UCOSIII Porting

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The document introduces how to port **μC/OS-III** on M451 and demonstration a sample code bases on **μC/OS-III** RTOS.
BSP Version	M451 Series BSP CMSIS v3.01.002
Hardware	NuTiny-EVB-M451-LQFP100 V1.3

1 Function Description

1.1 Introduction

This example code demonstrates how to port **μC/OS-III** on NuMicro® M451 Series MCU. Micro-Controller Operating Systems (MicroC/OS, stylized as μC/OS) is a real-time operating system (RTOS) designed by embedded software developer, Jean J. Labrosse in 1991. Based on the source code written for μC/OS, and introduced as a commercial product in 1998, **μC/OS-III** is a portable, ROM-able, scalable, preemptive, real-time, deterministic, multitasking kernel for microprocessors, and digital signal processors (DSPs). Most of **μC/OS-III** is written in highly portable ANSI C, with target microprocessor-specific code written in assembly language. Use of the latter is minimized to ease porting to other processors. (Refer to https://en.wikipedia.org/wiki/Micro-Controller_Operating_Systems).

This example code achieves porting the **μC/OS-III** on NuMicro® M451 Series MCU.

The main system configuration of the M451 MCU is as follows:

- System clock frequency is 48 MHz
- Clock source is from PLL, and PLL source is from HXT
- SysTick interrupt period is 1 ms

1.2 Porting μC/OS-III

An easiest way to port **μC/OS-III** is to download μC/OS-III BSP with same CPU core from the Micrium Web site.

1.2.1 Start up

To search a similar BSP with same CPU Core from following website.

https://www.micrium.com/downloadcenter/

1.2.2 Configuration μC/OS-III

Three files are used to configure **μC/OS-III** as highlighted in the figure below: os_cfg.h, os_cfg_app.h and os_type.h. If download **μC/OS-III** with same CPU Core, these files aren't necessary to change.

os_cfg.h

It is used to determine which features are needed from **µC/OS-III** for an application (i.e., product). Specifically, this file allows a user to determine whether to include semaphores, mutexes, event flags, run-time argument checking, etc.

os_cfg_app.h

It is used to configured at the application level through #define constants in os_cfg_app.h. uC/OS-III allows a user to specify stack sizes for all **µC/OS-III** internal tasks: the idle task, statistic task, tick task, timer task, and the ISR handler task.

os_type.h

It establishes µC/OS-III-specific data types used when building an application. It specifies the size of variables used to represent task priorities, the size of a semaphore count, and more. This file contains recommended data types for **µC/OS-III**, however these can be altered to make better use of the CPU's natural word size. For example, on some 32-bit CPUs, it is better to declare boolean variables as 32-bit values for performance considerations, even though an 8-bit quantity is more space efficient

1.2.3 Porting Files Under uCOSIII\BSP

The files under the folder refer to code associated with the actual evaluation board or the target board used. For example, the BSP defines functions to turn LEDs on or off, reads push-button switches, initializes peripheral clocks, etc. Basic functions list as following

- bsp.c and bsp.h

These files normally contain functions and their definitions such as:

BSP_Init()

This function is called by application code to initialize the BSP functionality. BSP_Init() could initialize I/O ports, setup timers, serial ports, SPI ports and so on.

- bsp_int.c

Hook interrupts between BSP and **µC/OS-III**.

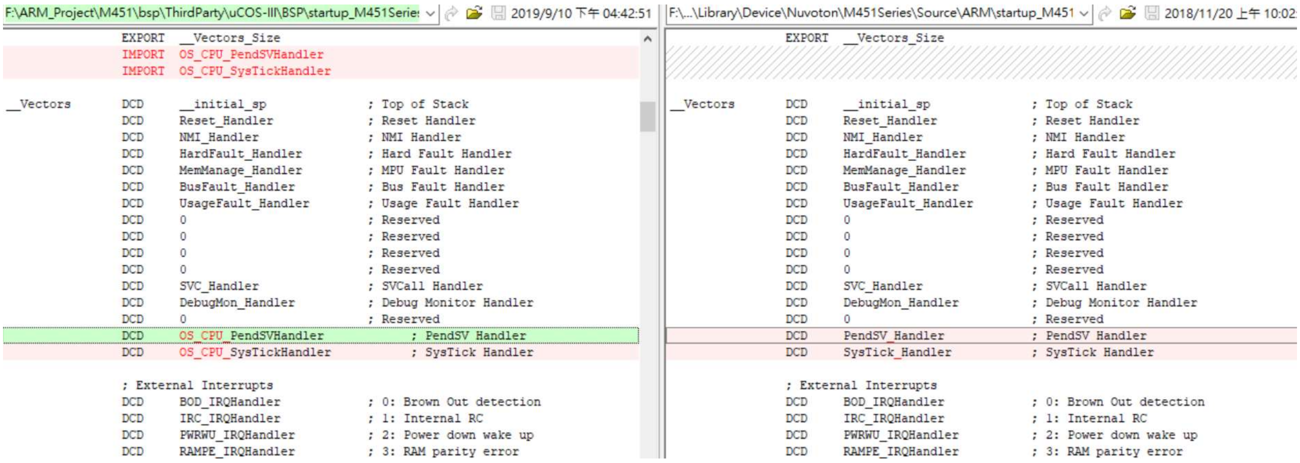
- bsp_periph.c

To get and set the working frequency of peripherals

- startup_M451Series.s

To modify the PendSV and SysTick Handler's name for the requirement of **µC/OS-III.** And modify Reset_Handler to enable Soft-VFP.

An illustration shows between following figures. The left side is for **µC/OS-III.** The right side is for standard (Non-OS) BSP.

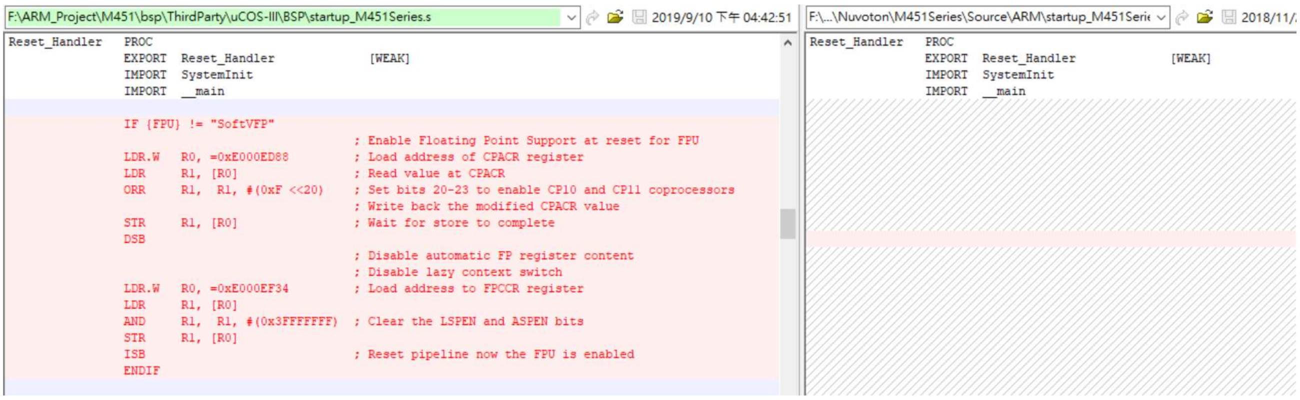


F:\ARM_Project\M451\bsp\ThirdParty\uCOS-III\BSP\startup_M451Serie 2019/9/10 下午 04:42:51

```
                EXPORT  __Vectors_Size
                IMPORT  OS_CPU_PendSVHandler
                IMPORT  OS_CPU_SysTickHandler

__Vectors       DCD     __initial_sp              ; Top of Stack
                DCD     Reset_Handler             ; Reset Handler
                DCD     NMI_Handler               ; NMI Handler
                DCD     HardFault_Handler         ; Hard Fault Handler
                DCD     MemManage_Handler         ; MPU Fault Handler
                DCD     BusFault_Handler          ; Bus Fault Handler
                DCD     UsageFault_Handler        ; Usage Fault Handler
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     SVC_Handler               ; SVCall Handler
                DCD     DebugMon_Handler          ; Debug Monitor Handler
                DCD     0                         ; Reserved
                DCD     OS_CPU_PendSVHandler            ; PendSV Handler
                DCD     OS_CPU_SysTickHandler           ; SysTick Handler

                ; External Interrupts
                DCD     BOD_IRQHandler            ; 0: Brown Out detection
                DCD     IRC_IRQHandler            ; 1: Internal RC
                DCD     PWRWU_IRQHandler          ; 2: Power down wake up
                DCD     RAMPE_IRQHandler          ; 3: RAM parity error
```

F:\...\Library\Device\Nuvoton\M451Series\Source\ARM\startup_M451 2018/11/20 上午 10:02

```
                EXPORT  __Vectors_Size


__Vectors       DCD     __initial_sp              ; Top of Stack
                DCD     Reset_Handler             ; Reset Handler
                DCD     NMI_Handler               ; NMI Handler
                DCD     HardFault_Handler         ; Hard Fault Handler
                DCD     MemManage_Handler         ; MPU Fault Handler
                DCD     BusFault_Handler          ; Bus Fault Handler
                DCD     UsageFault_Handler        ; Usage Fault Handler
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     0                         ; Reserved
                DCD     SVC_Handler               ; SVCall Handler
                DCD     DebugMon_Handler          ; Debug Monitor Handler
                DCD     0                         ; Reserved
                DCD     PendSV_Handler            ; PendSV Handler
                DCD     SysTick_Handler           ; SysTick Handler

                ; External Interrupts
                DCD     BOD_IRQHandler            ; 0: Brown Out detection
                DCD     IRC_IRQHandler            ; 1: Internal RC
                DCD     PWRWU_IRQHandler          ; 2: Power down wake up
                DCD     RAMPE_IRQHandler          ; 3: RAM parity error
```



F:\ARM_Project\M451\bsp\ThirdParty\uCOS-III\BSP\startup_M451Series.s 2019/9/10 下午 04:42:51

```
Reset_Handler   PROC
                EXPORT  Reset_Handler             [WEAK]
                IMPORT  SystemInit
                IMPORT  __main

                IF {FPU} != "SoftVFP"
                                                  ; Enable Floating Point Support at reset for FPU
                LDR.W   R0, =0xE000ED88           ; Load address of CPACR register
                LDR     R1, [R0]                  ; Read value at CPACR
                ORR     R1,  R1, #(0xF <<20)      ; Set bits 20-23 to enable CP10 and CP11 coprocessors
                                                  ; Write back the modified CPACR value
                STR     R1, [R0]                  ; Wait for store to complete
                DSB

                                                  ; Disable automatic FP register content
                                                  ; Disable lazy context switch
                LDR.W   R0, =0xE000EF34           ; Load address to FPCCR register
                LDR     R1, [R0]
                AND     R1,  R1, #(0x3FFFFFFF)    ; Clear the LSPEN and ASPEN bits
                STR     R1, [R0]
                ISB                               ; Reset pipeline now the FPU is enabled
                ENDIF
```

F:\...\Nuvoton\M451Series\Source\ARM\startup_M451Serie 2018/11/

```
Reset_Handler   PROC
                EXPORT  Reset_Handler             [WEAK]
                IMPORT  SystemInit
                IMPORT  __main
```

1.2.4 Porting Files Under uCOSIII\uCOSIII

All folders under uCOSIII\uCOSIII path, uC-CPU, uC-Lib and uCOS-III, will be kept and unchanged.

1.2.5 Porting File Under uCOSIII\USER

The folder is used to demonstrate the example code bases on the **µC/OS-III.** Project code is located under Keil folder. Code entry is located in file - app.c. Programmer can start to modify it from the file.

1.2.6 Demo Code Project Architecture

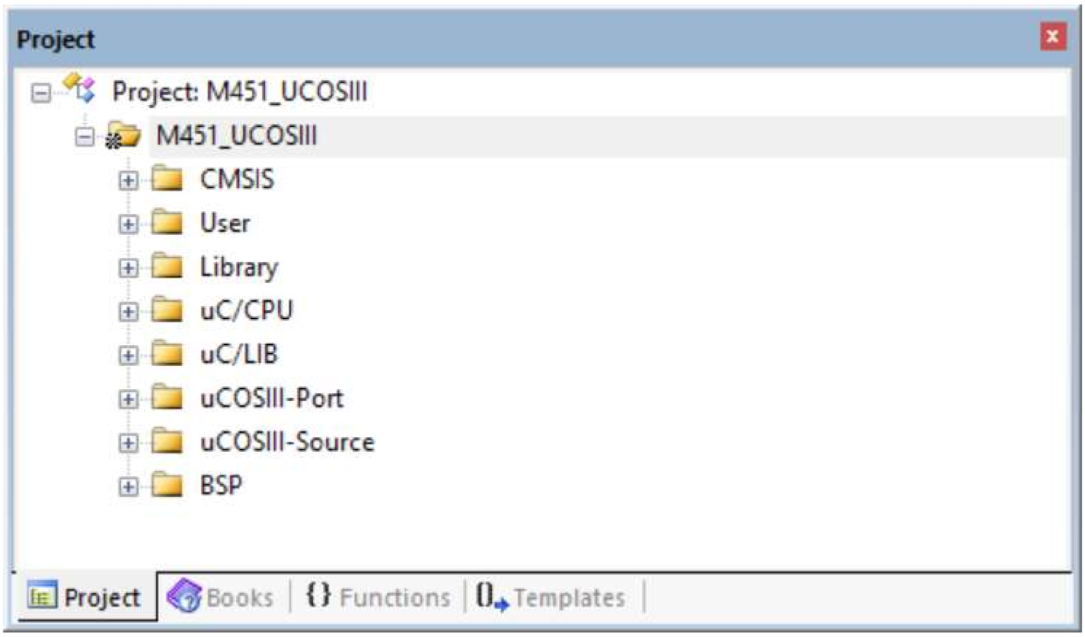
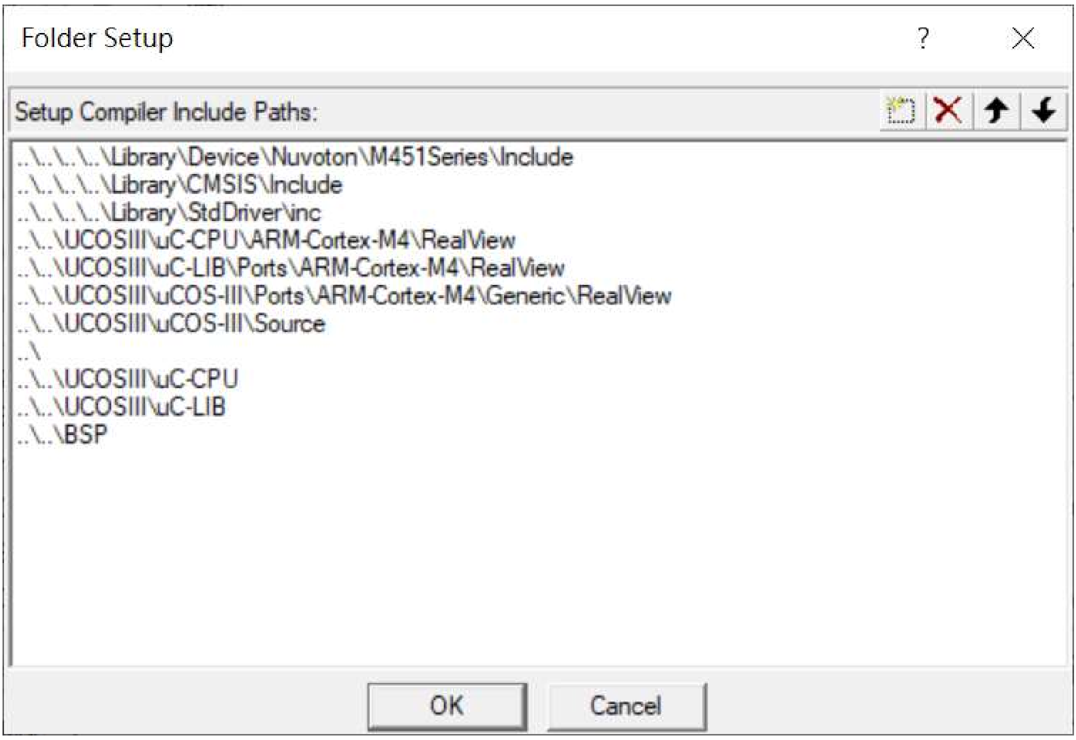


Figure 1-1 Demo code project architecture

1.2.7 Demo Code Project Include Path

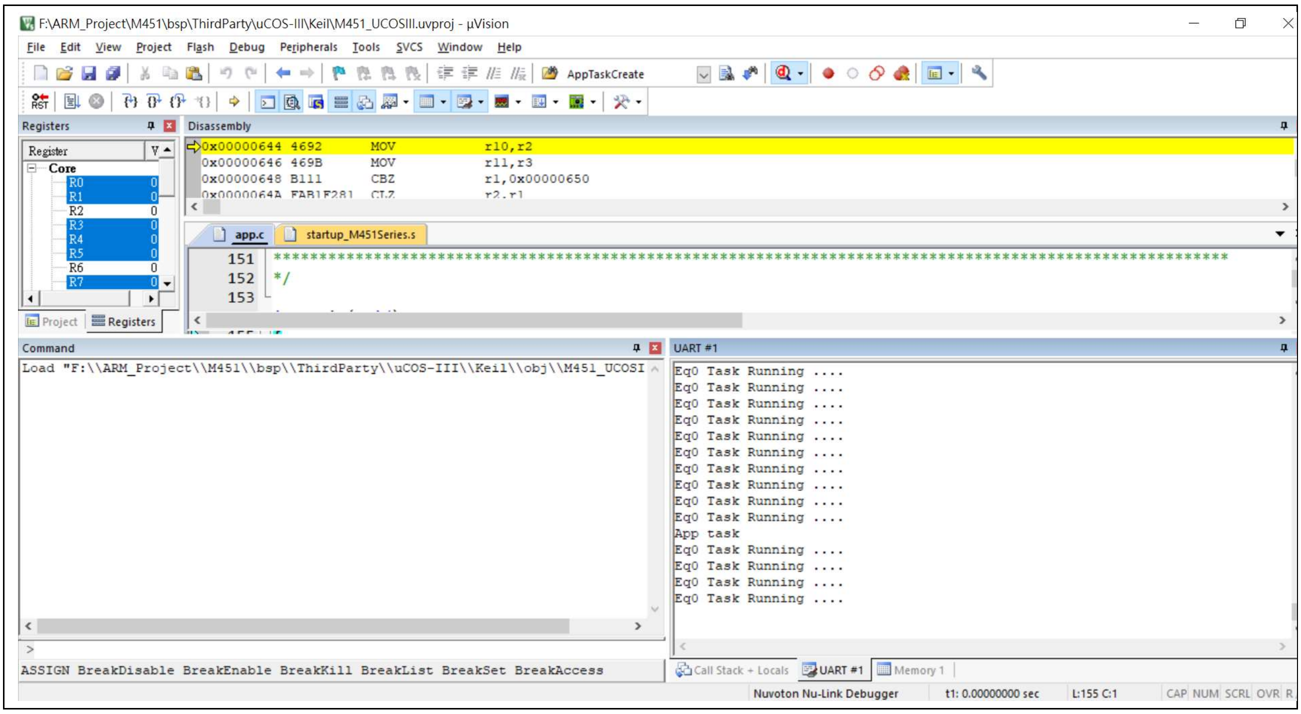


```
..\..\..\..\Library\Device\Nuvoton\M451Series\Include
..\..\..\..\Library\CMSIS\Include
..\..\..\..\Library\StdDriver\inc
..\..\UCOSIII\uC-CPU\ARM-Cortex-M4\RealView
..\..\UCOSIII\uC-LIB\Ports\ARM-Cortex-M4\RealView
..\..\UCOSIII\uCOS-III\Ports\ARM-Cortex-M4\Generic\RealView
..\..\UCOSIII\uCOS-III\Source
..\
..\..\UCOSIII\uC-CPU
..\..\UCOSIII\uC-LIB
..\..\BSP
```

Figure 1-2 Demo code project include path

1.3 Demo Result

The application execution result shows on the semi-hosting#UART1 as shown in Figure 1-3 Demo result



```
0x00000644 4692      MOV      r10,r2
0x00000646 469B      MOV      r11,r3
0x00000648 B111      CBZ      r1,0x00000650
```

```
151 ***************************************************************************************************
152 */
153
```

```
Load "F:\\ARM_Project\\M451\\bsp\\ThirdParty\\uCOS-III\\Keil\\obj\\M451_UCOSI
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

```
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
App task
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
Eq0 Task Running ....
```

Figure 1-3 Demo result

2 Demo Code Description

Demo code project file located under folder \\M451\bsp\ThirdParty\uCOS-III\Keil\

Initialize system, UART, Memory Management Module and Mathematical Module in main.c.

```
    SYS_Init();

    UART0_Init();

    /* Disable all interrupts                */
    BSP_IntDisAll();

    /* Initialize the uC/CPU Services        */
    CPU_Init();

    /* Initialize Memory Management Module   */
    Mem_Init();

    /* Initialize Mathematical Module        */
    Math_Init();
```

Following initialize OS and create start tasks.

```
    /* Init uC/OS-III */
    OSInit(&err);

    /* Create the start task */
    OSTaskCreate((OS_TCB       *)&AppTaskStartTCB,
                 (CPU_CHAR     *)"App Task Start",
                 (OS_TASK_PTR   )AppTaskStart,
                 (void         *)0u,
                 (OS_PRIO       )APP_CFG_TASK_START_PRIO,
                 (CPU_STK      *)&AppTaskStartStk[0u],
                 (CPU_STK_SIZE  )AppTaskStartStk[APP_CFG_TASK_START_STK_SIZE / 10u],
                 (CPU_STK_SIZE  )APP_CFG_TASK_START_STK_SIZE,
                 (OS_MSG_QTY    )0u,
                 (OS_TICK       )0u,
                 (void         *)0u,
                 (OS_OPT        )(OS_OPT_TASK_STK_CHK | OS_OPT_TASK_STK_CLR),
                  (OS_ERR       *)&err);

    /* Start multitasking (i.e. give control to uC/OS-III). */
    OSStart(&err);
```

Create application task in start task.

```
static  void  AppTaskStart (void *p_arg)
{
    OS_ERR  err;
    (void)p_arg;
    /* Initialize BSP functions  */
    BSP_Init();

    /* Initialize Tick Services  */
    BSP_Tick_Init();

    /* Compute CPU capacity with no task running */
#if OS_CFG_STAT_TASK_EN > 0u
    OSStatTaskCPUUsageInit(&err);
#endif

#ifdef CPU_CFG_INT_DIS_MEAS_EN
    CPU_IntDisMeasMaxCurReset();
#endif

    APP_TRACE_DBG(("Creating Application Kernel Objects\n\r"));
    /* Create Applicaiton kernel objects */
    AppObjCreate();

    APP_TRACE_DBG(("Creating Application Tasks\n\r"));
    /* Create Application tasks */
    AppTaskCreate();

    /* Task body, always written as an infinite loop. */
    while (DEF_TRUE) {
       OSTimeDlyHMSM(0u, 0u, 1u, 100u,
                  OS_OPT_TIME_HMSM_STRICT, &err);
    printf("App task \n");
    }
}
```

3 Software and Hardware Environment

- **Software Environment**
  - BSP version
    - M451 Series BSP CMSIS v3.01.002
  - IDE version
    - Keil uVersion 5.24
- **Hardware Environment**
  - Circuit component
    - NuTiny-EVB-M451-LQFP100 V1.3
  - Diagram
    - Connect Nu-Link ICE to JP2 for downloading code and debugging

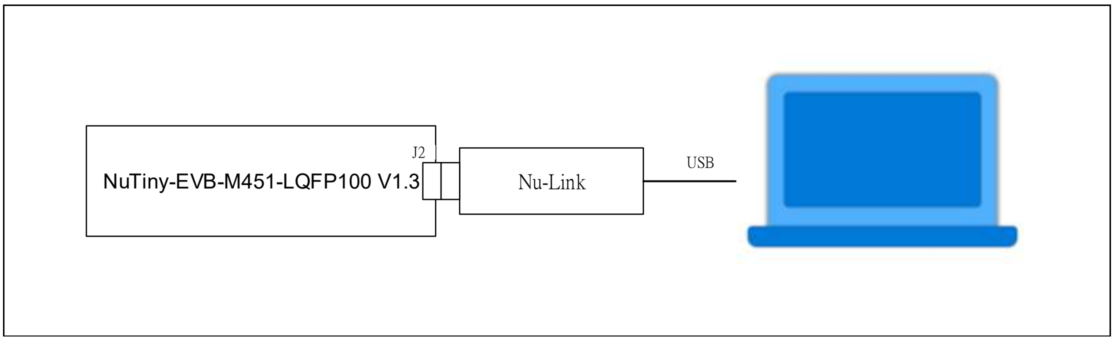


Figure 3-1 Hardware environment

4 Directory Information

🗁 EC_M451_UCOSIII_Porting_V1.00			
	🗁 Library		Sample code header and source files
		🗁 CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
		🗁 Device	CMSIS compliant device header file
		🗁 StdDriver	All peripheral driver header and source files
	🗁 ThirdParty		
		🗁 UCOSIII	**µC/OS-III** BSP and demo code

5 How to Execute Example Code

1. Opening the path ThirdParty\UCOSIII\USER\Keil folder by Directory Information (section 4) and double click M451_UCOSIII.uvproj.
2. Enter Keil compile mode
    a. Build
    b. Download
    c. Start/Stop debug session
    d. Open Semi-Host UART#1 through click menu View\Serial Windows\UART#1
3. Enter debug mode
    a. Run

6 Revision History

Date	Revision	Description
Sep 30, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, “Insecure Usage”.

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer’s risk, and in the event that third parties lay claims to Nuvoton as a result of customer’s Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.