AN0026
Application Note

Extending SRAM in User's Program

Introduction

This application note describes how to extend SRAM in the user program.

Applicable products:

MCU	AT32F403xx
	AT32F403Axx
	AT32F407xx
	AT32F413xx
	AT32F435xx
	AT32F437xx

Contents

List of tables

List of figures

1 Overview

The SRAM embedded in some AT32 MCU series offers a special SRAM extension mode for users to modify the SRAM size by programming the EOPB0 in the user system data area.

In general, it is recommended to use Artery ICP or ISP tool to configure this extension mode. When the ICP or ISP is not supported, this is done by executing program, which is detailed in this application note.

2 Configuration example

In this document, the SRAM extension mode is configured by modifying the EOPB0 value. Considering that the SRAM programmable size varies from MCU series to MCU series, each AT32 MCU series is provided with a demo for user's reference. For further details, please refer to the appropriate reference manual or data sheet.

Here we take the AT32F403A as an example and use AT-START-F403A evaluation board. The deom is placed in \project\at_start_f403a\examples\sram\extend_sram.

2.1 Example case

2.1.1 Function description

The factory-default SRAM size of the AT32F403A is 96 KB, which can be extended to 224 KB by modifying the EOPB0. The EOPB0 setting value is as follows:

Table 1. AT32F403A EOPB0 setting value

EOPB0: Extended option bytes	
Bit 7:0	224 KB_MODE 0xFE : on-chip memory is 224 K 0xFF : on-chip memory is 96 K Others reserved.

The EOPB0 can be modified through the extend_sram() function.

With the EXTEND_SRMA marco definition, the SRAM can be extended from 96 KB (default value) to 224 KB SRAM or from 224 KB bacl to 96 KB. The definition value of the EXTEND_SRAM is configured in the project configuration option. Note that the use of global variables is forbidden in this function.

After modifying EOPB0, the system reset must be performed so that the new EOPB0 value takes effect.

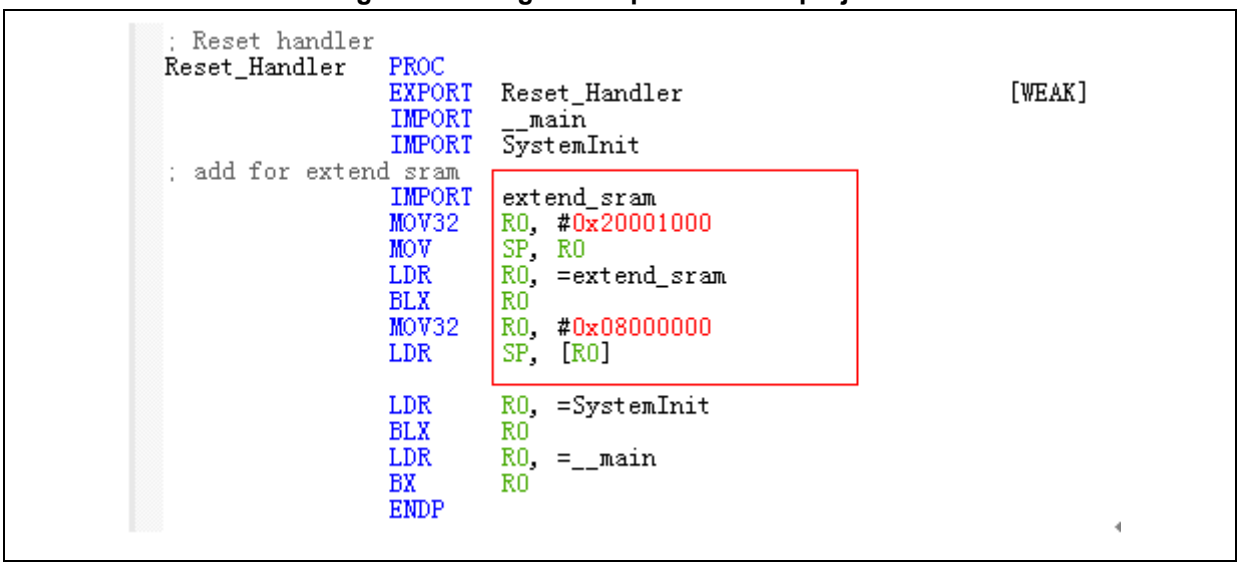
Figure 1. extend_SRAM function

```
void extend_sram(void)
{
  /* check if ram has been set to expectant size, if not, change eopb0 */
  if(((USD->eopb0) & 0xFF) != EXTEND_SRAM)
  {
    flash_unlock();
    /* erase user system data bytes */
    flash_user_system_data_erase();

    /* change sram size */
    flash_user_system_data_program((uint32_t)&USD->eopb0, EXTEND_SRAM);

    /* system reset */
    nvic_system_reset();
  }
}
```

Then the user must change the startup assembly code of the startup_at32f403a_407.s so that the extend_sram function can be executed before program initialization.

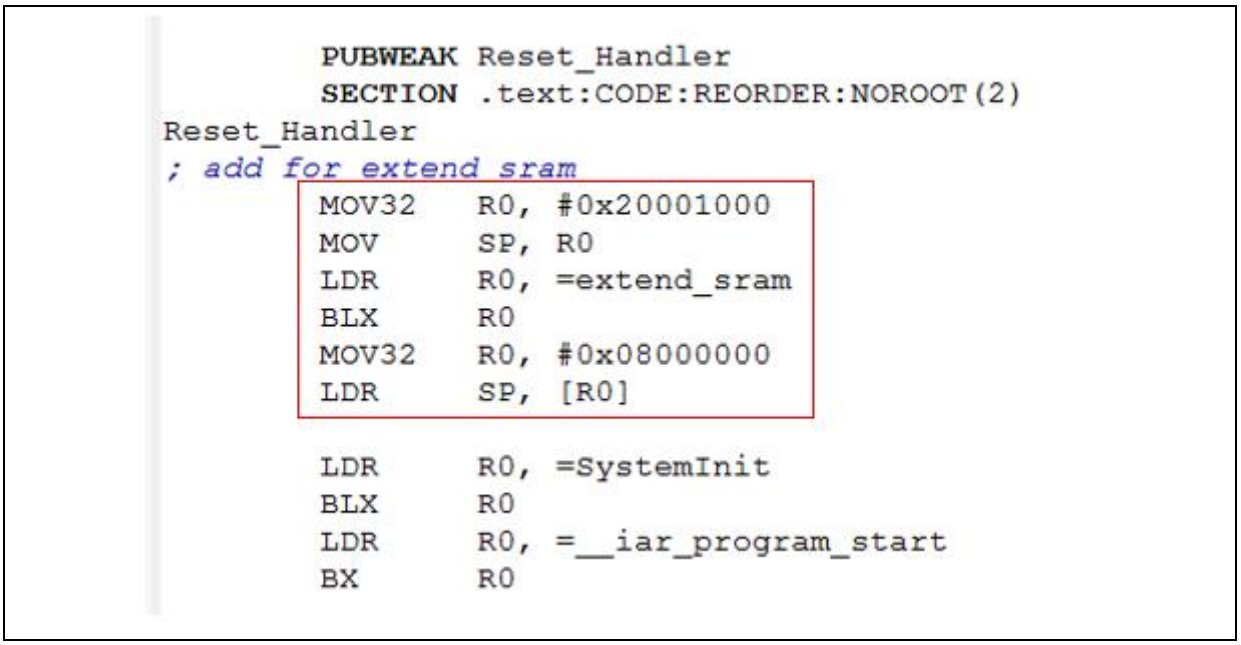
In this example, startup_at32f403a_407_ext_ram.s is the modified startup file. The figures below demonstrates how to change startup code in KEIL and IAR project.

Figure 2. Change startup file in KEIL project



```
; Reset handler
Reset_Handler   PROC
                EXPORT  Reset_Handler                   [WEAK]
                IMPORT  __main
                IMPORT  SystemInit
; add for extend sram
                IMPORT  extend_sram
                MOV32   R0, #0x20001000
                MOV     SP, R0
                LDR     R0, =extend_sram
                BLX     R0
                MOV32   R0, #0x08000000
                LDR     SP, [R0]

                LDR     R0, =SystemInit
                BLX     R0
                LDR     R0, =__main
                BX      R0
                ENDP
```

Figure 3. Change startup file in IAR project



```
        PUBWEAK Reset_Handler
        SECTION .text:CODE:REORDER:NOROOT(2)
Reset_Handler
; add for extend sram
        MOV32   R0, #0x20001000
        MOV     SP, R0
        LDR     R0, =extend_sram
        BLX     R0
        MOV32   R0, #0x08000000
        LDR     SP, [R0]

        LDR     R0, =SystemInit
        BLX     R0
        LDR     R0, =__iar_program_start
        BX      R0
```

Attention should be paid to the two points:

1) The EOPB0 must be changed at the beginning of Reset_Handler, instead of being modified in the SystemInit( ) function. Because the SRAM size set by the user in the Keil/IAR environment may be the extended 224 KB, and the actually used SRAM may exceed the default 96 KB, in this case, the initial value of the stack point will be set to the address after 96 KB, and error will occur when executing SystemInit( ), or even enter HardFault to trigger a crash.

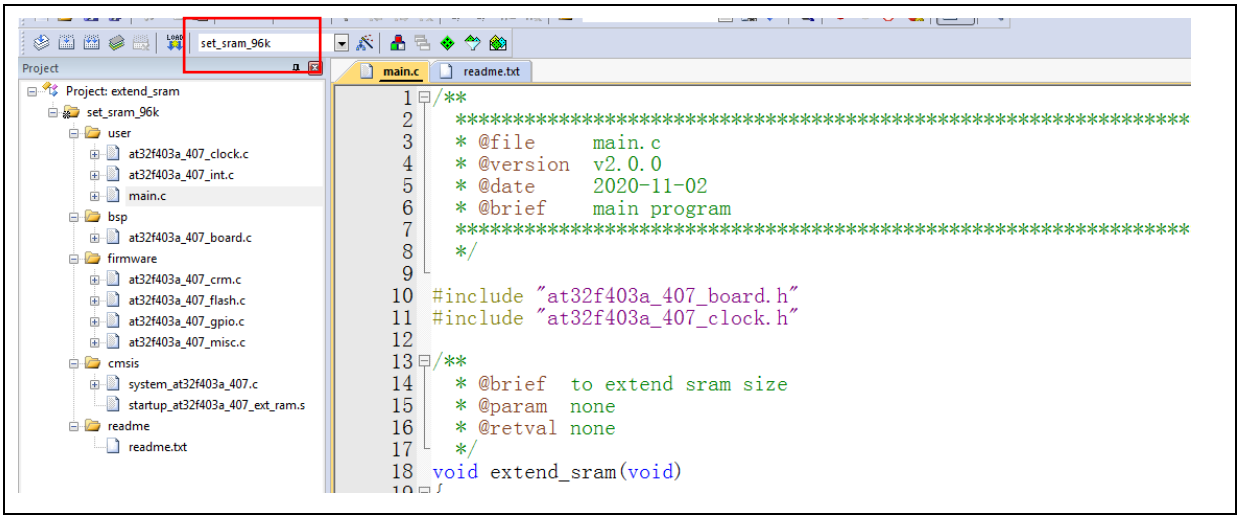
2) Before calling the extend_SRAM( ) function, the stack pointer must be placed within 96 KB (changed to 0x20001000 in this example) to prevent the initial value of the stack pointer from being set to the address outside 96 KB and cause an error during the execution of extend_SRAM( ).

2.1.2 Example demonstration

2.1.2.1 KEIL project

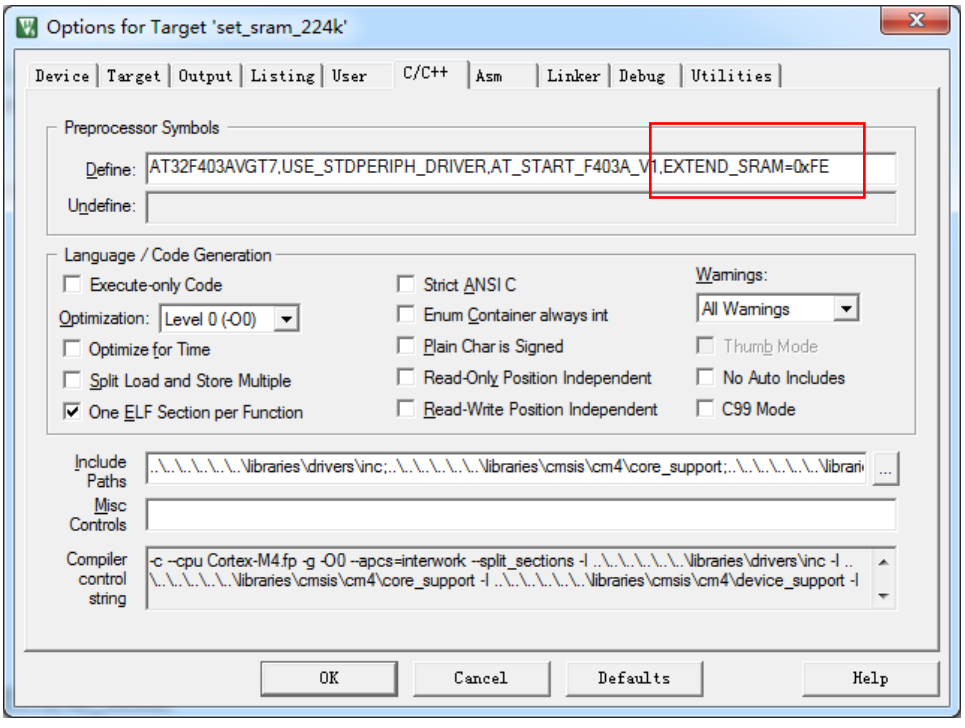
In this project, the user can select **set_SRAM_96 K or set_SRAM_224 K** through "*Select Target*" option, as shown in the figure below.

Figure 4. SRAM size selection



When the "**set_SRAM_96K**" or "**set_SRAM_224K**" is selected, the macro definition value of the EXTEND_SRAM will be displayed in the definition column of *C/C++ →Preprocessor Symbols*, and then the **extend_SRAM( )** function will select the corresponding configuration during compiling.

Figure 5. C/C++/Preprocessor Symbols configuration

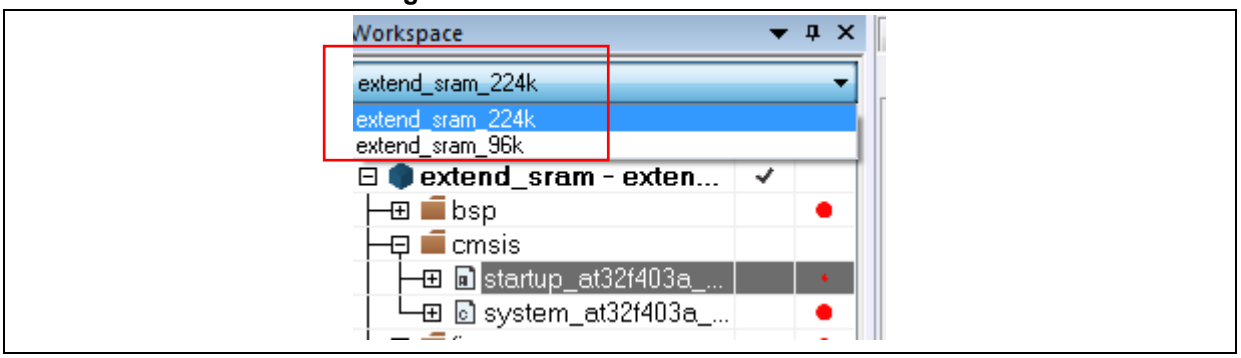


When the extension operation is complete, it will enter the **main( )** function to check the EOPB0 value in order to verify if the SRAM size is set correctly or not. The verify result is indicated by turning on LED4.

2.1.2.2 IAR project

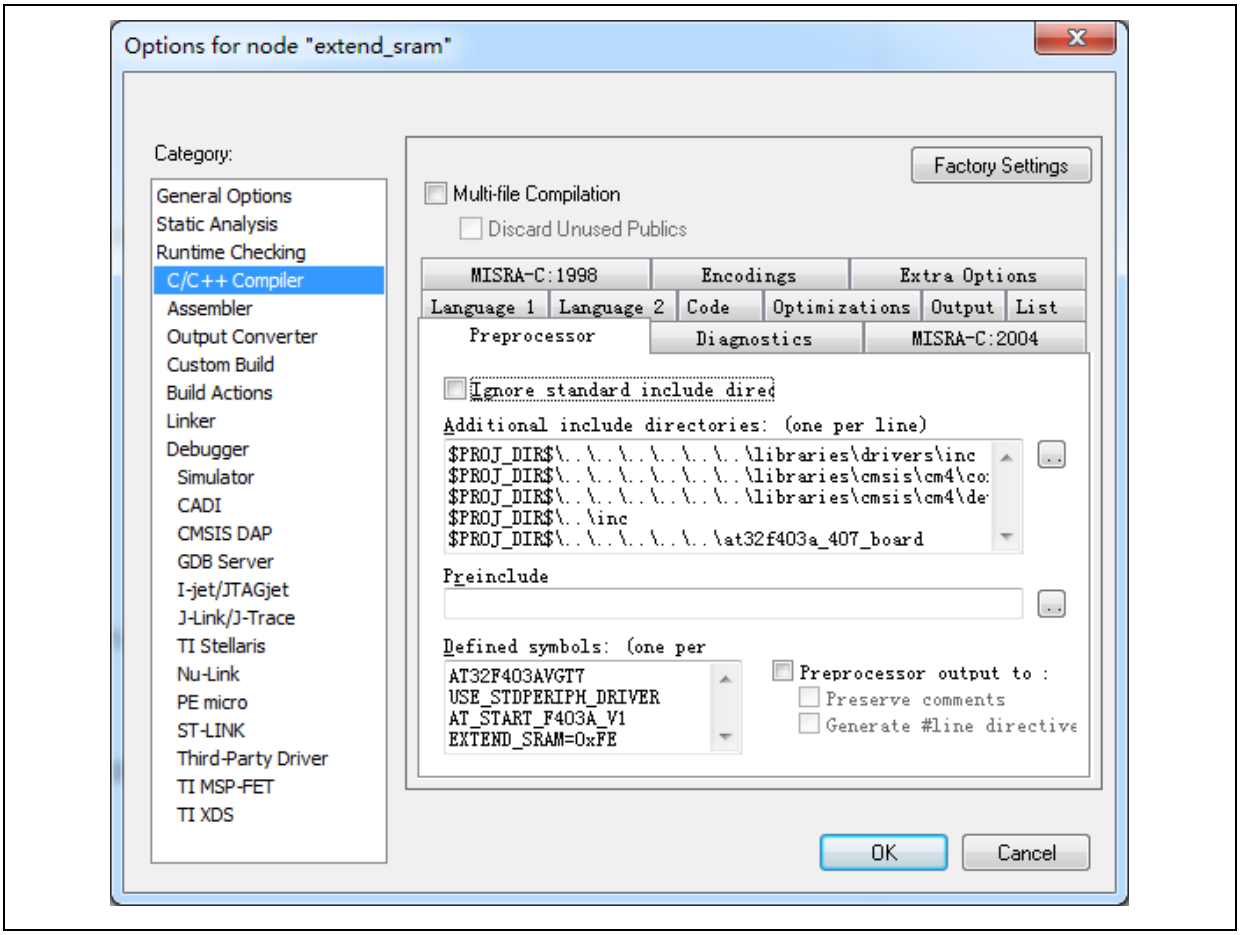
In this project, the user can select **set_SRAM_96 K or set_SRAM_224 K** through "*Select Target*" option, as shown in the figure below.

Figure 6. SRAM size selection in IAR



When the "**set_SRAM_96K**" or "**set_SRAM_224K**" is selected, the macro definition value of the EXTEND_SRAM will be displayed in the definition column of *C/C++ →Preprocessor Symbols*, and then the **extend_SRAM( )** function will select the corresponding configuration during compiling.

Figure 7. C/C++/Preprocessor Symbols configuration in IAR



When the extension operation is complete, it will enter the **main( )** function to check the EOPB0 value in order to verify if the SRAM size is set correctly or not. The verify result is indicated by turning on LED4.

3 Revision history

Table 2. Document revision history

Date	Revision	Changes
2021.05.24	2.0.0	Initial release

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Purchasers are solely responsible for the selection and use of ARTERY's products and services, and ARTERY assumes no liability whatsoever relating to the choice, selection or use of the ARTERY products and services described herein.

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document deals with any third party products or services, it shall not be deemed a license grant by ARTERY for the use of such third party products or services, or any intellectual property contained therein, or considered as a warranty regarding the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

Unless otherwise specified in ARTERY's terms and conditions of sale, ARTERY provides no warranties, express or implied, regarding the use and/or sale of ARTERY products, including but not limited to any implied warranties of merchantability, fitness for a particular purpose (and their equivalents under the laws of any jurisdiction), or infringement of any patent, copyright or other intellectual property right.

Purchasers hereby agrees that ARTERY's products are not designed or authorized for use in: (A) any application with special requirements of safety such as life support and active implantable device, or system with functional safety requirements; (B) any air craft application; (C) any automotive application or environment; (D) any space application or environment, and/or (E) any weapon application. Purchasers' unauthorized use of them in the aforementioned applications, even if with a written notice, is solely at purchasers' risk, and is solely responsible for meeting all legal and regulatory requirement in such use.

Resale of ARTERY products with provisions different from the statements and/or technical features stated in this document shall immediately void any warranty grant by ARTERY for ARTERY products or services described herein and shall not create or expand in any manner whatsoever, any liability of ARTERY.