GigaDevice Semiconductor Inc.

Arm® Cortex®-M33 32-bit MCU

Application Note
AN010

Table of Contents

List of Figures

List of Tables

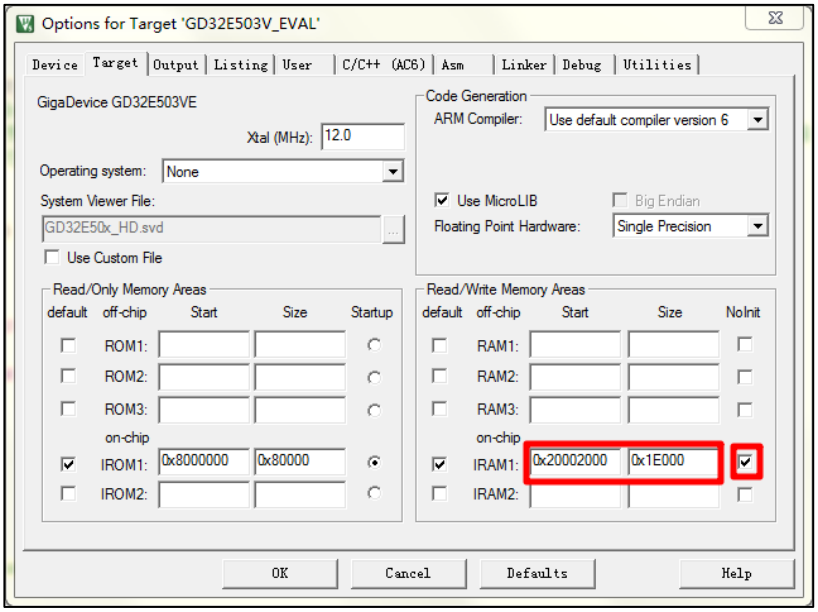
1. Introduction

In the field of low power control and application, the power consumption of MCU in low power mode is always paied much attention. GD32E5 series provide low power mode Deep-sleep 2, whose power consumption is much lower than that of Deep-sleep mode. In Deep-sleep 2 mode, all clocks in the 1.1V domain are off, and all of IRC8M, IRC48M, HXTAL and PLLs are disabled. The power of COREOFF0/COREOFF1 domain is cut off. The contents of SRAM except for the first 32KB and registers in COREOFF0/COREOFF1 domain are lost. In Deep-sleep 2 mode, part of the circuit including CPU will be power down, so when wakeup from the Deep-sleep 2 mode, the MCU will generate a system reset. The register configurations of peripherals except FMC, PMU, RCU, EXTI, GPIO, DBG, FWDGT, WWDGT, USART5 and I2C3 are lost. In order to keep the scene before entering the Deep-sleep 2 mode and recover the scene after waking up from Deep-sleep 2 mode, and meanwhile achieve low power consumption, a hardware and software cooperation mechanism is designed as follows.

2. SRAM partition

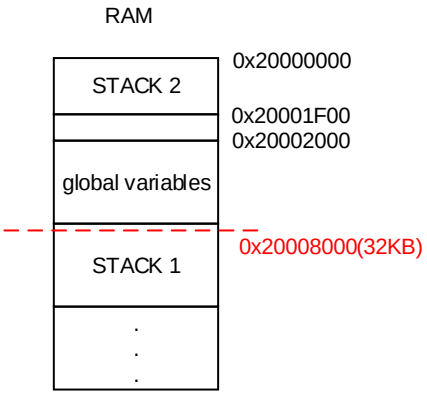
After waking up from the Deep-sleep 2 mode, the contents of the first 32KB of SRAM will not be lost. So the first 32KB of SRAM is retained for real-time data that needs to be recovered. Configure the IRAM in keil in ***Figure 2-1. Configuration of IRAM in keil***. The SRAM used by the keil start from the 0x20002000. And the "No Init" should be checked so that the zero space does not need to be initialized. The global variables must defined outside the main function without initialization, and initialized in the main function. If the global variables are initialized when defined outside the main function, when waked up from Deep-sleep 2 mode, the value of the global variables may be changed when initializing the stack before the main function.

Figure 2-1. Configuration of IRAM in keil



As is shown in ***Figure 2-2. SRAM partition***, two stacks are allocated, STACK 1 is allocated by compiler automatically, STACK 2 is allocated by user. Suppose that the global variables take up amount of SRAM space, and there is not enough space for the STACK 1 in the first 32KB of SRAM, the data in the STACK 1 will be lost when the MCU enters Deep-sleep 2 mode. The STACK 2 and the global variables locates in the first 32KB of SRAM, when waked up from the Deep-sleep 2 mode, the data in the STACK 2 and the global variables will not be lost. When waked up from the Deep-sleep 2 mode, the values of RCU registers are retained, but PLL will be reset, so it is recommended to configure the clock before main function.

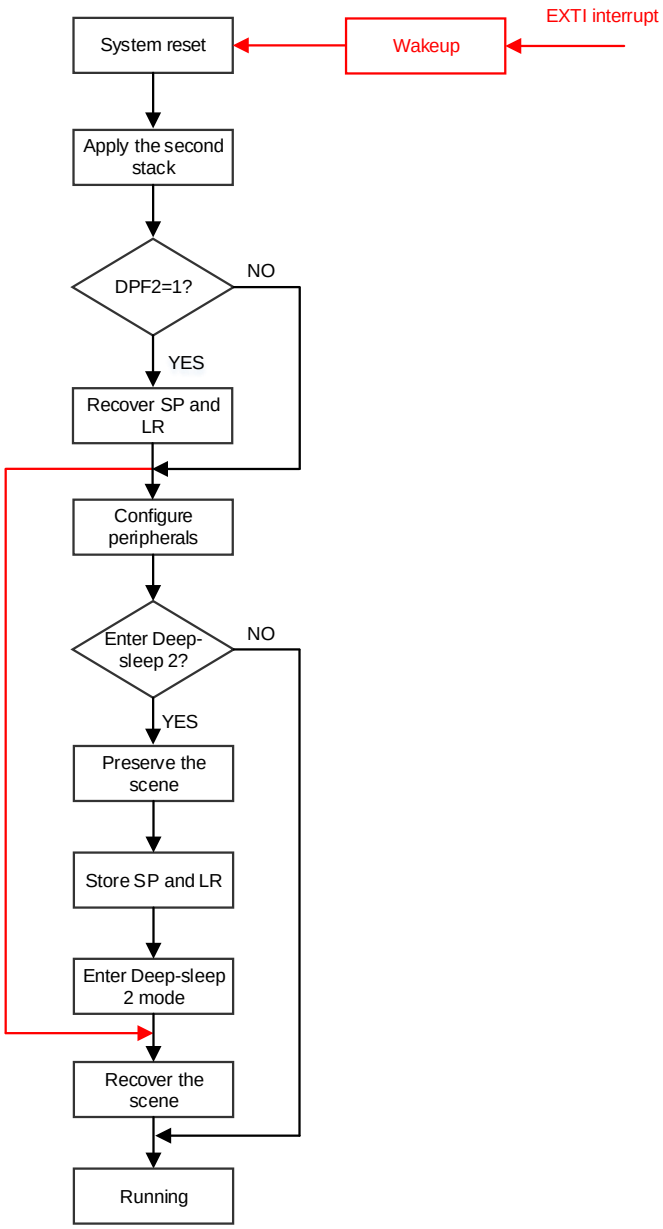
Figure 2-2. SRAM partition


RAM
STACK 2
0x20000000
0x20001F00
0x20002000
global variables
STACK 1
0x20008000(32KB)

3. Preserve and recover the scene

In Deep-sleep 2 mode, some circuitry (including CPU) will power down, so it needs to be reset after waking up. A mechanism is designed for software and hardware to work together to recover the scene, so that the program could continue to run where it had entered low power mode. The main program flow is shown in ***Figure 3-1. The main program flowchart***. The red line is only for the wakeup flow.

Figure 3-1. The main program flowchart



3.1. Apply for the second STACK

At the beginning of the main function, STACK 2 is applied for the application. The stack pointer points to address 0x20001F00, then the application will use STACK 2. The STACK 2 can be modified by user according to the actual stack size need by the application.

Table 3-1. Apply for STACK 2

```
__set_MSP(0x20001F00);
```

3.2. Preserve the sence

Before entering the Deep-sleep 2 mode, the values of registers that should not be reset can be stored in the global variables in the first 32KB of SRAM. As is shown in ***Table 3-2. Preserve the sence***.

Table 3-2. Preserve the sence

```
/* store the register configuration of EXMC */
exmc_config[0] = EXMC_SNCTL0;
exmc_config[1] = EXMC_SNTCFG0;
exmc_config[2] = EXMC_SNWTCFG0;
/* store the register configuration of USART */
usart_config[0] = USART_BAUD(EVAL_COM0);
usart_config[1] = USART_CTL0(EVAL_COM0);
```

Then store the SP and LR at address 0x20000000 and 0x20000004 respectively before entering the Deep-sleep 2 mode.

Table 3-3. Store the SP and LR

```
void pmu_deepsleep2_status_store(void)
{
    __asm__ __volatile__
    (
        "push {r0-r12}\n"
        "ldr r0,=0x20000000\n"
        /* store the value of sp at 0x20000000 */
        "str sp,[r0]\n"
        "ldr r0,=0x20000004\n"
        /* store the value of lr at 0x20000004 */
        "str lr,[r0]\n"
        "wfi\n"
    );
}
```

Table 3-4. Enter Deep-sleep 2 mode

```
void pmu_to_deepsleepmode_2(uint32_t ldo, uint8_t deepsleepmode2cmd)
{
    /* clear STBMOD and LDOLP bits */
    PMU_CTL0 &= ~((uint32_t)(PMU_CTL0_STBMOD | PMU_CTL0_LDOLP));
    /* clear deep-sleep 1 mode enable bit */
    PMU_CTL1 &= ~PMU_CTL1_DPMOD1;
    /* enable deep-sleep 2 mode */
    PMU_CTL1 |= PMU_CTL1_DPMOD2;
    /* configure LDOLP bit */
    PMU_CTL0 |= ldo;
    /* set sleepdeep bit of Cortex-M33 system control register */
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;
    /* select WFI or WFE command to enter deepsleep mode 2 */
    if(WFI_CMD == deepsleepmode2cmd){
        pmu_deepsleep2_status_store();
    }else{
        __SEV();
        __WFE();
        __WFE();
    }
    /* reset sleepdeep bit of Cortex-M33 system control register */
    SCB->SCR &= ~((uint32_t)SCB_SCR_SLEEPDEEP_Msk);
    PMU_CTL1 &= ~PMU_CTL1_DPMOD2;
}
```

3.3. Recover the sence

The DPF2 bit in PMU_CS1 register defines the Deep-sleep 2 mode status. If the DPF2 bit has been set, it indicates that the MCU has been entered Deep-sleep 2 mode. This bit is cleared by software by writing 0. When waked up from the Deep-sleep 2 mode, after system reset and apply the STACK 2, DPF2 will be checked, if DPF2 is 1, the SP and LR will be loaded with the values stored at address 0x20000000 and 0x20000004 respectively. Then the program jumps to the next line of the function pmu_deepsleep2_status_store and goes on.

Table 3-5. Recover the SP and LR

```
if((PMU_CS1 & PMU_CS1_DPF2) == 2){
    PMU_CS1 = 0;
    __asm__ __volatile__
    (
        "ldr r0,=0x20000000\n"
        /* load the value of 0x20000000 to SP */
```

```
            "ldr sp,[r0]\n"
            "ldr r0,=0x20000004\n"
            /* load the value of 0x20000004 to LR */
            "ldr lr,[r0]\n"
            "pop {r0-r12}\n"
            "bx lr\n"
        );
}
```

When the program runs out of the function pmu_to_deepsleepmode_2, the configuration of the registers can be recovered by loading the values stored before entering the Deep-sleep 2 mode in the the global variables.

Table 3-6. Recover the sence

```
pmu_to_deepsleepmode_2(PMU_LDO_NORMAL, WFI_CMD);
/* recover the register configuration of EXMC */
EXMC_SNCTL0 = exmc_config[0];
EXMC_SNTCFG0 = exmc_config[1];
EXMC_SNWTCFG0 = exmc_config[2];
/* recover the register configuration of USART */
USART_BAUD(EVAL_COM0) = usart_config[0];
USART_CTL0(EVAL_COM0) = usart_config[1];
```

So far, a mechanism designed to recover the application scene has achieved the similar feature of Deep-sleep mode that the program could continue to run where it had entered low power mode.

Further more, if the application is much complex and the global variables exceed the first 32KB of SRAM, the real-time or important data can be scatter loading in the first 32KB of SRAM. And discard unimportant global variables when entering the Deep-sleep 2 mode.

4. Revision history

Table 4-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Feb.24, 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.