Freescale Semiconductor
Application Note

Document Number: AN4111
Rev. 0, 04/2010

Changing the i.MX25 NAND Flash Model for Windows Embedded CE 6.0

by *Multimedia Applications Division*
Freescale Semiconductor, Inc.
Austin, TX

This application note describes how to incorporate a new NAND Flash model in a custom design using the i.MX25 and Windows Embedded CE 6.0.

Most custom boards incorporate several components that are not 100% compatible with Freescale's Board Support Package (BSP). Frequently, the new i.MX hardware designs use different components, when compared to the ones used on Freescale development tool. In general, memory suppliers promote changes on their product portfolio very often. Therefore, memories like DDR, Nand Flash, and NOR Flash technologies are often replaced by models which are not supported by Freescale's BSP drivers. This application note is intended to be a guideline to include new Nand Flash models to the Freescale's BSP.

WinCE and Eboot share the same low level driver (Figure 1) on the Nand Flash implementation. When a new memory model is incorporated, most of the changes are done in the Flash Media Driver (FMD) layer. After building this layer, both Eboot and NK images support the new flash memory.

The Xloader (XLDR) available on the Freescale BSP does not use FMD and due to this, some changes are done to the XLDR code.

Contents

freescale semiconductor

Figure 1 shows the driver architecture of WinCE 6.0 and Eboot on NAND Flash implementation.

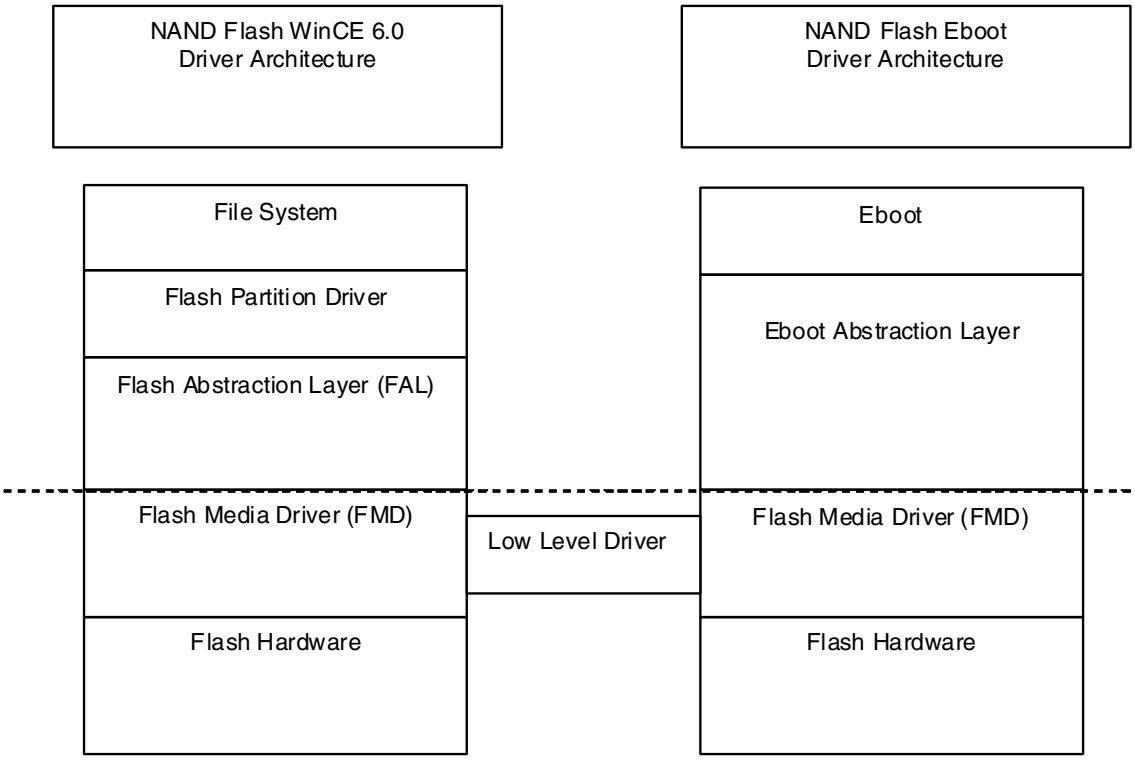

Figure 1. Driver Architecture of WinCE 6.0 and Eboot on NAND Flash Implementation

1 Implementation

The following section explains how to implement the NAND Flash.

1.1 Nand Flash Parameter Definition

The following are the parameter definitions, which must be configured when a new Nand Flash is added and these parameters are found on the Nand Flash specification document.

- NAND MARKER—Manufacture's ID code
- NAND DEVICE ID—Device ID code
- NAND BLOCK COUNT—Total number of blocks contained on the Flash
- NAND PAGE COUNT—Total number of pages contained on the Flash
- NAND PAGE SIZE—Size of each page
- NAND SPARE SIZE—Total size of spare area contained in a single page
- NAND BUS WIDTH—Size of data bus
- BBI MAIN ADDR—Address page where the BBI is located (See Section A.2, "BBI MAIN ADDR")
- BBI NUM—Number of pages in a block that contains the BBI (See Section A.3, "BBI NUM")

- BBIMarkPage[]—Defines the page(s) where the BBI is/are located in a block (See Section A.4, “BBIMarkPage”)

1.2 Inclusion of New NAND Flash Models

The following steps show how to include a new NAND Flash model by modifying the FMD files and the XLDR code.

NOTE

Ensure that the i.MX25 BSP is installed on the Visual Studio 2005.

1. Extract the following information from the NAND Flash specification. For more information, see Section 1.1, “NAND Flash Parameter Definition.”
   - NAND MARKER
   - NAND DEVICE ID
   - NAND BLOCK COUNT
   - NAND PAGE COUNT
   - NAND PAGE SIZE
   - NAND SPARE SIZE
   - NAND BUS WIDTH
   - BBI MAIN ADDR
   - BBI NUM
   - BBIMarkPage

2. Create a new header file with the following information below, name it with the NAND Flash part number. (For example - `K9LAG08U0M.h`). Store this file in the following location:

   `\WINCE600\PLATFORM\COMMON\SRC\SOC\COMMON_FSL_V2_PDK1_x\NAND\INC\`

   The empty values marked with red X’s and red arrows should be filled with the information extracted from the NAND Flash model specification.

```
#ifndef __K9LAG08U0M_H__          -> __Header File Name_H__
#define  __K9LAG08U0M_H__         -> __Header File Name_H__

// NAND Flash Chip CMD
#define CMD_READID                (0x90)      // Read ID
#define CMD_READ                  (0x00)      // Read data 1st cycle
#define CMD_READ2                 (0x30)      // Read data 2nd cycle
#define CMD_RESET                 (0xFF)      // Reset
#define CMD_ERASE                 (0x60)      // Erase setup
#define CMD_ERASE2                (0xD0)      // Erase
#define CMD_WRITE                 (0x80)      // Sequential data input
#define CMD_WRITE2                (0x10)      // Program
#define CMD_STATUS                (0x70)      // Read status

// NAND Flash Chip Size
#define NAND_BLOCK_CNT            (XXXX)      -> NAND BLOCK COUNT
#define NAND_PAGE_CNT             (XXX)       -> NAND PAGE COUNT
#define NAND_PAGE_SIZE            (XXXX)      -> NAND PAGE SIZE
```

```
#define NAND_SPARE_SIZE         (XX)        -> NAND SPARE SIZE
#define NAND_BUS_WIDTH             (X)          -> BUS WIDTH

// NAND Flash Chip
#define NAND_NUM_OF_CS          (1)

// NAND Flash Chip ID
#define NAND_MAKER_CODE         (0xXX)      -> NAND MARKER
#define NAND_DEVICE_CODE        (0xXX)      -> NAND DEVICE ID
#define NAND_ID_CODE            ((NAND_DEVICE_CODE << 8) | NAND_MAKER_CODE)

// NAND Flash Chip Operation Status
#define NAND_STATUS_ERROR_BIT   (0)      // Status Bit0 indicates error
#define NAND_STATUS_BUSY_BIT    (6)      // Status Bit6 indicates busy

// SWAP BBI
#define BBI_MAIN_ADDR           (XXX)       -> BBI MAIN ADDR
#define BBI_NUM                 (X)         -> BBI NUM
BYTE    BBIMarkPage[1] = {XXX};             -> BBIMarkPage

#endif
```

3. Create a new include file with the following information below, name it with the NAND flash part number. (For example - `K9LAG08U0M.inc`). Store this file in the following location:

`\WINCE600\PLATFORM\COMMON\SRC\SOC\COMMON_FSL_V2_PDK1_x\NAND\INC\`

The empty values marked with red X's and red arrows should be filled with the information extracted from the NAND Flash model specification.

```
;-------------------------------------------------------------------------------
;
;   File:  K9LAG08U0M.inc
;
;   Contains definitions for K9LAG08U0M NAND
;   flash memory device.
;
;-------------------------------------------------------------------------------

CMD_READID              EQU     0x90        ; Read ID
CMD_READ                EQU     0x00        ; Read data field
CMD_READ2CYCLE          EQU     0x30        ; Read CMD second cycle
CMD_READ2               EQU     0x50        ; Read spare field
CMD_RESET               EQU     0xFF        ; Reset
CMD_ERASE               EQU     0x60        ; Erase setup
CMD_ERASE2              EQU     0xD0        ; Erase
CMD_WRITE               EQU     0x80        ; Sequential data input
CMD_WRITE2              EQU     0x10        ; Program
CMD_STATUS              EQU     0x70        ; Read status

NAND_PAGE_CNT_LSH       EQU     (X)  -> NAND PAGE COUNT, in this format 2^x
NAND_PAGE_SIZE_LSH      EQU     (X)  -> NAND PAGE SIZE, in this format 2^x
NAND_BLOCK_SIZE_LSH     EQU     (NAND_PAGE_CNT_LSH+NAND_PAGE_SIZE_LSH)
NAND_PAGE_CNT           EQU     (1 << NAND_PAGE_CNT_LSH)
NAND_PAGE_SIZE          EQU     (1 << NAND_PAGE_SIZE_LSH)
NAND_BLOCK_SIZE         EQU     (1 << NAND_BLOCK_SIZE_LSH)
NAND_BLOCK_CNT          EQU     (XXXX)      -> NAND BLOCK COUNT
NAND_SPARE_SIZE         EQU     (XX)        -> NAND SPARE SIZE
BBI_PAGE_NUM            EQU     (X)         -> BBI NUM
BBI_PAGE_ADDR_1         EQU     (NAND_PAGE_CNT - 1)     ;NAND flash bbi page address
```

```
BBI_PAGE_ADDR_2         EQU     (NAND_PAGE_CNT - 1)     ;NAND flash bbi page address
NUM_OF_NAND_DEVICES     EQU     1           ;  Number of NAND device
NUM_OF_NAND_DEVICES_LSH EQU     0           ;
NAND_BUS_WIDTH            EQU      X          -> BUS WIDTH
```

4. Modify the header file - `nandbsp.h` located at:
`\WINCE600\PLATFORM\iMX25-3DS-PDK1_x\SRC\COMMON\NANDFMD\nandbsp.h` and add the `created.h` file from the step above. The following is the modified example and the changes are marked in bold.

```
//
// Copyright (c) Microsoft Corporation.  All rights reserved.
//
//
// Use of this source code is subject to the terms of the Microsoft end-user
// license agreement (EULA) under which you licensed this SOFTWARE PRODUCT.
// If you did not accept the terms of the EULA, you are not authorized to use
// this source code. For a copy of the EULA, please see the LICENSE.RTF on your
// install media.
//
//------------------------------------------------------------------------------
//
//  Copyright (C) 2007-2008, Freescale Semiconductor, Inc. All Rights Reserved.
//  THIS SOURCE CODE, AND ITS USE AND DISTRIBUTION, IS SUBJECT TO THE TERMS
//  AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT
//
//------------------------------------------------------------------------------
//
//  File:  nandbsp.h
//
//  Contains definitions for FMD implementation of the SoC NAND flash controller
//  and NAND memory device.
//
//------------------------------------------------------------------------------
#ifndef __NANDBSP_H__
#define __NANDBSP_H__
#ifdef BSP_NAND_K9LBG08U0M
#include "K9LBG08U0M.h"
#else ifdef BSP_NAND_K9LAG08U0M
#include "K9LAG08U0M.h"
#endif

#endif    // __NANDBSP_H__
```

5. Modify the include file - `nandchip.inc` located at:
`\WINCE600\PLATFORM\iMX25-3DS-PDK1_x\src\BOOTLOADER\XLDR\NAND\nandchip.inc` and add the `created.h` file from the step above. The following is the modified example and the changes are marked in bold.

```
;-------------------------------------------------------------------------------
;
;   Copyright (C) 2008-2009, Freescale Semiconductor, Inc. All Rights Reserved.
;   THIS SOURCE CODE, AND ITS USE AND DISTRIBUTION, IS SUBJECT TO THE TERMS
;   AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT
;
```

```
;-------------------------------------------------------------------------------
;
;   File:  nandchip.inc
;
;   Contains definitions of NAND flash for XLDR access.
;
;-------------------------------------------------------------------------------

; Include definitions for selected NAND flash device
IF :DEF: BSP_NAND_K9LBG08U0M
    INCLUDE K9LBG08U0M.inc
ELSE
    IF :DEF: BSP_NAND_K9LBG08U0D
        INCLUDE K9LBG08U0D.inc
    ELSE
         INCLUDE K9LAG08U0M.inc
    ENDIF
ENDIF

IMAGE_XLDR_NAND_BLOCK_OFFSET        EQU     (0)
IMAGE_XLDR_NAND_BLOCK_SIZE          EQU     (IMAGE_BOOT_BOOTIMAGE_NAND_OFFSET >>
NAND_BLOCK_SIZE_LSH)
IMAGE_EBOOT_NAND_BLOCK_OFFSET       EQU
(IMAGE_XLDR_NAND_BLOCK_OFFSET+IMAGE_XLDR_NAND_BLOCK_SIZE)

NUM_SEGMENT_NAND_USED               EQU     (NAND_PAGE_SIZE/NANDFC_MAIN_BUFF_SIZE)
NFC_BBI_MAIN_SEGMENT                EQU     (NANDFC_MAIN_BUFF0_OFFSET +
NAND_PAGE_SIZE-NANDFC_MAIN_BUFF_SIZE)
NFC_SPARE_SEGMENT_HIGH              EQU
(NANDFC_SPARE_BUFF0_OFFSET+(NUM_SEGMENT_NAND_USED-1)*NANDFC_SPARE_BUFF_SIZE)
NFC_SPARE_SEGMENT_LOW            EQU    (NFC_SPARE_SEGMENT_HIGH-NANDFC_SPARE_BUFF_SIZE)
NAND_BBI_COL_ADDR                   EQU
(((NANDFC_MAIN_BUFF_SIZE-(NUM_SEGMENT_NAND_USED-1)*(NAND_SPARE_SIZE/(NUM_SEGMENT_NAND_U
SED*2)*2))/4)*4);NAND flash bbi column address

END
```

6. Edit the `sources` file located at:
`\WINCE600\PLATFORM\iMX25-3DS-PDK1_x\SRC\COMMON\NANDFMD\sources`. This file has the definitions of which INCLUDE is to be selected on the `nandbsp.h` file. Based on the same memory model example, the file must include the following lines of code and the changes are marked in bold.

```
!if 0
Copyright (C) 2007-2008, Freescale Semiconductor, Inc. All Rights Reserved.
THIS SOURCE CODE, AND ITS USE AND DISTRIBUTION, IS SUBJECT TO THE TERMS
AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT
!endif

TARGETNAME=nandfmd_lib
TARGETTYPE=LIBRARY
RELEASETYPE=PLATFORM
SYNCHRONIZE_BLOCK=1

WINCEOEM=1
WINCECPU=1
NOMIPS16CODE=1
```

```
INCLUDES=\
    $(INCLUDES); \
    $(_PUBLICROOT)\common\oak\drivers\block\msflashfmd\inc; \
    $(_PLATFORMROOT)\common\src\soc\$(_COMMONSOCDIR)\nand\inc; \
      $(_PLATFORMROOT)\common\src\soc\$(_COMMONSOCDIR)\boot\fmd\nand

!IF "$(BSP_NAND_K9LAG08U0M)" == "1"       -> Environment variable to select the memory to
be used. This has to be named the same as in the nandbsp.h definition just added.

CDEFINES=$(CDEFINES) -DBSP_NAND_K9LAG08U0M    -> Modify the last part with the name model
just added on the nandbsp.h file.

!ENDIF                                                    -> End of IF definition


!IF "$(BSP_NAND_K9LBG08U0M)" == "1"
CDEFINES=$(CDEFINES) -DBSP_NAND_K9LBG08U0M

!ENDIF

SOURCES=nandbsp.cpp
```

7. Edit the `sources` file located at:
   `\WINCE600\PLATFORM\iMX25-3DS-PDK1_x\src\BOOTLOADER\XLDR\NAND\sources`. This file has the definitions of which INCLUDE is to be selected on the `nandchip.inc` file. Based on the same memory model example, the file must include the following lines of code and the changes are marked in bold.

```
!IF "$(IMGSDMMC)" == "1" || "$(IMGCSPIFLASH)" == "1"
SKIPBUILD=1
!ENDIF

TARGETNAME=xldr
TARGETTYPE=PROGRAM

RELEASETYPE=PLATFORM
EXEENTRY=StartUp
NOMIPS16CODE=1
SYNCHRONIZE_DRAIN=1

ADEFINES=-pd "_TGTCPU SETS \"$(_TGTCPU)\"" $(ADEFINES)

!IF "$(BSP_NAND_K9LBG08U0M)" == "1"
ADEFINES=$(ADEFINES) -pd "BSP_NAND_K9LBG08U0M SETL {TRUE}"
!ELSE IF "$(BSP_NAND_K9LBG08U0D)" == "1"
ADEFINES=$(ADEFINES) -pd "BSP_NAND_K9LBG08U0D SETL {TRUE}"
!ELSE
ADEFINES=$(ADEFINES) -pd "BSP_NAND_K9LAG08U0M SETL {TRUE}"
!ENDIF

INCLUDES=\
$(INCLUDES); \
$(_PLATFORMROOT)\common\src\soc\$(_COMMONSOCDIR)\nand\inc

LDEFINES=$(LDEFINES) -subsystem:native /DEBUG /DEBUGTYPE:CV /merge:.asecure=.astart
/FIXED:NO
CDEFINES=$(CDEFINES) -DEBOOTSHIP /Os /Og
SOURCES= \
  xldr.s \
  xldr_init.s
```

```
NOLIBC=1

WINCETARGETFILES=xldr_bin
```

1.3 Configuring the Environment

To configure the environment, perform the following steps:

1. Add the new NAND model environment variable to the project.
   a) On platform builder, go to Project > (Project Name) Properties > Configuration Properties > Environment > New... as shown in Figure 2.

Figure 2 shows how to add an environment variable to a project.

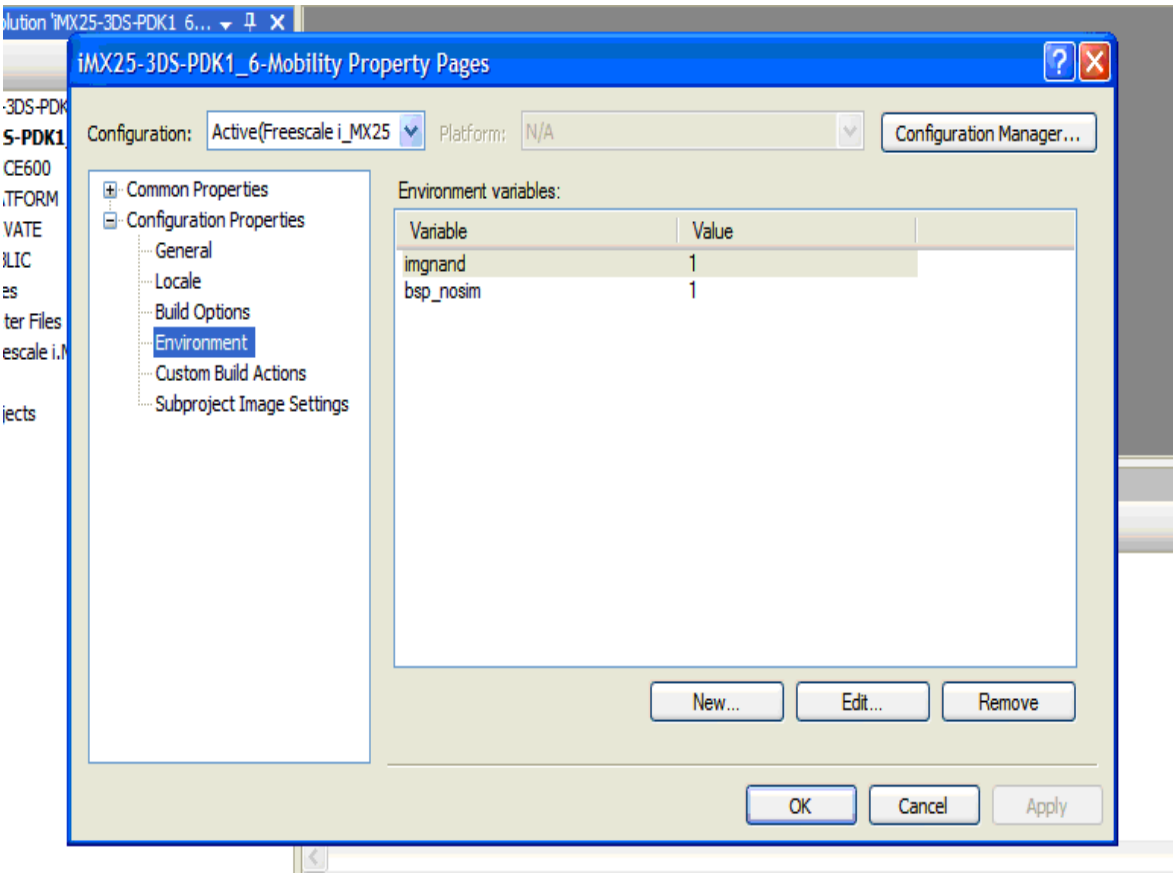

Figure 2. Adding Environment Variable to the Project

2. In the Variable name field, enter the variable defined in the sources file, which was modified in step 4 in Section 1.2, “Inclusion of New NAND Flash Models.” In the example shown in Figure 3, the variable name is BSP_NAND_K9LAG08U0M and the variable value must be equal to 1.

Figure 3 shows the variable name and its value.

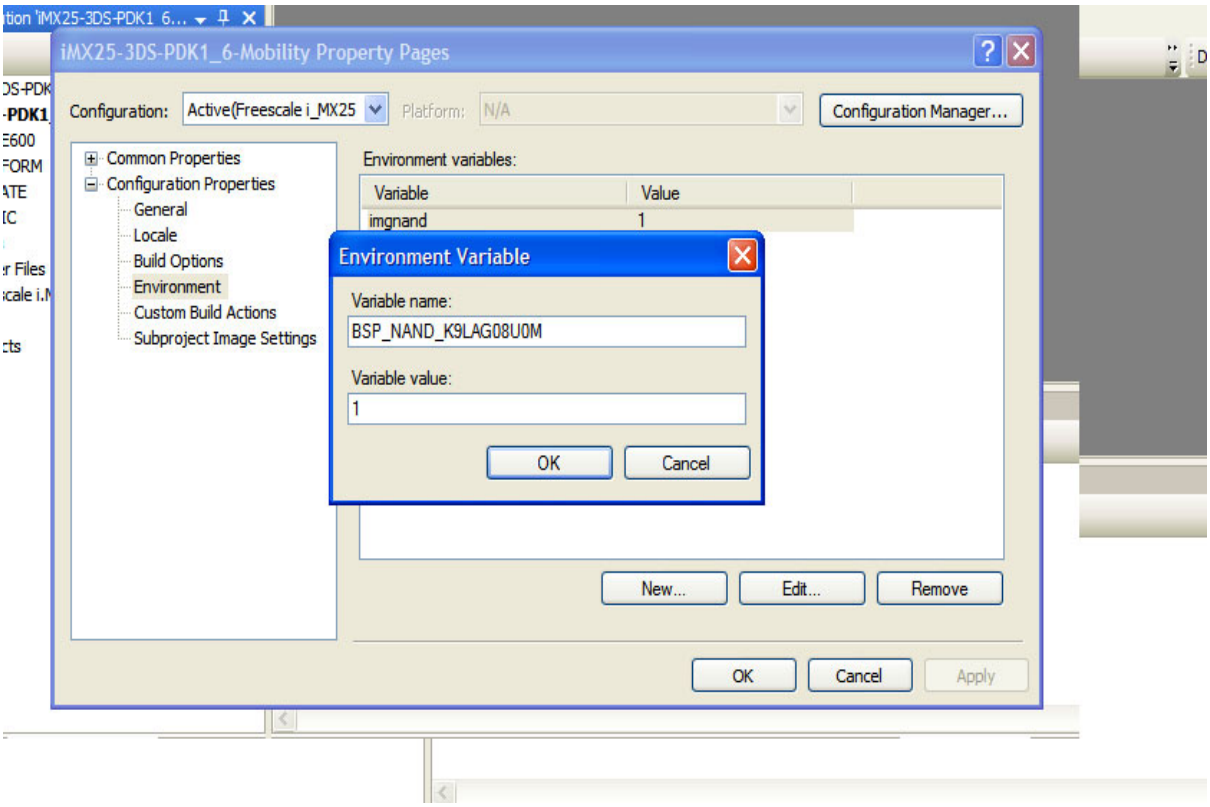

Figure 3. Entering Variable Name and Value

3. Build the project within the FMD changes.
4. On platform builder, go to Build > Advanced Build Commands > Build Current BSP and Subprojects as shown in Figure 4. This process builds the image, the boot loader, and the XLDR.

Figure 4 shows how to build the project.

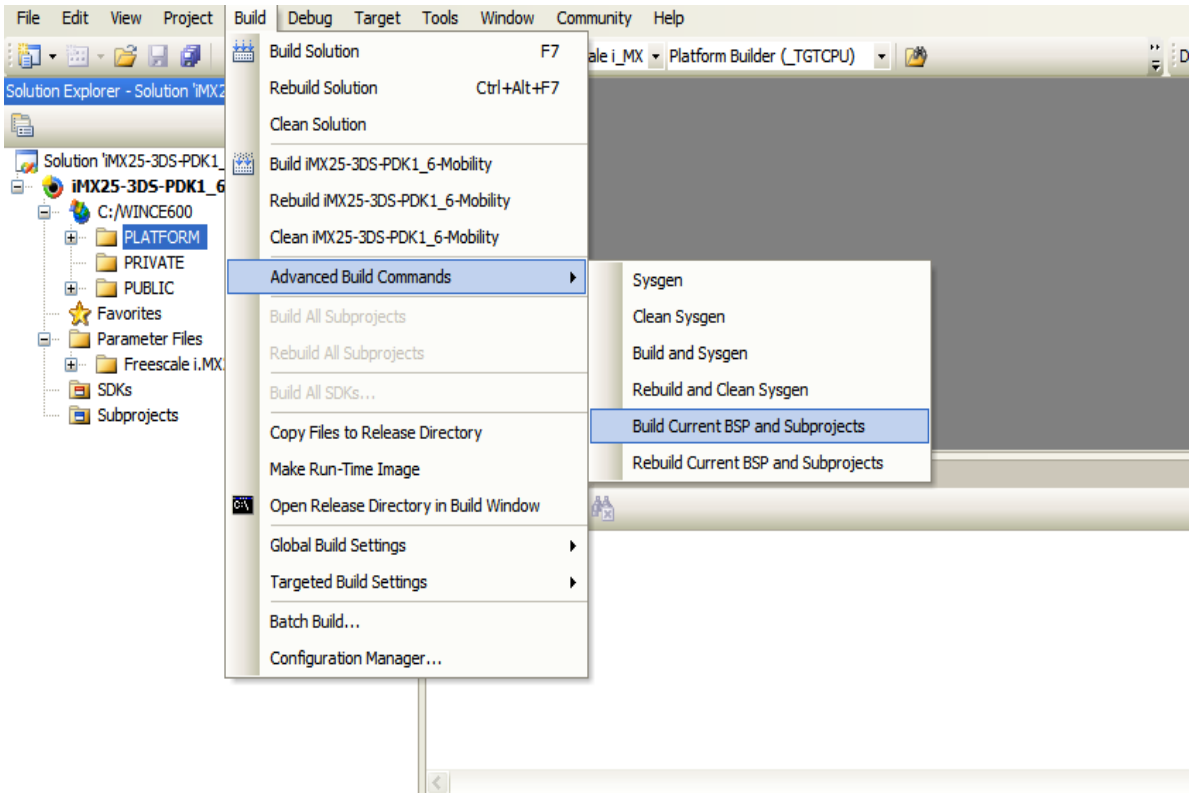

Figure 4. Building Current BSP and Subprojects

NOTE

To download the binary image to a blank NAND Flash, refer to Chapter 5 and 6 of the *User Guide document* of the SDK in use. This document provides guidelines to program the design through the ATK application.

2 Revision History

Table 1 provides a revision history for this application note.

Table 1. Document Revision History

Rev. Number	Date	Substantive Change(s)
0	04/2010	Initial release

Appendix A NAND Flash Parameters

The following section decribes the NAND Flash parameters.

A.1 Bad Block Information (BBI)

Every NAND flash defines a BBI byte(s) that contains the invalid block(s). When the Near Field Communication (NFC) reads or writes to a specific NAND Flash, it first checks the BBI byte in order to validate the block to be used. The location of this BBI byte depends on the page size and the spare area of each NAND Flash model.

The NAND flash driver for WinCE6.0 defines three parameters to store the BBI:

- BBI MAIN ADDR
- BBI NUM
- BBIMarkPage

A.2 BBI MAIN ADDR

This parameter defines the address location on the page, where the BBI is to be stored. This address depends on the page size and the number of bytes available on the spare area. Generally, the BBI is stored on the first byte of the spare area and below are two examples of how to get the BBI address.

A.2.1 Using 2 Kbyte page size with 64 bytes of spare area

Generally, the spare area for this NAND flash model is at the beginning, which is the 2048 address. However, the NFC divides the 2 Kbyte page into four pages (512 + 16) with its corresponding spare area, which is 16 bytes in this case. According to the NFC format, the value for the address should be $2048 - ((512 + 16) \times 3) = 464$.

The following points describe each of the values that are used to manipulate the address:

- 2048—First byte of the spare area on the NAND flash.
- 512—Page size on the internal NFC.
- 16—Size of spare area of each internal buffer on NFC.
- 3—Represents the last buffer of NFC minus one, to reach the last buffer of the NAND flash.

A.2.2 Using 4 Kbyte page size with 218 bytes of spare area

Generally, the spare area for this NAND flash model is at the beginning, which is the 4096 address. However, the NFC divides the 4 Kbyte page into eight pages (512 + 26) with its corresponding spare area, which is 26 bytes in this case. According to the NFC format, the value for the address should be $4096 - ((512 + 26) \times 7) = 330$.

The following points describe each of the values that are used to manipulate the address:

- 4096—First byte of the spare area on the NAND flash.
- 512—Page size on the internal NFC.

- 26—Size of spare area of each internal buffer on NFC.
- 7—Represents the last buffer of NFC minus one, to reach the last buffer of the NAND flash.

The following are the most common values of BBI MAIN ADDR:

```
#DEFINE BBI_MAIN_ADDR (464) //Use this for 2 K page NAND
#DEFINE BBI_MAIN_ADDR (400) //Use this for 4 K page 128 byte spare NAND, 4 bit ECC
#DEFINE BBI_MAIN_ADDR (330) //Use this for 4 K page 218 byte spare NAND, 8 bit ECC
```

A.3 BBI NUM

This parameter handles the number of pages in a block to check for BBI, and is equal to one in most cases. However, some vendors also use two. For more information, check the *NAND Flash manual* of each model in order to know this value.

A.4 BBIMarkPage

This parameter is an array that contains the address page, where the BBI is to be checked for each block. Generally, this page is the first or the last page. For instance, on a 128 page per block NAND Flash model, this value is either 0 or 127 for the first or last page. If two or more BBI bytes are used in a page, the BBIMarkPage array must contain all these pages.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or
+1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064
Japan
0120 191014 or
+81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
1-800 441-2447 or
+1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

Document Number: AN4111
Rev. 0
04/2010