NuMicro® Cortex®-M Code Protection

Application Note for 32-bit NuMicro® Family

Document Information

Abstract	This document is aimed to introduce many kinds of ways provided by Nuvoton to protect code or data from being pirated. Meanwhile, the ICP, ISP and NuGang Programming tools and software programming sequences that can help users to utilize in the system or in mass production are described in this document.
Apply to	NuMicro® Cortex®-M0/M4 Series

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

www.nuvoton.com

Table of Contents

1 Overview

Nowadays microcontrollers (MCUs) can be seen anywhere as they are flexible, robustness and easy to develop. Users always consider “code or data security” an important factor while the MCU is implemented in the system.

In view of code protection, there is an easy way to force flash inside the MCU to be locked after mass production. However, some applications need advanced functions in case someone tries to break the chassis to steal the important data therein.

Nuvoton considered “code or data security” in the very early stage of design. To protect the flash in NuMicro® Cortex®-M0/M4 series, the security lock bit in “User Configuration” is a simple but powerful way to lock the chip data after mass production.

User can update the MCU program memory under software control by Nuvoton ICP programming tool without removing the mounted MCU chip from a target PCB. The ICP tool provides a very friendly way to all the flash programming settings, including User Configuration.

Nuvoton also provides a particular register, spare register, and a corresponded pin, Tamper pin, for customers who have specific key or data to protect. While the voltage change happened on the Tamper pin, the data in the spare register will be automatically cleared.

In addition, Nuvoton provides special mechanism, UCID (Unique Customer Identification), to strengthen code protection. Users could add this special and customized ID into source code to strengthen the barrier of stealing code by others. Meanwhile, no one else but customer signed the non-disclosure agreement (NDA) could purchase the chip with same UCID.

In summary, this document is aimed to introduce five ways to protect the code or data from being pirated for NuMicro® Cortex®-M0/M4 series. Users could easily implement these mechanisms in each system and select a suitable way based on different applications.

- Security lock in Config0 register
- Code protection by Nuvoton tools
- 96-bit Unique Identification (UID)
- 128-bit Unique Customer Identification (UCID)
- RTC spare register and Tamper / Snooper pin

2 Security Lock

The protection mechanism can prevent the original source code from being stolen. In addition, Nuvoton provides other applications to enhance and strengthen the protection of user's source code. Please refer to the UID & UCID Security Protection section for details.

2.1 Associated Registers

Nuvoton provides a function for the NuMicro®-M0/M4 to lock the chip securely by means of the user configuration register, Config0[1], LOCK bit. As shown in Table 2-1, if the LOCK bit is set as 0, user can only get the chip's data in Config0 and Config1 through Nuvoton's ICP programming tool, NuGang programmer, or a third party programming tool, and the other data in flash will be shown as 0xFFFF_FFFF.

Config0 (Address = 0x0030_0000)

31	30	29	28	27	26	25	24
CWDTEN[2]	CWDTPDEN	Reserved		CGPFMFP	CFOSC		
23	22	21	20	19	18	17	16
CBODEN	CBOV		CBORST	Reserved			
15	14	13	12	11	10	9	8
Reserved					CIOINI	Reserved	
7	6	5	4	3	2	1	0
CBS		Reserved	CWDTE[1:0]		DFVSEN	LOCK	DFEN

[1]	**LOCK**	**Security Lock** 0 = Flash data is locked. 1 = Flash data is not locked. When flash data is locked, only device ID, CONFIG0 and CONFIG1 can be read by writer and ICP through serial debug interface. Others data is locked as 0xFFFFFFFF. ISP can read data anywhere regardless of LOCK bit value. User need to erase whole chip by ICP/Writer tool or erase user configuration by ISP to unlock.

Table 2-1 LOCK Bit in Config0 Register

3 Code Protection by Nuvoton Tools

3.1 Enable Security Lock via ICP or ISP Tool

Besides the third party writer, user can also use Nuvoton's NuMicro ICP or ISP Programming Tool to lock the source code during the process of chip programming. Moreover, Nuvoton provides an interface for user to get the contents of flash data from the On-board Flash window wherein the flash data is changed to 0xFFFF_FFFF after the ICP tool finishes "Security Lock" process. Through such a convenient tool, source code protection can be greatly improved. If someone wants to read the flash data of the locked chip, the ICP tool will pop out a warning window to enforce the whole chip erase. Figure 3-1 shows the flow of how to set the chip locked.

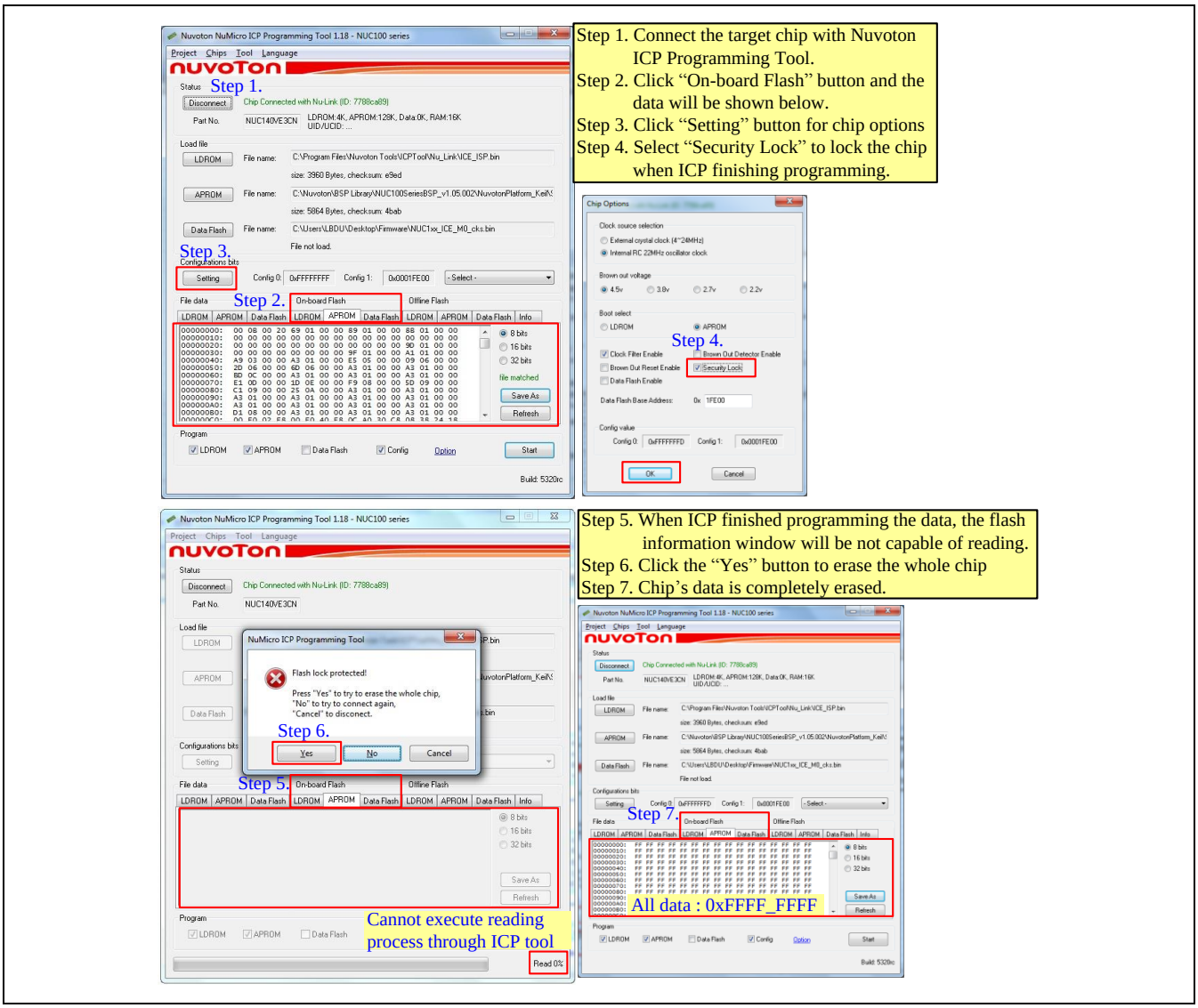

Figure 3-1 "Security Lock" Flow in the ICP Tool

3.2 Code Protection in ICP Offline Mode

Except enabling the **Security Lock** option to set the chip locked for source code protection as described in the previous section, Nuvoton provides another chip protection mechanism in Offline mode with the ICP tool.

When using Nuvoton's "Nu-Link" or "Nu-Link Pro" programming tools (as shown in Figure 3-2) to program a chip in Offline mode, there are two methods to protect chips — "setting password for offline data" or "limiting the number of offline programming". Figure 3-3 shows the steps of setting the password or the maximum number of programming in Offline mode.

Figure 3-2 Nu-Link and Nu-Link Pro Programming Tool

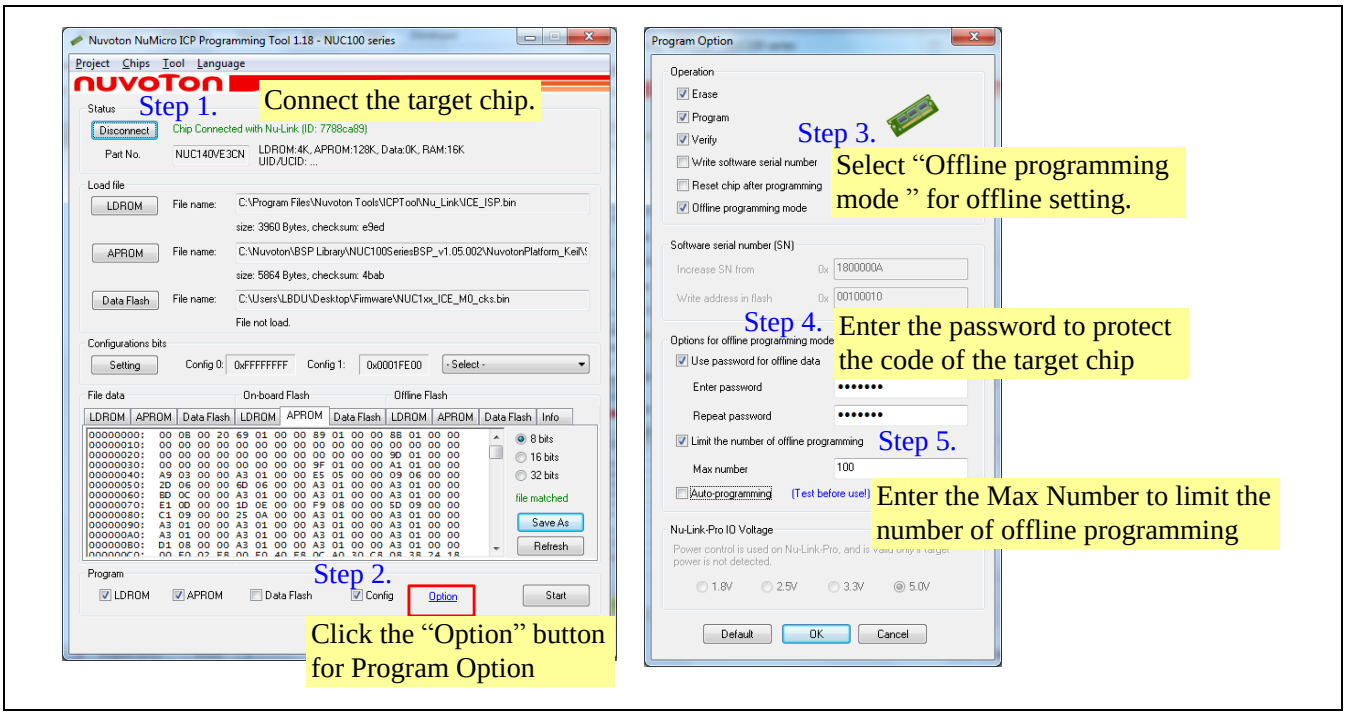

Figure 3-3 ICP Offline Mode

For Step 4 in Figure 3-3, user can set any form of password and re-enter the password again to confirm the setting, and then click the Start button to execute the chip programming

process. When the ICP tool detects the chip is connected next time, a request form will appear to ask the user to enter the right password to unlock the chip. If the entered password does not match the preset password, the request window will not disappear until the right one is entered. If user wants to remove the “Password” setting, just enter the right password and undo the click in Step 4 or erase the whole chip data could achieve.

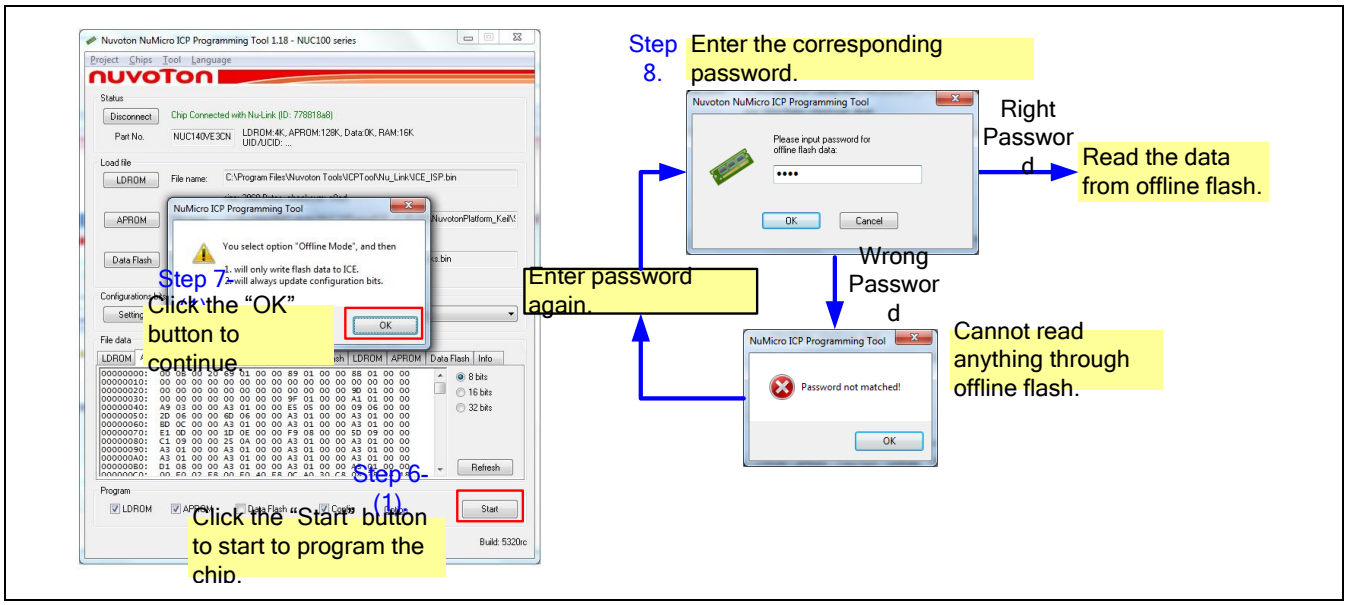

Figure 3-4 Password Function in ICP Tool Offline Mode

Another offline protection mechanism is to limit the number of programming chip when using the “Nu-Link” or “Nu-Link Pro” programming tools to program a chip. When the number of chip programming meets the limitation number, the “Nu-Link” or “Nu-Link Pro” will not be allowed to program any other chips. User needs to erase and re-program “Nu-Link” or “Nu-Link Pro” to continue chip programming again. User can get the flash data in “Nu-Link” or “Nu-Link Pro” from the Offline Flash window as shown in Figure 3-5.

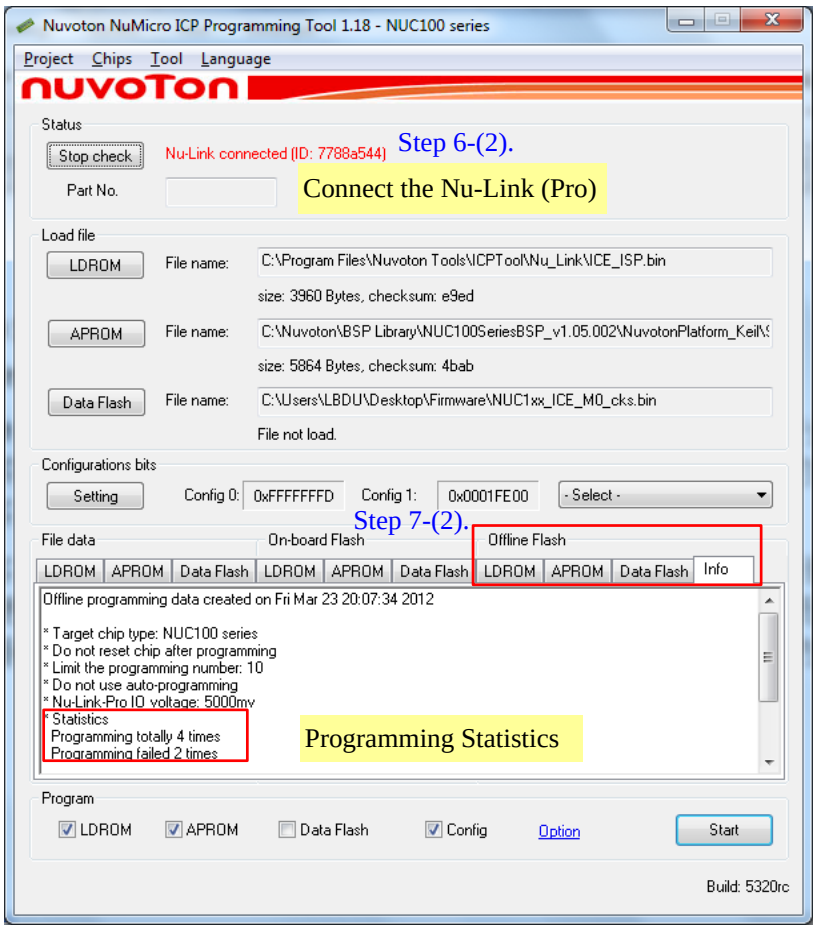

Figure 3-5 Offline Flash Data in the ICP Tool

Both setting the password or limiting the maximum number of chip programming in Offline mode provide further protection for chip programming.

4 UID & UCID Security Protection

4.1 What is UID?

UID (Unique Identification) stands for a specific code for every chip dispatched from Nuvoton, just like an identification card which is unique for everyone. UID represents chip's part number and date of dispatching with length of 96 bits.

Why UID is so special for the chip's encryption issue? User may worry or concern about if any industrial agent or person with intensions steals the source code when a commodity is under development or developed stage. Then, the person may try to get the same IC to achieve the same function as user designed. Therefore, Nuvoton provides a unique ID for every NuMicro®-M0/M4 chip so that user could employ UID into source code. The UID protection mechanism will be introduced in the next section.

4.2 UID Security Protection

To enhance the degree of safety for user's source code, each NuMicro®-M0/M4 chip dispatched from Nuvoton will be planted with a unique ID in the chip. User can put UID, which is dealt with DES (Data Encryption Standard), into Data Flash or some specified area. Also, users can design the standard of encryption by themselves to protect the source code in the chip from being stolen and produced by other people with specific intensions.

Figure 4-1 shows the flow that UID needs to be dealt with DES and put into Data Flash. Then user can add a judging method in the firmware code to compare the current chip's UID with the previous one. If the result is not the same, the program will fall into dead loop. Consequently, it will greatly prevent user' commodity to be mass produced from source code being stolen, under this double protection mechanism from UID (Nuvoton) and DES (user). Moreover it also significantly increases user's confidence and reliability to use Nuvoton IC.

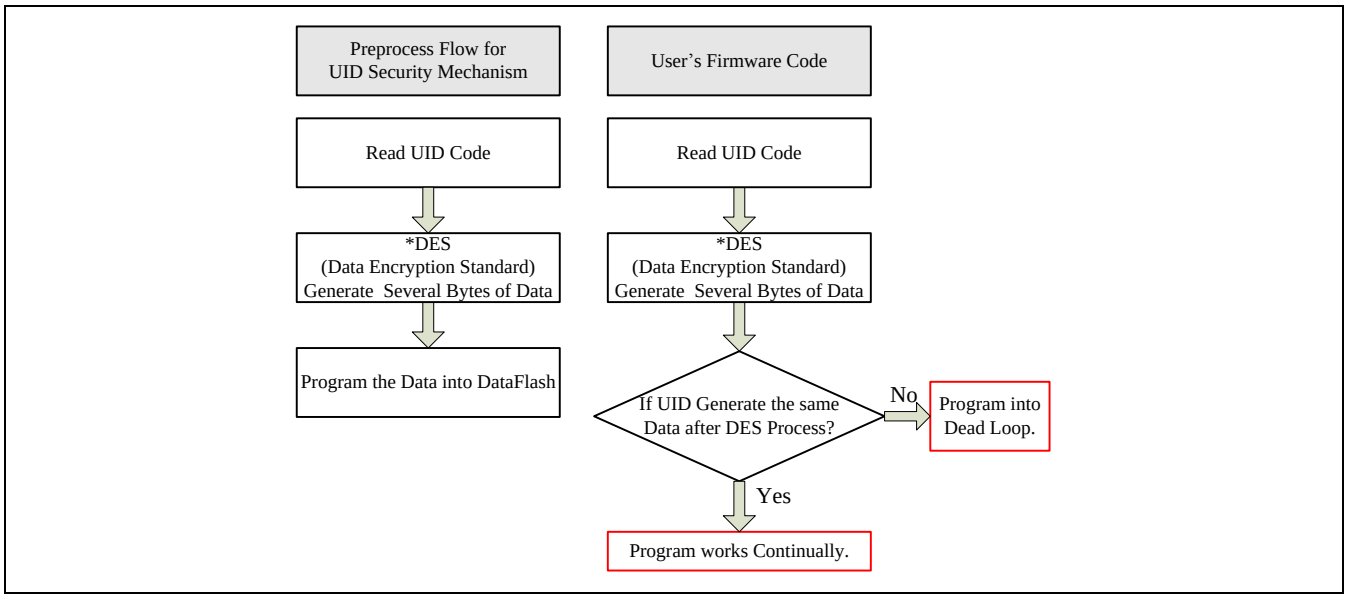

Figure 4-1 UID Protection Flow

4.3 What is UCID?

UCID (Unique Customer ID) is an initiative technique provided by Nuvoton to protect chip's source code. In addition to the previous UID section, user can apply for customized UCID with Nuvoton to ensure chip safety.

4.4 UCID Security Protection

Firstly, user provides the product number or specific code to put into the UCID. User data will be encoded and the conversion result will be planted into the chip to become customized and highly protected. Figure 4-2 shows the flow about how UCID protects the chip's source code, and Figure 4-3 shows the information to be put into the UCID.

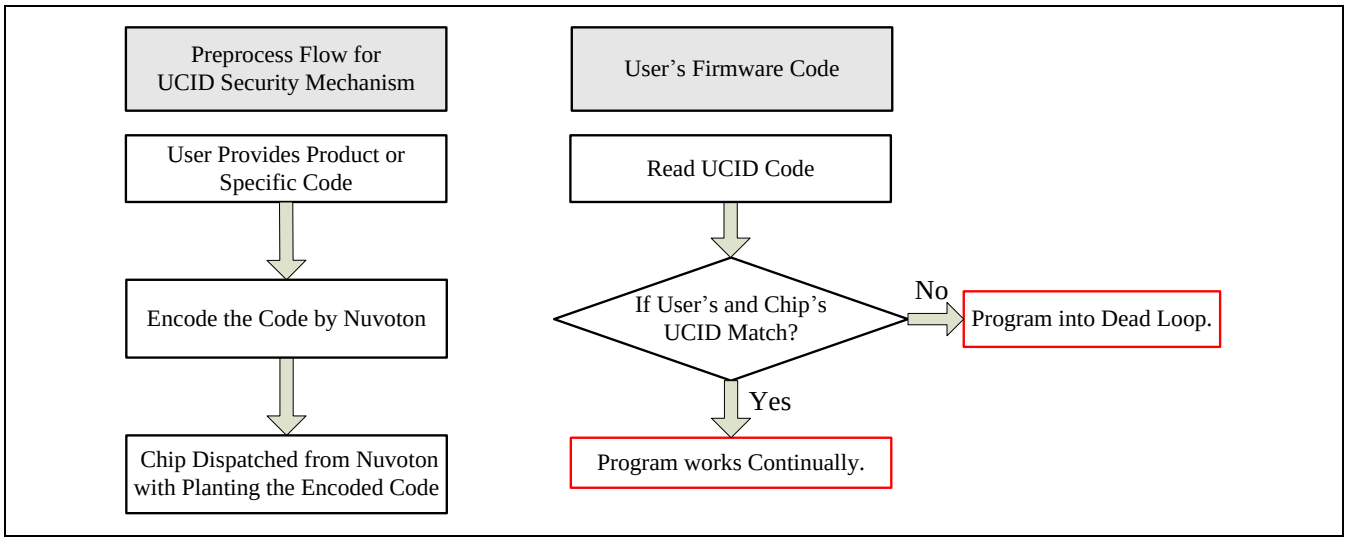

Figure 4-2 UCID Protection Flow

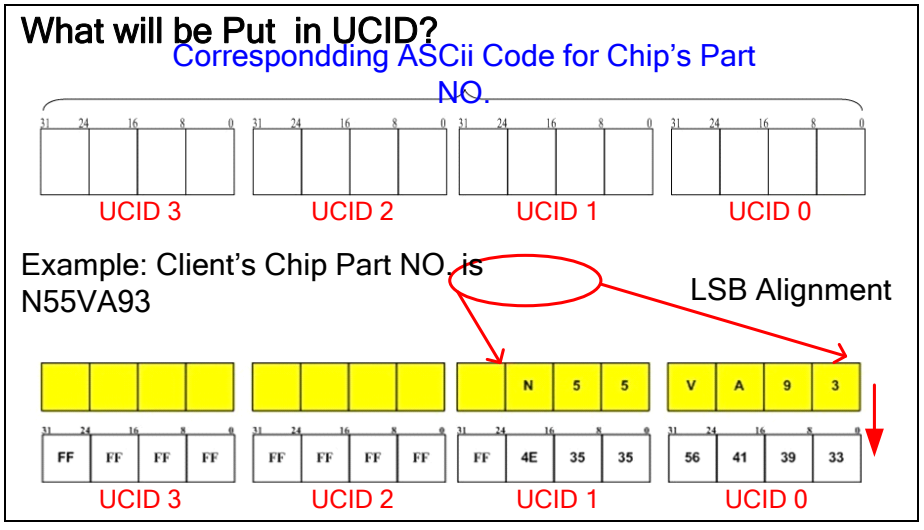

Figure 4-3 UCID Information

User can connect to a chip through Nuvoton's "NuMicro ICP or ISP Programming Tool" to get the current UCID in the chip as shown in Figure 4-4.

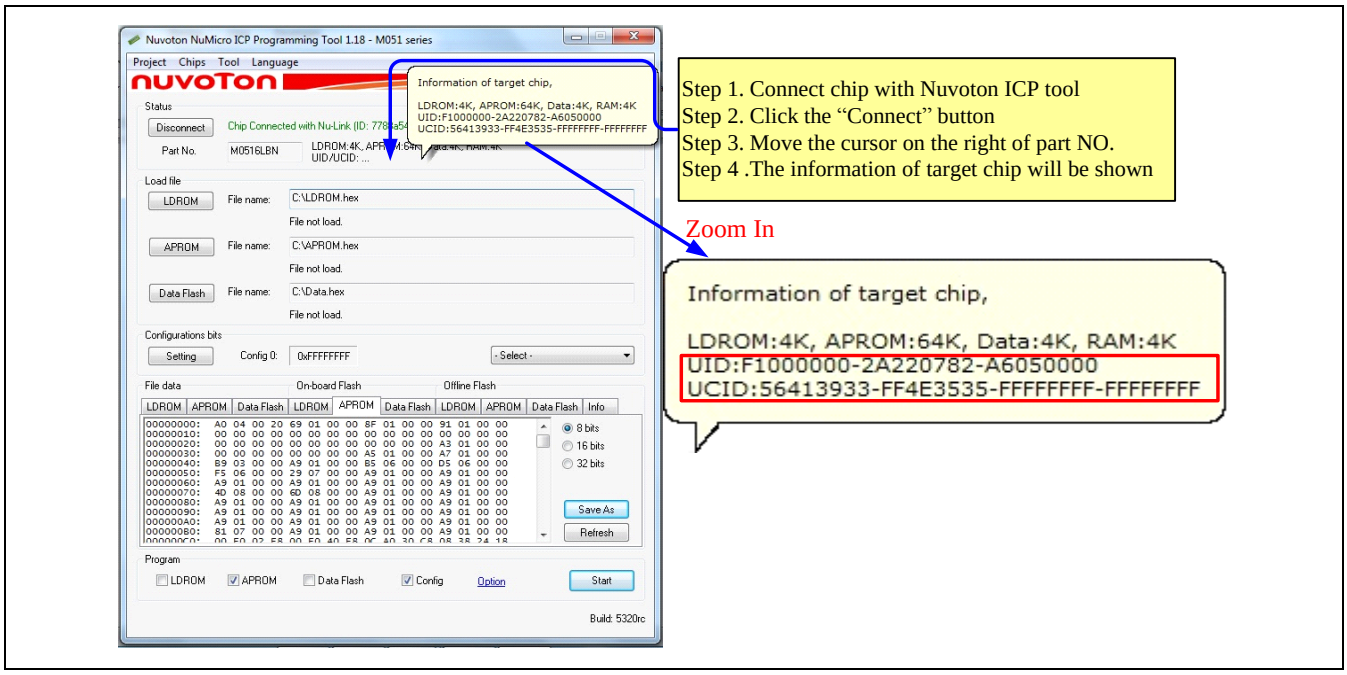

Figure 4-4 UID and UCID Data via ICP Tool

As to the protection mechanism for UCID, user can add a judging method in the firmware code to compare the current chip's UCID with the one encoded by Nuvoton. If the result is not the same, the program will fall into dead loop. Even user connects the chip with the ICP tool, an error window will appear to enforce the whole chip erase. According to this particular protection mechanism, the possibility of source code being stolen will be decreased, and product competition will be enhanced.

The NuMicro®-M0/M4 series chips from Nuvoton are normally dispatched without UCID and top-printed on chips. If user wants to add this protection mechanism to the purchased chips, please contact numicro@nuvoton.com for further information.

4.5 How to Read UID & UCID?

Before reading UID or UCID, user needs to unlock a register to get the access to control the flash related register first and then enable ISP functions.

```
SYS_UnlockReg();

void FMC_Open(void)
{
    FMC->ISPCON |=  FMC_ISPCON_ISPEN_Msk;
}
```

The following illustrates the programming sequences about how to read UID & UCID back via FMC related commands.

```
//This function reads one of the three UID.
uint32_t FMC_ReadUID(uint32_t u32Index)
{
    FMC->ISPCMD = FMC_ISPCMD_READ_UID;
    FMC->ISPADR = 0x04 * u32Index;          //u32Index must be 0, 1, or 2.
    FMC->ISPTRG = FMC_ISPTRG_ISPGO_Msk;

    while (FMC->ISPTRG & FMC_ISPTRG_ISPGO_Msk) ;

    return FMC->ISPDAT;
}
```

Table 4-1 Macro of Reading UID

```
//This function reads one of the three UID.
uint32_t FMC_ReadUCID(uint32_t u32Index)
{
    FMC->ISPCMD = FMC_ISPCMD_READ_UID;
    FMC->ISPADR = (0x04 * u32Index) + 0x10; //u32Index must be 0, 1, 2 or 3.
    FMC->ISPTRG = FMC_ISPTRG_ISPGO_Msk;

    while (FMC->ISPTRG & FMC_ISPTRG_ISPGO_Msk) ;

    return FMC->ISPDAT;
}
```

Table 4-2 Macro of Reading UCID

5 Spare Register in RTC

Some applications need to protect certain critical data like electronic safety deposit box or specific ID cards. User can utilize a special pin to detect any voltage change happened on this pin. If someone wants to tamper the data inside, the chip will activate the protection mechanism.

Some NuMicro® Cortex®-M0/M4 series, such as NUC100, NUC200/220AN, NUC230/240AE, Nano100AN, Nano100BN, Nano102/112, NUC442/472, M451 and NUC505 series, are embedded with RTC peripheral. Also, some of them are provided with a special register to store important information. The spare register content is cleared when specified event on tamper pin is detected.

This special function related to data protection is “Spare Register”. The following table lists all the NuMicro® Cortex®-M series which support spare register function or not.

NuMicro® Cortex®-M	RTC Function	Spare Function	Spare Reg. Length
NUC505	√	√	32 bytes
NUC442/472	√	√	96 bytes
M451	√	√	80 bytes
Nano100	√	√	80 bytes
Nano102/112	√	√	80 bytes
NUC230/240AE	√	√	80 bytes
NUC200/220AN	√		
NUC100	√		
NUC122/123			
NUC029			
M058S			
M0518			
M051			
Mini51			

Table 5-1 RTC and Spare Function Support List

5.1 Tamper Pin

Taking M451LG6AE as an example, the RTC is equipped with 80 bytes spare registers to store important user information, and also has a snoop function to detect the transition of snooper pin.

User needs to enable SPRRWEN (RTC_SPRCTL[2]) before writing one of 20 spare registers (RTC_SPR0 ~ RTC_SPR19). User could read SPRRWRDY (RTC_SPRCTL[7]) to check if data has been written into registers or not. User could only access the spare registers again once SPRRWRDY is 1. Any access to spare registers is available if SPRRWRDY is 0.

The snoop detection function is used to detect the transition of TAMPER pin. When the transition condition defined in SNPTYPE1 (RTC_SPRCTL[3]) and SNPTYPE0 (RTC_SPRCTL[1]) is detected then 80 bytes spare registers (RTC_SPR0 ~ RTC_SPR19) content will be cleared by hardware automatically to prevent the security data from being disclosed

NuMicro® Cortex®-M	Tamper Pin Function	Pin Configuration
NUC442/472	√	PA.0/PA.1
M451	√	PF.2
Nano100	√	PC.13/PB.15
Nano102/112	√	PB.8
NUC230/240AE	**Note:** Because NUC230/240AE and NUC505 do not have Tamper pin so that its spare register could not be used for code protection.	
NUC505		

Table 5-2 Tamper Pin in Each Series

5.2 RTC Spare Register

Table 5-3 shows 20 separated spare registers for storing important information. Each register is 4 bytes long and a total of 80 bytes important data could be stored.

RTC Spare Register X (RTC_SPRx)

Register	Offset	R/W	Description	Reset Value
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
		.		
		.		
		.		
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000

31	30	29	28	27	26	25	24
SPARE							
23	22	21	20	19	18	17	16
SPARE							
15	14	13	12	11	10	9	8
SPARE							
7	6	5	4	3	2	1	0
SPARE							

Bits	Description	
[31:0]	**SPARE**	**Spare Register** This field is used to store back-up information defined by user. This field will be cleared by hardware automatically once a snooper pin event is detected. Before storing back-up information in to RTC_SPRx register, user should write 0xA965 to RTC_RWEN[15:0] to make sure register read/write enable bit REWNF (RTC_RWEN[16]) is enabled.

Table 5-3 RTC Spare Register Table and Description

Revision History

Date	Revision	Description
2015.07.28	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.