Flash Driver for STM32F103

Generated by Doxygen 1.8.18

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 FLITF_t Struct Reference

Data Fields

- volatile uint_32t **FLASH_ACR**
- volatile uint_32t **FLASH_KEYR**
- volatile uint_32t **FLASH_OPTKEYR**
- volatile uint_32t **FLASH_SR**
- volatile uint_32t **FLASH_CR**
- volatile uint_32t **FLASH_AR**
- volatile uint_32t **FLASH_RESERVED**
- volatile uint_32t **FLASH_OPR**
- volatile uint_32t **FLASH_WRPR**

The documentation for this struct was generated from the following file:

- FLASH/FLITF.c

Chapter 4

File Documentation

4.1 FLASH/FLITF.c File Reference

This file is the Implementation for Flash Driver Interface for STM32F103.

```
#include "STD_TYPES.h"
#include "FLITF.h"
```

Data Structures

- struct FLITF_t

Macros

- #define FPEC ((FLITF_t *)(0x40022000))

  Casting Base Address of Flash Driver as Pointer to struct FLITF_t.
- #define HALF_WORD_LEN 2

  HNo of bytes of Half Word.
- #define KEY1 (0x45670123)

  KEY1 to unlock Flash to write on it.
- #define KEY2 (0xcdef89ab)

  KEY2 to unlock Flash to write on it.
- #define LOCK 0x00000080

  used to lock Flash
- #define START 0x00000040

  used to trigger an ERASE operation when set
- #define MASS_ERASE 0x00000004

  used to earse all the Flash
- #define PAGE_ERASE 0x00000002

  used to earse page
- #define PROGRAM_ENABLE 0x00000001

  used to enable falsh programming
- #define MER_RESET 0x00001FFB

  used to reset Mass Erase bit.
- #define EOP 0x00000020

  used to indicate that a Flash operation is completed
- #define BUSY 0x00000001

  used to indicate that a Flash operation is in progress.

Functions

- void Flash_Lock (void)

  Function to Lock The Flash.
- void Flash_Unlock (void)

  Function to Unlock The Flash.
- void Flash_MassErase (void)

  Function to erase all The Flash.
- void Flash_ErasePage (uint_32t PageAddress)

  Function to erase page from flash.
- void Flash_ProgramWrite (void ∗StartAddress, void ∗DataAddress, uint_32t DataLength)

  Function to write Full Data.
- void Flash_HalfWord (uint_16t ∗StartAddress, uint_16t Data)

  Function to write Half Word Data.
- void Flash_FullWord (uint_32t ∗StartAddress, uint_32t Data)

  Function to write Full Word Data.

4.1.1 Detailed Description

This file is the Implementation for Flash Driver Interface for STM32F103.

This file is a user interface for Flash Driver Interface for STM32F103.

Author

Amr (Ibrahimamr222@gmail.com)

Version

0.1

Date

2020-06-05

Copyright

Copyright (c) 2020

4.1.2 Macro Definition Documentation

4.1.2.1 PROGRAM_ENABLE

```
#define PROGRAM_ENABLE 0x00000001
```

used to enable falsh programming

4.1.3 Function Documentation

4.1.3.1 Flash_ErasePage()

```
void Flash_ErasePage (
            uint_32t PageAddress )
```

Function to erase page from flash.

Parameters

NA	

Returns

NA

4.1.3.2 Flash_FullWord()

```
void Flash_FullWord (
            uint_32t * StartAddress,
            uint_32t Data )
```

Function to write Full Word Data.

Parameters

StartAddress	pointer to uint_32t , Start address to write
Data	variabe of uint_32t , Data to be written

Returns

NA

4.1.3.3 Flash_HalfWord()

```
void Flash_HalfWord (
            uint_16t * StartAddress,
            uint_16t Data )
```

Function to write Half Word Data.

Parameters

StartAddress	pointer to uint_16t , Start address to write
Data	variabe of uint_16t , Data to be written

Returns

NA

4.1.3.4 Flash_Lock()

```
void Flash_Lock (
            void  )
```

Function to Lock The Flash.

Parameters

NA	

Returns

NA

4.1.3.5 Flash_MassErase()

```
void Flash_MassErase (
            void  )
```

Function to erase all The Flash.

Parameters

NA	

Returns

NA

4.1.3.6 Flash_ProgramWrite()

```
void Flash_ProgramWrite (
            void * StartAddress,
```

```
            void * DataAddress,
            uint_32t DataLength )
```

Function to write Full Data.

Parameters

StartAddress	pointer to void , Start address to write
DataAddress	pointer to void , Address of buffer of data
DataLength	variabe of uint_32t , Length of Data to be written

Returns

NA

4.1.3.7 Flash_Unlock()

```
void Flash_Unlock (
            void  )
```

Function to Unlock The Flash.

Parameters

NA	

Returns

NA

Index