GigaDevice Semiconductor Inc.

Arm® Cortex®- M3/M4/M23/M33 32-bit MCU

Application Note
AN035

Table of Contents

List of Figures

List of Tables

1. Introduction

The GD32F10x devices provide three kinds of boot sources which can be selected by the BOOT0 and BOOT1 pins. The details are shown in ***Table 1-1. Boot modes***. The value on the two pins is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT0 and BOOT1 pins after a power-on reset or a system reset to select the required boot source. Once the two pins have been sampled, they are free and can be used for other purposes.

Table 1-1. Boot modes

Selected boot source	Boot mode selection pins	
	Boot1	Boot0
Main Flash Memory	x	0
Boot loader	0	1
On-chip SRAM	1	1

Note: When the boot source is hoped to be set as “Main Flash Memory”, the Boot0 pin has to be connected with GND definitely and can not be floating.

After power-on sequence or a system reset, the Arm® Cortex®-M3/M4/M23/M33 processor fetches the top-of-stack value from address 0x0000 0000 and the base address of boot code from 0x0000 0004 in sequence. Then, it starts executing code from the base address of boot code.

Due to the selected boot source, either the main flash memory (original memory space beginning at 0x0800 0000) or the system memory (original memory space beginning at 0x1FFF F000) is aliased in the boot memory space which begins at the address 0x0000 0000. When the on-chip SRAM whose memory space is beginning at 0x2000 0000 is selected as the boot source, in the application initialization code, you have to relocate the vector table in SRAM using the NVIC exception table and offset register.

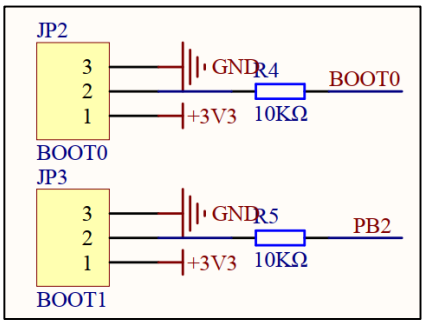
The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory.

2. Boot from SRAM

2.1. Hardware configuration

When boot from SRAM, the level of BOOT0 and BOOT1 must be configured as high, as is shown in ***Table 1-1. Boot modes***. When designing the circuit, a jumper cap is usually used to switch the high and low levels of boot pins, as is shown in ***Figure 2-1. Schematic of BOOT pins***.

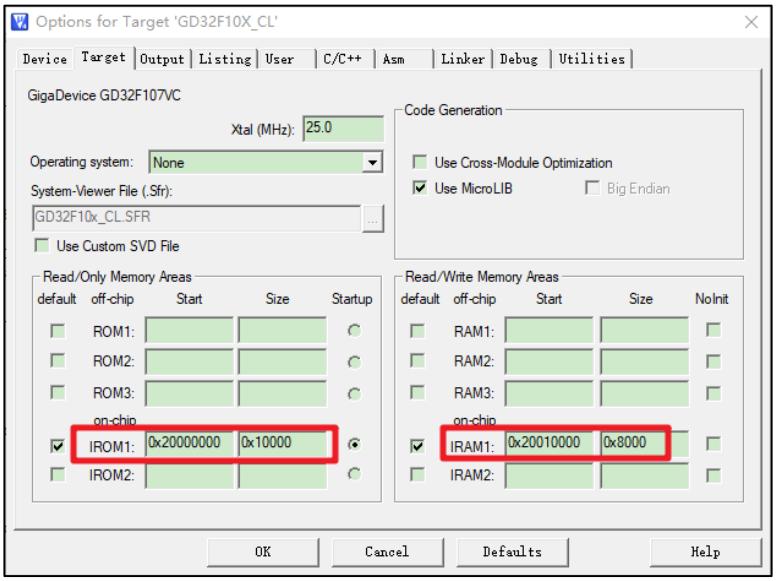
Figure 2-1. Schematic of BOOT pins



2.2. Configuration steps in Keil

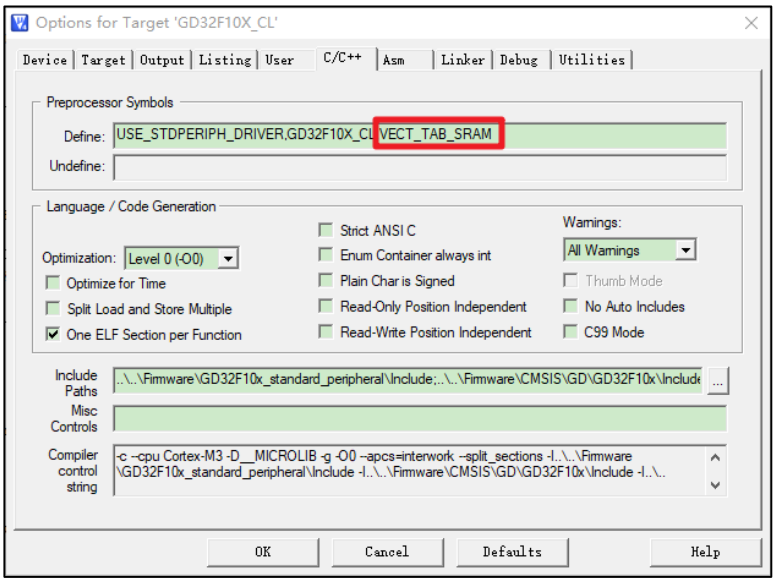
1. Configure IROM1 and IRAM1 as SRAM address in “Option for Target -> Target”, as is shown in ***Figure 2-2. Configuration of IROM1 and IRAM1 address***.

Figure 2-2. Configuration of IROM1 and IRAM1 address



2. Use the NVIC exception table and offset register to realocate the vector table to SRAM. Add the global macro “VECT_TAB_SRAM” to “Option for Target -> c/c++ -> Define”, as is shown in ***Figure 2-3. Add the global macro “VECT_TAB_SRAM”***.

Figure 2-3. Add the global macro "VECT_TAB_SRAM"



Add the code related to the macro " VECT_TAB_SRAM " in the SystemInit() function, as is shown in ***Table 2-1. Add the code related to the macro " VECT_TAB_SRAM***.

Table 2-1. Add the code related to the macro " VECT_TAB_SRAM

```
/*!
    \brief      setup the microcontroller system, initialize the system
    \param[in]  none
    \param[out] none
    \retval     none
*/
void SystemInit(void)
{
    /* reset the RCU clock configuration to the default reset state */
    /* enable IRC8M */
    RCU_CTL |= RCU_CTL_IRC8MEN;
    /* reset SCS, AHBPSC, APB1PSC, APB2PSC, ADCPSC, CKOUT0SEL bits */
    RCU_CFG0 &= ~(RCU_CFG0_SCS | RCU_CFG0_AHBPSC | RCU_CFG0_APB1PSC | RCU_CFG0_APB2PSC |
                  RCU_CFG0_ADCPSC | RCU_CFG0_ADCPSC_2 | RCU_CFG0_CKOUT0SEL);
    /* reset HXTALEN, CKMEN, PLLEN bits */
    RCU_CTL &= ~(RCU_CTL_HXTALEN | RCU_CTL_CKMEN | RCU_CTL_PLLEN);
    /* Reset HXTALBPS bit */
    RCU_CTL &= ~(RCU_CTL_HXTALBPS);
    /* reset PLLSEL, PREDV0_LSB, PLLMF, USBFSPSC bits */
#ifdef GD32F10X_CL
    RCU_CFG0 &= ~(RCU_CFG0_PLLSEL | RCU_CFG0_PREDV0_LSB | RCU_CFG0_PLLMF |
                  RCU_CFG0_USBFSPSC | RCU_CFG0_PLLMF_4);
```

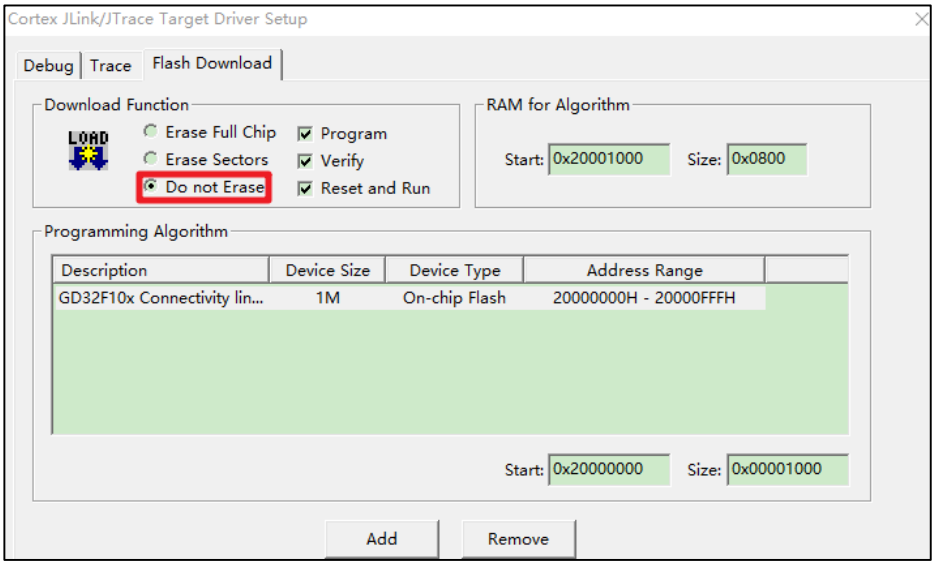
```
    RCU_CFG1 = 0x00000000U;
#else
    RCU_CFG0 &= ~(RCU_CFG0_PLLSEL | RCU_CFG0_PREDV0 | RCU_CFG0_PLLMF |
                  RCU_CFG0_USBDPSC | RCU_CFG0_PLLMF_4);
#endif /* GD32F10X_CL */

#if (defined(GD32F10X_MD) || defined(GD32F10X_HD) || defined(GD32F10X_XD))
    /* reset HXTALEN, CKMEN and PLLEN bits */
    RCU_CTL &= ~(RCU_CTL_PLLEN | RCU_CTL_CKMEN | RCU_CTL_HXTALEN);
    /* disable all interrupts */
    RCU_INT = 0x009F0000U;
#elif defined(GD32F10X_CL)
    /* Reset HXTALEN, CKMEN, PLLEN, PLL1EN and PLL2EN bits */
    RCU_CTL &= ~(RCU_CTL_PLLEN | RCU_CTL_PLL1EN | RCU_CTL_PLL2EN |
RCU_CTL_CKMEN | RCU_CTL_HXTALEN);
    /* disable all interrupts */
    RCU_INT = 0x00FF0000U;
#endif

    /* Configure the System clock source, PLL Multiplier, AHB/APBx prescalers and Flash settings */
    system_clock_config();
#ifdef VECT_TAB_SRAM
    nvic_vector_table_set(NVIC_VECTTAB_RAM,VECT_TAB_OFFSET);
#else
    nvic_vector_table_set(NVIC_VECTTAB_FLASH,VECT_TAB_OFFSET);
#endif
}
```

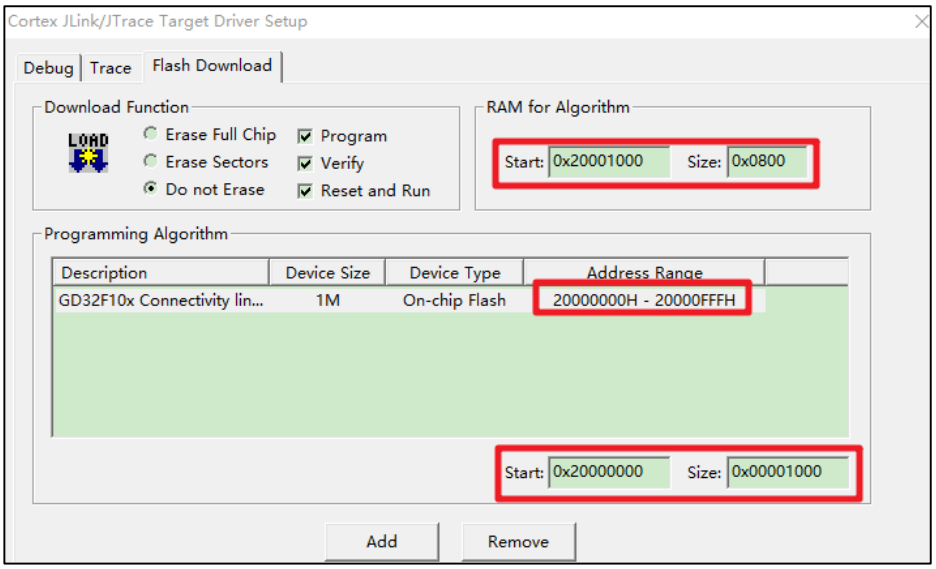
3. Configure the erase mode as “Do not Erase” in “Option for Target -> Debug -> Setting -> Flash Download”, as is shown in ***Figure 2-4. Select the erase mode***.

Figure 2-4. Select the erase mode



4. Configure the algorithm address as SRAM address in “Option for Target -> Debug -> Setting -> Flash Download”, as is shown in ***Figure 2-5. Configure the algorithm address***.

Figure 2-5. Configure the algorithm address



5. Before Reset_Handler in the startup file (such as startup_gd32f10x_cl.s), use SPACE to apply for a section of empty memory, as is shown in ***Figure 2-6. Use SPACE to apply empty memory***. So as to locate the Reset_Handler at address 0x200001E0, as is shown in ***Figure 2-7. Realocate the address of Reset_Handler***.

Figure 2-6. Use SPACE to apply empty memory

```
Skip_Mem        SPACE   0x7C
                ;DCD     0xF1E0F85F
__Vectors_End

__Vectors_Size  EQU  __Vectors_End - __Vectors

                AREA    |.text|, CODE, READONLY

;/* reset Handler */
Reset_Handler   PROC
                EXPORT  Reset_Handler                   [WEAK]
                IMPORT  __main
                IMPORT  SystemInit
                LDR     R0, =SystemInit
                BLX     R0
                LDR     R0, =__main
                BX      R0
                ENDP
```

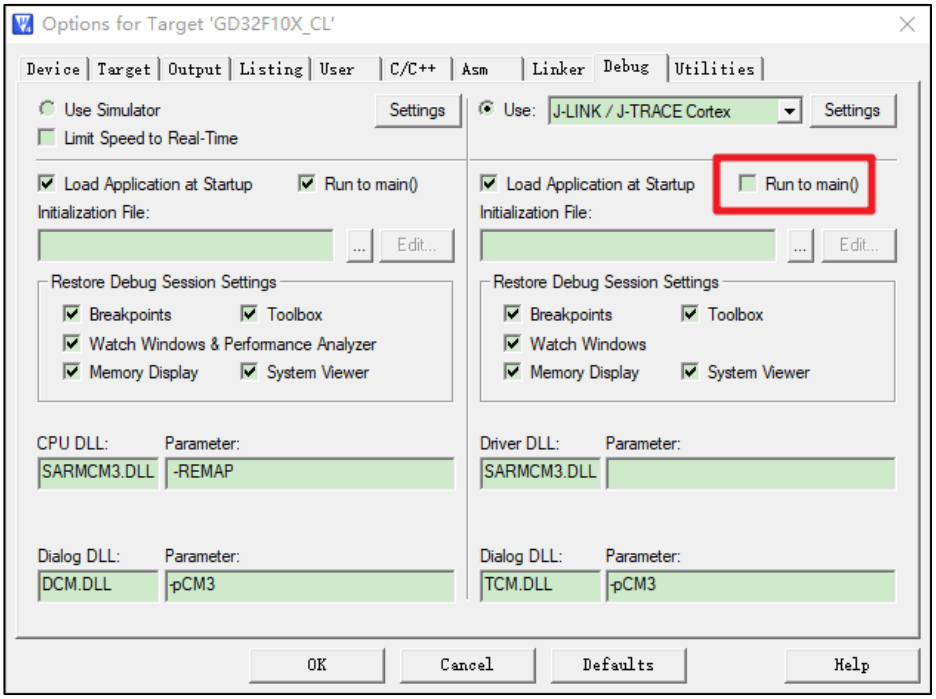
Figure 2-7. Realocate the address of Reset_Handler

```
__rt_final_cpp                  0x200001dd   Thumb Code     0
__rt_final_exit                 0x200001dd   Thumb Code     0
Reset_Handler                   0x200001e1   Thumb Code     8
ADC0_1_IRQHandler               0x200001fb   Thumb Code     0
CAN0_EWMC_IRQHandler            0x200001fb   Thumb Code     0
CAN0_RX0_IRQHandler             0x200001fb   Thumb Code     0
CAN0_RX1_IRQHandler             0x200001fb   Thumb Code     0
CAN0_TX_IRQHandler              0x200001fb   Thumb Code     0
```

3. Demonstration in debug mode

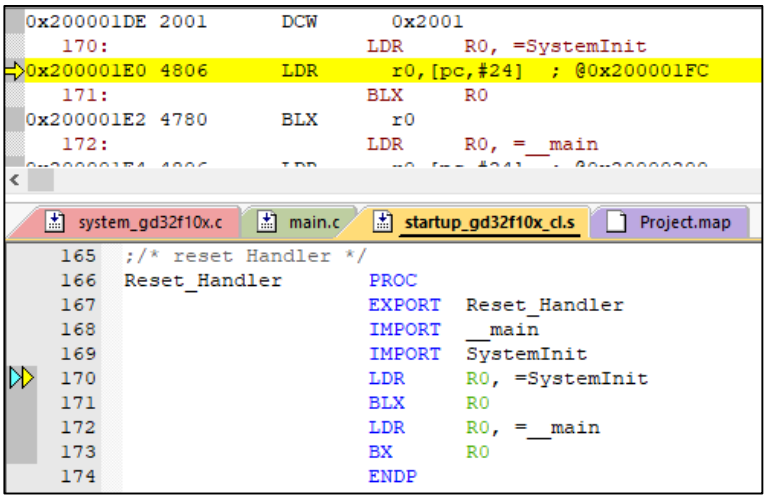
Remove “Run to main()” in “Option for Target -> Debug”, as is shown in ***Figure 3-1. Remove “Run to main()”***.

Figure 3-1. Remove “Run to main()”



Enter the debug mode, the program starts running at the address 0x200001E0.

Figure 3-2. Debug the program



So far, boot from SRAM is successfully. As long as the power is not cut off, the program can run after reset.

4. Revision history

Table 4-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.01, 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.