# MCAL Configuration Verification Manual for FlsLoader

## 32-bit TriCore™ AURIX™ TC3xx microcontroller family

## About this document

### Scope and purpose

This Configuration Data Reference document is applicable to all TC3xx devices in the TriCore™ AURIX™ family of 32-bit microcontrollers.

The purpose of this document is to facilitate the integrator to verify the generated code based on the input configuration parameters. This document describes details of structures, defines, macros and variables generated from the configuration parameters.

### Intended audience

This document is intended for integrators who need to understand the logic of the generated configuration code.

### Reference documents

This document should be read in conjunction with the following documents:

- AURIX™ TC3xx MCAL User Manual FlsLoader

# Table of contents

Table of contents

Table of contents

# 1 FlsLoader driver

This chapter describes the details of the configuration data generated from the FLSLOADER driver.

## 1.1 File: FlsLoader_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in 'inc' folder.

### 1.1.1 Macro: FLSLOADER_AR_RELEASE_MAJOR_VERSION

**Table 1 FLSLOADER_AR_RELEASE_MAJOR_VERSION**

| | | |
|---|---|---|
| **Name** | FLSLOADER_AR_RELEASE_MAJOR_VERSION | |
| **Description** | Major version number of AUTOSAR release on which the FlsLoader implementation is based on. | |
| **Verification method** | The macro is generated as 4.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Generate FlsLoader_Cfg.h file | `#define FLSLOADER_AR_RELEASE_MAJOR_VERSION (4U)` |

### 1.1.2 Macro: FLSLOADER_ AR_RELEASE_MINOR_VERSION

**Table 2 FLSLOADER_ AR_RELEASE_MINOR_VERSION**

| | | |
|---|---|---|
| **Name** | FLSLOADER_ AR_RELEASE_MINOR_VERSION | |
| **Description** | Minor version number of AUTOSAR release on which the FlsLoader implementation is based on. | |
| **Verification method** | The macro is generated as 2.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Generate FlsLoader_Cfg.h file | `#define FLSLOADER_AR_RELEASE_MINOR_VERSION (2U)` |

### 1.1.3 Macro: FLSLOADER_ AR_RELEASE_REVISION_VERSION

**Table 3 FLSLOADER_ AR_RELEASE_REVISION_VERSION**

| | |
|---|---|
| **Name** | FLSLOADER_AR_RELEASE_REVISION_VERSION |
| **Description** | Revision version number of AUTOSAR release on which the FlsLoader implementation is based on. |

| | | |
|---|---|---|
| **Verification method** | The macro is generated as 2.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Generate FlsLoader_Cfg.h file | `#define FLSLOADER_AR_RELEASE_REVISION_VERSION (2U)` |

### 1.1.4 Macro: FLSLOADER_SW_MAJOR_VERSION

**Table 4 FLSLOADER_SW_MAJOR_VERSION**

| | | |
|---|---|---|
| **Name** | FLSLOADER_SW_MAJOR_VERSION | |
| **Description** | Major version number of the FlsLoader module. | |
| **Verification method** | The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion'.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Generate FlsLoader_Cfg.h file with SwMajorVersion 10 | `#define FLSLOADER_SW_MAJOR_VERSION (10U)` |

### 1.1.5 Macro: FLSLOADER_ SW_MINOR_VERSION

**Table 5 FLSLOADER_ SW_MINOR_VERSION**

| | | |
|---|---|---|
| **Name** | FLSLOADER_ SW_MINOR_VERSION | |
| **Description** | Minor version number of the FlsLoader module. | |
| **Verification method** | The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion'.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Generate FlsLoader_Cfg.h file with SwMinorVersion 10 | `#define FLSLOADER_SW_MINOR_VERSION (10U)` |

### 1.1.6 Macro: FLSLOADER_ SW_PATCH_VERSION

**Table 6 FLSLOADER_ SW_PATCH_VERSION**

| | |
|---|---|
| **Name** | FLSLOADER_SW_PATCH_VERSION |
| **Description** | Patch level version number of the FlsLoader module. |

| | | |
|---|---|---|
| **Verification method** | The macro is generated with the value present in 'CommonPublishedInformation/SwPatchVersion'.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Generate FlsLoader_Cfg.h file with SwPatchVersion 0 | `#define FLSLOADER_SW_PATCH_VERSION (0U)` |

### 1.1.7 Macro: FLSLOADER_DEV_ERROR_DETECT

**Table 7 FLSLOADER_DEV_ERROR_DETECT**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DEV_ERROR_DETECT | |
| **Description** | Enables/disables the Development Error Detection. | |
| **Verification method** | The macro is generated as STD_ON if FlsLoaderDevErrorDetect configuration parameter is set to 'True' else the macro is generated as STD_OFF. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDevErrorDetect = True | `#define FLSLOADER_DEV_ERROR_DETECT (STD_ON)` |
| | FlsLoaderDevErrorDetect = False | `#define FLSLOADER_DEV_ERROR_DETECT (STD_OFF)` |

### 1.1.8 Macro: FLSLOADER_DEINIT_API

**Table 8 FLSLOADER_DEINIT_API**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DEINIT_API | |
| **Description** | Enables/disables FlsLoader_DeInit API. | |
| **Verification method** | The macro is generated as STD_ON if FlsLoaderDeInitApi configuration parameter is set to 'True' else the macro is generated as STD_OFF. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDeInitApi = True | `#define FLSLOADER_DEINIT_API (STD_ON)` |
| | FlsLoaderDeInitApi = False | `#define FLSLOADER_DEINIT_API (STD_OFF)` |

### 1.1.9 Macro: FLSLOADER _LOCK_UNLOCK_API

**Table 9 FLSLOADER_LOCK_UNLOCK_API**

| | |
|---|---|
| **Name** | FLSLOADER_LOCK_UNLOCK_API |
| **Description** | Enables/disables FlsLoader_Lock and FlsLoaderUnLock APIs. |
| **Verification method** | The macro is generated as STD_ON if FlsLoaderLockUnlockApi configuration parameter is set to 'True' else the macro is generated as STD_OFF. |

| | | |
|---|---|---|
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderLockUnlockApi = True | `#define FLSLOADER_LOCK_UNLOCK_API (STD_ON)` |
| | FlsLoaderLockUnlockApi = False | `#define FLSLOADER_LOCK_UNLOCK_API (STD_OFF)` |

### 1.1.10 Macro: FLSLOADER _VERSION_INFO_API

**Table 10 FLSLOADER_VERSION_INFO_API**

| | | |
|---|---|---|
| **Name** | FLSLOADER_VERSION_INFO_API | |
| **Description** | Enables/disables FlsLoader_GetVersionInfo API. | |
| **Verification method** | The macro is generated as STD_ON if FlsLoaderVersionInfoApi configuration parameter is set to 'True' else the macro is generated as STD_OFF. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderVersionInfoApi = True | `#define FLSLOADER_VERSION_INFO_API (STD_ON)` |
| | FlsLoaderVersionInfoApi = False | `#define FLSLOADER_VERSION_INFO_API (STD_OFF)` |

### 1.1.11 Macro: FLSLOADER_ CALLOUT_FUNC

**Table 11 FLSLOADER_ CALLOUT_FUNC**

| | | |
|---|---|---|
| **Name** | FLSLOADER_CALLOUT_FUNC | |
| **Description** | Enables/disables call-out functionality | |
| **Verification method** | The macro is generated as STD_OFF if FlsLoaderCallOutFunction configuration parameter is set to 'NULL_PTR' else the macro is generated as STD_ON. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderCallOutFunction = NULL_PTR | `#define FLSLOADER_CALLOUT_FUNC (STD_OFF)` |
| | FlsLoaderCallOutFunction = FlsLoader_LoopCallOut | `#define FLSLOADER_CALLOUT_FUNC (STD_ON)` |
| | FlsLoaderCallOutFunction = 0x80005478 | `#define FLSLOADER_CALLOUT_FUNC (STD_ON)` |

### 1.1.12 Macro: FLSLOADER_CALLOUT_TIME

**Table 12 FLSLOADER_CALLOUT_TIME**

| | |
|---|---|
| **Name** | FLSLOADER_CALLOUT_TIME |
| **Description** | Specifies the time rate in nanoseconds at which the configured callout function is invoked periodically by the driver during write and erases operations. |
| **Verification method** | The macro is assigned with the numeric value configured in configuration parameter |

| | | |
|---|---|---|
| | FlsLoaderCallOutTime. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderCallOutTime = 10000 | `#define FLSLOADER_CALLOUT_TIME (10000U)` |
| | FlsLoaderCallOutTime = 5000000 | `#define FLSLOADER_CALLOUT_TIME (5000000U)` |

### 1.1.13 Macro: FLSLOADER_ ENABLE_LOCKCHECK

**Table 13 FLSLOADER_ ENABLE_LOCKCHECK**

| | | |
|---|---|---|
| **Name** | FLSLOADER_ENABLE_LOCKCHECK | |
| **Description** | Enables/disables lock-check functionality. | |
| **Verification method** | The macro is generated as STD_ON if FlsLoaderEnableLockCheck configuration parameter is set to 'True' else the macro is generated as STD_OFF. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderEnableLockCheck = True | `#define FLSLOADER_ENABLE_LOCKCHECK (STD_ON)` |
| | FlsLoaderEnableLockCheck = False | `#define FLSLOADER_ENABLE_LOCKCHECK (STD_OFF)` |

### 1.1.14 Macro: FLSLOADER_ DF0_PROT

**Table 14 FLSLOADER_ DF0_PROT**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PROT | |
| **Description** | Specifies the protection set for DFlash0 at bank level. | |
| **Verification method** | The macro is generated with the protection value set by configuration parameter FlsLoaderDF0Prot.<br><br>*Note: Supported protection values for DFlash0 are NO_PROTECTION, WRITE_PROTECTION and READ_PROTECTION* | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = NO_PROTECTION | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)` |
| | FlsLoaderDF0Prot = WRITE_PROTECTION | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)` |

### 1.1.15 Macro: FLSLOADER_DF0_PW00

**Table 15 FLSLOADER_DF0_PW00**

| | |
|---|---|
| **Name** | FLSLOADER_DF0_PW00 |

| | | |
|---|---|---|
| **Description** | First 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW0_0 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW0_0 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW00 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW0_0 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW00 ((uint32)0x0U)` |

### 1.1.16 Macro: FLSLOADER_DF0_PW01

**Table 16 FLSLOADER_DF0_PW01**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PW01 | |
| **Description** | Second 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW0_1 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW0_1 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW01 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW0_1 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW01 ((uint32)0x0U)` |

### 1.1.17 Macro: FLSLOADER_DF0_PW10

**Table 17 FLSLOADER_DF0_PW10**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PW10 | |
| **Description** | Third 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW1_0 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = | `#define FLSLOADER_DF0_PROT` |

| | | |
|---|---|---|
| | WRITE_PROTECTION<br>FlsLoaderDF0UcbPW1_0 = 2857740885 | `(WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW10 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW1_0 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW10 ((uint32)0x0U)` |

### 1.1.18 Macro: FLSLOADER_DF0_PW11

**Table 18 FLSLOADER_DF0_PW11**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PW11 | |
| **Description** | Fourth 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW1_1 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW1_1 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW11 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW1_1 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW11 ((uint32)0x0U)` |

### 1.1.19 Macro: FLSLOADER_DF0_PW20

**Table 19 FLSLOADER_DF0_PW20**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PW20 | |
| **Description** | Fifth 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW2_0 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW2_0 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW20 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW2_0 = | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW20` |

| | | |
|---|---|---|
| | 1432769894 | ((uint32)0x0U) |

## 1.1.20 Macro: FLSLOADER_DF0_PW21

**Table 20 FLSLOADER_DF0_PW21**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PW21 | |
| **Description** | Sixth 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW2_1 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW2_1 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW21 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW2_1 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW21 ((uint32)0x0U)` |

## 1.1.21 Macro: FLSLOADER_DF0_PW30

**Table 21 FLSLOADER_DF0_PW30**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_PW30 | |
| **Description** | Seventh 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW3_0 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW3_0 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW30 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW3_0 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW30 ((uint32)0x0U)` |

## 1.1.22 Macro: FLSLOADER_DF0_PW31

**Table 22 FLSLOADER_DF0_PW31**

| | |
|---|---|
| **Name** | FLSLOADER_DF0_PW31 |

| | | |
|---|---|---|
| **Description** | Eighth 32-bit word of 256-bit password for setting DFlash0 protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderDF0UcbPW3_1 if active DFlash0 protection (READ_PROTECTION, WRITE_PROTECTION) is enabled using FlsLoaderDF0Prot, else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = WRITE_PROTECTION<br>FlsLoaderDF0UcbPW3_1 = 2857740885 | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_DF0_PW31 ((uint32)0xAA55AA55U)` |
| | FlsLoaderDF0Prot = NO_PROTECTION<br>FlsLoaderDF0UcbPW3_1 = 1432769894 | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_DF0_PW31 ((uint32)0x0U)` |

### 1.1.23 Macro: FLSLOADER_PF0_PW00

**Table 23 FLSLOADER_PF0_PW00**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW00 | |
| **Description** | First 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW0_0 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW0_0 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW00 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW0_0 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW00 ((uint32)0x0U)` |

### 1.1.24 Macro: FLSLOADER_PF0_PW01

**Table 24 FLSLOADER_PF0_PW01**

| | |
|---|---|
| **Name** | FLSLOADER_PF0_PW01 |
| **Description** | Second 32-bit word of 256-bit password for setting PFlash protection. |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW0_1 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the |

| | | |
|---|---|---|
| | selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW0_1 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW01 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW0_1 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW01 ((uint32)0x0U)` |

### 1.1.25 Macro: FLSLOADER_PF0_PW10

**Table 25 FLSLOADER_PF0_PW10**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW10 | |
| **Description** | Third 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW1_0 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW1_0 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW10 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW1_0 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW10 ((uint32)0x0U)` |

### 1.1.26 Macro: FLSLOADER_PF0_PW11

**Table 26 FLSLOADER_PF0_PW11**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW11 | |
| **Description** | Fourth 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW1_1 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |

| | | |
|---|---|---|
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW1_1 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW11 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW1_1 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW11 ((uint32)0x0U)` |

### 1.1.27 Macro: FLSLOADER_PF0_PW20

**Table 27 FLSLOADER_PF0_PW20**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW20 | |
| **Description** | Fifth 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW2_0 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW2_0 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW20 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW2_0 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW20 ((uint32)0x0U)` |

### 1.1.28 Macro: FLSLOADER_PF0_PW21

**Table 28 FLSLOADER_PF0_PW21**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW21 | |
| **Description** | Sixth 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW2_1 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)` |

| | | |
|---|---|---|
| | FlsLoaderPFUcbPW2_1 = 2857740885 | `#define FLSLOADER_PF0_PW21 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW2_1 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW21 ((uint32)0x0U)` |

### 1.1.29 Macro: FLSLOADER_PF0_PW30

**Table 29 FLSLOADER_PF0_PW30**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW30 | |
| **Description** | Seventh 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW3_0 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW3_0 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW30 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = NO_PROTECTION<br>FlsLoaderPFUcbPW3_0 = 1432769894 | `#define FLSLOADER_PF2_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF0_PW30 ((uint32)0x0U)` |

### 1.1.30 Macro: FLSLOADER_PF0_PW31

**Table 30 FLSLOADER_PF0_PW31**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF0_PW31 | |
| **Description** | Eighth 32-bit word of 256-bit password for setting PFlash protection. | |
| **Verification method** | The macro is generated as hex converted numeric value set in the configuration parameter FlsLoaderPFUcbPW3_1 if active protection (WRITE_PROTECTION, WOP_PROTECTION, OTP_PROTECTION) is enabled for any of the PFlash bank using FlsLoaderPFLash<x>WriteProt(x=0 to n-1, n = Number of PFlash banks available in the selected device), else it is generated as 0x0. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderPFLash2WriteProt = WRITE_PROTECTION<br>FlsLoaderPFUcbPW3_1 = 2857740885 | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF0_PW31 ((uint32)0xAA55AA55U)` |
| | FlsLoaderPFLash2WriteProt = | `#define FLSLOADER_PF2_PROT` |

| | | |
|---|---|---|
| | NO_PROTECTION<br>FlsLoaderPFUcbPW3_1 = 1432769894 | (NO_PROTECTION)<br>#define FLSLOADER_PF0_PW31 ((uint32)0x0U) |

### 1.1.31 Macro: FLSLOADER_DFLASH<x>_START_ADDRESS

**Table 31 FLSLOADER_DFLASH<x>_START_ADDRESS**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DFLASH<x>_START_ADDRESS (x = 0 and 1) | |
| **Description** | Specifies the start address of DFlash bank 'x' in the selected device. | |
| **Verification method** | The macro is generated as hex value specifying the start address of DFlash bank 'x' in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC375:<br>DFlash0 start address is 0xAF000000,<br>DFlash1 start address is 0xAFC00000 | #define FLSLOADER_DFLASH0_START_ADDRESS ((FlsLoader_AddressType)(0xaf000000U))<br>#define FLSLOADER_DFLASH1_START_ADDRESS ((FlsLoader_AddressType)(0xafc00000U)) |

### 1.1.32 Macro: FLSLOADER_DFLASH<x>_END

**Table 32 FLSLOADER_DFLASH<x>_END**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DFLASH<x>_END (x = 0 and 1) | |
| **Description** | Specifies the end address of DFlash bank 'x'. | |
| **Verification method** | The macro is generated as hex value specifying the end address of DFlash bank 'x' in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC375:<br>DFlash0 end address is 0xAF0FFFFF,<br>DFlash1 end address is 0xAFC1FFFF | #define FLSLOADER_DFLASH0_END ((FlsLoader_AddressType)(0xaf0fffffU))<br>#define FLSLOADER_DFLASH1_END ((FlsLoader_AddressType)(0xafc1ffffU)) |

### 1.1.33 Macro: FLSLOADER_DFLASH<x>_BANK_SIZE

**Table 33 FLSLOADER_DFLASH<x>_BANK_SIZE**

| | |
|---|---|
| **Name** | FLSLOADER_DFLASH<x>_BANK_SIZE (x = 0 and 1) |

| | | |
|---|---|---|
| **Description** | Specifies size of DFlash bank ‘x’ in the selected device in bytes. | |
| **Verification method** | The macro is generated as hex value specifying size of DFlash bank ‘x’ in bytes in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC375:<br>DFlash0 size is 256KB,<br>DFlash1 size is 128KB | `#define FLSLOADER_DFLASH0_BANK_SIZE (0x40000U)`<br>`#define FLSLOADER_DFLASH1_BANK_SIZE (0x20000U)` |

## 1.1.34 Macro: FLSLOADER_NUM_OF_DF<x>_SECTORS

**Table 34 FLSLOADER_NUM_OF_DF<x>_SECTORS**

| | | |
|---|---|---|
| **Name** | FLSLOADER_NUM_OF_DF<x>_SECTORS (x = 0 and 1) | |
| **Description** | Specifies the number of sectors present in DFlash bank ‘x’ in the selected device. | |
| **Verification method** | The macro is generated as numeric value specifying the number of sectors present in DFlash bank ‘x’ in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC375:<br>DFlash0 has 64 sectors,<br>DFlash1 has 32 sectors | `#define FLSLOADER_NUM_OF_DF0_SECTORS (64U)`<br>`#define FLSLOADER_NUM_OF_DF1_SECTORS (32U)` |

## 1.1.35 Macro: FLSLOADER_PFLASH<x>_START_ADDRESS

**Table 35 FLSLOADER_PFLASH<x>_START_ADDRESS**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PFLASH<x>_START_ADDRESS (x ranges from 0 to n-1, n= Number of PFlash banks available in the selected device) | |
| **Description** | Specifies the start address of PFlash bank ‘x’ in the selected device. | |
| **Verification method** | The macro is generated as hex value specifying the start address of PFlash bank ‘x’ in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC399:<br>PFlash0 start address is 0xA0000000,<br>PFlash1 start address is | `#define FLSLOADER_PFLASH0_START_ADDRESS ((uint32)(0xa0000000U))`<br>`#define FLSLOADER_PFLASH1_START_ADDRESS ((uint32)(0xa0300000U))` |

| | | |
|---|---|---|
| | 0xA0300000,<br>PFlash2 start address is 0xA0600000,<br>PFlash3 start address is 0xA0900000,<br>PFlash4 start address is 0xA0C00000,<br>PFlash5 start address is 0xA0F00000 | `#define FLSLOADER_PFLASH2_START_ADDRESS ((uint32)(0xa0600000U))`<br>`#define FLSLOADER_PFLASH3_START_ADDRESS ((uint32)(0xa0900000U))`<br>`#define FLSLOADER_PFLASH4_START_ADDRESS ((uint32)(0xa0c00000U))`<br>`#define FLSLOADER_PFLASH5_START_ADDRESS ((uint32)(0xa0f00000U))` |
| | If selected device is TC389:<br>PFlash0 start address is 0xA0000000,<br>PFlash1 start address is 0xA0300000,<br>PFlash2 start address is 0xA0600000,<br>PFlash3 start address is 0xA0900000 | `#define FLSLOADER_PFLASH0_START_ADDRESS ((uint32)(0xa0000000U))`<br>`#define FLSLOADER_PFLASH1_START_ADDRESS ((uint32)(0xa0300000U))`<br>`#define FLSLOADER_PFLASH2_START_ADDRESS ((uint32)(0xa0600000U))`<br>`#define FLSLOADER_PFLASH3_START_ADDRESS ((uint32)(0xa0900000U))` |

### 1.1.36 Macro: FLSLOADER_PFLASH<x>_END

**Table 36 FLSLOADER_PFLASH<x>_END**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PFLASH<x>_END (x ranges from 0 to n-1, n= Number of PFlash banks available in the selected device) | |
| **Description** | Specifies the end address of PFlash bank ‘x’ in the selected device. | |
| **Verification method** | The macro is generated as hex value specifying the end address of PFlash bank ‘x’ in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC399:<br>PFlash0 end address is 0xA02FFFFF,<br>PFlash1 end address is 0xA05FFFFF,<br>PFlash2 end address is 0xA08FFFFF,<br>PFlash3 end address is 0xA0BFFFFF,<br>PFlash4 end address is 0xA0EFFFFF,<br>PFlash5 end address is 0xA0FFFFFF | `#define FLSLOADER_PFLASH0_END ((FlsLoader_AddressType)(0xa02fffffU))`<br>`#define FLSLOADER_PFLASH1_END ((FlsLoader_AddressType)(0xa05fffffU))`<br>`#define FLSLOADER_PFLASH2_END ((FlsLoader_AddressType)(0xa08fffffU))`<br>`#define FLSLOADER_PFLASH3_END ((FlsLoader_AddressType)(0xa0bfffffU))`<br>`#define FLSLOADER_PFLASH4_END ((FlsLoader_AddressType)(0xa0efffffU))`<br>`#define FLSLOADER_PFLASH5_END ((FlsLoader_AddressType)(0xa0ffffffU))` |

| | | |
|---|---|---|
| | If selected device is TC389:<br>PFlash0 end address is 0xA02FFFFF,<br>PFlash1 end address is 0xA05FFFFF,<br>PFlash2 end address is 0xA08FFFFF,<br>PFlash3 end address is 0xA09FFFFF | `#define FLSLOADER_PFLASH0_END ((FlsLoader_AddressType)(0xa02fffffU))`<br>`#define FLSLOADER_PFLASH1_END ((FlsLoader_AddressType)(0xa05fffffU))`<br>`#define FLSLOADER_PFLASH2_END ((FlsLoader_AddressType)(0xa08fffffU))`<br>`#define FLSLOADER_PFLASH3_END ((FlsLoader_AddressType)(0xa09fffffU))` |

### 1.1.37 Macro: FLSLOADER_PFLASH<x>_SIZE

**Table 37 FLSLOADER_PFLASH<x>_SIZE**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PFLASH<x>_SIZE (x ranges from 0 to n-1, n= Number of PFlash banks available in the selected device) | |
| **Description** | Specifies size of PFlash bank ‘x’ in the selected device in bytes. | |
| **Verification method** | The macro is generated as hex value specifying size of PFlash bank ‘x’ in the selected device in bytes.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | If selected device is TC399:<br>PFlash0 size is 3MB,<br>PFlash1 size is 3MB<br>PFlash2 size is 3MB,<br>PFlash3 size is 3MB,<br>PFlash4 size is 3MB,<br>PFlash5 size is 1MB | `#define FLSLOADER_PFLASH0_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH1_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH2_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH3_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH4_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH5_SIZE (0x100000U)` |
| | If selected device is TC389:<br>PFlash0 size is 3MB,<br>PFlash1 size is 3MB,<br>PFlash2 size is 3MB,<br>PFlash3 size is 1MB | `#define FLSLOADER_PFLASH0_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH1_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH2_SIZE (0x300000U)`<br>`#define FLSLOADER_PFLASH3_SIZE (0x100000U)` |

### 1.1.38 Macro: FLSLOADER_NUM_OF_PF<x>_SECTORS

**Table 38 FLSLOADER_NUM_OF_PF<x>_SECTORS**

| | | |
|---|---|---|
| **Name** | FLSLOADER_NUM_OF_PF<x>_SECTORS (x ranges from 0 to n-1, n= Number of PFlash banks available in the selected device) | |
| **Description** | Specifies the number of sectors present in PFlash bank 'x' in the selected device. | |
| **Verification method** | The macro is generated as numeric value specifying the number of sectors present in PFlash bank 'x' in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | <u>If selected device is TC399:</u><br>PFlash0 has 192 sectors,<br>PFlash1 has 192 sectors<br>PFlash2 has 192 sectors,<br>PFlash3 has 192 sectors<br>PFlash4 has 192 sectors,<br>PFlash5 has 64 sectors | `#define FLSLOADER_NUM_OF_PF0_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF1_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF2_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF3_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF4_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF5_SECTORS (64U)` |
| | <u>If selected device is TC389:</u><br>PFlash0 has 192 sectors,<br>PFlash1 has 192 sectors<br>PFlash2 has 192 sectors,<br>PFlash3 has 64 sectors | `#define FLSLOADER_NUM_OF_PF0_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF1_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF2_SECTORS (192U)`<br>`#define FLSLOADER_NUM_OF_PF3_SECTORS (64U)` |

### 1.1.39 Macro: FLSLOADER_NUM_OF_DFLASH_BANK

**Table 39 FLSLOADER_NUM_OF_DFLASH_BANK**

| | |
|---|---|
| **Name** | FLSLOADER_NUM_OF_DFLASH_BANK |
| **Description** | Specifies the number of DFlash banks present in the selected device. |
| **Verification method** | The macro is generated as numeric value specifying the number of DFlash banks present in the selected device.<br><br>*Note: This macro is not configurable by the user. All devices of TC3xx family have 2 DFlash banks.* |

| Example(s) | Action | Generated output |
|---|---|---|
| | Device TC399 has 2 DFlash banks | `#define FLSLOADER_NUM_OF_DFLASH_BANK ((uint8)2U)` |

### 1.1.40 Macro: FLSLOADER_NUM_OF_PFLASH_BANK

**Table 40 FLSLOADER_NUM_OF_PFLASH_BANK**

| | | |
|---|---|---|
| **Name** | FLSLOADER_NUM_OF_PFLASH_BANK | |
| **Description** | Specifies the number of PFlash banks present in the selected device. | |
| **Verification method** | The macro is generated as numeric value specifying the number of PFlash banks present in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Selected device (TC399) has 6 PFlash banks | `#define FLSLOADER_NUM_OF_PFLASH_BANK ((uint8)6U)` |
| | Selected device (TC389) has 4 PFlash banks | `#define FLSLOADER_NUM_OF_PFLASH_BANK ((uint8)4U)` |

### 1.1.41 Macro: FLSLOADER_PF<x>_PROT

**Table 41 FLSLOADER_PF<x>_PROT**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF<x>_PROT (x ranges from 0 to 5) | |
| **Description** | Specifies the protection set for PFlash bank ‘x’ at bank level. | |
| **Verification method** | The macro is generated with the protection value set by the configuration parameter FlsLoaderPFLash<x>WriteProt for the respective PFlash bank ‘x’. If a device does not have PFlash bank ‘x’, respective FLSLOADER_PF<x>_PROT is generated with NO_PROTECTION.<br><br>*Note: Supported protection types for PFlash are NO_PROTECTION, WRITE_PROTECTION, WOP_PROTECTION and OTP_PROTECTION.* | |
| **Example(s)** | **Action** | **Generated output** |
| | For device TC399 with 6 PFlash banks:<br>FlsLoaderPFLash0WriteProt = NO_PROTECTION,<br>FlsLoaderPFLash1WriteProt = NO_PROTECTION,<br>FlsLoaderPFLash2WriteProt = WRITE_PROTECTION,<br>FlsLoaderPFLash3WriteProt = | `#define FLSLOADER_PF0_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF1_PROT (NO_PROTECTION)`<br>`#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF3_PROT (WRITE_PROTECTION)`<br>`#define FLSLOADER_PF4_PROT` |

| | | |
|---|---|---|
| | WRITE_PROTECTION,<br>FlsLoaderPFLash4WriteProt = OTP_PROTECTION,<br>FlsLoaderPFLash5WriteProt = WOP_PROTECTION | (OTP_PROTECTION)<br>#define FLSLOADER_PF5_PROT (WOP_PROTECTION) |
| | For device TC389 with 4 PFlash banks:<br>FlsLoaderPFLash0WriteProt = NO_PROTECTION,<br>FlsLoaderPFLash1WriteProt = NO_PROTECTION,<br>FlsLoaderPFLash2WriteProt = WRITE_PROTECTION,<br>FlsLoaderPFLash3WriteProt = WRITE_PROTECTION | #define FLSLOADER_PF0_PROT (NO_PROTECTION)<br>#define FLSLOADER_PF1_PROT (NO_PROTECTION)<br>#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)<br>#define FLSLOADER_PF3_PROT (WRITE_PROTECTION)<br>#define FLSLOADER_PF4_PROT (NO_PROTECTION)<br>#define FLSLOADER_PF5_PROT (NO_PROTECTION) |

### 1.1.42 Macro: FLSLOADER_PROCONP0<x>

**Table 42 FLSLOADER_PROCONP0<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONP0<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether write protection is set or not for the sectors of PFlash bank0 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents write protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank0 (refer to FLSLOADER_PF<x>_PROT) is set to WRITE_PROTECTION and<br>• Protection of sector 'y' is set to WRITE_PROTECTION using configuration parameter FlsLoaderPF0Sector/FlsLoaderPF0Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONP0<x> is generated with 0x00000000. |

| Example(s) | Action | Generated output |
|---|---|---|
| | PFlash bank0 sectors 0, 32 and 64 are write protected:<br><br>FlsLoaderPFLash0WriteProt = WRITE_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector0/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector32/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector64/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION | `#define FLSLOADER_PF0_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCONP00 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP01 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP02 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP03 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP04 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP05 ((uint32)0x00000000U)` |

### 1.1.43 Macro: FLSLOADER_PROCONP1<x>

**Table 43 FLSLOADER_PROCONP1<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONP1<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether write protection is set or not for the sectors of PFlash bank1 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents write protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank1 (refer to FLSLOADER_PF<x>_PROT) is set to WRITE_PROTECTION and<br>• Protection of sector 'y' is set to WRITE_PROTECTION using configuration parameter FlsLoaderPF1Sector/FlsLoaderPF1Sector<y>/FlsLoaderPFSectorWriteProtection. |

| | | |
|---|---|---|
| | If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONP1<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank1 sectors 0, 32 and 64 are write protected:<br><br>FlsLoaderPFLash1WriteProt = WRITE_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector0/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector32/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector64/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION | `#define FLSLOADER_PF1_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCONP10 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP11 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP12 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP13 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP14 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP15 ((uint32)0x00000000U)` |

## 1.1.44 Macro: FLSLOADER_PROCONP2<x>

**Table 44 FLSLOADER_PROCONP2<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONP2<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether write protection is set or not for the sectors of PFlash bank2 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents write protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if, |

| | | |
|---|---|---|
| | • Protection of PFlash bank2 (refer to FLSLOADER_PF<x>_PROT) is set to WRITE_PROTECTION and<br>• Protection of sector 'y' is set to WRITE_PROTECTION using configuration parameter FlsLoaderPF2Sector/FlsLoaderPF2Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONP2<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank2 sectors 0, 32 and 64 are write protected:<br><br>FlsLoaderPFLash2WriteProt = WRITE_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector0/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector32/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector64/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION | `#define FLSLOADER_PF2_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCONP20 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP21 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP22 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP23 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP24 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP25 ((uint32)0x00000000U)` |

## 1.1.45 Macro: FLSLOADER_PROCONP3<x>

**Table 45 FLSLOADER_PROCONP3<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONP3<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether write protection is set or not for the sectors of PFlash bank3 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |

| | | |
|---|---|---|
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents write protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank3 (refer to FLSLOADER_PF<x>_PROT) is set to WRITE_PROTECTION and<br>• Protection of sector 'y' is set to WRITE_PROTECTION using configuration parameter FlsLoaderPF3Sector/FlsLoaderPF3Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONP3<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank3 sectors 0, 32 and 64 are write protected:<br><br>FlsLoaderPFLash3WriteProt = WRITE_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector0/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector32/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector64/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION | `#define FLSLOADER_PF3_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCONP30 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP31 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP32 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP33 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP34 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP35 ((uint32)0x00000000U)` |

## 1.1.46 Macro: FLSLOADER_PROCONP4<x>

**Table 46 FLSLOADER_PROCONP4<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONP4<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether write protection is set or not for the sectors of PFlash bank4 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63, |

| | |
|---|---|
| | x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents write protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank4 (refer to FLSLOADER_PF<x>_PROT) is set to WRITE_PROTECTION and<br>• Protection of sector 'y' is set to WRITE_PROTECTION using configuration parameter FlsLoaderPF4Sector/FlsLoaderPF4Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONP4<x> is generated with 0x00000000. |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | PFlash bank4 sectors 0, 32 and 64 are write protected:<br><br>FlsLoaderPFLash4WriteProt = WRITE_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector0/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector32/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector64/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION | `#define FLSLOADER_PF4_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCONP40 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP41 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP42 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP43 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP44 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONP45 ((uint32)0x00000000U)` |

### 1.1.47 Macro: FLSLOADER_PROCONP5<x>

**Table 47 FLSLOADER_PROCONP5<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONP5<x> (x ranges from 0 to 1) |
| **Description** | Specifies whether write protection is set or not for the sectors of PFlash bank5 based on the value of 'x' as described below: |

| | |
|---|---|
| | For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents write protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank5 (refer to FLSLOADER_PF<x>_PROT) is set to WRITE_PROTECTION and<br>• Protection of sector 'y' is set to WRITE_PROTECTION using configuration parameter FlsLoaderPF5Sector/FlsLoaderPF5Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONP5<x> is generated with 0x00000000. |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | PFlash bank5 sector 0 is write protected:<br><br>FlsLoaderPFLash5WriteProt = WRITE_PROTECTION,<br><br>FlsLoaderPF5Sector/FlsLoaderPF5Sector0/FlsLoaderPFSectorWriteProtection = WRITE_PROTECTION | `#define FLSLOADER_PF5_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCONP50 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONP51 ((uint32)0x00000000U)` |

## 1.1.48 Macro: FLSLOADER_PROCONOTP0<x>

**Table 48 FLSLOADER_PROCONOTP0<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONOTP0<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether OTP protection is set or not for the sectors of PFlash bank0 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents OTP protection status of set of 32-sectors identified by 'x'. |

| | | |
|---|---|---|
| | Bit ‘y’ of the macro is set if,<br>• Protection of PFlash bank0 (refer to FLSLOADER_PF<x>_PROT) is set to OTP_PROTECTION and<br>• Protection of sector ‘y’ is set to OTP_PROTECTION using configuration parameter FlsLoaderPF0Sector/FlsLoaderPF0Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by ‘x’) then the respective FLSLOADER_PROCONOTP0<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank0 sectors 0, 32 and 64 are OTP protected:<br><br>FlsLoaderPFLash0WriteProt = OTP_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector0/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector32/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector64/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION | `#define FLSLOADER_PF0_PROT (OTP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONOTP00 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP01 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP02 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP03 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP04 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP05 ((uint32)0x00000000U)` |

## 1.1.49 Macro: FLSLOADER_PROCONOTP1<x>

**Table 49 FLSLOADER_PROCONOTP1<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONOTP1<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether OTP protection is set or not for the sectors of PFlash bank1 based on the value of ‘x’ as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |

| | | |
|---|---|---|
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents OTP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank1 (refer to FLSLOADER_PF<x>_PROT) is set to OTP_PROTECTION and<br>• Protection of sector 'y' is set to OTP_PROTECTION using configuration parameter FlsLoaderPF1Sector/FlsLoaderPF1Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONOTP1<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank1 sectors 0, 32 and 64 are OTP protected:<br><br>FlsLoaderPFLash1WriteProt = OTP_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector0/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector32/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector64/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION | `#define FLSLOADER_PF1_PROT (OTP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONOTP10 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP11 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP12 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP13 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP14 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP15 ((uint32)0x00000000U)` |

## 1.1.50 Macro: FLSLOADER_PROCONOTP2<x>

**Table 50 FLSLOADER_PROCONOTP2<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONOTP2<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether OTP protection is set or not for the sectors of PFlash bank2 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63, |

| | |
|---|---|
| | x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents OTP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank2 (refer to FLSLOADER_PF<x>_PROT) is set to OTP_PROTECTION and<br>• Protection of sector 'y' is set to OTP_PROTECTION using configuration parameter FlsLoaderPF2Sector/FlsLoaderPF2Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONOTP2<x> is generated with 0x00000000. |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | PFlash bank2 sectors 0, 32 and 64 are OTP protected:<br><br>FlsLoaderPFLash2WriteProt = OTP_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector0/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector32/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector64/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION | `#define FLSLOADER_PF2_PROT (OTP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONOTP20 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP21 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP22 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP23 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP24 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP25 ((uint32)0x00000000U)` |

## 1.1.51 Macro: FLSLOADER_PROCONOTP3<x>

**Table 51 FLSLOADER_PROCONOTP3<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONOTP3<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether OTP protection is set or not for the sectors of PFlash bank3 based on the value of 'x' as described below: |

| | | |
|---|---|---|
| | For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents OTP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank3 (refer to FLSLOADER_PF<x>_PROT) is set to OTP_PROTECTION and<br>• Protection of sector 'y' is set to OTP_PROTECTION using configuration parameter FlsLoaderPF3Sector/FlsLoaderPF3Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONOTP3<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank3 sectors 0, 32 and 64 are OTP protected:<br><br>FlsLoaderPFLash3WriteProt = OTP_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector0/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector32/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector64/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION | `#define FLSLOADER_PF3_PROT (OTP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONOTP30 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP31 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP32 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP33 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP34 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP35 ((uint32)0x00000000U)` |

### 1.1.52 Macro: FLSLOADER_PROCONOTP4<x>

**Table 52 FLSLOADER_PROCONOTP4<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONOTP4<x> (x ranges from 0 to 5) | |
| **Description** | Specifies whether OTP protection is set or not for the sectors of PFlash bank4 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents OTP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank4 (refer to FLSLOADER_PF<x>_PROT) is set to OTP_PROTECTION and<br>• Protection of sector 'y' is set to OTP_PROTECTION using configuration parameter FlsLoaderPF4Sector/FlsLoaderPF4Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONOTP4<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank4 sectors 0, 32 and 64 are OTP protected:<br><br>FlsLoaderPFLash4WriteProt = OTP_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector0/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector32/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector64/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION | `#define FLSLOADER_PF4_PROT (OTP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONOTP40 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP41 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP42 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP43 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONOTP44 ((uint32)0x00000000U)` |

| | | |
|---|---|---|
| | | `#define FLSLOADER_PROCONOTP45 ((uint32)0x00000000U)` |

## 1.1.53 Macro: FLSLOADER_PROCONOTP5<x>

**Table 53 FLSLOADER_PROCONOTP5<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONOTP5<x> (x ranges from 0 to 1) | |
| **Description** | Specifies whether OTP protection is set or not for the sectors of PFlash bank5 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents OTP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank5 (refer to FLSLOADER_PF<x>_PROT) is set to OTP_PROTECTION and<br>• Protection of sector 'y' is set to OTP_PROTECTION using configuration parameter FlsLoaderPF5Sector/FlsLoaderPF5Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONOTP5<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank5 sector 0 is OTP protected:<br><br>FlsLoaderPFLash5WriteProt = OTP_PROTECTION,<br><br>FlsLoaderPF5Sector/FlsLoaderPF5Sector0/FlsLoaderPFSectorWriteProtection = OTP_PROTECTION | `#define FLSLOADER_PF5_PROT (OTP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONOTP50 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONOTP51 ((uint32)0x00000000U)` |

## 1.1.54 Macro: FLSLOADER_PROCONWOP0<x>

**Table 54 FLSLOADER_PROCONWOP0<x>**

| | |
|---|---|
| **Name** | FLSLOADER_PROCONWOP0<x> (x ranges from 0 to 5) |
| **Description** | Specifies whether WOP protection is set or not for the sectors of PFlash bank0 based on the value of 'x' as described below: |

| | | |
|---|---|---|
| | For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents WOP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank0 (refer to FLSLOADER_PF<x>_PROT) is set to WOP_PROTECTION and<br>• Protection of sector 'y' is set to WOP_PROTECTION using configuration parameter FlsLoaderPF0Sector/FlsLoaderPF0Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONWOP0<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank0 sectors 0, 32 and 64 are WOP protected:<br><br>FlsLoaderPFLash0WriteProt = WOP_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector0/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector32/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF0Sector/FlsLoaderPF0Sector64/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION | `#define FLSLOADER_PF0_PROT (WOP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONWOP00 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP01 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP02 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP03 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP04 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP05 ((uint32)0x00000000U)` |

## 1.1.55 Macro: FLSLOADER_PROCONWOP1<x>

**Table 55 FLSLOADER_PROCONWOP1<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONWOP1<x> (x ranges from 0 to 5) | |
| **Description** | Specifies whether WOP protection is set or not for the sectors of PFlash bank1 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents WOP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank1 (refer to FLSLOADER_PF<x>_PROT) is set to WOP_PROTECTION and<br>• Protection of sector 'y' is set to WOP_PROTECTION using configuration parameter FlsLoaderPF1Sector/FlsLoaderPF1Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONWOP1<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank1 sectors 0, 32 and 64 are WOP protected:<br><br>FlsLoaderPFLash1WriteProt = WOP_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector0/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector32/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF1Sector/FlsLoaderPF1Sector64/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION | `#define FLSLOADER_PF1_PROT (WOP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONWOP10 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP11 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP12 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP13 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP14 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP15 ((uint32)0x00000000U)` |

### 1.1.56 Macro: FLSLOADER_PROCONWOP2<x>

**Table 56 FLSLOADER_PROCONWOP2<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONWOP2<x> (x ranges from 0 to 5) | |
| **Description** | Specifies whether WOP protection is set or not for the sectors of PFlash bank2 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents WOP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank2 (refer to FLSLOADER_PF<x>_PROT) is set to WOP_PROTECTION and<br>• Protection of sector 'y' is set to WOP_PROTECTION using configuration parameter FlsLoaderPF2Sector/FlsLoaderPF2Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONWOP2<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank2 sectors 0, 32 and 64 are WOP protected:<br><br>FlsLoaderPFLash2WriteProt = WOP_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector0/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector32/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF2Sector/FlsLoaderPF2Sector64/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION | `#define FLSLOADER_PF2_PROT (WOP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONWOP20 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP21 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP22 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP23 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP24 ((uint32)0x00000000U)` |

| | | |
|---|---|---|
| | | `#define FLSLOADER_PROCONWOP25 ((uint32)0x00000000U)` |

## 1.1.57 Macro: FLSLOADER_PROCONWOP3<x>

**Table 57 FLSLOADER_PROCONWOP3<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONWOP3<x> (x ranges from 0 to 5) | |
| **Description** | Specifies whether WOP protection is set or not for the sectors of PFlash bank3 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents WOP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank3 (refer to FLSLOADER_PF<x>_PROT) is set to WOP_PROTECTION and<br>• Protection of sector 'y' is set to WOP_PROTECTION using configuration parameter FlsLoaderPF3Sector/FlsLoaderPF3Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONWOP3<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank3 sectors 0, 32 and 64 are WOP protected:<br><br>FlsLoaderPFLash3WriteProt = WOP_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector0/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector32/FlsLoaderPFSectorWriteProte | `#define FLSLOADER_PF3_PROT (WOP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONWOP30 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP31 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP32 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP33` |

| | | |
|---|---|---|
| | ction = WOP_PROTECTION,<br><br>FlsLoaderPF3Sector/FlsLoaderPF3Sector64/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION | ((uint32)0x00000000U)<br><br>#define FLSLOADER_PROCONWOP34 ((uint32)0x00000000U)<br><br>#define FLSLOADER_PROCONWOP35 ((uint32)0x00000000U) |

## 1.1.58 Macro: FLSLOADER_PROCONWOP4<x>

**Table 58 FLSLOADER_PROCONWOP4<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONWOP4<x> (x ranges from 0 to 5) | |
| **Description** | Specifies whether WOP protection is set or not for the sectors of PFlash bank4 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63,<br>x=2, sectors = 64 to 95,<br>x=3, sectors = 96 to 127,<br>x=4, sectors = 128 to 159,<br>x=5, sectors = 160 to 191 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents WOP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank4 (refer to FLSLOADER_PF<x>_PROT) is set to WOP_PROTECTION and<br>• Protection of sector 'y' is set to WOP_PROTECTION using configuration parameter FlsLoaderPF4Sector/FlsLoaderPF4Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONWOP4<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank4 sectors 0, 32 and 64 are WOP protected:<br><br>FlsLoaderPFLash4WriteProt = WOP_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector0/FlsLoaderPFSectorWriteProtect | #define FLSLOADER_PF4_PROT (WOP_PROTECTION)<br><br>#define FLSLOADER_PROCONWOP40 ((uint32)0x00000001U)<br><br>#define FLSLOADER_PROCONWOP41 ((uint32)0x00000001U) |

| | | |
|---|---|---|
| | ion = WOP_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector32/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION,<br><br>FlsLoaderPF4Sector/FlsLoaderPF4Sector64/FlsLoaderPFSectorWriteProtection = WOP_PROTECTION | `#define FLSLOADER_PROCONWOP42 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP43 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP44 ((uint32)0x00000000U)`<br><br>`#define FLSLOADER_PROCONWOP45 ((uint32)0x00000000U)` |

## 1.1.59 Macro: FLSLOADER_PROCONWOP5<x>

**Table 59 FLSLOADER_PROCONWOP5<x>**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCONWOP5<x> (x ranges from 0 to 1) | |
| **Description** | Specifies whether WOP protection is set or not for the sectors of PFlash bank5 based on the value of 'x' as described below:<br><br>For,<br>x=0, sectors = 0 to 31,<br>x=1, sectors = 32 to 63 | |
| **Verification method** | The macro is generated as a 32-bit numeric value where each bit (y=0 to 31) represents WOP protection status of set of 32-sectors identified by 'x'.<br><br>Bit 'y' of the macro is set if,<br>• Protection of PFlash bank5 (refer to FLSLOADER_PF<x>_PROT) is set to WOP_PROTECTION and<br>• Protection of sector 'y' is set to WOP_PROTECTION using configuration parameter FlsLoaderPF5Sector/FlsLoaderPF5Sector<y>/FlsLoaderPFSectorWriteProtection.<br><br>If a device does not have all the mentioned sectors (as per the specified range denoted by 'x') then the respective FLSLOADER_PROCONWOP5<x> is generated with 0x00000000. | |
| **Example(s)** | **Action** | **Generated output** |
| | PFlash bank5 sector 0 is WOP protected:<br><br>FlsLoaderPFLash5WriteProt = WOP_PROTECTION,<br><br>FlsLoaderPF5Sector/FlsLoaderPF5Sector0/FlsLoaderPFSectorWriteProtect | `#define FLSLOADER_PF5_PROT (WOP_PROTECTION)`<br><br>`#define FLSLOADER_PROCONWOP50 ((uint32)0x00000001U)`<br><br>`#define FLSLOADER_PROCONWOP51 ((uint32)0x00000000U)` |

| | | |
|---|---|---|
| | ion = WOP_PROTECTION | |

### 1.1.60 Macro: FLSLOADER_UCB_PFLASH_ORGINAL_START

**Table 60 FLSLOADER_UCB_PFLASH_ORGINAL_START**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_PFLASH_ORGINAL_START | |
| **Description** | Specifies the start address of PFlash original UCB (UCB_PFLASH_ORIG). | |
| **Verification method** | The macro is generated as a hex value specifying the start address of PFlash original UCB (UCB_PFLASH_ORIG). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Start address of PFlash original UCB is 0xAF402000 | `#define FLSLOADER_UCB_PFLASH_ORGINAL_START ((FlsLoader_AddressType)(0xaf402000U))` |

### 1.1.61 Macro: FLSLOADER_UCB_PFLASH_ORGINAL_END

**Table 61 FLSLOADER_UCB_PFLASH_ORGINAL_END**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_PFLASH_ORGINAL_END | |
| **Description** | Specifies the end address of PFlash original UCB (UCB_PFLASH_ORIG). | |
| **Verification method** | The macro is generated as a hex value specifying the end address of PFlash original UCB (UCB_PFLASH_ORIG). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | End address of PFlash original UCB is 0xAF4021FF | `#define FLSLOADER_UCB_PFLASH_ORGINAL_END ((FlsLoader_AddressType)(0xaf4021ffU))` |

### 1.1.62 Macro: FLSLOADER_UCB_PFLASH_ORGINAL_SIZE

**Table 62 FLSLOADER_UCB_PFLASH_ORGINAL_SIZE**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_PFLASH_ORGINAL_SIZE |
| **Description** | Specifies the size of PFlash original UCB (UCB_PFLASH_ORIG). |
| **Verification method** | The macro is generated as a hex value derived by dividing the size of PFlash original UCB (UCB_PFLASH_ORIG) by 4. This value is same for all devices of the TC3xx family. |

| | | |
|---|---|---|
| | *Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Size of PFlash original UCB (UCB_PFLASH_ORIG) is 512 bytes. | `#define FLSLOADER_UCB_PFLASH_ORGINAL_SIZE (0x80U)` |

### 1.1.63 Macro: FLSLOADER_UCB_PFLASH_COPY_START

**Table 63 FLSLOADER_UCB_PFLASH_COPY_START**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_PFLASH_COPY_START | |
| **Description** | Specifies the start address of PFlash copy UCB (UCB_PFLASH_COPY). | |
| **Verification method** | The macro is generated as a hex value specifying the start address of PFlash copy UCB (UCB_PFLASH_COPY). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Start address of PFlash copy UCB is 0xAF403000 | `#define FLSLOADER_UCB_PFLASH_COPY_START ((FlsLoader_AddressType)(0xaf403000U))` |

### 1.1.64 Macro: FLSLOADER_UCB_PFLASH_COPY_END

**Table 64 FLSLOADER_UCB_PFLASH_COPY_END**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_PFLASH_COPY_END | |
| **Description** | Specifies the end address of PFlash copy UCB (UCB_PFLASH_COPY). | |
| **Verification method** | The macro is generated as a hex value specifying the end address of PFlash copy UCB (UCB_PFLASH_COPY). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | End address of PFlash copy UCB is 0xAF4031FF | `#define FLSLOADER_UCB_PFLASH_COPY_END ((FlsLoader_AddressType)(0xaf4031ffU))` |

### 1.1.65 Macro: FLSLOADER_UCB_PFLASH_COPY_SIZE

**Table 65 FLSLOADER_UCB_PFLASH_COPY_SIZE**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_PFLASH_COPY_SIZE |
| **Description** | Specifies the size of PFlash copy UCB (UCB_PFLASH_COPY). |

| | |
|---|---|
| **Verification method** | The macro is generated as a hex value specifying the size of PFlash copy UCB (UCB_PFLASH_COPY). This value is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | Size of PFlash copy UCB (UCB_PFLASH_COPY) is 512 bytes. | `#define FLSLOADER_UCB_PFLASH_COPY_SIZE (0x200U)` |

### 1.1.66 Macro: FLSLOADER_UCB_DFLASH_ORIGINAL_START

**Table 66 FLSLOADER_UCB_DFLASH_ORIGINAL_START**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_DFLASH_ORIGINAL_START |
| **Description** | Specifies the start address of DFlash original UCB (UCB_DFLASH_ORIG). |
| **Verification method** | The macro is generated as a hex value specifying the start address of DFlash original UCB (UCB_DFLASH_ORIG). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | Start address of DFlash original UCB is 0xAF402200 | `#define FLSLOADER_UCB_DFLASH_ORIGINAL_START ((FlsLoader_AddressType)(0xaf402200U))` |

### 1.1.67 Macro: FLSLOADER_UCB_DFLASH_ORGINAL_END

**Table 67 FLSLOADER_UCB_DFLASH_ORGINAL_END**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_DFLASH_ORGINAL_END |
| **Description** | Specifies the end address of DFlash original UCB (UCB_DFLASH_ORIG). |
| **Verification method** | The macro is generated as a hex value specifying the end address of DFlash original UCB (UCB_DFLASH_ORIG). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | End address of DFlash original UCB is 0xAF4023FF | `#define FLSLOADER_UCB_DFLASH_ORGINAL_END ((FlsLoader_AddressType)(0xaf4023ffU))` |

### 1.1.68 Macro: FLSLOADER_UCB_DFLASH_ORGINAL_SIZE

**Table 68 FLSLOADER_UCB_DFLASH_ORGINAL_SIZE**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_DFLASH_ORGINAL_SIZE |
| **Description** | Specifies the size of DFlash original UCB (UCB_DFLASH_ORIG). |
| **Verification method** | The macro is generated as a hex value derived by dividing the size of DFlash original UCB (UCB_DFLASH_ORIG) by 4. This value is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | Size of DFlash original UCB (UCB_DFLASH_ORIG) is 512 bytes. | `#define FLSLOADER_UCB_DFLASH_ORGINAL_SIZE (0x80U)` |

### 1.1.69 Macro: FLSLOADER_UCB_DFLASH_COPY_START

**Table 69 FLSLOADER_UCB_DFLASH_COPY_START**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_DFLASH_COPY_START |
| **Description** | Specifies the start address of DFlash copy UCB (UCB_DFLASH_COPY). |
| **Verification method** | The macro is generated as a hex value specifying the start address of DFlash copy UCB (UCB_DFLASH_COPY). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | Start address of DFlash copy UCB is 0xAF403200 | `#define FLSLOADER_UCB_DFLASH_COPY_START ((FlsLoader_AddressType)(0xaf403200U))` |

### 1.1.70 Macro: FLSLOADER_UCB_DFLASH_COPY_END

**Table 70 FLSLOADER_UCB_DFLASH_COPY_END**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_DFLASH_COPY_END |
| **Description** | Specifies the end address of DFlash copy UCB (UCB_DFLASH_COPY). |
| **Verification method** | The macro is generated as a hex value specifying the end address of DFlash copy UCB (UCB_DFLASH_COPY). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | End address of DFlash copy UCB is 0xAF4033FF | `#define FLSLOADER_UCB_DFLASH_COPY_END ((FlsLoader_AddressType)(0xaf4033ffU))` |

### 1.1.71 Macro: FLSLOADER_UCB_DFLASH_COPY_SIZE

**Table 71 FLSLOADER_UCB_DFLASH_COPY_SIZE**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_DFLASH_COPY_SIZE | |
| **Description** | Specifies the size of DFlash copy UCB (UCB_DFLASH_COPY). | |
| **Verification method** | The macro is generated as a hex value specifying the size of PFlash copy UCB (UCB_PFLASH_COPY). This value is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Size of DFlash copy UCB (UCB_DFLASH_COPY) is 512 bytes. | `#define FLSLOADER_UCB_DFLASH_COPY_SIZE (0x200U)` |

### 1.1.72 Macro: FLSLOADER_UCB_OTP0_START

**Table 72 FLSLOADER_UCB_OTP0_START**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_OTP0_START | |
| **Description** | Specifies the start address of OTP0 original UCB (UCB_OTP0_ORIG). | |
| **Verification method** | The macro is generated as a hex value specifying the start address of OTP0 original UCB (UCB_OTP0_ORIG). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Start address of OTP0 original UCB is 0xAF404000 | `#define FLSLOADER_UCB_OTP0_START ((FlsLoader_AddressType)(0xaf404000U))` |

### 1.1.73 Macro: FLSLOADER_UCB_OTP0_END

**Table 73 FLSLOADER_UCB_OTP0_END**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_OTP0_END | |
| **Description** | Specifies the end address of OTP0 original UCB (UCB_OTP0_ORIG). | |
| **Verification method** | The macro is generated as a hex value specifying the end address of OTP0 original UCB (UCB_OTP0_ORIG). This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | End address of OTP0 original UCB is | `#define FLSLOADER_UCB_OTP0_END` |

| | | |
|---|---|---|
| | 0xAF4041FF | `((FlsLoader_AddressType)(0xaf4041ffU))` |

### 1.1.74 Macro: FLSLOADER_UCB_OTP0_SIZE

**Table 74 FLSLOADER_UCB_OTP0_SIZE**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_OTP0_SIZE | |
| **Description** | Specifies the size of OTP0 original UCB (UCB_OTP0_ORIG). | |
| **Verification method** | The macro is generated as a hex value derived by dividing the size of DFlash original UCB (UCB_OTP0_ORIG) by 4. This value is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Size of OTP0 original UCB (UCB_OTP0_ORIG) is 512 bytes. | `#define FLSLOADER_UCB_OTP0_SIZE (0x80U)` |

### 1.1.75 Macro: FLSLOADER_UCB_START

**Table 75 FLSLOADER_UCB_START**

| | | |
|---|---|---|
| **Name** | FLSLOADER_UCB_START | |
| **Description** | Specifies the start address of UCB block (DFLASH0_UCB) in DFlash0. | |
| **Verification method** | The macro is generated as a hex value specifying the start address of UCB block (DFLASH0_UCB) in DFlash0. This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Start address of UCB block (DFLASH0_UCB) is 0xAF400000 | `#define FLSLOADER_UCB_START ((FlsLoader_AddressType)(0xaf400000U))` |

### 1.1.76 Macro: FLSLOADER_UCB_END

**Table 76 FLSLOADER_UCB_END**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_END |
| **Description** | Specifies the end address of UCB block (DFLASH0_UCB) in DFlash0. |
| **Verification method** | The macro is generated as a hex value specifying the end address of UCB block (DFLASH0_UCB) in DFlash0. This address is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| Example(s) | Action | Generated output |
|---|---|---|
| | End address of UCB block (DFLASH0_UCB) is 0xAF405FFF | `#define FLSLOADER_UCB_END ((FlsLoader_AddressType)(0xaf405fffU))` |

### 1.1.77 Macro: FLSLOADER_UCB_SIZE

**Table 77 FLSLOADER_UCB_SIZE**

| | |
|---|---|
| **Name** | FLSLOADER_UCB_SIZE |
| **Description** | Specifies the total size of UCB block (DFLASH0_UCB) in DFlash0. |
| **Verification method** | The macro is generated as a hex value specifying the total size of UCB block (DFLASH0_UCB) in DFlash0. This value is same for all devices of the TC3xx family.<br><br>*Note: This macro is not configurable by the user.* |

| Example(s) | Action | Generated output |
|---|---|---|
| | Size of UCB block (DFLASH0_UCB) is 24KB. | `#define FLSLOADER_UCB_SIZE (0x6000U)` |

### 1.1.78 Macro: FLSLOADER_FLASH_BUSY_MASK

**Table 78 FLSLOADER_FLASH_BUSY_MASK**

| | |
|---|---|
| **Name** | FLSLOADER_FLASH_BUSY_MASK |
| **Description** | Specifies mask for checking the busy status of DFlash0 and all available PFlash banks from HF_STATUS register. |
| **Verification method** | The macro is generated as a 32-bit hex value specifying the mask to check the busy status bits of DFlash0 and all available PFlash banks of HF_STATUS register.<br><br>*Note: This macro is not configurable by the user.* |

| Example(s) | Action | Generated output |
|---|---|---|
| | Device has 6 PFlash banks | `#define FLSLOADER_FLASH_BUSY_MASK (0x000000FDU)` |
| | Device has 4 PFlash banks | `#define FLSLOADER_FLASH_BUSY_MASK (0x0000003DU)` |
| | Device has 2 PFlash banks | `#define FLSLOADER_FLASH_BUSY_MASK (0x0000000DU)` |

### 1.1.79 Macro: FLSLOADER_DERIVATIVE

**Table 79 FLSLOADER_DERIVATIVE**

| | |
|---|---|
| **Name** | FLSLOADER_DERIVATIVE |
| **Description** | Indicates selected sub-derivative of TC3xx family. |

| | | |
|---|---|---|
| **Verification method** | The macro is generated with sub-derivative of TC3xx family selected by the user.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Configuration is for TC399 device | `#define FLSLOADER_DERIVATIVE (FLSLOADER_TC399x)` |
| | Configuration is for TC387 device | `#define FLSLOADER_DERIVATIVE (FLSLOADER_TC387x)` |

### 1.1.80 Macro: FLSLOADER_DF0_USERMODE

**Table 80 FLSLOADER_DF0_USERMODE**

| | | |
|---|---|---|
| **Name** | FLSLOADER_DF0_USERMODE | |
| **Description** | Indicates single sensing user mode for DFlash0. | |
| **Verification method** | The macro is assigned with constant value of 0U.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |

### 1.1.81 Macro: FLSLOADER_PF_BANKS

**Table 81 FLSLOADER_PF_BANKS**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PF_BANKS | |
| **Description** | Indicates number of PFlash banks available in the device. | |
| **Verification method** | The macro is generated with number of PFlash banks available in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Selected device has 6 PFlash banks | `#define FLSLOADER_PF_BANKS (6U)` |
| | Selected device has 4 PFlash banks | `#define FLSLOADER_PF_BANKS (4U)` |

### 1.1.82 Macro: FLSLOADER_PFLASH_END

**Table 82 FLSLOADER_PFLASH_END**

| | |
|---|---|
| **Name** | FLSLOADER_PFLASH_END |
| **Description** | Indicates end address of last PFlash bank in the selected device. |

| | | |
|---|---|---|
| **Verification method** | The macro is generated with the end address of last PFlash bank in the selected device.<br><br>*Note: This macro is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Selected device has 6 PFlash banks (ranging from 0 to 5) | `#define FLSLOADER_PFLASH_END (FLSLOADER_PFLASH5_END)` |
| | Selected device has 4 PFlash banks (ranging from 0 to 3) | `#define FLSLOADER_PFLASH_END (FLSLOADER_PFLASH3_END)` |

### 1.1.83 Macro: FLSLOADER_PROCOND0

**Table 83 FLSLOADER_PROCOND0**

| | | |
|---|---|---|
| **Name** | FLSLOADER_PROCOND0 | |
| **Description** | Indicates the 32-bit data to be programmed in to UCB_DFLASH to set the DFlash0 protection selected by FlsLoaderDF0Prot. | |
| **Verification method** | The macro is assigned with a 32-bit hex value as mentioned below based on the protection value selected in FlsLoaderDF0Prot.<br><br>NO_PROTECTION = 0x00000000,<br>WRITE_PROTECTION = 0x00000001,<br>READ_PROTECTION = 0x80000001. | |
| **Example(s)** | **Action** | **Generated output** |
| | FlsLoaderDF0Prot = NO_PROTECTION | `#define FLSLOADER_DF0_PROT (NO_PROTECTION)`<br><br>`#define FLSLOADER_PROCOND0 ((uint32)(0x00000000U))` |
| | FlsLoaderDF0Prot = READ_PROTECTION | `#define FLSLOADER_DF0_PROT (READ_PROTECTION)`<br><br>`#define FLSLOADER_PROCOND0 ((uint32)(0x80000001U))` |
| | FlsLoaderDF0Prot = WRITE_PROTECTION | `#define FLSLOADER_DF0_PROT (WRITE_PROTECTION)`<br><br>`#define FLSLOADER_PROCOND0 ((uint32)(0x00000001U))` |

## 1.2 File: FlsLoader_Cfg.c

The generated source file contains pre-compile configuration parameters and constant data generated based on the selected derivative of the TC3xx family. The file is generated in ‘src’ folder.

### 1.2.1 Function Pointer: FlsLoader_CallOutPtr

**Table 84 FlsLoader_CallOutPtr**

| | |
|---|---|
| **Name** | FlsLoader_CallOutPtr |
| **Type** | FlsLoader_CallOutFunc |
| **Description** | Function pointer pointing to configured callout function. |
| **Verification method** | The function pointer is generated with address of callout function configured using configuration parameter FlsLoaderCallOutFunction.<br><br>*Note: User can configure FlsLoaderCallOutFunction either with name or with address of the callout function.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | FlsLoaderCallOutFunction = FlsLoader_LoopCallOut | `const FlsLoader_CallOutFunc FlsLoader_CallOutPtr = &FlsLoader_LoopCallOut;` |
| | FlsLoaderCallOutFunction = `0x80005000` | `const FlsLoader_CallOutFunc FlsLoader_CallOutPtr = \ (FlsLoader_CallOutFunc) 0X80005000U;` |
| | FlsLoaderCallOutFunction = `1879089152` | `const FlsLoader_CallOutFunc FlsLoader_CallOutPtr = \ (FlsLoader_CallOutFunc) 0X7000A000U;` |

### 1.2.2 Array: FlsLoader_PFlashOffset

**Table 85 FlsLoader_PFlashOffset**

| | |
|---|---|
| **Name** | FlsLoader_PFlashOffset |
| **Type** | uint32 |
| **Description** | Constant array containing start address of each PFlash bank. |
| **Verification method** | The array is generated with its size equal to number of PFlash banks available in the selected device and contains the start address of each PFlash bank.<br><br>*Note: This array is not configurable by the user.* |

| **Example(s)** | **Action** | **Generated output** |
|---|---|---|
| | Selected device TC375 has 2 PFlash banks | `const uint32 FlsLoader_PFlashOffset[FLSLOADER_NUM_OF_PFLASH_BANK] = { (uint32)0XA0000000U, (uint32)0XA0300000U };` |
| | Selected device TC389 has 4 PFlash banks | `const uint32 FlsLoader_PFlashOffset[FLSLOADER_NUM_OF_PFLASH_BANK] = { (uint32)0XA0000000U, (uint32)0XA0300000U, (uint32)0XA0600000U, (uint32)0XA0900000U` |

| | | |
|---|---|---|
| | | }; |

### 1.2.3 Array: FlsLoader_PFlashSectorCount

**Table 86 FlsLoader_PFlashSectorCount**

| | | |
|---|---|---|
| **Name** | FlsLoader_PFlashSectorCount | |
| **Type** | uint8 | |
| **Description** | Constant array containing number of sectors present in each PFlash bank. | |
| **Verification method** | The array is generated with size x (x=1 to maximum number of PFlash banks available in the selected device) and contains the macro `FLSLOADER_NUM_OF_PF<x>_SECTORS` representing number of sectors present in PFlash bank 'x'.<br><br>*Note: This array is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Selected device TC375 has 2 PFlash banks | `const uint8 FlsLoader_PFlashSectorCount[FLSLOADER_NUM_OF_PFLASH_BANK] = { FLSLOADER_NUM_OF_PF0_SECTORS, FLSLOADER_NUM_OF_PF1_SECTORS };` |
| | Selected device TC389 has 4 PFlash banks | `const uint8 FlsLoader_PFlashSectorCount[FLSLOADER_NUM_OF_PFLASH_BANK] = { FLSLOADER_NUM_OF_PF0_SECTORS, FLSLOADER_NUM_OF_PF1_SECTORS, FLSLOADER_NUM_OF_PF2_SECTORS, FLSLOADER_NUM_OF_PF3_SECTORS };` |

### 1.2.4 Array: FlsLoader_PFlashEndAddress

**Table 87 FlsLoader_PFlashEndAddress**

| | | |
|---|---|---|
| **Name** | FlsLoader_PFlashEndAddress | |
| **Type** | uint32 | |
| **Description** | Constant array containing the end address of all available PFlash banks. | |
| **Verification method** | The array is generated with its size equal to number of PFlash banks available in the selected device and contains the end address of each PFlash bank.<br><br>*Note: This array is not configurable by the user.* | |
| **Example(s)** | **Action** | **Generated output** |
| | Selected device TC375 has 2 PFlash banks | `const uint32 FlsLoader_PFlashEndAddress[FLSLOADER_NUM_OF_PFLASH_BANK] = { (uint32)0XA02FFFFFU, (uint32)0XA05FFFFFU };` |
| | Selected device TC389 has 4 PFlash banks | `const uint32 FlsLoader_PFlashEndAddress[FLSLOADER_NUM_OF_PFLASH_BANK] = { (uint32)0XA02FFFFFU, (uint32)0XA05FFFFFU, (uint32)0XA08FFFFFU, (uint32)0XA09FFFFFU };` |

# Revision history

### Major changes since the last revision

| Date | Version | Description |
|---|---|---|
| 2020-12-10 | 2.0 | Released. |
| 2020-12-10 | 1.1 | Configuration parameter names corrected to FlsLoaderPFUcbPW0_0, FlsLoaderPFUcbPW0_1, FlsLoaderPFUcbPW1_0, FlsLoaderPFUcbPW1_1, FlsLoaderPFUcbPW2_0, FlsLoaderPFUcbPW2_1, FlsLoaderPFUcbPW3_0, FlsLoaderPFUcbPW3_1. |
| 2020-11-26 | 1.0 | Released. |
| 2020-11-25 | 0.1 | • Aligned with template, no functional changes.<br>• FLSLOADER driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_CD.pdf to this document. |

**Trademarks**
All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2020-12-10**
**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**



**Do you have a question about this document?**

**Email: erratum@infineon.com**

**Document reference**
**Doc_Number**

**IMPORTANT NOTICE**
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

**WARNINGS**
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.