AN4246
Application note

Proprietary Code Read Out Protection on STM32L1 microcontrollers

Introduction

The protection of the intellectual property of embedded code has become a high importance issue concerning the microcontrollers. In order to provide this protection, STM32 microcontrollers have different means of protecting Flash code against copy and reverse engineering.

This application note describes the generic STM32 family Flash protection features. The focus is on the Proprietary Code Read Out Protection (PCROP) which is embedded in medium-density plus STM32L151xC, STM32L152xC, STM32L162xC and STM32L100xC microcontrollers.

Table 1 lists the microcontrollers concerned by this application note.

Table 1. Applicable products

Type	Applicable products
Microcontrollers	STM32L1 (STM32L151xC, STM32L152xC, STM32L162xC and STM32L100xC)

Contents

1 Flash code protection

The STM32 microcontroller family is provided with the following code protection features:

1. Global Read-out Protection (RDP)
2. Write protection
3. Proprietary Code Read Out Protection (PCROP)

These features are meant to protect the intellectual property of the embedded firmware code, which represents an increasing interest for complex embedded systems.

1.1 Global Read Out Protection (RDP)

The global Read Out Protection allows the embedded firmware code (preloaded in the Flash memory) to protect against reverse engineering, dumping using debug tools or other means of intrusive attack.

This protection is set by the user after the binary code is loaded to the embedded Flash memory.

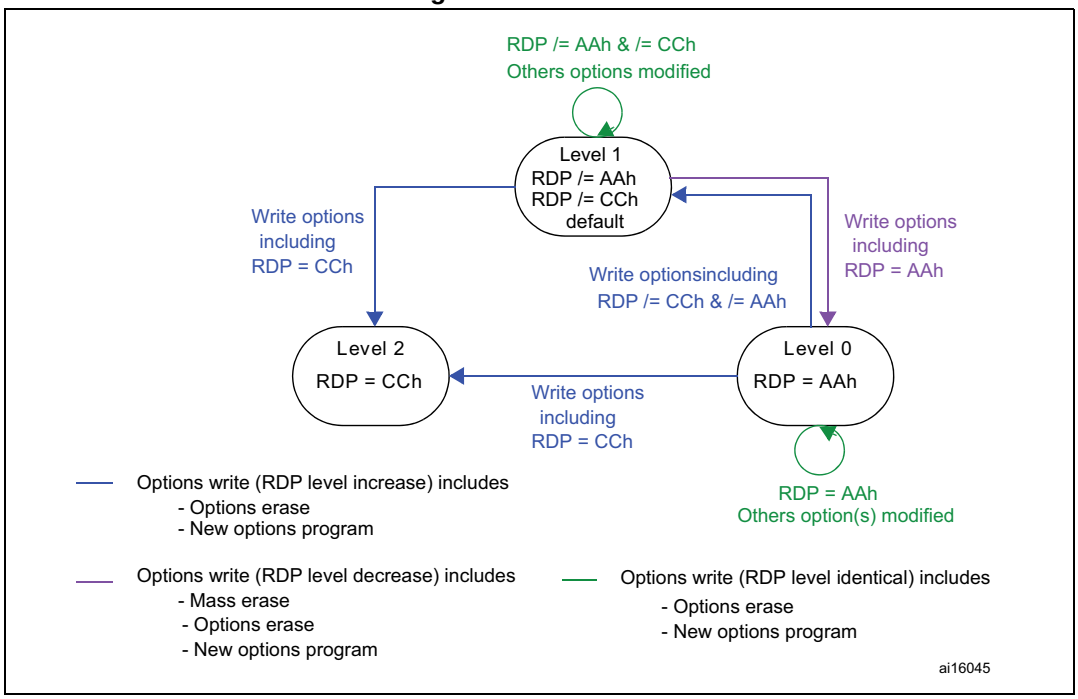
Table 2 describes the 3 user-defined protection levels.

Table 2. RDP Protection Level

Levels	Description
Level 0	No protection (default)
Level 1	Flash memory is protected against reading by debugging or code dumping by the RAM loaded code
Level 2	All debug features are disabled

Once the user code is loaded in the Flash memory of the product, it can be protected against code dumping.This is possible by activating either Level 1 or Level 2 protection, otherwise by RDP option byte programming, following the rules described in *Figure 1.*

Figure 1. RDP Levels



Both protection levels (1 and 2) have the same abilities to protect the Flash memory. Its content cannot be read by Serial Wire or JTAG Debug access, bootloader system software or by loading any other SW to the volatile RAM memory.

The main difference between the two protection levels is the volatile data (RAM content) protection which only exists on Level 2.

When RDP protection is set to Level 1, debug tools still can be connected and access all the volatile resources of the MCU (RAM and registers). These tools are used to check the part and/or system, by loading some test code to the RAM.

Also, Level 1 protection allows to recover a programmed part by erasing the entire Flash content. This is done by re-programming the RDP option byte from Level 1 to Level 0 (see *Figure 1*).

On the other hand, Level 2 protection is irreversible (fuse). Once the RDP is set to Level 2, the RDP option byte and all the other option bytes are frozen and can non longer be modified.

However, the user Flash content, with the exception of all the write-protected sectors (see *Section 1.2: Write Protection*), still can be updated under the control of the user code itself. An IAP (In Application Programming) bootloader code can be implemented in order to allow a firmware update of some sectors.

In order to ensure the protection of previously programmed user code, the bootloader protocol can be a user specified (implementing the relevant protection against attacks, dumping and/or malicious code update).

Note: Some examples of Secure Bootloader implementation using the embedded AES accelerator available on STM32 are described in application note AN4023 - STM32 secure firmware upgrade.

For additional details on Read Protection, refer to the microcontroller reference manuals.

1.2 Write Protection

The Write protection, applied by a Flash area (sector), protects the content of the specified sectors against code update or erase.

One option bit is used to activate the write protection for each Flash sector. When the Write protection is set for sector i (option bit `nWRPi` = 0), this sector cannot be erased or programmed.

Table 3 shows the sector Write protection depending on the RDP Level.

Table 3. Write Protection

Levels	Description
Level 0 or 1	The other option bytes still can be modified.(1)
Level 2	All the option bytes are definitively frozen.(2)

1. The sector Write protection is very important for safety functions. If they are programmed in the write protected sectors, these functions are fully protected against accidental erase or update.
2. A write protected sector cannot be erased or modified, either intentionally or not.

Note: Under these conditions, the integrity of the embedded firmware written in these sectors is guaranteed against any modification.

1.3 Proprietary Code Read Out Protection

The Proprietary Code Read Out Protection (PCROP) is an alternative protection which is applied also by sector, allowing the protection of specific code (intellectual property) against attacks.

The PCROP implements 2 main features on the microcontroller code protection and the code management.

Table 4 compares both PCROP features to the RDP protection method.

Table 4. Protection against attacks

Type of Protection	Comparison
External attacks	Similar to the protection offered by RDP (but which can be restricted to a specific Flash area)
Internal attacks (such as Trojan horse type)	Possible use of some “unsecured” third party code in an application, while still preserving the privacy of some parts of the code

This protection is based on an execute-only mechanism. The Flash code area can only be reached by the STM32 CPU (as an instruction code), while all other accesses (DMA, debug and CPU data read) are strictly prohibited.

While protecting the executable code against reading, a side effect generated by this execute-only mechanism makes the protected code itself (executed from this area) unable to access the associated data values stored in the same area (e.g. literal pool). In order to avoid the need of data accesses in this area (specially for literal pool accesses), a specific command line option must be chosen in the ARM/Keil compiler:
(`armcc --no_literal_pools --max_string_in_code = 0`).

This command line option translates the literal pool operations with alternative instructions. These instructions build the register values without any data read access. It is mainly needed for loading registers with variable addresses. As an alternative method is less efficient, this option translates these operations in a slightly less effective code. However, the loss of performance is limited (below 5%), which is acceptable for the protected parts of the code.

The PCROP sector is selected by using the same option bytes as the Write protection. As a result, these 2 options are exclusive each other. However, the sectors protected against reading (PCROP) are also protected against writing/erasing. Therefore, the PCROP may be considered as a superset of the sector write protection.

In order to activate the PCROP (change the function of the nWRP option bits), the SPRMOD option bit must be activated. This operation is irreversible.

Also in PCROP mode, a sector which was set to be read-protected cannot be reset to the unprotected state. As a result, new sectors may be added to the read protected area (when RDP is set to Level 0 or 1), but the protected ones cannot be unprotected, either erased or modified.

Depending on the RDP level, there is a possible workaround for recovering a protected chip. If the STM32 is in RDP Level 1 and the RDP option byte is set to Level 0, the user’s Flash area will be totally erased. This is the only case where the SPRMOD and nWRP bits may be reset and all the protected sectors may be unprotected.

However, as this operation is always associated to the global erase of the user Flash area, the code protection is not affected.

When the RDP is set to Level 2, all the option bytes are frozen and can no longer be modified. As a result, the protected sectors never can be erased or modified, so the protection becomes permanent.

2 Examples

2.1 Secure Firmware Update (SFU) bootloader protection

A secure firmware update bootloader (as described in AN4023) can be included. It allows programming a third party code in the STM32 Flash memory, without compromising the secure bootloader mechanism and/or keys.

2.2 Preloaded third-party IP code

The third-party code which contains critical intellectual property code can be preloaded (e.g. through a fast ROM procedure) in the STM32 Flash memory and protected against reading by activating the PCROP mechanism.

Then, the STM32 microcontrollers including the protected code can be used/programmed by the end user, without affecting the protected code.

3 Conclusion

STM32 microcontrollers are provided with various Flash protection mechanisms to fulfill the different needs of the intellectual property protection. These range from a single user global code protection to a finer grain code protection where multiple IP firmware can coexist in the STM32 microcontroller memory. This solution allows the application to operate in potentially unsafe environments without compromising code protection or integrity.

4 Reference documents

Programming manual (PM0062), STMicroelectronics

Reference manual (RM0038), STMicroelectronics

5 Revision history

Table 5. Document revision history

Date	Revision	Changes
03-Apr-2013	1	Initial release.