

# TN00037

## LPC54018 Code Partition between SRAM and SPIFI memory

1.0 — 14 March 2018

Technical note

### Document information

Info	Content
<b>Keywords</b>	LPC540xx, SRAM, SPIFI, XIP, linker scripts
<b>Abstract</b>	This technical note describes how to partition code between SRAM and SPIFI memory regions using linker scripts.



**Revision history**

Rev	Date	Description
1.0	20180314	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](mailto:salesaddresses@nxp.com)

## 1. Introduction

The LPC540xx is a family of ARM Cortex-M4 based microcontrollers used in embedded applications. This technical note references the LPCXpresso development board for LPC540xx MCUs. For details of the board, see:

<https://www.nxp.com/demoboard/om40003>

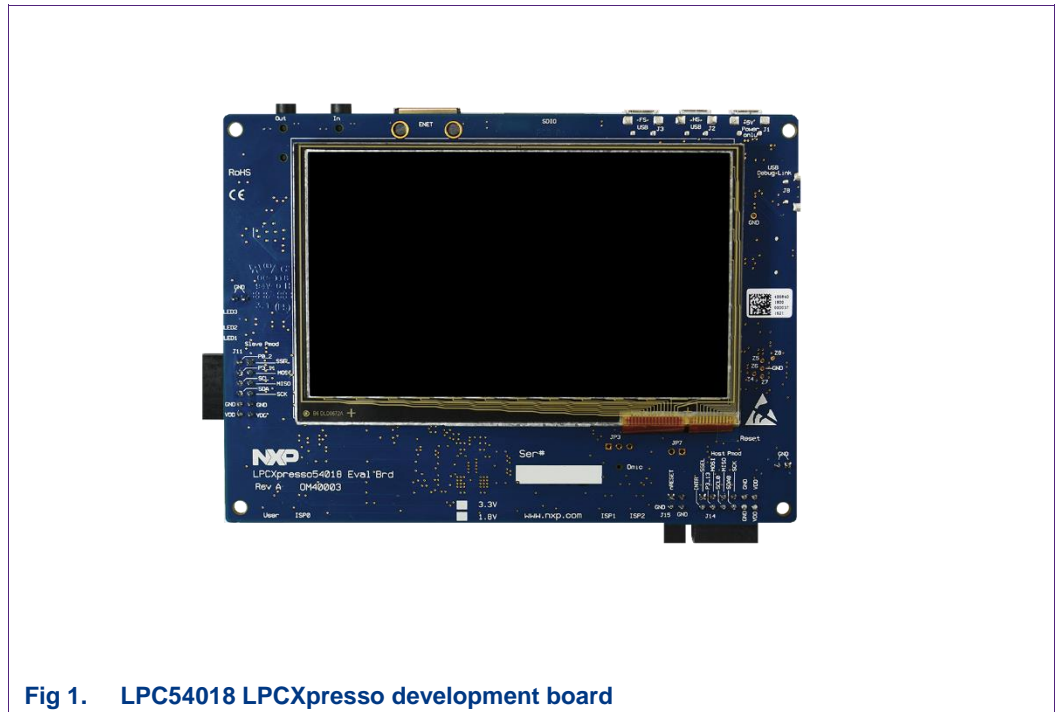


Fig 1. LPC54018 LPCXpresso development board

## 2. Description

The "blinky\_sram\_xip\_hybrid" example illustrates how to split code between SRAMX and SPIFI memory regions. The example is a simple blinky application toggling LED3 on the LPC54018 LPCXpresso development board. The delay() function in "delay\_nop.c" file is placed in SPIFI memory location 0x10040000 and is executed-in-place (XIP). The rest of the code is linked to SRAMX memory region and is executed in plain load fashion.

**Note:** See the "Getting Started with MCUXpresso SDK for LPC540xx.pdf" document in "docs" folder of SDK package. This document explains how to configure the IDEs for various debug configurations (SRAMX, Plain Load, and XIP).

The example is available in two tool chains:

- IAR embedded Workbench
- Keil MDK

The Keil and IAR examples are found in:

**lpc54018\_blinky\_sram\_xip\_hybrid\_keil\_iar\boards\lpcpresso54018\demo\_apps\blinky\_sram\_xip\_hybrid**

## 2.1 Keil MDK IDE

The Keil linker script modification is shown below. Refer to “LPC54018\_sram\_spifi\_flash\_hybrid.scf” file in the example. An XIP region is defined and the “delay\_nop.o” object file is placed in this defined region.

```
#define m_usb_sram_start          0x40100000
#define m_usb_sram_size          0x00002000

#if (!defined(XIP_IMAGE))
#define m_xip_start              0x10040000
#define m_xip_size              0x00100000
#endif
```

**Fig 2. Keil linker script – definition memory region**

```
#if (!defined(XIP_IMAGE))
LR_m_xip m_xip_start m_xip_size {
    ER_m_xip m_xip_start m_xip_size {
        delay_nop.o (+RO)
    }
}
#endif
```

**Fig 3. Keil linker script – placement of delay\_nop.o**

To program the plain load example with the XIP code region, add the Quad SPI flash flashloaders. See [Fig 4](#). One flashloader has address range from 0x00000000 – 0x00FFFFFF to program Plain load image and another flashloader has the address range 0x10000000 – 0x10FFFFFF to program the XIP code section.

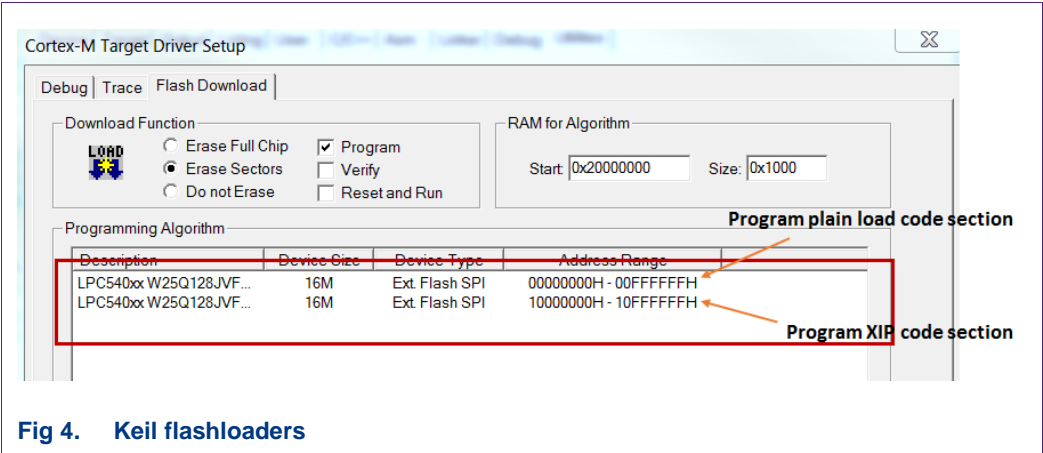


Fig 4. Keil flashloaders

In Keil MDK, to program the application into external Quad SPI flash device, set the Reset type in Keil IDE to SYSRESETREQ (See Project->Options->Debug tab ->Settings). To debug the application set the Reset type to VECTRESET.

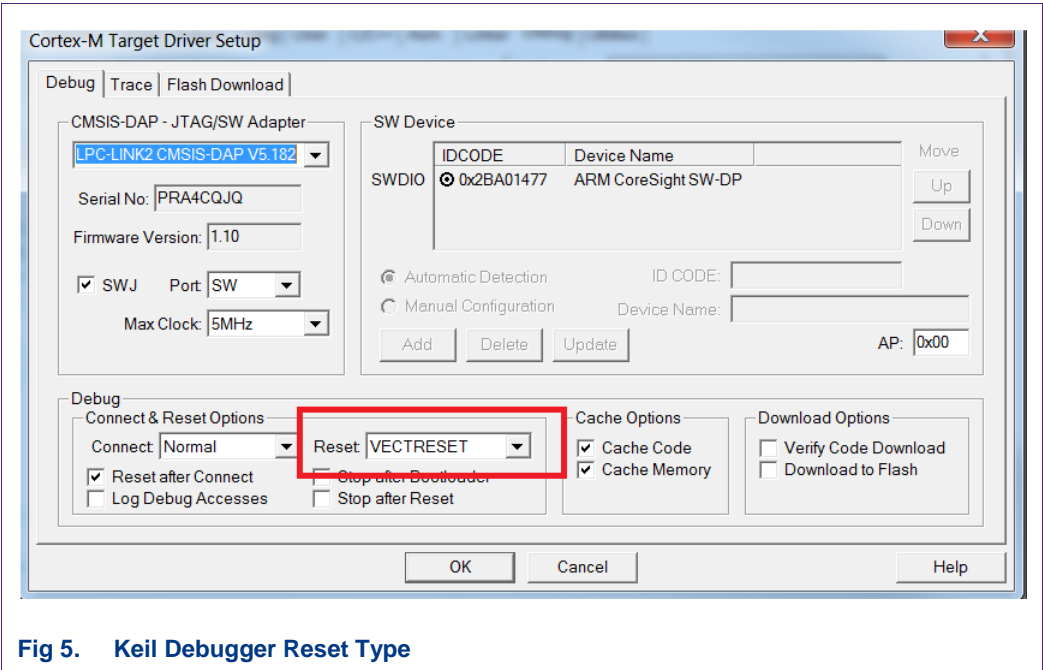


Fig 5. Keil Debugger Reset Type

## 2.2 IAR Embedded Workbench IDE

The IAR linker script modification is shown below. See the “LPC54018\_sram\_spifi\_flash\_hybrid.icf” file in the example.

```
define symbol m_usb_sram_start          = 0x40100000;
define symbol m_usb_sram_end            = 0x40101FFF;

if (!isdefinedsymbol(XIP_IMAGE)) {
define symbol m_xip_start                = 0x10040000;
define symbol m_xip_end                  = 0x100FFFFF;
}

define memory mem with size = 4G;

define region TEXT_region                = mem:[from m_interrupts_start to m_interrupts_end]
| mem:[from m_text_start to m_text_end];
define region DATA_region              = mem:[from m_data_start to m_stack_start - 1];
define region CSTACK_region             = mem:[from m_stack_start to m_stack_end];

if (!isdefinedsymbol(XIP_IMAGE)) {
define region XIP_region                 = mem:[from m_xip_start to m_xip_end];
}
```

**Fig 6. IAR linker script – definition of memory region**

```
if (!isdefinedsymbol(XIP_IMAGE)) {
place in XIP_region                { readonly object delay_nop.o };
}
```

**Fig 7. IAR linker script - placement of delay\_nop.o**

In IAR IDE, no changes to the flashloaders are required because the IAR in-built flashloaders take care of programming both the plain load and XIP sections.

The MCUXpresso IDE version of example will be released with the next release of MCUXpresso IDE.

## 3. Legal information

### 3.1 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.

### 3.2 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.

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 accepts 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.

### 3.3 Trademarks

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

4. Contents

1. Introduction .....3

2. Description.....3

2.1 Keil MDK IDE .....4

2.2 IAR Embedded Workbench IDE.....6

3. Legal information .....7

3.1 Definitions .....7

3.2 Disclaimers.....7

3.3 Trademarks.....7

4. Contents.....8

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