26 HAL FLASH Generic Driver

26.1 FLASH Firmware driver registers structures

26.1.1 FLASH_ProcessTypeDef

FLASH_ProcessTypeDef is defined in the stm32f4xx_hal_flash.h

Data Fields

- ***__IO FLASH_ProcedureTypeDef ProcedureOnGoing***
- ***__IO uint32_t NbSectorsToErase***
- ***__IO uint8_t VoltageForErase***
- ***__IO uint32_t Sector***
- ***__IO uint32_t Bank***
- ***__IO uint32_t Address***
- ***HAL_LockTypeDef Lock***
- ***__IO uint32_t ErrorCode***

Field Documentation

- ***__IO FLASH_ProcedureTypeDef FLASH_ProcessTypeDef::ProcedureOnGoing***
- ***__IO uint32_t FLASH_ProcessTypeDef::NbSectorsToErase***
- ***__IO uint8_t FLASH_ProcessTypeDef::VoltageForErase***
- ***__IO uint32_t FLASH_ProcessTypeDef::Sector***
- ***__IO uint32_t FLASH_ProcessTypeDef::Bank***
- ***__IO uint32_t FLASH_ProcessTypeDef::Address***
- ***HAL_LockTypeDef FLASH_ProcessTypeDef::Lock***
- ***__IO uint32_t FLASH_ProcessTypeDef::ErrorCode***

26.2 FLASH Firmware driver API description

The following section lists the various functions of the FLASH library.

26.2.1 FLASH peripheral features

The Flash memory interface manages CPU AHB I-Code and D-Code accesses to the Flash memory. It implements the erase and program Flash memory operations and the read and write protection mechanisms.

The Flash memory interface accelerates code execution with a system of instruction prefetch and cache lines.

The FLASH main features are:

- Flash memory read operations
- Flash memory program/erase operations
- Read / write protections
- Prefetch on I-Code
- 64 cache lines of 128 bits on I-Code
- 8 cache lines of 128 bits on D-Code

26.2.2 How to use this driver

This driver provides functions and macros to configure and program the FLASH memory of all STM32F4xx devices.

1. FLASH Memory IO Programming functions:
   - Lock and Unlock the FLASH interface using HAL_FLASH_Unlock() and HAL_FLASH_Lock() functions
   - Program functions: byte, half word, word and double word
   - There Two modes of programming :
     - Polling mode using HAL_FLASH_Program() function
     - Interrupt mode using HAL_FLASH_Program_IT() function
2. Interrupts and flags management functions :
   - Handle FLASH interrupts by calling HAL_FLASH_IRQHandler()
   - Wait for last FLASH operation according to its status
   - Get error flag status by calling HAL_SetErrorCode()

In addition to these functions, this driver includes a set of macros allowing to handle the following operations:

- Set the latency
- Enable/Disable the prefetch buffer
- Enable/Disable the Instruction cache and the Data cache
- Reset the Instruction cache and the Data cache
- Enable/Disable the FLASH interrupts
- Monitor the FLASH flags status

26.2.3 Programming operation functions

This subsection provides a set of functions allowing to manage the FLASH program operations.

This section contains the following APIs:

- ***HAL_FLASH_Program()***
- ***HAL_FLASH_Program_IT()***
- ***HAL_FLASH_IRQHandler()***
- ***HAL_FLASH_EndOfOperationCallback()***
- ***HAL_FLASH_OperationErrorCallback()***

26.2.4 Peripheral Control functions

This subsection provides a set of functions allowing to control the FLASH memory operations.

This section contains the following APIs:

- ***HAL_FLASH_Unlock()***
- ***HAL_FLASH_Lock()***
- ***HAL_FLASH_OB_Unlock()***
- ***HAL_FLASH_OB_Lock()***
- ***HAL_FLASH_OB_Launch()***

26.2.5 Peripheral Errors functions

This subsection permits to get in run-time Errors of the FLASH peripheral.

This section contains the following APIs:

- ***HAL_FLASH_GetError()***
- ***FLASH_WaitForLastOperation()***

26.2.6 Detailed description of functions

HAL_FLASH_Program

Function name

HAL_StatusTypeDef HAL_FLASH_Program (uint32_t TypeProgram, uint32_t Address, uint64_t Data)

Function description

Program byte, halfword, word or double word at a specified address.

Parameters

- **TypeProgram:** Indicate the way to program at a specified address. This parameter can be a value of FLASH Type Program
- **Address:** specifies the address to be programmed.
- **Data:** specifies the data to be programmed

Return values

- **HAL_StatusTypeDef:** HAL Status

HAL_FLASH_Program_IT

Function name

HAL_StatusTypeDef HAL_FLASH_Program_IT (uint32_t TypeProgram, uint32_t Address, uint64_t Data)

Function description

Program byte, halfword, word or double word at a specified address with interrupt enabled.

Parameters

- **TypeProgram:** Indicate the way to program at a specified address. This parameter can be a value of FLASH Type Program
- **Address:** specifies the address to be programmed.
- **Data:** specifies the data to be programmed

Return values

- **HAL:** Status

HAL_FLASH_IRQHandler

Function name

void HAL_FLASH_IRQHandler (void)

Function description

This function handles FLASH interrupt request.

Return values

- **None:**

HAL_FLASH_EndOfOperationCallback

Function name

void HAL_FLASH_EndOfOperationCallback (uint32_t ReturnValue)

Function description

FLASH end of operation interrupt callback.

Parameters

- **ReturnValue:** The value saved in this parameter depends on the ongoing procedure Mass Erase: Bank number which has been requested to erase Sectors Erase: Sector which has been erased (if 0xFFFFFFFFU, it means that all the selected sectors have been erased) Program: Address which was selected for data program

Return values

- **None:**

HAL_FLASH_OperationErrorCallback

Function name

void HAL_FLASH_OperationErrorCallback (uint32_t ReturnValue)

Function description

FLASH operation error interrupt callback.

Parameters

- **ReturnValue:** The value saved in this parameter depends on the ongoing procedure Mass Erase: Bank number which has been requested to erase Sectors Erase: Sector number which returned an error Program: Address which was selected for data program

Return values

- **None:**

HAL_FLASH_Unlock

Function name

HAL_StatusTypeDef HAL_FLASH_Unlock (void)

Function description

Unlock the FLASH control register access.

Return values

- **HAL:** Status

HAL_FLASH_Lock

Function name

HAL_StatusTypeDef HAL_FLASH_Lock (void)

Function description

Locks the FLASH control register access.

Return values

- **HAL:** Status

HAL_FLASH_OB_Unlock

Function name

HAL_StatusTypeDef HAL_FLASH_OB_Unlock (void)

Function description

Unlock the FLASH Option Control Registers access.

Return values

- **HAL:** Status

HAL_FLASH_OB_Lock

Function name

HAL_StatusTypeDef HAL_FLASH_OB_Lock (void)

Function description

Lock the FLASH Option Control Registers access.

Return values

- **HAL:** Status

HAL_FLASH_OB_Launch

Function name

HAL_StatusTypeDef HAL_FLASH_OB_Launch (void)

Function description

Launch the option byte loading.

Return values

- **HAL:** Status

HAL_FLASH_GetError

Function name

uint32_t HAL_FLASH_GetError (void)

Function description

Get the specific FLASH error flag.

Return values

- **FLASH_ErrorCode:** The returned value can be a combination of:
  - HAL_FLASH_ERROR_RD: FLASH Read Protection error flag (PCROP)
  - HAL_FLASH_ERROR_PGS: FLASH Programming Sequence error flag
  - HAL_FLASH_ERROR_PGP: FLASH Programming Parallelism error flag
  - HAL_FLASH_ERROR_PGA: FLASH Programming Alignment error flag
  - HAL_FLASH_ERROR_WRP: FLASH Write protected error flag
  - HAL_FLASH_ERROR_OPERATION: FLASH operation Error flag

FLASH_WaitForLastOperation

Function name

HAL_StatusTypeDef FLASH_WaitForLastOperation (uint32_t Timeout)

Function description

Wait for a FLASH operation to complete.

Parameters

- **Timeout:** maximum flash operationtimeout

Return values

- **HAL:** Status

26.3 FLASH Firmware driver defines

The following section lists the various define and macros of the module.

26.3.1 FLASH

FLASH

FLASH Error Code

HAL_FLASH_ERROR_NONE

No error

HAL_FLASH_ERROR_RD

Read Protection error

HAL_FLASH_ERROR_PGS

Programming Sequence error

HAL_FLASH_ERROR_PGP

Programming Parallelism error

HAL_FLASH_ERROR_PGA

Programming Alignment error

HAL_FLASH_ERROR_WRP

Write protection error

HAL_FLASH_ERROR_OPERATION

Operation Error

FLASH Exported Macros

__HAL_FLASH_SET_LATENCY

Description:

- Set the FLASH Latency.

Parameters:

- __LATENCY__: FLASH Latency The value of this parameter depend on device used within the same series

Return value:

- none

__HAL_FLASH_GET_LATENCY

Description:

- Get the FLASH Latency.

Return value:

- FLASH: Latency The value of this parameter depend on device used within the same series

__HAL_FLASH_PREFETCH_BUFFER_ENABLE

Description:

- Enable the FLASH prefetch buffer.

Return value:

- none

__HAL_FLASH_PREFETCH_BUFFER_DISABLE

Description:

- Disable the FLASH prefetch buffer.

Return value:

- none

__HAL_FLASH_INSTRUCTION_CACHE_ENABLE

Description:

- Enable the FLASH instruction cache.

Return value:

- none

__HAL_FLASH_INSTRUCTION_CACHE_DISABLE

Description:

- Disable the FLASH instruction cache.

Return value:

- none

__HAL_FLASH_DATA_CACHE_ENABLE

Description:

- Enable the FLASH data cache.

Return value:

- none

__HAL_FLASH_DATA_CACHE_DISABLE

Description:

- Disable the FLASH data cache.

Return value:

- none

__HAL_FLASH_INSTRUCTION_CACHE_RESET

Description:

- Resets the FLASH instruction Cache.

Return value:

- None

Notes:

- This function must be used only when the Instruction Cache is disabled.

__HAL_FLASH_DATA_CACHE_RESET

Description:

- Resets the FLASH data Cache.

Return value:

- None

Notes:

- This function must be used only when the data Cache is disabled.

__HAL_FLASH_ENABLE_IT

Description:

- Enable the specified FLASH interrupt.

Parameters:

- __INTERRUPT__: FLASH interrupt This parameter can be any combination of the following values:
  - FLASH_IT_EOP: End of FLASH Operation Interrupt
  - FLASH_IT_ERR: Error Interrupt

Return value:

- none

__HAL_FLASH_DISABLE_IT

Description:

- Disable the specified FLASH interrupt.

Parameters:

- __INTERRUPT__: FLASH interrupt This parameter can be any combination of the following values:
  - FLASH_IT_EOP: End of FLASH Operation Interrupt
  - FLASH_IT_ERR: Error Interrupt

Return value:

- none

__HAL_FLASH_GET_FLAG

Description:

- Get the specified FLASH flag status.

Parameters:

- __FLAG__: specifies the FLASH flags to check. This parameter can be any combination of the following values:
  - FLASH_FLAG_EOP : FLASH End of Operation flag
  - FLASH_FLAG_OPERR : FLASH operation Error flag
  - FLASH_FLAG_WRPERR: FLASH Write protected error flag
  - FLASH_FLAG_PGAERR: FLASH Programming Alignment error flag
  - FLASH_FLAG_PGPERR: FLASH Programming Parallelism error flag
  - FLASH_FLAG_PGSERR: FLASH Programming Sequence error flag
  - FLASH_FLAG_RDERR : FLASH Read Protection error flag (PCROP) (*)
  - FLASH_FLAG_BSY : FLASH Busy flag (*) FLASH_FLAG_RDERR is not available for STM32F405xx/407xx/415xx/417xx devices

Return value:

- The: new state of __FLAG__ (SET or RESET).

__HAL_FLASH_CLEAR_FLAG

Description:

- Clear the specified FLASH flags.

Parameters:

- __FLAG__: specifies the FLASH flags to clear. This parameter can be any combination of the following values:
  - FLASH_FLAG_EOP : FLASH End of Operation flag
  - FLASH_FLAG_OPERR : FLASH operation Error flag
  - FLASH_FLAG_WRPERR: FLASH Write protected error flag
  - FLASH_FLAG_PGAERR: FLASH Programming Alignment error flag
  - FLASH_FLAG_PGPERR: FLASH Programming Parallelism error flag
  - FLASH_FLAG_PGSERR: FLASH Programming Sequence error flag
  - FLASH_FLAG_RDERR : FLASH Read Protection error flag (PCROP) (*) (*) FLASH_FLAG_RDERR is not available for STM32F405xx/407xx/415xx/417xx devices

Return value:

- none

FLASH Flag definition

FLASH_FLAG_EOP

FLASH End of Operation flag

FLASH_FLAG_OPERR

FLASH operation Error flag

FLASH_FLAG_WRPERR

FLASH Write protected error flag

FLASH_FLAG_PGAERR

FLASH Programming Alignment error flag

FLASH_FLAG_PGPERR

FLASH Programming Parallelism error flag

FLASH_FLAG_PGSERR

FLASH Programming Sequence error flag

FLASH_FLAG_RDERR

Read Protection error flag (PCROP)

FLASH_FLAG_BSY

FLASH Busy flag

FLASH Interrupt definition

FLASH_IT_EOP

End of FLASH Operation Interrupt source

FLASH_IT_ERR

Error Interrupt source

FLASH Private macros to check input parameters

IS_FLASH_TYPEPROGRAM

FLASH Keys

RDP_KEY

FLASH_KEY1

FLASH_KEY2

FLASH_OPT_KEY1

FLASH_OPT_KEY2

FLASH Latency

FLASH_LATENCY_0

FLASH Zero Latency cycle

FLASH_LATENCY_1

FLASH One Latency cycle

FLASH_LATENCY_2

FLASH Two Latency cycles

FLASH_LATENCY_3

FLASH Three Latency cycles

FLASH_LATENCY_4

FLASH Four Latency cycles

FLASH_LATENCY_5

FLASH Five Latency cycles

FLASH_LATENCY_6

FLASH Six Latency cycles

FLASH_LATENCY_7

FLASH Seven Latency cycles

FLASH_LATENCY_8

FLASH Eight Latency cycles

FLASH_LATENCY_9

FLASH Nine Latency cycles

FLASH_LATENCY_10

FLASH Ten Latency cycles

FLASH_LATENCY_11

FLASH Eleven Latency cycles

FLASH_LATENCY_12

FLASH Twelve Latency cycles

FLASH_LATENCY_13

FLASH Thirteen Latency cycles

FLASH_LATENCY_14

FLASH Fourteen Latency cycles

FLASH_LATENCY_15

FLASH Fifteen Latency cycles

FLASH Program Parallelism

FLASH_PSIZE_BYTE

FLASH_PSIZE_HALF_WORD

FLASH_PSIZE_WORD

FLASH_PSIZE_DOUBLE_WORD

CR_PSIZE_MASK

FLASH Type Program

FLASH_TYPEPROGRAM_BYTE

Program byte (8-bit) at a specified address

FLASH_TYPEPROGRAM_HALFWORD

Program a half-word (16-bit) at a specified address

FLASH_TYPEPROGRAM_WORD

Program a word (32-bit) at a specified address

FLASH_TYPEPROGRAM_DOUBLEWORD

Program a double word (64-bit) at a specified address

27 HAL FLASH Extension Driver

27.1 FLASHEx Firmware driver registers structures

27.1.1 FLASH_EraseInitTypeDef

FLASH_EraseInitTypeDef is defined in the stm32f4xx_hal_flash_ex.h

Data Fields

- ***uint32_t TypeErase***
- ***uint32_t Banks***
- ***uint32_t Sector***
- ***uint32_t NbSectors***
- ***uint32_t VoltageRange***

Field Documentation

- ***uint32_t FLASH_EraseInitTypeDef::TypeErase***
  Mass erase or sector Erase. This parameter can be a value of ***FLASHEx_Type_Erase***
- ***uint32_t FLASH_EraseInitTypeDef::Banks***
  Select banks to erase when Mass erase is enabled. This parameter must be a value of ***FLASHEx_Banks***
- ***uint32_t FLASH_EraseInitTypeDef::Sector***
  Initial FLASH sector to erase when Mass erase is disabled This parameter must be a value of ***FLASHEx_Sectors***
- ***uint32_t FLASH_EraseInitTypeDef::NbSectors***
  Number of sectors to be erased. This parameter must be a value between 1 and (max number of sectors - value of Initial sector)
- ***uint32_t FLASH_EraseInitTypeDef::VoltageRange***
  The device voltage range which defines the erase parallelism This parameter must be a value of ***FLASHEx_Voltage_Range***

27.1.2 FLASH_OBProgramInitTypeDef

FLASH_OBProgramInitTypeDef is defined in the stm32f4xx_hal_flash_ex.h

Data Fields

- ***uint32_t OptionType***
- ***uint32_t WRPState***
- ***uint32_t WRPSector***
- ***uint32_t Banks***
- ***uint32_t RDPLevel***
- ***uint32_t BORLevel***
- ***uint8_t USERConfig***

Field Documentation

- ***uint32_t FLASH_OBProgramInitTypeDef::OptionType***
  Option byte to be configured. This parameter can be a value of ***FLASHEx_Option_Type***
- ***uint32_t FLASH_OBProgramInitTypeDef::WRPState***
  Write protection activation or deactivation. This parameter can be a value of ***FLASHEx_WRP_State***
- ***uint32_t FLASH_OBProgramInitTypeDef::WRPSector***
  Specifies the sector(s) to be write protected. The value of this parameter depend on device used within the same series
- ***uint32_t FLASH_OBProgramInitTypeDef::Banks***
  Select banks for WRP activation/deactivation of all sectors. This parameter must be a value of ***FLASHEx_Banks***
- ***uint32_t FLASH_OBProgramInitTypeDef::RDPLevel***
  Set the read protection level. This parameter can be a value of ***FLASHEx_Option_Bytes_Read_Protection***

- ***uint32_t FLASH_OBProgramInitTypeDef::BORLevel***
  Set the BOR Level. This parameter can be a value of ***FLASHEx_BOR_Reset_Level***
- ***uint8_t FLASH_OBProgramInitTypeDef::USERConfig***
  Program the FLASH User Option Byte: IWDG_SW / RST_STOP / RST_STDBY.

27.1.3 FLASH_AdvOBProgramInitTypeDef

FLASH_AdvOBProgramInitTypeDef is defined in the stm32f4xx_hal_flash_ex.h

Data Fields

- ***uint32_t OptionType***
- ***uint32_t PCROPState***
- ***uint32_t Banks***
- ***uint16_t SectorsBank1***
- ***uint16_t SectorsBank2***
- ***uint8_t BootConfig***

Field Documentation

- ***uint32_t FLASH_AdvOBProgramInitTypeDef::OptionType***
  Option byte to be configured for extension. This parameter can be a value of ***FLASHEx_Advanced_Option_Type***
- ***uint32_t FLASH_AdvOBProgramInitTypeDef::PCROPState***
  PCROP activation or deactivation. This parameter can be a value of ***FLASHEx_PCROP_State***
- ***uint32_t FLASH_AdvOBProgramInitTypeDef::Banks***
  Select banks for PCROP activation/deactivation of all sectors. This parameter must be a value of ***FLASHEx_Banks***
- ***uint16_t FLASH_AdvOBProgramInitTypeDef::SectorsBank1***
  Specifies the sector(s) set for PCROP for Bank1. This parameter can be a value of ***FLASHEx_Option_Bytes_PC_ReadWrite_Protection***
- ***uint16_t FLASH_AdvOBProgramInitTypeDef::SectorsBank2***
  Specifies the sector(s) set for PCROP for Bank2. This parameter can be a value of ***FLASHEx_Option_Bytes_PC_ReadWrite_Protection***
- ***uint8_t FLASH_AdvOBProgramInitTypeDef::BootConfig***
  Specifies Option bytes for boot config. This parameter can be a value of ***FLASHEx_Dual_Boot***

27.2 FLASHEx Firmware driver API description

The following section lists the various functions of the FLASHEx library.

27.2.1 Flash Extension features

Comparing to other previous devices, the FLASH interface for STM32F427xx/437xx and STM32F429xx/439xx devices contains the following additional features

- Capacity up to 2 Mbyte with dual bank architecture supporting read-while-write capability (RWW)
- Dual bank memory organization
- PCROP protection for all banks

27.2.2 How to use this driver

This driver provides functions to configure and program the FLASH memory of all STM32F427xx/437xx, STM32F429xx/439xx, STM32F469xx/479xx and STM32F446xx devices. It includes

1. FLASH Memory Erase functions:
   - Lock and Unlock the FLASH interface using HAL_FLASH_Unlock() and HAL_FLASH_Lock() functions
   - Erase function: Erase sector, erase all sectors
   - There are two modes of erase :
     - Polling Mode using HAL_FLASHEx_Erase()
     - Interrupt Mode using HAL_FLASHEx_Erase_IT()

2. Option Bytes Programming functions: Use HAL_FLASHEx_OBProgram() to :
   - Set/Reset the write protection
   - Set the Read protection Level
   - Set the BOR level
   - Program the user Option Bytes
3. Advanced Option Bytes Programming functions: Use HAL_FLASHEx_AdvOBProgram() to :
   - Extended space (bank 2) erase function
   - Full FLASH space (2 Mo) erase (bank 1 and bank 2)
   - Dual Boot activation
   - Write protection configuration for bank 2
   - PCROP protection configuration and control for both banks

27.2.3 Extended programming operation functions

This subsection provides a set of functions allowing to manage the Extension FLASH programming operations.
This section contains the following APIs:

- *HAL_FLASHEx_Erase()*
- *HAL_FLASHEx_Erase_IT()*
- *HAL_FLASHEx_OBProgram()*
- *HAL_FLASHEx_OBGetConfig()*
- *HAL_FLASHEx_AdvOBProgram()*
- *HAL_FLASHEx_AdvOBGetConfig()*
- *HAL_FLASHEx_OB_SelectPCROP()*
- *HAL_FLASHEx_OB_DeSelectPCROP()*
- *HAL_FLASHEx_OB_GetBank2WRP()*

27.2.4 Detailed description of functions

HAL_FLASHEx_Erase

Function name

HAL_StatusTypeDef HAL_FLASHEx_Erase (FLASH_EraseInitTypeDef * pEraseInit, uint32_t * SectorError)

Function description

Perform a mass erase or erase the specified FLASH memory sectors.

Parameters

- **pEraseInit:** pointer to an FLASH_EraseInitTypeDef structure that contains the configuration information for the erasing.
- **SectorError:** pointer to variable that contains the configuration information on faulty sector in case of error (0xFFFFFFFFU means that all the sectors have been correctly erased)

Return values

- **HAL:** Status

HAL_FLASHEx_Erase_IT

Function name

HAL_StatusTypeDef HAL_FLASHEx_Erase_IT (FLASH_EraseInitTypeDef * pEraseInit)

Function description

Perform a mass erase or erase the specified FLASH memory sectors with interrupt enabled.

Parameters

- **pEraseInit:** pointer to an FLASH_EraseInitTypeDef structure that contains the configuration information for the erasing.

Return values

- **HAL:** Status

HAL_FLASHEx_OBProgram

Function name

HAL_StatusTypeDef HAL_FLASHEx_OBProgram (FLASH_OBProgramInitTypeDef * pOBInit)

Function description

Program option bytes.

Parameters

- **pOBInit:** pointer to an FLASH_OBInitStruct structure that contains the configuration information for the programming.

Return values

- **HAL:** Status

HAL_FLASHEx_OBGetConfig

Function name

void HAL_FLASHEx_OBGetConfig (FLASH_OBProgramInitTypeDef * pOBInit)

Function description

Get the Option byte configuration.

Parameters

- **pOBInit:** pointer to an FLASH_OBInitStruct structure that contains the configuration information for the programming.

Return values

- **None:**

HAL_FLASHEx_AdvOBProgram

Function name

HAL_StatusTypeDef HAL_FLASHEx_AdvOBProgram (FLASH_AdvOBProgramInitTypeDef * pAdvOBInit)

Function description

Program option bytes.

Parameters

- **pAdvOBInit:** pointer to an FLASH_AdvOBProgramInitTypeDef structure that contains the configuration information for the programming.

Return values

- **HAL:** Status

HAL_FLASHEx_AdvOBGetConfig

Function name

void HAL_FLASHEx_AdvOBGetConfig (FLASH_AdvOBProgramInitTypeDef * pAdvOBInit)

Function description

Get the OBEX byte configuration.

Parameters

- **pAdvOBInit:** pointer to an FLASH_AdvOBProgramInitTypeDef structure that contains the configuration information for the programming.

Return values

- **None:**

HAL_FLASHEx_OB_SelectPCROP

Function name

HAL_StatusTypeDef HAL_FLASHEx_OB_SelectPCROP (void)

Function description

Select the Protection Mode.

Return values

- **HAL:** Status

Notes

- After PCROP activated Option Byte modification NOT POSSIBLE! excepted Global Read Out Protection modification (from level1 to level0)
- Once SPRMOD bit is active unprotection of a protected sector is not possible
- Read a protected sector will set RDERR Flag and write a protected sector will set WRPERR Flag
- This function can be used only for STM32F42xxx/STM32F43xxx/STM32F401xx/STM32F411xx/ STM32F446xx/ STM32F469xx/STM32F479xx/STM32F412xx/STM32F413xx devices.

HAL_FLASHEx_OB_DeSelectPCROP

Function name

HAL_StatusTypeDef HAL_FLASHEx_OB_DeSelectPCROP (void)

Function description

Deselect the Protection Mode.

Return values

- **HAL:** Status

Notes

- After PCROP activated Option Byte modification NOT POSSIBLE! excepted Global Read Out Protection modification (from level1 to level0)
- Once SPRMOD bit is active unprotection of a protected sector is not possible
- Read a protected sector will set RDERR Flag and write a protected sector will set WRPERR Flag
- This function can be used only for STM32F42xxx/STM32F43xxx/STM32F401xx/STM32F411xx/ STM32F446xx/ STM32F469xx/STM32F479xx/STM32F412xx/STM32F413xx devices.

HAL_FLASHEx_OB_GetBank2WRP

Function name

uint16_t HAL_FLASHEx_OB_GetBank2WRP (void)

Function description

Returns the FLASH Write Protection Option Bytes value for Bank 2.

Return values

- **The:** FLASH Write Protection Option Bytes value

Notes

- This function can be used only for STM32F42xxx/STM32F43xxx/STM32F469xx/STM32F479xx devices.

FLASH_Erase_Sector

Function name

void FLASH_Erase_Sector (uint32_t Sector, uint8_t VoltageRange)

Function description

Erase the specified FLASH memory sector.

Parameters

- **Sector:** FLASH sector to erase The value of this parameter depend on device used within the same series
- **VoltageRange:** The device voltage range which defines the erase parallelism. This parameter can be one of the following values:
  - FLASH_VOLTAGE_RANGE_1: when the device voltage range is 1.8V to 2.1V, the operation will be done by byte (8-bit)
  - FLASH_VOLTAGE_RANGE_2: when the device voltage range is 2.1V to 2.7V, the operation will be done by half word (16-bit)
  - FLASH_VOLTAGE_RANGE_3: when the device voltage range is 2.7V to 3.6V, the operation will be done by word (32-bit)
  - FLASH_VOLTAGE_RANGE_4: when the device voltage range is 2.7V to 3.6V + External Vpp, the operation will be done by double word (64-bit)

Return values

- **None:**

FLASH_FlushCaches

Function name

void FLASH_FlushCaches (void)

Function description

Flush the instruction and data caches.

Return values

- **None:**

27.3 FLASHEx Firmware driver defines

The following section lists the various define and macros of the module.

27.3.1 FLASHEx

FLASHEx

FLASH Advanced Option Type

OPTIONBYTE_PCROP

PCROP option byte configuration

OPTIONBYTE_BOOTCONFIG

BOOTConfig option byte configuration

FLASH Banks

FLASH_BANK_1

Bank 1

FLASH_BANK_2

Bank 2

FLASH_BANK_BOTH

Bank1 and Bank2

FLASH BOR Reset Level

OB_BOR_LEVEL3

Supply voltage ranges from 2.70 to 3.60 V

OB_BOR_LEVEL2

Supply voltage ranges from 2.40 to 2.70 V

OB_BOR_LEVEL1

Supply voltage ranges from 2.10 to 2.40 V

OB_BOR_OFF

Supply voltage ranges from 1.62 to 2.10 V

FLASH Dual Boot

OB_DUAL_BOOT_ENABLE

Dual Bank Boot Enable

OB_DUAL_BOOT_DISABLE

Dual Bank Boot Disable, always boot on User Flash

FLASH Private macros to check input parameters

IS_FLASH_TYPEERASE

IS_VOLTAGERANGE

IS_WRPSTATE

IS_OPTIONBYTE

IS_OB_RDP_LEVEL

IS_OB_IWDG_SOURCE

IS_OB_STOP_SOURCE

IS_OB_STDBY_SOURCE

IS_OB_BOR_LEVEL

IS_PCROPSTATE

IS_OBEX

IS_FLASH_LATENCY

IS_FLASH_BANK

IS_FLASH_SECTOR

IS_FLASH_ADDRESS

IS_FLASH_NBSECTORS

IS_OB_WRP_SECTOR

IS_OB_PCROP

IS_OB_BOOT

IS_OB_PCROP_SELECT

FLASH Mass Erase bit

FLASH_MER_BIT

2 MER bits here to clear

FLASH Option Bytes IWatchdog

OB_IWDG_SW

Software IWDG selected

OB_IWDG_HW

Hardware IWDG selected

FLASH Option Bytes nRST_STDBY

OB_STDBY_NO_RST

No reset generated when entering in STANDBY

OB_STDBY_RST

Reset generated when entering in STANDBY

FLASH Option Bytes nRST_STOP

OB_STOP_NO_RST

No reset generated when entering in STOP

OB_STOP_RST

Reset generated when entering in STOP

FLASH Option Bytes PC ReadWrite Protection

OB_PCROP_SECTOR_0

PC Read/Write protection of Sector0

OB_PCROP_SECTOR_1

PC Read/Write protection of Sector1

OB_PCROP_SECTOR_2

PC Read/Write protection of Sector2

OB_PCROP_SECTOR_3

PC Read/Write protection of Sector3

OB_PCROP_SECTOR_4

PC Read/Write protection of Sector4

OB_PCROP_SECTOR_5

PC Read/Write protection of Sector5

OB_PCROP_SECTOR_6

PC Read/Write protection of Sector6

OB_PCROP_SECTOR_7

PC Read/Write protection of Sector7

OB_PCROP_SECTOR_8

PC Read/Write protection of Sector8

OB_PCROP_SECTOR_9

PC Read/Write protection of Sector9

OB_PCROP_SECTOR_10

PC Read/Write protection of Sector10

OB_PCROP_SECTOR_11

PC Read/Write protection of Sector11

OB_PCROP_SECTOR_12

PC Read/Write protection of Sector12

OB_PCROP_SECTOR_13

PC Read/Write protection of Sector13

OB_PCROP_SECTOR_14

PC Read/Write protection of Sector14

OB_PCROP_SECTOR_15

PC Read/Write protection of Sector15

OB_PCROP_SECTOR_16

PC Read/Write protection of Sector16

OB_PCROP_SECTOR_17

PC Read/Write protection of Sector17

OB_PCROP_SECTOR_18

PC Read/Write protection of Sector18

OB_PCROP_SECTOR_19

PC Read/Write protection of Sector19

OB_PCROP_SECTOR_20

PC Read/Write protection of Sector20

OB_PCROP_SECTOR_21

PC Read/Write protection of Sector21

OB_PCROP_SECTOR_22

PC Read/Write protection of Sector22

OB_PCROP_SECTOR_23

PC Read/Write protection of Sector23

OB_PCROP_SECTOR_All

PC Read/Write protection of all Sectors

FLASH Option Bytes Read Protection

OB_RDP_LEVEL_0

OB_RDP_LEVEL_1

OB_RDP_LEVEL_2

Warning: When enabling read protection level 2 it s no more possible to go back to level 1 or 0

FLASH Option Bytes Write Protection

OB_WRP_SECTOR_0

Write protection of Sector0

OB_WRP_SECTOR_1

Write protection of Sector1

OB_WRP_SECTOR_2

Write protection of Sector2

OB_WRP_SECTOR_3

Write protection of Sector3

OB_WRP_SECTOR_4

Write protection of Sector4

OB_WRP_SECTOR_5

Write protection of Sector5

OB_WRP_SECTOR_6

Write protection of Sector6

OB_WRP_SECTOR_7

Write protection of Sector7

OB_WRP_SECTOR_8

Write protection of Sector8

OB_WRP_SECTOR_9

Write protection of Sector9

OB_WRP_SECTOR_10

Write protection of Sector10

OB_WRP_SECTOR_11

Write protection of Sector11

OB_WRP_SECTOR_12

Write protection of Sector12

OB_WRP_SECTOR_13

Write protection of Sector13

OB_WRP_SECTOR_14

Write protection of Sector14

OB_WRP_SECTOR_15

Write protection of Sector15

OB_WRP_SECTOR_16

Write protection of Sector16

OB_WRP_SECTOR_17

Write protection of Sector17

OB_WRP_SECTOR_18

Write protection of Sector18

OB_WRP_SECTOR_19

Write protection of Sector19

OB_WRP_SECTOR_20

Write protection of Sector20

OB_WRP_SECTOR_21

Write protection of Sector21

OB_WRP_SECTOR_22

Write protection of Sector22

OB_WRP_SECTOR_23

Write protection of Sector23

OB_WRP_SECTOR_All

Write protection of all Sectors

FLASH Option Type

OPTIONBYTE_WRP

WRP option byte configuration

OPTIONBYTE_RDP

RDP option byte configuration

OPTIONBYTE_USER

USER option byte configuration

OPTIONBYTE_BOR

BOR option byte configuration

FLASH PCROP State

OB_PCROP_STATE_DISABLE

Disable PCROP

OB_PCROP_STATE_ENABLE

Enable PCROP

FLASH Sectors

FLASH_SECTOR_0

Sector Number 0

FLASH_SECTOR_1

Sector Number 1

FLASH_SECTOR_2

Sector Number 2

FLASH_SECTOR_3

Sector Number 3

FLASH_SECTOR_4

Sector Number 4

FLASH_SECTOR_5

Sector Number 5

FLASH_SECTOR_6

Sector Number 6

FLASH_SECTOR_7

Sector Number 7

FLASH_SECTOR_8

Sector Number 8

FLASH_SECTOR_9

Sector Number 9

FLASH_SECTOR_10

Sector Number 10

FLASH_SECTOR_11

Sector Number 11

FLASH_SECTOR_12

Sector Number 12

FLASH_SECTOR_13

Sector Number 13

FLASH_SECTOR_14

Sector Number 14

FLASH_SECTOR_15

Sector Number 15

FLASH_SECTOR_16

Sector Number 16

FLASH_SECTOR_17

Sector Number 17

FLASH_SECTOR_18

Sector Number 18

FLASH_SECTOR_19

Sector Number 19

FLASH_SECTOR_20

Sector Number 20

FLASH_SECTOR_21

Sector Number 21

FLASH_SECTOR_22

Sector Number 22

FLASH_SECTOR_23

Sector Number 23

FLASH Selection Protection Mode

OB_PCROP_DESELECTED

Disabled PcROP, nWPRi bits used for Write Protection on sector i

OB_PCROP_SELECTED

Enable PcROP, nWPRi bits used for PCRoP Protection on sector i

FLASH Type Erase

FLASH_TYPEERASE_SECTORS

Sectors erase only

FLASH_TYPEERASE_MASSERASE

Flash Mass erase activation

FLASH Voltage Range

FLASH_VOLTAGE_RANGE_1

Device operating range: 1.8V to 2.1V

FLASH_VOLTAGE_RANGE_2

Device operating range: 2.1V to 2.7V

FLASH_VOLTAGE_RANGE_3

Device operating range: 2.7V to 3.6V

FLASH_VOLTAGE_RANGE_4

Device operating range: 2.7V to 3.6V + External Vpp

FLASH WRP State

OB_WRPSTATE_DISABLE

Disable the write protection of the desired bank 1 sectors

OB_WRPSTATE_ENABLE

Enable the write protection of the desired bank 1 sectors