

**GigaDevice Semiconductor Inc.**

**Arm<sup>®</sup> Cortex<sup>®</sup>-M33 32-bit MCU**

**应用笔记**

**AN010**

# 目录

目录.....	2
图索引.....	3
表索引.....	4
1. 简介.....	5
2. <b>SRAM</b> 分区.....	6
3. 保护和恢复现场.....	7
3.1. 申请第二个 <b>STACK</b> .....	7
3.2. 保护现场.....	8
3.3. 恢复现场.....	9
4. 版本历史.....	11

## 图索引

图 2-1. Keil 中 IRAM 配置.....	6
图 2-2. SRAM 分区.....	6
图 3-1. 主程序流程图.....	7

## 表索引

表 3-1. 申请 STACK 2 .....	8
表 3-2. 保护现场 .....	8
表 3-3. 保存 SP 和 LR .....	8
表 3-4. 进入深度睡眠模式 2 .....	8
表 3-5. 恢复 SP 和 LR .....	9
表 3-6. 恢复现场 .....	10
表 4-1. 版本历史 .....	11

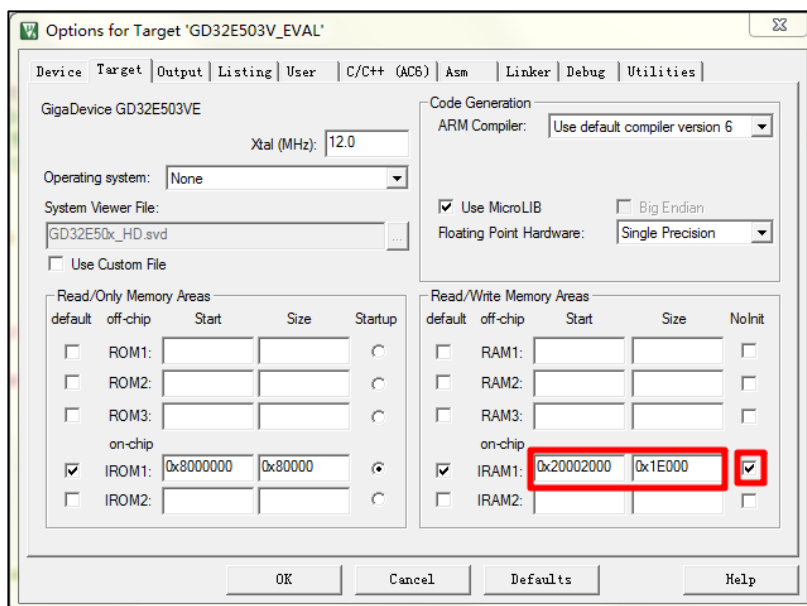
## 1. 简介

在低功耗控制和应用领域，单片机在低功耗模式下的功耗一直是备受关注的问题。GD32E5 系列提供低功耗模式深度睡眠模式 2，其功耗远低于深度睡眠模式。在深度睡眠模式 2 下，1.1V 域的所有时钟关闭，IRC8M，IRC48M，HXTAL 和 PLLs 被禁用。COREOFF0/COREOFF1 域电源被切断。SRAM 除前 32KB 和 COREOFF0/COREOFF1 域寄存器内容丢失。在深度睡眠模式 2 下，包括 CPU 在内的部分电路会断电，所以当从深度睡眠模式 2 中唤醒时，MCU 会产生一个系统复位。外设除 FMC、PMU、RCU、EXTI、GPIO、DBG、FWDGT、WWDGT、USART5、I2C3 外的寄存器配置丢失。为了在进入深度睡眠模式 2 前保持现场，从深度睡眠模式 2 醒来后恢复现场，同时实现低功耗，设计了如下文所述软硬件协同机制。

## 2. SRAM 分区

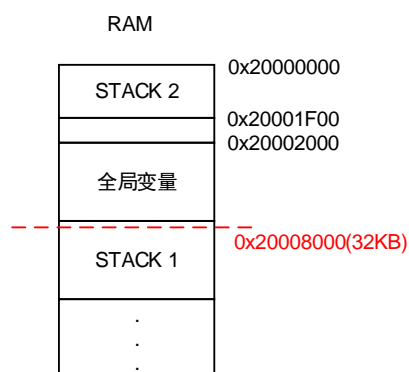
从深度睡眠模式 2 中唤醒后，SRAM 的前 32KB 内容不会丢失。因此，前 32KB 的 SRAM 被预留用于需要恢复的实时数据。在 keil 中配置如 [图 2-1. Keil 中 IRAM 配置](#) 所示 IRAM。keil 使用的 SRAM 从 0x20002000 开始。并且将 “No Init” 勾选上，这样零空间就不会进行初始化。全局变量必须在 main 函数外部定义，且在定义时不进行初始化，在 main 函数中初始化。如果全局变量在 main 函数外部定义时进行初始化，当从深度睡眠模式 2 中唤醒时，全局变量的值可能会在 main 函数之前初始化堆栈时被改变。

图 2-1. Keil 中 IRAM 配置



如 [图 2-2. SRAM 分区](#) 所示，分配了两个堆栈，STACK 1 由编译器自动分配，STACK 2 由用户分配。假设全局变量占用了大量的 SRAM 空间，SRAM 的前 32KB 没有足够的空间存放 STACK 1，当 MCU 进入深度睡眠模式 2 时，STACK 1 中的数据将丢失。STACK 2 和全局变量位于 SRAM 的前 32KB，当从深度睡眠模式 2 中唤醒时，STACK 2 中的数据和全局变量不会丢失。当从深度睡眠模式 2 唤醒时，RCU 寄存器的值保留，但是 PLL 会被重置，所以建议在 main 功能之前配置时钟。

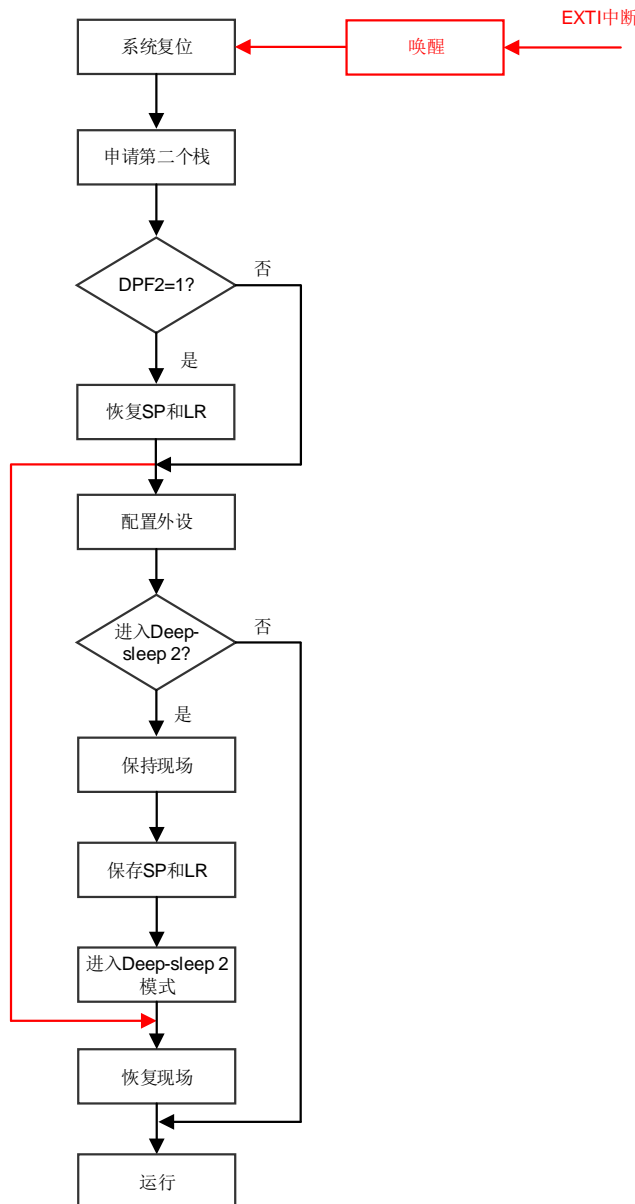
图 2-2. SRAM 分区



### 3. 保护和恢复现场

在深度睡眠模式 2 下，部分电路（包括 CPU）会断电，所以需要在唤醒后会复位。本文设计了一种软硬件协同的机制来恢复现场，这样程序就可以在唤醒后，从进入低功耗模式处继续运行。主程序流程如 [图 3-1. 主程序流程图](#) 所示。红线仅用于唤醒流程。

图 3-1. 主程序流程图



#### 3.1. 申请第二个 STACK

在主函数的开始，申请 STACK 2 用于应用程序。堆栈指针指向地址 0x20001F00，然后应用程序将使用 STACK 2。STACK 2 可以由用户根据应用程序实际需要的堆栈大小进行修改。

**表 3-1. 申请 STACK 2**

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

## 3.2. 保护现场

在进入深度睡眠模式 2 之前，不能被复位的寄存器值可以存储在 SRAM 前 32KB 的全局变量中。如 [表 3-2. 保护现场](#) 所示。

**表 3-2. 保护现场**

```
/* 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);
```

在进入深度睡眠模式 2 之前，将 SP 和 LR 分别存储在 0x20000000 和 0x20000004。

**表 3-3. 保存 SP 和 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"
    );
}
```

**表 3-4. 进入深度睡眠模式 2**

```
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 |= Ido;
/* 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. 恢复现场

PMU\_CS1 寄存器中的 DPF2 位定义了深度睡眠模式 2 状态。如果 DPF2 位置位，则表示 MCU 进入过深度睡眠模式 2。该位通过写 0 清零。当从深度睡眠模式 2 中唤醒后，系统复位并应用 STACK 2 后，将检查 DPF2，如果 DPF2 为 1，SP 和 LR 将分别加载存储在地址 0x20000000 和 0x20000004 的值。然后，程序跳转到函数 pmu\_deepsleep2\_status\_store 的下一行并继续运行。

表 3-5. 恢复 SP 和 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"
    );
}

```

当程序运行完 pmu\_to\_deepsleepmode\_2 函数时，可以通过加载进入深度睡眠模式 2 之前存储在全局变量中的值来恢复寄存器的配置。

表 3-6. 恢复现场

```
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];
```

到目前为止，一种旨在恢复应用现场的机制已经实现了与深度睡眠模式类似的功能，即程序可以在进入低功耗模式处继续运行。

此外，如果应用程序非常复杂，全局变量超过了 **SRAM** 的前 **32KB**，则实时数据或重要数据可以分散加载在 **SRAM** 的前 **32KB** 中。当进入深度睡眠模式 2 时，丢弃不重要的全局变量。

## 4. 版本历史

表 4-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2021 年 2 月 24 日

## 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 its 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 its 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.