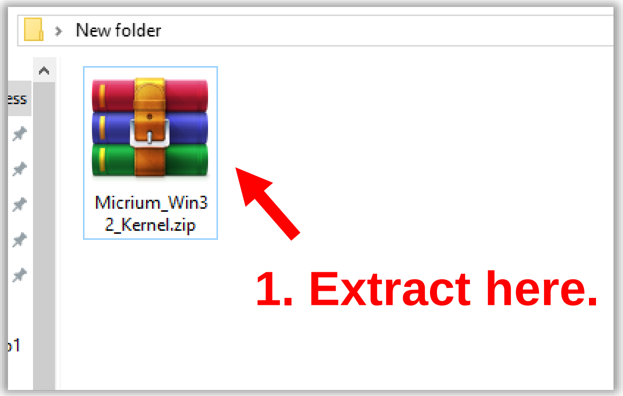
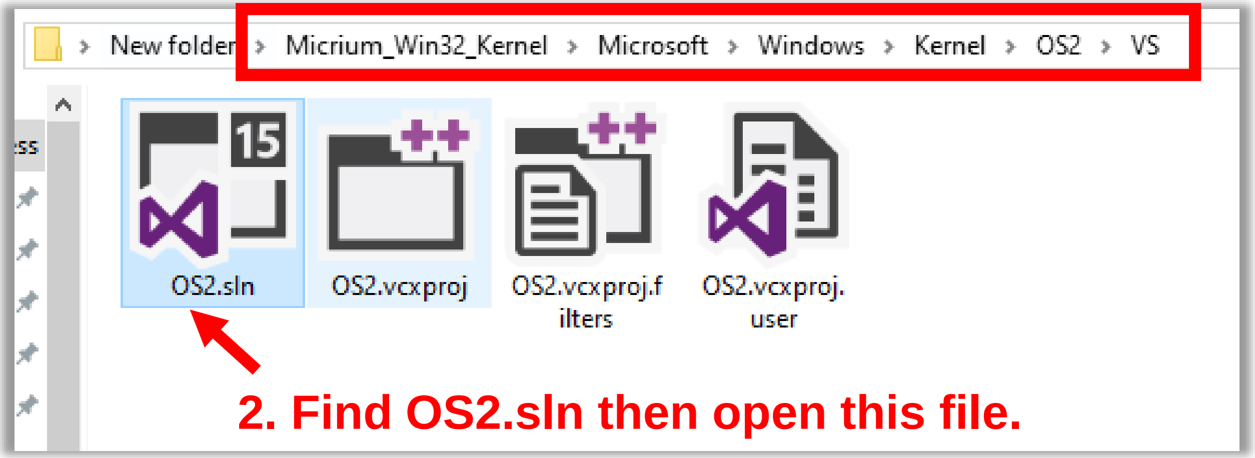
Run µC/OS-II

2020.10.14

Open example project

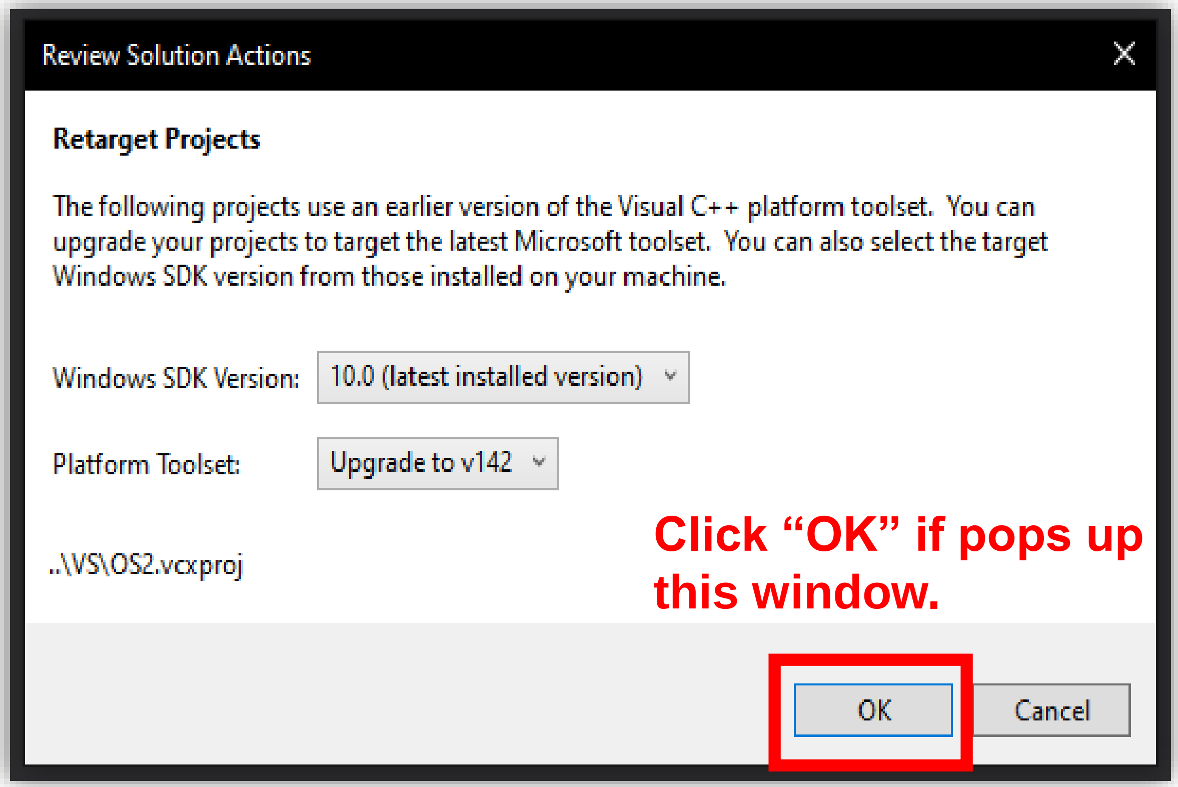


1. Extract here.


New folder › Micrium_Win32_Kernel › Microsoft › Windows › Kernel › OS2 › VS
OS2.sln
OS2.vcxproj
OS2.vcxproj.filters
OS2.vcxproj.user

2. Find OS2.sln then open this file.

Open example project



Review Solution Actions

Retarget Projects

The following projects use an earlier version of the Visual C++ platform toolset. You can upgrade your projects to target the latest Microsoft toolset. You can also select the target Windows SDK version from those installed on your machine.

Windows SDK Version: 10.0 (latest installed version)

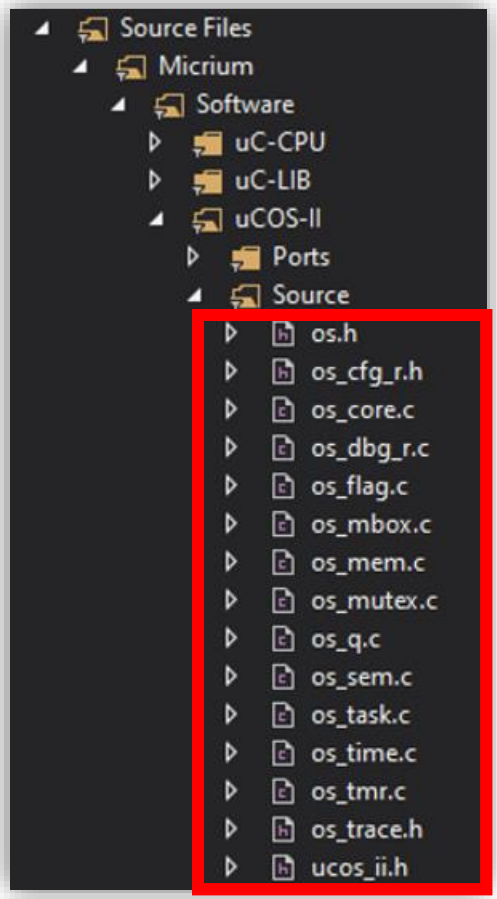
Platform Toolset: Upgrade to v142

..\VS\OS2.vcxproj

Click "OK" if pops up this window.

Find µC/OS-II source code

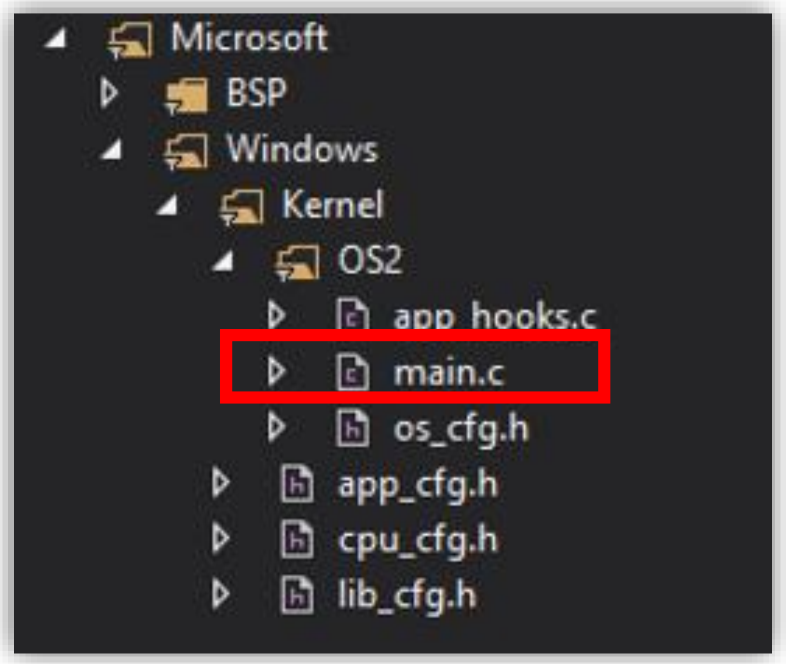
- Source code path: Source Files\Micrium\
  Software\uCOS-II\Source



- Source Files
  - Micrium
    - Software
      - uC-CPU
      - uC-LIB
      - uCOS-II
        - Ports
        - Source
          - os.h
          - os_cfg_r.h
          - os_core.c
          - os_dbg_r.c
          - os_flag.c
          - os_mbox.c
          - os_mem.c
          - os_mutex.c
          - os_q.c
          - os_sem.c
          - os_task.c
          - os_time.c
          - os_tmr.c
          - os_trace.h
          - ucos_ii.h

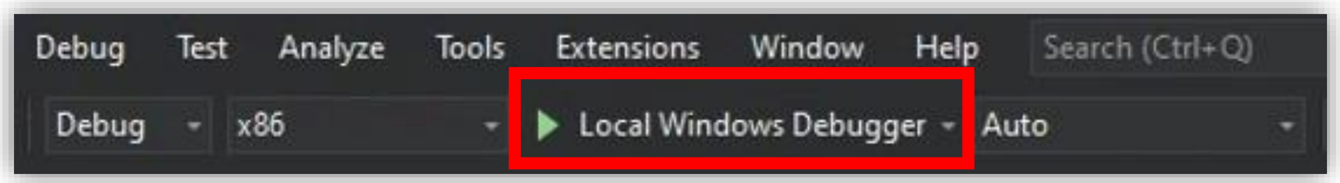
Find µC/OS-II source code

- Main.c path: Microsoft\Windows\Kernel\OS2



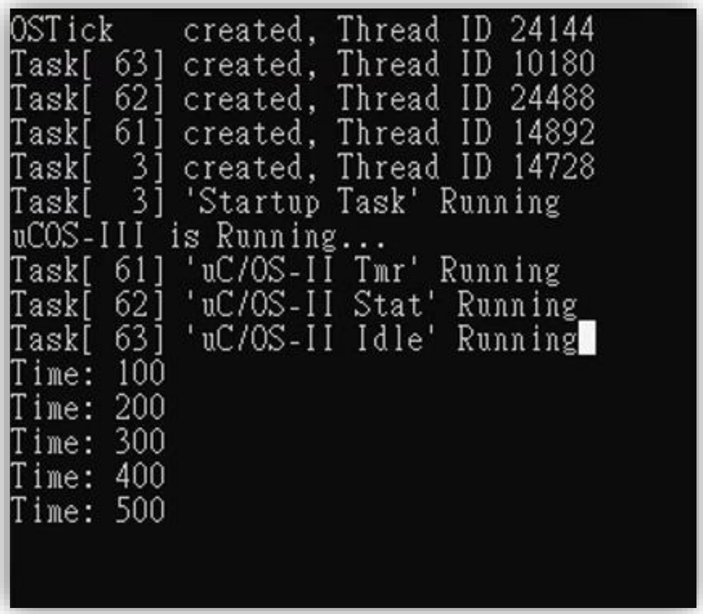
Run example project

- Open main.c and then click “Local Windows Debugger” or press F5.



Run example project

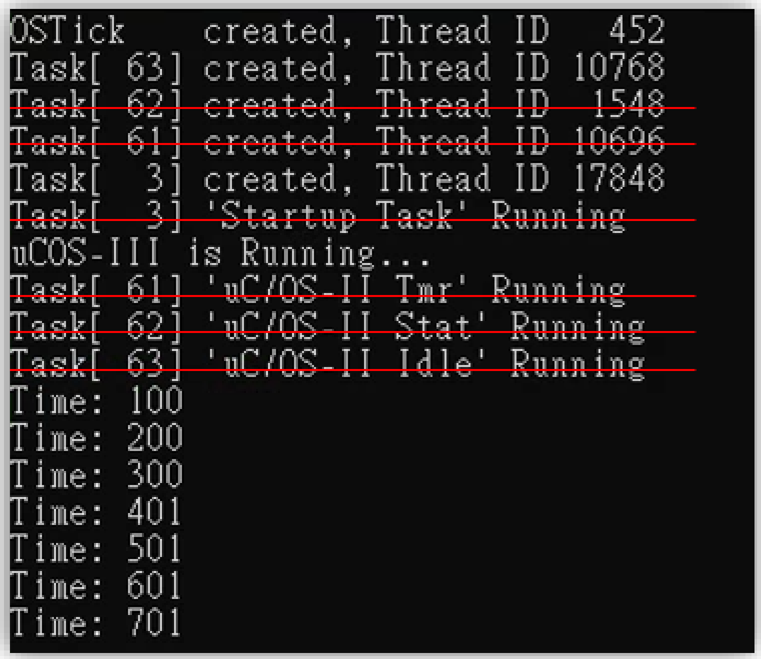
- You can see the tasks information in command prompt if the project has been run successfully.



```
OSTick    created, Thread ID 24144
Task[ 63] created, Thread ID 10180
Task[ 62] created, Thread ID 24488
Task[ 61] created, Thread ID 14892
Task[  3] created, Thread ID 14728
Task[  3] 'Startup Task' Running
uCOS-III is Running...
Task[ 61] 'uC/OS-II Tmr' Running
Task[ 62] 'uC/OS-II Stat' Running
Task[ 63] 'uC/OS-II Idle' Running
Time: 100
Time: 200
Time: 300
Time: 400
Time: 500
```

Modify example project

- You need to disable two tasks and messages before submitting your project.

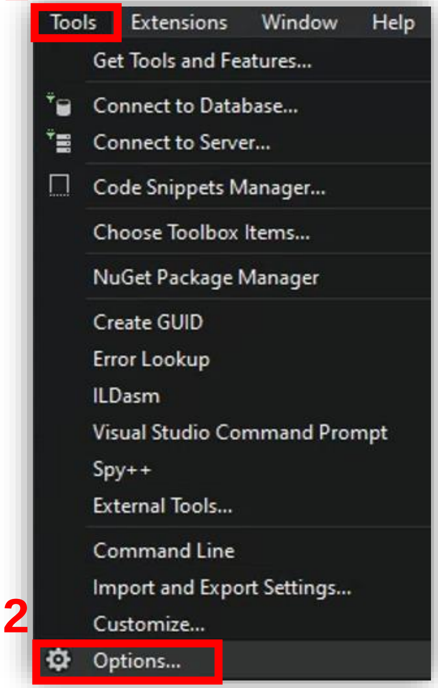


```
OSTick     created, Thread ID   452
Task[ 63] created, Thread ID 10768
~~Task[ 62] created, Thread ID  1548~~
~~Task[ 61] created, Thread ID 10696~~
Task[  3] created, Thread ID 17848
~~Task[  3] 'Startup Task' Running~~
uCOS-III is Running...
~~Task[ 61] 'uC/OS-II Tmr' Running~~
~~Task[ 62] 'uC/OS-II Stat' Running~~
~~Task[ 63] 'uC/OS-II Idle' Running~~
Time: 100
Time: 200
Time: 300
Time: 401
Time: 501
Time: 601
Time: 701
```

Modify example project

- First, open the line numbers.

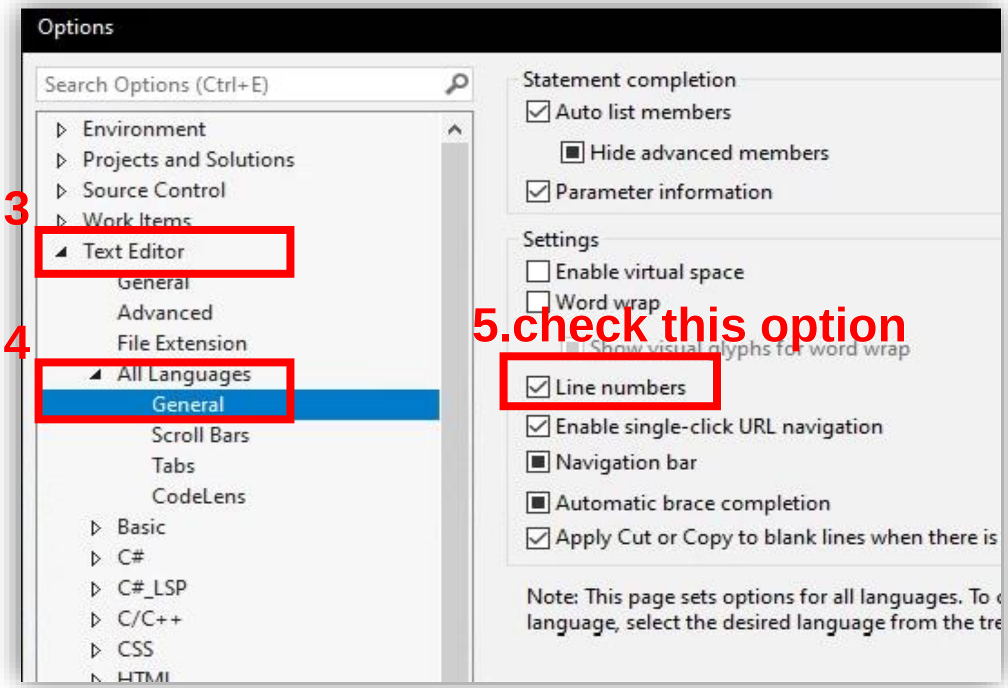
1



2

3

4



Statement completion

☑ Auto list members

■ Hide advanced members

☑ Parameter information

Settings

☐ Enable virtual space

☐ Word wrap

■ Show visual glyphs for word wrap

☑ Line numbers

☑ Enable single-click URL navigation

■ Navigation bar

■ Automatic brace completion

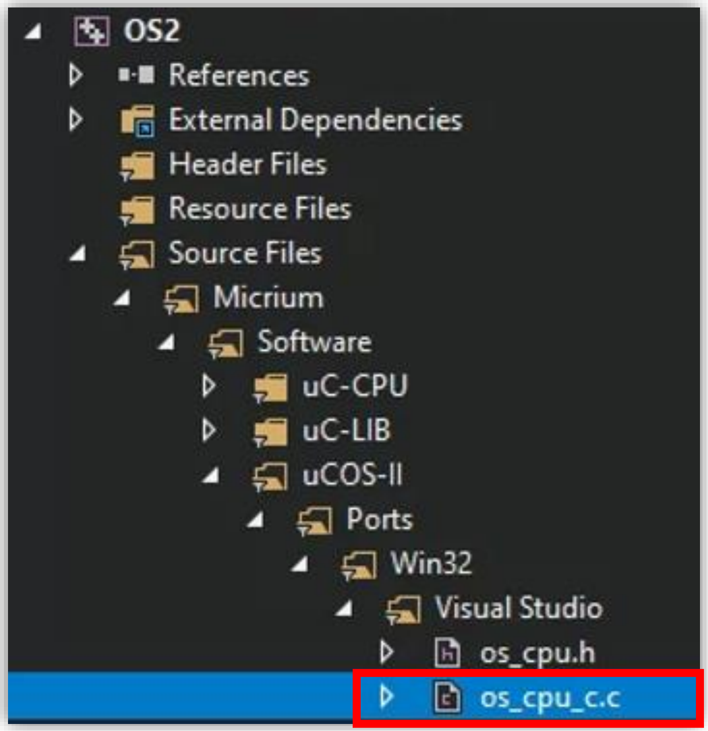
☑ Apply Cut or Copy to blank lines when there is

Note: This page sets options for all languages. To c
language, select the desired language from the tre

5.check this option

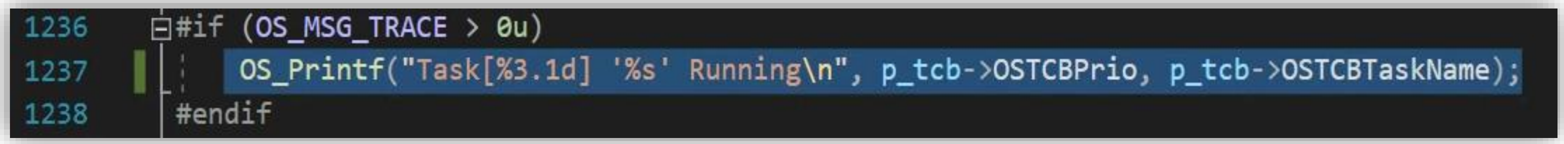
Modify example project

- Find os_cpu_c.c and then open it.
- os_cpu_c.c path:Micrium\Software\uCOS-II\Ports\Win32\Visual Studio

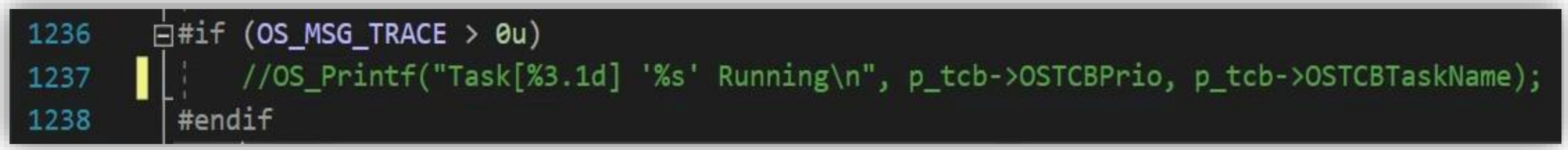


Modify example project

- Comment out the 1237th line.



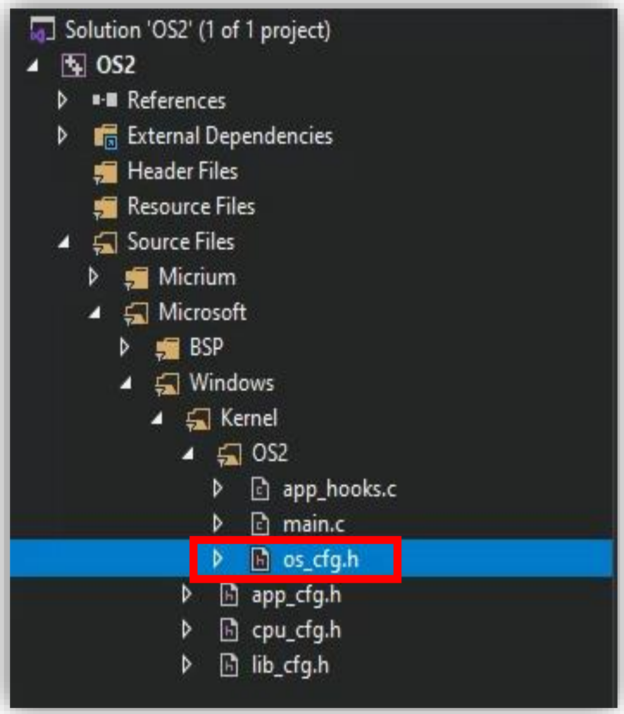
```
1236  #if (OS_MSG_TRACE > 0u)
1237      OS_Printf("Task[%3.1d] '%s' Running\n", p_tcb->OSTCBPrio, p_tcb->OSTCBTaskName);
1238  #endif
```



```
1236  #if (OS_MSG_TRACE > 0u)
1237      //OS_Printf("Task[%3.1d] '%s' Running\n", p_tcb->OSTCBPrio, p_tcb->OSTCBTaskName);
1238  #endif
```

Modify example project

- Next, find os_cfg.h and then open it.



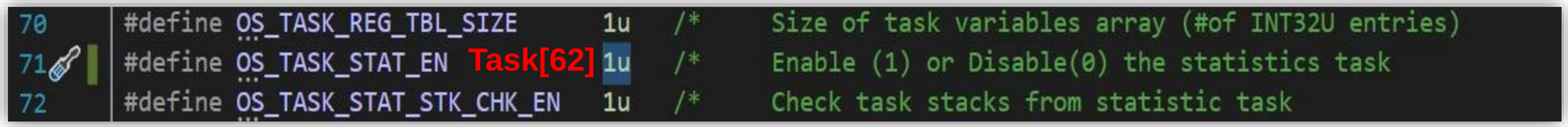
```
Solution 'OS2' (1 of 1 project)
  OS2
    References
    External Dependencies
    Header Files
    Resource Files
    Source Files
      Micrium
      Microsoft
        BSP
        Windows
          Kernel
            OS2
              app_hooks.c
              main.c
              os_cfg.h
            app_cfg.h
            cpu_cfg.h
            lib_cfg.h
```

- os_cfg.h path:

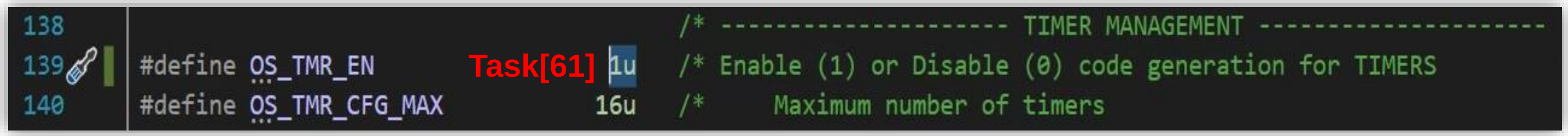
Micrium_Win32_Kernel\Microsoft\
Windows\Kernel\OS2

Modify example project

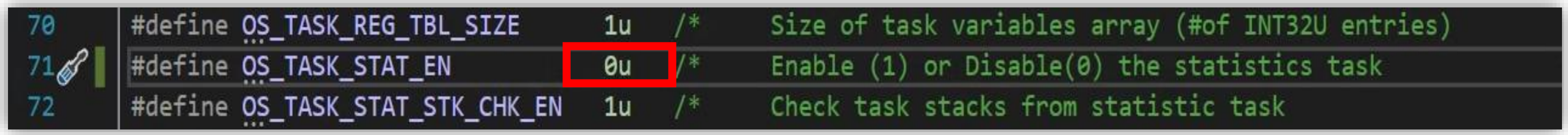
- Go to the 71th line and 139th line and then DISABLE them.



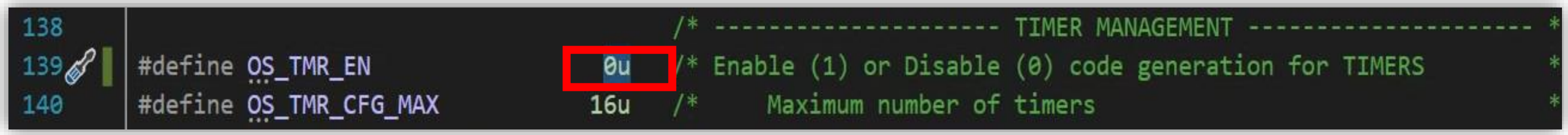
```
70   #define OS_TASK_REG_TBL_SIZE       1u   /*      Size of task variables array (#of INT32U entries)
71   #define OS_TASK_STAT_EN  Task[62]  1u   /*      Enable (1) or Disable(0) the statistics task
72   #define OS_TASK_STAT_STK_CHK_EN    1u   /*      Check task stacks from statistic task
```



```
138                                          /* -------------------- TIMER MANAGEMENT --------------------
139  #define OS_TMR_EN        Task[61]  1u   /* Enable (1) or Disable (0) code generation for TIMERS
140  #define OS_TMR_CFG_MAX            16u   /*      Maximum number of timers
```



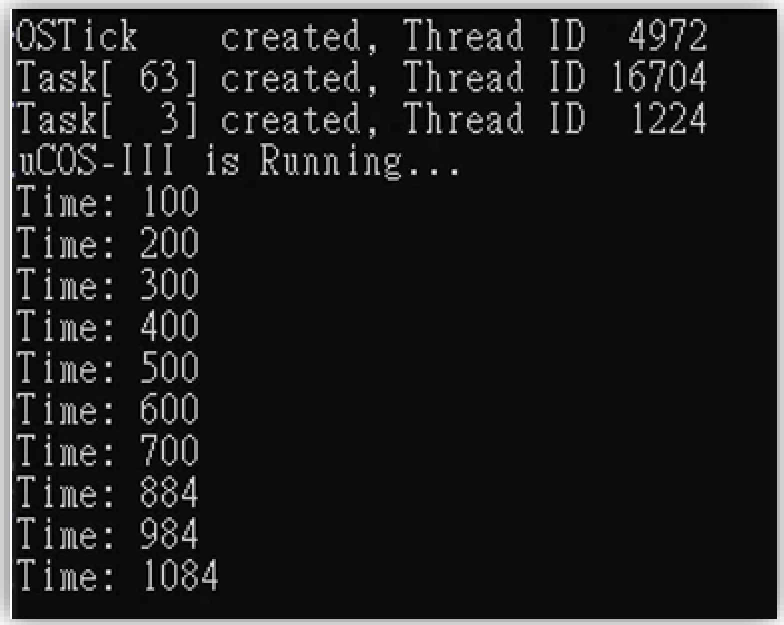
```
70   #define OS_TASK_REG_TBL_SIZE       1u   /*      Size of task variables array (#of INT32U entries)
71   #define OS_TASK_STAT_EN            0u   /*      Enable (1) or Disable(0) the statistics task
72   #define OS_TASK_STAT_STK_CHK_EN    1u   /*      Check task stacks from statistic task
```



```
138                                          /* -------------------- TIMER MANAGEMENT -------------------- *
139  #define OS_TMR_EN                  0u   /* Enable (1) or Disable (0) code generation for TIMERS       *
140  #define OS_TMR_CFG_MAX            16u   /*      Maximum number of timers                             *
```

Modify example project

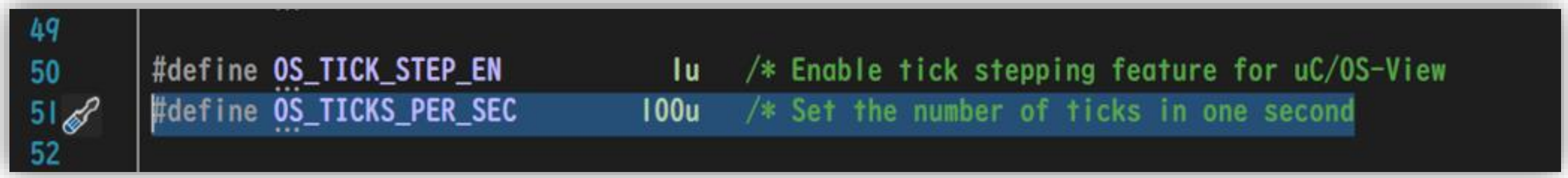
- Finally, rerun this project and you can see the modified tasks information.



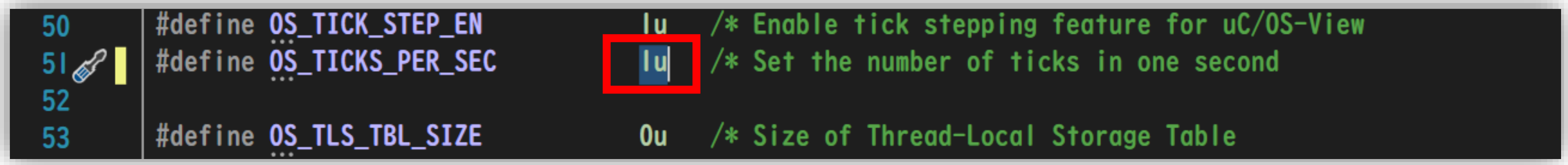
```
OSTick     created, Thread ID  4972
Task[ 63] created, Thread ID 16704
Task[  3] created, Thread ID  1224
uCOS-III is Running...
Time: 100
Time: 200
Time: 300
Time: 400
Time: 500
Time: 600
Time: 700
Time: 884
Time: 984
Time: 1084
```

Create the initial tasks of HW1

- First, go to os_cfg.h and then set 1 tick in 1 second.



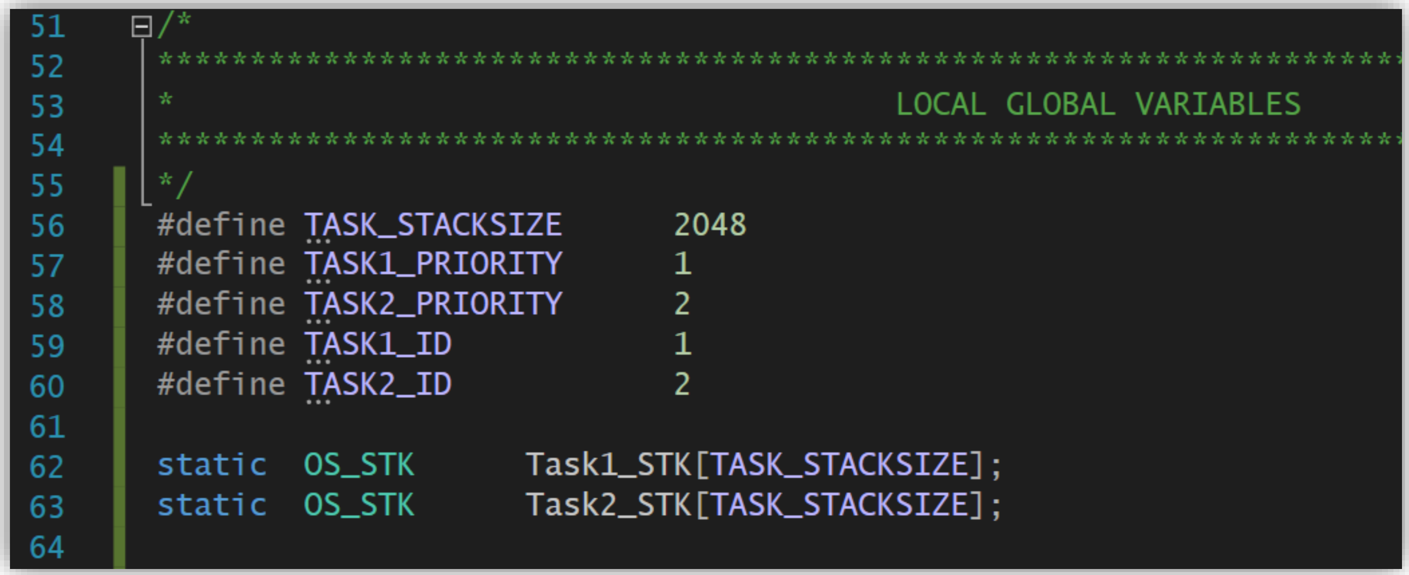
```
#define OS_TICK_STEP_EN             1u    /* Enable tick stepping feature for uC/OS-View
#define OS_TICKS_PER_SEC          100u    /* Set the number of ticks in one second
```



```
#define OS_TICK_STEP_EN             1u    /* Enable tick stepping feature for uC/OS-View
#define OS_TICKS_PER_SEC            1u    /* Set the number of ticks in one second

#define OS_TLS_TBL_SIZE             0u    /* Size of Thread-Local Storage Table
```

Create the initial tasks of HW1

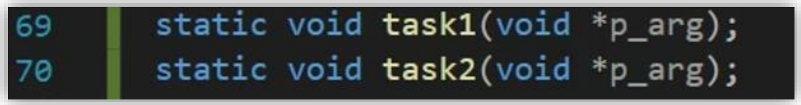
- You need to declare the task **priority, ID, stack size** and **stack space** as **GLOBAL** variables in main.c or app_cfg.h.



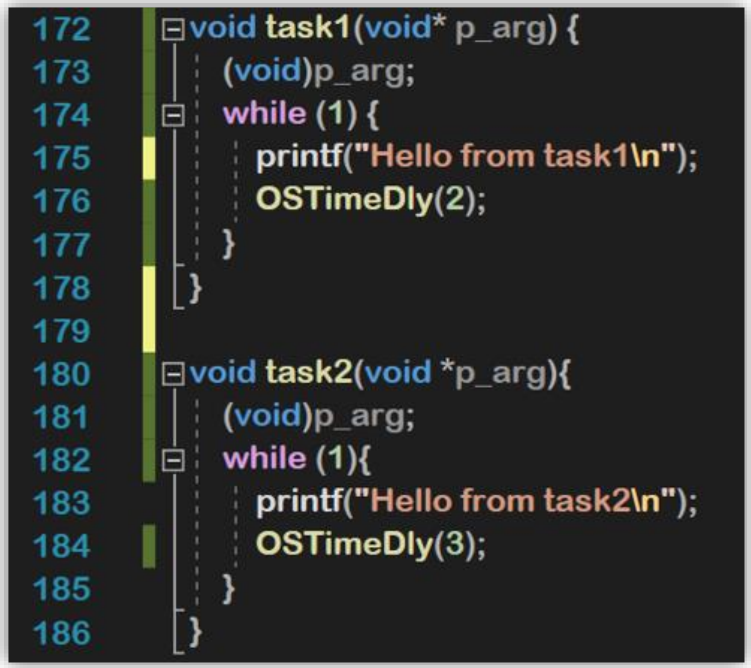
```
/*
********************************************************************
*                                       LOCAL GLOBAL VARIABLES
********************************************************************
*/
#define TASK_STACKSIZE      2048
#define TASK1_PRIORITY      1
#define TASK2_PRIORITY      2
#define TASK1_ID            1
#define TASK2_ID            2

static  OS_STK      Task1_STK[TASK_STACKSIZE];
static  OS_STK      Task2_STK[TASK_STACKSIZE];
```

Create the initial tasks of HW1

- Declare and define task function.



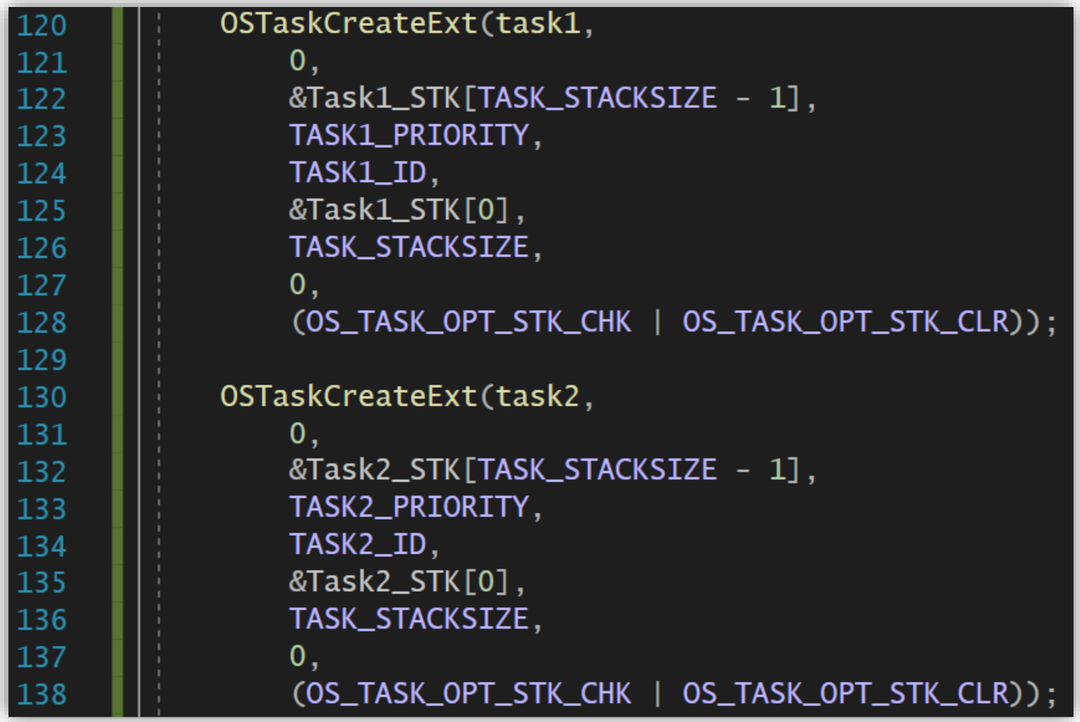
```c
static void task1(void *p_arg);
static void task2(void *p_arg);
```



```c
void task1(void* p_arg) {
    (void)p_arg;
    while (1) {
        printf("Hello from task1\n");
        OSTimeDly(2);
    }
}

void task2(void *p_arg){
    (void)p_arg;
    while (1){
        printf("Hello from task2\n");
        OSTimeDly(3);
    }
}
```

Create the initial tasks of HW1

- Call ***OSTaskCreateExt(…)*** in main function to create a new task.

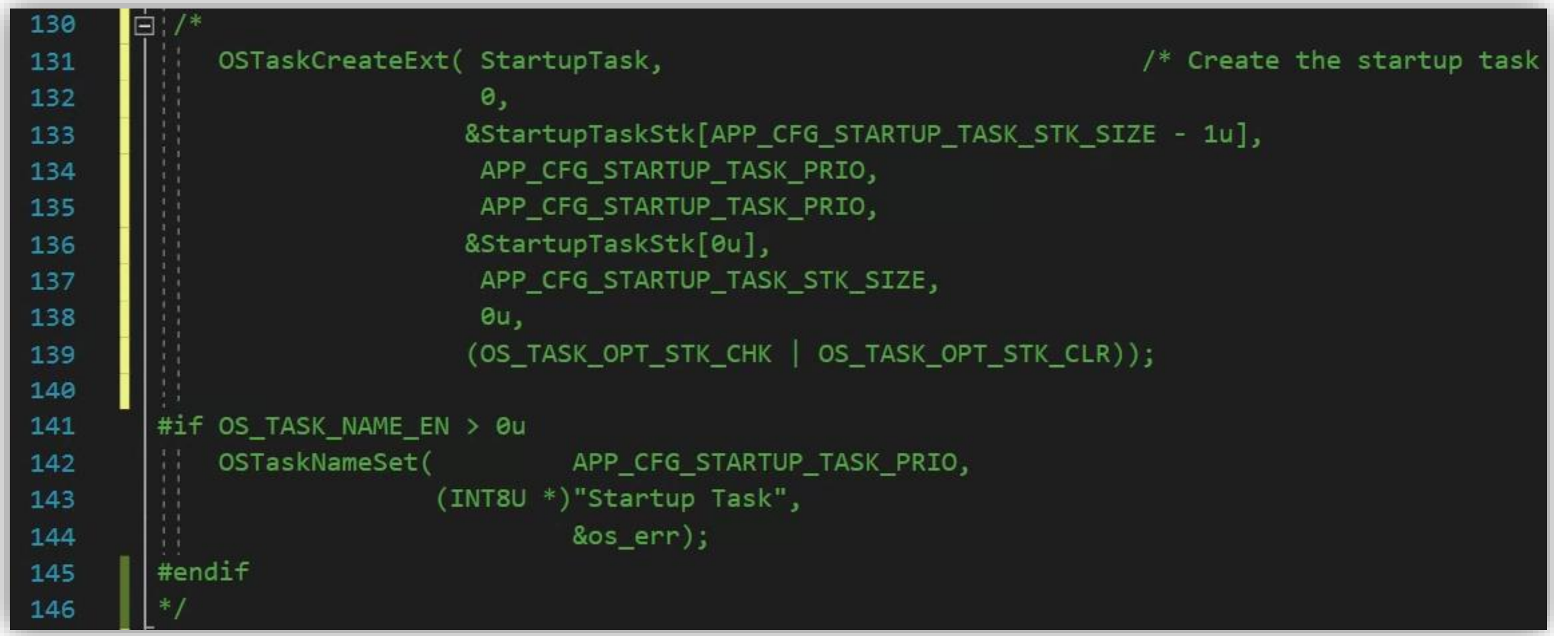


```
    OSTaskCreateExt(task1,
        0,
        &Task1_STK[TASK_STACKSIZE - 1],
        TASK1_PRIORITY,
        TASK1_ID,
        &Task1_STK[0],
        TASK_STACKSIZE,
        0,
        (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));

    OSTaskCreateExt(task2,
        0,
        &Task2_STK[TASK_STACKSIZE - 1],
        TASK2_PRIORITY,
        TASK2_ID,
        &Task2_STK[0],
        TASK_STACKSIZE,
        0,
        (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
```

You can find its definition by right-click its name.

Create the initial tasks of HW1

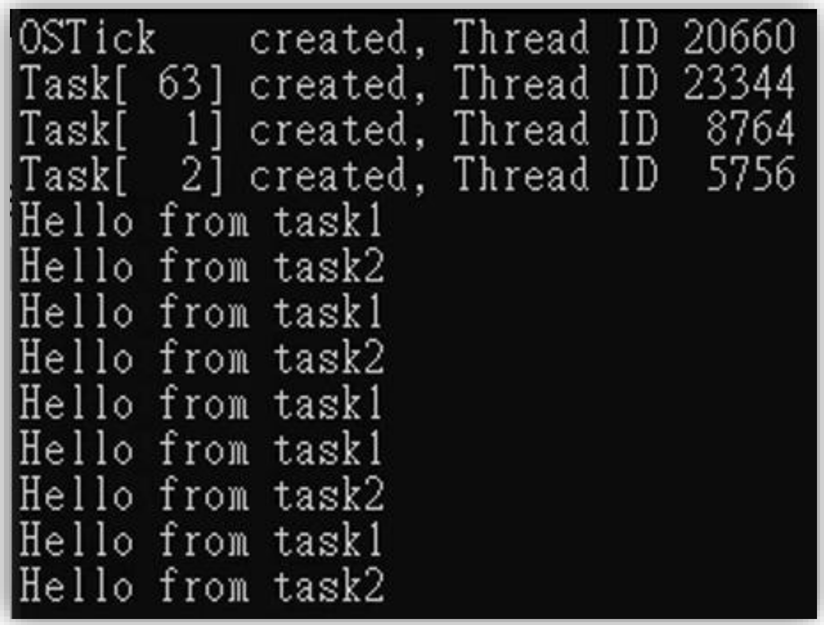
- Finally, comment out ***“create the startup task”*** and ***“OSTaskNameSet(…)”***



```
 /*
    OSTaskCreateExt( StartupTask,                                           /* Create the startup task
                     0,
                    &StartupTaskStk[APP_CFG_STARTUP_TASK_STK_SIZE - 1u],
                     APP_CFG_STARTUP_TASK_PRIO,
                     APP_CFG_STARTUP_TASK_PRIO,
                    &StartupTaskStk[0u],
                     APP_CFG_STARTUP_TASK_STK_SIZE,
                     0u,
                    (OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));

#if OS_TASK_NAME_EN > 0u
    OSTaskNameSet(         APP_CFG_STARTUP_TASK_PRIO,
                  (INT8U *)"Startup Task",
                           &os_err);
#endif
*/
```

Create the initial tasks of HW1

- Then rerun this project, you can see two tasks information in command prompt.

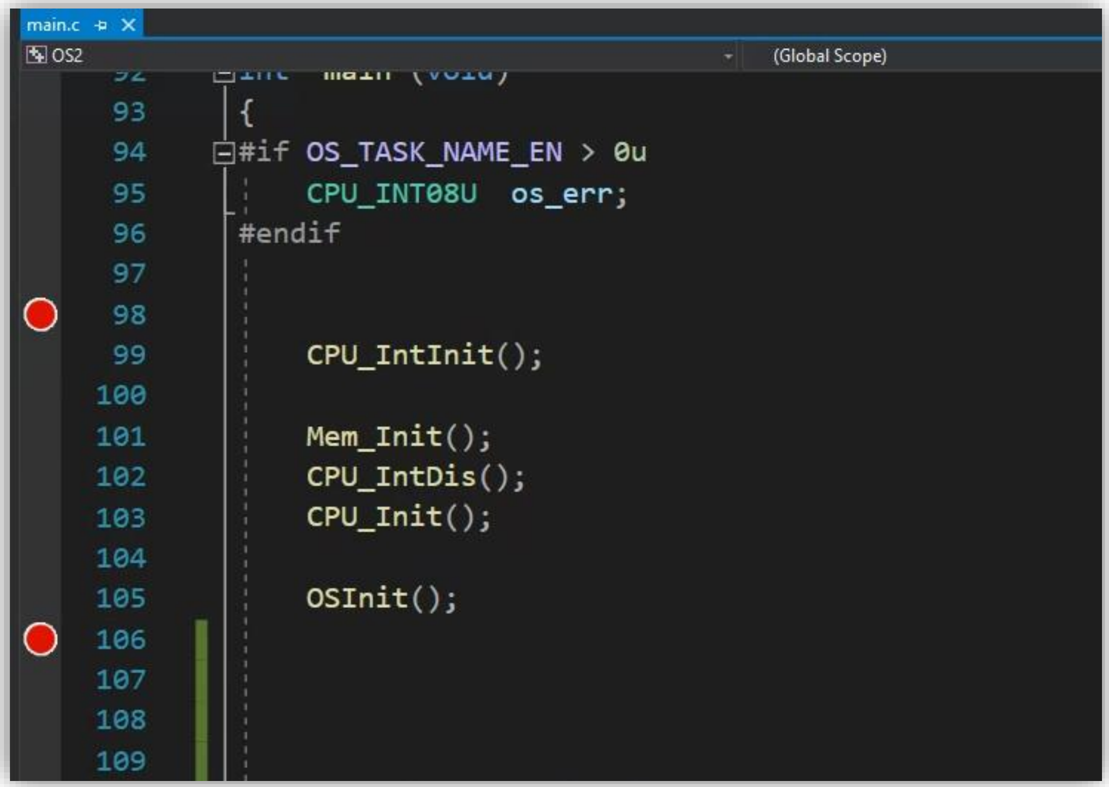


```
OSTick    created, Thread ID 20660
Task[ 63] created, Thread ID 23344
Task[  1] created, Thread ID  8764
Task[  2] created, Thread ID  5756
Hello from task1
Hello from task2
Hello from task1
Hello from task2
Hello from task1
Hello from task1
Hello from task2
Hello from task1
Hello from task2
```

Now you can use this project to do HW1!

Debug mode

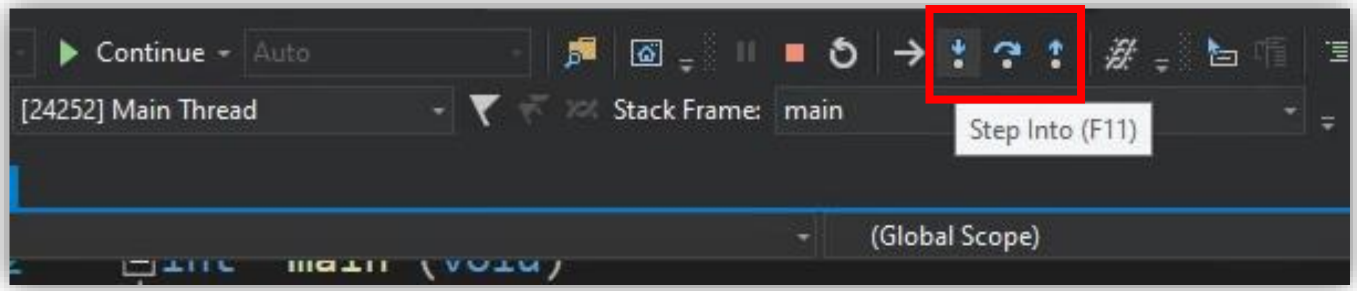
- First, add breakpoints by left-click.



```
int main (void)
{
#if OS_TASK_NAME_EN > 0u
    CPU_INT08U  os_err;
#endif


    CPU_IntInit();

    Mem_Init();
    CPU_IntDis();
    CPU_Init();

    OSInit();
```

Debug mode

- Then press F11 the program executes step by step.



Continue
Auto
[24252] Main Thread
Stack Frame:
main
Step Into (F11)
(Global Scope)
int main (void)