiseStack() at port.c:214 0x8006274
    Thread #4 [Task 2] 536875896 (Suspended : Container)
      vTaskDelay() at tasks.c:1,373 0x80059f8
      task2() at main.c:99 0x8000624
      pxPortInitialiseStack() at port.c:214 0x8006274
  arm-none-eabi-gdb (10.2.90.20210621)
  ST-LINK (ST-LINK GDB server)
  RTOS Proxy
```