PM0042
Programming manual

STM32F10xxx Flash programming

Introduction

This programming manual describes how to program the Flash memory of an STM32F10xxx microcontroller.

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 or SWD protocol 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, 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 MCUs supported by this programming manual are the STM32F101xx and STM32F103xx.

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). Read/Write protections and option bytes are also implemented.

Contents

List of tables

List of figures

Glossary

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

- 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/instruction 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 JTAG or SWD protocols 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 Service.
- SIF: small Information Block of option bytes (refer to *Figure 5 on page 20* for further details)
- Option bytes: product configuration bits stored in the Flash memory
- AHB: advanced high-performance bus.

1 Overview

1.1 Features

- 128 Kbyte Flash memory
- Memory organization:
  - Main memory block: 16 Kbits × 64 bits
  - Information block: 258 × 64 bits

Flash interface features:

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

1.2 Flash module organization

The memory organization is based on a main memory block containing 128 pages of 1 Kbyte, and an information block that consists of 2 pages (2 Kbytes and 16 bytes) as shown in *Table 1*.

Table 1. Flash module organization

Block	Name	Addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	4x 1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	
	Page 2	0x0800 0800 - 0x0800 0BFF	
	Page 3	0x0800 0C00 - 0x0800 0FFF	
	Page 4 to 7	0x0800 1000 - 0x0800 1FFF	4x 1 Kbyte
	Page 8 to 11	0x0800 2000 - 0x0800 2FFF	4x 1 Kbyte
	. . .	. . .	. . .
	Page 124 to 127	0x0801 F000 - 0x0801 FFFF	4x 1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbyte
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash 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
	Reserved	0x4002 2024 - 0x4002 2087	100

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*.

System memory is a sector used to boot the device in System memory Boot mode. The area is reserved for use by STMicroelectronics and contains the bootloader 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.

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

The 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 on page 18* 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.

2 Reading/programming the STM32F10xxx 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 32-bit 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 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.

The prefetch buffer can be switched off by resetting a bit in the Flash access control register. After reset, the state of the prefetch buffer is on.

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.

2.2.4 Information block access

The information block (user bootloader) can be executed through the I-Code bus and (data) read through the D-Code bus. The small information block is read- and write-accessible in all modes.
The option byte block contains configuration option bytes and other user-defined information.

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_WRPR)

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 = 0x000000A5
- KEY1 = 0x45670123
- KEY2 = 0xCDEF89AB

2.3.2 Unlocking the Flash memory

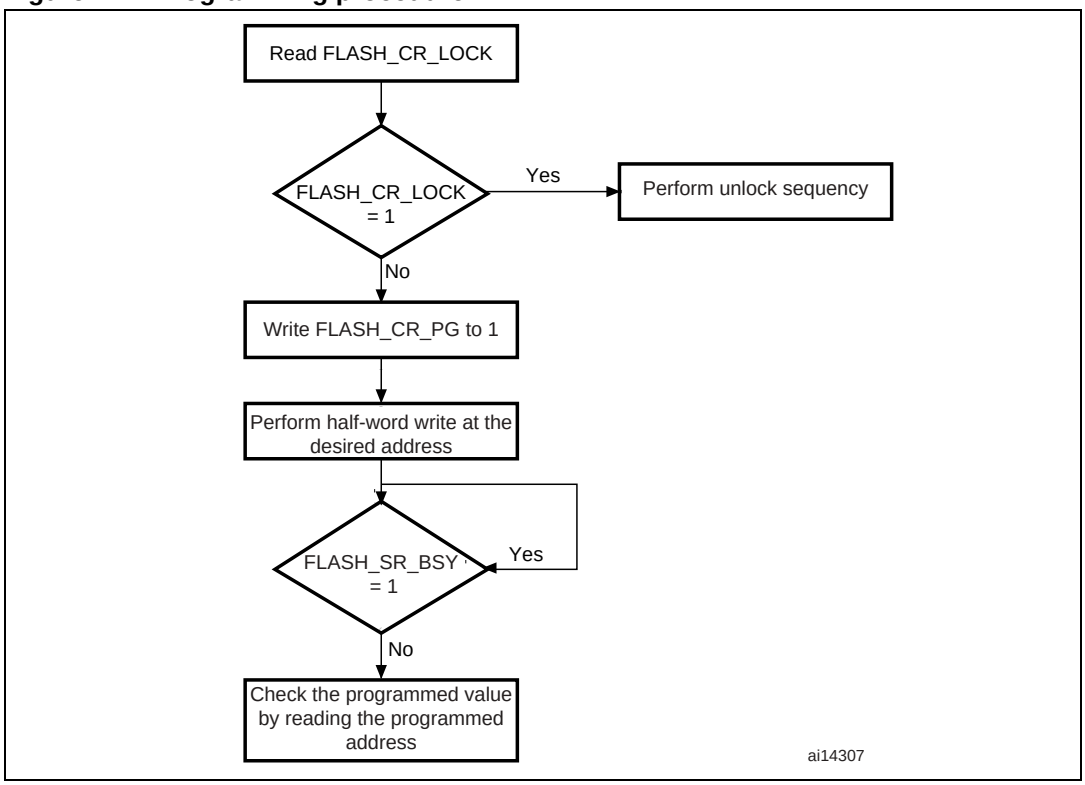
After reset, the FPEC block is protected. The FLASH_CR register is not accessible in write mode. An unlocking sequence should be written to the FLASH_KEYR register to open up the FPEC block. This sequence consists of two write cycles, where two key values (KEY1 and KEY2) are written to the FLASH_KEYR 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 Flash memory programming

The Flash memory can be programmed 16 bits at a time. The program operation is started when the CPU writes a half-word into a 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 Flash memory programming is over.

Figure 1. Programming procedure



Standard programming

In this mode the CPU programs the 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 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 Flash memory location is write-protected by the FLASH_WRPR register, the program operation is skipped and a warning is issued by the WRPRTERR bit in the FLASH_SR register. The end of the program operation is indicated by the EOP bit in the FLASH_SR register.

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

- Check that no 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 Information block programming

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 small information block 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 WRPRTERR 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 user Flash memory is performed before reprogramming the read protection option. If the user wants to change an option other than the protection option, then the mass erase is not performed. The erased state of the read protection option byte protects the Flash memory.

Data programming

Two bytes in the option byte block (at addresses 0x1FFF F804 and 0x1FFF F806) are available for use as data storage. These addresses can be programmed using the option byte programming procedure.

Erase procedure

The small information block 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 small information block and verify

2.3.5 Flash memory erase

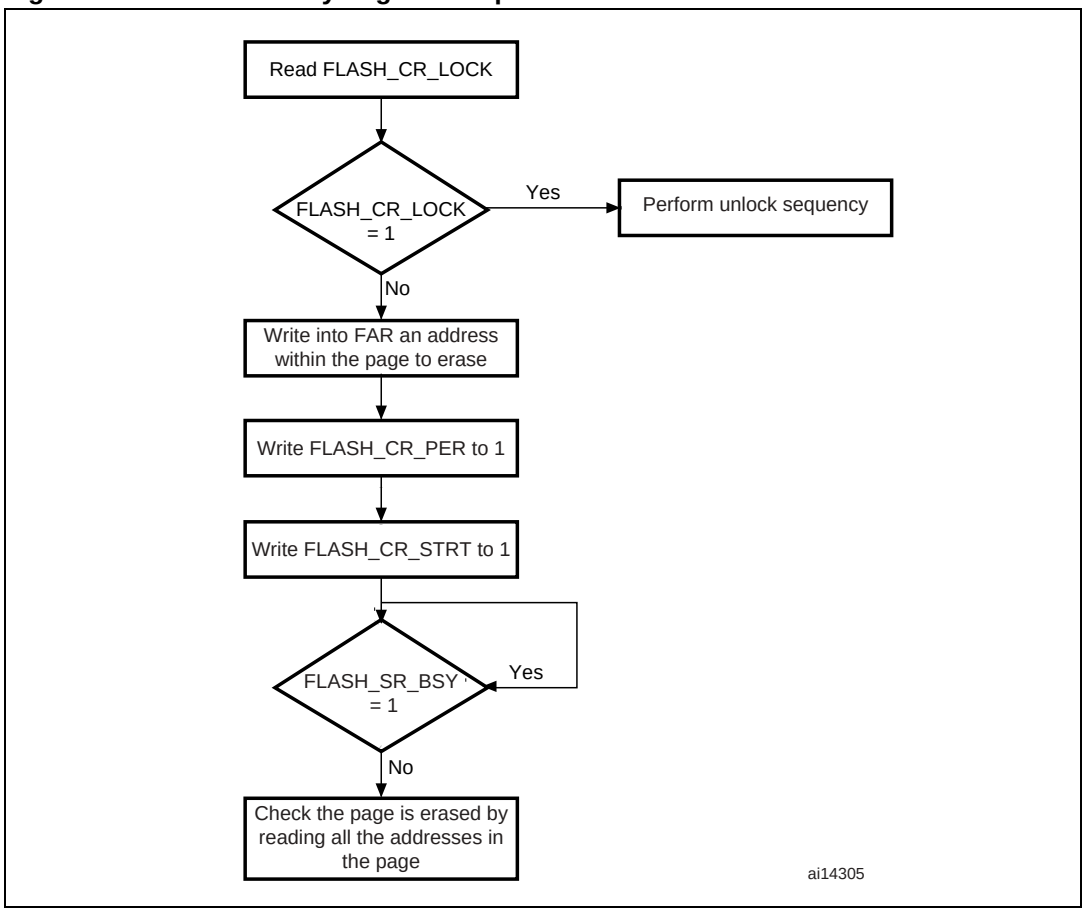
The Flash memory can be erased page by page or completely (Mass Erase).

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
- Program the FLASH_AR register to select a page to erase
- Set the PER bit in the FLASH_CR register
- 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

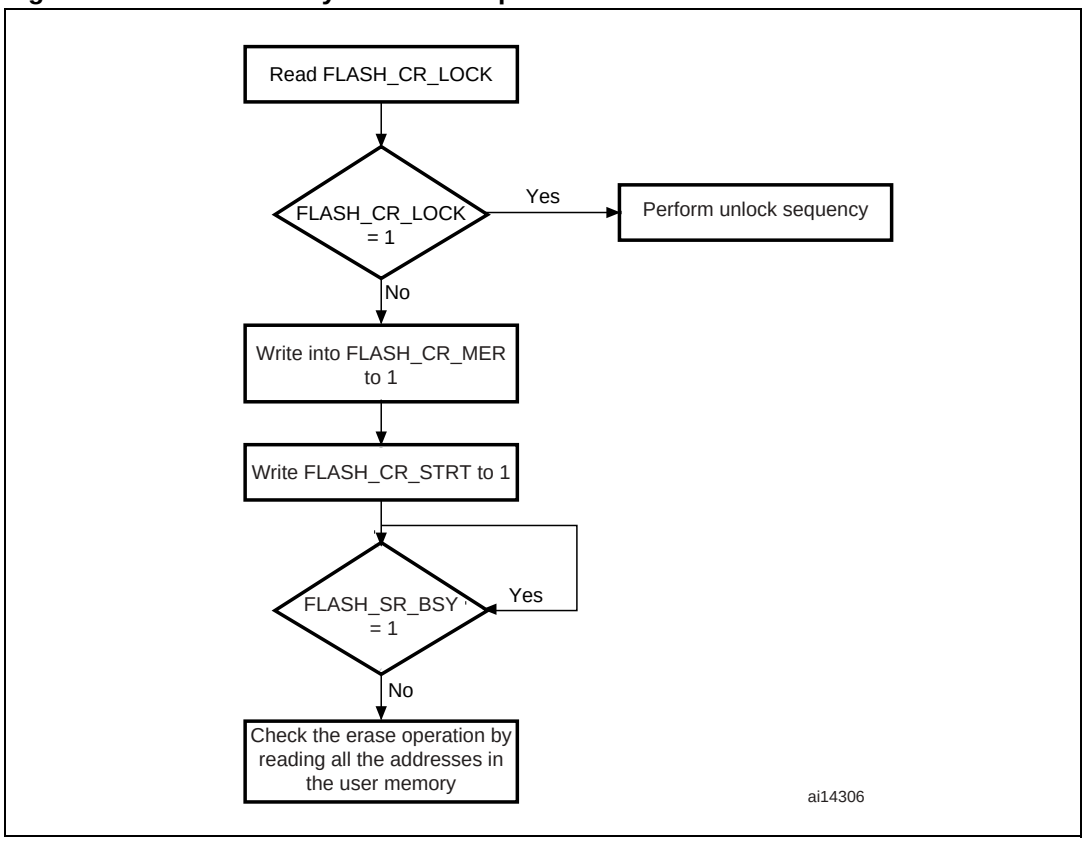


Mass Erase

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



2.4 Protections

The user area of the Flash memory can be protected against read by untrusted code. The pages of the Flash memory, with a granularity of four pages, can also be protected against unwanted write due to loss of program counter contexts.

2.4.1 Read protection

This protection is activated by setting an option byte in the information block. Once the protection byte is programmed to a value, Flash memory read accesses are not allowed when the device is in debug mode. All features linked to loading and executing code in RAM are still active (for example, JTAG/SWD, boot in RAM, etc.) and this can be used to disable the read protection (access to the Flash memory still denied). When the read protection is active, pages 0-3 are automatically write-protected. When the read protection option byte in the information block is altered to a memory-unprotect value, a Mass Erase is performed.

The Flash memory is protected when the RDP option byte and its complement contain the pair of values shown in *Table 2*.

Table 2. Flash memory protection status

RDP byte value	RDP complement value	Read protection status
0xFF	0xFF	Protected
RDP	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:

- Erase the entire small information block. 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 0xA5 to unprotect the memory. This operation first forces a Mass Erase of the main block.
- Reset the device (POR Reset) to reload the option bytes (and the new RDP code), and to disable the read protection.

2.4.2 Write protection

The write protection is implemented with a granularity of four pages at a time. This reduces the number of option bits needed to 32 for the total size of 128 Kbytes. Also it is reasonable to protect 4 Kbytes at a time as any boot code is usually bigger than 1 Kbyte. A summary of protection for the user pages is shown in *Table 3*.

Table 3. Protection of the user pages

RDP	WRP	Action
Active	Active	CPU Read only Debug/intrusive access forbidden
Active	Inactive	CPU R/W Debug/intrusive access forbidden Page 0 write protected
Inactive	Active	CPU Read Debug/intrusive access allowed
Inactive	Inactive	CPU R/W Debug/intrusive access allowed

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).

Unprotection

To disable the write protection:

- Erase the entire small information block by using the OPTER bit in the Flash memory control register (FLASH_CR).
- Program the RDP code (to disable the read protection)
- Program the correct RDP code 0xA5 to enable read access
- 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 Information block protection

Option byte block write protection

The option byte block is always read-accessible and write-protected by default. To gain write access (Program/Erase) to this block, 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 byte block 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 loader

In the information block of the Flash memory, a set of option bytes is stored. These bits contain information on the configuration of the product (HSI RC calibration value, for example). The option bytes are typically selected by the end user depending on their application. An example is the selection of the watchdog in hardware or software mode.

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

Table 4. 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 5*.

Table 5. Small information block organization

Block	Address	[31:24]	[23:16]	[15:8]	[7:0]
Small information block (SIF)	0x1FFFF800	nUSER	USER	nRDP	RDP
	0x1FFFF804	nData1	Data1	nData0	Data0
	0x1FFFF808	nWRP1	WRP1	nWRP0	WRP0
	0x1FFFF80C	nWRP3	WRP3	nWRP2	WRP2

The user option bytes consist of 8 bytes and are mainly used internally within the Flash interface for protection purposes (read and write protection). Only three bytes are used for user purposes.

Table 6. Description of the option bytes

Option bytes	
RDP: *Read protection option byte.* The read protection helps the user protect the software code stored in Flash memory. It is activated by setting an option byte in the information block. When this option byte is programmed to a correct value (RDPRT key = 0x00A5), read access to the Flash memory is allowed.	
USER: *User option byte.* This byte is used to configure the following features: – Select the watchdog event: Hardware or software. – Reset event when entering STOP mode. – Reset event when entering Standby mode.	
Bit 19:23	0xFF: Not used
Bit 18	*nRST_STDBY* 0: Reset generated when entering Standby mode. 1: No reset generated.
Bit 17	*nRST_STOP* 0: Reset generated when entering STOP mode. 1: No reset generated.

Table 6. Description of the option bytes (continued)

Option bytes	
Bit 16	*WDG_SW* 1: Software watchdog. 0: Hardware watchdog.
WRPx: *Flash memory write protection option bytes.* One bit of the user option bytes WRPx is used to protect 4 pages of 1 Kbyte in main memory block. – 0: Write protection active – 1: Write protection not active In total, four user option bytes are used to protect the 128 Kbyte main Flash memory. WRP0: Write-protects pages 0 to 31. WRP1: Write-protects pages 32 to 63. WRP2: Write-protects pages 64 to 95. WRP3: Write-protects pages 96 to 127.	
Datax: two bytes for user data storage. These addresses can be programmed using the option byte programming procedure.	

On every system reset, the option byte loader reads the information block and stores the data into the registers. Each option bit 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 bits (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.

2.5.1 Low-power management

In low-power modes all Flash memory accesses are aborted. Refer to the *STM32F10x Reference Manual* for further information.

3 Register descriptions

In this section, the following abbreviations are used:

Table 7. Abbreviations

Abbreviation	Meaning
read/write (rw)	Software can read from and write to these bits.
read-only (r)	Software can only read these bits.
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.

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															
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										PRFTBS	PRFTBE	HLFCYA	LATENCY		
Res.										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 < SYSCLK≤24 MHz
001 One wait state, if 24 MHz < SYSCLK ≤48 MHz
010 Two wait states, if 48 MHz < SYSCLK ≤72 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															
Res.															
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
Res.										rw	rw		rw		rw

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 ‘0’ before programming.
Reset by writing 1.

Note: The STRT bit should be reset before writing a ‘0’.

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															
Res.															
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	OPT PG	Res.	MER	PER	PG
Res.			rw	Res.	rw	rw	Res.	rw	rw	rw	rw	Res.	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 / small information block 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 / small information block 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
AR[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
AR[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
Reserved															
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						Not used					nRST_STDBY	nRST_STOP	WDG_SW	RDPRT	OPTERR
Res.						r	r	r	r	r	r	r	r	r	r

Bits 31:10 Reserved, must be kept cleared.

Bits 9:2 **USER**: *User Option Bytes*
This contains the user option byte loaded by the OBL.
Bits [9:5]: Not used
Bit 4: nRST_STDBY
Bit 3: nRST_STOP
Bit 2: WDG_SW

Bit 1 **RDPRT**: *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.
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 8. 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**	Reserved																										PRFTBS	PRFTBE	HLFCYA	LATENCY [2:0]		
	Reset Value																											1	1	0	0	0	0
0x004	**FLASH_ KEYR**	FKEYR[31:0]																															
	Reset Value	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**	OPTKEYR[31:0]																															
	Reset Value	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**	Reserved																										EOP	WRPRTERR	Reserved	PGERR	ERLYBSY	BSY
	Reset Value																											0	0		0	0	0
0x010	**FLASH_ CR**	Reserved																			EOPIE	ERLYBSYIE	ERRIE	OPTWRE	Reserved	LOCK	STRT	OPTER	OPTPG	Reserved	MER	PER	PG
	Reset Value																				0	0	0	0		1	0	0	0		0	0	0
0x014	**FLASH_ AR**	FAR[31:0]																															
	Reset Value	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**	Reserved																						Not used					nRST_STDBY	nRST_STOP	WDG_SW	RDPRT	OPTERR
	Reset Value																							1	1	1	1	1	1	1	1	0	0
0x020	**FLASH_ WRPR**	WRP[31:0]																															
	Reset Value	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 9. Document revision history

Date	Revision	Changes
22-Mar-2007	0.1	Initial release.
01-Jun-2007	1	Small text changes. Definition update of STRT bit in *Section 3.5: Flash Control Register (FLASH_CR) on page 25* Alteration of text in *Data programming on page 14* Latency bits description updated *Section 3.1: Flash Access Control Register (FLASH_ACR) on page 22* Definitions of ICP updated *on page 1* and *on page 6* SWD added to debug port definitions with JTAG *on page 6*. OBR reset value modified in *Section 3.7: Option Byte Register (FLASH_OBR)*.
12-Oct-2007	2	Definition of FPEC modified in *Glossary on page 6*. Data at address 0x1FFFF800 updated in *Table 5: Small information block organization*. **Datax** added to and **WRPx** updated in *Table 6: Description of the option bytes*. *Data programming on page 14* updated. *Flash Access Control Register (FLASH_ACR)* reset value corrected. *Option Byte Register (FLASH_OBR)* reset value corrected. *Access time tuner on page 11* corrected. *Prefetch controller* modified. *Section 2.5: Option byte loader* modified. FLASH_ACR and FLASH_OBR reset values updated in *Table 8: Flash interface - register map and reset values*. Information block modified in *Section 1.1: Features* and in *Section 1.2: Flash module organization* (see *Table 1: Flash module organization*). The number of option bytes is limited to 8 instead of 6 (see *Section 2.3.4: Information block programming*). Reserved block removed from *Table 5: Small information block organization*.

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 AN AUTHORIZED ST REPRESENTATIVE, 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.

© 2007 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 - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com