AN11555

LPC15xx In-Application Programming

Rev. 1 — 3 June 2014 Application note

Document information

Info	Content
Keywords	LPC15xx, IAP
Abstract	This application note describes the In-Application Programming capabilities of LPC15xx.

Revision history

Rev	Date	Description
1	20140603	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

The LPC15xx are an ARM Cortex-M3 based microcontroller family that can operate up to 72 MHz. The LPC15xx support up to 256 kB of flash memory, a 4 kB EEPROM, and 36 kB of SRAM.

The ARM Cortex-M3 uses a 3-stage pipeline and includes an internal prefetch unit that supports speculative branching. The peripheral compliment includes one full-speed USB 2.0 device, two SPI interfaces, three USARTs, one Fast-mode Plus I^2C-bus interface, one C_CAN module, PWM/timer subsystem with four configurable, multi-purpose State Configurable Timers(SCTimer/PWM) with input-preprocessing unit (SCT-IPU), a Real-time clock module with independent power supply and a dedicated oscillator, two 12-channel/12-bit, 2 Msample/s ADCs, one 12-bit, 500 ksample/s DAC, four voltage comparators with internal voltage reference, and a temperature sensor. A DMA engine can service most peripherals.

The In-Application Programming (IAP) allows manipulation of the on-chip flash memory while running user application code. The IAP routines located in the BOOT ROM can be used to operate on flash or to get certain information stored in on-chip ROM like the boot code version, part ID and unique ID of the chip. This can include:

- Field firmware upgrade
- EEPROM content replacement
- Data storage

2. Flash specifications

The on-chip flash memory of the LPC15xx is grouped as sectors. The flash memory is divided into 64 sectors and the size of each sector is 4 kB. Individual pages of 256 byte each can be erased using the IAP erase page command. One sector contains 16 pages.

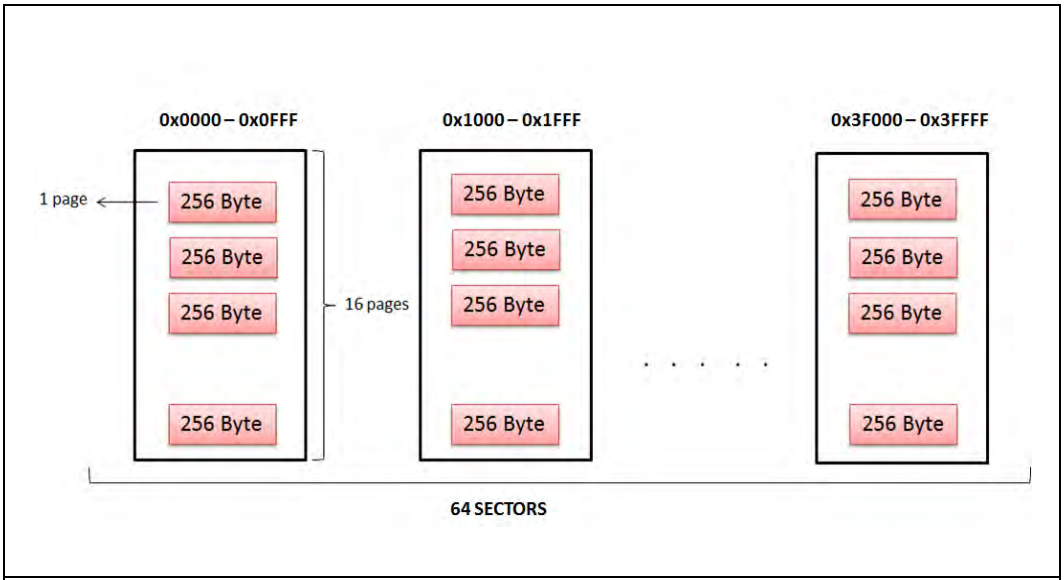

Fig 1. LPC15xx flash structure

Sector number	Sector size [kB]	Page number	Address range
0 - 15	4	0 -255	0x0000 0000 - 0x0000 FFFF
16 - 31	4	256 - 511	0x0001 0000 - 0x0001 FFFF
32 - 47	4	512 - 767	0x0002 0000 - 0x0002 FFFF
48 - 63	4	768 - 1023	0x0003 0000 - 0x0003 FFFF

Fig 2. LPC15xx flash address mapping

3. EEPROM

The LPC15xx contains 4 kB of on-chip byte-erasable and byte-programmable EEPROM data memory. The EEPROM can be programmed using In-Application Programming via the on-chip boot loader software.

4. Introduction to In-Application Programming

4.1 IAP initialization

The IAP routines are located in the Boot ROM. The IAP routine resides at 0x0300 0205 location and it is thumb code. To access the IAP routines the entry point for IAP has to be defined. For LPC15xx, the address is 0x0300 0205.

```
1    #define IAP_LOCATION 0x0300 0205
```

The IAP routines take two unsigned 32-bit integer arrays as input; the command_param and status_result. For each IAP command the number of parameters and results vary. The maximum number of parameters is 5, passed to the "Copy RAM to FLASH" command. The maximum number of results is 5, returned by the "ReadUID" command. Hence both command_param and status_result are 5 element arrays. The arrays can be defined as:

```
2    unsigned int command_param[5];
3    unsigned int status_result[5];
```

or they can be defined as:

```
4    unsigned int * command_param;
5    unsigned int * status_result;
6    command_param = (unsigned int *) <address>
7    status_result = (unsigned int *) <address>
```

4.2 IAP routines

Table 1. IAP routines

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).	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 and configures the peripherals for ISP (UART, C_CAN or USB). Before calling this command, the clocks to GPIO0/1/2 blocks should be enabled in the SYSAHBCLKCTRL0 register.
Read device serial number	58	Returns the part’s unique serial number.	None
Erase Page	59	Erases a page or multiple pages of on-chip flash memory.	The page has to be prepared for write operation before it can be erased. Ensure no other flash accesses are performed during the erase procedure.

IAP command	Code (base 10)	Functional description	Precautions
EEPROM Write	61	Data is copied from the RAM address to the EEPROM address	The top 64 bytes of the 4kB EEPROM are reserved and cannot be written to.
EEPROM Read	62	Data is copied from the EEPROM address to the RAM address.	None

4.3 IAP precautions

The IAP manipulates the memory during run-time. Therefore, certain precautions have to be taken to ensure proper operations.

4.3.1 Interrupts

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

The LPC15xx has the ability to remap the interrupt vector table to the RAM by changing the MAP bits in the SYSMEMREMAP register. This allows interrupts to occur even during the erase and write operations. But as the flash cannot be accessed during this time, the interrupt handlers must be executed from the RAM. Hence, all the code related to the interrupt handlers must be copied from Flash into the RAM.

4.3.2 RAM usage

The IAP routines utilize 32 bytes of space in the top portion of the on-chip RAM for execution (address 0x2008FDF to 0x2008FFF) and up to 128 bytes of stack space. The user program should not use this space if the IAP flash programming is permitted in the application. Furthermore, if the interrupt vector table is remapped to the SRAM, the bottom 512 bytes of the memory map should not be used.

5. IAP sample project and interrupt handling

5.1 Software setup

5.1.1 SRAM memory mapping

The demonstration code relocates the interrupt vector table to SRAM and uses the IAP code. This means that the compiler must be configured such that the bottom 512 bytes and the 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 address 0x0200 0000. Since the interrupt vector table uses 512 bytes of the bottom of SRAM, the start location is set to 0x0200 0200. The SRAM size of LPC15xx is 36 kB. With the IAP using the 32 bytes in the top of SRAM, this means the usable SRAM size is 36kB – 32 bytes i.e. 36832 bytes. But since 512 bytes is also being used by the interrupt vector table, the SRAM size now becomes: (36864 – 32 – 512) = 36320 bytes i.e. 0x8DE0.

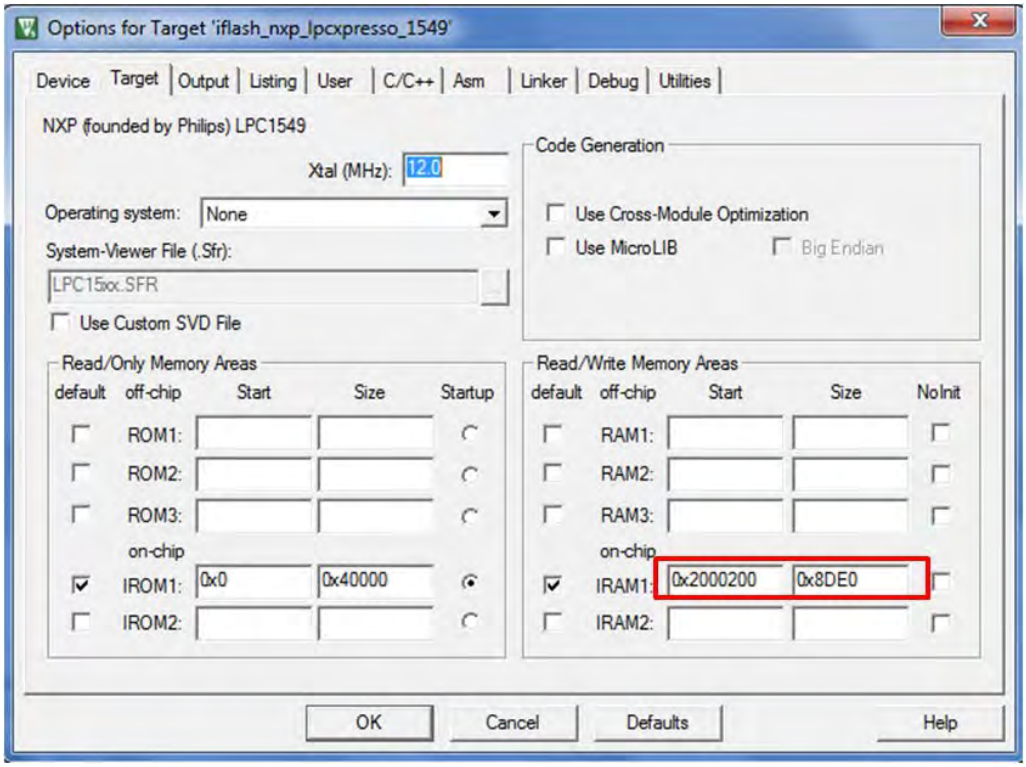

Fig 3. Keil IROM and SRAM remapping

In LPCXpresso IDE, the same task is accomplished by changing the MCU settings. Right click on '*periph_iap'* choose Properties -> C/C++ Build -> MCU settings.

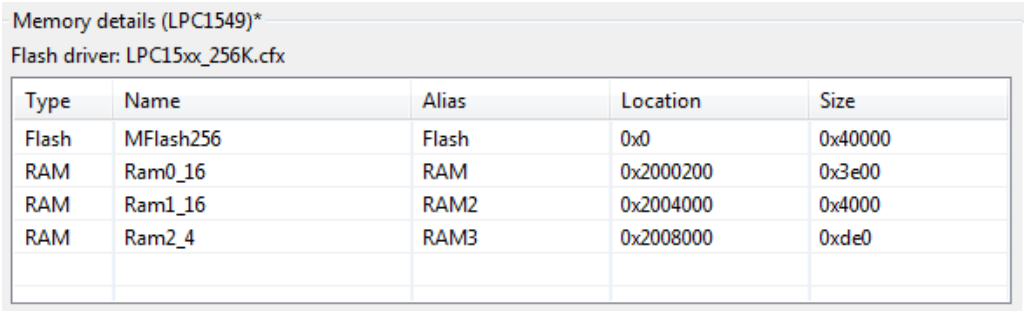

Fig 4. LPCXpresso flash and SRAM remapping

In IAR Embedded Workbench, the SRAM remapping is achieved by changing the linker configuration settings. Right click on '*periph*_iap' and click Options->Linker and change the linker configuration file. The RAM address is set to 0x0200 0200 and the end address is set to 0x0200 8DE0.

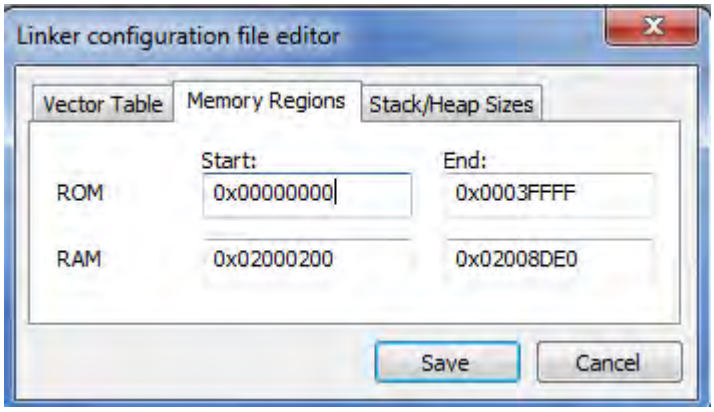

Fig 5. IAR Embedded Workbench ROM and SRAM remapping

5.1.2 Interrupt remapping

The system remap register SYSMEMREMAP on NXP's LPC15xx selects whether the exception vectors are read from the boot ROM, flash or SRAM. By default, the flash memory is mapped to the address 0X0000 0000. When the MAP bits in the SYSMEMREMAP register are set to 0x0 or 0x1, the boot ROM or RAM are respectively mapped to the bottom 512 bytes of the memory map (address 0x0000 0000 to 0x0000 0200).

Bit	Symbol	Value	Description	Reset value
1:0	MAP		System memory remap. Value 0x3 is reserved.	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.	
31:2	-	-	Reserved	-

Fig 6. SYSMEMREMAP register

So for interrupt handling during IAP, user code should copy the interrupt vector table from 0x0000 0000 to 0x0200 0000 and then set the MAP bits to be 0x1 to select the exception vector from RAM. The entire lower 512 byte flash block should be copied to RAM.

5.1.3 SysTick interrupt

The SysTick is used to create a periodic interrupt while the software is running. Since during the IAP call the flash is not accessible to the software, the SysTick interrupt handler is relocated to the SRAM.

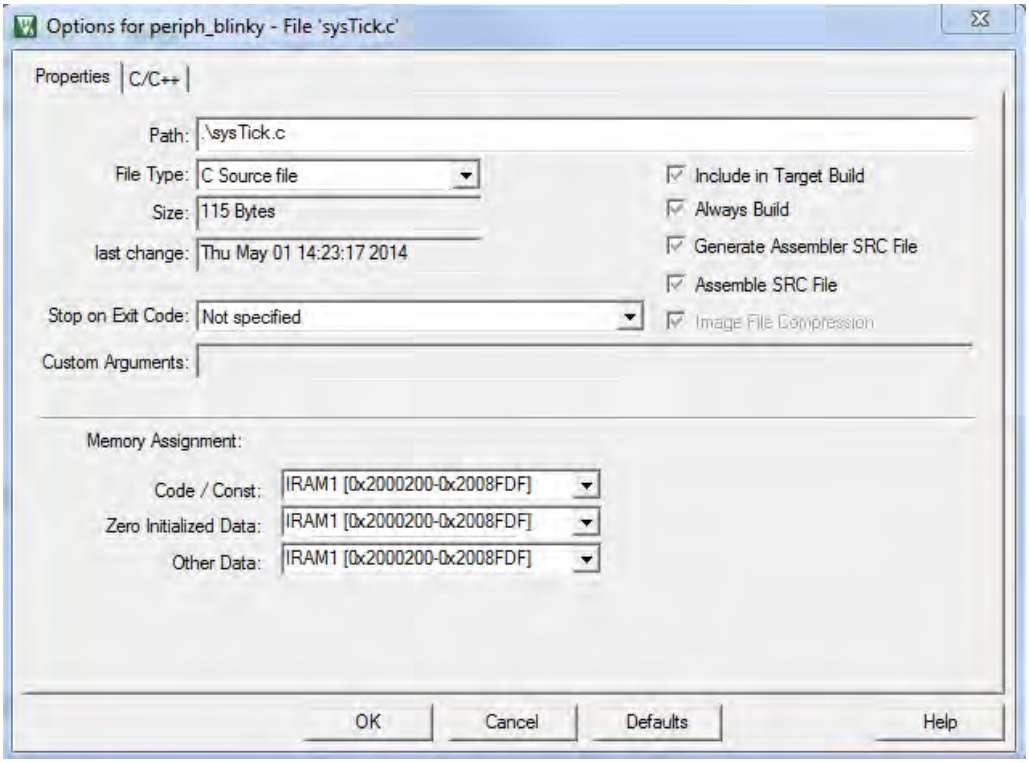

Fig 7. SysTick interrupt handler settings in Keil

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

```
8    attribute__ ((__section__(".data.ramfunc")))
9    void SysTick_Handler(void){
10   LPC_GPIO->NOT[0] = (1<<7);}
```

In the IAR Embedded Workbench, the systick handler function is placed into the SRAM by using the compiler directive **__ramfunc.**

```
11   __ramfunc void SysTick_Handler(void){
12   LPC_GPIO->NOT[0] = (1<<7);}
```

5.1.4 Handling interrupts during IAP

The LPC15xx flash is not accessible when the IAP routines are being called; interrupts are disabled during this time. In order to allow interrupts during IAP calls the interrupt vector table is relocated to SRAM.

The MAP bits in the SYSMEMREMAP register is set to 0x1, indicating the vector table is located in the SRAM and not in the flash.

```
13 CopyInterruptToSRAM();          //remap interrupt vector to SRAM
14 LPC_SYSCON->SYSMEMREMAP = 0x1;  //change memory map
```

The interrupt vector table is copied to the SRAM using the function ‘CopyInterruptToSRAM’. The function call is hardcoded to copy from flash address 0x00 to SRAM address 0x200 0000.

```
void CopyInterruptToSRAM(void)
{
  unsigned int * flashPtr, * ramPtr;
  unsigned int * uLimit = (unsigned int *) 0x200;

    ramPtr = (unsigned int *)0x2000000; //load RAM starting at 0x2000000,
    flashPtr = (unsigned int *)0x00;      //start of interrupt vector table
    while(flashPtr < uLimit)
      {
        *ramPtr = *flashPtr;
        ramPtr++;
        flashPtr++;
      }
}
```

Fig 8. Copy the IRQ handler to SRAM

In LPC15xx, during EEPROM write and read operations, the interrupts do not need to be disabled for proper IAP operation.

5.2 Hardware setup

In the LPC15xx family of microcontrollers the CCLK parameter is not used in the IAP APIs. However for compatibility with other LPC microcontroller families, the CCLK parameter is passed in the application note.

The LPC1549 Xpresso board is used to implement the demonstration code. Connect a mini-USB power cable and debugger to the board.

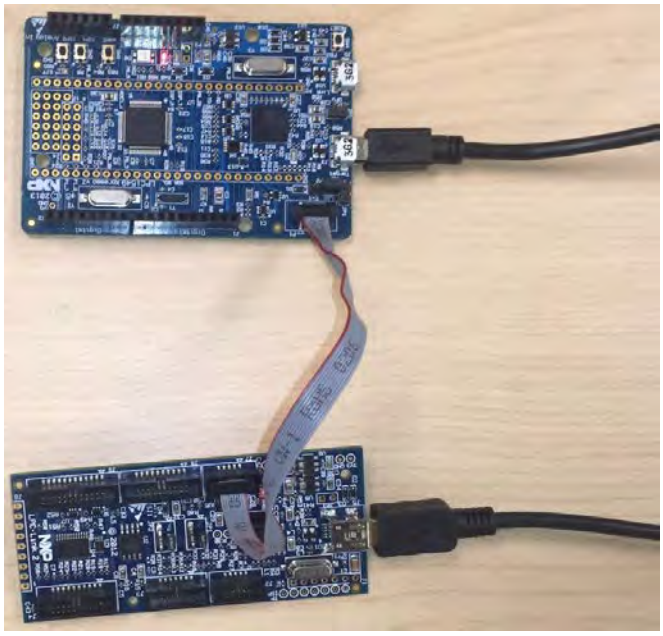

Fig 9. LPC1549 Xpresso board and LPCLink2 debugger

For the Keil IDE use the ULINK debugger (ex. UNLINK2,ULINKME, ULINKPro), for LPCXpresso IDE use the LPC Link-2 and for IAR Embedded Workbench IDE use Jlink.

The LPC1549 Xpresso board also supports CMSIS DAP which is compatible with all three tool chains: Keil, LPCXpresso and IAR. This eliminates the use of an external debugger.

5.3 Application example

The demonstration software is executed on the LPC1549 Xpresso Board (OM13056).

The board can be ordered from the following website:

http://www.nxp.com/demoboard/OM13056.html

The demonstration software is implemented in Keil, LPCXpresso and IAR IDE.

On opening the project, first build the chip library 'lib_lpc_chip_15xx'. Then build the board library 'lib_lpc_board_nxp_lpcxpresso_1549'. Finally build the IAP project 'periph_iap'. Once the build is successful click debug and run the project.

An LED on the board is used to indicate the status of the IAP operation. If the IAP operation is a success, the LED on the board glows green. If the IAP operation fails, the LED glows red.

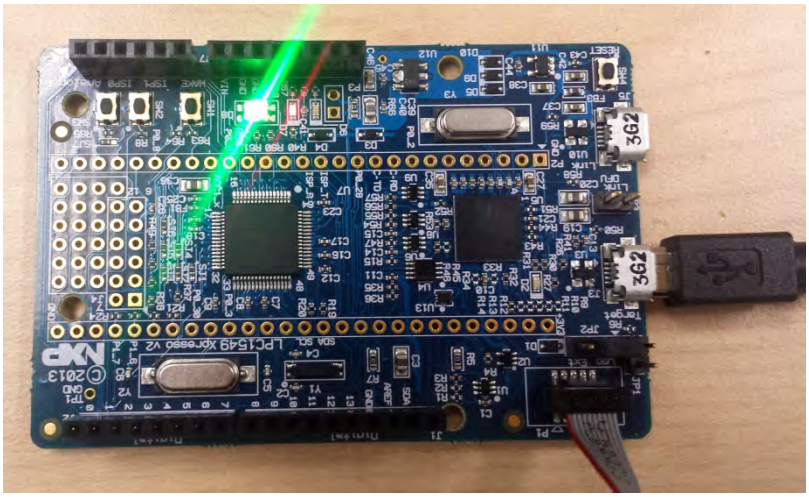

Fig 10. Green LED on Xpresso Board

6. Conclusion

This application note provides an example implementation for In-Application Programming (IAP) in LPC15xx MCU families. The IAP routines available on the LPC15xx provide an easy and simple way for data storage or for program updates. As these routines are stored on the on-chip ROM, the user application's code space used is minimized.

For additional details on how the IAP routines operate, refer to the LPC15xx user manual UM10736.

7. Legal information

7.1 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.

7.2 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.

7.3 Trademarks

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

8. List of figures

Fig 1. LPC15xx flash structure 3
Fig 2. LPC15xx flash address mapping 4
Fig 3. Keil IROM and SRAM remapping 7
Fig 4. LPCXpresso flash and SRAM remapping 7
Fig 5. IAR Embedded Workbench ROM and SRAM remapping 8
Fig 6. SYSMEMREMAP register 8
Fig 7. SysTick interrupt handler settings in Keil 9
Fig 8. Copy the IRQ handler to SRAM 10
Fig 9. LPC1549 Xpresso board and LPCLink2 debugger 10
Fig 10. Green LED on Xpresso Board 11

9. List of tables

Table 1. IAP routines 5

10. Contents

1. Introduction 3
2. Flash specifications 3
3. EEPROM 4
4. Introduction to In-Application Programming 4
4.1 IAP initialization 4
4.2 IAP routines 5
4.3 IAP precautions 6
4.3.1 Interrupts 6
4.3.2 RAM usage 6
5. IAP sample project and interrupt handling 6
5.1 Software setup 6
5.1.1 SRAM memory mapping 6
5.1.2 Interrupt remapping 8
5.1.3 SysTick interrupt 8
5.1.4 Handling interrupts during IAP 9
5.2 Hardware setup 10
5.3 Application example 11
6. Conclusion 11
7. Legal information 12
7.1 Definitions 12
7.2 Disclaimers 12
7.3 Trademarks 12
8. List of figures 13
9. List of tables 14
10. Contents 15

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

Date of release: 3 June 2014
Document identifier: AN11555