AN11387

In-Application Programming for the LPC11Axx

Rev. 1 — 12 August 2013

Application note

Document information

Info	Content
Keywords	In-Application Programming, IAP, LPC11Axx
Abstract	This application note will describe the In-Application Programming capabilities of the LPC11Axx

Revision history

Rev	Date	Description
1	20130812	Initial version.

Contact information

For more information, please visit: http://www.nxp.com

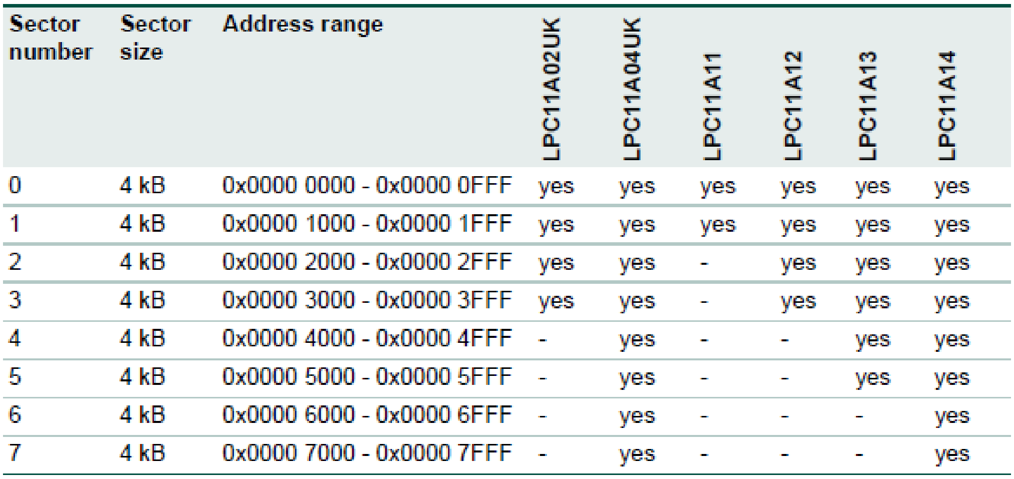
For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

In-Application Programming (IAP) allows manipulation of the on-chip flash memory while running the user application code. The IAP routines, located in the Boot ROM, can be used for in-field application programming updates or as a data storage element in flash memory.

2. Flash memory layout

The on-chip flash memory of the LPC11Axx is grouped in sectors. Each sector is 4 kB in size. The amount of on-chip flash memory available to the user is dependent on the particular LPC11Axx device. The flash layout is shown in Fig 1.



Sector number	Sector size	Address range	LPC11A02UK	LPC11A04UK	LPC11A11	LPC11A12	LPC11A13	LPC11A14
0	4 kB	0x0000 0000 - 0x0000 0FFF	yes	yes	yes	yes	yes	yes
1	4 kB	0x0000 1000 - 0x0000 1FFF	yes	yes	yes	yes	yes	yes
2	4 kB	0x0000 2000 - 0x0000 2FFF	yes	yes	-	yes	yes	yes
3	4 kB	0x0000 3000 - 0x0000 3FFF	yes	yes	-	yes	yes	yes
4	4 kB	0x0000 4000 - 0x0000 4FFF	-	yes	-	-	yes	yes
5	4 kB	0x0000 5000 - 0x0000 5FFF	-	yes	-	-	yes	yes
6	4 kB	0x0000 6000 - 0x0000 6FFF	-	yes	-	-	-	yes
7	4 kB	0x0000 7000 - 0x0000 7FFF	-	yes	-	-	-	yes

Fig 1. LPC11Axx flash memory layout

The IAP erase and write operations are always sector-by-sector, i.e., when any portion of a sector needs to be modified, the erase/write operation must be performed on the entire flash sector.

The IAP does not allow direct flash memory manipulation. To write data from one flash location to another, place the data into RAM, then write the data into the target flash location.

3. IAP initialization

The IAP routines are located in the Boot ROM. In order to access the IAP routines, the entry point of the IAP must be defined. For the LPC11Axx, the address is 0x1FFF1FF1.

```
#define IAP_LOCATION 0x1FFF1FF1
```

A function pointer to the function type is then defined.

```
typdef void (*IAP)(unsigned int [], unsigned int[]);
IAP iap_entry;
```

The IAP routines take two unsigned 32-bit integer arrays, the command_param and status_result, as input. The command_param is a 5 element array and the status_result is a 4 element array. The arrays can be defined as:

```
unsigned int command_param[5];
unsigned int status_result[4];
```

or they can be defined as:

```
unsigned int * command_param;
unsigned int * status_result;
command_param = (unsigned int *) <address>
status_result = (unsigned int *) <address>
```

To make a call to the IAP routines, the following code can be used:

```
Iap_entry (command_param, status_result);
```

3.1 IAP routines

The list of available IAP routines is listed below.

IAP command	Code (base 10)	Functional description	Precautions
Prepare sector(s) for write operation	50	Turns off the write protection for the specified flash sectors.	This function must be called prior to executing “Copy RAM to Flash” or “Erase Sector(s)” commands.
Copy RAM to Flash	51	Performs a write operation from RAM to flash memory.	A flash sector must be prepared for write operation before contents can be written. Ensure no other flash accesses are performed during the copy procedure. Source data must be located in RAM.
Erase Sector(s)	52	Erases the contents of the entire flash sector(s). Erased flash sector(s) will read back with all bits set to 1’s.	A flash sector must be prepared for write operation before it can be erased. Ensure no other flash accesses are performed during the erase procedure.
Blank check sector(s)	53	Determines if flash sector(s) is (are) erased.	None
Read part identification number	54	Returns the identification number of a particular part. See the user manual for the specific part identification numbers.	None
Read boot code version number	55	Returns the boot ROM version number.	None
Compare (memory)	56	Compares memory contents at two locations.	None
Re-invoke ISP	57	This function call will invoke the ISP routine located on the boot ROM.	Calling this function will remap the boot vectors, enable UART0 and Timer1 and change their PCLK values to CCLK/4.
Read device serial number	58	Returns the part’s unique serial number.	None

3.2 Command and result array

The IAP routines require two array functions to be passed to it: the command and result array. The first element of the command array will always be the command code with the subsequent elements as optional parameters.

The first element of the result array will always be the status code. Subsequent elements will be the optional results.

4. IAP precautions

The IAP manipulates the flash memory during run-time. As such, certain precautions must be observed to ensure proper operations.

4.1 Interrupts

When IAP routines are used, any access to the flash memory must be avoided during erase or write operations. If the vector interrupt table is located in the flash, all interrupts must be disabled prior to an erase or write operation.

The LPC11Axx has the ability to remap the interrupt vector to the RAM by changing the value of the MAP bits in the SYSMEMREMAP register. If the interrupts are remapped, then it is possible to allow interrupts to occur during an erase or write operation. However, since the flash cannot be accessed during this time, the interrupt handlers must execute from RAM. Therefore, all code related to interrupt handlers must be copied from flash into RAM.

4.2 RAM usage

The IAP routines utilize 32 bytes of space in the top portion of the on-chip RAM for execution and up to 128 bytes of stack space. The user program should not use this space if IAP flash programming is permitted in the application. Furthermore, if the interrupt vector is remapped to the SRAM, the bottom 512 bytes of the memory map should not be used.

4.3 Sector 0

The following items are stored in sector 0 of the flash:

- Interrupt vector table
- Code Read Protection (CRP)
- Valid user code checksum

The user application must ensure these three items are properly assigned. If these items are written incorrectly or accidentally erased, the device will not perform as expected.

4.4 Preparing the sector

Before a sector can be manipulated, the prepare sector command must be executed. On each successful completion of “Copy RAM to flash” or “Erase Sector(s)” command, the associated sector will be protected. Thus, to make incremental changes, i.e., number of bytes less than the sector size (4 kB), the prepare sector command must be executed multiple times.

4.5 Invoking ISP

Before the Invoke ISP can be called, some housekeeping must be done to ensure the proper entry into ISP. These items must be done before making a call to Invoke ISP:

- Enable Timer 1
- Enable the GPIO domain
- Enable the IO configuration clock
- Set the SYSAHBCLKDIV to a 1:1 ratio
- Set the stack pointer to the reset default value of 0x1FFF0000

5. Software setup

The demonstration software includes the IAP routines in two files: iap.h and iap.c. The IAP commands and results are represented as enum.

NOTE – The commands are represented in decimal.

```
typedef enum
{
    IAP_PREPARE = 50,            // Prepare sector(s) for write operation
    IAP_COPY_RAM2FLASH = 51,     // Copy RAM to Flash
    IAP_ERASE = 52,              // Erase sector(s)
    IAP_BLANK_CHECK = 53,        // Blank check sector(s)
    IAP_READ_PART_ID = 54,       // Read chip part ID
    IAP_READ_BOOT_VER = 55,      // Read chip boot code version
    IAP_COMPARE = 56,            // Compare memory areas
    IAP_REINVOKE_ISP = 57,       // Reinvoke ISP
    IAP_READ_SERIAL_NUMBER = 58, // Read serial number
}  IAP_COMMAND_CODE;
```

The status code for the IAP is also represented as an enum

```
typedef enum
{
    CMD_SUCCESS,        // Command is executed successfully.
    INVALID_COMMAND,    // Invalid command.
    SRC_ADDR_ERROR,     // Source address is not on a word boundary.
    DST_ADDR_ERROR,     // Destination address is not on a correct boundary.
    SRC_ADDR_NOT_MAPPED, // Source address is not mapped in the memory map.
    DST_ADDR_NOT_MAPPED, // Destination address is not mapped in the memory map.
    COUNT_ERROR,        // Byte count is not multiple of 4 or is not a permitted
value.
    INVALID_SECTOR,     // Sector number is invalid.
    SECTOR_NOT_BLANK,   // Sector is not blank.
    SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION,     // Command to prepare sector
for write operation was not executed.
    COMPARE_ERROR,      // Source and destination data is not same.
    BUSY,              // Flash programming hardware interface is busy.
} IAP_STATUS_CODE;
```

Conversely, the commands and results can be represented with individual #define.

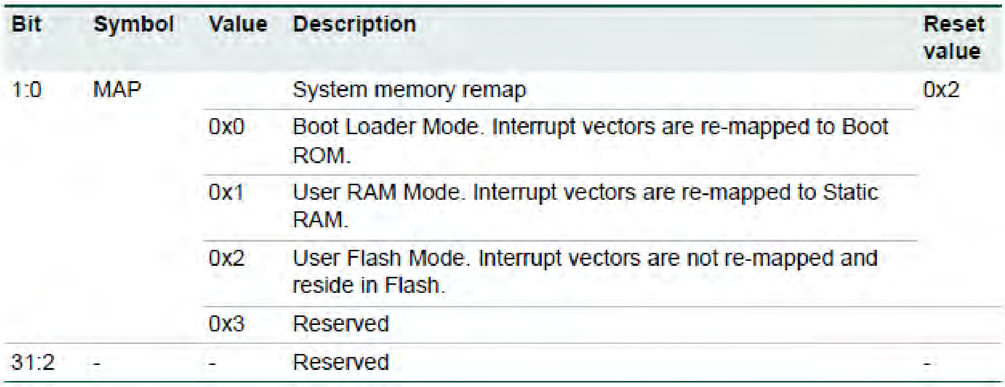
5.1 Functions return variables

The IAP functions return an IAP_STATUS_CODE type to indicate the result of the function call. As such, a variable of the IAP_STATUS_CODE needs to be defined:

```
IAP_STATUS_CODE iap_ret;
```

5.2 Interrupt remapping

The system memory remap register SYSMEMREMAP on NXP's LPC11Axx, LPC11Uxx and LPC11xx MCUs selects whether the exception vectors are read from boot ROM, flash, or SRAM. By default, the flash memory is mapped to address 0x00000000. When the MAP bits in the SYSMEMREMAP register are set to 0x0 or 0x1, the boot ROM or RAM respectively are mapped to the bottom 512 bytes of the memory map (addresses 0x0000 0000 to 0x0000 0200).



Bit	Symbol	Value	Description	Reset value
1:0	MAP		System memory remap	0x2
		0x0	Boot Loader Mode. Interrupt vectors are re-mapped to Boot ROM.	
		0x1	User RAM Mode. Interrupt vectors are re-mapped to Static RAM.	
		0x2	User Flash Mode. Interrupt vectors are not re-mapped and reside in Flash.	
		0x3	Reserved	
31:2	-	-	Reserved	-

Fig 2. SYSMEMREMAP Register

For interrupt handling during IAP on these MCU families, user code should copy the interrupt vector table from 0x00000000 to 0x10000000 and then set MAP bits to be 0x1 to select the exception vector from RAM. Note that the entire lower 512 byte flash block should be copied to RAM.

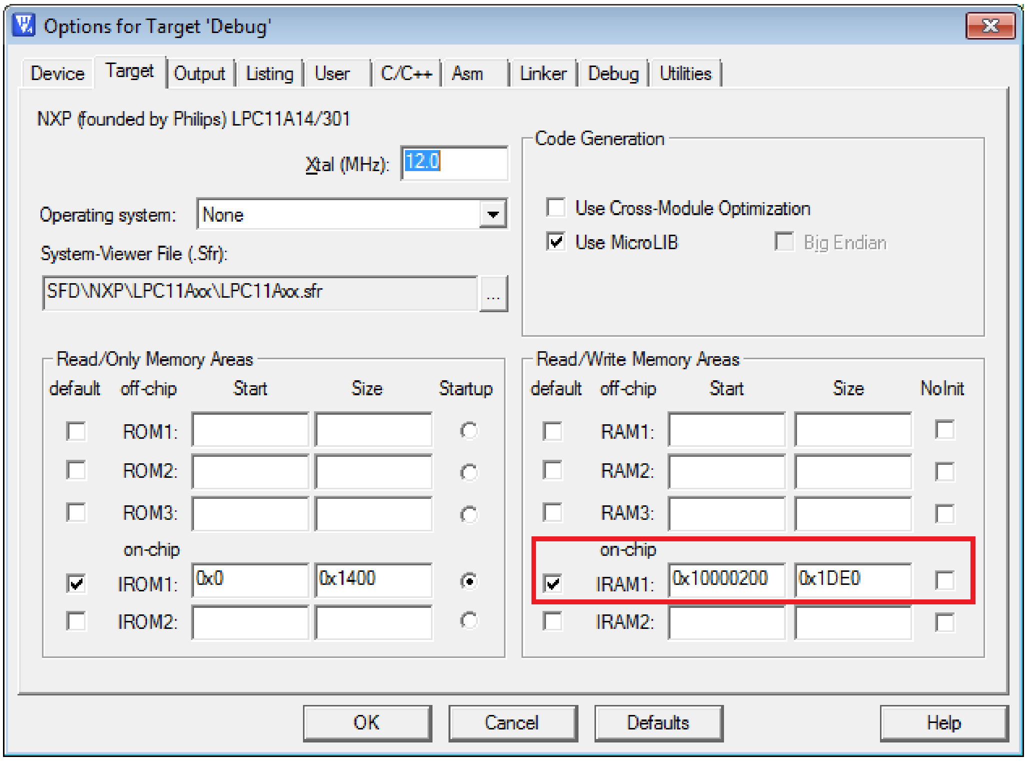
5.3 SRAM memory mapping

The demo software relocates the interrupt vector to the SRAM and uses the IAP code, i.e., the compiler must be configured such that the bottom 512 bytes and top 32 bytes of the memory cannot be touched. In the Keil environment, the IRAM1 section should be specified to be smaller than the actual SRAM size to prevent the compiler from using these areas.

The SRAM starts at location 0x10000000. Since the interrupt vector table uses 512 bytes of the bottom of the SRAM, the start location is now set to 0x10000200.

The total SRAM size for the LPC11A14 is 8 kB. With IAP using 32 bytes in the top of the SRAM, this means the usable SRAM size is 8 kB – 32 bytes, or 8160 bytes. But since 512 bytes is also being used by the interrupt vector table, the SRAM size now becomes:

8 kB – 32 bytes – 512 bytes = 7648 bytes.



NXP (founded by Philips) LPC11A14/301

Xtal (MHz): 12.0

Operating system: None

System-Viewer File (.Sfr):
SFD\NXP\LPC11Axx\LPC11Axx.sfr

Code Generation
☐ Use Cross-Module Optimization
☑ Use MicroLIB ☐ Big Endian

Read/Only Memory Areas

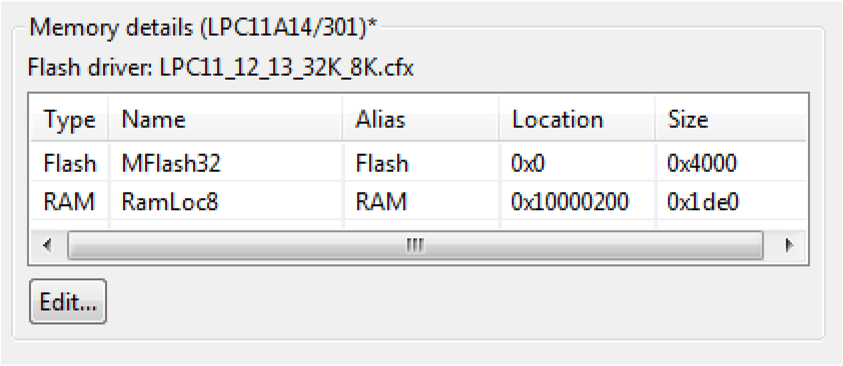
default	off-chip	Start	Size	Startup
☐	ROM1:			○
☐	ROM2:			○
☐	ROM3:			○
	on-chip			
☑	IROM1:	0x0	0x1400	●
☐	IROM2:			○

Read/Write Memory Areas

default	off-chip	Start	Size	NoInit
☐	RAM1:			☐
☐	RAM2:			☐
☐	RAM3:			☐
	on-chip			
☑	IRAM1:	0x10000200	0x1DE0	☐
☐	IRAM2:			☐

Fig 3. Keil IROM and SRAM remapping

In the LPCXPresso IDE, the same task is accomplished by changing the MCU settings.



Memory details (LPC11A14/301)*

Flash driver: LPC11_12_13_32K_8K.cfx

Type	Name	Alias	Location	Size
Flash	MFlash32	Flash	0x0	0x4000
RAM	RamLoc8	RAM	0x10000200	0x1de0

Fig 4. LPCXPresso flash and SRAM remapping

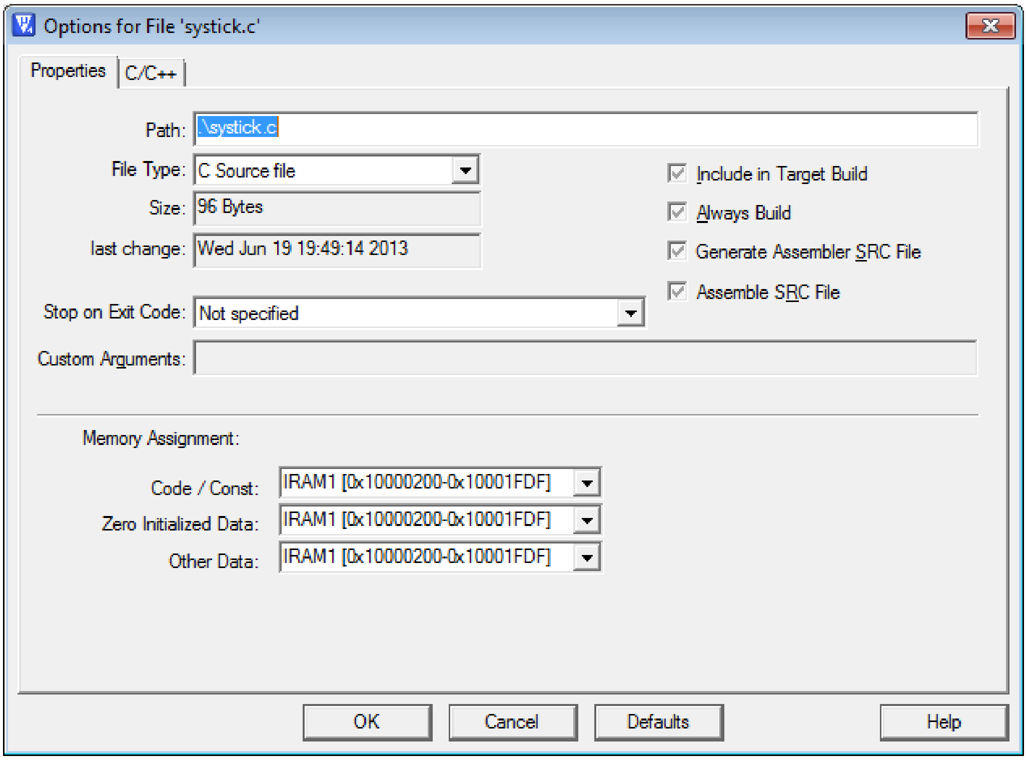
5.4 Flash memory remapping

The demo software manipulates sector 2 to sector 7 of the flash. To prevent the compiler from using those sectors, an artificial limitation of 5 kB is imposed on it. To do this, the IROM1 is set to a size of 5 kB (0x1400) for the Keil environment.

In the LPCXpresso IDE, sector 4 to sector 7 is manipulated. For this, the flash size is artificially limited to 16 kB in size.

5.5 Systick interrupt

The systick is used to create a periodic interrupt while the software is running. On each interrupt, PIO0_16 is toggled. Since during an IAP call the flash is not accessible to the software, the Systick interrupt handler is relocated to the SRAM. This is done within the Keil environment as opposed to manually editing the scatter file.



Path: .\systick.c
File Type: C Source file
Size: 96 Bytes
last change: Wed Jun 19 19:49:14 2013
Stop on Exit Code: Not specified
Custom Arguments:

☑ Include in Target Build
☑ Always Build
☑ Generate Assembler SRC File
☑ Assemble SRC File

Memory Assignment:
Code / Const: IRAM1 [0x10000200-0x10001FDF]
Zero Initialized Data: IRAM1 [0x10000200-0x10001FDF]
Other Data: IRAM1 [0x10000200-0x10001FDF]

Fig 5. Systick interrupt handler settings in Keil

For the LPCXpresso IDE, the systick handler function is directed to be placed into the SRAM by the using the **.data.ramfunc** directive.

```
__attribute__ ((__section__(".data.ramfunc")))
void SysTick_Handler(void){
        LPC_GPIO_PORT->NOT[0] = (1<<16);}
```

5.6 Application example

The demo software sends a menu system through the UART at 9600, 8, N, 1. The menu system allows erasing and writing to sector 2 through sector 7 in the Keil IDE and sector 4 through sector 7 for the LPCXpresso IDE. Sector 0 contains the software so modification of this sector is not allowed.

The demo software demonstrates the IAP calls of:

- Erasing sector 2 – 7 (Keil) or sector 4 – 7 (LPCXpresso)
- Writing sector 2 – 7 (Keil) or sector 4 -7 (LPCXpresso) with a fill pattern of 0xAA
- Reading the Boot Version
- Reading the device serial number. The serial number is presented as HEX format. To do a comparison against the serial number provided by the ISP, the serial number will have to be converted from HEX to decimal.
- Reading the device
- Invoking ISP

The demo software also demonstrates how interrupts can be handled while using IAP calls by remapping the interrupt vector table to the SRAM.

6. Legal information

6.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

6.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

6.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

7. List of figures

8. Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

For more information, visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 12 August 2013
Document identifier: AN11387