GigaDevice Semiconductor Inc.

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

Application Note AN035



Table of Contents

Tabl	le of Contents	2
List	of Figures	3
List	of Tables	4
1.	Introduction	5
2.	Boot from SRAM	6
2.1	1. Hardware configuration	6
2.2	2. Configuration steps in Keil	6
3.	Demonstration in debug mode	
4.	Revision history	



List of Figures

Figure 2-1. Schematic of BOOT pins	6
Figure 2-2. Configuration of IROM1 and IRAM1 address	6
Figure 2-3. Add the global macro "VECT_TAB_SRAM"	7
Figure 2-4. Select the erase mode	9
Figure 2-5. Configure the algorithm address	9
Figure 2-6. Use SPACE to apply empty memory	10
Figure 2-7. Realocate the address of Reset_Handler	10
Figure 3-1. Remove "Run to main()"	11
Figure 3-2. Debug the program	11



List of Tables

Table 1-1. Boot modes	. 5
Table 2-1. Add the code related to the macro " VECT_TAB_SRAM	. 7
Table 4-1. Revision history	12



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 <u>Table 1-1. Boot modes</u>. 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	х	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 <u>Table 1-1. Boot modes</u>. When designing the circuit, a jumper cap is usually used to switch the high and low levels of boot pins, as is shown in <u>Figure 2-1. Schematic of BOOT</u> <u>pins</u>.

Figure 2-1. Schematic of BOOT pins



2.2. Configuration steps in Keil

 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

Options for Target 'GD32F10X_CL' X							
Device Target Output Listing Vser C/C++ Asm Linker Debug Utilities							
GigaDevice GD32F107VC							
Xtal (MHz): 25.0							
Operating system: None	•	Use Cross-Module Optimization					
System-Viewer File (.Sfr):		Use MicroLIB 🔲 Big Endiar	n				
GD32F10x_CL.SFR							
Use Custom SVD File							
Read/Only Memory Areas —		Read/Write Memory Areas					
default off-chip Start	Size Startup	default off-chip Start Size	Nolnit				
ROM1:	0	RAM1:					
ROM2:	c	RAM2:					
ROM3:	C	□ RAM3:					
on-chip		on-chip					
IROM1: 0x2000000	00 0x10000 (*	✓ IRAM1: 0x20010000 0x8000					
IROM2:	0	IRAM2:					
,	,	,,					
	OK Can	cel Defaults	Help				

 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"

Options for Target 'GD32 Device Target Output Li	2F10X_CL' isting User C/C++ Asm Linker Debug	; Utilities			
Preprocessor Symbols Define: USE_STDPERIPH_DRIVER,GD32F10X_CL_VECT_TAB_SRAM Undefine:					
Language / Code Generation Optimization: Level 0 (-00) Optimize for Time Split Load and Store Mult Image: One ELF Section per Function One ELF Section per Function	n Strict ANSI C Enum Container always int Plain Char is Signed tiple Read-Only Position Independent nction Read-Write Position Independent	Warnings: All Warnings Thumb Mode No Auto Includes C99 Mode			
Include Paths Misc Controls Compiler control string Compiler Contex-M3-D_MICROLIB-g-00 ~apcs=interwork -split_sections -1,\Firmware (GD32F10x_standard_peripheral\Include -1,\Firmware\CMSIS\GD\GD32F10x\Include -1, Virmware					
	OK Cancel Default:	s Help			

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 = 0x0000000U;
#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
```

 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

ex JLink/JTrace Target Driver	Setup				
bug Trace Flash Download	H				
Download Function	RAM	for Algorithm ——			
Erase Sectors 🖉 Do not Erase	✓ Verify ✓ Reset and	Sta Id Run	art: 0x20001000	Size: 0x08	300
Programming Algorithm			1		
Description	Device Size	Device Type	Address R	ange	
GD32F10x Connectivity lin	1M	On-chip Flash	2000000H - 20	DOOOFFFH	
		Sta	art: 0x20000000	Size: 0x00	0001000
	Ad	ld Rem	ove		

 Configure the algorithm address as SRAM address in "Option for Target -> Debug -> Setting -> Flash Download", as is shown in <u>Figure 2-5. Configure the algorithm</u> <u>address</u>.

Figure 2-5. Configure the algorithm address

ug Trace	Flash Download	4				
ownload Fi	© Erase Full Chip © Erase Sectors © Do not Erase	o IV Program IV Verify IV Reset an	d Run	for Algorithm	Size: 0x08	800
ogrammin	ig Algonum		Davies Tures	Adda.co. D		1
Descriptio GD32F10x	on Connectivity lin	1M	On-chip Flash	20000000H - 20	0000FFFH	<u> </u>
Descriptio GD32F10x	on Connectivity lin	1M	On-chip Flash	20000000H - 20	ange DOOOFFFH	1
Descriptio GD32F10x	on Connectivity lin	1M	On-chip Flash	20000000H - 20	ange DOOOFFFH]
Descriptio GD32F10x	n Connectivity lin	1M	On-chip Flash	Address N 20000000H - 20	Sizel 0x00	0001000

 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* <u>empty memory</u>. 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 ; DCD	0x7C 0xF1E0F85F	
Vectors_End	,		
Vectors_Size	EQU	Vectors_EndVectors	
	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	RO	
	FNDP		

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()*".

🛛 Options for Target 'GD32F10X_CL' × Device | Target | Output | Listing | User | C/C++ | Asm | Linker Debug | Utilities | C Use Simulator Settings Use: J-LINK / J-TRACE Cortex Settings Limit Speed to Real-Time Load Application at Startup Run to main() Load Application at Startup Run to main() Initialization File: Initialization File: E dit... E dit. Restore Debug Session Settings Restore Debug Session Settings ▼ Toolbox ▼ Toolbox Breakpoints Breakpoints ✓ Watch Windows & Performance Analyzer ✓ Watch Windows Memory Display System Viewer Memory Display System Viewer CPU DLL: Parameter: Driver DLL: Parameter SARMCM3.DLL -REMAP SARMCM3.DLL Dialog DLL: Parameter Dialog DLL: Parameter DCM.DLL TCM.DLL -рСМЗ -pCM3 OK Cancel Defaults Help

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

	0 x 200001	DE 2	001	DCW	0 x 2	001
	170:				LDR	R0, =SystemInit
₽	0 x 200001	LEO 4	806	LDR	r0,	[pc,#24] ; @0x200001FC
	171:				BLX	R0
	0 x 200001	E2 4	780	BLX	rO	
	172:				LDR	R0, =main
<	0	TE 4 4	000	TOD		+041 - 00-00000000
-	_					
	📩 syste	m_gd32	2f10x.c	📩 main.c	📩 sta	rtup_gd32f10x_cl.s Project.map
	165	;/*	reset	Handler */	/	
	166	Rese	t_Hand	ller	PROC	
	167				EXPOR	T Reset_Handler
	168				IMPOR	Tmain
	169				IMPOR	T SystemInit
	> 170				LDR	R0, =SystemInit
	171				BLX	R0
	172				LDR	RO, =main
	173				BX	R0
	174				ENDP	

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.

© 2021 GigaDevice - All rights reserved