

Abstract

This application note describes how to use the Keil™ MDK-ARM toolkit with the Freescale's Kinetis devices that are based on a Cortex™-M4 processor. This note focuses on the configuration of the FlexMemory.

MDK-ARM Version 4.13a or later is required for Kinetis device support. This version also contains the example projects that are referred in this application note.

Contents

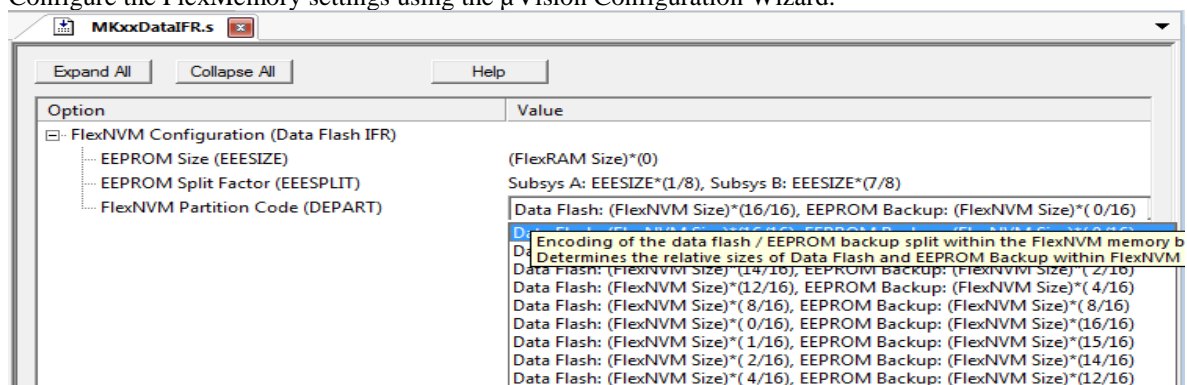
Abstract.....	1
FlexMemory Configuration	1
Flash Configuration Field & MCU Security.....	2
Program Once Field	3
Using µ Vision Multi-Project	4
Other Useful Information.....	5
Revision History	5

FlexMemory Configuration

Freescale's Kinetis device offers FlexMemory technology, which provides versatile and powerful solutions for flexible memory usage. FlexMemory is composed of FlexNVM and FlexRAM. FlexNVM is a non-volatile flash memory that can execute program code, store data, or backup EEPROM data. FlexRAM can be used as standard RAM or high-endurance EEPROM.

A µ Vision project shows the usage of the FlexMemory Configuration:

1. Open the project file **FlexMem_Cfg.uvproj** located in the folder **..\ARM\Boards\Freescale\TWR-Kxxxxxx\FlexMem_Cfg**.
2. Open the file **MKxxFlashIFR.s**.
3. Configure the FlexMemory settings using the µ Vision Configuration Wizard.



4. Build the Target.
5. Click the button **LOAD**, to download the application to Flash.

Note:

- All memory content (Program Flash and FlexNVM) is erased with the steps above.

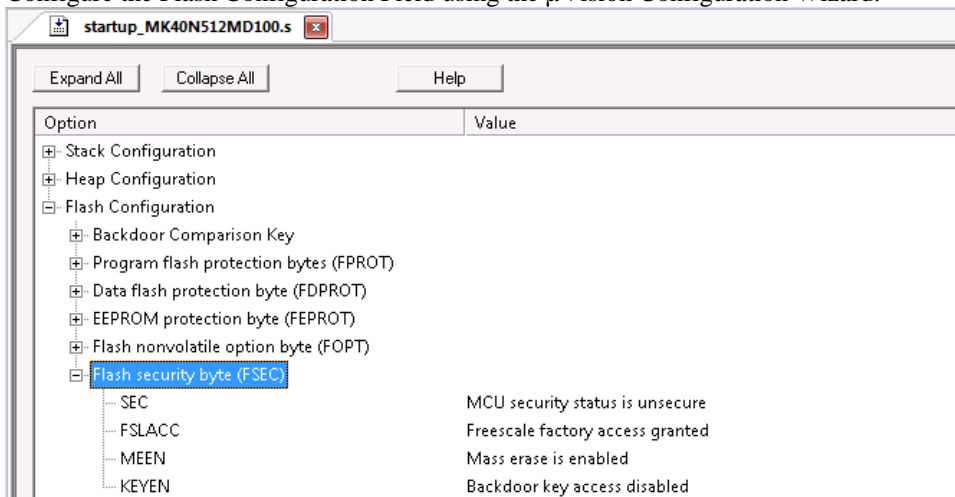
Flash Configuration Field & MCU Security

Program flash memory contains a 16-byte Flash Configuration Field that stores default protection settings and security information allowing the MCU to restrict access to the FTFL module. During reset, the FTFL module initializes the FSEC register using the security byte data of the Flash Configuration Field.

Configuration of Flash Configuration Field

A μ Vision project shows the usage of the Flash Configuration Field:

1. Open project file **Blinky.uvproj** located in the folder `..\ARM\Boards\Freescale\ TWR-Kxxxxxx\Blinky`.
2. Open the startup file **startup_MKxxxxxxx.s**
3. Configure the Flash Configuration Field using the μ Vision Configuration Wizard.



4. Build the Target.
5. Click the button **LOAD**, to download to Flash.

WARNING:

- If, in the section **Flash security byte (FSEC)**, the field SEC is configured as "MCU security status is secure" and the field MEEN is configured as "Mass erase is disabled", the MCU's security status cannot be reset to unsecure state because even Mass Erase via the debugger is blocked!

Debugger access when MCU security is secure

If field SEC is set to "MCU security status is secure", most debugging functionalities, including "Flash Programming" and "Start a debug session", are disabled. If the device has to be set back to unsecure state, Mass Erase via the debugger can be executed when field MEEN is set to "Mass erase is enabled". When executing "Flash Programming" or "Start a debug session", a warning informs that a Mass Erase has to be executed to use normal debugging capabilities.



Note:

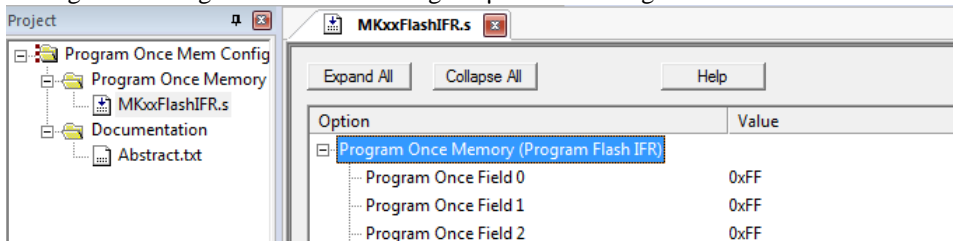
- Mass Erase via the debugger erases all contents located in Program Flash, FlexNVM, including the FlexNVM configuration information located in Data Flash IFR.
- To unsecure the MCU, the field SEC must be configured as "MCU security status is unsecure".

Program Once Field

The **Program Once Field** is a user data storage of 64 bytes, separated from the program flash main array. The **Program Once Field** can be programmed only once as there is no program flash IFR erase mechanism available to the user.

A μ Vision project shows the usage of the Program Once Field:

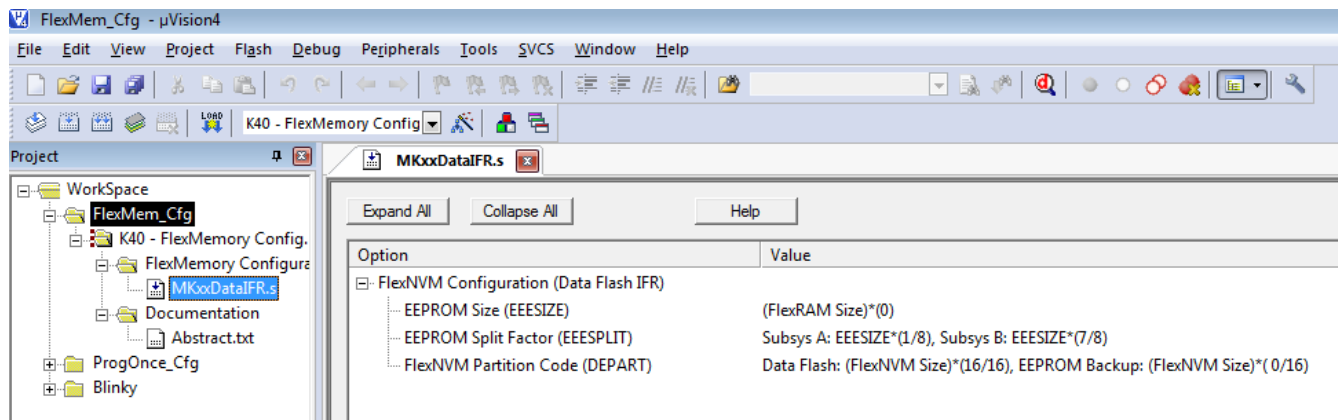
1. Open the project file **ProgOnce_Cfg.uvproj** located in the folder `..\ARM\Boards\Freescale\TWR-Kxxxxxx\ProgOnce_Cfg`.
2. Open the file **MKxxFlashIFR.s**
3. Configure the Program Once Field using the μ Vision Configuration Wizard.



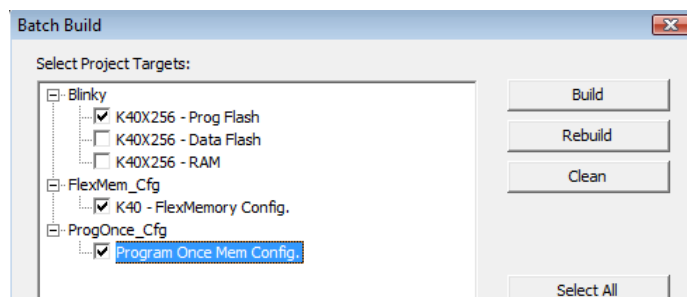
4. Build the Target.
5. Click the button **LOAD**, to download to flash.

Using µVision Multi-Project

Individual projects can be combined into a Multi-Project for a better application overview.



Configure each project individually. The **Batch Build** feature builds the targets in the order established in the Project Window.



Other Useful Information

Using FlexRAM as Read/Write Area

Define the FlexRAM start address and size in the **Read/Write Memory Areas** fields to use the memory area globally for the application.

Read/Write Memory Areas				
default	off-chip	Start	Size	NoInit
<input checked="" type="checkbox"/>	RAM1:	0x14000000	0x800	<input type="checkbox"/>
<input type="checkbox"/>	RAM2:			<input type="checkbox"/>
<input type="checkbox"/>	RAM3:			<input type="checkbox"/>
on-chip				
<input checked="" type="checkbox"/>	IRAM1:	0x1FFF8000	0x8000	<input type="checkbox"/>
<input checked="" type="checkbox"/>	IRAM2:	0x20000000	0x8000	<input type="checkbox"/>

Using Data Flash of FlexNVM as Read/Only Area

Define the Data Flash start address and size in the **Read/Only Memory Areas** fields to use the memory area globally for the application. Data Flash can be used for code execution or static data storage.

Read/Only Memory Areas				
default	off-chip	Start	Size	Startup
<input type="checkbox"/>	ROM1:			<input type="radio"/>
<input type="checkbox"/>	ROM2:			<input type="radio"/>
<input type="checkbox"/>	ROM3:			<input type="radio"/>
on-chip				
<input checked="" type="checkbox"/>	IROM1:	0x0	0x40000	<input checked="" type="radio"/>
<input checked="" type="checkbox"/>	ROM2:	0x10000000	0x40000	<input type="radio"/>

Revision History

- Nov. 2010: Initial Version