M480 ISP固件更新与修复

NuMicro® 32位系列微控制器范例代码介绍

文件信息

应用简述	本范例程序为 LDROM 之应用，将 APROM 分割为 2 块，在上电后对执行块与备份块进行比较与更新，以实现固件更新与修复功能。
BSP 版本	M480 Series BSP CMSIS V3.05.003
开发平台	NuMaker-PFM-M487 V3.0

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 and microprocessor 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. 概述

本范例为 LDROM 之应用，支援了以 UART 更新 APROM 并透过 Bank 拆分的方式对其进行固件校验与修复。由于 M480 系列 IC 没有切换 Bank 映射的功能，因此本范例将 APROM 的 2 个 bank 区分为 Bank0 的执行块与 Bank1 的备份块如图 1-1 所示，当使用 ICE 或 UART 进行固件更新时，将程序更新至 APROM_SIZE/2 的地址，待重新上电后，本范例将对 Bank0 与 Bank1 逐页进行比较，若出现差异，则更新相对应的页，以达到修复应用程序的目的。在一般情况下重新上电，也会对 Bank0 与 Bank1 进行比较，修正完错误的页后才执行应用。

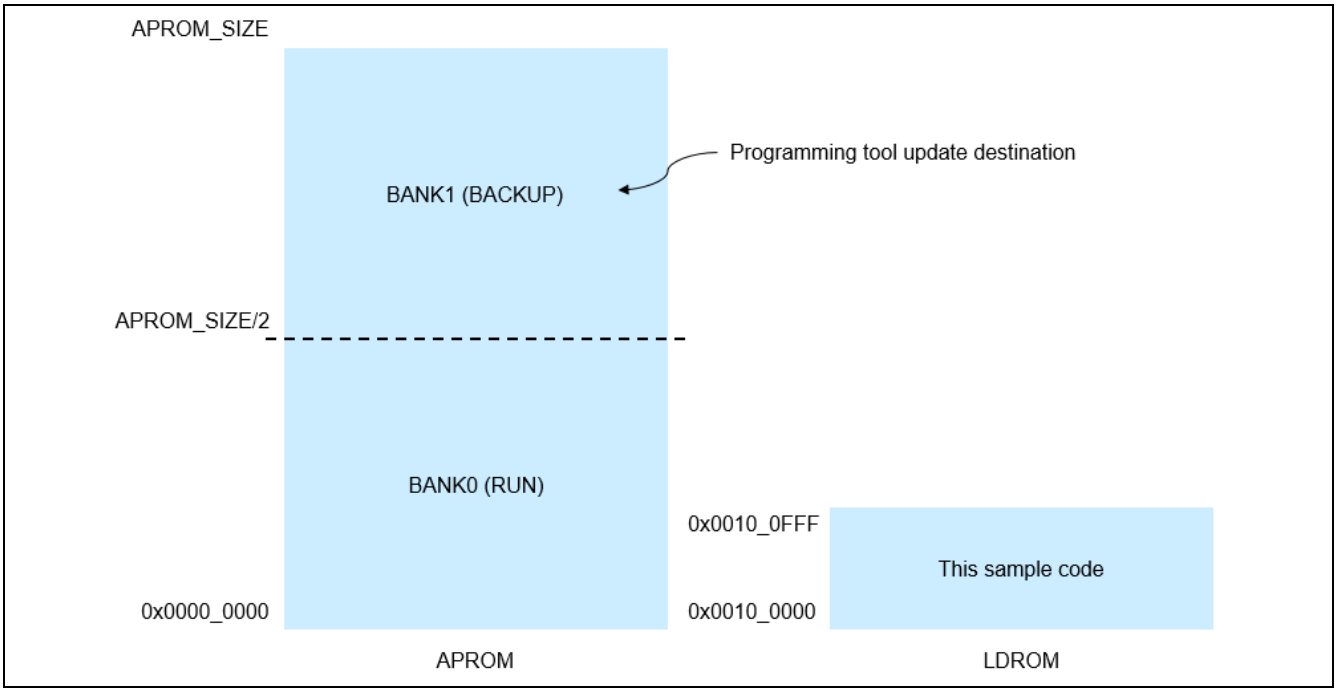

图 1-1 快闪记忆体分配

1.1 软件架构

软件流程如图 1-2 所示， 启动模式选择由 LDROM 启动，此时位于 LDROM 的程序会在等待 UART 命令的同时对 Bank0 与 Bank1 做逐页比对并记录结果，比较的长度为 APROM_SIZE 的一半，当比对完成且接收 UART 命令超时，则会根据比对结果区分为: 1. 比对内容一致，跳转至 APROM 执行应用，2. 内容不一致，针对 Bank0 相对应的页进行擦除与写入，待执行完毕，再跳转至 APROM 执行应用。

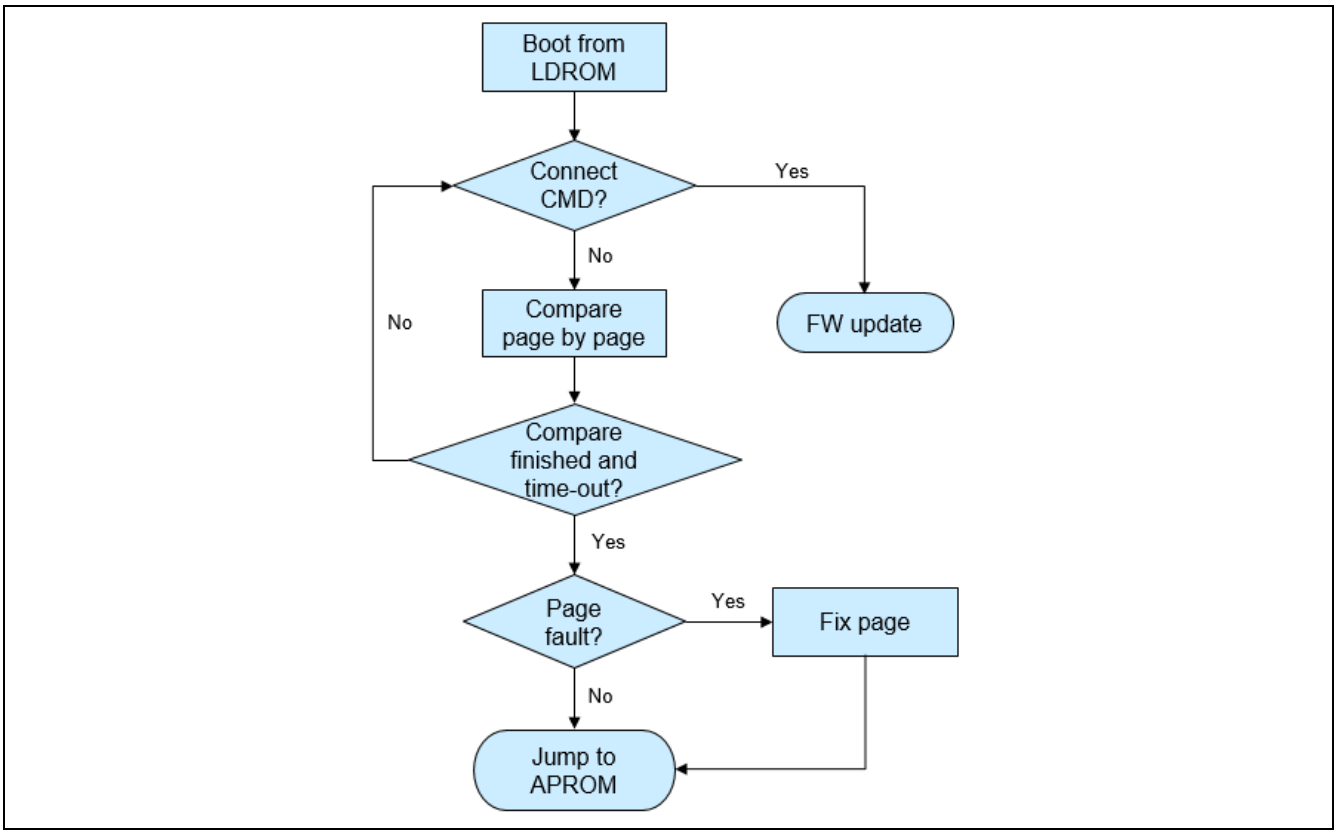

图1-2 软件流程图

1.2 功能介绍

1.2.1 启动模式设定

- 启动模式：

  M480 系列 MCU 支援了多种启动模式，包含 LDROM with IAP，LDROM without IAP，APROM with IAP，APROM without IAP，其中设定启动模式的地址为 0x0030_0000 的 CONFIG0[7:6] CBS，本范例建立于 LDROM，使用 CBS 为 01 的 LDROM 启动，使固件更新与 APROM 的应用彼此独立。
  当使用 LDROM 开机时，会将 LDROM 映射至起始地址为 0 处，此时若想存取 APROM 的地址则需使用 FMC 模块。

1.2.2 FMC 操作

- 擦除：

  M480 系列 IC 擦除的最小单位为页，操作时将 FMC_ISPCMD 填入擦除命令 0x22，FMC_ISPADDR 填入每页的实体起始地址，再将 FMC_ISPTRG 写 1 完成一次擦除流程。

- 读：

  读取每次以 32 位元为单位，操作时将 FMC_ISPCMD 写入读命令 0x0，FMC_ISPADDR 填入欲读取的实体地址，再将 FMC_ISPTRG 进行读取，读取的资料可从 FMC_ISPDAT 取得。

- 写：

  写的最小单位为 32 位元，操作时将 FMC_ISPCMD 填入写命令 0x21，FMC_ISPDAT 填入欲写入的资料，FMC_ISPADDR 填入目标实体地址，再将 FMC_ISPTRG 写 1 完成写入流程。

1.2.3 执行说明

1.2.3.1 前置作业

第一次烧录需连接 ICE 并搭配 Nuvoton NuMicro ICP Programming Tool 将此范例程序烧录至 LDROM，并将 CONFIG0[7:6] CBS 写为 0x1，使启动模式改为由 LDROM 启动，应用程序的烧录以及后续的更新亦可使用 Nuvoton NuMicro ISP Programming tool 透过 UART 进行固件更新，烧录前须注意应用程序的大小需小于 APROM 大小的一半。

1.2.3.2 执行结果

实验以大小为 512KB 之 APROM 的固件更新为例：

固件更新前，程序#1 在地址 0x200 的资料如图 1-3。

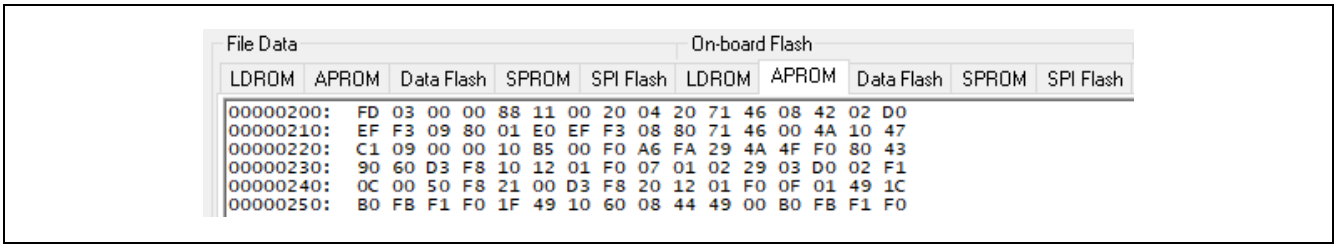

图 1-3 更新前执行区块

将程序#2 更新至起始地址为 APROM 大小的一半，地址 0x40200 的资料如图 1-4 。

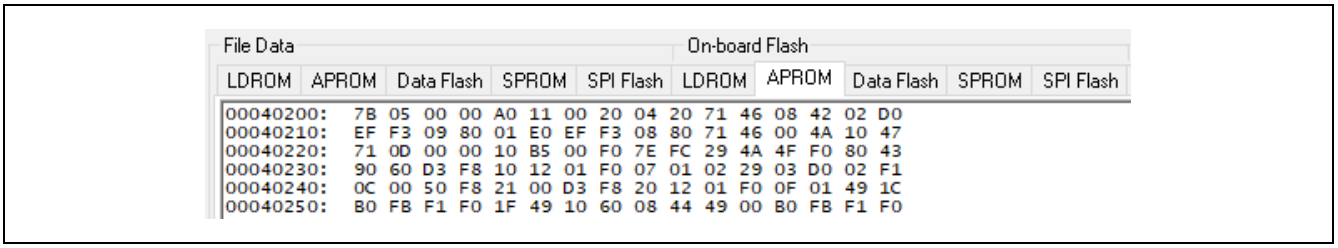

图 1-4 更新后备份区块

重新上电后，位于 LDROM 的范例程序对 Bank0 与 Bank1 进行比较与更新，可得在地址 0x200 与 0x40200 资料相同如图 1-5。

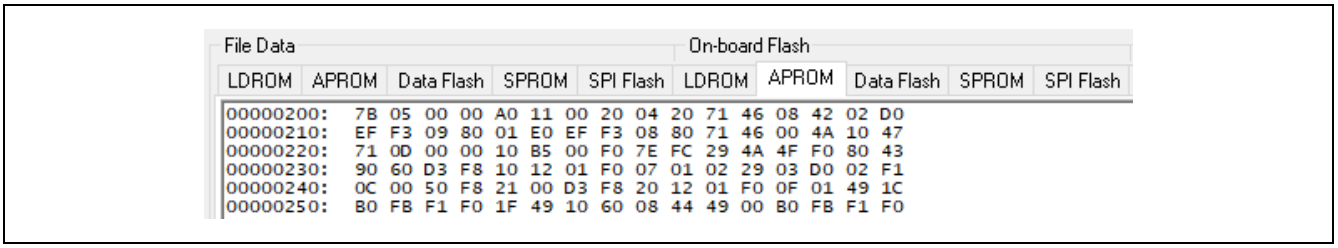
File Data
On-board Flash
LDROM APROM Data Flash SPROM SPI Flash LDROM APROM Data Flash SPROM SPI Flash
00000200: 7B 05 00 00 A0 11 00 20 04 20 71 46 08 42 02 D0
00000210: EF F3 09 80 01 E0 EF F3 08 80 71 46 00 4A 10 47
00000220: 71 0D 00 00 10 B5 00 F0 7E FC 29 4A 4F F0 80 43
00000230: 90 60 D3 F8 10 12 01 F0 07 01 02 29 03 D0 02 F1
00000240: 0C 00 50 F8 21 00 D3 F8 20 12 01 F0 0F 01 49 1C
00000250: B0 FB F1 F0 1F 49 10 60 08 44 49 00 B0 FB F1 F0

图 1-5 更新后执行区块

2. 程序介绍

以下程序实现了章节 1.1 的架构，包含 1.2.2 所述的 FMC 使用流程。

```
/* Word compare between source address and destination address */
int flashcmp(uint32_t desAddr, uint32_t srcAddr)
{
    uint32_t srcData, desData;
    FMC->ISPCMD = FMC_ISPCMD_READ;
    FMC->ISPADDR = srcAddr;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
    srcData = FMC->ISPDAT;

    FMC->ISPCMD = FMC_ISPCMD_READ;
    FMC->ISPADDR = desAddr;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
    desData = FMC->ISPDAT;

    return srcData != desData;
}

/* Word copy from source address to destination address */
void flashcpy(uint32_t desAddr, uint32_t srcAddr)
{
    uint32_t srcData;
    FMC->ISPCMD = FMC_ISPCMD_READ;
    FMC->ISPADDR = srcAddr;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
    srcData = FMC->ISPDAT;

    FMC->ISPCMD = FMC_ISPCMD_PROGRAM;
    FMC->ISPADDR = desAddr;
    FMC->ISPDAT = srcData;
    FMC->ISPTRG = 0x1;
    __ISB();
    while (FMC->ISPTRG & 0x1) ;
}

int main(void)
{
    uint16_t cp[4] = {0,0,0,0};
    uint32_t i = 0, j, checked = 0;
    uint32_t bsize, pnum;

    /* Unlock protected registers */
    SYS_UnlockReg();
    /* Init System, peripheral clock and multi-function I/O */
```

```
    SYS_Init();
    /* Init UART to 115200-8n1 */
    UART_Init();
    CLK->AHBCLK |= CLK_AHBCLK_ISPCKEN_Msk;
    FMC->ISPCTL |= FMC_ISPCTL_ISPEN_Msk;
    g_apromSize = GetApromSize();
    bsize = g_apromSize >> 1;
    pnum = bsize >> 12;
    GetDataFlashInfo(&g_dataFlashAddr, &g_dataFlashSize);
    SysTick->LOAD = 300000 * CyclesPerUs;
    SysTick->VAL   = (0x00);
    //using cpu clock
    SysTick->CTRL = SysTick->CTRL | SysTick_CTRL_CLKSOURCE_Msk | SysTick_CTRL_ENABLE_Msk;

    while (1)
    {
    /* Command catch */
        if ((bufhead >= 4) || (bUartDataReady == TRUE))
        {
            uint32_t lcmd;
            lcmd = inpw(uart_rcvbuf);

            if (lcmd == CMD_CONNECT)
            {
                goto _ISP;
            }
            else
            {
                bUartDataReady = FALSE;
                bufhead = 0;
            }
        }

    /* Command timeout and flash compare finished */
        if ((SysTick->CTRL & SysTick_CTRL_COUNTFLAG_Msk) && checked)
        {
         // Fix page
         for(i = 0; i < pnum; i++)
         {
             if(cp[i/16] & (1 << (i%16)))
             {
                 FMC->ISPCMD = FMC_ISPCMD_PAGE_ERASE;
                 FMC->ISPADDR = FMC_APROM_BASE + PAGE_SIZE * i;
                 FMC->ISPTRG = 0x1;
                 __ISB();
                 while (FMC->ISPTRG & 0x1) ;

                 for(j = 0; j < PAGE_SIZE; j += 4)
                 {
                     flashcpy(PAGE_SIZE * i + j, PAGE_SIZE * i + j + bsize);
                 }
             }
         }
            goto _APROM;
        }
```

```
    /* Compare flash page by page */
    if(!checked)
    {
        for(j = 0; j < PAGE_SIZE; j+=4)
        {
            if(flashcmp(i * PAGE_SIZE + j, i * PAGE_SIZE + j + bsize))
            {
                cp[i/16] |= 1 << (i%16);
                break;
            }
        }

        if(++i == pnum)
            checked = 1;
    }
    }

_ISP:

    while (1)
    {
        if (bUartDataReady == TRUE)
        {
            bUartDataReady = FALSE;
            ParseCmd(uart_rcvbuf, 64);
            PutString();
        }
    }

_APROM:
    SYS->RSTSTS = (SYS_RSTSTS_PORF_Msk | SYS_RSTSTS_PINRF_Msk);
    FMC->ISPCTL &= ~(FMC_ISPCTL_ISPEN_Msk | FMC_ISPCTL_BS_Msk);
    SCB->AIRCR = (V6M_AIRCR_VECTKEY_DATA | V6M_AIRCR_SYSRESETREQ);

    /* Trap the CPU */
    while (1);
}
```

3. 软硬体需求

3.1 软件需求

- BSP 版本
  - M480 Series BSP CMSIS V3.05.003
- IDE 版本
  - Keil uVersion 5.28
- 烧录工具
  - Nuvoton NuMicro ICP Programming Tools
  - Nuvoton NuMicro ISP Programming Tool(可选)

3.2 硬体需求

- NuMaker-PFM-M487 V3.0

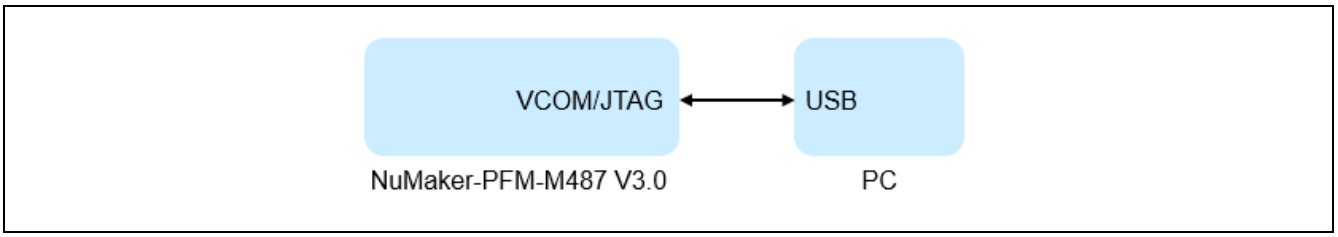

图 3-1 连接至电脑

4. 目录资讯

目录架构如下所示。

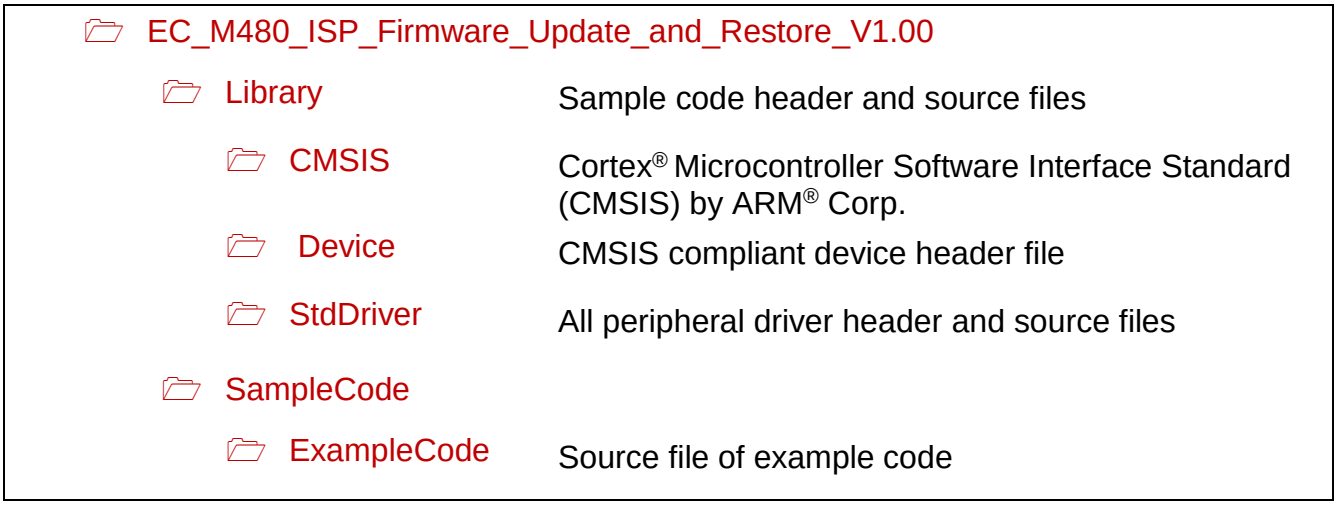

图 4-1 目录资讯

5. 范例程序执行

1. 编译范例程序。
2. 透过 ICE 连接 Nuvoton NuMicro ICP Programming Tool，选择 M480 Series。
   - 勾选 LDROM 以及 Options 内 Erase Whole Chip，点击 LDROM 选择编译后的文件后 Start 开始烧录 LDROM。
   - 勾选 Config 并取消勾选 LDROM 与 Options 内 Erase Whole Chip，点击 Setting 勾选 LDROM，确认后 Start 开始烧录 CONFIG。
   - 勾选 APROM 并取消勾选 Config，于 Offset 处填入 APROM_SIZE/2，点击 APROM 选择 BSP 内任意编译过的范例程序后 Start 开始烧录 APROM。
3. 烧录完成后程序将由 LDROM 启动并开始更新执行区块，执行结果即为写入备份区块的应用程序。

6. 修订纪录

Date	Revision	Description
2022.08.24	1.00	初始发布。

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.