PM0042
Programming manual

STM32F10x Flash programming

Introduction

This programming manual describes how to program the Flash memory of an STM32F10x microcontroller.

The STM32F10x 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 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...) to download programming data into memory. IAP allows you to re-program the Flash memory while the application is executing. Nevertheless, part of the application has to have been previously programmed in one of the Flash banks using ICP.

The MCUs supported by this programming manual are the STM32F101 and STM32F103.

The Flash interface implements the instruction and data accesses over the AHB protocol. It implements a memory partition by implementing a prefetch buffer to speed up memory access. It also implements the logic necessary to carry out Flash operations (Program/Erase). Access/Write protections and option bytes are also implemented.

Glossary

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

- Option Bytes: Product configuration bits stored in the Flash
- Word: Data/Instruction of 32-bit length
- Half word: Data/Instruction of 16 bit length
- Byte: Data/Instruction of 8-bit length
- FPEC (FLASH Program/Erase controller): The write operations to the 2 banks are managed by an embedded FPEC.
- IAP (In-Application Programming): The IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.
- ICP (In-Circuit Programming): The ICP is the ability to program the Flash memory of a microcontroller using JTAG protocol while the device is mounted on the user application board.
- JTAG (Joint Test Action Group): The debug interface of the Cortex-M3 core is based on the Joint Test Action Group (JTAG) protocol.
- 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 DCode bus (literal load and debug access) of the Cortex-M3 to the Flash Data Service.
- SIF: Small Information Block of user option bytes (refer to *Figure 4 on page 15* for further details)

Contents

1 Overview

1.1 Features

- 128 Kbytes Flash
- Endurance: 1000 cycles
- Memory organization:
  - Main memory block: 16K x 64 bits
  - Information block: 320 x 64 bits

Flash interface features:

- Read interface with pre-fetch buffer (2x64-bit words)
- Option byte Loader
- Flash Program / Erase Operation
- Readout / Write Protection

1.2 Flash module organization

Memory organization is based on main memory block with 128 pages of 1Kbyte, and information block composed by 2 pages (2K and 0.5Kbyte) as shown in *Table 1*.

Table 1. Flash module organization

Block	Name	Addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	4x 1K
	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 1K
	Page 8 to 11	0x0800 2000 - 0x0800 2FFF	4x 1K
	. . .	. . .	. . .
	Page 124 to 127	0x0801 F000 - 0x0801 FFFF	4x 1K
Information block	Boot Loader Code	0x1FFF F000 - 0x1FFF F7FF	2K
	User Option Bytes	0x1FFF F800 - 0x1FFF F9FF	512
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 program memory is organized in 32-bit wide memory cells which 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 STM32F10x Microcontroller type. For the base address, please refer to the related *STM32F10x Reference Manual*.

SystemMemory is a sector used to boot the device in SystemMemory Boot Mode. The area is reserved for use by STMicroelectronics. It is programmed by ST when the device is manufactured and protected against spurious write/erase operations.

The write operations of the main memory 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
- Readout Protection

Refer to *Section 2.4 on page 14* for more details.

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

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

You can program Flash memory using In-Circuit Programming and In-Application programming.

2 Reading/programming the STM32F10x embedded Flash

2.1 Introduction

This section describes how to read or to program the STM32F10x embedded Flash.

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 and an AHB interface on the other side to interface with the CPU. The main task of this interface is to generate the control signals for the Flash read and to pre-fetch the blocks required by the CPU. The pre-fetch block is only used for instruction fetches over I-Code bus. The Literal pool is accessed over D-Code bus. Since the target of these 2 buses are the same Flash, the D-code bus accesses have priority over the pre-fetch accesses.

2.2.1 Instruction Fetch

The Cortex-M3 fetches the instruction over I-Code bus and the literal pool over D-code bus. The Pre-fetch block aims at increasing the efficiency of the I-Code bus accesses.

Prefetch buffer

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

The implementation of this pre-fetch 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 depending 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 pre-fetch buffer can be switched off by resetting a bit in the Flash access control register. After reset, the state of the pre-fetch buffer is off.

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

In case of non-availability of a high frequency clock in the system, the Flash accesses can be made on a half cycle of the HCLK if the frequency of HCLK permits (the half-cycle access can only be used with a low frequency clock of <8 MHz which can be obtained with the use of HSI or HSE and no PLL use). This mode can be chosen by setting a control bit in Flash access control register.

Note: The half cycle access cannot be used when the prefetch buffer is used and when there is a prescaler other than 1 on the AHB clock.

Access time tuner

In order to maintain the control signals for read of the Flash, the ratio of Pre-fetch controller clock period with respect to the access time of the Flash 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 to correctly read the required data. After reset, the value is one and the Flash is accessed in two cycles (FLASH_ACR reset value is 01, latency=1).

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. The D-code accesses have priority over the pre-fetch accesses. This interface uses the Access Time Tuner block of the pre-fetch buffer.

2.2.3 Flash access controller

This block is essentially a simple arbiter between the read requests of the pre-fetch/I-code and D-Code interfaces.

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

2.2.4 Information block access

The information block can be executed through I-Code bus and read (data) through D-Code bus. The small information block is accessible in read as well as write 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 programming and erase operations of the Flash. The FPEC consists of 7 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 on-going Flash operation does not block the CPU as long as CPU does not access the Flash.

2.3.1 Key values

The key values are as follows:

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

2.3.2 Unlocking the Flash

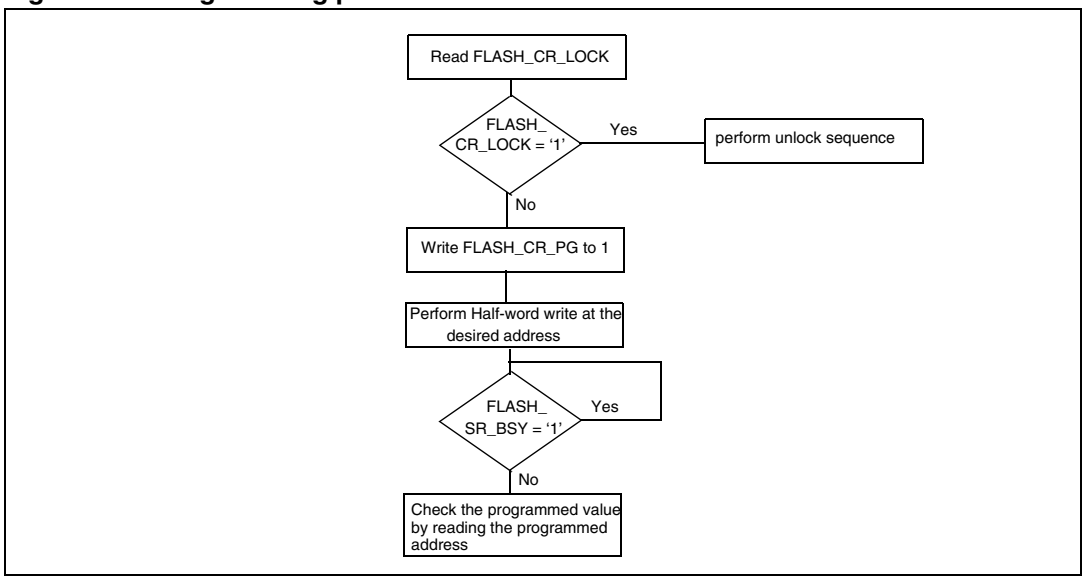
After reset, the FPEC block is protected. The FLASH_CR register is not accessible in write mode. An unlocking sequence should be written into the FLASH_KEYR register to open up the FPEC block. The sequence consists of writing 2 key values (KEY1 and KEY2) into the address of FLASH_KEYR (refer to *Section 2.3.1* for key values). Any wrong sequence will lock 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 if the KEY1 does not match or during the second write if KEY1 has been correctly written and KEY2 value does not match. The FPEC block and FLASH_CR register can be locked by the user 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 Programming

The Flash can be programmed 16-bits at a time. Programming operation is started when the CPU writes a half-word into a Flash address with the PG bit of the FLASH_CR register set. Any writes of size other than half-word will result in the bus error response from the FPEC. If a read/write occurs during programming, (BSY bit set), the CPU is stalled until the ongoing Flash programming is over.

Figure 1. Programming procedure



Standard programming

In this mode the CPU programs the Flash in the same way as a standard half-word write. The PG bit in FLASH_CR register must be set. FPEC preliminarily reads the value of the

addressed Flash location and checks it is erased then program operation is skipped and a warning is issued by PGERR bit in FLASH_SR register (the only exception to this is when a 0x0000 is programmed. Here, the location is correctly programmed to 0x0000 and the PGERR bit is not set). If the addressed Flash 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. End of program operation is signaled by the EOP bit in the FLASH_SR register.

The Flash programming sequence in standard mode is as follows:

- Check that no Flash operation is on-going 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 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 a normal user address. The number of option bytes is limited to 6 (4 for write protection, 1 for Readout protection and 1 for configuration). After unlocking the FPEC, the user has to authorize the small information block programming 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 followed by a half-word write on the desired Flash address.

FPEC preliminarily reads the value of the addressed option byte and checks it is erased, otherwise program operation is skipped and a warning is issued by the WRPRTERR bit in the FLASH_SR register. End of program operation is signaled by the EOP bit in the FLASH_SR register.

The FPEC takes the LSB and computes automatically 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 operation is on-going 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
- Perform the write the data (half-word) at the desired address
- Wait for BSY bit to be reset.
- Read the programmed value and verify.

When the protection option is changed on read protected Flash from protected to unprotected, a mass erase of user Flash is performed before reprogramming the readout protection option. If the user wants to change an option other than protection option, then

the mass erase is not performed. The erased state of readout protection option byte protects the Flash.

Data programming

The remaining bytes after the option bytes block (the non-used OPT bytes at 0x1FFF F804) are available for use for data storage. The programming of these addresses can be done through the standard programming procedure.

Erase procedure

The small information block erase sequence (OPTERASE) is as follows:

- Check that no Flash 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 Erase

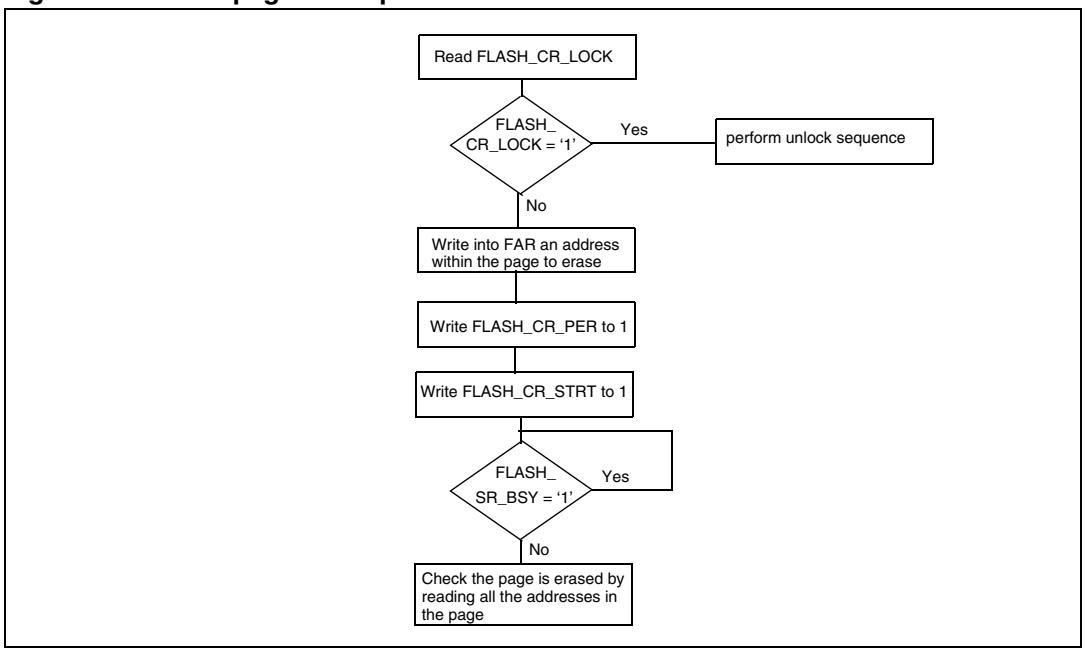
Flash can be erased page by page or completely (Mass Erase).

Page Erase

A page of the Flash 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 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 BSY bit to be reset
- Read the Erased page and verify

Figure 2. Flash page erase procedure

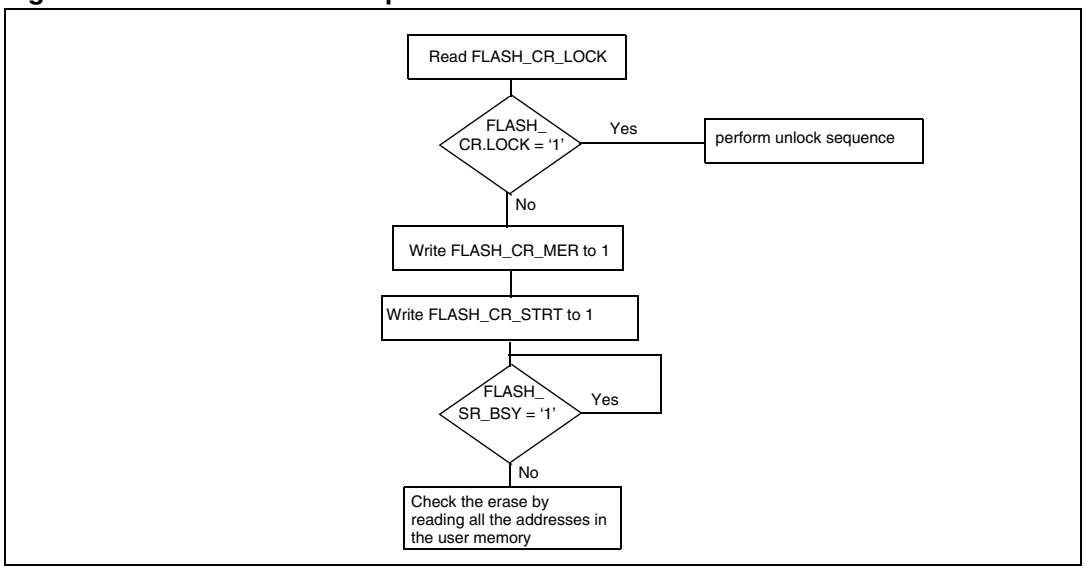


Mass Erase

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

- Check that no Flash operation is on-going by checking the BSY bit in the0 FLASH_SR register
- Set the MER bit in the FLASH_CR register
- Set the STRT bit in the FLASH_CR register
- Wait for BSY bit to be reset
- Read all the pages and verify

Figure 3. Flash mass erase procedure



2.4 Protections

The user area of the Flash can be protected against a read by untrusted code. The individual pages of the Flash with a granularity of 4 pages at a time can also be protected against unwanted write due to loss of program counter contexts.

2.4.1 Readout protection

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

The Flash is protected when the RDP option byte and its complement contains the following pair of values:

RDP Byte value	RDP Complementary value	Readout 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 readout protection:

- Erase the entire Small Information block (user part). The result of Readout protection code (RDP) will be 0xFF, however the Readout protection will be still enabled.
- Program the correct code 0xA5 of the RDP to unprotect the memory. This operation willfirst force 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 Readout protection.

2.4.2 Write protection

The write protection is implemented with a choice of protecting 4 pages at a time. This reduces the number of option bits needed to 32 for the total size of 128k. Also it is reasonable to protect 4k at a time as any boot code is usually bigger than 1k. A summary of protection for the user pages are as follows:

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

RDP	WRP	Action
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 returns a protection error flag on the Flash status register (FLASH_SR).

Unprotection

To disable the write protection:

- Erase the entire Small information block (user part) by using the OPTER bit in the Flash control register (FLASH_CR).
- Program the RDP code (for readout protection disabling)
- Program the correct code 0xA5 of RDP to enable the 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 (Prog/Erase) to this block a sequence of keys (Same as that of lock) have to be written into the OPTKEYR. A correct sequence of keys will open up the option byte block to write accesses and this is indicated by OPTWRE in the FLASH_CR register being set. The write accesses can be disabled by resetting the bit through software.

2.5 Option byte loader

- On the information block of the Flash, a set of option bytes are stored. These bits contain information on the configuration of the product (the package, for example). The user option bytes are typically selected by the end user depending on his application. An example is the selection of watchdog in hardware or software mode.

A 32-bit word is divided as follows in the option byte.

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 below

Figure 4. Information block organization

Block	Address	[31:24]	[23:16]	[15:8]	[7:0]
Reserved	0x1FFFF7F8	Reserved			
	0x1FFFF7FC	Reserved			

Block	Address	[31:24]	[23:16]	[15:8]	[7:0]
Small information block (SIF)	0x1FFFF800	nUSER	USER	nRDP	RDP
	0x1FFFF804	Not used			
	0x1FFFF808	nWRP1	WRP1	nWRP0	WRP0
	0x1FFFF80C	nWRP3	WRP3	nWRP2	WRP2

Note: 1 The RDPRT KEY consists of 8 bits which is decoded into a single option which is output from this IP as rdprot.

The User Option Bytes are composed by 6 bytes and are mainly used internally within the Flash interface for protection purposes (Readout and Write protection). Only one byte is used for User purposes.

Table 2. User Option Bytes description

RDP: *ReadOut protection option Byte.* The readout protection helps the user to protect the software code stored in Flash. 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), the Flash is open for any access.	
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.
Bit 16	*WDG_SW* 1: Software watchdog. 0: Hardware watchdog.
WRPx: *Flash Write Protection Option Bytes.* One bit of the user option bytes WRPx is used to protect 4 pages of 1Kbytes in main memory block. A total of 4 user option bytes are used to protect all the 128K main Flash. WRP0: Write protect pages 0 to 31. WRP1: Write protect pages 32 to 63. WRP2: Write protect pages 64 to 95. WRP3: Write protect pages 96 to 128.	

On every system reset, the option byte loader performs a read of the information block and stores the data into the registers. Each option bit has also its complement in the information block. During option loading, by verification of the option bit and its complement the loading can be checked that it 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 by the CPU. See *Section 3: Register descriptions* for more details.

The option byte register is accessible in read mode by the CPU.

2.6 Low power management

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

3 Register descriptions

In this section, the following abbreviations are used:

Table 3. Abbreviations

read/write (rw)	Software can read 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: 0x00h
Reset value: 0x0000 0001h

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										PRFTBS	PRFTBE	HLFCYA	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 HCLK period to the Flash Access time. 000: Zero wait state, if 0 < HCLK ≤ 24 MHz 001: One wait state, if 24 MHz < HCLK ≤ 56 MHz 010. Two wait states, if 56 MHz < HCLK ≤ 72 MHz

3.2 FPEC Key Register (FLASH_KEYR)

Address offset: 0x04h
Reset value: xxxx xxxxh

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: 0x08h
Reset value: xxxx xxxxh

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: 0x0Ch
Reset value: 0x0000 0000h

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		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. 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: 0x10h
Reset value: 0x0000 0080h

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	OPT PG	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, this allows the access to the programming the option bytes / small information block. 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 set, this 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 set.
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: 0x14h
Reset value: 0x0000 0000h

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 the programming is chosen or a page to erase when the page erase is chosen. **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 0x1Ch
Reset value: 0xFFFF FFFEh

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															

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
						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**: *Readout protection* When set, this indicates that the Flash 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: 0x20h

Reset value: 0xFFFF FFFFh

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 4. 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
000h	**FLASH_ ACR**	Reserved																										PRFTBS	PRFTBE	HLFCYA	LATENCY [2:0]		
	Reset Value																											0	0	0	0	0	1
004h	**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
008h	**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
00Ch	**FLASH_ SR**	Reserved																										EOP	WRPRTERR	Reserved	PGERR	ERLYBSY	BSY
	Reset Value																											0	0		0	0	0
010h	**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
014h	**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
018h	Reserved																																
01Ch	**FLASH_ OBR**	Reserved																						Not used					nRST_STDBY	nRST_STOP	WDG_SW	RDPRT	OPTERR
	Reset Value																							1	1	1	1	1	1	1	1	1	0
020h	**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 5. Document revision history

Date	Revision	Changes
22-Mar-2007	0.1	Initial release