

TOSHIBA

**TMPM4K4Group
Data Flash Emulation
Sample Program Manual**

Rev 1.0
September 2018

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Preface

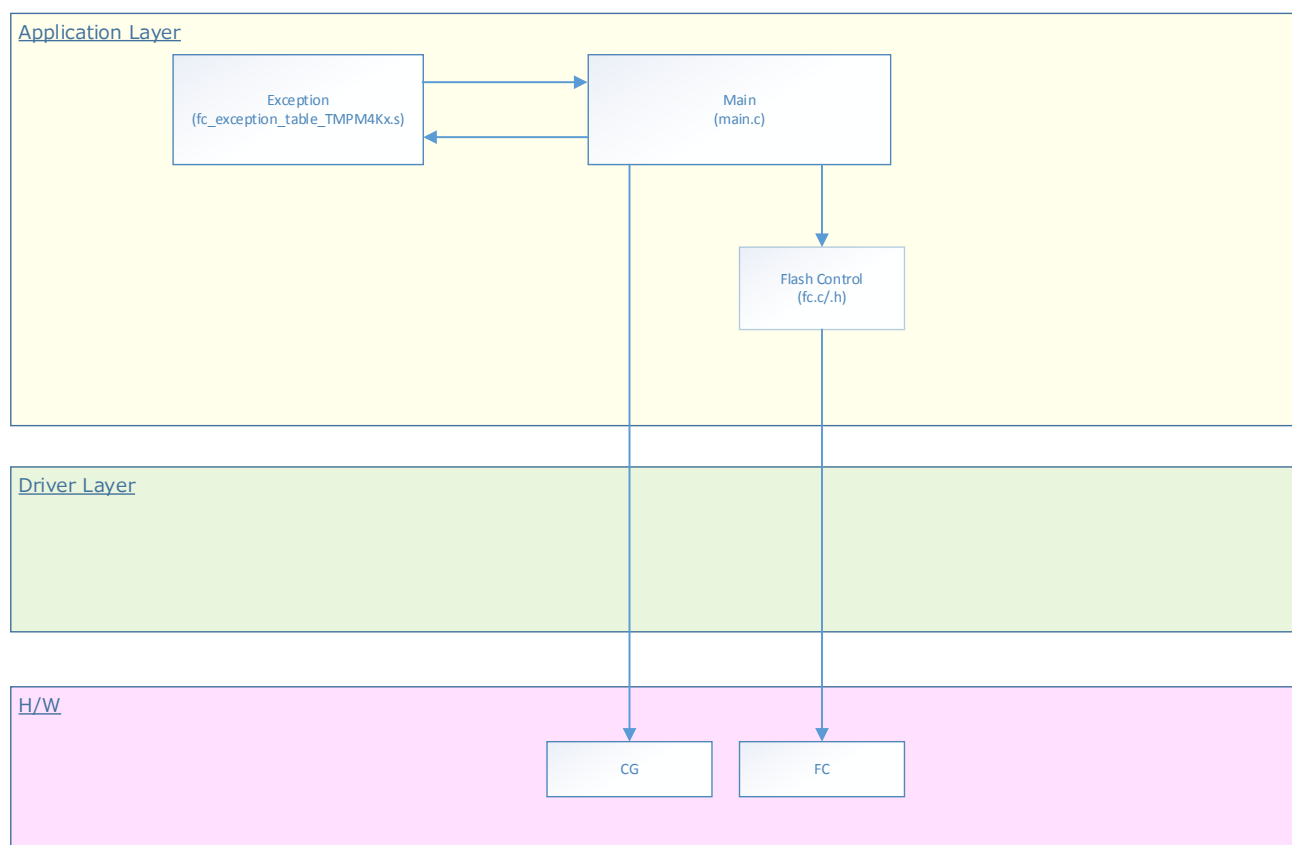
This document is a reference material for developing products that use Code Flash as Data Flash using the M4K Group (1).

This document helps the user check operation of the product and develop its program

Target sample program: CodeFlashE2PROMEmulation

1. Overview

The sample program described in this document uses the Code Flash as a substitute of the Data Flash, controlling the number of writing times and the writing area of the Code Flash.



2. Reference Document

- Datasheet
 TPM4K Group (1) datasheet Rev3.0
- Reference manual
 Flash Memory Code Flash: 265KB/128KB/96KB/64KB (FLASH256-B) Rev1.1
 (Japanese edition)

3. Function to Use

Function	Operation mode
Flash Memory	Automatic program

4. Target Device

The target devices of application note are as follows.

TPM4K4FYAUG	TPM4K4FWAUG	TPM4K4FUAUG	TPM4K4FSAUG
TPM4K4FYAFG	TPM4K4FWAFG	TPM4K4FUAFG	TPM4K4FSAFG
TPM4K2FYADUG	TPM4K2FWADUG	TPM4K2FUADUG	TPM4K2FSADUG
TPM4K1FYAUG	TPM4K1FWAUG	TPM4K1FUAUG	TPM4K1FSAUG
TPM4K0FSADUG			

5. Description of Software

5.1. Operation Overview of the sample program

This sample program uses the Code Flash area as a substitute of the Data Flash for data storage. The sample program tries to keep the number of writing time equal to that of EEPROM, by shifting the addresses of the variables when data write, and erasing by page unit when data erase.

5.2. Parameters Setting

User can set the following parameters.

Start page of Flash write

Number of using pages (page size is 4KB)

Write data size (maximum 4079 bytes)

Write data storage address

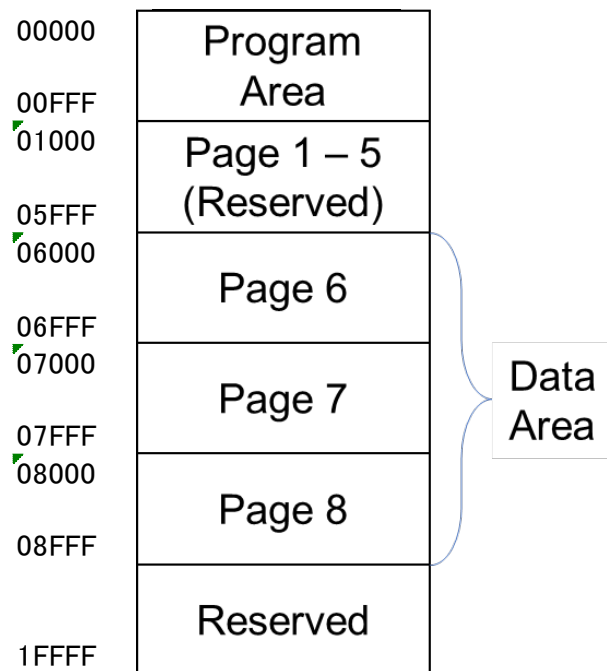
5.3. Operation Example (in case of 128KB Code Flash)

Start page of Flash write =0x06

Number of using pages =3

Write data size =100byte

Write data storage address =0x2000xxxx0 (User Data)



Memory Map

When writing 100 byte data, the program divides a page by the unit of 112 bytes into 36 Records, because the write unit is 16 bytes. It writes “0xFF” to the first 11 bytes in the residual 12 bytes, and writes the written mark: 0xA5 at the last 1 byte. When the unit of the data write is a multiple of 16 bytes, the program increases the Record size by 16 bytes, and writes “0xFF” to the first 15 bytes in the increased 16 bytes and written mark: 0xA5 to the last 1 byte, respectively. The program uses the last 16 bytes in a page as “Record D” for the storage of the write times count.

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0	
06000	0							1							2		
06100	2					3							4				
06200	4			5							6						
06300	6	7							8							9	
06400	9						10							11			
06500	11				12							13					
06600	13		14						15								
06700	16							17							18		
06800	18					19							20				
06900	20			21							22						
06A00	22	23							24							25	
06B00	25						26							27			
06C00	27				28							29					
06D00	29		30						31								
06E00	32							33							34		
06F00	34					35							Reserved			D	

Mapping in a page (in case of page 6)

Write Operation

The program reads the data from the address of 0x20000000, and write them to the Flash sequentially from “page 6, Record 0”.

When it has written the data to “page 6, Record 35” (36 Records), it moves to the next page. After finishing to write to “page 8, Record 35” (108 Records), it erases the page 6 and start to write from “page 6, Record 0”.

When multiple pages are set for the number of using pages, the program will erase the first page and start the second write after finishing to write all pages.

Write Times Control

At the first write, the program writes “0x0000” to the last 4 bytes in the “Record D”. After the data erase before the second write, the program write “0x0001” to the last 4 bytes of the “Record D”. The program controls the write times by performing this operation to each page. After the write times control data, “Record D” reaches at “0x2710” (10000 times), erase operation will not be performed.

Note

(1) When the write to the Record, the program will not write the data if the User Data is same as the User Data of the Cord Flash

(2) Behavior during Code Flash ROM Operation

When the program is running on RAM, it uses the vector table for RAM. It disables exceptions and interrupts, and accepts the exceptions which cannot be disabled using the vector table for RAM. All the exception handler implements an infinite loop.

(3) Code Flash ROM Write/Erase Failure

This program does not implement retry for the failure, and just returns an error. In order not to control the unusable Record, the functions of this program will be disabled when an error occurs. After successful restore at Reset, the restored User Data can be read from the User Data area.

(4) Behavior after Reset

After the successful restore at Reset, EEPROM emulation function gets usable. The restored User Data can be read from the User Data area.

6. Appendix

The record format

