eYs3D

Flash Protection APN

FW Div.

V2.0
20210906

History

Version.	Date	Description
V1.0	20210824	• Initial release
V2.0	**20210906**	**• Create user file id as the shadow of factory file id, p.19** **• Add file id 201 as a backup of UNP, p.20**

Theory

1. Flash type we use is MX25R series
   a. MX25R2035F, 2M bits (256K Bytes)
   b. MX25R4035F, 4M bits (512K Bytes)
   c. MX25R8035F, 8M bits (1M Bytes)
2. Flash protection is block base. Each block is **64KB**.
3. Flash protection scheme is separated as program data and user data.
   a. Program data and protected parameters need to be protected by flash chip level.
   b. User data is unprotected.
4. Flash protection blocks must start at lowest address and the size must be able to store all protected data, item 2.a. (See p.9)
5. $BP_{3\sim0}$ is protection bits in flash. BP stands for Block Protection. There are 4 bits for this setting. $BP_{3\sim0}$ determines protection blocks (See pp.10 - 12).
6. Numbers of protection blocks, N = BPX > 0 ? 2^{BPX-1} : 0, BPX = $BP_{3\sim0}$,(N = 0,1,2,4,8…) , N <= Total blocks

Block Top/Bottom

1. TB bit sets protection block from top to bottom or bottom to top.

Protected Area Sizes (TB bit = 0)

Status bit				Protect Level
BP3	BP2	BP1	BP0	
0	0	0	0	0 (none)
0	0	0	1	1 (1block, block 3rd)
0	0	1	0	2 (2blocks, block 2nd~3rd)
0	0	1	1	3 (4blocks, protect all)
0	1	0	0	4 (4blocks, protect all)
0	1	0	1	5 (4blocks, protect all)
0	1	1	0	6 (4blocks, protect all)
0	1	1	1	7 (4blocks, protect all)
1	0	0	0	8 (4blocks, protect all)
1	0	0	1	9 (4blocks, protect all)
1	0	1	0	10 (4blocks, protect all)
1	0	1	1	11 (4blocks, protect all)
1	1	0	0	12 (4blocks, protect all)
1	1	0	1	13 (4blocks, protect all)
1	1	1	0	14 (4blocks, protect all)
1	1	1	1	15 (4blocks, protect all)

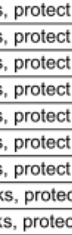
High

Low

Protected Area Sizes (TB bit = 1)

Status bit				Protect Level
BP3	BP2	BP1	BP0	
0	0	0	0	0 (none)
0	0	0	1	1 (1block, block 0th)
0	0	1	0	2 (2blocks, block 0th-1st)
0	0	1	1	3 (4blocks, protect all)
0	1	0	0	4 (4blocks, protect all)
0	1	0	1	5 (4blocks, protect all)
0	1	1	0	6 (4blocks, protect all)
0	1	1	1	7 (4blocks, protect all)
1	0	0	0	8 (4blocks, protect all)
1	0	0	1	9 (4blocks, protect all)
1	0	1	0	10 (4blocks, protect all)
1	0	1	1	11 (4blocks, protect all)
1	1	0	0	12 (4blocks, protect all)
1	1	0	1	13 (4blocks, protect all)
1	1	1	0	14 (4blocks, protect all)
1	1	1	1	15 (4blocks, protect all)

Low



High

BPX description for 2Mb/256KB

1. To meet the criteria, MX25R2035F (2Mb/256KB), can be configured as lower 2 blocks (0^{th} ~ 1^{st} /128KB) are protected, higher 2 blocks (2^{nd} ~ 3^{rd} /128KB) are unprotected.

Protected Area Sizes (TB bit = 1)

Status bit				Protect Level
BP3	BP2	BP1	BP0	
0	0	0	0	0 (none)
0	0	0	1	1 (1block, block 0th)
0	0	1	0	2 (2blocks, block 0th-1st)
0	0	1	1	3 (4blocks, protect all)
0	1	0	0	4 (4blocks, protect all)
0	1	0	1	5 (4blocks, protect all)
0	1	1	0	6 (4blocks, protect all)
0	1	1	1	7 (4blocks, protect all)
1	0	0	0	8 (4blocks, protect all)
1	0	0	1	9 (4blocks, protect all)
1	0	1	0	10 (4blocks, protect all)
1	0	1	1	11 (4blocks, protect all)
1	1	0	0	12 (4blocks, protect all)
1	1	0	1	13 (4blocks, protect all)
1	1	1	0	14 (4blocks, protect all)
1	1	1	1	15 (4blocks, protect all)

BPX description for 4Mb/512KB

2. If 128KB protected size is not enough, it can upgrade to 4Mb, MX25R4035F (4Mb/512KB). This part number can be configured as lower 4 blocks (0^{th} ~ 3^{rd} /256KB) are protected, higher 4 blocks (4^{th} ~ 7^{th} /256KB) are unprotected.

Protected Area Sizes (TB bit = 1)

Status bit				Protect Level
BP3	BP2	BP1	BP0	
0	0	0	0	0 (none)
0	0	0	1	1 (1block, block 0th)
0	0	1	0	2 (2blocks, block 0th-1st)
0	0	1	1	3 (4blocks, block 0th-3th)
0	1	0	0	4 (8blocks, protect all)
0	1	0	1	5 (8blocks, protect all)
0	1	1	0	6 (8blocks, protect all)
0	1	1	1	7 (8blocks, protect all)
1	0	0	0	8 (8blocks, protect all)
1	0	0	1	9 (8blocks, protect all)
1	0	1	0	10 (8blocks, protect all)
1	0	1	1	11 (8blocks, protect all)
1	1	0	0	12 (8blocks, protect all)
1	1	0	1	13 (8blocks, protect all)
1	1	1	0	14 (8blocks, protect all)
1	1	1	1	15 (8blocks, protect all)

BPX description for 8Mb/1MB

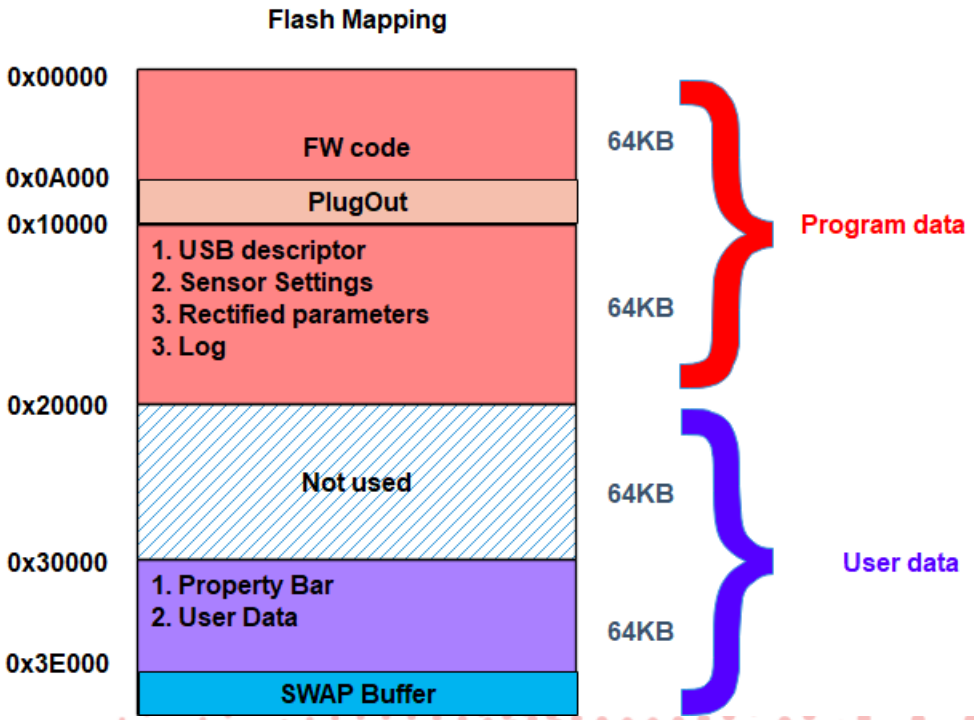
3. This is 8Mb(1MB), MX25R8035F, configuration.

Protected Area Sizes (TB bit = 1)

Status bit				Protect Level
BP3	BP2	BP1	BP0	8Mb
0	0	0	0	0 (none)
0	0	0	1	1 (1block, block 0th)
0	0	1	0	2 (2blocks, block 0th-1st)
0	0	1	1	3 (4blocks, block 0th-3rd)
0	1	0	0	4 (8blocks, block 0th-7th)
0	1	0	1	5 (16blocks, protect all)
0	1	1	0	6 (16blocks, protect all)
0	1	1	1	7 (16blocks, protect all)
1	0	0	0	8 (16blocks, protect all)
1	0	0	1	9 (16blocks, protect all)
1	0	1	0	10 (16blocks, protect all)
1	0	1	1	11 (16blocks, protect all)
1	1	0	0	12 (16blocks, protect all)
1	1	0	1	13 (16blocks, protect all)
1	1	1	0	14 (16blocks, protect all)
1	1	1	1	15 (16blocks, protect all)

Flash Mapping

1. Protected parameters:
    1. File table structure
    2. Sensor settings
    3. USB description table
    4. Y-offset (file id 3x)
    5. Rectify table (file id 4x)
    6. ZD Table (file 5x)
    7. Log file (file 24x)
2. Property bar locates at unprotected lock area.

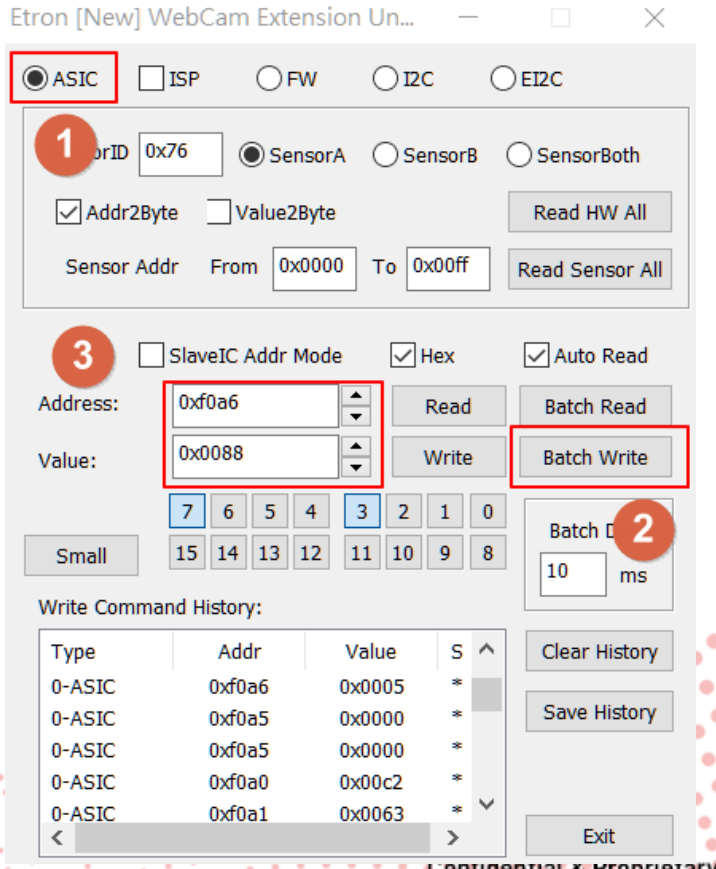


Read BPX

Status Register

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SRWD (status register write protect)	QE (Quad Enable)	BP3 (level of protected block)	BP2 (level of protected block)	BP1 (level of protected block)	BP0 (level of protected block)	WEL (write enable latch)	WIP (write in progress bit)
1=status register write disable	1=Quad Enable 0=not Quad Enable	(note 1)	(note 1)	(note 1)	(note 1)	1=write enable 0=not write enable	1=write operation 0=not in write operation
Non-volatile bit	Non-volatile bit	Non-volatile bit	Non-volatile bit	Non-volatile bit	Non-volatile bit	volatile bit	volatile bit

1. Check BP status by CapW:
    1. Set ASIC type
    2. Use Batch Write to load script file, RD_Flash_STS_(SPI_RDSR_CMD).txt, to read status register
    3. Set the address to 0xF0A6 to read the content
        1. [0xF0A6] = 0x88 >> BPX = 0010.
2. Refer to slide: Improvement 2/6, Improvement 3/6
3. Note: It needs to reload script file to read the register again

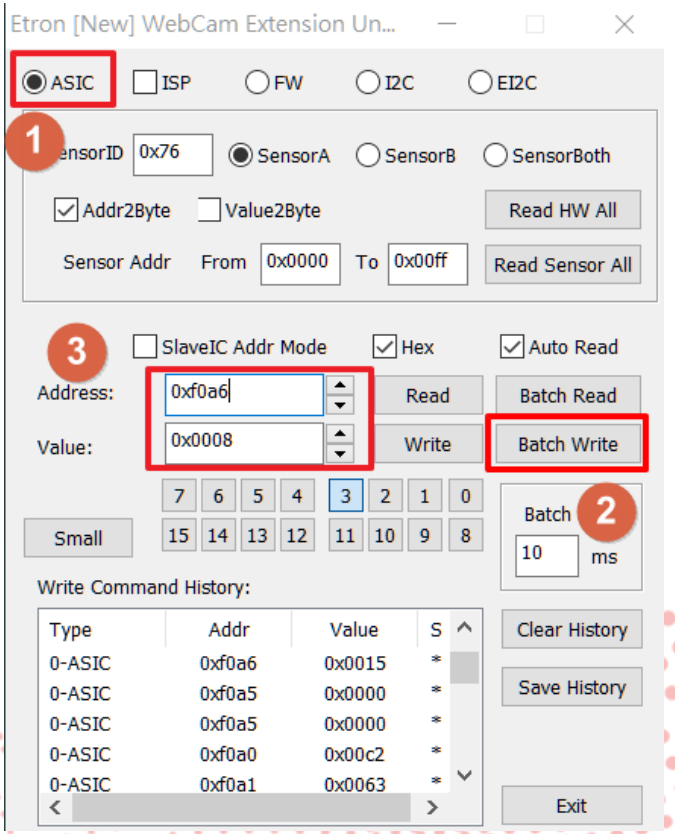


Read Top/Bottom Bit

Configuration Register - 1

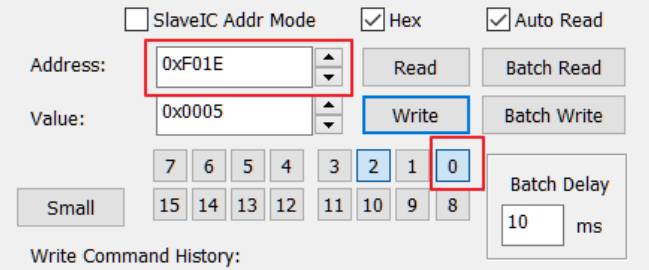
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Reserved	DC (Dummy Cycle)	Reserved	Reserved	TB (top/bottom selected)	Reserved	Reserved	Reserved
x	2READ/ 4READ Dummy Cycle	x	x	0=Top area protect 1=Bottom area protect (Default=0)	x	x	x
x	Volatile bit	x	x	OTP	x	x	x

1. Check Top/Bottom setting by CapW
   1. Set ASIC type
   2. Use Batch Write to load script file, RD_Flash_Configuration_(SPI_RDCR_CMD).txt, to read configuration register
   3. Set the address to 0xF0A6 to read the content
      1. [0xF0A6] = 0x08 >> bottom area
2. Refer to slide: Improvement 2/6, Improvement 3/6
3. Note: It needs to reload script file to read the register again

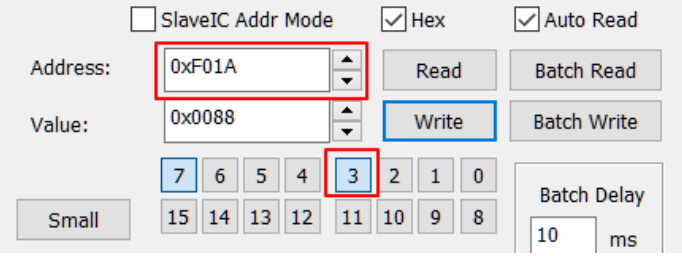


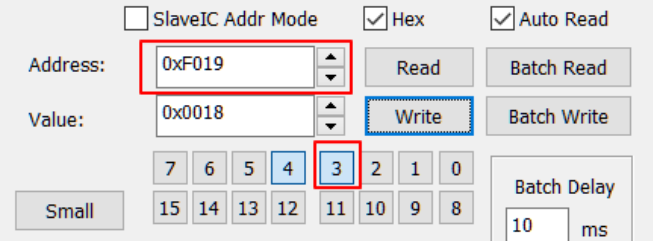
Unprotect Flash

1. Do the following steps to unprotect flash
   1. Set GPIO23 from UART RX to GPIO mode
      1. Set bit 0 of 0xF01E to 1



   2. Set GPIO23 as output
      1. Set bit 3 of 0xF01A to 1
      2. Set bit 3 of 0xF019 to 1





   3. Use batch write to load script file, Flash_Wr_Protect_None -00.txt, to unprotect flash.
   4. Read BPX to check the status. $BP_{3\sim0}$ should be cleared.

Write on protected flash

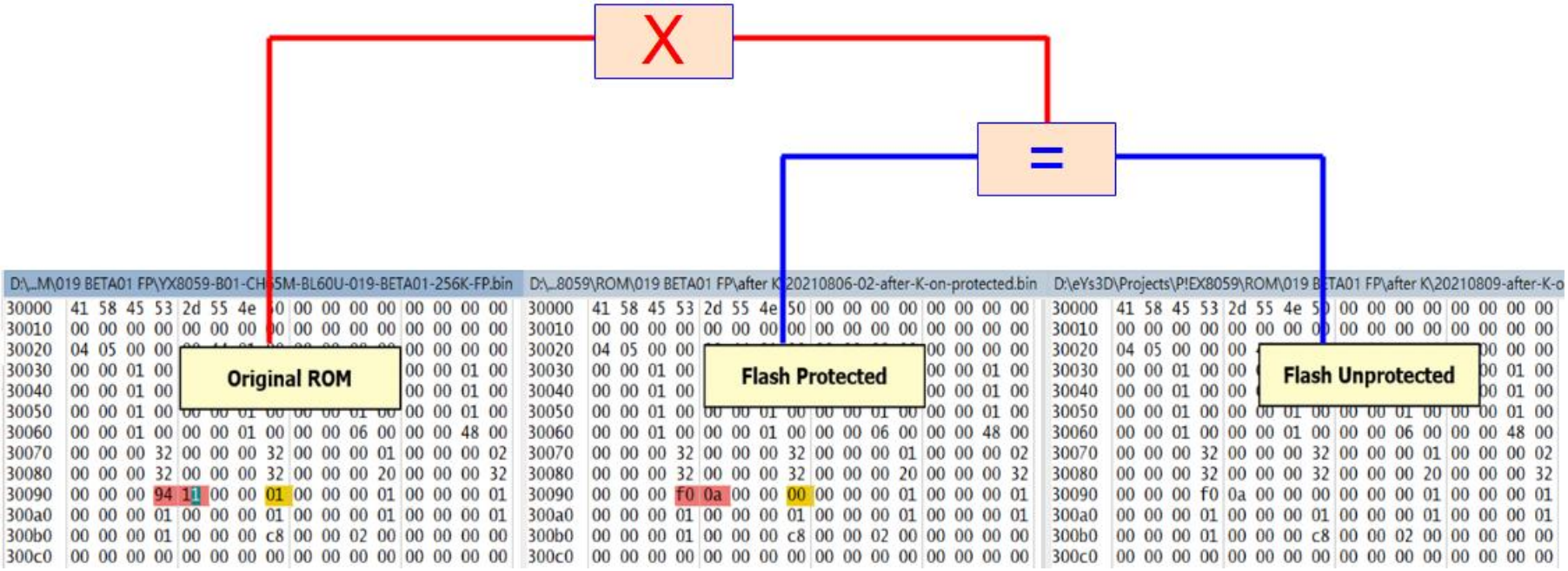
1. If you find you can't write to flash as you want like calibration, please run unprotect flash procedure.
2. Run your tool like calibration to do your job.

Note: ***Don't remove the module after you unprotect flash. Or, the flash will be protected again when the module is powered on.***

Test result

	Protected Flash	Unprotected Flash
Property Bar/ User Data	V	V
Calibration / Program Data	X	V

Property bar test



Calibration test

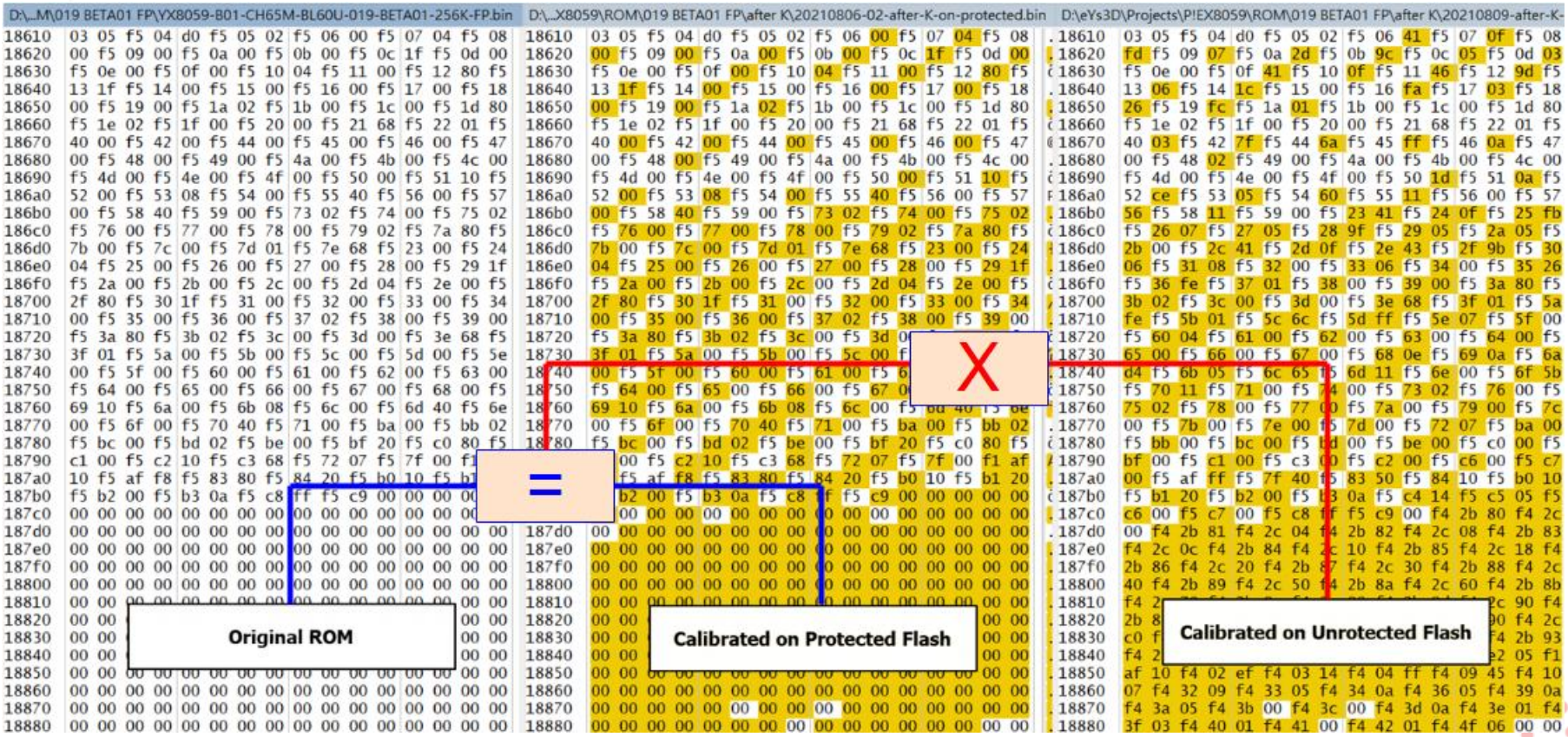


Table explanation

2. Header

Offset	Field	Size	Description
0x00	Signature	8	String "AXES-CFG"
0x08	Version	2	1 for initial release.
0x0A	STI_Offset	2	Offset of "Storage Type Information" 0xFFFF means use default SPI Flash Storage

- Find the STI offset from header table illustrated at the left table, and check the StructLen value.

Old structure

11 Storage Type Information

Offset	Field	Size	Description
0	StructLen	1	Struct Length of Storage Information(不包含長度欄位本身，因此目前為 11).
1	SpiPageShift	1	Power of SPI Page Size. (Page: minimal writing size) It should be 8 (256) now
2	SpiSecShift	1	Power of SPI Sector Size. (Sector: minimal erasing size)
3	Reserved	1	Always set as 32
4	SpiTotalSec	2	SPI Total Sector Number.
6	UserAreaStartSec	2	Start Sector Number of User Area. Set as 0 to Disabled User Area.
8	UserAreaEndSec	2	End Sector Number of User Area
10	SpiWpGpioNum	1	SPI Flash Write Protection GPIO Register (Low Byte)
11	SpiWpGpioPin	1	SPI Flash Write Protection GPIO Pin
…	…		

New structure

3. Storage Type Information

Offset	Field	Size	Description
0	StructLen	1	Struct Length of Storage Information(不包含長度欄位本身，因此目前為 17).
1	SpiPageShift	1	Power of SPI Page Size. (Page: minimal writing size) It should be 8 (256) now
2	SpiSecShift	1	Power of SPI Sector Size. (Sector: minimal erasing size)
3	Reserved	1	Always set as 32
4	SpiTotalSec	2	SPI Total Sector Number.
6	UserAreaStartSec	2	Start Sector Number of User Area. Set as 0 to Disabled User Area.
8	UserAreaEndSec	2	End Sector Number of User Area
10	SpiWpGpioNum	1	SPI Flash Write Protection GPIO Register (Low Byte)
11	SpiWpGpioPin	1	SPI Flash Write Protection GPIO Pin
12	UnpAreaStartSec	2	UnProtection Area Start Sector Set as 0 to Disabled UNP Area
14	UupAreaSizeShift	1	Power of UnProtection Area Size (Ex. 16=64K)
15	SpiWpSts	1	SPI Flash Write Protection Status
16	SpiInitCfg	2	SPI Flash Initial Config Value
…	…		

Protected bin file check

1. If the StructLen is 0x0B, this bin file doesn't support flash protection.
2. If it is other than 0x0B, 0x11 for current case, this bin file supports flash protection.
    1. The first 128KB is protected.
    2. If UnpAreaStartSec is 0, this bin file doesn't have unprotected area. If it is not 0, the unprotected address starts at N * 4096. N is UnpAreaStartSec value.
    3. For example, the current setting is 48 x 4096 = 196608 = 0x30000. The UNP starts at address 0x30000.

Protected bin file example

Without flash protection scheme

Header

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00010000	41	58	45	53	2D	43	46	47	00	19	0D	12	00	20	01	00
00010010	0C	DC	08	DE	7A	A0	7A	84	FF	FF	0D	1E	00	00	00	00
00010020	03	02	01	0F	08	00	00	00	09	00	00	00	01	00	00	00
00010030	08	00	00	00	03	01	03	00	00	00	00	01	00	00	00	00
00010040	00	00	00	01	00	00	00	04	04	0F	01	00	00	00	01	00

Storage Type Information

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00010D10	0D	12	0B	08	0C	20	00	20	00	00	00	00	19	03	36	00
00010D20	00	00	00	EC	01	00	EC	02	CA	02	03	B6	01	F3	03	05
00010D30	A9	00	E4	04	06	8D	03	9F	05	0A	2C	03	A8	06	0D	D4
00010D40	06	6E	07	14	42	02	99	08	16	DB	03	97	09	1A	72	01
00010D50	0E	0A	1B	80	00	E5	0B	1C	65	02	93	0C	1E	F8	01	C4

With flash protection scheme

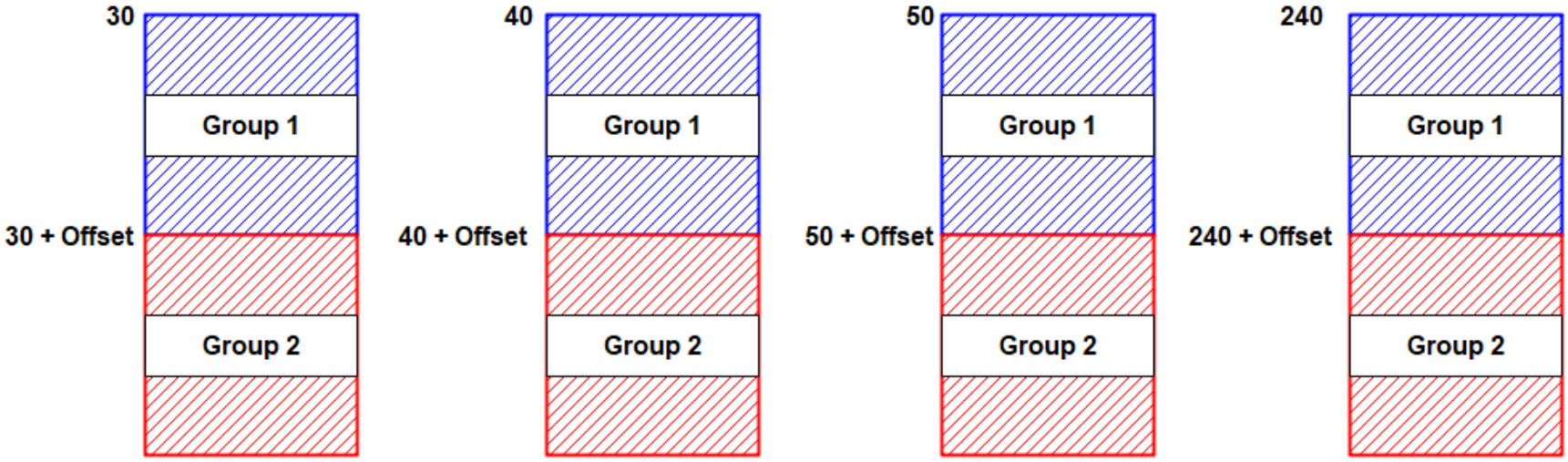
Header

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00010000	41	58	45	53	2D	43	46	47	00	1C	0D	44	00	20	01	00
00010010	0D	0E	09	10	7C	D6	7C	BA	FF	FF	0D	56	00	00	00	00
00010020	03	02	01	0F	08	00	00	00	09	00	00	00	01	00	00	00
00010030	08	00	00	00	03	01	03	00	00	00	00	01	00	00	00	00
00010040	00	00	00	01	00	00	00	04	04	0F	01	00	00	00	01	00

Storage Type Information

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00010D40	00	00	0D	44	11	08	0C	20	00	40	00	00	00	00	19	03
00010D50	00	30	10	88	00	08	36	00	00	00	00	EC	01	00	EC	02
00010D60	CA	02	03	B6	01	F3	03	05	A9	00	E4	04	06	8D	03	9F
00010D70	05	0A	2C	03	A8	06	0D	D4	06	6E	07	14	42	02	99	08
00010D80	16	DB	03	97	09	1A	72	01	0E	0A	1B	80	00	E5	0B	1C

Factory and user settings



1. The gropu 1 is the factory settings which are calibrated before shipment. The group 2 is user settings which are the shadow of factory settings after post calibration.
2. FW Register 0xF6 is the offset.
3. The final file ID is loaded = file id + offset which is tored at 0xF6.
4. The default offset is set as 5 which means 10 is divided by 2 (groups) = 5
5. For example, 30/40/50/240 are factory settings, 35/45/55/245 are user settings
6. File ID:
   030 ~ 039: Y-Offset
   040 ~ 049: Rectification
   050 ~ 059: ZD table
   240 ~ 249: Log

Backup UNP data

1. The UNP (**UNP**rotect ion) area still has a chance to be destroyed. Hence, data in this area will be compressed and saved in the file id 201.
2. If the application needs to restore to factory settings for UNP area, it can get UNP backup from file id 201 and restore to UNP area.
3. Default length of file id 201 is 512 bytes.

Script files

Flash_Wr_Protect_None -00.txt

RD_Flash_Configuration_(SPI_RDCR_CMD).txt

RD_Flash_STS_(SPI_RDSR_CMD).txt

Reference

1. Axes Plugin Data Binary File Format.doc

Thanks for your attention.
Any comment is sincerely appreciated.