MCU-AN-300213-E-V13

F²MC-16FX FAMILY

16-BIT MICROCONTROLLER

ALL SERIES

FLASH SECURITY

APPLICATION NOTE

Revision History

Date	Issue
2006-08-31	V1.0 PHu Initial Version
2006-10-16	V1.1 PHu Corrected UNLOCK command
2007-06-22	V1.2 HPi Updated Configuration
2007-08-01	V1.3 HPi Add memory map, flowchart for unlocking

This document contains 12 pages.

 © Fujitsu Microelectronics Europe GmbH

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products delivered free of charge** (e.g. software include or header files, application examples, target boards, evaluation boards, engineering samples of IC's etc.), its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling. **Note, all these products are intended and must only be used in an evaluation laboratory environment**.

1. Fujitsu Microelectronics Europe GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials for a period of 90 days form the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Microelectronics Europe GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.

2. Should a Product turn out to be defect, Fujitsu Microelectronics Europe GmbH´s entire liability and the customer's exclusive remedy shall be, at Fujitsu Microelectronics Europe GmbH´s sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Microelectronics Europe GmbH in original packing and without further defects resulting from the customer's use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Microelectronics Europe GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Microelectronics Europe GmbH.

3. To the maximum extent permitted by applicable law Fujitsu Microelectronics Europe GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.

4. To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH´s and its suppliers´ liability is restricted to intention and gross negligence.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

To the maximum extent permitted by applicable law, in no event shall Fujitsu Microelectronics Europe GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect

Contents

1 Introduction

The 16FX Family of microcontrollers features a Flash Security function. This function ensures that the contents of the Flash memory cannot be read. Instead, when reading the Flash memory, only the data 0xFF is returned for every location.

2 Flash Security

The Flash Security feature protects the content of the Flash Memory. In the following, it is explained how this feature can be used.

2.1 Features

Often, it is desirable to protect the content of the Flash Memory from read-out. For this, the 16FX Family MCUs offer the Flash Security feature. When the flash security is enabled, flash memory cannot be read by

- Program activated by external boot vector fetch (modes 0/1/6)
- External parallel flash programmer (mode 7)
- Serial communication mode (mode 2)

However internal user program that has been started in internal vector mode (mode 3) has access to the content of the Flash Memory.

Generally, there are 3 different access levels for the Flash Memory:

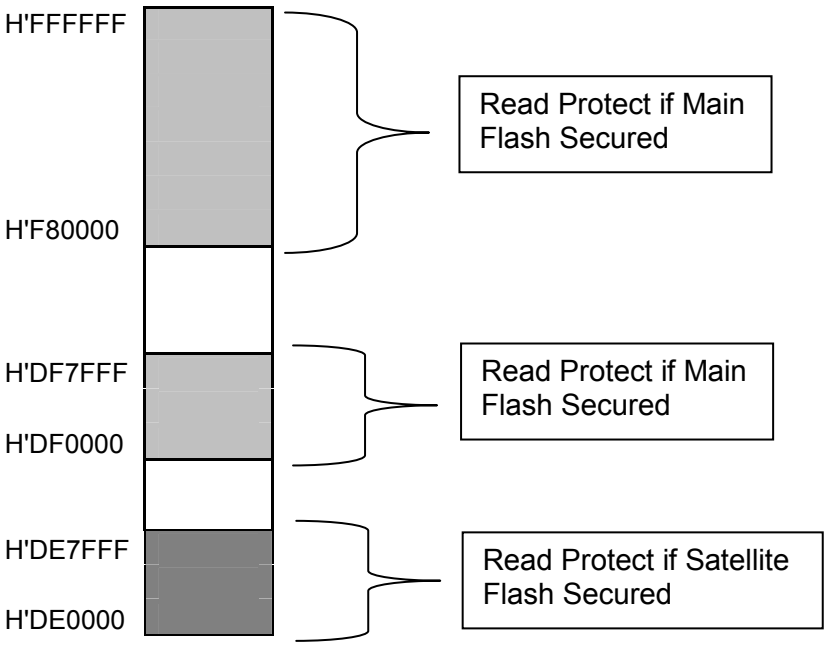
- Unprotected (Flash Security disabled)
- Flash Security enabled, but can be disabled by an Unlock Key
- Flash Security enabled and cannot be disabled

The default is that the Flash Security is disabled. The Flash Security can be enabled by storing a certain pattern at a specific location. See below for more details. If the Flash Security is enabled, the user can also supply a 16-Byte long Unlock Key that allows disabling the Flash Security. If the supplied Unlock Key is all 0, the Flash Security cannot be disabled at all.

Note: The Flash Memory normally allows erasing either the complete memory (“Chip Erase”) or individual sectors (“Sector Erase”). Because the Flash Security is enabled by storing a pattern at a specific location, it would possible to disable the Flash Security by erasing just the single sector that contains this location. To avoid this, a Sector Erase cannot be performed when the Flash security is enabled. Only a Chip Erase is allowed. In case of Chip Erase, the flash is erased from the highest memory address to the lowest memory address because of that security byte is erased at last after erasing entire content of flash.

 © Fujitsu Microelectronics Europe GmbH

2.2 Memory Map showing Secured Areas



2.3 Configuration

The Flash Security feature is configured using the ROM Configuration Block. This is done separately for the Main Flash Memory and the Satellite Flash Memory, if available. The settings can be made manually. Because this is rather tedious, it can be configured easily using the `start.asm` file.

2.3.1 Start.asm Configuration

The configuration block of the Flash Security looks like shown below.

```
;====================================================================
; 4.12 Flash Security
;====================================================================

#set      MAIN_SECURITY_ENABLE       OFF ; <<< enable Flash Security for
Main Flash
#set      SATELLITE_FLASH            OFF ; <<< select if Satellite Flash
is available
#set      SATELLITE_SECURITY_ENABLE  OFF ; <<< enable Flash Security for
Satellite Flash

; set the Flash Security unlock key (16 bytes)
; all 0: unlock not possible
#set      MAIN_UNLOCK_0              0x00
#set      MAIN_UNLOCK_1              0x00
#set      MAIN_UNLOCK_2              0x00
#set      MAIN_UNLOCK_3              0x00
#set      MAIN_UNLOCK_4              0x00
#set      MAIN_UNLOCK_5              0x00
#set      MAIN_UNLOCK_6              0x00
#set      MAIN_UNLOCK_7              0x00
#set      MAIN_UNLOCK_8              0x00
#set      MAIN_UNLOCK_9              0x00
#set      MAIN_UNLOCK_10             0x00
#set      MAIN_UNLOCK_11             0x00
#set      MAIN_UNLOCK_12             0x00
#set      MAIN_UNLOCK_13             0x00
#set      MAIN_UNLOCK_14             0x00
#set      MAIN_UNLOCK_15             0x00

#set      SATELLITE_UNLOCK_0         0x00
#set      SATELLITE_UNLOCK_1         0x00
#set      SATELLITE_UNLOCK_2         0x00
#set      SATELLITE_UNLOCK_3         0x00
#set      SATELLITE_UNLOCK_4         0x00
#set      SATELLITE_UNLOCK_5         0x00
#set      SATELLITE_UNLOCK_6         0x00
#set      SATELLITE_UNLOCK_7         0x00
#set      SATELLITE_UNLOCK_8         0x00
#set      SATELLITE_UNLOCK_9         0x00
#set      SATELLITE_UNLOCK_10        0x00
#set      SATELLITE_UNLOCK_11        0x00
#set      SATELLITE_UNLOCK_12        0x00
#set      SATELLITE_UNLOCK_13        0x00
#set      SATELLITE_UNLOCK_14        0x00
#set      SATELLITE_UNLOCK_15        0x00
```

Enabling the Flash Security for the Main Flash is as easy as setting:

```
;====================================================================
; 4.12 Flash Security
;====================================================================

#set      MAIN_SECURITY_ENABLE       ON ; <<< enable Flash Security for
Main Flash
```

To enable the Flash Security for the Satellite Flash requires the following setting:

```
#set      SATELLITE_FLASH            ON ; <<< select if Satellite Flash
is available
#set      SATELLITE_SECURITY_ENABLE  ON ; <<< enable Flash Security for
Satellite Flash
```

These settings enable the Flash Security. In this case, it is not possible to disable the feature using an Unlock Key. To allow this unlocking, the desired Unlock Key needs to be specified. This can be done separately for the Main Flash Memory and the Satellite Flash Memory. An example for the Main Flash Memory is shown below:

```
; set the Flash Security unlock key (16 bytes)
; all 0: unlock not possible
#set      MAIN_UNLOCK_0              0x01
#set      MAIN_UNLOCK_1              0x23
#set      MAIN_UNLOCK_2              0x45
#set      MAIN_UNLOCK_3              0x67
#set      MAIN_UNLOCK_4              0x89
#set      MAIN_UNLOCK_5              0xAB
#set      MAIN_UNLOCK_6              0xCD
#set      MAIN_UNLOCK_7              0xEF
#set      MAIN_UNLOCK_8              0x01
#set      MAIN_UNLOCK_9              0x23
#set      MAIN_UNLOCK_10             0x45
#set      MAIN_UNLOCK_11             0x67
#set      MAIN_UNLOCK_12             0x89
#set      MAIN_UNLOCK_13             0xAB
#set      MAIN_UNLOCK_14             0xCD
#set      MAIN_UNLOCK_15             0xEF
```

2.3.2 Manual Configuration

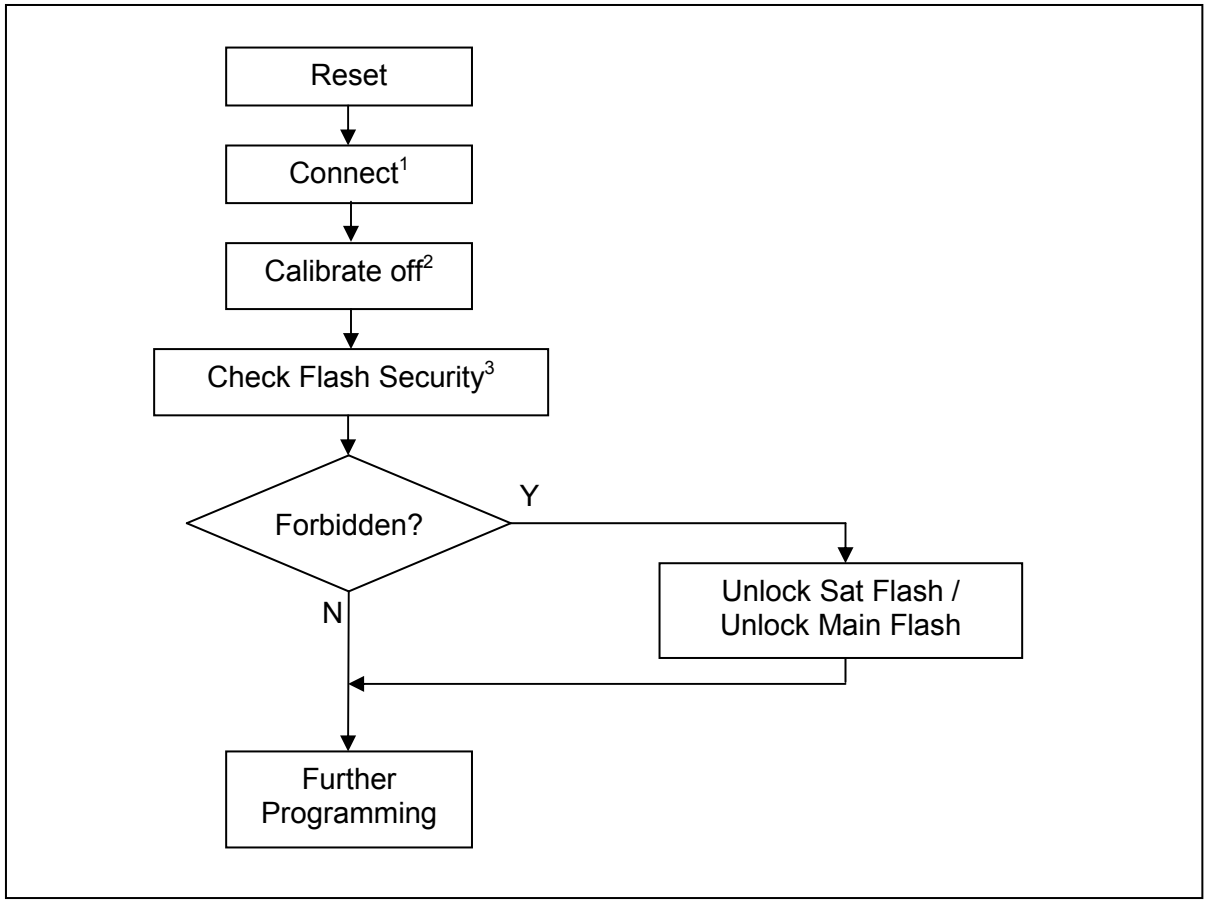
The Flash Security feature can be configured by writing certain patterns at specific Flash Memory locations. The table below details these locations.

	Main Flash Memory	Satellite Flash Memory
Flash Security Byte	MFSB @ 0xDF0000	SFSB @ 0xDE0000
Enable protection	MFSB = 0x99	SFSB = 0x99
Disable Protection	Set MFSB to any value but 0x99	Set SFSB to any value but 0x99
Security Unlock Key location	0xDF0002 – 0xDF0011	0xDE0002 – 0xDE0011

2.4 Unlocking the Flash Security

The Flash Security feature can be unlocked temporarily by communicating with the Boot-ROM. For details refer to the Application Note MCU-AN-300224-E “MCU Flash Programming and Boot-ROM Protocol”.

The following flow chart shows the flow of sequence of communication from a PC to the Boot ROM in order to unlock the Flash Security.



When the communication is established and flash is memory is secured, the UNLOCK command must used to unlock it. The format of the command is as follows:

Header		Payload					Check-sum
Cmd	**Select**	**Key 0**	**Key 1**	**…**	**Key 14**	**Key 15**	
0x0A	*0xXX*	*Key0*	*Key1*	…	*Key14*	*Key15*	*0xYY*
PC → MCU							

Select: selects the ROM/Flash module. (0x00: Main ROM/Flash, 0x01: Satellite Flash.)

Key: is the 16-Byte code to be compared against stored key.

[1,2,3] Further information about these command can be found in

MCU-AN-300224-E “MCU Flash Programming and Boot-ROM Protocol”

Checksum: is calculated with the following equation. It includes all bytes (*n*) except the calibration header[4] (`0x00`, `0x55`):

$$Checksum = 0xFF - \left(\sum_{i=1}^{n} data_i\right) \bmod 0x100 - \left\lfloor \frac{\sum_{i=1}^{n} data_i}{0x100} \right\rfloor - \left\lfloor \frac{\sum_{i=1}^{n} data_i}{0x10000} \right\rfloor$$

The last two summands are correction values, if the overall sum crosses each `0x100` boundary and the `0x10000` boundary.

For the unlocking Satellite Flash with Unlock Key as shown in section 2.3.1, checksum can be calculated as follows:

$$Sum = 0x0A + 0x01 + 0x01 + 0x23 + 0x45 + 0x67 + 0x89 + 0xAB + 0xCD + 0xEF + 0x01 + 0x23 + 0x45 + 0x67 + 0x89 + 0xAB + 0xCD + 0xEF$$

$$Checksum = 0xFF - (Sum \mod 0x100) - \left\lfloor \frac{Sum}{0x100} \right\rfloor$$

$$Checksum = 0xFF - (0x78B \mod 0x100) - \left\lfloor \frac{0x78B}{0x100} \right\rfloor$$

$$Checksum = 0xFF - 0x8B - 0x07 = 0x6D$$

Depending on the result, the following response and behaviour must be expected:

- The ROM/Flash Memory is unlocked successfully:

Response
`0x69`
MCU → PC

  After this response, all other commands are allowed.

- If the ROM/Flash Memory cannot be unlocked because the saved Unlock Key is all 0:

Response
`0x96`
MCU → PC

  After this response, serial commands are still handled.

- If the ROM/Flash Memory cannot be unlocked because the wrong Unlock Key was transmitted:

Response
`0x96`
MCU → PC

  After this response, no more commands are handled. The MCU must be reset.

[4] Further detail about calibration header can be found in application note MCU-AN-300224-E “MCU Flash Programming and Boot-ROM Protocol”

3 Appendix

FURTHER INFORMATION

Information about FUJITSU Microcontrollers can be found on the following Internet page:

http://mcu.emea.fujitsu.com/

3.1 Related Documents

- *MCU-AN-300224-e-16fx_mcu_flash_prog_boot_rom*
  This application note shows flash programming and boot ROM mechanism.