NUC1262 CRC 验证代码完整性

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 NUC1262 来做代码完整性的自我验证
BSP 版本	NUC1262_Series_BSP_CMSIS_V3.00.001
开发平台	NuMaker-NUC1262SE V1.0

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

本范例代码使用 NUC1262 CRC 来做代码完整性的自我验证。

确保执行代码的完整性是一个终端产品应具备的安全条件之一。这可避免产品在执行过程中因为代码的不正确而产生异常的结果和造成死机，或者是因代码被窜改而造成数据被窃取。

本示例代码不需另接外部硬件模块。

需下载 SRecord 工具包 (http://srecord.sourceforge.net/)，使用 srec_cat.exe 来计算源代码的 CRC32 校验值，并另外产生一个包含源代码和校验值的 binary 格式文件，再将此 binary 格式文件转换成可直接由 Keil IDE 做烧录的 HEX 格式文件。

1.1 原理

因本示例代码不需另接外部硬件模块，此章节将着重于介绍验证源代码 CRC32 校验值的步骤，和在软件工程中如何使用 srec_cat.exe 来产生包含 CRC32 校验值文件的说明。

应用程序源代码和其 CRC32 校验值位于芯片内存的位置关系如图1-1所示。

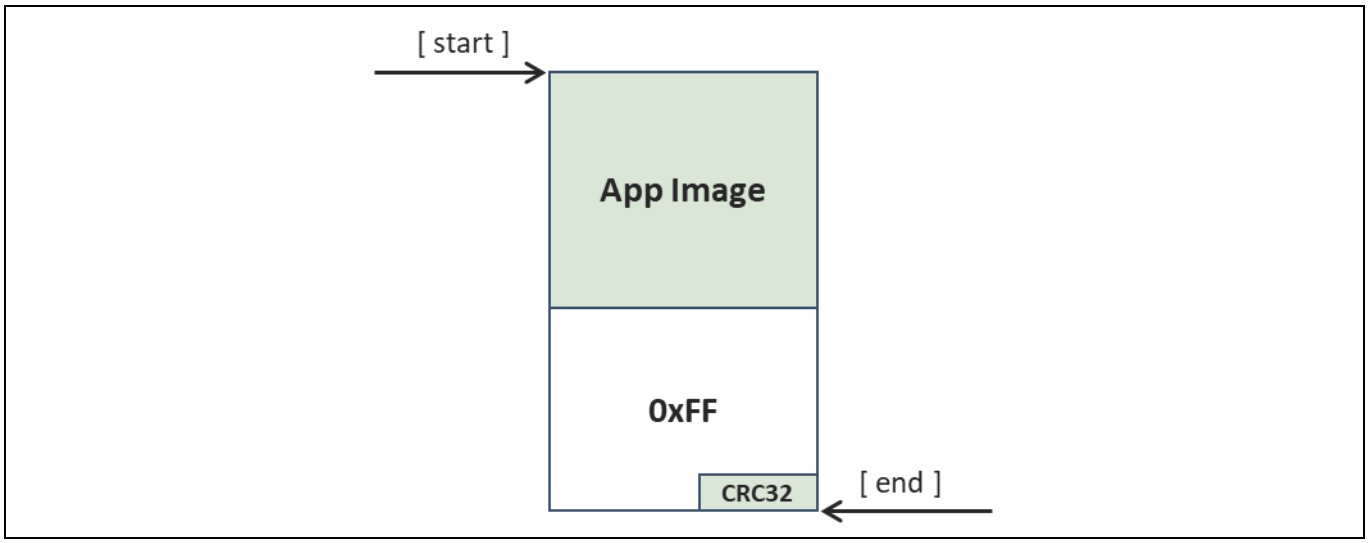

图 1-1 应用程序源代码 和 CRC32 校验值

验证源代码 CRC32 校验值的步骤如图1-2所示。

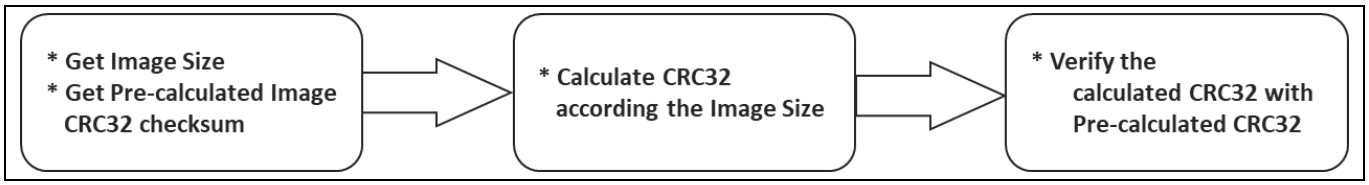

图1-2 验证源代码 CRC32 校验值

[Get Image Size] :

可由工程内的 ER_RO 和 ER_RW 长度计算出源代码的字节数。

(可参考 NUC1262_Verify_Image_CRC32.map)

[Get Pre-calculated Image CRC32 checksum] :

可经由定义在 main.c 内 IMAGE_CRC32_ADDRESS 的地址读出源代码的校验值。

[Calculate CRC32 according the Image Size] :

根据源代码的字节数计算出 CRC32 校验值。

[Verify the calculated CRC32 with Pre-calculated CRC32] :

对比计算出来的校验值和预先写入的校验值是否相同。

本示例代码将使用 GenCRC.bat 来执行 srec_cat.exe 的相关命令，GenCRC.bat 的内容和在工程内的配置如图1-3所示。

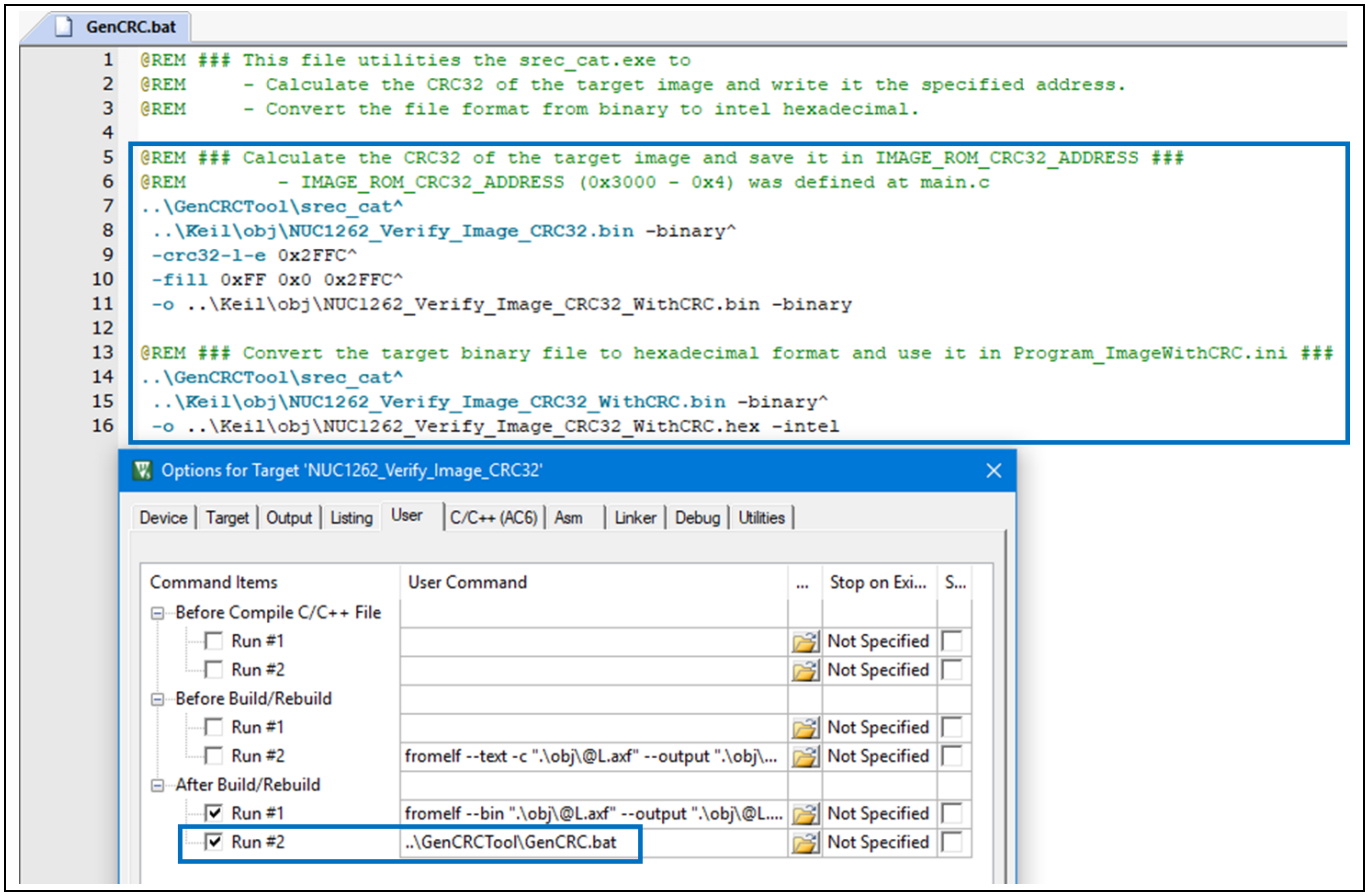

图 1-3 GenCRC.bat 内容和配置

在编译工程并产生源代码的 binary 格式文件后，将会执行 GenCRC.bat 内的第一个 srec_cat 命令。此命令会先做源代码 CRC32 校验值的计算，并将此校验值写入到指定的地址 (如: 0x2FFC)，之后便可产生一个新的 binary 格式文件，包含源代码和其 CRC32。而位于源代码之后和存放校验值之间的区域将被写为全 0xFF。第二个 srec_cat 命令则是将此新的 binary 格式文件转成 intel HEX 格式的文件，其目的是为了可直接由 Keil IDE 做烧写源代码和校验值到指定的位置。实现 Keil IDE 做烧写所用到的文件为 ProgramTargetImageWithCRC.ini，其内容和在工程内的配置如图1-4所示。

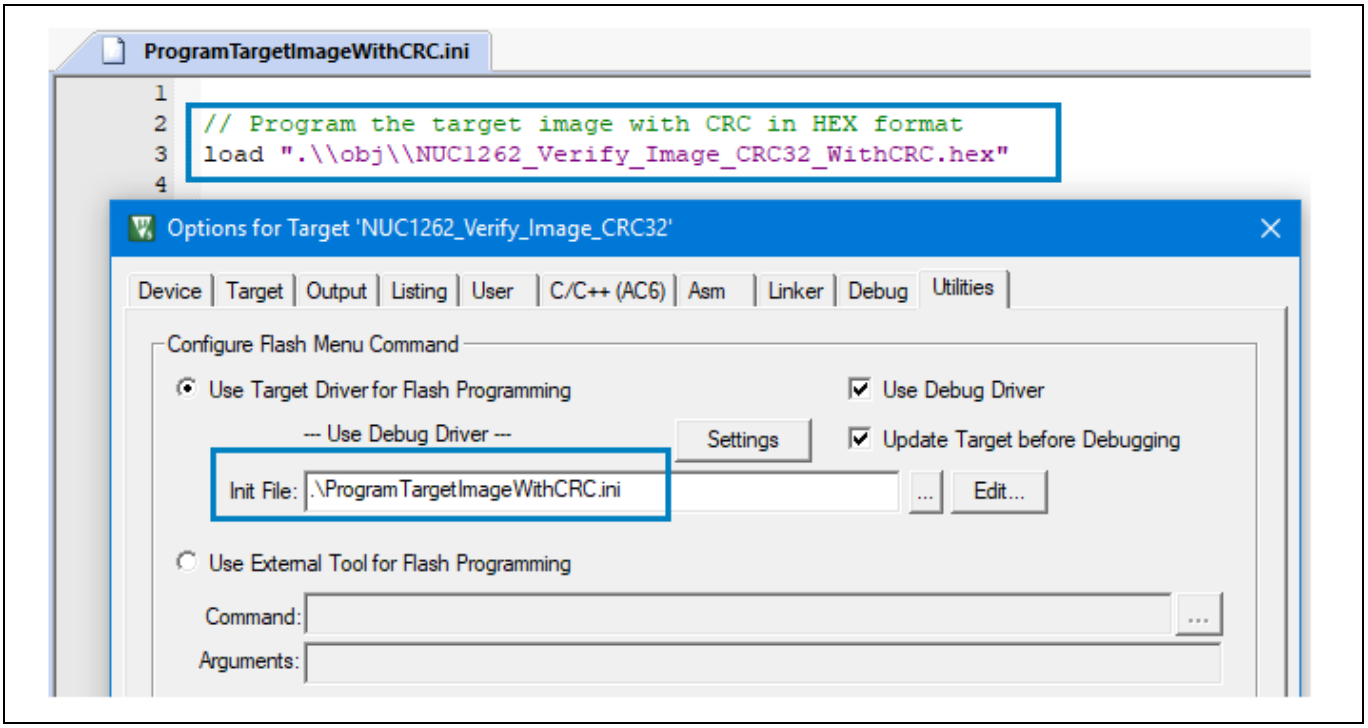

图 1-4 ProgramTargetImageWithCRC.ini 内容和配置

1.2 执行结果

本示例代码执行编译和烧录的结果如图1-5所示。执行 GenCRC.bat 内的两次 srec_cat 命令和烧写 ProgramTargetImageWithCRC.ini 内指定的 HEX 文件。

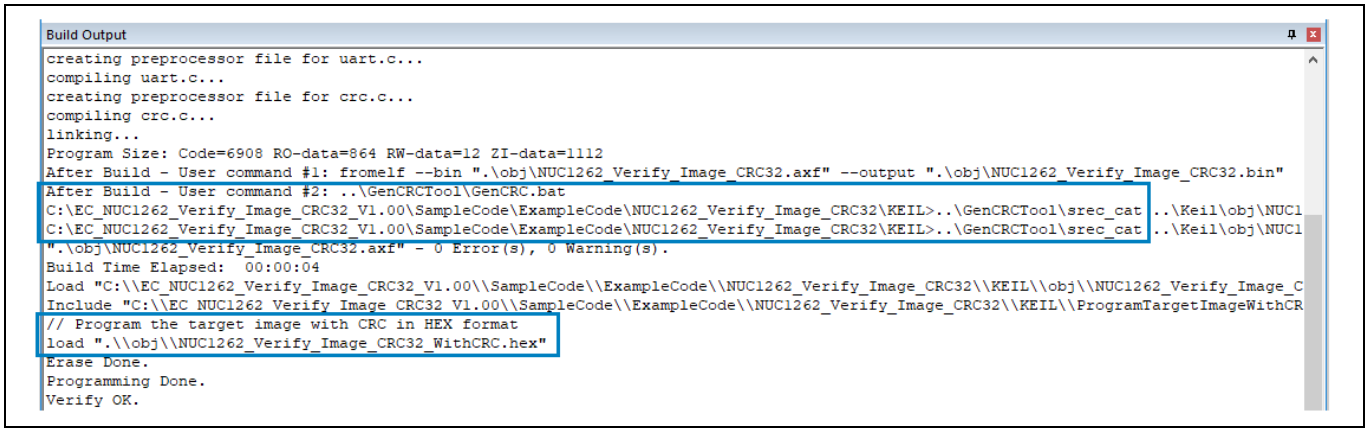

图 1-5 执行编译和烧录结果

图1-6说明开启 NUC1262_Verify_Image_CRC32_WithCRC.bin 文件，可看到源代码数据字节数为 0x1e68 和其 CRC32 校验值为 0x129c3e3f。

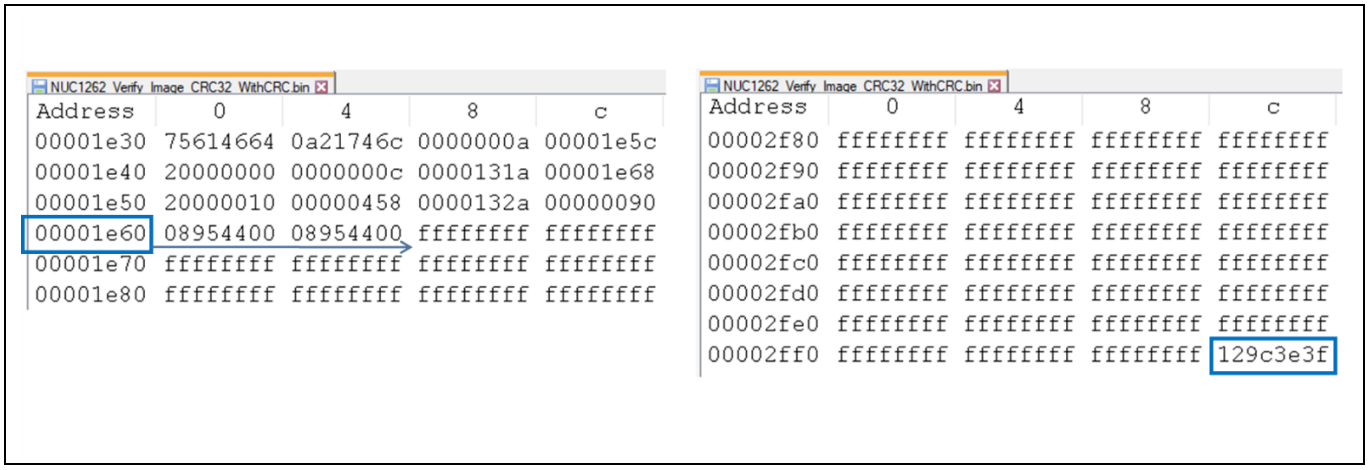

图 1-6 源代码 CRC32 校验值

执行结果会由 PC UART 打印出源代码的字节数，CRC32 校验值和根据源代码字节数计算得到的 CRC32 校验值等信息，如图1-7所示。

```
[ Perform CRC32 calculation to verify image integrity ]
    # CPU @ 72000000 Hz.
    # The target Image ROM size is 0x1e68 (7784).
    # The pre-calculated CRC32 value 0x129c3e3f is stored in 0x00002ffc.

Start to perform HW CRC to claculate the Image ROM CRC32 checksum......
        - CRC32 checksum: 0x129c3e3f

[Compare Image ROM checksum ... PASS]
```

图 1-7 执行结果

2 代码介绍

计算源代码字节数的函数，代码位于 main.c。

```
/**
  * @brief      Get Image Size
  * @param[in]  None
  * @return     Image ROM size (bytes)
  * @details    This function gets the target image size according to the Image RO and RW size.
  */
static uint32_t GetImageSize(void)
{
    extern uint32_t Image$$ER_RO$$Length; // Execution Region ER_RO can be found in NUC1262_Verify_Image_CRC32.map
    extern uint32_t Image$$ER_RW$$Length; // Execution Region ER_RW can be found in NUC1262_Verify_Image_CRC32.map
    uint32_t u32ROSize, u32RWSize;

    u32ROSize = (uint32_t)&Image$$ER_RO$$Length;
    u32RWSize = (uint32_t)&Image$$ER_RW$$Length;

    //printf("\tRO: %d, RW: %d.\n", u32ROSize, u32RWSize);

    return (u32ROSize + u32RWSize);
}
```

初始化系统后，经由IMAGE_CRC32_ADDRESS地址可取得源代码 CRC32 校验值，再和使用 CRC模块计算出来的 CRC32 校验值做对比。代码位于 main.c。

```
    ……
 #define IMAGE_CRC32_ADDRESS         (0x3000 - 0x4) // 0x2FFC, the flash address to store pre-calculated CRC32
    ……
int main(void)
{
    volatile uint32_t addr;
    uint32_t u32ImageROMSize;
    uint32_t u32PreChks, u32CalChks;
```

```
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Init UART0 for printf */
    UART0_Init();

    /* Get the target image size and pre-calculated CRC32 checksum */
    u32ImageROMSize = GetImageSize();
    u32PreChks      = inpw(IMAGE_CRC32_ADDRESS);

    printf("\n\n[ Perform CRC32 calculation to verify image integrity ]\n");
    printf("    # CPU @ %d Hz.\n", SystemCoreClock);
    printf("    # The target Image ROM size is 0x%x (%d).\n", u32ImageROMSize,
u32ImageROMSize);
    printf("    # The pre-calculated CRC32 value 0x%08x is stored in 0x%08x.\n",
u32PreChks, IMAGE_CRC32_ADDRESS);
    printf("\n");



    printf("Start to perform HW CRC to claculate the Image ROM CRC32 checksum......\n");

    /* Configure CRC controller for CRC32 calculation */
    CRC_Open(CRC_32, (CRC_WDATA_RVS | CRC_CHECKSUM_RVS | CRC_CHECKSUM_COM), 0xFFFFFFFF,
CRC_CPU_WDATA_32);

    /* Calculate the Image ROM CRC32 checksum */
    for(addr=0; addr<u32ImageROMSize; addr+=4)
    {
        CRC_WRITE_DATA(inpw(addr));
    }
    u32CalChks = CRC_GetChecksum();
    printf("\t- CRC32 checksum: 0x%08x\n", u32CalChks);

    if(u32PreChks == u32CalChks)
    {
        printf("\n[Compare Image ROM checksum ... PASS]\n");
    }
    else
    {
```

```
        printf("\n[Compare Image ROM checksum ... WRONG]\n");
        printf("\tPre-calculated checksum:  0x%08x\n", u32PreChks);
        printf("\tCalculate CRC32 checksum: 0x%08x\n", u32CalChks);
    }

    /* Disable CRC function */
    CRC->CTL &= ~CRC_CTL_CRCEN_Msk;

    while(1) {}
}
```

3 软件与硬件需求

- 软件需求
  - BSP 版本
    - NUC1262_Series_BSP_CMSIS_V3.00.001
  - IDE 版本
    - Keil uVersion 5.28
- 硬件需求
  - 电路组件
    - NuMaker-NUC1262SE V1.0
  - 示意图
    - 将 UART0 TX (PB.13) pin 脚连接到 PC UART RX，以显示示例代码的执行结果

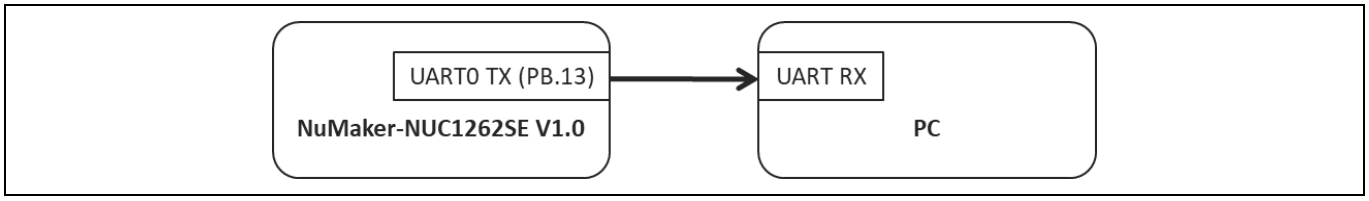

图 3-1 NUC1262 UART0 TX 和 PC UART RX 连接

4 目录信息

EC_NUC1262_Verify_Image_CRC32_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

图 4-1 目录信息

5 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 NUC1262_Verify_Image_CRC32.uvprojx。

2. 进入编译模式界面

   a. 编译

   b. 下载代码至内存

   c. 进入 / 离开仿真模式

3. 进入仿真模式界面

   a. 执行代码

6 修订纪录

Date	Revision	Description
2021.09.24	1.00	1. 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.