



## STM32F10xxx Flash memory microcontrollers

### Introduction

This programming manual describes how to program the Flash memory of STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx microcontrollers. For convenience, these will be referred to as STM32F10xxx in the rest of this document unless otherwise specified.

The STM32F10xxx embedded Flash memory can be programmed using in-circuit programming or in-application programming.

The **in-circuit programming (ICP)** method is used to update the entire contents of the Flash memory, using the JTAG, SWD protocol or the boot loader to load the user application into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

In contrast to the ICP method, **in-application programming (IAP)** can use any communication interface supported by the microcontroller (I/Os, USB, CAN, UART, I<sup>2</sup>C, SPI, etc.) to download programming data into memory. IAP allows the user to re-program the Flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the Flash memory using ICP.

The Flash interface implements instruction access and data access based on the AHB protocol. It implements a prefetch buffer that speeds up CPU code execution. It also implements the logic necessary to carry out Flash memory operations (Program/Erase). Program/Erase operations can be performed over the whole product voltage range. Read/Write protections and option bytes are also implemented.

[Table 1](#) lists the microcontrollers and evaluation tool concerned by this programming manual.

**Table 1. Applicable products**

Type	Applicable products
Microcontrollers	STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx microcontrollers
Evaluation tool	STM3210C-EVAL evaluation board

# Contents

- 1 Overview ..... 6**
  - 1.1 Features ..... 6
  - 1.2 Flash module organization ..... 6
  
- 2 Reading/programming the embedded Flash memory ..... 11**
  - 2.1 Introduction ..... 11
  - 2.2 Read operation ..... 11
    - 2.2.1 Instruction fetch ..... 11
    - 2.2.2 D-Code interface ..... 12
    - 2.2.3 Flash access controller ..... 12
  - 2.3 Flash program and erase controller (FPEC) ..... 12
    - 2.3.1 Key values ..... 12
    - 2.3.2 Unlocking the Flash memory ..... 13
    - 2.3.3 Main Flash memory programming ..... 13
    - 2.3.4 Flash memory erase ..... 14
    - 2.3.5 Option byte programming ..... 16
  - 2.4 Protections ..... 17
    - 2.4.1 Read protection ..... 17
    - 2.4.2 Write protection ..... 18
    - 2.4.3 Option byte block write protection ..... 19
  - 2.5 Option byte description ..... 19
  
- 3 Register descriptions ..... 23**
  - 3.1 Flash access control register (FLASH\_ACR) ..... 23
  - 3.2 FPEC key register (FLASH\_KEYR) ..... 24
  - 3.3 Flash OPTKEY register (FLASH\_OPTKEYR) ..... 24
  - 3.4 Flash status register (FLASH\_SR) ..... 25
  - 3.5 Flash control register (FLASH\_CR) ..... 26
  - 3.6 Flash address register (FLASH\_AR) ..... 27
  - 3.7 Option byte register (FLASH\_OBR) ..... 27
  - 3.8 Write protection register (FLASH\_WRPR) ..... 28
  - 3.9 Flash register map ..... 29
  
- 4 Revision history ..... 30**

## List of tables

Table 1.	Applicable products . . . . .	1
Table 2.	Flash module organization (low-density devices) . . . . .	7
Table 3.	Flash module organization (medium-density devices) . . . . .	7
Table 4.	Flash module organization (high-density devices) . . . . .	8
Table 5.	Flash module organization (connectivity line devices) . . . . .	9
Table 6.	Flash memory protection status . . . . .	18
Table 7.	Option byte format . . . . .	19
Table 8.	Option byte organization . . . . .	20
Table 9.	Description of the option bytes . . . . .	20
Table 10.	Abbreviations . . . . .	23
Table 11.	Flash interface - register map and reset values . . . . .	29
Table 12.	Document revision history . . . . .	30

## List of figures

Figure 1.	Programming procedure . . . . .	13
Figure 2.	Flash memory Page Erase procedure . . . . .	15
Figure 3.	Flash memory Mass Erase procedure . . . . .	16

## Glossary

This section gives a brief definition of acronyms and abbreviations used in this document:

- **Low-density devices** are STM32F101xx, STM32F102xx and STM32F103xx microcontrollers where the Flash memory density ranges between 16 and 32 Kbytes.
- **Medium-density devices** are STM32F101xx, STM32F102xx and STM32F103xx microcontrollers where the Flash memory density ranges between 64 and 128 Kbytes.
- **High-density devices** are STM32F101xx and STM32F103xx microcontrollers where the Flash memory density ranges between 256 and 512 Kbytes.
- **Connectivity line devices** are STM32F105xx and STM32F107xx microcontrollers.
- The Cortex-M3 core integrates two debug ports:
  - JTAG debug port (JTAG-DP) provides a 5-pin standard interface based on the Joint Test Action Group (JTAG) protocol.
  - SWD debug port (SWD-DP) provides a 2-pin (clock and data) interface based on the Serial Wire Debug (SWD) protocol.  
For both the JTAG and SWD protocols please refer to the *Cortex M3 Technical Reference Manual*
- Word: data/instruction of 32-bit length
- Half word: data/instruction of 16-bit length
- Byte: data of 8-bit length
- FPEC (Flash memory program/erase controller): write operations to the main memory and the information block are managed by an embedded Flash program/erase controller (FPEC).
- IAP (in-application programming): IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
- ICP (in-circuit programming): ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the boot loader while the device is mounted on the user application board.
- I-Code: this bus connects the Instruction bus of the Cortex-M3 core to the Flash instruction interface. Prefetch is performed on this bus.
- D-Code: this bus connects the D-Code bus (literal load and debug access) of the Cortex-M3 to the Flash Data Interface.
- Option bytes: product configuration bits stored in the Flash memory
- OBL: option byte loader.
- AHB: advanced high-performance bus.

# 1 Overview

## 1.1 Features

- up to 512 Kbytes of Flash memory
- Memory organization:
  - Main memory block:
    - 4 Kbits × 64 bits for low-density devices
    - 16 Kbits × 64 bits for medium-density devices
    - 64 Kbits × 64 bits for high-density devices
    - 32 Kbits × 64 bits for connectivity line devices
  - Information block:
    - 2306 × 64 bits for connectivity line devices
    - 258 × 64 bits for other devices

Flash memory interface (FLITF) features:

- Read interface with prefetch buffer (2 × 64-bit words)
- Option byte Loader
- Flash Program / Erase operation
- Read / Write protection
- Low-power mode

## 1.2 Flash module organization

The memory organization is based on a main memory block containing 32 pages of 1 Kbyte (for low-density devices), 128 pages of 1 Kbyte (for medium-density devices), 128 pages of 2 Kbyte (for connectivity line devices) or 256 pages of 2 Kbyte (for high-density devices), and an information block as shown in [Table 3](#) and [Table 4](#).

**Table 2. Flash module organization (low-density devices)**

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	⋮	⋮	⋮
	Page 31	0x0800 7C00 - 0x0800 7FFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

**Table 3. Flash module organization (medium-density devices)**

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	⋮	⋮	⋮
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16

**Table 3. Flash module organization (medium-density devices) (continued)**

Block	Name	Base addresses	Size (bytes)
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

**Table 4. Flash module organization (high-density devices)**

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 07FF	2 Kbytes
	Page 1	0x0800 0800 - 0x0800 0FFF	2 Kbytes
	Page 2	0x0800 1000 - 0x0800 17FF	2 Kbytes
	Page 3	0x0800 1800 - 0x0800 1FFF	2 Kbytes
	⋮	⋮	⋮
	Page 255	0x0807 F800 - 0x0807 FFFF	2 Kbytes
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4



**Table 5. Flash module organization (connectivity line devices)**

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 07FF	2 Kbytes
	Page 1	0x0800 0800 - 0x0800 0FFF	2 Kbytes
	Page 2	0x0800 1000 - 0x0800 17FF	2 Kbytes
	Page 3	0x0800 1800 - 0x0800 1FFF	2 Kbytes
	⋮	⋮	⋮
	Page 127	0x0803 F800 - 0x0803 FFFF	2 Kbytes
Information block	System memory	0x1FFF B000 - 0x1FFF F7FF	18 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

The Flash memory is organized as 32-bit wide memory cells that can be used for storing both code and data constants. The Flash module is located at a specific base address in the memory map of each STM32F10xxx microcontroller type. For the base address, please refer to the related *STM32F10xxx reference manual*.

The information block is divided into two parts:

- System memory is used to boot the device in System memory boot mode. The area is reserved for use by STMicroelectronics and contains the boot loader which is used to reprogram the Flash memory using the USART1 serial interface. It is programmed by ST when the device is manufactured, and protected against spurious write/erase operations. For further details please refer to AN2606.

In connectivity line devices the boot loader can be activated through one of the following interfaces: USART1, USART2 (remapped), CAN2 (remapped) or USB OTG FS in Device mode (DFU: device firmware upgrade). The USART peripheral operates with the internal 8 MHz oscillator (HSI). The CAN and USB OTG FS, however, can only function if an external 8 MHz, 14.7456 MHz or 25 MHz clock (HSE) is present. For further details, please refer to AN2662 (“STM32F105xx and STM32F107xx system memory boot mode”) available from [www.st.com](http://www.st.com).

- Option bytes

Write operations to the main memory block and the option bytes are managed by an embedded Flash Program/Erase Controller (FPEC). The high voltage needed for Program/Erase operations is internally generated.

The main Flash memory can be protected against different types of unwanted access (read/write/erase). There are two types of protection:

- Page Write Protection
- Read Protection

Refer to [Section 2.4](#) for more details.

During a write operation to the Flash memory, any attempt to read the Flash memory will stall the bus. The read operation will proceed correctly once the write operation has completed. This means that code or data fetches cannot be made while a write/erase operation is ongoing.

For write and erase operations on the Flash memory (write/erase), the internal RC oscillator (HSI) must be ON.

The Flash memory can be programmed and erased using in-circuit programming and in-application programming.

*Note: In the low-power modes, all Flash memory accesses are aborted. Refer to the STM32F10xxx reference manual for further information.*

## 2 Reading/programming the embedded Flash memory

### 2.1 Introduction

This section describes how to read from or program to the STM32F10xxx embedded Flash memory.

### 2.2 Read operation

The embedded Flash module can be addressed directly, as a common memory space. Any data read operation accesses the content of the Flash module through dedicated read senses and provides the requested data.

The read interface consists of a read controller on one side to access the Flash memory and an AHB interface on the other side to interface with the CPU. The main task of the read interface is to generate the control signals to read from the Flash memory and to prefetch the blocks required by the CPU. The prefetch block is only used for instruction fetches over the I-Code bus. The Literal pool is accessed over the D-Code bus. Since these two buses have the same Flash memory as target, D-code bus accesses have priority over prefetch accesses.

#### 2.2.1 Instruction fetch

The Cortex-M3 fetches the instruction over the I-Code bus and the literal pool (constant/data) over the D-code bus. The prefetch block aims at increasing the efficiency of I-Code bus accesses.

##### Prefetch buffer

The prefetch buffer is 2 blocks wide where each block consists of 8 bytes. The prefetch blocks are direct-mapped. A block can be completely replaced on a single read to the Flash memory as the size of the block matches the bandwidth of the Flash memory.

The implementation of this prefetch buffer makes a faster CPU execution possible as the CPU fetches one word at a time with the next word readily available in the prefetch buffer. This implies that the acceleration ratio will be of the order of 2 assuming that the code is aligned at a 64-bit boundary for the jumps.

##### Prefetch controller

The prefetch controller decides to access the Flash memory depending on the available space in the prefetch buffer. The Controller initiates a read request when there is at least one block free in the prefetch buffer.

After reset, the state of the prefetch buffer is on.

The prefetch buffer should be switched on/off only when SYSCLK is lower than 24 MHz and no prescaler is applied on the AHB clock (SYSCLK must be equal to HCLK). The prefetch buffer is usually switched on/off during the initialization routine, while the microcontroller is running on the internal 8 MHz RC (HSI) oscillator.

*Note:* The prefetch buffer must be kept on (`FLASH_ACR[4]='1'`) when using a prescaler different from 1 on the AHB clock.

In case of non-availability of a high frequency clock in the system, Flash memory accesses can be made on a half cycle of HCLK (AHB clock), the frequency of HCLK permitting (half-cycle access can only be used with a low-frequency clock of less than 8 MHz that can be obtained with the use of HSI or HSE but not of PLL). This mode can be chosen by setting a control bit in the Flash access control register.

*Note:* Half-cycle access cannot be used when there is a prescaler different from 1 on the AHB clock.

### Access time tuner

In order to maintain the control signals to read the Flash memory, the ratio of the prefetch controller clock period to the access time of the Flash memory has to be programmed in the Flash access control register. This value gives the number of cycles needed to maintain the control signals of the Flash memory and correctly read the required data. After reset, the value is zero and only one cycle is required to access the Flash memory.

## 2.2.2 D-Code interface

The D-Code interface consists of a simple AHB interface on the CPU side and a request generator to the Arbiter of the Flash access controller. D-code accesses have priority over prefetch accesses. This interface uses the Access Time Tuner block of the prefetch buffer.

## 2.2.3 Flash access controller

Mainly, this block is a simple arbiter between the read requests of the prefetch/I-code and D-Code interfaces.

D-Code interface requests have priority over I-Code requests.

## 2.3 Flash program and erase controller (FPEC)

The FPEC block handles the program and erase operations of the Flash memory. The FPEC consists of seven 32-bit registers.

- FPEC key register (FLASH\_KEYR)
- Option byte key register (FLASH\_OPTKEYR)
- Flash control register (FLASH\_CR)
- Flash status register (FLASH\_SR)
- Flash address register (FLASH\_AR)
- Option byte register (FLASH\_OBR)
- Write protection register (FLASH\_WRP)

An ongoing Flash memory operation will not block the CPU as long as the CPU does not access the Flash memory.

### 2.3.1 Key values

The key values are as follows:

- RDPRT key = 0x00A5
- KEY1 = 0x45670123
- KEY2 = 0xCDEF89AB

### 2.3.2 Unlocking the Flash memory

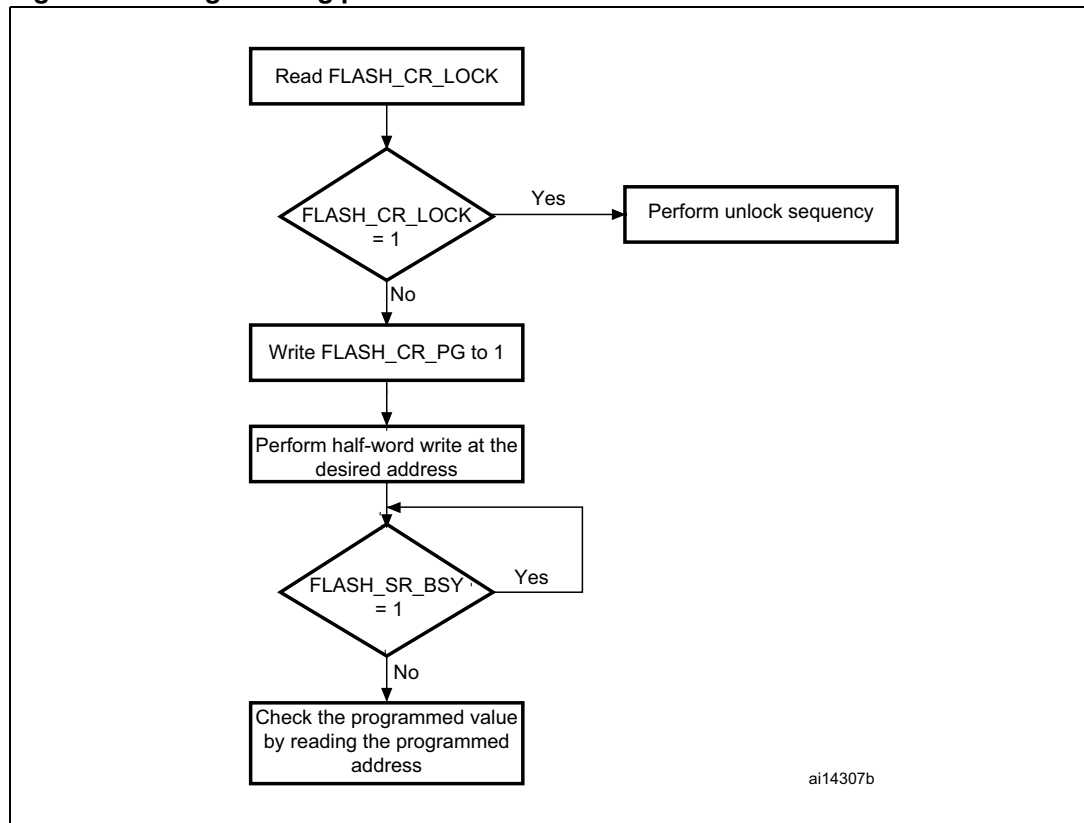
After reset, the FPEC block is protected. The FLASH\_CR register is not accessible in write mode. An unlocking sequence should be written to the FLASH\_KEYR register to open up the FPEC block. This sequence consists of two write cycles, where two key values (KEY1 and KEY2) are written to the FLASH\_KEYR address (refer to [Section 2.3.1](#) for key values). Any wrong sequence locks up the FPEC block and FLASH\_CR register until the next reset.

Also a bus error is returned on a wrong key sequence. This is done after the first write cycle if KEY1 does not match, or during the second write cycle if KEY1 has been correctly written but KEY2 does not match. The FPEC block and FLASH\_CR register can be locked by the user's software by writing the LOCK bit of the FLASH\_CR register to 1. In this case, the FPEC can be unlocked by writing the correct sequence of keys into FLASH\_KEYR.

### 2.3.3 Main Flash memory programming

The main Flash memory can be programmed 16 bits at a time. The program operation is started when the CPU writes a half-word into a main Flash memory address with the PG bit of the FLASH\_CR register set. Any attempt to write data that are not half-word long will result in a bus error response from the FPEC. If a read/write operation is initiated during programming, (BSY bit set), the CPU stalls until the ongoing main Flash memory programming is over.

Figure 1. Programming procedure



## Standard programming

In this mode the CPU programs the main Flash memory by performing standard half-word write operations. The PG bit in the FLASH\_CR register must be set. FPEC preliminarily reads the value at the addressed main Flash memory location and checks that it has been erased. If not, the program operation is skipped and a warning is issued by the PGERR bit in FLASH\_SR register (the only exception to this is when 0x0000 is programmed. In this case, the location is correctly programmed to 0x0000 and the PGERR bit is not set). If the addressed main Flash memory location is write-protected by the FLASH\_WRP register, the program operation is skipped and a warning is issued by the WRPRERR bit in the FLASH\_SR register. The end of the program operation is indicated by the EOP bit in the FLASH\_SR register.

The main Flash memory programming sequence in standard mode is as follows:

- Check that no main Flash memory operation is ongoing by checking the BSY bit in the FLASH\_SR register.
- Set the PG bit in the FLASH\_CR register.
- Perform the data write (half-word) at the desired address.
- Wait for the BSY bit to be reset.
- Read the programmed value and verify.

*Note:* The registers are not accessible in write mode when the BSY bit of the FLASH\_SR register is set.

### 2.3.4 Flash memory erase

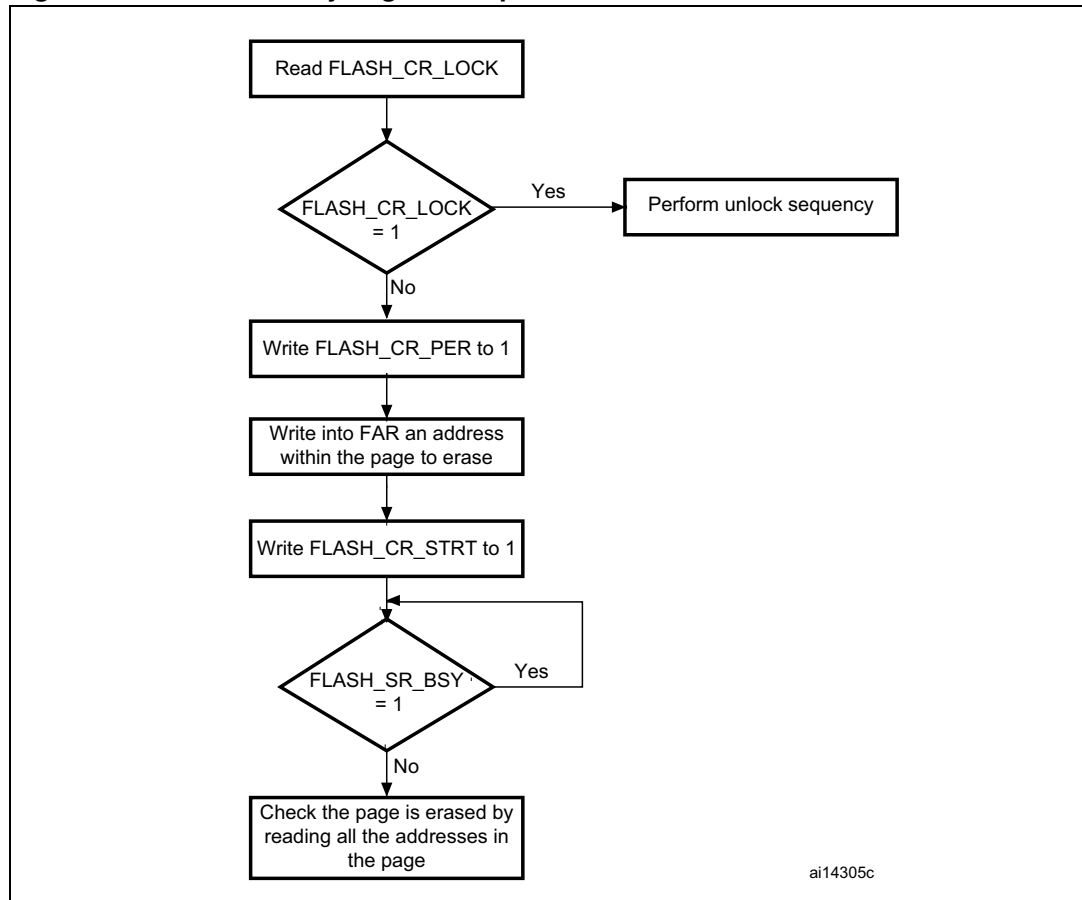
The Flash memory can be erased page by page or completely (Mass Erase).

#### Page Erase

A page of the Flash memory can be erased using the Page Erase feature of the FPEC. To erase a page, the procedure below should be followed:

- Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH\_CR register
- Set the PER bit in the FLASH\_CR register
- Program the FLASH\_AR register to select a page to erase
- Set the STRT bit in the FLASH\_CR register
- Wait for the BSY bit to be reset
- Read the erased page and verify

Figure 2. Flash memory Page Erase procedure

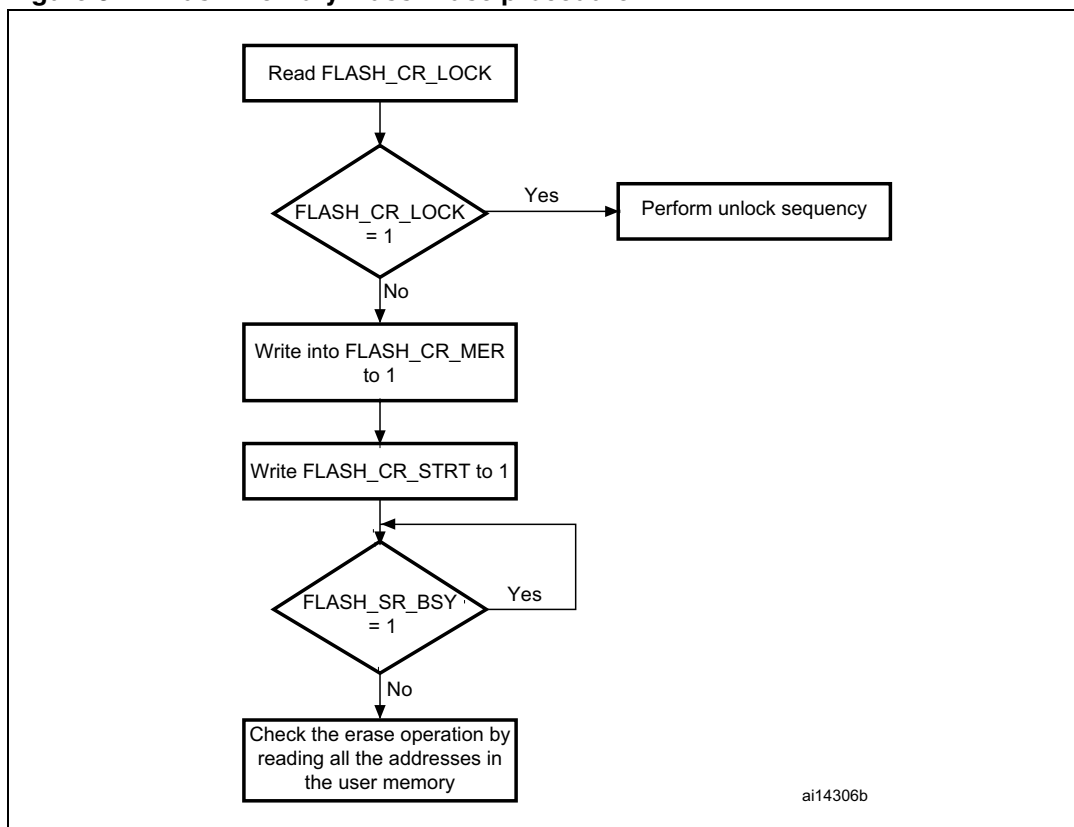


### Mass Erase

The Mass Erase command can be used to completely erase the user pages of the Flash memory. The information block is unaffected by this procedure. The following sequence is recommended:

- Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH\_SR register
- Set the MER bit in the FLASH\_CR register
- Set the STRT bit in the FLASH\_CR register
- Wait for the BSY bit to be reset
- Read all the pages and verify

Figure 3. Flash memory Mass Erase procedure



### 2.3.5 Option byte programming

The option bytes are programmed differently from normal user addresses. The number of option bytes is limited to 8 (4 for write protection, 1 for read protection, 1 for configuration and 2 for user data storage). After unlocking the FPEC, the user has to authorize the programming of the option bytes by writing the same set of KEYS (KEY1 and KEY2) to the FLASH\_OPTKEYR register to set the OPTWRE bit in the FLASH\_CR register (refer to [Section 2.3.1](#) for key values). Then the user has to set the OPTPG bit in the FLASH\_CR register and perform a half-word write operation at the desired Flash address.

FPEC preliminarily reads the value of the addressed option byte and checks that it has been erased. If not, the program operation is skipped and a warning is issued by the PGERR bit in the FLASH\_SR register. The end of the program operation is indicated by the EOP bit in the FLASH\_SR register.

The FPEC takes the LSB and automatically computes the MSB (which is the complement of the LSB) and starts the programming operation. This guarantees that the option byte and its complement are always correct.



The sequence is as follows:

- Check that no Flash memory operation is ongoing by checking the BSY bit in the FLASH\_SR register.
- Unlock the OPTWRE bit in the FLASH\_CR register.
- Set the OPTPG bit in the FLASH\_CR register
- Write the data (half-word) to the desired address
- Wait for the BSY bit to be reset.
- Read the programmed value and verify.

When the Flash memory read protection option is changed from protected to unprotected, a Mass Erase of the main Flash memory is performed before reprogramming the read protection option. If the user wants to change an option other than the read protection option, then the mass erase is not performed. The erased state of the read protection option byte protects the Flash memory.

### Erase procedure

The option byte erase sequence (OPTERASE) is as follows:

- Check that no Flash memory operation is ongoing by reading the BSY bit in the FLASH\_SR register
- Unlock the OPTWRE bit in the FLASH\_CR register
- Set the OPTER bit in the FLASH\_CR register
- Set the STRT bit in the FLASH\_CR register
- Wait for BSY to reset
- Read the erased option bytes and verify

## 2.4 Protections

The user area of the Flash memory can be protected against read by untrusted code. The pages of the Flash memory can also be protected against unwanted write due to loss of program counter contexts. The write-protection granularity is then of:

- four pages for low- and medium-density devices
- two pages for high-density and connectivity line devices.

### 2.4.1 Read protection

The read protection is activated by setting the RDP option byte and then, by applying a system reset to reload the new RDP option byte.

*Note: If the read protection is set while the debugger is still connected through JTAG/SWD, apply a POR (power-on reset) instead of a system reset (without debugger connection).*

Once the protection byte has been programmed:

- Main Flash memory read access is not allowed except for the user code (when booting from main Flash memory itself with the debug mode not active).
- Pages 0-3 (for low- and medium-density devices), or pages 0-1 (for high-density and connectivity line devices) are automatically write-protected. The rest of the memory can be programmed by the code executed from the main Flash memory (for IAP, constant storage, etc.), but it is protected against write/erase (but not against mass erase) in debug mode or when booting from the embedded SRAM.

- All features linked to loading code into and executing code from the embedded SRAM are still active (JTAG/SWD and boot from embedded SRAM) and this can be used to disable the read protection. When the read protection option byte is altered to a memory-unprotect value, a mass erase is performed.
- When booting from the embedded SRAM, Flash memory accesses through the code and through data read using DMA1 and DMA2 are not allowed.
- Flash memory access through data read using JTAG, SWV (serial wire viewer), SWD (serial wire debug), ETM and boundary scan are not allowed.

The Flash memory is protected when the RDP option byte and its complement contain the pair of values shown in [Table 6](#).

**Table 6. Flash memory protection status**

RDP byte value	RDP complement value	Read protection status
0xFF	0xFF	Protected
RDPRT	Complement of RDP byte	Not protected
Any value	Not the complement value of RDP	Protected

*Note:* Erasing the option byte block will not trigger a mass erase as the erased value (0xFF) corresponds to a protected value.

### Unprotection

To disable the read protection from the embedded SRAM:

- Erase the entire option byte area. As a result, the read protection code (RDP) will be 0xFF. At this stage the read protection is still enabled.
- Program the correct RDP code 0x00A5 to unprotect the memory. This operation first forces a Mass Erase of the main Flash memory.
- Reset the device (POR Reset) to reload the option bytes (and the new RDP code) and, to disable the read protection.

*Note:* The read protection can be disabled using the boot loader (in this case only a System Reset is necessary to reload the option bytes). For more details refer to AN2606.

### 2.4.2 Write protection

In high-density and connectivity line devices, from page 0 to page 61, write protection is implemented with a granularity of two pages at a time. The remaining memory block (from page 62 to page 255 in high-density devices, and from page 62 to page 127 in connectivity line devices) is write-protected at once.

In low- and medium-density devices, write protection is implemented with a granularity of four pages at a time.

If a program or an erase operation is performed on a protected page, the Flash memory returns a protection error flag on the Flash memory Status Register (FLASH\_SR).

The write protection is activated by configuring the WRP[3:0] option bytes, and then by applying a system reset to reload the new WRPx option bytes.

## Unprotection

To disable the write protection, two application cases are provided:

- Case 1: Read protection disabled after the write unprotection:
  - Erase the entire option byte area by using the OPTER bit in the Flash memory control register (FLASH\_CR)
  - Program the correct RDP code 0x00A5 to unprotect the memory. This operation first forces a Mass Erase of the main Flash memory.
  - Reset the device (system reset) to reload the option bytes (and the new WRP[3:0] bytes), and to disable the write protection
- Case 2: Read protection maintained active after the write unprotection, useful for in-application programming with a user boot loader:
  - Erase the entire option byte area by using the OPTER bit in the Flash memory control register (FLASH\_CR)
  - Reset the device (system reset) to reload the option bytes (and the new WRP[3:0] bytes), and to disable the write protection.

### 2.4.3 Option byte block write protection

The option bytes are always read-accessible and write-protected by default. To gain write access (Program/Erase) to the option bytes, a sequence of keys (same as for lock) has to be written into the OPTKEYR. A correct sequence of keys gives write access to the option bytes and this is indicated by OPTWRE in the FLASH\_CR register being set. Write access can be disabled by resetting the bit through software.

## 2.5 Option byte description

There are eight option bytes. They are configured by the end user depending on the application requirements. As a configuration example, the watchdog may be selected in hardware or software mode.

A 32-bit word is split up as follows in the option bytes.

**Table 7. Option byte format**

31-24	23-16	15 -8	7-0
complemented option byte1	Option byte 1	complemented option byte0	Option byte 0

The organization of these bytes inside the information block is as shown in [Table 8](#).

The option bytes can be read from the memory locations listed in [Table 8](#) or from the Option byte register (FLASH\_OBR).

*Note:* The new programmed option bytes (user, read/write protection) are loaded after a system reset.

**Table 8. Option byte organization**

Address	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nData1	Data1	nData0	Data0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF F80C	nWRP3	WRP3	nWRP2	WRP2

**Table 9. Description of the option bytes**

Flash memory address	Option bytes
0x1FFF F800	<p>Bits [31:24] <b>nUSER</b></p> <p>Bits [23:16] <b>USER</b>: User option byte (stored in FLASH_OBR[9:2])                      This byte is used to configure the following features:</p> <ul style="list-style-type: none"> <li>- Select the watchdog event: Hardware or software.</li> <li>- Reset event when entering Stop mode.</li> <li>- Reset event when entering Standby mode.</li> </ul> <p><i>Note: Only bits [16:18] are used, bits [23:19]: 0x1F are not used.</i></p> <p>Bit 18: <b>nRST_STDBY</b>                      0: Reset generated when entering Standby mode.                      1: No reset generated.</p> <p>Bit 17: <b>nRST_STOP</b>                      0: Reset generated when entering Stop mode                      1: No reset generated</p> <p>Bit 16: <b>WDG_SW</b>                      0: Hardware watchdog                      1: Software watchdog</p> <p>Bits [15:8]: <b>nRDP</b></p> <p>Bits [7:0]: <b>RDP</b>: Read protection option byte                      The read protection helps the user protect the software code stored in Flash memory. It is activated by setting the RDP option byte.                      When this option byte is programmed to a correct value (RDPRT key = 0x00A5), read access to the Flash memory is allowed.                      (The result of RDP level enabled/disabled is stored in FLASH_OBR[1].)</p>
0x1FFF F804	<p><b>Datax</b>: Two bytes for user data storage.                      These addresses can be programmed using the option byte programming procedure.</p> <p>Bits [31:24]: <b>nData1</b></p> <p>Bits [23:16]: <b>Data1</b> (stored in FLASH_OBR[25:18])</p> <p>Bits [15:8]: <b>nData0</b></p> <p>Bits [7:0]: <b>Data0</b> (stored in FLASH_OBR[17:10])</p>
0x1FFF F808	<p><b>WRPx</b>: Flash memory write protection option bytes</p> <p>Bits [31:24]: <b>nWRP1</b></p> <p>Bits [23:16]: <b>WRP1</b> (stored in FLASH_WRPR[15:8])</p> <p>Bits [15:8]: <b>nWRP0</b></p> <p>Bits [7:0]: <b>WRP0</b> (stored in FLASH_WRPR[7:0])</p>

Table 9. Description of the option bytes (continued)

Flash memory address	Option bytes
0x1FFF F80C	<p><b>WRP<sub>x</sub></b>: Flash memory write protection option bytes</p> <p>Bits [31:24]: <b>nWRP3</b></p> <p>Bits [23:16]: <b>WRP3</b> (stored in FLASH_WRP[31:24])</p> <p>Bits [15:8]: <b>nWRP2</b></p> <p>Bits [7:0]: <b>WRP2</b> (stored in FLASH_WRP[23:16])</p> <ul style="list-style-type: none"> <li>● For low-density devices, one bit of the user option bytes WRP<sub>x</sub> is used to protect 4 pages of 1 Kbyte in main memory block. <ul style="list-style-type: none"> <li>– 0: Write protection active</li> <li>– 1: Write protection not active</li> </ul> <p>In total, one user option byte is used to protect the 32-Kbyte main Flash memory.</p> <p>WRP0: Write-protects pages 0 to 31</p> </li> <li>● For medium-density devices, one bit of the user option bytes WRP<sub>x</sub> is used to protect 4 pages of 1 Kbyte in main memory block. <ul style="list-style-type: none"> <li>– 0: Write protection active</li> <li>– 1: Write protection not active</li> </ul> <p>In total, four user option bytes are used to protect the 128-Kbyte main Flash memory.</p> <p>WRP0: Write-protects pages 0 to 31</p> <p>WRP1: Write-protects pages 32 to 63</p> <p>WRP2: Write-protects pages 64 to 95</p> <p>WRP3: Write-protects pages 96 to 127</p> </li> <li>● For high-density devices, one bit of the user option bytes WRP<sub>x</sub> is used to protect 2 pages of 2 Kbytes in main memory block. However, the bit 7 of WRP3 write protects pages 62 to 255. <ul style="list-style-type: none"> <li>– 0: Write protection active</li> <li>– 1: Write protection not active</li> </ul> <p>In total, four user option bytes are used to protect the 512-Kbyte main Flash memory.</p> <p>WRP0: Write-protects pages 0 to 15.</p> <p>WRP1: Write-protects pages 16 to 31.</p> <p>WRP2: Write-protects pages 32 to 47.</p> <p>WRP3: bits 0-6 write-protect pages 48 to 61</p> <p>bit 7 write-protects pages 62 to 255.</p> </li> <li>● For connectivity line devices, one bit of the user option bytes WRP<sub>x</sub> is used to protect 2 pages of 2 Kbytes in main memory block. However, the bit 7 of WRP3 write-protects pages 62 to 127. <ul style="list-style-type: none"> <li>– 0: Write protection active</li> <li>– 1: Write protection not active</li> </ul> <p>In total, four user option bytes are used to protect the 256-Kbyte main Flash memory.</p> <p>WRP0: Write-protects pages 0 to 15.</p> <p>WRP1: Write-protects pages 16 to 31.</p> <p>WRP2: Write-protects pages 32 to 47.</p> <p>WRP3: bits 0-6 write-protect pages 48 to 61</p> <p>bit 7 write-protects pages 62 to 127.</p> </li> </ul>

On every system reset, the option byte loader (OBL) reads the information block and stores the data into the Option byte register (FLASH\_OBR) and the Write protection register (FLASH\_WRPR). Each option byte also has its complement in the information block. During option loading, by verifying the option bit and its complement, it is possible to check that the loading has correctly taken place. If this is not the case, an option byte error (OPTERR) is generated. When a comparison error occurs the corresponding option byte is forced to 0xFF. The comparator is disabled when the option byte and its complement are both equal to 0xFF (Electrical Erase state).

All option bytes (but not their complements) are available to configure the product. The option registers are accessible in read mode by the CPU. See [Section 3: Register descriptions](#) for more details.

### 3 Register descriptions

In this section, the following abbreviations are used:

**Table 10. Abbreviations**

Abbreviation	Meaning
read/write (rw)	Software can read from and write to these bits.
read-only (r)	Software can only read these bits.
write-only (w)	Software can only write to this bit. Reading the bit returns the reset value.
read/clear (rc_w0)	Software can read as well as clear this bit by writing '0'. Writing '1' has no effect on the bit value.
read/set (rs)	Software can read as well as set this bit. Writing '0' has no effect on the bit value.
Reserved (Res.)	Reserved bit, must be kept at reset value.

*Note:* The Flash memory registers have to be accessed by 32-bit words (half-word and byte accesses are not allowed).

#### 3.1 Flash access control register (FLASH\_ACR)

Address offset: 0x00

Reset value: 0x0000 0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										PRFT BS	PRFT BE	HLF CYA	LATENCY		
										r	rw	rw	rw	rw	rw

Bits 31:6 Reserved, must be kept cleared.

Bit 5 **PRFTBS**: Prefetch buffer status

This bit provides the status of the prefetch buffer.

0: Prefetch buffer is disabled

1: Prefetch buffer is enabled

Bit 4 **PRFTBE**: Prefetch buffer enable

0: Prefetch is disabled

1: Prefetch is enabled

Bit 3 **HLFCYA**: Flash half cycle access enable

0: Half cycle is disabled

1: Half cycle is enabled

Bits 2:0 **LATENCY**: Latency

These bits represent the ratio of the SYSCLK (system clock) period to the Flash access time.

000 Zero wait state, if  $0 < \text{SYSCLK} \leq 24 \text{ MHz}$

001 One wait state, if  $24 \text{ MHz} < \text{SYSCLK} \leq 48 \text{ MHz}$

010 Two wait states, if  $48 \text{ MHz} < \text{SYSCLK} \leq 72 \text{ MHz}$

### 3.2 FPEC key register (FLASH\_KEYR)

Address offset: 0x04

Reset value: xxxx xxxx

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

*Note:* These bits are all write-only and will return a 0 when read.

Bits 31:0 **FKEYR**: FPEC key

These bits represent the keys to unlock the FPEC.

### 3.3 Flash OPTKEY register (FLASH\_OPTKEYR)

Address offset: 0x08

Reset value: xxxx xxxx

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

*Note:* These bits are all write-only and will return a 0 when read.

Bits 31:0 **OPTKEYR**: Option byte key

These bits represent the keys to unlock the OPTWRE.



### 3.4 Flash status register (FLASH\_SR)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EOP	WRPRT ERR	Res.	PG ERR	Res.	BSY
										rw	rw		rw		r

Bits 31:6 Reserved, must be kept cleared.

Bit 5 **EOP**: End of operation

Set by hardware when a Flash operation (programming / erase) is completed. Reset by writing a 1

*Note: EOP is asserted at the end of each successful program or erase operation*

Bit 4 **WRPRTERR**: Write protection error

Set by hardware when programming a write-protected address of the Flash memory. Reset by writing 1.

Bit 3 Reserved, must be kept cleared.

Bit 2 **PGERR**: Programming error

Set by hardware when an address to be programmed contains a value different from '0xFFFF' before programming.

Reset by writing 1.

*Note: The STRT bit in the FLASH\_CR register should be reset before starting a programming operation.*

Bit 1 Reserved, must be kept cleared

Bit 0 **BSY**: Busy

This indicates that a Flash operation is in progress. This is set on the beginning of a Flash operation and reset when the operation finishes or when an error occurs.

### 3.5 Flash control register (FLASH\_CR)

Address offset: 0x10

Reset value: 0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
			rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

Bits 31:13 Reserved, must be kept cleared.

Bit 12 **EOPIE**: End of operation interrupt enable

This bit enables the interrupt generation when the EOP bit in the FLASH\_SR register goes to 1.

- 0: Interrupt generation disabled
- 1: Interrupt generation enabled

Bit 11 Reserved, must be kept cleared

Bit 10 **ERRIE**: Error interrupt enable

This bit enables the interrupt generation on an FPEC error (when PGERR / WRPRTErr are set in the FLASH\_SR register).

- 0: Interrupt generation disabled
- 1: Interrupt generation enabled

Bit 9 **OPTWRE**: Option bytes write enable

When set, the option bytes can be programmed. This bit is set on writing the correct key sequence to the FLASH\_OPTKEYR register.  
This bit can be reset by software

Bit 8 Reserved, must be kept cleared.

Bit 7 **LOCK**: Lock

Write to 1 only. When it is set, it indicates that the FPEC and FLASH\_CR are locked. This bit is reset by hardware after detecting the unlock sequence.

In the event of unsuccessful unlock operation, this bit remains set until the next reset.

Bit 6 **STRT**: Start

This bit triggers an ERASE operation when set. This bit is set only by software and reset when the BSY bit is reset.

Bit 5 **OPTER**: Option byte erase

Option byte erase chosen.

Bit 4 **OPTPG**: Option byte programming

Option byte programming chosen.

Bit 3 Reserved, must be kept cleared.

Bit 2 **MER**: Mass erase

Erase of all user pages chosen.

Bit 1 **PER**: Page erase

Page Erase chosen.

Bit 0 **PG**: Programming

Flash programming chosen.

### 3.6 Flash address register (FLASH\_AR)

Address offset: 0x14

Reset value: 0x0000 0000

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

Updated by hardware with the currently/last used address. For Page Erase operations, this should be updated by software to indicate the chosen page.

Bits 31:0 **FAR**: Flash Address

Chooses the address to program when programming is selected, or a page to erase when Page Erase is selected.

*Note:* Write access to this register is blocked when the BSY bit in the FLASH\_SR register is set.

### 3.7 Option byte register (FLASH\_OBR)

Address offset 0x1C

Reset value: 0x03FF FFFC

*Note:* The reset value of this register depends on the value programmed in the option byte and the OPTERR bit reset value depends on the comparison of the option byte and its complement during the option byte loading phase.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Reserved						Data1						Data0						Not used				nRST_STDBY	nRST_STOP	WDG_SW	RDPRT	OPTERR							
						r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:26 Reserved, must be kept cleared.

Bits 25:18 Data1

Bits 17:10 Data0

Bits 9:2 **USER**: User option bytes

This contains the user option byte loaded by the OBL.

Bits [9:5]: Not used (if these bits are written in the Flash option byte, they will be read in this register with no effect on the device.)

Bit 4: nRST\_STDBY

Bit 3: nRST\_STOP

Bit 2: WDG\_SW

Bit 1 **RDPR**: Read protection

When set, this indicates that the Flash memory is read-protected.

*Note: This bit is read-only.*

Bit 0 **OPTERR**: Option byte error

When set, this indicates that the loaded option byte and its complement do not match. The corresponding byte and its complement are read as 0xFF in the FLASH\_OBR or FLASH\_WRPR register.

*Note: This bit is read-only.*

### 3.8 Write protection register (FLASH\_WRPR)

Address offset: 0x20

Reset value: 0xFFFF FFFF

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

Bits 31:0 **WRP**: Write protect

This register contains the write-protection option bytes loaded by the OBL.

0: Write protection active

1: Write protection not active

*Note: These bits are read-only.*

### 3.9 Flash register map

Table 11. Flash interface - register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
0x000	FLASH_ACR Reset value	Reserved																								1	PRFTBS	1	PRFTBE	0	HLFCYA	0	LATENCY [2:0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x004	FLASH_KEYR Reset value	FKEYR[31:0]																								x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x								
0x008	FLASH_OPTKEYR Reset Value	OPTKEYR[31:0]																								x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x								
0x00C	FLASH_SR Reset value	Reserved																								0	EOP	0	WRPRTERR	0	Reserved	0	PGERR	0	ERLYBSY	0	BSY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x010	FLASH_CR Reset value	Reserved																								0	EOPIE	0	Reserved	0	ERRIE	0	OPTWRE	0	Reserved	0	LOCK	1	STRT	0	OPTER	0	OPTPG	0	Reserved	0	MER	0	PER	0	PG	0	0	0	0	0	0								
0x014	FLASH_AR Reset value	FAR[31:0]																								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x018	Reserved																																																																
0x01C	FLASH_OBR Reset value	Reserved								Data1								Data0								Not used								1	nRST_STDBY	1	nRST_STOP	1	WDG_SW	0	RDPR	0	OPTERR																						
0x020	FLASH_WRPR Reset value	WRP[31:0]																																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

## 4 Revision history

**Table 12. Document revision history**

Date	Revision	Changes
30-Aug-2010	1	Initial release. Same content as PM0042 revision 8.
27-Aug-2012	2	Updated <a href="#">Section 2.3.5: Option byte programming on page 16</a> .

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2012 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

[www.st.com](http://www.st.com)

