

eMMC Programming with Data I/O Flashcore

eMMC 4.3+ feature set introduces boot area and partitioning features. To use these features a single image file must be provided to Data I/O TaskLink software. Currently there are two file formats supported:

- 1) "SuperHeader": a single image file consisting of:
 - 1 MByte header "Superheader" containing:
 - Data for "Extended CSD register"
 - Definition of partitions to be written
 - Optional: Re-partition parameters
 - Optional: CMD56 Refresh parameters
 - User data to be written

"SuperHeader" is the recommended format for programming eMMC.

For the specification of "SuperHeader" refer to <http://ftp.dataio.com/FCNotes/Footnote/Super%20Partition%20Organization.pdf>

A Windows-tool to demonstrate creation of a SuperHeader is available under below link: <ftp://ftp.dataio.com/FCNotes/Footnote/SuperHeaderEditor.zip>

- 2) "Multi Partition Header" ": a single image file consisting of:
 - 1 MByte header "Multi Partition header" containing:
 - Data for "Extended CSD register"
 - User data to be written

For the specification of "Multi Partition Header" refer to <http://ftp.dataio.com/FCNotes/Footnote/eMMC%20on%20Data%20IO.pdf>

"Multi Partition Header" is supported for backwards compatibility only and not recommended for new designs.

Mandatory User Options in TaskLink -> Special Features

SDMP: multi-partition header Default: *Auto Check*

Auto Check	Auto checks if image includes header. Recommended option.
Has Multi-partition header	The programming image must contain either a multi-partition-header or a super-header.
NO Multi-partition header	The programming image only targets the user-area partition. No data will be written to extended-CSD-register or boot-partitions. (Standard programming operation when no configuration is required.) The programming image must be a flat binary file without header.

SDMP: super-partition function Default: *Auto Check*

Auto Check	Auto checks if image includes a super-header. Recommended option. Note: when reading a Master device, please select "Enable".
Disable	Disables the super-header regardless of header configuration.
Enable	The programming image must contain a super-header.

SDMP: multi-partition function Default: *Auto Check*

Auto Check	Auto check if image includes multi-partition-header. Not recommended for new designs.
Disable	Disables multi-partition function regardless of a header configuration files.
Enable	The programming image must contain a multi-partition header.
Remapping	Only used for remapping the file format.

Optional User Options in TaskLink -> Special Features

Required Sector Num (SD part) Default: <Number of Sectors of the device>

Use this option to disable some sectors during programming/reading. The value should be less than or equal to the maximum sector table number. **Decimal value.**

Smart Whole Chip Read –above 2G device only– Default: <Enable>
Boolean value

This is a smart read method for e-MMC devices. If the function is enabled, the programmer will skip blank areas (all bytes are blank state) that are larger than 32 MBs, and read back the image without the skipped blank areas. However it may include blank blocks smaller than 32MBs.

NOTE: The function only supports devices above 2 GB.

NOTE: For more details, please refer to
http://ftp.dataio.com/FCNotes/Footnote/Smart_Whole_Chip_Read_User_Manual.pdf

Check device PRV Default: <Disable> Boolean value

This is a switch to enable PRV (product revision) check function.
Please define the expected PRV value in “**Product Version (PRV)**” setting.

NOTE: if the “Check device PRV” exists, but there is no Product Version (PRV) value, then an exception gets thrown when the algorithm runs.

Product Version (PRV) Default: <0> Hexadecimal value

This setting to define the expected PRV value.

NOTE: The feature is only available when “Check device PRV” is set to <Enable>.

Partitions – Following settings are for “Multi Partition header” only

When using “Super header”, the following settings are not used.

Boot Partition #0 Data Size(block) Default: <0> Decimal value

Optional, used with “Multi Partition header” only.
This is a boot partition capacity.

FORMULA: If Boot Partition capacity = n bytes then set it to = (n/512) Blocks

EXAMPLE 1:

If you want a target boot partition size of 128 KB then $(128 \times 1024)/512 = 256$ blocks.

EXAMPLE 2:

If you want a target boot partition size of 2 MB then $(2 \times 1024 \times 1024)/512 = 4096$ blocks.

Boot Partition #1 Data Size(block) Default: <0> Decimal value

Optional, used with “Multi Partition header” only.
This is a boot partition capacity.

FORMULA: If Boot Partition capacity = n bytes then set it to = (n/512) Blocks
For examples see *Boot Partition #0 Data Size(block)* above.

General Purpose Partition #0 Data Size(block) Default <0> Hexadecimal value

General Purpose Partition #1 Data Size(block) Default <0> Hexadecimal value

General Purpose Partition #2 Data Size(block) **Default <0> Hexadecimal value**

General Purpose Partition #3 Data Size(block) **Default <0> Hexadecimal value**

These are general purpose partitions size for programming/reading.

User Partition #0 Data Start(block) Default: <0> Hexadecimal value

Optional, used with “Multi Partition header” only.

This is a user partition start address for reading.

FORMULA: The address at which to start reading is = $(0xABCDEF0x200)$.

Example:

To start reading at the 25th sector, the address of the 25th sector = $0xC80000$ bytes.

Data start = $0xC80000 / 0x200$ bytes = $0x6400$

User Partition #0 Data Size(block)

Default: <0> Hexadecimal value

Optional, used with “Multi Partition header” only.

This is a user partition size for reading.

FORMULA: If the device has multiple partitions, then this parameter is calculated as $[\text{Address of last sector Programmed} - (1\text{MB Header} + n \text{ bytes Boot Partition Data Size})] / 512$ bytes. (Hint– To work in a Hex value; 512 bytes = 200 hex).

Example 1:

If the last sector programmed = 20th sector, and you want a Boot Partition = 2 MB, then User Partition #0 Data Size(block) =

$[0xA80000 - (0x100000 + 0x200000)] / 0x200 = 0x3C00$

Example 2:

If the last sector programmed = 20th sector, and you want a Boot Partition = 0 MB, then User Partition #0 Data Size(block) =

$[0xA80000 - (0x100000 + 0)] / 0x200 = 0x4C00$

Boot Partition Size Change Enable –Samsung moviNAND only– Default: *Disable*

Optional, used with “Multi Partition header” only.

Enable/Disable the ability to change boot partition capacity of capable devices.

Changed Boot Partition Size –Samsung moviNAND only–

Default: <0> Decimal value

Optional, used with “Multi Partition header” only.

This is the parameter for boot area size change. (Refer to spec for capable devices.)

FORMULA: If Boot Partition capacity = $128k * n$ bytes then set it to = n

For examples $n = 1$ indicate Boot Partition capacity = 128k bytes.

RPMB Partition Size Change Enable: –Samsung moviNAND only–

Default: *Disable*

Optional, used with “Multi Partition header” only.

Enable/Disable the ability to change RPMB partition capacity.

Changed RPMB Partition Size –Samsung moviNAND only–

Default: <0> Decimal value

Optional, used with “Multi Partition header” only.

This is the parameter for RPMB area size change. (Refer to spec for capable devices.)

FORMULA: If RPMB Partition capacity = $128k * n$ bytes then set it to = n

For examples see *Changed Boot Partition Size* above.

Program Boot Area in Single Block Mode

Default: No

An alternative Method to program the Boot (may be required in some cases).

Special Notes

NOTE: Partition information one block indicates 512 Bytes according to e-MMC4.4 spec.

USER CASE 1: Multiple Partition File Format (Necessary options only)

SDMP: multi-partition header
SDMP: multi-partition function
Boot Partition #0 Data Size(block)
Boot Partition #1 Data Size(block)
General Purpose Partition #0 Data Size(block)
General Purpose Partition #1 Data Size(block)
General Purpose Partition #2 Data Size(block)
General Purpose Partition #3 Data Size(block)

USER CASE 2: Super Partition File Format (Necessary options only)

SDMP: multi-partition header
SDMP: multi-partition function
SDMP: super-partition function

HINT: Check whether the value required are Hexadecimal or Decimal units.

Revision History

V0.1 June 27, 2017
 Created doc

Appendix

Current documentation is available under <http://ftp.dataio.com/FCNotes/Footnote/>