

**mDOC H3 Flash Memories
and
Programming mDOC H3 Flash Memories Using
Dataman Device Programmers**

Application Note

January, 2009
an_dataman_mdock_h3_flash, version 1.00

Table of contents

Table of contents.....	2
List of tables.....	2
List of figures.....	2
Introduction.....	3
Image access mode.....	3
mDOC H3 Image Format	3
Creating optimized image file by reading of master device.....	5
Creating optimized image file by RAM MTD utilization.....	5
Programming mDOC H3 using image file.....	5
Docshell utility access mode.....	6
DFORMAT Utility.....	6
DINFO UTILITY.....	8
DIMAGE UTILITY.....	8
References.....	10

List of tables

Table 1: mDOC H3 Image Format.....	4
Table 2: mDOC H3 Image Format	4
Table 3: Non-optimally version of image file cumulates only 32 sectors.....	4
Table 4: Optimally version of image file cumulates 256 sectors.....	5
Table 5: DFORMAT Flags.....	7
Table 6: RAM MTD Supported Devices.....	7
Table 7: DINFO Flags.....	8
Table 8: DIMAGE Flags.....	9
Table 9: DIMAGE Error Messages.....	9
Table 10: DIMAGE Socket Status Messages.....	9

List of figures

Figure 1: TrueFFS - Legacy mDOC vs. mDOC H3 Architecture.....	3
---	---

Introduction

This manual describes the work with mDOC H3 devices on Dataman programmers. SanDisk's mDOC H3 products are high-performance, single-chip flash disks that provide full hard-drive emulation for all major operating systems (Oss) and platforms. The mDOC H3 uses advanced Multi-Level Cell(MLC) and binary (SLC) NAND flash technologies, enhanced by SanDisk's proprietary TrueFFS embedded flash management software running as firmware on the flash controller. The mDOC H3 provides superior performance through a true 16-bit bus interface, X2 technology and offers security-enabling and data-protection features. SanDisk's proprietary TrueFFS flash management software is now embedded within the mDOC H3 device and runs as firmware from the flash controller. Embedded TrueFFS overcomes SLC and MLC NAND-related error patterns by using a robust error detection and correction (EDC/ECC) mechanism.

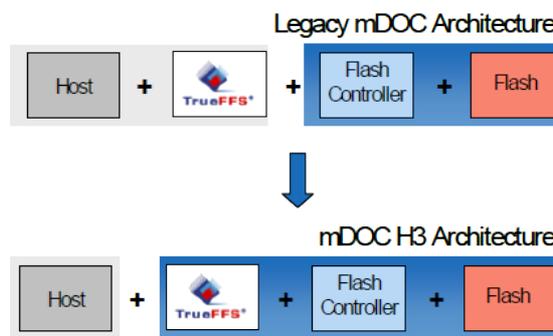


Figure 1: TrueFFS - Legacy mDOC vs. mDOC H3 Architecture

Dataman software provides two possibilities of programming of mDOC H3 devices: "Image access mode" or "Docshell utility access mode". The low-level programming hardware access layer is implemented over the NOR physical interfaces.

Image access mode

The "Image access mode" is default mode when you select device name from PG4UW software. Programming algorithm requires the mDOC H3 file image format for programming of mDOC H3 device. It is suitable for mass production. The image file of legacy DiskOnChip G3, G4 H1 and DiskOnChip-based MCP devices does not apply. Image files can be created with the SanDisk Docshell utility or PG4UW (recommended for better programming speed).

mDOC H3 Image Format

The mDOC H3 image is a binary file that contains all the data required for duplicating of mDOC H3 device. It is comprised of a common header and a series of command blocks as shown in Table 1. Each block represents a single ATA command including all of its 7 registers and an optional data payload that is to be transferred to the device.

Common Header (256 bytes)
Command Block #0
...
Command Block #n-1

Table 1: mDOC H3 Image Format

The common header consists of 256 bytes. Table 2 describes the Common Header fields.

Field Size (Bytes)	Field Name	Description
4	Image Compatibility Number	This number changes when image format is changed by SanDisk.
4	READ Version	Version of the READ application used to generate this image.
8	Signature	ASCII string specified by SanDisk to verify image compatibility. "EYOPYPRE"
4	Primary Device Capacity	In sectors, MUST not be 0
4	Secondary Device Capacity	In sectors, 0 for no secondary device
1	Number of Partitions in Primary Device	
1	Number of Partitions in Secondary Device	
2	Buffer Size Used for Reading Image	In sectors, default is 32, Dataman required 256 for better speed
...		

Table 2: mDOC H3 Image Format

There is important value of “Buffer Size Used for Reading Image” that represents amount of cumulated sectors. It is important to cumulate as many as possible sequential sectors in the same data block of image file for better programming speed. Please see differences in tables below, that contains contents between twice optimized and standard image file.

Common Header of image	
00000000:	01 00 00 00 02 00 00 00 45 59 4F 50 59 50 52 45 # # EYOPYPRE
00000010:	00 00 08 00 00 00 00 00 07 00 20 00 44 4F 43 2D # # #DOC-
Command Block Header	
02	20 10 00 00 40 83 03 20 00 00 FF 00 00 00 00 # # @□# .
02	20 10 00 00 40 82 04 20 00 00 FF 00 00 00 00 # # @,# .

Table 3: Non-optimally version of image file cumulates only 32 sectors

Common Header of image																					
00000000:	01	00	00	00	02	00	00	00	00	45	59	4F	50	59	50	52	45		#	#	EYOPYPRE
00000010:	00	00	08	00	00	00	00	00	00	07	00	00	01	44	4F	43	2D		#	#	#DOC-
Command Block Header																					
	02	00	10	00	00	40	83	03	00	01	00	FF	00	00	00	00		#	#	@□# # `	
	02	00	10	00	00	40	82	04	00	01	00	FF	00	00	00	00		#	#	@,# # `	

Table 4: Optimally version of image file cumulates 256 sectors

Program Pg4uw detects validity of image and usage of non-optimized image display by attention in log window: “Attention: Buffer Size Used for Reading Image is 32 sector(s)! Better performance requires 256 sectors to be united in the same data block.”

If you use PG4UW software to reach maximum speed, please select device with suffix (DSH) (e.g. for SDED7-256M-N9 [FBGA115] device select SDED7-256M-N9 [FBGA115](DSH)) and create optimized image file by reading of master device or RAM MTD simulation of appropriate device.

Creating optimized image file by reading of master device

After selection of appropriate device with suffix (DSH) that represents “Docshell utility access mode”, load command batch file with DIMAGE utility, that reads the master mDOC H3 image to a file.

```
DIMAGE /WINSRC:00000 /FILETRG:master_image_file_name.BIN
```

Note: When duplicating a mDOC H3 with active hardware protection, you must use DIMAGE with the /SRCBDTL# flag in order to disable the protection. The password will be included in the virtual image for later use by DIMAGE.

Example

```
DIMAGE /WINSRC:00000 /FILETRG:MYDOC.IMG /SRCBTL0:12345678
```

Copies the master mDOC H3 contents with a protected partition to the MYDOC.IMG file.

There is possibility to verifies that the image file contains the contents of the device.

```
DIMAGE /VERIFYIMG:master_image_file_name.BIN /WINTRG:00000
```

Example

```
DIMAGE /VERIFYIMG:MYDOC.IMG /WINTRG:00000
```

If you have virgin device you can load batch file, that is formatting (initializes the flash media and creates the partitions) and is copying all target application files to mDOC H3 device by using the DFORMAT utility at first time and next reading master device in last step as written above.

Creating optimized image file by RAM MTD utilization

Use this option when you need create an optimized file without real device insertion into socket. RAM MTD simulates mDOC H3 device. The mDOC H3 data are stored in a simulated file that is located in the same directory as loaded batch file. After selection of appropriate device with suffix (DSH) that represents “Docshell utility access mode”, load command batch file with DFORMAT and DIMAGE utilities. For those utilities is necessary to replace memory address at which mDOC H3 is located from /WIN:00000, /WINSRC:00000 to RAMDEVTYPE, RAMDEVFILE flags. The RAM MTD utilization requires an inserted converter without device in programmer and insertion test deactivated as well. For more details, please read section “Docshell utility access mode”.

Programming mDOC H3 using image file

First of all, select your mDOC H3 device by **[Select]** button or press key <Alt+F5>. Load binary mDOC H3 image file. Use **Device Operation Options** (shortcut <Alt+O>) to define operation features.

Insertion Test

Test of proper device insertion.

Verify after programming

Enable or disable verification after programming. Default value is False. The Program operation verifies the programming of operation on the fly. Verify programming is a process of data block reading immediately after programming it and comparing the result with the original data block. Please note that the mDOC H3 image contains verify command blocks intertwined with the program command blocks.

Possible device operations:

Verify operation:

This command compares content of the whole device including its available special areas with data in buffer.

Program operation:

Copying loaded image file in buffer to the target mDOC H3 devices results in identical mDOC H3 target devices.

Notes: Device ID check is executed internally. Project file is supported. Project file contains device configuration buffer data saved and user interface configuration.

Docshell utility access mode

When you select device name with suffix (DSH) from PG4UW software, the “Docshell utility access mode” is used. The docshell utility provides: DFORMAT, DINFO, DIMAGE utilities.

DFORMAT Utility

DFORMAT utility may be used for the following purposes:

- Format the entire mDOC H3 device and program it.
- Format (update) single partition. In this case DFORMAT should be used with the /NOFORMAT flag. The size of the partition MUST remain the same.
- Read the IPL data or partition data located on one or more partitions. In this case /READ flag MUST be used.
- Return mDOC H3 to virgin state using the /UNFORMAT flag.

Syntax: DFORMAT <<[/WIN:00000]> [/Flag[n]:parameter[suffix]]

DFORMAT Option	Description
/WIN:00000	Memory address at which mDOC H3 is located, for Dataman programmers is equal 00000.
/NOFORMAT	Use this flag to update the mDOC H3 with the IPL or partition(s) without reformatting the entire device. When /READ flag used /NOFORMAT is not required.
/READ	Causes DFORMAT to perform a read operation. Conjunction with NOFORMAT is not required.
/BDTLL[n]:Partition Size	Sets the size of the partition [n]. The size of the last partition does not have to be defined.
/BDTLF[n]:<[Partition Image file][!]>	In update or format modes - Places a partition image file on the partition [n]. When '!' is specified – Deletes all the data from corresponding partition. Other partitions will stay in tact.
/EP<n>: <size>[s m k p]	Format Enhanced performance partition.
/OTWBDTL[n]	Places OTW protection on partition 'n'. OTW protection may be placed only on the partitions, which have changeable protection flag ON.
/LEAVE:[n]	Keep the n first partitions while formatting; (n = 0-13).
/BDTLP[n]:WCL: Password [Default value = no protection]	Sets the protection type (Write/Change/Lock or Read-Write) and the protection key (password) of the partition [n] (n = 0-13).
/BDTLZ[n]: Password	Inserts the key of the partition number [n] (n=0-13, default is 0)
/IPLP:WC: Password	Sets the protection type (Write/Change) and the protection key (password) of the IPL partition.
/IPLZ: Password	Inserts the key of the IPL partition.
/MBR<n>	Creates/formats logical partition on partition number 'n'. When <n> is not specified, the default is bdtl0.
[:fs<file system>]	Place in MBR file system indicator <file system>.
/LOGF<n>:<file_name>	Can be used only in conjunction with the /READ flag, /LOGF specifies the name of the file and holds the data that will be read from the logical partition of partition number 'n'.

mDOC H3 Flash Memories Application Note

/IPL:File !	File is written as a custom IPL mDOC H3.
/OTP:<file name>	Write/Read OTP to/from the file.
/UNFORMAT	Removes any existing mDOC H3 formatting and restores mDOC H3 to its virgin state.
/IPLMODE:<p v>	Write IPL in the customized mode mDOC H3 P – for paged RAM boot mode V – for virtual RAM boot mode
/XIPMODE:<xip_mode>	<XIP mode> is a bitmap, 2 bytes long in hexadecimal notation (bit 16 is a most significant) where: bit 5 ON means 128K window mode bit 7 ON means SWAP bytes mode bit 6 ON means No Address shift mode
RAMDEVTYPE:<type>	Specifies the NEW RAM device type simulation. This flag should be specified only when a new virtual mDOC H3 is to be created. See Table 6.
RAMDEVFILE:<filename>	Specifies existing RAM MTD file name, when it should be used; or defines the RAM MTD file name, when file should be created by type. See Table 6.

Table 5: DFORMAT Flags

mDOC H3 Device	RAMDEVTYPE	Default File Name
mDOC H3 128MB	30	"H3_1Gb.img"
mDOC H3 1GB	32	"H3_8Gb.img"
mDOC H3 2GB	34	"H3_16Gb.img"
mDOC H3 256MB	36	"H3_2Gb.img"

Table 6: RAM MTD Supported Devices

DINFO UTILITY

DINFO displays mDOC H3 information, such as physical size and partition number.

Syntax: DINFO [/WIN:00000] [/FLAG:parameter]

DINFO Option	Description
/WIN:00000	Memory address at which mDOC H3 is located, for Dataman programmers is equal 00000.
/OTP	Displays OTP and device ID information.
/BDTL	Displays the partition information for the selected mDOC H3 device, including the number of partitions on the media, capacity data, and protection status for each individual partition.
/LOGD[bdtl][:drive]	Total number of logical partitions per mDOC H3 device, per partition.
/IPL	Provides information about the mDOC H3 IPL.
RAMDEVFILE:<filename>	Specifies existing RAM MTD file name, when it should be used; or defines the RAM MTD file name, when file should be created by type. See Table 6.

Table 7: DINFO Flags

DIMAGE UTILITY

The DIMAGE utility is used to perform the following:

- Read the master mDOC H3 image to a file.
- Duplicate mDOC H3, writing an image to target mDOC H3 devices.
- Verify that the contents of the target devices match the master image file.

Syntax: DIMAGE /WINSRC:00000 /FILETRG:file PROTECTION-FLAGS
 DIMAGE /FILESRC:file /WINTRG:00000 PROTECTION-FLAGS

Where PROTECTION-FLAGS can be a subset of the following:

[/SRCBDTL#:password]
 [/SRCIPL: password]

DIMAGE Option	Description
/FILESRC:file	Name of the source image.
/FILETRG:file	Name of the target image.
/WINSRC:00000	Memory address at which mDOC H3 is located, for Dataman programmers is equal 00000.
/WINTRG:00000	Memory address at which mDOC H3 is located, for Dataman programmers is equal 00000.
/SRCBDTL#:password	/SRCBDTL#:password Protection key of the protected partition on the source device (the default is 0). Inserting this flag disables the write protection of the specific partition.
/SRCIPL: password	Protection key of the protected IPL partition on the source device. Inserting this flag disables the write protection of the specific partition.
RAMDEVFILE:<filename>	Specifies existing RAM MTD file name, when it should be used; or defines the RAM MTD file name, when file should be created by type. See Table 6.
/VERIFYIMG	Performs verify of the image file with the device contents. This option verifies that the devices contain the contents of the image file, this option verifies also the reverse (e.g. the image file contains the contents of the device).

Table 8: DIMAGE Flags

Table 9 describes possible DIMAGE error messages.

DIMAGE Option	Description
Not enough memory	There is not enough system memory to complete the operation.
Can't open file	The system cannot open the image file. This may occur during write or read cycles.
Adapter not found	The system cannot find a mDOC H3 adapter.
Can't write to file	The system cannot write to the image file.
Source file format error	DIMAGE detected that the image file being written to the target device is not a valid image file.

mDOC H3 Flash Memories Application Note

Wrong parameters passed	There was an error in the parameters given in the command line.
Socket does not exist	The socket chosen by user does not exist.
Failed initialize access layer	Initialization of the access layer (such as NOR, etc.) failed.
Socket is not active	Socket failed.

Table 9: DIMAGE Error Messages

Table 10 describes the DIMAGE socket status messages.

DIMAGE Option	Description
Passed	Operation passed on this socket
Socket failed	Operation failed on this socket
ATA command failed	ATA command that was sent during PUTIMAGE failed
Socket not compatible	mDOC H3 not compatible with image file

Table 10: DIMAGE Socket Status Messages

References

- [1] SanDisk, Inc.: mDOC H3 Embedded Flash Drive (EFD) featuring Embedded TrueFFS® Flash Management Software, Rev. 1.3, May 2008
- [2] SanDisk, Inc.: DOC Driver 1.0 Block Device (BD), Software Developer Kit (SDK) Developer Guide, Rev. 1.0, January 2007
- [3] SanDisk, Inc.: DOC Driver 1.0 Software Utilities Developer Guide, Rev. 1.0, January 2007
- [4] M-Systems, Inc.: DOC H Device Programming Programming Specification, Rev. 0.1, March 2006