AN13628

SRAM3 Usage in LPC55(s)06

Rev. 0 — 8 June 2022 Application Note

Contents

1 Introduction

LPC55(s)06 has 96 kB of on-chip SRAM, including SRAMX, SRAM0,1,2,3. SRAM3 is reserved by default, user cannot use it as stack space in SDK code. The application note introduces a method to use SRAM3 as stack space in the code.

2 SRAM allocation in LPC55(s)06

This chapter introduces the memory map for LPC55(s)06 and also gives the address range for SRAM3. Table 1 is the memory map overview, as described in *LPC55S0x/LPC550x User manual* (document UM11424).

Table 1. Memory map overview

AHB port	Non-secure start address	Non-secure end address	Secure start address	Secure end address	Function
0	0x0000 0000	0x0003 FFFF	0x1000 0000	0x1003 FFFF	Flash memory, on CM33 code bus. The last 17 pages (12 kB) are reserved on the 256 kB flash devices resulting in 244 kB internal flash memory
	0x0300 0000	0x0301 FFFF	0x1300 0000	0x1301 FFFF	Boot ROM, on CM33 code bus
1	0x0400 0000	0x0400 3FFF	0x1400 0000	0x1400 3FFF	SRAMX on CM33 code bus, 32 kB. SRAMX_0 (0x1400 0000 to 0x1400 0FFF) and SRAMX_1 (0x1400 1000 to 0x1400 1FFF) are used for Casper (total 8 kB). If CPU retention used in power-down mode, SRAMX_2 (0x1400 2000 to 0x1400 25FF) is used (total 1.5 kB) by default in power API. This function is user configurable within SRAMX_2 and SRAMX_3
2	0x2000 0000	0x2000 7FFF	0x3000 0000	0x3000 7FFF	SRAM0 on CM33 data bus, 32 kB
3	0x2000 8000	0x2000 BFFF	0x3000 8000	0x3000 BFFF	SRAM1 on CM33 data bus, 16 kB
4	0x2000 C000	0x2000 FFFF	0x3000 C000	0x3000 FFFF	SRAM2 on CM33 data bus, 16 kB
5	**0x2001 0000**	**0x2001 3FFF**	**0x3001 0000**	**0x3001 3FFF**	**SRAM 3, 16 kB**
6	0x4000 0000	0x4001 FFFF	0x5000 0000	0x5001 FFFF	AHB to APB bridge 0
	0x4002 0000	0x4003 FFFF	0x5002 0000	0x5003 FFFF	AHB to APB bridge 1
7	0x4008 0000	0x4008 FFFF	0x5008 0000	0x5008 FFFF	AHB peripherals

Table continues on the next page...

Table 1. Memory map overview (continued)

AHB port	Non-secure start address	Non-secure end address	Secure start address	Secure end address	Function
8	0x4009 0000	0x4009 FFFF	0x5009 0000	0x5009 FFFF	AHB peripherals
9	0x400A 0000	0x400A FFFF	0x500A 0000	0x500A FFFF	AHB peripherals

NOTE

The address of SRAM3 is next to the address of other SRAMs.

3 SRAM3 usage limitations

This section lists the limitations on SRAM3 usage as follows:

- Do not use SRAM3 as stack storage, because the ROM code provides the verification for value of SP in the VTOR table.
- If the value of SP register is greater than 0x20010000, the ROM code does not allow PC jump into the app code.
- Enable SRAM3 clock before use.

The ROM code verifies the value of the stack pointer before the PC pointer jumps into the application code. If the value of SP is above 0x20010000, the ROM code does not jump into application code and the PC pointer enters the while (1) loop in ROM code. However, the SRAM3 space address range is above 0x20010000. Therefore, if user defines the stack pointer value in the SRAM3, the PC cannot jump into the application code. SRAM3 is at the end of SRAM storage in the chip, user often allocates the stack at the end of SRAM. This is the limitation of SRAM3 usage in the chip.

The document introduces two methods to use the SRAM3 for stack storage as described in Workarounds.

4 Workarounds

The two workarounds to allocate the stack pointer in the SRAM3 are described in this section.

4.1 Workaround to allocate the STACK space at SRAM0/1/2/X

This method is easy, however, to protect the overflow, the user may allocate the stack at the end of the RAM. If stack is allocated at SRAM0/1/2, the whole SRAM is divided into three parts as shown in Figure 1:

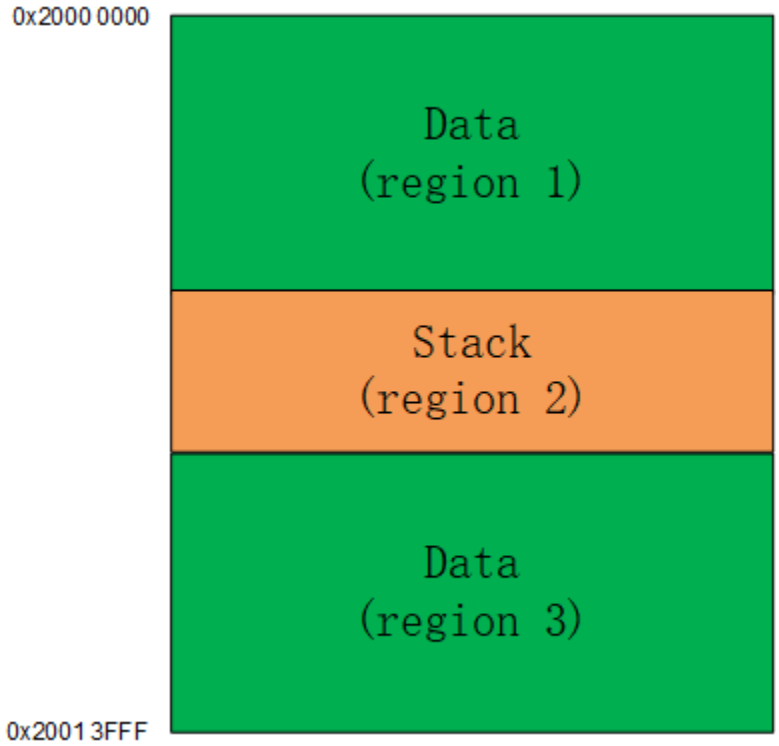

Figure 1. SRAM allocation map

NOTE

The data region 3 in Figure 1 is difficult to use in order to store the data. Assign the data address at region 3 in the project.

4.2 Workaround to cheat the ROM to verify the SP value successfully

Set an acceptable value first word of vector table, to **cheat** the ROM to verify the SP value successfully. Then give a correct value to SP register before using stack.

NOTE

The SP value in VTOR table is 0x20010000 which is fixed and is a wrong value. The correct value is Image$$ARM_LIB_STACK$$ZI$$Limit in Keil IDE and _vStackTop in MCUXpresso IDE.

Therefore, take care when using the SP value in VTOR table in application code.

4.2.1 Keil IDE

The section introduces how to modify the project code to implement the function. The steps are described below:

Step 1: Modify the scatter file in IDE as shown in Figure 2.

```
43 #define  m_interrupts_start              0x00000000
44 #define  m_interrupts_size               0x00000200
45
46 #define m_text_start                     0x00000200
47 #define m_text_size                      0x00037E00
48
49 #define  m_data_start                    0x20000000
50 #define  m_data_size                     0x00014000
51
52 #define  m_sramx_start                   0x04000000
53 #define  m_sramx_size                    0x00004000
54
55 //#define  m_sram3_start                   0x20010000
56 //#define  m_sram3_size                    0x00004000
```

Figure 2. Modifying the RAM range

Step 2: Set the first word of vector table with acceptable value as shown in Figure 3.

```
26 __Vectors:
27     .long    0x20010000                    /* Top of Stack */
28     .long    Reset_Handler                 /* Reset Handler */
29     .long    NMI_Handler                   /* NMI Handler*/
30     .long    HardFault_Handler             /* Hard Fault Handler*/
31     .long    MemManage_Handler             /* MPU Fault Handler*/
```

Figure 3. Setting the first word of vector table in MDK IDE

Step3: Enable SRAM clock and reconfig the SP register with correct value as shown in Figure 4 and Figure 5.

```
117 Reset_Handler:
118     cpsid   i               /* Mask interrupts */
119     ldr     r0, =0x50000228
120     ldr     r1, =0x40
121     str     r1, [r0]
122     .equ    VTOR, 0xE000ED08
123     ldr     r0, =VTOR
124     ldr     r1, =__Vectors
125     str     r1, [r0]
126     ldr     r2, =Image$$ARM_LIB_STACK$$ZI$$Limit
127     msr     msp, r2
128     ldr     R0, =Image$$ARM_LIB_STACK$$ZI$$Base
129     msr     msplim, R0
130     ldr     r0,=SystemInit
131     blx     r0
132     cpsie   i               /* Unmask interrupts */
133     ldr     r0,=__main
134     bx      r0
```

Figure 4. Enable SRAM clock and set SP register in MDK IDE

```
Image$$ARM_LIB_HEAP$$ZI$$Base            0x20000060   Number         0  anon$$obj.o ABSOLUTE
Image$$ARM_LIB_HEAP$$ZI$$Limit           0x20000460   Number         0  anon$$obj.o ABSOLUTE
Image$$ARM_LIB_STACK$$ZI$$Base           0x20013c00   Number         0  anon$$obj.o ABSOLUTE
Image$$ARM_LIB_STACK$$ZI$$Limit          0x20014000   Number         0  anon$$obj.o ABSOLUTE
```

Figure 5. Define stack address in MDK IDE

Code Snippet:

```
Reset_Handler:
cpsid   i
```

```
ldr     r0, =0x50000228
ldr     r1, =0x40
str     r1, [r0]
.equ    VTOR, 0xE000ED08
ldr     r0, =VTOR
ldr     r1, =__Vectors
str     r1, [r0]
ldr     r2, =Image$$ARM_LIB_STACK$$ZI$$Limit
msr    msp, r2
ldr     R0, =Image$$ARM_LIB_STACK$$ZI$$Base
msr     msplim, R0
ldr     r0,=SystemInit
blx     r0
cpsie   i
ldr     r0,=__main
bx      r0
```

4.2.2 MCUXpresso IDE

The section introduces how to modify the project code to implement the function in MCUXpresso IDE. The steps are described below:

Step 1: Modify memory details as shown in Figure 6.

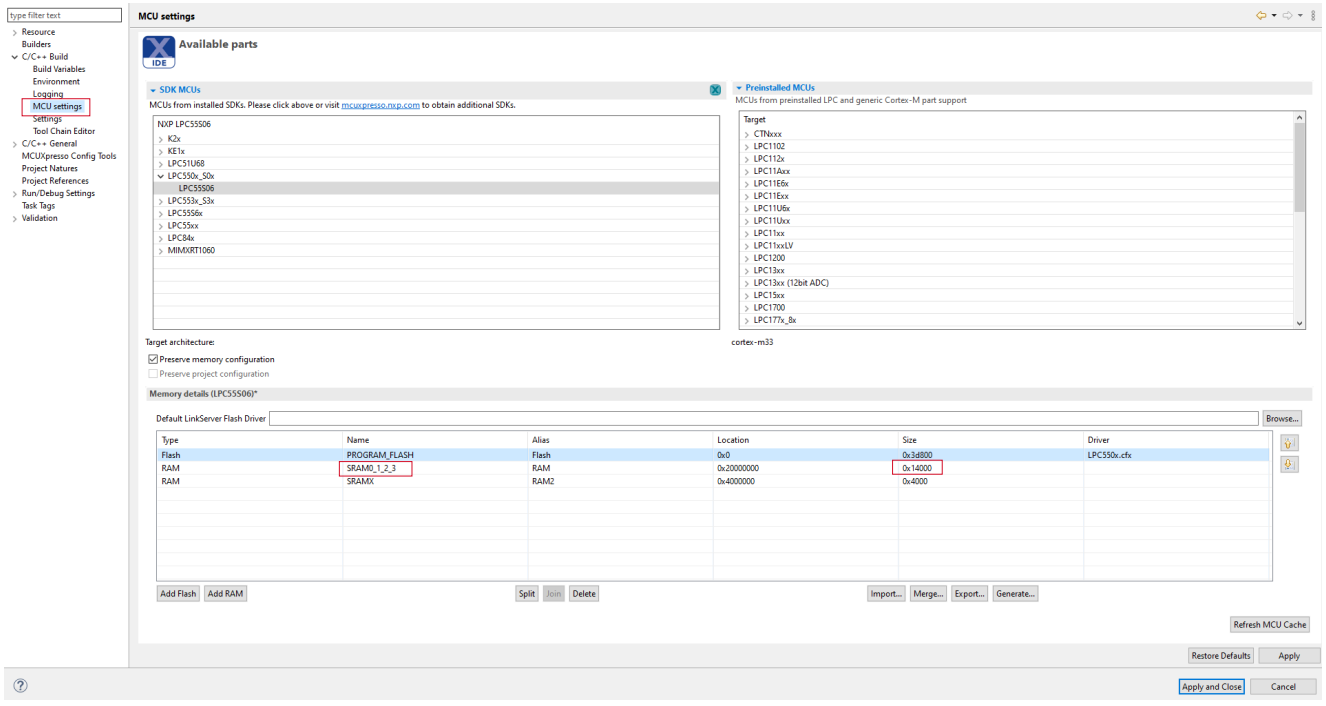

Figure 6. SRAM space setting in MCUXpresso IDE

Step 2: Set the first word of vector table with acceptable value as shown in Figure 7.

```
__attribute__ ((used, section(".isr_vector")))
void (* const g_pfnVectors[])(void) = {
    // Core Level - CM33
    (void (*)(void)) 0x20010000,       // set stack address with acceptable value for ROM
    ResetISR,                          // The reset handler
    NMI_Handler,                       // The NMI handler
    HardFault_Handler,                 // The hard fault handler
    MemManage_Handler,                 // The MPU fault handler
```

Figure 7. Set the first word of vector table in MCUXpresso IDE

Step3: Enable SRAM clock and reconfig the SP register with correct value as shown in Figure 8.

```
__attribute__ ((naked, section(".after_vectors.reset")))
void ResetISR(void) {


    // Disable interrupts
    __asm volatile ("cpsid i");

    // Config VTOR & MSPLIM register
    __asm volatile (
                    "LDR R0, =0xE000ED08  \n"
                    "STR %0, [R0]         \n"
                    "LDR R0, =0x50000228  \n"//Enable SRAM3 clock
                    "LDR R1, =0x40        \n"
                    "STR R1, [R0]         \n"
                    "MSR MSP, %1          \n"//configure SP register with actual value of stack address
                    "MSR MSPLIM, %2       \n"
                    :
                    : "r"(g_pfnVectors), "r"(_vStackTop), "r"(_vStackBase)
                    : "r0", "r1", "r2"
                    );
#if defined (__USE_CMSIS)
// If __USE_CMSIS defined, then call CMSIS SystemInit code
    SystemInit();

#endif // (__USE_CMSIS)
```

Figure 8. Enable SRAM3 and set SP register in MCUXpresso IDE

Code snippet:

```
// Config VTOR & MSPLIM register
__asm volatile
"LDR R0, =0xE000ED08  \n"
"STR %0, [R0]         \n"
"LDR R0, =0x50000228  \n"//Enable SRAM3 clock
"LDR R1, =0x40        \n"
"STR R1, [R0]       \n"
"MSR MSP, %1        \n"//configure SP register with actual value of stack address
"MSR MSPLIM, %2     \n"
        :
        : "r"(g_pfnVectors), "r"(_vStackTop), "r"(_vStackBase)
        : "r0", "r1", "r2");
```

5 Revision history

Table 2 summarizes the changes done to this document since the initial release.

Rev	Date	Substantive Changes
0	8 June 2022	Initial release

Legal information

Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

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.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

Airfast — is a trademark of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Cadence — the Cadence logo, and the other Cadence marks found at www.cadence.com/go/trademarks are trademarks or registered trademarks of Cadence Design Systems, Inc. All rights reserved worldwide.

CodeWarrior — is a trademark of NXP B.V.

ColdFire — is a trademark of NXP B.V.

ColdFire+ — is a trademark of NXP B.V.

EdgeLock — is a trademark of NXP B.V.

EdgeScale — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

eIQ — is a trademark of NXP B.V.

FeliCa — is a trademark of Sony Corporation.

Freescale — is a trademark of NXP B.V.

HITAG — is a trademark of NXP B.V.

ICODE and I-CODE — are trademarks of NXP B.V.

Immersiv3D — is a trademark of NXP B.V.

I2C-bus — logo is a trademark of NXP B.V.

Kinetis — is a trademark of NXP B.V.

Layerscape — is a trademark of NXP B.V.

Mantis — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MOBILEGT — is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

Processor Expert — is a trademark of NXP B.V.

QorIQ — is a trademark of NXP B.V.

SafeAssure — is a trademark of NXP B.V.

SafeAssure — logo is a trademark of NXP B.V.

StarCore — is a trademark of NXP B.V.

Synopsys — Portions Copyright © 2021 Synopsys, Inc. Used with permission. All rights reserved.

Tower — is a trademark of NXP B.V.

UCODE — is a trademark of NXP B.V.

VortiQa — is a trademark of NXP B.V.

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

Date of release: 8 June 2022
Document identifier: AN13628