

Use M480 LDROM As APROM

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	The NuMicro® M480 series provides embedded Flash memory (includes APROM and LDROM). APROM is main memory for user applications. LDROM is for a loader to implement In-System-Programming (ISP) function. LDROM is a 4KB embedded Flash memory, the Flash address range is from 0x0010_0000 to 0x0010_0FFF. M480 will be treated LDROM as an extension space for APROM when no ISP update executed.
BSP Version	M480 Series BSP CMSIS V3.03.000
Hardware	NuMaker-ETM-M487 v1.1

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

1.1 Introduction

The NuMicro® M480 series provides two kinds of embedded Flash memory, including APROM and LDROM, M480 will be treated LDROM as an extension space for APROM when no ISP update executed. The following chapters will demonstrate this function with a sample.

1.2 Principle

In order to illustrate this sample, the IAP(In-application-programming) function must be enabled by re-booting chip and setting the chip boot selection bits in CBS (CONFIG0[7:6]) as 2(APROM with IAP mode). IAP is a function for user to switch the program executing between APROM and LDROM without reset. Figure 1 shows the executable range of program with IAP function enabled of NuMicro® M480 series. There are programs stored in APROM and LDROM both, and the executing program will jump from APROM to LDROM via LDROM's predetermined function address.

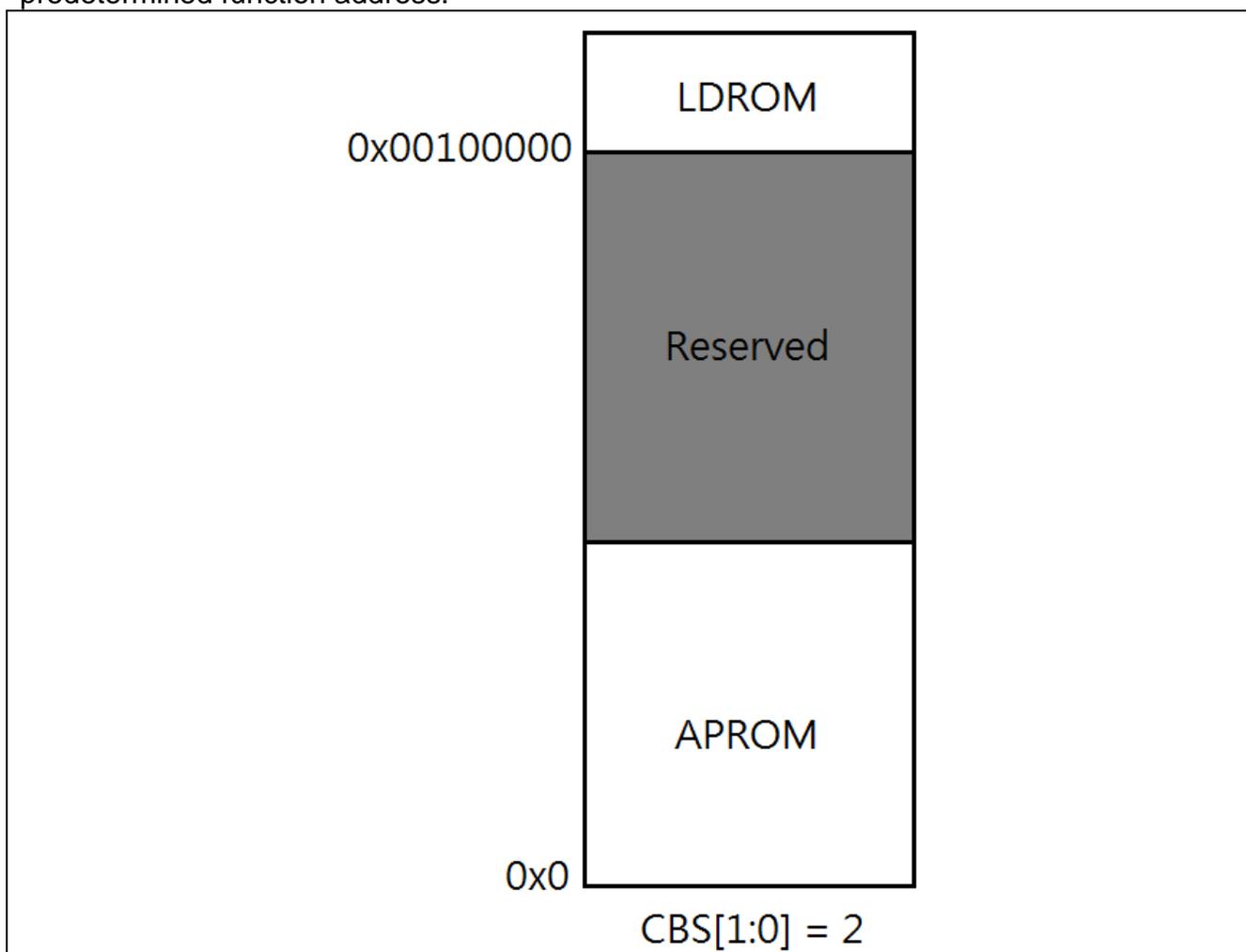


Figure 1 IAP function enable

1.3 Demo Result

After the program compile completed, user can download it to the NuMaker-ETM-M487 development board for verification. User can see PH.0 (Red LED) constantly changes status, the waveform as show in Figure 2.

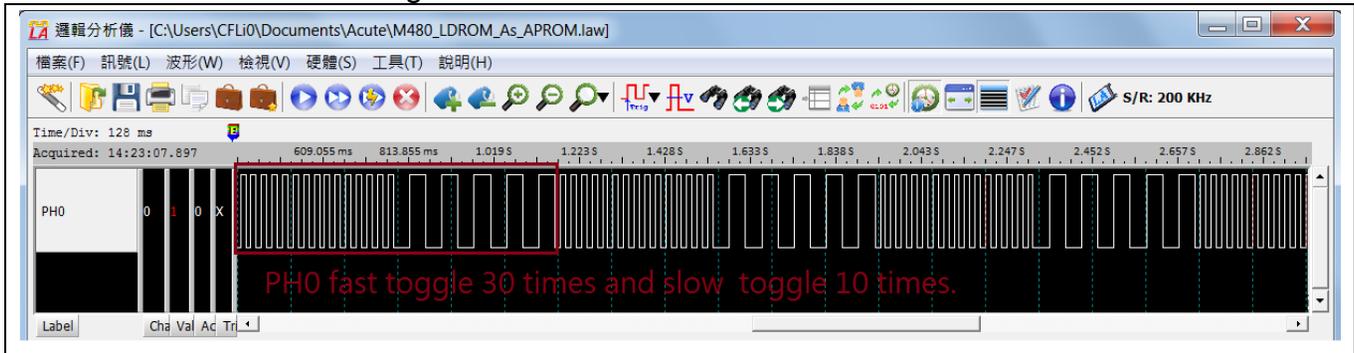


Figure 2 PH.0 Fast/Slow toggle

2 Code Description

The example is using Keil uVersion 5.26. There are two KEIL projects in this example. One is APROM program, the other is LDROM program. These projects need to set parameters and download to APROM and LDROM respectively. The following steps will introduce how to demonstrate this example.

Step 1. First, open LDROM.uvproj to select Target Options. Choice the Linker tab page, and input "0x00100000" in the R/O Base field, as shown in Figure 3.

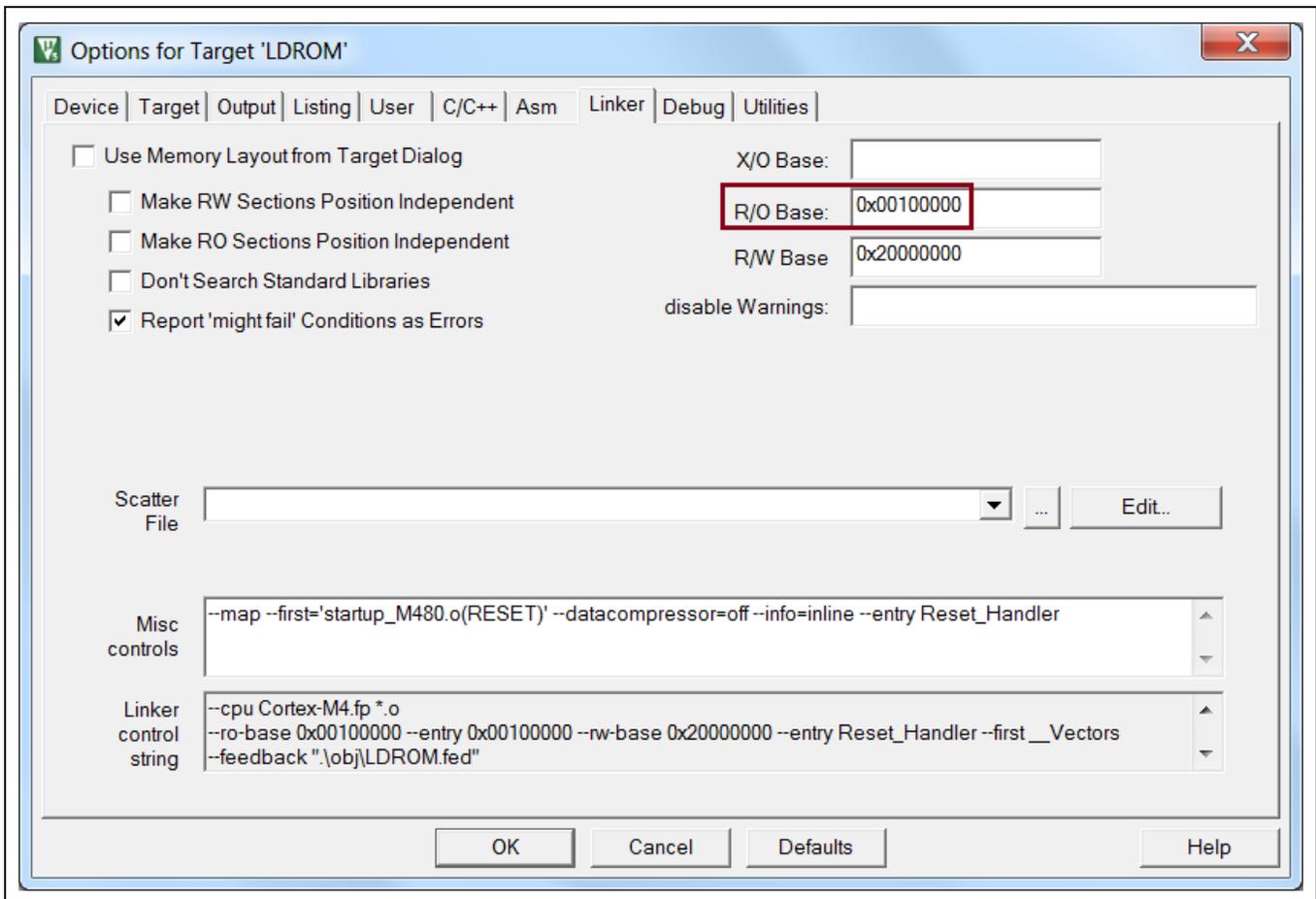


Figure 3 Add data to R/O Base item

Step 2: Go to the Utilities tab page and double click Settings button. It will pop the window and select "LDROM" in the Flash Select item, as shown in Figure 4.

Step 3: Finish above settings, user compiles LDROM program. When compile complete and without build error, user go to the example folder location to find the LDROM.txt file. As shown in Figure 5, user search function name (Fastflash and Slowflash) to know these functions address with the LDROM.

Step 4: Running this example, chip must enable APROM with IAP mode (CBS[1:0]=2).

Step 5: Open APROM.uvproj to start Keil to select Target Options, and double click Settings button in the Utilities tab page. It will pop the window and select “APROM with IAP” in the Flash Select item, as shown in Figure 6.

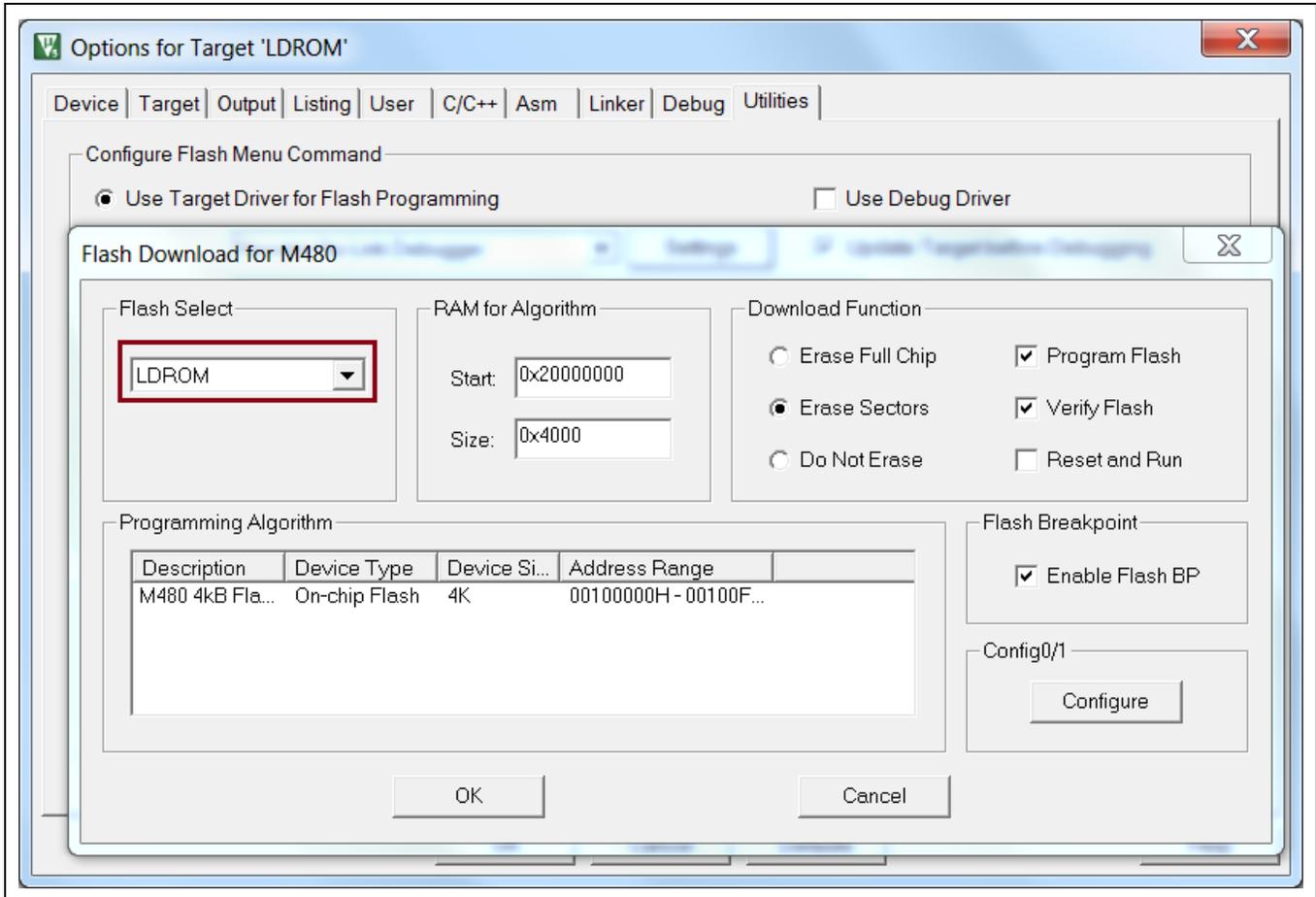


Figure 4 Flash Select LDROM

```

Fastflash
0x00100496:  b570      p.      PUSH    {r4-r6,lr}
0x00100498:  4604      .F      MOV     r4,r0
0x0010049a:  4d14      .M      LDR     r5,[pc,#80] ; [0x1004ec] = 0x400049c0
0x0010049c:  4e14      .N      LDR     r6,[pc,#80] ; [0x1004f0] = 0x186a0
0x0010049e:  e007      ..      B       0x1004b0 ; Fastflash + 26
0x001004a0:  6828      (h     LDR     r0,[r5,#0]
0x001004a2:  f0800001  ....   EOR     r0,r0,#1
0x001004a6:  6028      (`    STR     r0,[r5,#0]
0x001004a8:  4630      0F     MOV     r0,r6
0x001004aa:  f7ffffe5  ....   BL      CLK_SysTickDelay ; 0x100478
0x001004ae:  1e64      d.     SUBS   r4,r4,#1
0x001004b0:  2c00      .,     CMP     r4,#0
0x001004b2:  d1f5      ..     BNE    0x1004a0 ; Fastflash + 10
0x001004b4:  bd70      p.     POP    {r4-r6,pc}

Slowflash
0x001004b6:  b570      p.     PUSH    {r4-r6,lr}
0x001004b8:  4604      .F     MOV     r4,r0
0x001004ba:  4d0c      .M     LDR     r5,[pc,#48] ; [0x1004ec] = 0x400049c0
0x001004bc:  4e0d      .N     LDR     r6,[pc,#52] ; [0x1004f4] = 0xf4240
0x001004be:  e007      ..     B       0x1004d0 ; Slowflash + 26
0x001004c0:  6828      (h     LDR     r0,[r5,#0]
0x001004c2:  f0800001  ....   EOR     r0,r0,#1
0x001004c6:  6028      (`    STR     r0,[r5,#0]
0x001004c8:  4630      0F     MOV     r0,r6
0x001004ca:  f7ffffd5  ....   BL      CLK_SysTickDelay ; 0x100478
0x001004ce:  1e64      d.     SUBS   r4,r4,#1
0x001004d0:  2c00      .,     CMP     r4,#0
0x001004d2:  d1f5      ..     BNE    0x1004c0 ; Slowflash + 10
0x001004d4:  bd70      p.     POP    {r4-r6,pc}
    
```

Figure 5 LDROM.txt

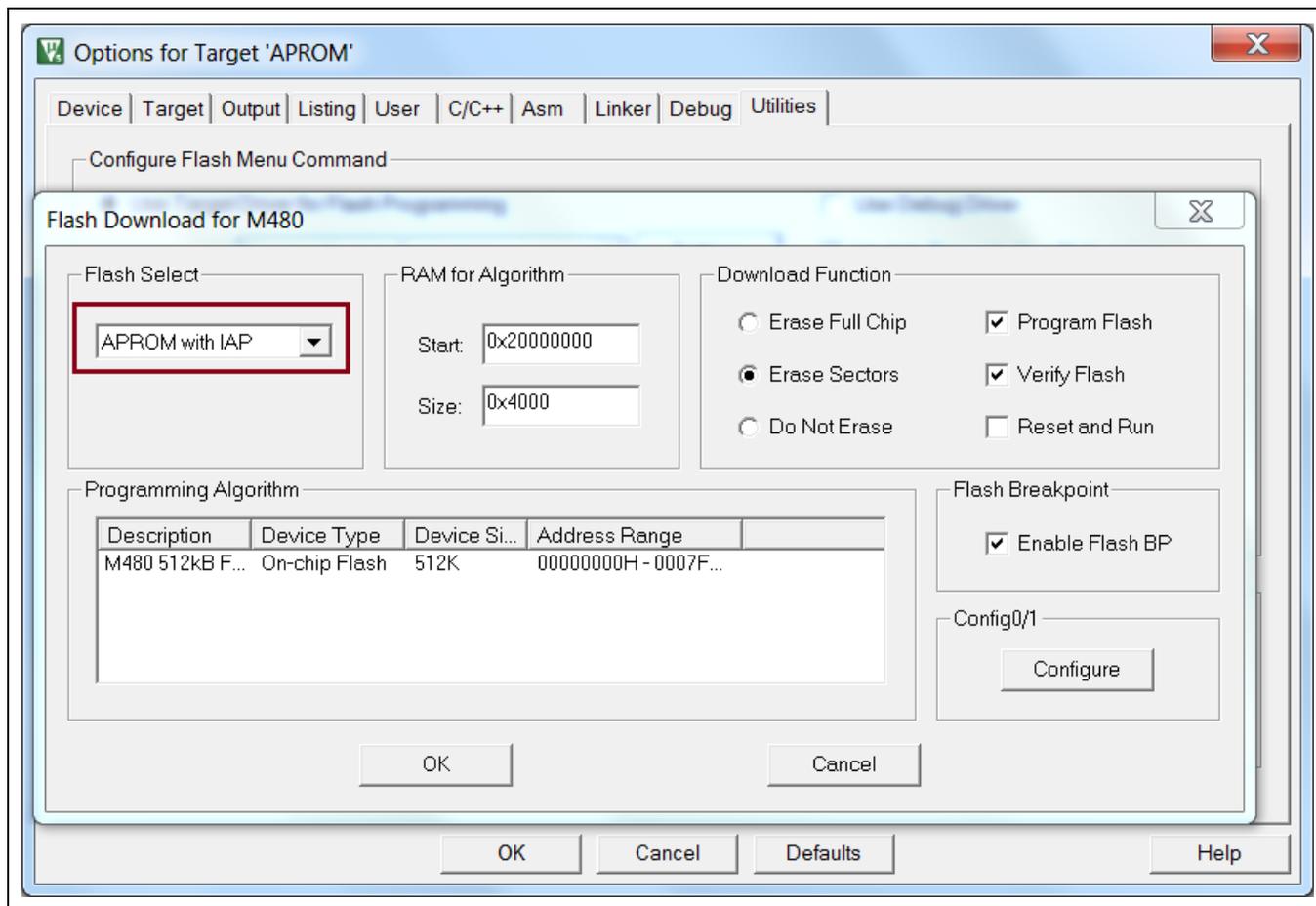


Figure 6 Flash Select APROM with IAP

In this example, func1() and func2() are LDROM programs. Two programs constantly changes status on PH.0(Red LED).

```
int main()
{
    // Changes PH.0(Red LED) status
    func1=(pointfunc)Fastflash;
    // Changes PH.0(Red LED) status
    func2=(pointfunc)Slowflash;

    while(1);
}

// Fast toggle PH.0(Red LED)
void Fastflash(uint32_t i)
{
    while(i > 0)
    {
```

```

        PH0 ^=1;
        CLK_SysTickDelay(100000);
        i--;
    }
}

// Slow toggle PH.0(Red LED)
void Slowflash(uint32_t i)
{
    while(i > 0)
    {
        PH0 ^=1;
        CLK_SysTickDelay(1000000);
        i--;
    }
}

```

The following code lists APROM program. APROM program will switch to LDROM running the functions. User need to know the two functions address within LDROM and apply these address to APROM main() to execute functions.

```

int main()
{
    /* Init System, IP clock and multi-function I/O */
    SYS_Init();

    /* Configure PH.0 to Output mode */
    GPIO_SetMode(PH, BIT0, GPIO_MODE_OUTPUT);

    /* Switch to LDROM to execute LDROM program */
    while(1)
    {
        /*func Fastflash() in LDROM
        func = (pointfunc )(0x100000+0x497);
        func(30);
        /*func Slowflash() in LDROM
        func = (pointfunc )(0x100000+0x4b7);
        func(10);
    }
}

```

3 Software and Hardware Environment

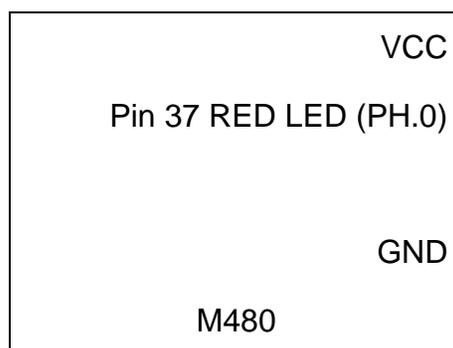
- **Software Environment**

- BSP version
 - ◆ M480 Series BSP CMSIS V3.03.000
- IDE version
 - ◆ Keil uVersion 5.26

- **Hardware Environment**

- Circuit components
 - ◆ NuMaker-ETM-M487
- Diagram

User can see PH.0(LED) blink or output PH.0 (pin 37) waveform.



4 Directory Information

📁 EC_M480_Use_LDROM_As_APROM_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
📁 SampleCode	
📁 ExampleCode	Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click APROM.uvproj.
2. Enter Keil compile mode
 - a. Build
 - b. Download
 - c. Start/Stop debug session
3. Enter debug mode
 - a. Run

6 Revision History

Date	Revision	Description
Jul. 26, 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.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*