Using Logic's lpd_MSFLASHFMD Driver with the LH7A400-10

Application Note 279

Mike Aanenson

Logic Product Development

Published: July 2004

Abstract

This doc explains how to use the lpd_MSFLASHFMD driver with the LH7A400-10 Card Engine.

No warranties of any nature are extended by this document. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipments. The only warranties made by Logic Product Development, if any, with respect to the products described in this document are set forth in such license or agreement. Logic Product Development cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

Logic Product Development may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering the subject matter in this document. Except as expressly provided in any written agreement from Logic Product Development, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

REVISION HISTORY

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	Mike Aanenson	Release – written for beta_001 of lpd_MSFlashFMD	HAR	1/3/05

1 Introduction

Logic Product Development has created a packaged driver, the lpd_MSFLASHFMD_xxx_nnn driver, based on the MSFLASHFMD source code included with Windows Platform Builder.

This driver allows a user to use a portion of the card engine's onboard StrataFlash (NOR Flash) for a FAT File System repository in Windows CE. A folder will show up in the system that can be used much like a CompactFlash card or other storage devices that support a file system. This allows for persistent file storage of files in Flash. Another feature is that a hive registry hive files (persistent registry) can be placed here (please note that to use the lpd_MSFLASHFMD driver for a hive registry repository, additional registry modifications are required).

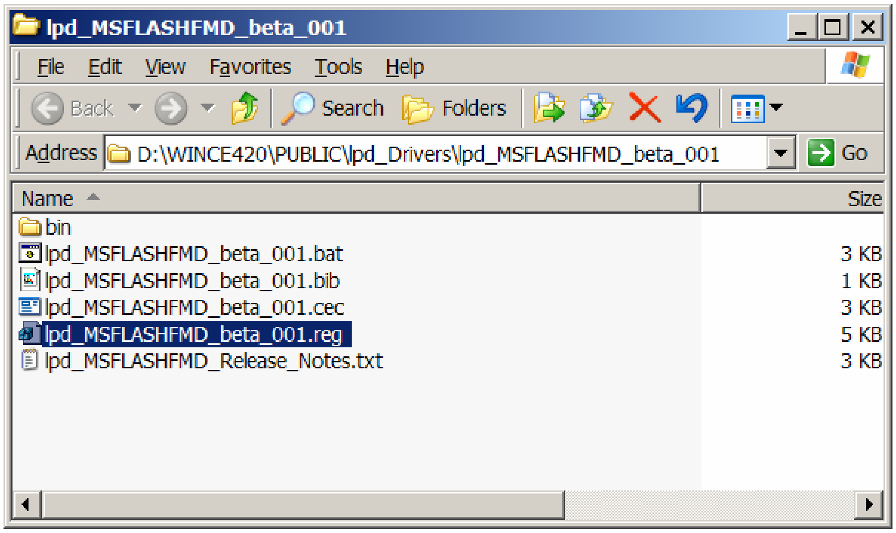
When using this driver:

- The Windows CE image must be running out of RAM instead of flash memory
    - The reason for this is due to the system being unable to execute out of flash while programming or erasing the same flash chip.
- To create an image that runs out of RAM, the CE image must be built as a RAM image or as a RELOCATE_FROM_FLASH image (burned into Flash, but is copied to and executed from the RAM)

Important Note: Since this driver allows writing to the onboard flash, there is the possibility that blocks 0 and 1 (where LogicLoader (Bootloader) is stored) could be overwritten – this would render the board unbootable. **Calculate the values for the Address and Size of the flash Storage folder carefully**. If the folder's end address goes past the end of the available flash, it will wrap and start at block 0.

2 Calculating Address and Size Parameters for the Registry

Find the lpd_MSFLASHFMD_xxx_nnn folder and open the .reg file as shown below in Figure 2.1. This file contains the values for the location and size of the Flash Storage folder that the driver uses when it boots. It is critical to find the correct values for the address and size so that the Flash Storage card does not overwrite other critical portions of the Flash.



Address: D:\WINCE420\PUBLIC\lpd_Drivers\lpd_MSFLASHFMD_beta_001

Name	Size
bin	
lpd_MSFLASHFMD_beta_001.bat	3 KB
lpd_MSFLASHFMD_beta_001.bib	1 KB
lpd_MSFLASHFMD_beta_001.cec	3 KB
lpd_MSFLASHFMD_beta_001.reg	5 KB
lpd_MSFLASHFMD_Release_Notes.txt	3 KB

Figure 2.1: Location of lpd_MSFLASHFMD_beta_001.reg

2.1 lpd_MSFLASHFMD_beta_001.reg

- The contents of this section have been taken from lpd_MSFLASHFMD_beta_001.
- The “MemBase” is set to the address of ‘B1A00000’ and the size, “MemLen”, is set to ‘00400000’.
- With the release of the beta_001 driver, the address here is a *virtual* rather than *physical* address (e.g. 0xB1A00000 corresponds to physical address 0x00A00000 – 10Mbytes up from the bottom of flash memory)

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\lpd_MSFLASHFMD]
   "Prefix"="DSK"
   "Dll"="lpd_MSFLASHFMD_beta_001.dll"
   "Order"=dword:2
   "Index"=dword:2
   "Ioctl"=dword:4
   "Profile"="lpd_MSFLASHFMD"
   "IClass"="{A4E7EDDA-E575-4252-9D6B-4195D48BB865}"
   "MemBase"=dword:B1A00000    ;A400 Base
   "MemLen"=dword:00400000
   "FSD"="FATFSD.DLL"
   "ScratchReg"=dword:C0000000
   "FlashReg"=dword:71000000

; Override names in default profile
[HKEY_LOCAL_MACHINE\System\StorageManager\Profiles\lpd_MSFLASHFMD]
    "Name"="MSFLASH for STRATAFLASH"
    "Folder"="NOR Flash"
    "AutoPart"=dword:1
    "AutoFormat"=dword:1

[HKEY_LOCAL_MACHINE\System\StorageManager\AutoLoad\lpd_MSFLASHFMD]
    "DriverPath"="Drivers\\BuiltIn\\StrataFMD"
    ; LoadFlags 0x01 == load synchronously
    "LoadFlags"=dword:1
    "Order"=dword:0

[HKEY_LOCAL_MACHINE\Windows CE Tools]
"Platform"="{23DD5F5A-E4FF-4348-A830-6AEDED32E830}"
```

2.2 Flash Memory

The Flash Memory is mapped out below.

Please note the differences between 1) RAM and RELOCATE_FROM_FLASH images, and 2) 16MB and 32MB card engine flash sizes.

Typical Windows CE Flash Memory Mapping for the LH7A400-10

-

	Physical Address		*Virtual Kernel Address*	*Area Size*
16MB Flash on Card Engine				
RAM Image				
	0x00000000	Beginning of Flash (Block 0)	0xB1000000	-
		Bootloader		256KB
	0x0003FFFF	End of Block 0	0xB103FFFF	-
	0x00040000	Beginning of Block 1	0xB1040000	-
		LogicLoader		256KB
	0x0007FFFF	End of Block 1	0xB107FFFF	-
	0x00080000	Beginning of Block 2	0xB1080000	-
		Open Flash Memory		15.5MB
		Available for Flash Storage		-
	0x00FFFFFF	End of Flash	0xB1FFFFFF	-
			Total	16MB
RELOCATE_FROM_FLASH Image				
	0x00000000	Beginning of Flash (Block 0)	0xB1000000	-
		Bootloader		256KB
	0x0003FFFF	End of Block 0	0xB103FFFF	-
	0x00040000	Beginning of Block 1	0xB1040000	-
		LogicLoader		256KB
	0x0007FFFF	End of Block 1	0xB107FFFF	-
	0x00080000	Beginning of Block 2	0xB1080000	-
		**Memory Reserved for CE Image*		15MB
				-
	0x00F7FFFF	End of Flash	0xB1F7FFFF	-
	0x00F80000	Beginning of possible Flash Storage Area	0xB1F80000	-
		Open Flash Memory		512KB
		Available for Flash Storage		-
	0x00FFFFFF	End of Flash	0xB1FFFFFF	-
			Total	16MB

*The size of the Windows CE image is set to a value in the lha400_10_config_bib.h file. The default value is 0x00F00000, which equals 15MB.

	Physical Address		Virtual Kernel Address	Area Size
32MB Flash on Card Engine				
RAM Image				
	0x00000000	Beginning of Flash (Block 0)	0xB1000000	-
		Bootloader		256KB
	0x0003FFFF	End of Block 0	0xB103FFFF	-
	0x00040000	Beginning of Block 1	0xB1040000	-
		LogicLoader		256KB
	0x0007FFFF	End of Block 1	0xB107FFFF	-
	0x00080000	Beginning of Block 2	0xB1080000	-
		Open Flash Memory		31.5MB
		Available for Flash Storage		-
	0x01FFFFFF	End of Flash	0xB2FFFFFF	-
			Total	32MB
RELOCATE_FROM_FLASH Image				
	0x00000000	Beginning of Flash (Block 0)	0xB1000000	-
		Bootloader		256KB
	0x0003FFFF	End of Block 0	0xB103FFFF	-
	0x00040000	Beginning of Block 1	0xB1040000	-
		LogicLoader		256KB
	0x0007FFFF	End of Block 1	0xB107FFFF	-
	0x00080000	Beginning of Block 2	0xB1080000	-
		**Memory Reserved for CE Image*		15MB
				-
	0x00F7FFFF	End of Flash	0xB1F7FFFF	-
	0x00F80000	Beginning of possible Flash Storage Area	0xB1F80000	-
		Open Flash Memory		16.5MB
		Available for Flash Storage		-
	0x01FFFFFF	End of Flash	0xB2FFFFFF	-
			Total	32MB

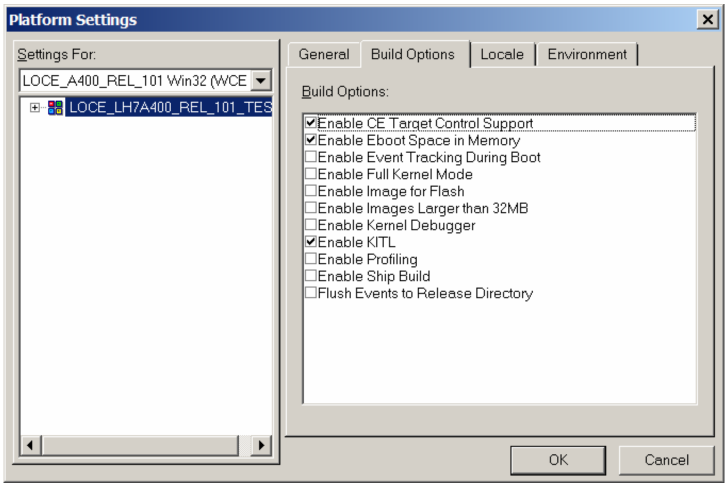
*The size of the Windows CE image is set to a value in the lha400_10_config_bib.h file.
The default value is 0x00F00000, which equals 15MB.

3 Adding the Driver to Your Project

1. Once the registry values have been correctly calculated and saved, the driver can be added to your project and built into the image. Please be sure that your image is being built as a RAM image or a RELOCATE_FROM_FLASH image. This can be checked in your Platform Settings in Platform Builder. Check that the "Enable Image for Flash" is not checked. The two environment variables you are looking for are presented in the figures on this page:

Environment Variables		
IMGFLASH	**RELOCATE_FROM_FLASH**	Image Type
		RAM Image
1		Flash Image
	1	Relocate From Flash Image

Figure 3.1: Environment Variables



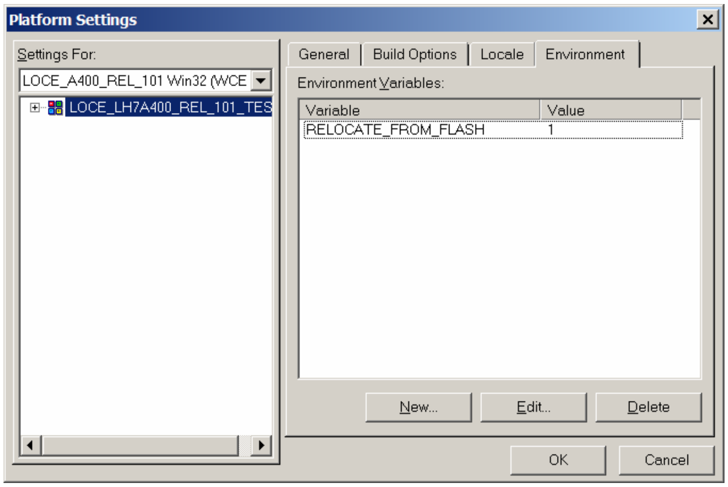
Settings For: LOCE_A400_REL_101 Win32 (WCE

LOCE_LH7A400_REL_101_TES

Build Options:

- ☑ Enable CE Target Control Support
- ☑ Enable Eboot Space in Memory
- ☐ Enable Event Tracking During Boot
- ☐ Enable Full Kernel Mode
- ☐ Enable Image for Flash
- ☐ Enable Images Larger than 32MB
- ☐ Enable Kernel Debugger
- ☑ Enable KITL
- ☐ Enable Profiling
- ☐ Enable Ship Build
- ☐ Flush Events to Release Directory

Figure 3.2: Platform Settings (1)



Settings For: LOCE_A400_REL_101 Win32 (WCE

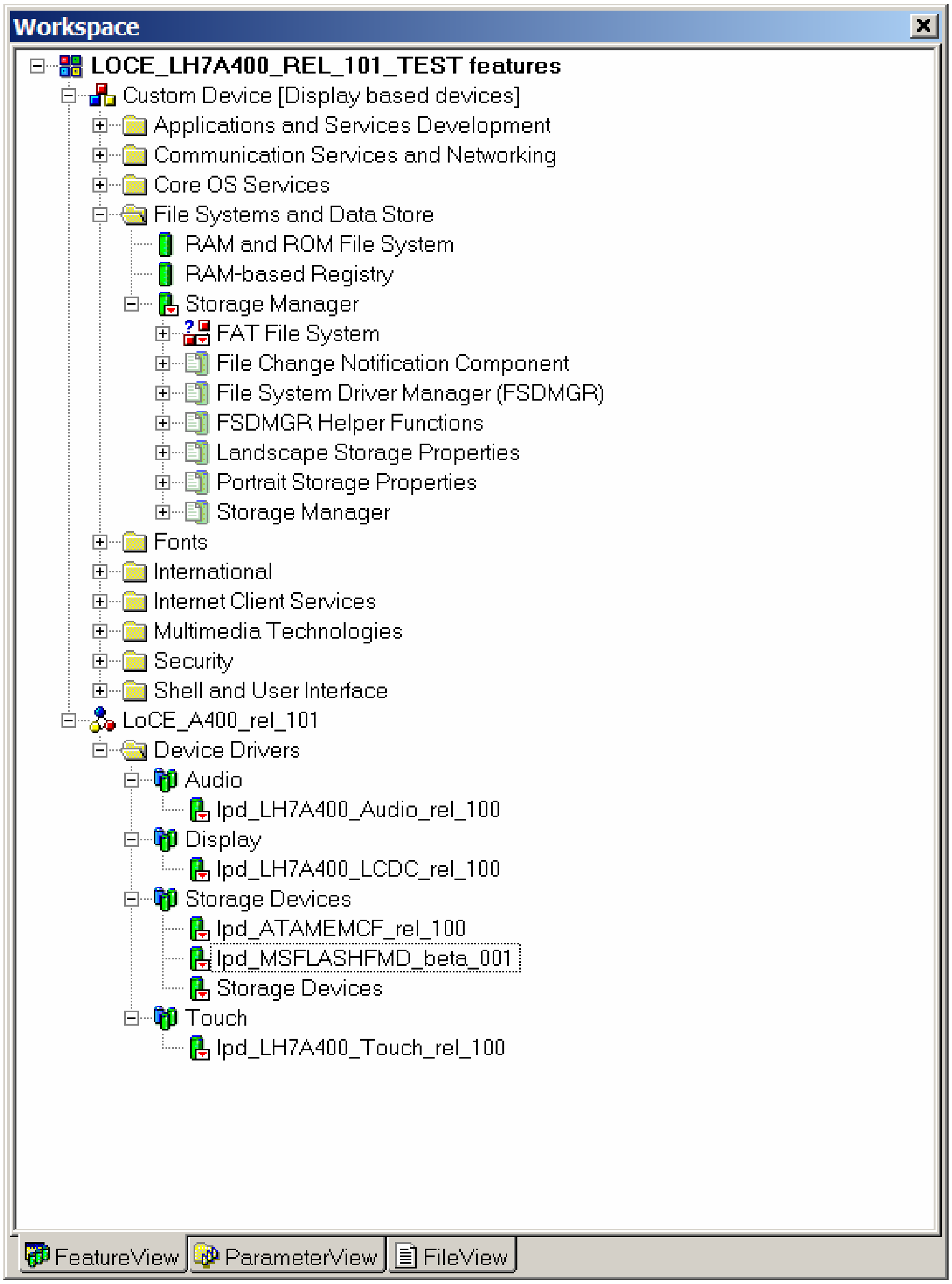
LOCE_LH7A400_REL_101_TES

Environment Variables:

Variable	Value
RELOCATE_FROM_FLASH	1

Figure 3.3: Platform Settings (2)

2. The driver added to your project will show up under the ‘Storage Devices’. Please be sure to add the ‘FAT File System’ component to your project in order for this driver to work properly.



- **LOCE_LH7A400_REL_101_TEST features**
  - Custom Device [Display based devices]
    - Applications and Services Development
    - Communication Services and Networking
    - Core OS Services
    - File Systems and Data Store
      - RAM and ROM File System
      - RAM-based Registry
      - Storage Manager
        - FAT File System
        - File Change Notification Component
        - File System Driver Manager (FSDMGR)
        - FSDMGR Helper Functions
        - Landscape Storage Properties
        - Portrait Storage Properties
        - Storage Manager
    - Fonts
    - International
    - Internet Client Services
    - Multimedia Technologies
    - Security
    - Shell and User Interface
  - LoCE_A400_rel_101
    - Device Drivers
      - Audio
        - lpd_LH7A400_Audio_rel_100
      - Display
        - lpd_LH7A400_LCDC_rel_100
      - Storage Devices
        - lpd_ATAMEMCF_rel_100
        - lpd_MSFLASHFMD_beta_001
        - Storage Devices
      - Touch
        - lpd_LH7A400_Touch_rel_100

Figure 3.4: Add the ‘FAT File System’ Component to Your Project

3. After the system has booted up, you will find a folder called “NOR Flash” located in “My Computer”.

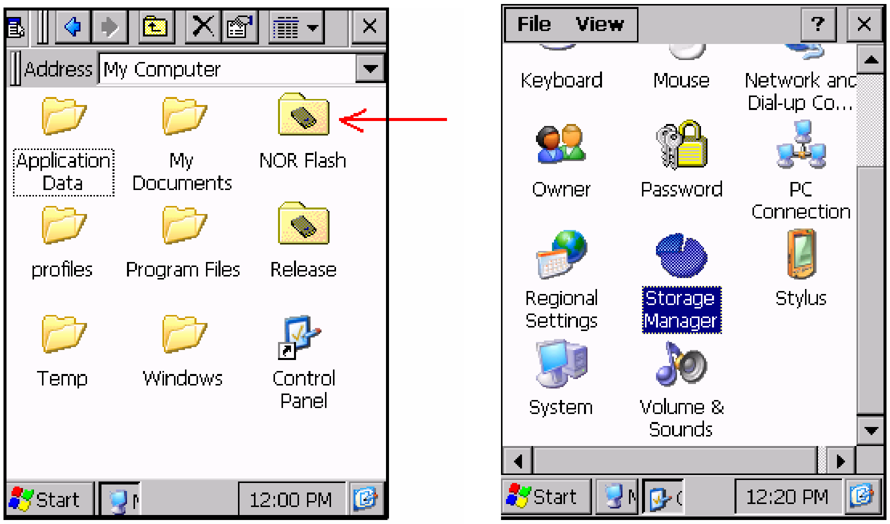


Figure 3.5: “NOR Flash” located in “My Computer”

4 Changing Parameters after Using the Driver

When the system has successfully booted and created the flash storage folder once, the partition must be deleted first in order to change the values for the starting address and size. Neither deleting the partition, nor erasing the flash memory is the #1 problem-cause that most user’s experience when developing with the MSFLASHFMD driver. The old disk must be erased if an aborted boot occurred or if the disk parameters need to be changed.

The partition can be deleted by using the ‘Storage Manager’ in the CE control panel. This feature is only available in a Graphical User Interface image. If the system does not have a Graphical User Interface, the file system tables must be overwritten in flash using other means, such as the LogicLoader.

Newer revisions of LogicLoader can also erase the partition by using the ‘erase’ command ***in conjunction with the physical addresses***. If the ‘erase’ command is not used properly in LogicLoader, the system may use previous values at the next boot up.

Please note that the GUI method must be done before updating the parameters in the registry. The Logic Loader ‘erase’ command can be performed at any time -- it will force the system to autoformat and autopartition during the next cold reset / boot cycle.

Be very cautious when using the “erase” command so that the system does not erase blocks “0” or “1” either by direct address assignment or by “wrapping” around the top of the flash memory. For example, if the system is told to erase 16MB of memory starting at block 2 on a 16MB flash system, the erase command will try to erase memory at 16MB + 2 blocks. The “+2 blocks” will wrap around the top address of the flash and will erase blocks 0 and 1 since the highest order address line is a NC on the flash memory (address line 24 in that case).