具UID进阶保护的IO切换范例

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	透过判断自定义 UID 决定 IO 是否切换的范例代码
BSP 版本	M051 Series BSP CMSIS v3.01.001
开发平台	NuTiny–EVB–M051_V3.0

1 功能介绍

1.1 简介

Nuvoton M051 系列微处理器提供UID (Unique Identifier)，开发者在应用上可透过识别UID的正确性，决定应用是否继续执行，确保产品不被任意复制仿冒。单纯的UID识别安全性似乎不足，因此将提供一个可自行增加识别复杂度范例，开发者可以基于微处理器的UID，再进行自定义的运算，使用运算过的UID识别有更佳的安全性。

1.2 原理

为了方便开发可直接将芯片UID读出并加以运算，接着合并应用程序再写入芯片，此范例分为三个区块，请参考图1-1，下列将对每区块做说明:

1. **NuLink.exe** : Nuvoton 提供的简易工具，必须搭配 NuLink ICE 使用，在此使用 NuLink.exe 进行 UID 读取及烧入修改后的应用。
2. **TCDEF.exe** : Visual C++ 所编译，提供将 NuLink 读回的 UID 经自定义运算后，与 Keil 所产生的应用 bin 档做合并。
3. **APROM.bin** : 开发者应用专案，必须包含预留运算后 UID 的区域，及如何识别运算后 UID 的判断程序。

方便操作，关于上述流程，最后会使用批文件将流程整合，若开发者已完成UID自定义运算，及如何识别UID自定义运算后的程序，直接操作批文件即可。

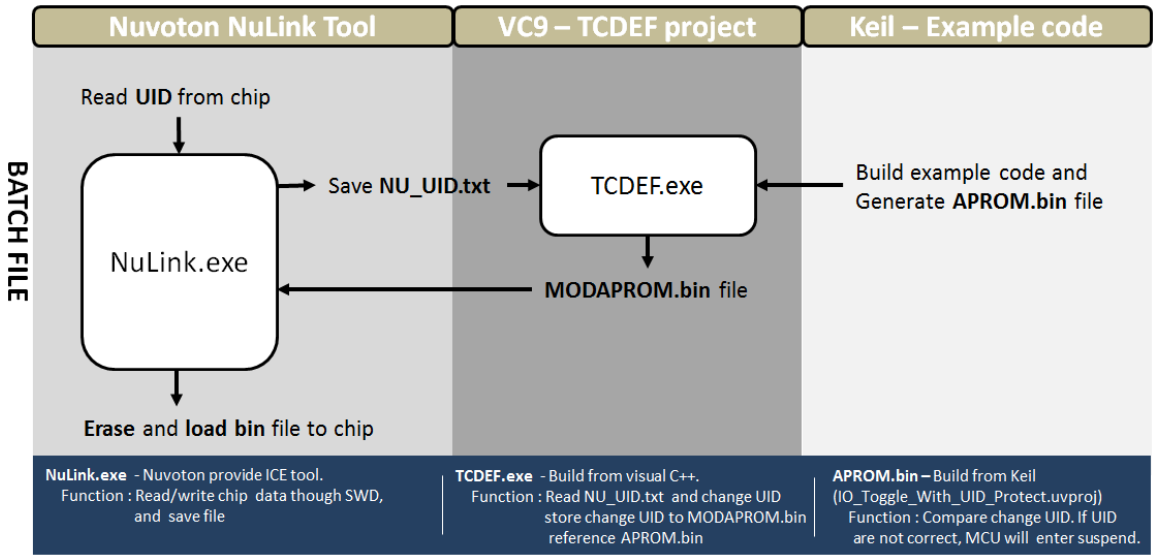

图1-1范例流程简介

从芯片读出的UID执行完自定义运算后，将存放于APROM保留位置，APROM保留位置在 example code 与 TCDEF.exe 定义要相同，如图1-2 。

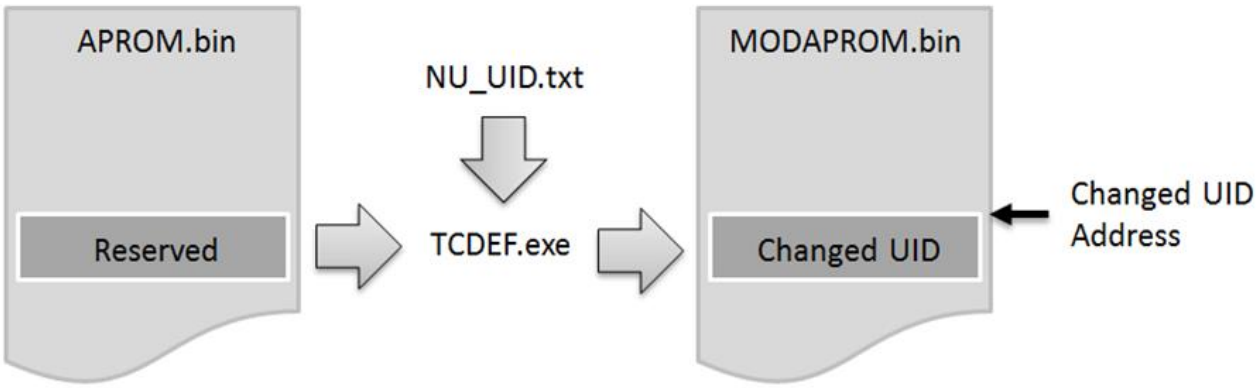

图1-2自定义运算后UID的存放位置

此范例自定义运算的部分预留，提供整个流程方法，最后比较芯片UID及存放于APROM的UID是否相同。

1.3 执行结果

1. 执行 IO_Toggle_With_UID_Protect.bat，显示如下图 1-3 command line 视窗，执行上述所介绍之流程，完成烧入后，按下任意键将结束 command line 视窗，晶片将重新执行自定义 UID 的应用程序。

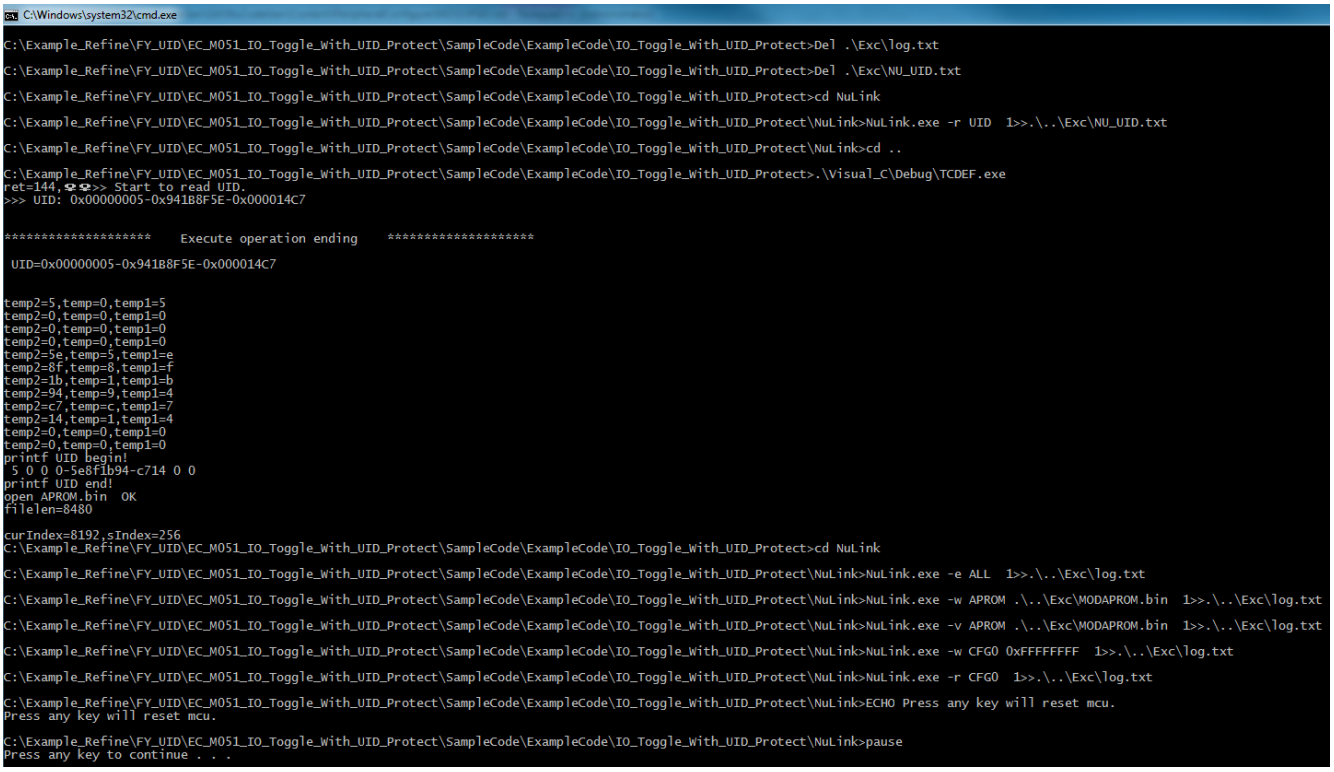

圖 1-3 批文件執行結果

2. 比对芯片 UID 与 自定义运算后 UID 相同，芯片将持续由 PB13 Toggle。

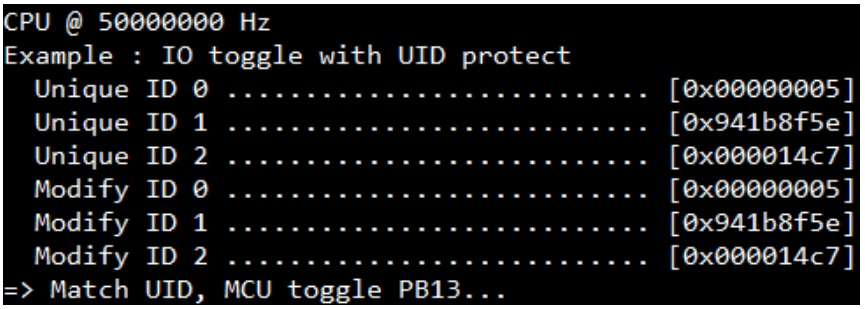

图 1-4 由 UART 输出范例执行结果

2 代码介绍

2.1 Batch file - IO_Toggle_With_UID_Protect.bat

删除 log.txt and NU_UID.txt file

```
Del .\Exc\log.txt
Del .\Exc\NU_UID.txt
```

读取芯片 UID 并储存于 NU_UID.txt file

```
NuLink.exe -r UID >> .\..\Exc\NU_UID.txt
```

执行 TCDEF.exe

```
.\Visual_C\Debug\TCDEF.exe
```

抹除芯片

```
NuLink.exe -e ALL >> .\..\Exc\log.txt
```

写入 MODAPROM.bin 并验证

```
NuLink.exe -w APROM .\..\Exc\MODAPROM.bin >> .\..\Exc\log.txt
NuLink.exe -v APROM .\..\Exc\MODAPROM.bin >> .\..\Exc\log.txt
```

写入 Config0 并读出存于 log.txt

```
NuLink.exe -w CFG0 0xFFFFFFFF >> .\..\Exc\log.txt
NuLink.exe -r CFG0 >> .\..\Exc\log.txt
```

芯片重新执行

```
NuLink.exe -reset
```

2.2 TCDEF.exe

自定义UID存放开始位置及位移

```
#define APROMUIDADD                 0x2000  //Store UID change data start address of APROM
#define UIDOFFSET                   0x100   //Store UID change data offset of APROM
```

开发者自定义UID运算编辑函式

```
void changUIDData()
{
    int i = 0;
    for (i = 0; i < 12; i++);
    /************* User change UID implement ***************/
}
```

2.3 KEIL – Example code

自定义UID存放位置保留及定义，必须与TCDEF.exe 定义相同

```
#define UID_START_ADDR              0x00002000   // UID start address of APROM
#define UID_ADDR_OFFSET             0x100        // UID address offset of APROM

__attribute__((at(UID_START_ADDR + UID_ADDR_OFFSET))) static const uint32_t
gau32UIDAddr[4]= {2,2,2};
```

判断芯片UID是否与自定义运算后的UID相同，若有自定义运算，需自行加入如何判读自定义UID。

```
/************* User change UID compare implement ***************/
/* Compare original UID and change UID */
if((au32mcuUID[0] == au32modfiyUID[0]) && (au32mcuUID[1] == au32modfiyUID[1]) &&
    (au32mcuUID[2] == au32modfiyUID[2]))
    {
        printf("=> Match UID, MCU toggle PB13...\n");
        while(1)
        {
            Delay_count(500);
            P35 = 0;
            Delay_count(500);
            P35 = 1;
        }
    }
```

3 软件与硬件环境

- 软件环境
  - BSP 版本
    - M051 Series BSP CMSIS v3.01.001
  - IDE 版本
    - Keil uVersion 5.24
  - Visual C++ 版本
    - Microsoft Visual C++ 2008
  - NuLink.exe 版本
    - Nuvoton NuLink Command Tool v2.06.6871

- 硬件环境
  - 电路组件
    - NuTiny–EVB–M051_V3.0

4 目录信息

- EC_M051_IO_Toggle_With_UID_Protect
  - 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
      - IO_Toggle_With_UID_Protect
        - KEIL — Example files of KEIL project.
        - Exc — Temporary and log files.
        - NuLink — Nuvoton NuLink tools.
        - Visual_C — Visual C++ project for UID changed.

5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 IO_Toggle_With_UID_Protect.uvproj。
2. 进入编译模式接口
   a. 编译
3. 执行 IO_Toggle_With_UID_Protect.bat
   a. 确认芯片及 NuLink 已正确连接至 PC.
   b. 执行 UID 运算及烧入程序
   c. 重新执行芯片程序

6 修订纪录

Date	Revision	Description
Jul. 9, 2019	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.