AN11511

LPC11U6x In-Application Programming

Rev. 1 — 6 February 2014 Application note

Document information

Info	Content
Keywords	LPC11U6X, IAP
Abstract	This application note describes the In-Application Programming capabilities of LPC11U6X. It also provides guidance to interrupt handling during flash IAP calls.

Revision history

Rev	Date	Description
1	20140206	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 LPC11U6x are an ARM Cortex-M0+ based, low-cost 32-bit MCU family operating at CPU frequencies of up to 50 MHz. The LPC11U6x support up to 256 kB of flash memory, a 4 kB EEPROM, and 36 kB of SRAM.

The ARM Cortex-M0+ is an easy-to-use, energy-efficient core using a two-stage pipeline and fast single-cycle I/O access. The peripheral complement of the LPC11U6x includes a DMA controller, a CRC engine, one full-speed USB device controller with XTAL-less low-speed mode, two I^2C-bus interfaces, up to five USARTs, two SSP interfaces, PWM/timer subsystem with six configurable multi-purpose timers, a Real-Time Clock, one 12-bit ADC, temperature sensor, function-configurable I/O ports, and up to 80 general-purpose I/O pins.

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-chip flash or to get some information stored in on-chip ROM. This can include:

- Field firmware upgrade
- EEPROM content replacement
- Data storage

2. Flash specifications

The on-chip flash memory of the LPC11U6X is grouped in sectors. The flash memory is divided into 24 x 4 kB and 5 x 32 kB sectors. Individual pages of 256 bytes each can be erased using the IAP erase page command.

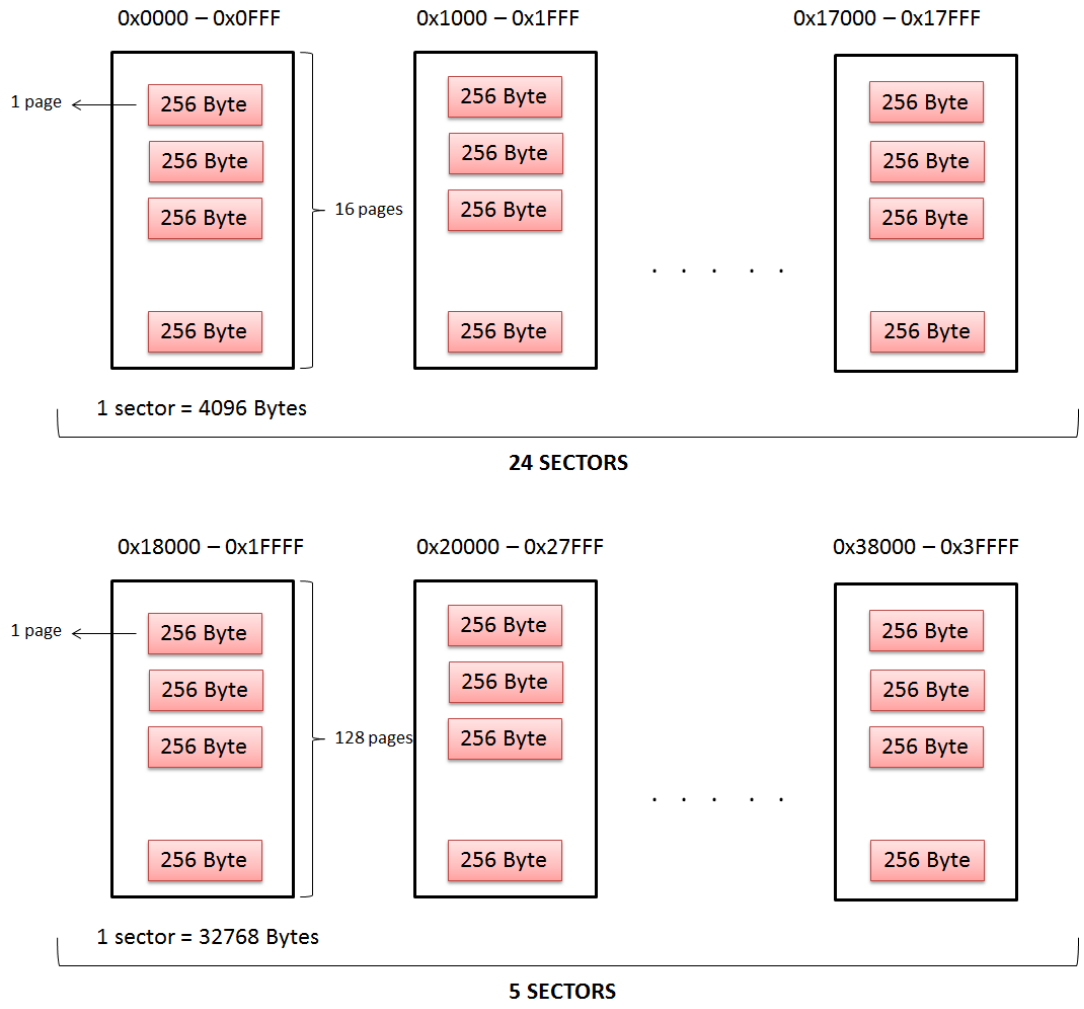

Fig 1. LPC11U6x flash structure

3. EEPROM

The LPC11U6x contains 2 kB or 4 kB of on-chip byte-erasable and byte-programmable EEPROM data memory. The EEPROM can be programmed using IAP via the on-chip boot loader software.

4. Introduction to IAP

4.1 IAP initialization

The IAP routines are located in the Boot ROM. The IAP routine resides at 0x1FFF 1FF0 location and it is thumb code. To access the IAP routines the entry point for IAP has to be defined. For LPC11U6x, the address is 0x1FFF1FF1.

```
1    #define IAP_LOCATION 0x1FFF1FF1
```

The IAP routines take two unsigned 32-bit integer array, 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:

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

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, enable UART0 and Timer1 and change their PCLK values to CCLK.

IAP command	Code (base 10)	Functional description	Precautions
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.
Write EEPROM	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.
Read EEPROM	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 operation.

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 LPC11U6x 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 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 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 0x1000 0000. Since the interrupt vector table uses 512 bytes of the bottom of SRAM, the start location is set to 0x1000 0200. The SRAM size of LPC11U6x is 32 kB. With the IAP using the 32 bytes in the top of SRAM, this

means the usable SRAM size is 32 kB – 32 bytes = 32736 bytes. But since 512 bytes is also being used by the interrupt vector table, the SRAM size now becomes:

32768 – 32 – 512 = 32224 bytes i.e. 0x7DE0.

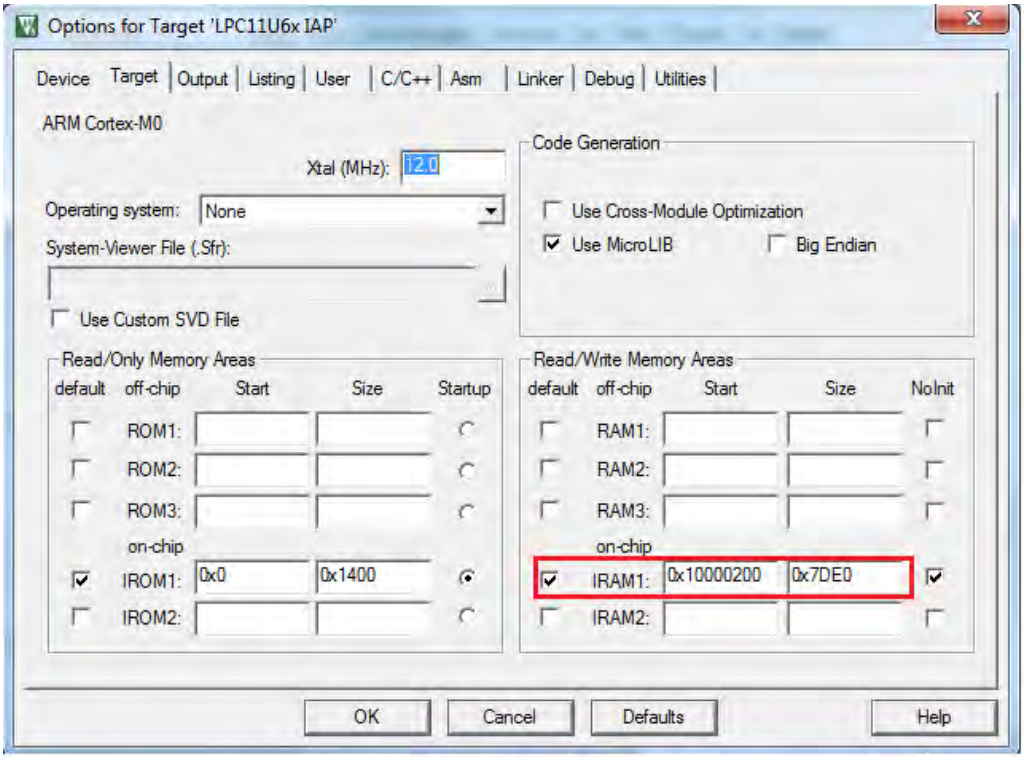

Fig 2. Keil IROM and SRAM remapping

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

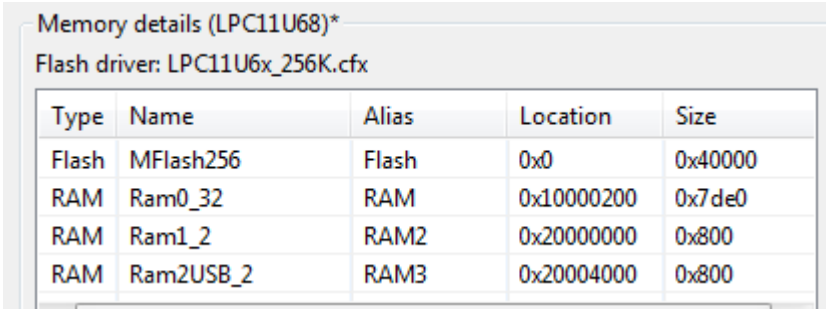

Fig 3. LPCXpresso flash and SRAM remapping

In IAR Embedded Workbench, the SRAM remapping is achieved by changing the linker configuration settings. The RAM address is set to 0x1000 0200 and the end address is set to 0x1000 7FE0.

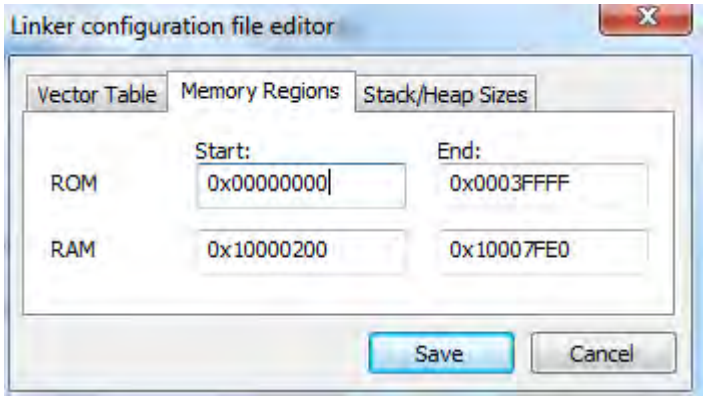

Fig 4. IAR Embedded Workbench ROM and SRAM remapping

5.1.2 Interrupt remapping

The system remap register SYSMEMREMAP on NXP's LPC11U6x 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 5. SYSMEMREMAP register

So for interrupt handling during IAP, user code should copy the interrupt vector table from 0x0000 0000 to 0x1000 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.

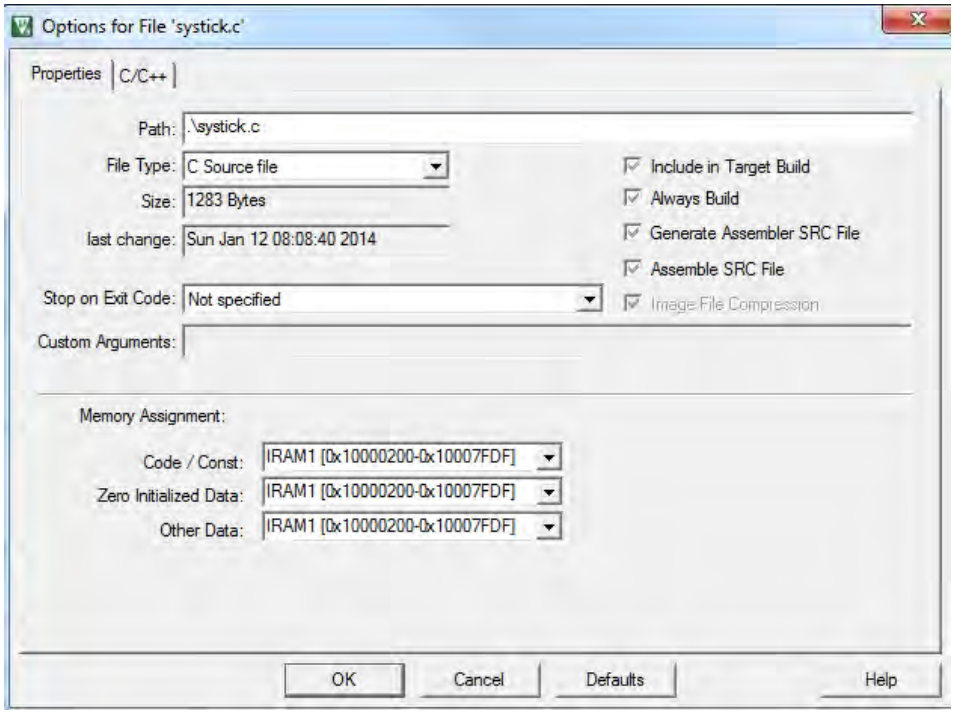

Fig 6. SysTick interrupt handler settings in Keil

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_PORT->NOT[2] = (1<<5);}
```

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_PORT->NOT[2] = (1<<5);}
```

5.1.4 Handling interrupts during IAP

The LPC11U6x flash is not accessible when the IAP routines are being called. This can be dealt with by relocating the interrupt table to SRAM.

The interrupt vector table is moved to the 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 0x1000 0000.

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

    ramPtr = (unsigned int *)0x10000000;  //load RAM starting at 0x10000000,
    flashPtr = (unsigned int *)0x00;      //start of interrupt vector table
    while(flashPtr < uLimit)
      {
        *ramPtr = *flashPtr;
        ramPtr++;
        flashPtr++;
      }
}
```

Fig 7. Copy the IRQ handler to SRAM

During EEPROM write and read operations, the interrupts must be disabled for proper IAP operation.

5.2 Hardware setup

The input parameter 'CCLK' of an IAP command should be equal to the CPU clock frequency in kilohertz (kHz). If the CCLK parameter is less than the CPU clock, the flash operation may be unstable. If the CCLK parameter is higher than the CPU clock, the flash operation may be slower than expected. If the CCLK is not equal to the CPU clock, the flash reliability cannot be guaranteed.

An LPC11U68 Manley board with UART and Xpresso board is used to implement the demonstration code.

Fig 8. LPC11U68 Manley board

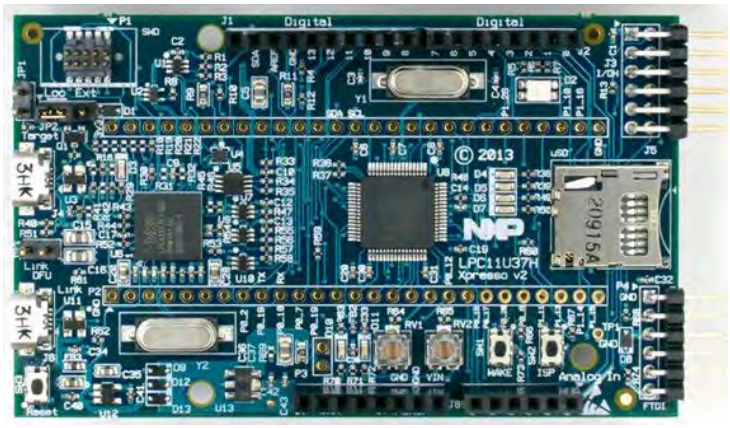

Fig 9. LPC11U68 Xpresso board

Fig 10. RS232 Serial Interface cable

Connect the Serial cable to the LPC11U68 Manley Board and the PC and run the code. Follow the instructions on the console. The LPC11U68 Xpresso board does not support UART connection.

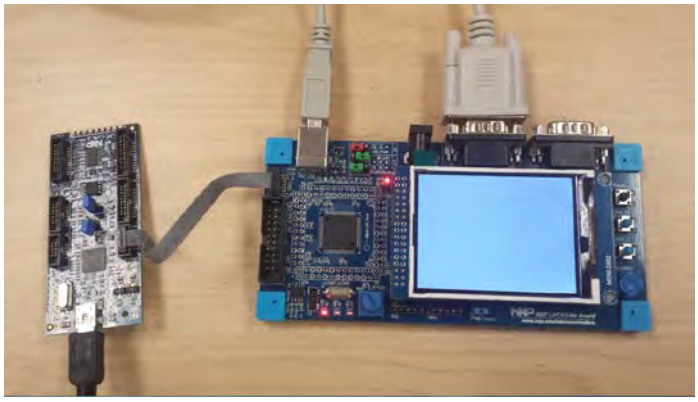

Fig 11. UART and debugger connection on LPC11U68 Manley Board

For the Keil IDE use the ULINK debugger, for LPCXpresso IDE use the LPC Link-2 and for IAR Embedded Workbench IDE the Jlink can be used.

Fig 12. ULINK debugger

Fig 13. LPC Link-2

Fig 14. Segger J-link

5.3 Application example

The demo software sends a menu system to the PC through the UART at 9600 baudrate.

The demo software demonstrates the IAP calls of:

- EEPROM write
- EEPROM read
- Reading the part ID of the device
- Reading the boot code version
- Erasing the sector
- Copying contents from RAM to flash
- Verifying the data in the memory locations
- Erase page

The following figure shows the UART menu display using Tera Term on the PC.

Fig 15. UART window on PC

In the IAR package for Manley board, the UART is not implemented. An LED glows when the IAP is a success.

For the LPC11U68 Xpresso Board, the LED on the board indicates the status of the IAP. The LED turns green if IAP succeeds, and turns red if IAP fails.

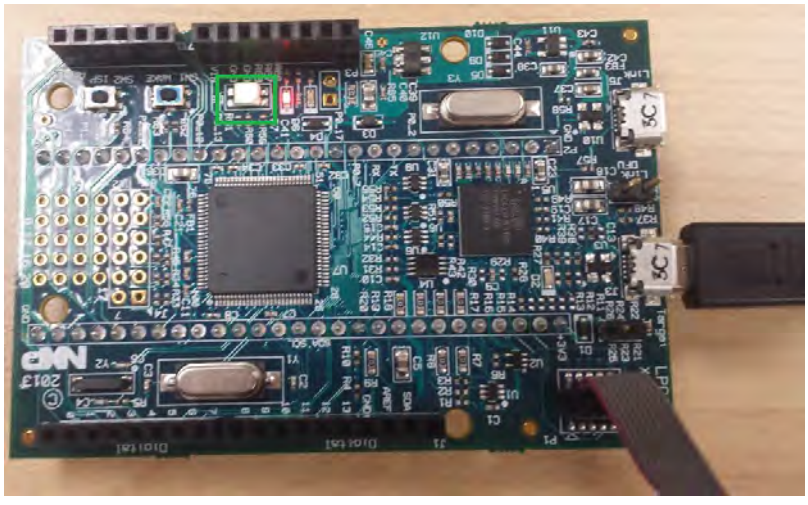

Fig 16. Status LED on XPresso board

6. Conclusion

This application note provides example implementation for IAP in LPC11U6x MCU families. The IAP routines available on the LPC11U6x 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 size is minimized.

For additional details on how the IAP routines operate, refer to the LPC11U6x user manual.

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

9. List of tables

10. Contents

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

Date of release: 6 February 2014
Document identifier: AN11511