GigaDevice Semiconductor Inc.

Migration from GD32E230 series to GD32F3x0 series

Application Note AN046



Table of Contents

Tabl	e of Contents	2
List	of Figures	3
List	of Tables	5
1.	Introduction	6
2.	Introduction to hardware differences	7
3.	Comparison of resource and peripheral addresses	9
4.	Comparison of development tools	12
5.	Software environment settings	13
5.1 ៖	I. Using Keil to develop GD32F3x05.1.1. Add GD32F3x0 MCU device in Keil45.1.2. Add GD32F3x0 MCU device in Keil5	13 13 15
5.2	2. Debugging and simulating GD32F3x0 with GD-Link	17
5.3	Debugging and simulating GD32F3x0 with J-Link	19
5.4 t	I.Using IAR to develop GD32F3x05.4.1Add gd32F3x0 MCU device in IAR5.4.2Debugging GD32F3x0 in IAR	
6.	Steps of GD32E23x firmware library adapting to GD32F3x0 Series M	CU 26
7.	Steps to replace GD32E23x project library with GD32F3x0 Library	31
8.	Revision history	



List of Figures

Figure 2-1. Comparison diagram of LQFP48 package of GD32F3x0 and GD32E230xx7
Figure 2-2. Comparison diagram of QFN32 package of GD32F3x0 and GD32E230xx7
Figure 2-3. Comparison diagram of QFN28 package of GD32F3x0 and GD32E230xx8
Figure 2-4. Comparison diagram of TSSOP20 package of GD32F330/F310xx and GD32E230xx 8
Figure 2-5. Comparison diagram of LQFP32 package of GD32F310xx and GD32E230xx
Figure 5-1. GD32F3x0 plug-in package details 13
Figure 5-2. Installation diagram of GD32F3x0 Series MCU plug-in package (Keil4) 13
Figure 5-3. Successful installation of GD32F3x0 Series MCU plug-in package (Keil4)14
Figure 5-4. GD32F3x0 series flash algorithm file selection diagram (Keil4) 14
Figure 5-5. GD32F3x0 plug-in package details 15
Figure 5-6. Installation diagram of GD32F3x0 Series MCU plug-in package (Keil5) 15
Figure 5-7. Successful installation of GD32F3x0 Series MCU plug-in package (Keil5) 16
Figure 5-8. GD32F3x0 series flash algorithm file selection diagram (Keil5) 16
Figure 5-9. Select the "CMSIS-DAP Debugger" option in the Debug interface (Keil4) 17
Figure 5-10. Select the "CMSIS-DAP Debugger" option in the Utilities interface (Keil4)
Figure 5-11. GD-Link tool successfully connected to the target MCU(Keil4) 18
Figure 5-12. Schematic diagram of adding flash algorithm file(Keil4) 19
Figure 5-13. Schematic diagram of GD-Link simulation(Keil4) 19
Figure 5-14. Select the "J-LINK/J-Trace Cortex" option in the Debug interface (Keil4) 20
Figure 5-15. Select the "J-LINK/J-Trace Cortex" option in the Utilities interface (Keil4)
Figure 5-16. J-Link tool successfully connected to the target MCU(Keil4)
Figure 5-17. Schematic diagram of adding flash algorithm file(Keil4)
Figure 5-18. Schematic diagram of J-Link simulation(Keil4)
Figure 5-19. Installation diagram of GD32F3x0 Series MCU plug-in package(IAR) 22
Figure 5-20. Successful installation of GD32F3x0 Series MCU plug-in package (IAR) 22
Figure 5-21. Select the GD device in the IAR "Options" interface
Figure 5-22. Add CMSIS file in IAR "Options" interface
Figure 5-23. Add ICF file in IAR "Options" interface
Figure 5-24. Select the debugger tool in the IAR "Options" interface
Figure 5-25. Configure flash loader in IAR "Options" interface
Figure 6-1. Open GD32E23x Keil project 26
Figure 6-2. Select GD32F3x0 device in GD32E23x project
Figure 6-3. Add the flash algorithm of GD32F3x0 27
Figure 6-4. Add Cortex M4 kernel files to GD32E23x firmware library 27
Figure 6-5. Modify the contents of "gd32e23x.h"
Figure 7-1. Copy h file in CMSIS of GD32F3x0 firmware library to GD32E23x
Figure 7-2. Copy and replace the Include and Source files in CMSIS under GD32F3x0 firmware
library into GD32E23x firmware library 31
Figure 7-3. Copy and replace the Include and Source files in standard_peripheral under
GD32F3x0 firmware library into GD32E23x firmware library



Figure 7-4. Copy the "gd32f3x0_libopt.h" file in GD32F3x0 firmware library into GD32E	23x
firmware library	. 31
Figure 7-5. Open the Keil project under the template file in the GD32E23x firmware library	. 32
Figure 7-6. Remove the Yellow marked files and add new files	. 33
Figure 7-7. Modify the contents of "main.c", "systick.c" files	. 33
Figure 7-8. Reselect GD32F3x0 MCU device	. 34
Figure 7-9. Reselect GD32F3x0 Flash algorithm	. 34



List of Tables

Table 3-1. GD32F3x0 series and gd32E230xx series resources comparison overview 9
Table 3-2. GD32F3x0 series and GD32E230xx series peripheral address comparison overview 9
Table 4-1. Comparison of IDE environment between GD32F3x0 series and GD32E230xx series
Table 4-2. Comparison of GD32F3x0 series and GD32E230xx series debugging tools 12
Table 6-1. Modify the contents of "gd32e23x.h" 27
Table 6-2. Modify the contents of "gd32e23x_misc.h"
Table 6-3. Modify the contents of "gd32e23x_misc.c" 28
Table 6-4. Modify the contents of "gd32e23x_misc.h"
Table 6-5. Modify the contents of "gd32e23x_misc.c" 28
Table 6-6. Remove the function of waiting period in GD32E23x project
Table 6-7. Add half word programming to "gd32e23x_fmc.h" of GD32E23x project 30
Table 6-8. Add half word programming to "gd32e23x_fmc.c" of GD32E23x project
Table 7-1. nvic_priority_group_set function 34
Table 7-2. nvic_irq_enable function
Table 8-1. Revision history



1. Introduction

This application note is designed to help you quickly migrate applications from GD32E230xx series MCU to GD32F3x0 series MCU.

In order to make better use of the information in this application note, you need to download it from the website www.GD32MCU.com, such as datasheet, user manual, official code and various development tools.



2. Introduction to hardware differences

The package types of GD32E230xx series include: TSSOP20, LGA20, QFN28, QFN32, LQFP32 and LQFP48; The package types of GD32F3x0 series include:TSSOP20 (GD32F330/F310xx series only), QFN28, QFN32, LQFP32(GD32F310xx series only), LQFP48, LQFP64(GD32F330/F350xx series only). The chip pins of the same package of the two series are compatible, see *Figure 2-1. Comparison diagram of LQFP48 package of GD32F3x0 and GD32E230xx*, *Figure 2-2. Comparison diagram of QFN32 package of GD32F3x0 and GD32E230xx*, *Figure 2-3. Comparison diagram of QFN28 package of GD32F3x0 and GD32E230xx*, *Figure 2-4. Comparison diagram of TSSOP20 package of GD32F330/F310xx and GD32E230xx*, *Figure 2-5. Comparison diagram of LQFP32 package of GD32F310xx and GD32E230xx*.

- 1. In the package of TSSOP20 and QFN28, PA9 and PA10 of GD32E230xx series can be mapped to PA11 and PA12. GD32F3x0 series does not have this function.
- LQFP48 package's pin 1 is V_{DD} on GD32E230xx series and Vbat on GD32F3x0, that is, GD32E230xx does not support power down RTC.











Figure 2-3. Comparison diagram of QFN28 package of GD32F3x0 and GD32E230xx









GD32E230Kx LQFP32 pinouts	GD32F310KxT6 LQFP32 pinouts
32 31 30 29 28 27 26 25	32 31 30 29 28 27 26 25
VDD 1 4 24 PA14 PF0-OSCIN 2 23 PA13 PF1-OSCOUT 3 22 PA12 NRST 4 GigaDevice GD32E230Kx 21 PA11 VDDA 5 20 PA10 PA0 6 19 PA9 PA1 7 18 PA8	VDD 1 ● 24 PA14 PF0-OSCIN 2 23 PA13 PF1-OSCOUT 3 GigaDevice 22 PA12 NRST 4 GD32F310KxT6 21 PA11 VDDA 5 LQFP32 20 PA10 PA0 6 19 PA9 PA1 7 18 PA8
PA2 8 17 VOD 9 10 11 12 13 14 15 16 9 10 11 12 13 14 15 16	PAZ U 8 9 10 11 12 13 14 15 16 9 10 11 12 13 14 15 16 A A A A A A A A A A A A A A A A A A A



3.

Comparison of resource and peripheral addresses

The resources of GD32F3x0 series and GD32E230xx series are slightly different:

- 1. TIMER1 is added to GD32F3x0 series, but TIMER5 is cut out(GD32F350xx has this peripheral). GD32E230xx series has TIMER5, but there is no TIMER1;
- 2. GD32E230xx and GD32F350xx series has a comparator, there is no one in GD32F330/F310xx;
- 3. GD32E230xx series adds 1K OTP area, which is not available in GD32F3x0 series;
- 4. GD32F350xx series has USBFS, HDMI-CEC and DAC peripheral, GD32F330/F310xx series and GD32E230xx series do not have these peripheral.

Please check for details in <u>Table 3-1. GD32F3x0 series and gd32E230xx series resources</u> <u>comparison overview</u> and <u>Table 3-2. GD32F3x0 series and GD32E230xx series</u> <u>peripheral address comparison overview</u>.

Peripheral	GD32F310xx	GD32F330xx	GD32F350xx	GD32E230xx
Core	Cortex-M4	Cortex-M4	Cortex-M4	Cortex-M23
Flash	16K-64K	16K-128K	16K-128K	16K-64K
RAM	4K-8K	4K-16K	4K-16K	4K-8K
Frequency	72MHz	84MHz	108MHz	72MHz
GPTM(32bit)	0	1	1	0
GPTM(16bit)	4/5	4/5	5	4/5
AdvTM	1	1	1	1
BaseTM	0	0	1	1
U(S)ART	1/2	1/2	1/2	1/2
I2C	1/2	1/2	1/2	1/2
SPI	1/2	1/2	1/2	1/2
I2S	1	0	1	1
USBFS	0	0	1	0
HDMI-CEC	0	0	1	0
TSI	0	0	1	0
COMP	0	0	2	1
ADC	1(9)/1(10)	1(9)/1(10)/1(16)	1(10)/1(16)	1(9)/1(10)
DAC	0	0	1	0
Operating Voltage	2.6-3.6V	2.6-3.6V	2.6-3.6V	1.8-3.6V
Temperature Range	-40-85°C	-40-85°C	-40-85°C	-40-85°C

Note: The above "/" represents a variety of situations, which need to be distinguished according to the specific chip part number.

Table 3-2. GD32F3x0 series and GD32E230xx series peripheral address comparison



Peripheral	BUS	GD32F3x0	GD32E230xx
GPIOF		0X48001400	0X48001400
GPIOD	GPIOD GPIOC AHB2 GPIOB GPIOA	0X48000C00	-
GPIOC		0X48000800	0x48000800
GPIOB		0X48000400	0X48000400
GPIOA		0X48000000	0X48000000
USBFS		0X5000000	-
TSI		0X40024400	-
CRC		0X40023000	0X40023000
FMC	AHB1	0X40022000	0X40022000
RCU		0X40021000	0X40021000
DMA		0X40020000	0X40020000
DBG		0xE0042000	0X40015800
TIMER16		0X40014800	0X40014800
TIMER15		0X40014400	0X40014400
TIMER14		0X40014000	0X40014000
USART0		0X40013800	0X40013800
SPI0/I2S0	APB2	0X40013000	0X40013000
TIMER0		0X40012C00	0X40012C00
ADC		0X40012400	0X40012400
EXTI		0X40010400	0X40010400
SYSCFG+CMP	-	0X40010000	0X40010000
CTC		0X4000C800	-
CEC		0X40007800	-
DAC		0X40007800	-
PMU		0X40007000	0X40007000
I2C1		0X40005800	0X40005800
I2C0		0X40005400	0X40005400
USART1		0X40004400	0X40004400
SPI1		0X40003800	0X40003800
FWDGT	APB1	0X40003000	0X40003000
WWDGT		0X40002C00	0X40002C00
RTC TIMER13 TIMER5		0X40002800	0X40002800
		0X40002000	0X40002000
		0X40001000	0X40001000
TIMER2	l F	0X40000400	0X40000400
TIMER1		0X4000000	-
SRAM		0x20000000	0x2000000
Option Byte		0x1FFFF800	0x1FFFF800
Main Flash		0x08000000	0x0800000
System Memory		0x1FFFEC00	0x1FFFEC00



AN046 Migration from GD32E230 series to GD32F3x0 series

OTP - 0x1FFF7000



4. Comparison of development tools

GD32F3x0 can be developed by using Keil4 and Keil5 of MDK for arm. It is recommended to install version 4.74 or above when using Keil4; Using Keil5, it is recommended to install version 5.20 or above. You can also use IAR for ARM development. It is recommended to install IAR 6.3 or above, As shown in <u>Table 4-1. Comparison of IDE environment between</u> <u>GD32F3x0 series and GD32E230xx series</u>.

Table 4-1. Comparison of IDE environment between GD32F3x0 series and GD32E230xx series

MCU series	GD32F3x0	GD32E230xx
KEIL	Both Keil4 and Keil5 support	KEIL 5.25 or above
IAR	IAR 6.3 or above	IAR 8.1 or above

GD32F3x0 can be developed with debugging tools such as J-LINK, ULINK and GD-LINK. As shown in <u>Table 4-2. Comparison of GD32F3x0 series and GD32E230xx series debugging</u> tools.

Table 4-2. Comparison of GD32F3x0 series and GD32E230xx series debugging tools

MCU series	GD32F3x0	GD32E230xx	
JLINK	JLINK ob, V8 and V9 all support	Only JLINK V9 and above	
ULINK	support	support	
GDLINK	support	support	



5. Software environment settings

5.1. Using Keil to develop GD32F3x0

At present, the common MDK for arm versions on the market include Keil4 and Keil5: it is recommended to install version 4.74 or above for Keil4 and version 5.20 or above for Keil5.

5.1.1. Add GD32F3x0 MCU device in Keil4

1. Download GD32F3x0 series plug-ins from gd32mcu website.

Figure 5-1. GD32F3x0 plug-in package details

gd32mcu.com/en/download/0?kw=GD32F3						
Full Data	GD32F3x0 Firmware Library Introduction: GD32F3x0 standard firmware library is suitable for GD32F3x0 ser	2.2.0 ies MCU. The libra	ary is compatib			
GD32L2 MCU	D32L2 MCU Software Interface Standard), and includes programs, data structures and mail the related drivers and sample programs					
GD32F1 MCU	GD32F3x0 AddOn	3.0.0	٩			
GD32F2 MCU	Introduction: Introduction: 1. GigaDevice.GD32F3x0 Addon.3.0.0.exe support Keil Version 4.74;					
GD32F3 MCU	2. GigaDevice.GD32F3x0_DFP.3.0.0.pack support Keil Version 5.26 above; 3. IAR_GD32F3x0_ADDON.3.0.0.exe support IAR Version 7.4 above.					
GD32F4 MCU						

2. Double click the installation file to install the plug-in to the directory of Keil4. Generally, it will be selected by default. If Keil4 and Keil5 are both installed, it needs to be selected manually.

Figure 5-2. Installation diagram of GD32F3x0 Series MCU plug-in package (Keil4)

Setup GigaDevice GD32F30x Device AddOn Package	to Keil MDK-ARM	Х
Folder Selection Select the folder where SETUP will install files.	ARM	M°KEIL° Itroller Tools
This Add-On will install into the following product folder.		
To install to this folder, press 'Next'. To install to a different fo folder.	lder, press 'Browse' and sele	ect another
Destination Folder		
C:\Keil		B <u>r</u> owse
— Keil MDK-ARM Setup —	<< Back Next >>	Cancel



3. After successful installation, reopen Keil4, and the drop-down option of "Database" can appear in "Options for Target ->Device". Click to view the GD32F3x0 part number.

Figure 5-3. Successful installation of GD32F3x0 Series MCU plug-in package (Keil4)

Options for Target 'GD3	2F330'	×
♥ Options for Target 'GD3 Device Target) Output] L Database: GigaDevice Device: GD32F330G4 Toolset: ARM □• GigaDevice □ GigaDevice □ GD32F330C4 □ GD32F330C6 □ GD32F330C6 □ GD32F330C6 □ GD32F330C6 □ GD32F330F8 □ GD32F330F8 □ GD32F330G8 □ GD32F330G8 □ GD32F330G8 □ GD32F330F8 □ GD32F330G8 □ GD32F330K4 □ GD32F330K6	2F330' isting User C/C++ Asm Linker Debug Utilities ce GD32F3x0 Devices Core: ARM 32-bit Cortex-M4 Microcontroller, 120MHz max) Memories: 16 Kbytes of Flash memory 4 Kbytes of SRAM with HW party checking 1 * general timer (32bit) 4 * general timer (32bit) 2 * watch dog timer RTC 1 * 12C 1 * 12C 1 * 2P 1 * ADC (10 channels) 	×
GD32E330R8	V < OK Cancel Defaults	> Help

4. For the smooth progress of subsequent debugging, it is recommended to check whether there is a download algorithm under the installation path. You can check it in the following way: open a project, select the device as GD32F3x0, and then "Options for Target -> Debug ->Settings -> Flash Download-> Add". If there is a flash download algorithm of GD32F3x0 in the drop-down option, the installation is successful.

Figure 5-4. GD32F3x0 series flash algorithm file selection diagram (Keil4)

Download F	unction − ○ Eras ○ Eras ○ Do r	e Full Chip	RAM for Algor Start: 0x20	ithm 000000 Size	e: 0x0800	
Programming	g Algorit _A	dd Flash Programming Alg	orithm		×	1
Descriptio	n					
		Description	Device Type	Device Size	A	
		GD32F30x High-density FMC	On-chip Flash	512k		
		GD32F30x Extra-density FMC	On-chip Flash	3M		
		GD32F3x0 FMC	On-chip Flash	128k		
		GD32F403 FMC	On-chip Flash	3M		
1	-	GD32F4xx_1MB FMC	On-chip Flash	1M		
		GD32F4XX_2MB FMC	On-chip Hash	ZM 2M		
		GD32F4XX_3IVIB FIVIC	On-chip Flash	31VI 5106		
		HT32 Series Flash	On-chip Flash	128		
		HT32 Series Flash Options	On-chip Flash	4k		
		K8P5615UQA Dual Flash	Ext. Flash 32-bit	64M		
		LM3Sxxx 128kB Flash	On-chip Flash	128k		
		LM3Sxxx 16kB Flash	On-chip Flash	16k		
		LM3Sxxx 256kB Flash	On-chip Flash	256k		
		LM3Sxxx 32kB Flash	On-chip Flash	32k		
		LM3Sxxx 384kB Flash	On-chip Flash	384k	¥	
		Add	Cancel	1		



5.1.2. Add GD32F3x0 MCU device in Keil5

1. Download GD32F3x0 series plug-ins from gd32mcu website.

Figure 5-5. GD32F3x0 plug-in package details

gd32mcu.con	n/en/download/0?kw=GD32F3		
Full Data	GD32F3x0 Firmware Library Introduction: GD32F3x0 standard firmware library is suitable for GD32F3x0 s	2.2.0 eries MCU. The libra	arv is compatib
GD32L2 MCU	Software Interface Standard), and includes programs, data structures and ma all the related drivers and sample programs	cro definitions. It cov	ers the feature
GD32F1 MCU	GD32F3x0 AddOn	3.0.0	٩
GD32F2 MCU	Introduction: Introduction: 1. GigaDevice.GD32F3x0 Addon.3.0.0.exe support Keil Version 4.74;		
GD32F3 MCU	 GigaDevice.GD32F3x0_DFP.3.0.0.pack support Keil Version 5.26 above; IAR_GD32F3x0_ADDON.3.0.0.exe support IAR Version 7.4 above. 		
GD32F4 MCU			

2. Extract and install it into the directory of Keil5.

Figure 5-6. Installation diagram of GD32F3x0 Series MCU plug-in package (Keil5)

Pack Unzip: GigaDevice GD32F30x_DFP 2.1.0			×
Welcome to Keil Pack Unzip Release 9/2020			
This program installs the Software Pack:			
GigaDevice GD32F30x_DFP 2.1.0 GigaDevice GD32F30x Series Device Support and Example	\$		
Destination Folder C:\Users\xian\AppData\Local\Arm\Packs\GigaDevice\C	D32F30x_DF	P\2.1.0	
Keil Pack Unzip			
Pack already installed. Click "Next" to replace.	<< Back	Next >>	Cancel

3. After installation, reopen Keil5 project, and you can find GD32F3x0 device in "Options for Target ->Device".



Figure 5-7. Successful installation of GD32F3x0 Series MCU plug-in package (Keil5)

Options for Target 'GD32F330'		\times
Device Target Output Listing User	C/C++ Asm Linker Debug Utilities	
Software Packs	*	
Vendor: GigaDevice	Software Pack	
Device: GD32F330C8	Pack: GigaDevice.GD32F3x0_DFP.2.0.0	
Toolset: ARM	URL: <u>http://gd32mcu.21ic.com/data/documer</u>	
Search:	GD32 is a new 32-bit high performance, low power consumption universal microcontroller family powered by the ARM Cortex-M4 RISC	^
GD32F3x0 Series GD32F3x0 Series GD32F330 GD32F330C4 GD32F330C6 GD32F330C6 GD32F330C8	core, which targeted at various MCU application areas. GD32 family integrates features to simplify system design and provide customers wide range of comprehensive and superior cost effective MCU potfolios with proven technology and great innovation. GD32 family includes entry line, performance line and connectivity line currently.	
- # GD32F330CB - # GD32F330F4 - # GD32F330F6	GD32F330 - ARM Cortex-M4 Core Frequency up to 84 MHz Flash access zero wait state Single-cycle multiplier and hardware divider	
	Memories	~
OK	Cancel Defaults Help	

 Add the flash algorithm in "Options for Target -> Debug ->Settings ->Flash Download", and the algorithm of GD32F3x0 will appear, which indicates that the installation is successful. Debug and download is now availabled.

Figure 5-8. GD32F3x0 series flash algorithm file selection diagram (Keil5)

Download F	O Erase Full Chip ♥ Pro C Erase Sectors ♥ Vei	ogram	RAM for	Algorithm	e: 0x1000	
_	Add Flash Programming Alg	gorithm			×	
Descript	Description GD32F3x0 FMC AM25tr28 Hash K8P5615UQA Dual Flash LPC18ax/43xx X25FU325F LPC18ax/43xx S25FU325 SP LPC18ax/43xx S25FU32 SP LPC407xx9k S25FU32 SP LPC5460x MT25QL128 SPIFI M29W640FB Flash MMXRT105x EcoXP Flash RC28F640J3x Dual Flash S25FL1032 V2C S29GL064N Dual Flash S23JL032H_BOT Flash S23JL032H_TOP Flash	Flash Size 128k 15M 64M 8M 4M 8M 4M 16M 8M 4M 16M 16M 16M 4M 4M	Device Type On-chip Flash Ext. Hash 15bit Ext. Rash 32bit Ext. Rash 32bit Ext. Rash 32bit Ext. Rash 5PI Ext. Rash 5PI Ext. Rash 5PI Ext. Rash 5PI Ext. Rash 5PI Ext. Rash 32bit Ext. Rash 32bit Ext. Rash 32bit Ext. Rash 16bit Ext. Rash 16bit	Odnin Device Family Packag MUK Core MDK Core	e	
		Adc	Cance	4		

5. Open Keil5 project file in Keil4 environment

If Keil5 environment is not installed, Keil4 environment can also be used to compile Keil5 project files. The method is to modify the suffix of the project file, and change the suffix



"xxxx. uvprojx" of Keil5 project file to "xxxx. uvproj", then Keil4 can be used for development.

6. Open Keil4 project file in Keil5 environment

If you use Keil5 environment to open the Keil4 project file, There will be no MCU devices found. In this case, you can directly modify the suffix of the Keil4 project file "xxxx.uvproj" to "xxxx.uvprojx", then Keil5 can be used for development.

5.2. Debugging and simulating GD32F3x0 with GD-Link

Debugging and simulating GD MCU with GD-Link, The hardware needs to be connected to the development board with GD-Link tool, and the specific configuration of the IDE is as follows.

 Open a GD32F3x0 project and select "CMSIS-DAP Debugger" in "Options for Target -> Debug". Some customers reported that this drive option could not be found because the MDK version is too low and only Keil4 The "CMSIS-DAP Debugger" option is only supported for versions above 4.74 and Keil5.

Figure 5-9. Select the "CMSIS-DAP Debugger" option in the Debug interface (Keil4)

😗 Options for Target 'GD32F330'	×
Device Target Output Listing User C/C++ A	Asm Linker Debug Utilities
C Use Simulator Settings	
Load Application at Startup Run to main() Initialization File:	Coad Application at Startup Run to main() Initialization File:
Restore Debug Session Settings	Restore Debug Session Settings
CPU DLL: Parameter: SARMCM3.DLL -REMAP	Driver DLL: Parameter: SARMCM3.DLL
Dialog DLL: Parameter: DCM.DLL PCM3	Dialog DLL: Parameter: TCM.DLL PCM3
OK Car	Local Defaults Help

2. In "Options for Target > Utilities", we also have to choose "CMSIS-DAP Debugger" option.



Figure 5-10. Select the "CMSIS-DAP Debugger" option in the Utilities interface (Keil4)

W Options for Target 'GD32F330'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Configure Flash Menu Command
C Use Target Driver for Flash Programming
CMSIS-DAP Debugger Settings Update Target before Debugging
Init File: Edit
C Use External Tool for Flash Programming
Command:
Arguments:
E Run Independent
OK Cancel Defaults Help

 In the "Options for Target > Debug ->Settings" check SWJ and Port select SW. "0xXBAXXXX" will appear in the idcode in the right box, indicates that the target MCU device is successfully connected.

Figure 5-11. GD-Link tool successfully connected to the target MCU(Keil4)

😨 Options for Target 'GD32F330'		×
Cortex-M Target Driver Setup		×
Debug Flash Download		
CMSIS-DAP - JTAG/SW Adapter CMSIS-DAP	SW Device	Move Up Down
Max Clock: 1MHz	Automatic Detection ID CODE: Automatic Detection Device Name: Add Delete Update	AP: 0x00
Debug Connect & Reset Options Connect: Normal Rese Reset after Connect	t: Autodetect Cache Options Download Og Verify Co Verify Co Download Download	otions de Download d to Flash
-	OK Cancel	Help
	DK Cancel Defaults	Help

 Add the flash algorithm of GD32F3x0 in "Options for Target -> Debug ->Settings -> Flash Download".



Figure 5-12. Schematic diagram of adding flash algorithm file(Keil4)

😨 Options for Target 'GD32F3:	30'			×
Cortex-IVI Target Driver Setup				×
Debug Flash Download				
Download Function C Erase Full Chip C Erase Sectors C Do not Erase	 ✓ Program ✓ Verify ✓ Reset and Run 	RAM for A Start: 0	vgorithm x20000000 Size: 0x0800	
Programming Algorithm				
Description GD32F3x0 FMC	Device Type On-chip Flash	Device Size 128k	Address Range 08000000H - 0807FFFFH	
		Start:	Size:	_
	Add	Remove		
	ОК	Cance	el	Help
	ОК	Cancel	Defaults	Help

 Click the shortcut in the red box in <u>Figure 5-13. Schematic diagram of GD-Link</u> <u>simulation(Keil4)</u> to start debugging, and you can use GD-Link for simulation.

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help 🗋 🚰 🛃 🐉 🕹 🛍 🖄 🗢 🖭 🦛 🚽 🥐 🎘 🎘 🎘 🎼 🎘 🕍 word 🖃 🗟 🦸 0 🔗 🚓 🔲 🔍 🗣 🔝 Disassembly egisters 54: 0x080004A8 Register V rcu_configuration(); F000F832 BL.W rcu Core led_config(); 55: 56: main.c startup_gd32f1x0.s R4 0. 51 ^L*/ 52 in R5 R6 R7 R8 R9 R10 R11 0. 0. int main (void) 0. 0. 0. 0. 53 🖂 { > 54 • 55 rcu_configuration(); led_config(); 56 57 /* setup SysTick Timer for 1ms interrupts */ R12 R13 (SP) 0. 0. 58 59 systick_config(); 60 白 61 Banked System Internal if(0 == gpio_input_bit_get(GPIOA, GPIO_PIN_0)){
 /* delay 100ms */ 62 🗄 63 delay_ims(100);
/* LED1 on*/
gpio_bit_set(GPIOA, GPIO_PIN_1); Т. Р. М. 64 Mode Privilege 65 66 Stack States Sec 6 0 67 68 delay_1ms(35);
/* LED1 off*/ gpio_bit_reset(GPIOA, GPIO_PIN_1); delay_1ms(35); 69 70 71 }else{ 🔃 Project 🛛 🚟 Regist

Figure 5-13. Schematic diagram of GD-Link simulation(Keil4)

5.3. Debugging and simulating GD32F3x0 with J-Link

Debugging and simulating GD MCU with J-Link, The hardware needs to be connected to the



development board with J-Link tool, and the specific configuration of the IDE is as follows:

 Open a GD32F3x0 project file and select "J-LINK/J-Trace Corte " in "Options for Target -> Debug".

Figure 5-14. Select the "J-LINK/J-Trace Cortex" option in the Debug interface (Keil4)

🔣 Options for Target 'GD32F330'	×
Device Target Output Listing User C/C++	Asm Linker Debug Utilities
C Use <u>Simulator</u> Settings ☐ Limit Speed to Real-Time	
Iversity Initialization File:	✓ Load Application at Startup ✓ Run to main() Initialization File:
Edit	Edit
Restore Debug Session Settings	Restore Debug Session Settings
Image: Seakpoints Image: Toolbox Image: Watch Windows & Performance Analyzer	Image: weak points Image:

2. In "Options for Target > Utilities", we also have to choose "J-LINK/J-Trace Cortex" option.

Figure 5-15. Select the "J-LINK/J-Trace Cortex" option in the Utilities interface (Keil4)

🕎 Options for Target 'GD32F330'	×
Device Target Output Listing User C/C++ Asm Linker Debug Utilities	
Configure Flash Menu Command	
Use Target Driver for Flash Programming	
J-LINK / J-Trace Cortex Settings Vpdate Target before Debugging	
Init File: Edit	

 In the "Options for Target > Debug ->Settings" Port select SW. "0xXBAXXXXX" will appear in the idcode in the right box, indicates that the target MCU device is successfully connected.

Figure 5-16. J-Link tool successfully connected to the target MCU(Keil4)

Cortex JLink/JTrace Target Driver Setup		×
Debug Trace Flash Download		
J-Link / J-Trace Adapter	SW Device	
SN: 4294967295	IDCODE Device Name	Move
Device: J-Link	SWD Occ2BA01477 ARM CoreSight SW-DP	Up
HW : V9.20 dll : V5.02c		Down
FW: J-Link V9 compiled Sep 1 20	Automatic Detection ID CODE:	_
SW 💌 10MHz 💌	C Manual Configuration Device Name:	
Auto Clk	Add Delete Update IR len:	

 Add the flash algorithm of GD32F3x0 in "Options for Target -> Debug ->Settings -> Flash Download".



Figure 5-17. Schematic diagram of adding flash algorithm file(Keil4)

Programming Algorithm Description Device Type Device Size Address Range GD32F3x0 FMC On-chip Flash 128k 08000000H - 0807FFFFH Start: Size: Add Remove	Download Function C Erase Full Chip C Erase Sectors C Do not Erase	 ✓ Program ✓ Verify ✓ Reset and Run 	RAM for A	Ngonithm 0x20000000 Size: 0x0800	
Start: Size: Size:	Programming Algorithm Description GD32F3x0 FMC	Device Type On-chip Flash	Device Size 128k	Address Range 08000000H - 0807FFFFH	
Add Remove			Start:	Size:	
			1	,	
		Add	Remove		
		Add	Remove		

 Click the shortcut in the red box in <u>Figure 5-18. Schematic diagram of J-Link</u> <u>simulation(Keil4)</u> to start debugging, and you can use J-Link for simulation.

Figure 5-18. Schematic diagram of J-Link simulation(Keil4)

File Edit View P	rojec	t Flash	Debug Peripherals Tools SVCS Window Help
🗋 💕 🖬 🗿	<u>ж</u> 1	6 🛍 🖂	이 안 석 → 隆 豫 豫 華 岸 <i>旧</i> 版 🦉 word 🕢 🗟 ≰ 🗐 🔍 🔍 🔗 🌒 🖬 🗨
🏦 🖹 🕄 🕅	{} •	{} * * {}	
Registers 🛛 🗜		Disassemb	ly
Register	v	5.	4: rcu_configuration();
Core	-	0x080	004A8 F000F832 BL.W rcu_configuration (0x08000510)
N	0.	5	5: led_config();
R1	0.	5	б:
R2	0.	•	
R3	0.	(+) m	haine the startup addotts of a
R4	0.		ante in startep_gozitios
RD pg	U.	51	*/
ND 	0.	52	int main (void)
RB	0.	53 🖯	{
R9	0.	54	rcu_configuration();
R10	0.	0 55	<pre>led_config();</pre>
R1 1	0.	56	
R12	0.	57	/* setup SysTick Timer for 1ms interrupts */
R13 (SP)	0.	58	<pre>systick_config();</pre>
R15 (PC)		59	
* xPSR	ŏ.	60 🖯	while(1){
🛨 🛛 Banked		61	/* PAO:Keyl is pressed */
🗄 System		62 🖯	<pre>if(0 == gpio_input_bit_get(GPIOA, GPIO_PIN_0)){</pre>
Internal		63	/* delay 100ms */
Mode	T.	64	delay_1ms(100);
frivilege Stock	P.	65	/* LED1 on*/
States	6	66	<pre>gpio_bit_set(GPIOA, GPI0_PIN_1);</pre>
Sec	0.	67	delay_1ms(35);
		68	/* LED1 off*/
		69	<pre>gpio_bit_reset(GPIOA, GPIO_PIN_1);</pre>
		70	delay_1ms(35);
	1	71	}else{
🖭 Project 🛛 🚟 Regist	ers	1	11

5.4. Using IAR to develop GD32F3x0

There are many versions of IAR, and the compatibility between versions is not good. If you use it for the first time, it is recommended to install versions above 7.3. After installing IAR, add the device of GD according to this document for debugging



5.4.1 Add gd32F3x0 MCU device in IAR

- 1. Download GD32F3x0 series plug-ins: IAR_GD32F3x0_ADDON_2.0.0.exe.
- 2. Run IAR_ GD32F3x0_ ADDON_ 2.0.0. exe, click start to start installing the plug-in.

Figure 5-19. Installation diagram of GD32F3x0 Series MCU plug-in package(IAR)

a Setup GigaDevice GD32F3x0 Device AddOn Package to IAR v2.0.0	×
This SETUP program installs:	7
GigaDevice GD32F3x0 Device AddOn Package to IAR	
This AddOn will install into the following product folder.	
To install to this folder, press 'Start'. To Install to a different folder, press 'Browse' and select another folder	r.
C:\Program Files (v86)\14R Systems\Emhedded Workhench & 3	
Realtime Status	
0%	-
Start Cancel	

3. After the installation is successful, click Finish to end the plug-in installation.

Figure 5-20. Successful installation of GD32F3x0 Series MCU plug-in package (IAR)

羄 Setup GigaDevice GD32F3x0 Device AddOn Package to IAR v2.0.0	Х
- This SETUP program installs:	
GigaDevice GD32F3x0 Device AddOn Package to IAR	
This AddOn will install into the following product folder.	
To install to this folder, press 'Start'. To Install to a different folder, press 'Browse' and select another fold	er.
C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.3 Browse	
Realtime Status	
100%	
IAR Setup has performed all requested operations successfully Finish Cancel	

5.4.2 Debugging GD32F3x0 in IAR

In the previous section, we have added the plug-in of GD32F3x0 series. In this section, we will introduce how to use it.

 There are two ways to use IAR to compile GD MCU. One is to use the existing project for modification, and the other is to re-establish the project. Here, we will not introduced how to establish the project. The project establishment of GD is consistent with that of other



platforms.



Options for node "Project"					>
Options for node "Project" Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver	Library Target Processo O Core Devic O CMSI Endian n © Little Big	Options 2 Output or variant e G IS-Pack	MISRA Library Co ortex-M4 D GD32F330x8 lone Floating point FPU D registers	-C:2004 Infiguration	MISRA-C:1998 Library Options 1
PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	© Little	32	FPU D registers	VFPv4 single	e precision 🗸
	● BE	xtension	NEON)	☐ TrustZa Mode Se	one cure v
L			[OK	Cancel

 IAR after version 6.1 does not need to add CMSIS files (core_cm3.c and core_cm3.h), but you need to check use CMSIS in "General Options->Library Configuration". If the software code uses printf function, you also need to modify the"Library" to"FULL".



|--|

egory: eral Options tic Analysis			
time Checking C++ Compiler	Ontions 2	MISRA-C·2004	MISRA-C-1998
sembler Target	Output	Library Configuration	Library Ontions 1
Itput Converter	Output		Library options i
ild Actions		Description:	
ker Full	~	Use the full configuration	of the C/C++
bugger		runtime library. Full local	e interface, C locale,
imulator		file descriptor support, n	nultibytes in printf
ADI		and scanf, and hex floats	in strtod.
DB Server			
-jet Configurati	on file:		
-Link/J-Trace \$TOOLKIT	DIR\$\inc\c\l	DLib_Config_Full.h	
I Stellaris			
Emicro Enable th	nread suppo	ort in library	
T-LINK Library lov	w-level inter	face implementation	CMSIS
hird-Party Driver			
I MSP-FET		Studutystaen	Use CMSIS
I XDS (Semihos	sted	Via semihosting	DSP library
O IAR brea	akpoint	○ Via SWO	

 The Link file of the chip will be selected by default according to the device when establishing the project, but you should still have the habit of checking before compiling. Check whether the ICF file is configured and correct.

Figure 5-23. Add ICF file in IAR "Options" interface

Options for node "Project"							×
Category: General Options						Factory	Settings
Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build	#define Config	Diag Library	nostics Input	Checksum Optimizations	Encodings Advanced	Extra C Output	Dptions List
Build Actions Linker Debugger Simulator CADI	⊡ Ov \$	erride de	fault DIR\$\cc	onfig\linker\GD\@	6D32F330x8.ic	f	
CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris	Configu	Edit ration file	e symbol	definitions: (one	per line)		~

4. Configure the "Debugger->Setup" option. The newly created project is simulator option by default. If debugging is required, you need to choose according to the actual situation: use GD-Link to select CMSIS DAP (poor compatibility, not recommended under IAR) or J-Link to select J-Link/J-Trace.



Figure 5-24. Select the debugger tool in the IAR "Options" interface

Category:	Fi	actory Settings
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler	Setup Download Images Extra Options Multicore Plugins	2009 22003
Custom Build Build Actions	Driver V Run to	
Cablugger Smulator CADI CADI CBS Server L-jet J-ink()-Trace TI Stellaris Nu-Link PE-mico ST-LINK Tirtd-Party Driver Tirtd-Party Driver Tirtd-Party Driver	Simulator CADI CMSIS DAP GDB Server L-jet I-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI MSP S Stdebugger/GD/GD32F330x8.ddf	

5. Configure the "Debugger->Download" option. The new project may not be configured with the download option. If we need to debug the code, we must check the "User flash loader" option and ensure that the "board file" is accurate, otherwise the program cannot be downloaded to the chip normally.

Figure 5-25. Configure flash loader in IAR "Options" interface



6. Steps of GD32E23x firmware library adapting to GD32F3x0 Series MCU

This chapter will use GD32E23x_Firmware_Library_V1.1.1 take the project in the template as an example to introduce how to adapt GD32F3x0 series MCU.

1. Open Keil project.

Figure 6-1. Open GD32E23x Keil project

GD32E23x_Firmware_Library_V1.1.1 > Template >					
^	Name	Date modified	Туре		
	IAR_project	3/10/2022 10:33 AM	File folder		
	Keil_project	3/10/2022 10:11 PM	File folder		
	🗐 gd32e23x_it.c	5/18/2021 2:21 PM	C File		
	📗 gd32e23x_it.h	3/10/2022 1:18 PM	H File		
	🗐 gd32e23x_libopt.h	5/18/2021 2:21 PM	H File		
	🗐 gd32f3x0_libopt.h	1/6/2022 4:58 PM	H File		
	🧾 main.c	3/10/2022 1:19 PM	C File		
	🧾 main.h	5/18/2021 2:21 PM	H File		
	readme.txt	5/18/2021 2:21 PM	Text Document		
	systick.c	3/10/2022 1:26 PM	C File		
	🥘 systick.h	5/18/2021 2:21 PM	H File		

2. After opening the project, "Options for Target -> Device", select GD32F3x0 MCU part number.

D:\Desktop\G	B:\Desktop\GD32E23x_Firmware_Library_V1.1.1\Template\Keil_project\Project.uvprojx - µVision						
File Edit View	Project Flash Debug Peripherals Tools SVCS Window Help						
🗋 💕 🛃 🕔	🐰 ங 🏡 🔊 🤨 ሩ 🔿 🥐 隆 隆 隆 隆 隆 🕸 🎼 🎼 🎼 🖉 LIS2DW12_SAD_R1 👘 😡 🧖						
• 🗳 🛍 🛍 🍣 •	📖 🙀 GD32E23x 🛛 🗸 着 着 💠 🗇 🏙						
Project	I Options for Target 'GD32E23x'						
E Sproject: Project:	CC Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities						
	Software Packs						
÷) s	Vendor: GigaDevice Software Pack						
🕀 🧰 Perip	he Device: GD32F330F8 Pack: GigaDevice.GD32F3x0_DFP.2.0.0						
🗉 🚞 Startı	IP Toolset: ARM URL: <u>http://gd32mcu.21ic.com/data/documer</u>						
🕀 🧰 Utiliti	es Search:						
🗉 🛅 Doc							
S CMS	S GD32F330C6 ← GD32F330C6 ← GD32F330C6 ← GD32F330C8 GD32F330C8 ← GD32F330C8 ← GD32F330C8 ← GD32F330F4 ← GD32F330F4 ← GD32F330F6 ← GD32F330F6 ← GD32F330F6 ← GD32F330G6 ← GD32F330G6 ← GD32F330G6 ← GD32F330G6 ← GD32F330G6 ← GD32F330G8 ← ← Frequency up to 84 MHz ← Fash access zero wait state Single-cycle multiplier and hardware divider ← Memories ← Memo						
Proj Books	0K Cancel Defaults Help						

Figure 6-2. Select GD32F3x0 device in GD32E23x project

 Add the flash algorithm of GD32F3x0 in "Options for Target -> Debug ->Settings -> Flash Download".



Figure 6-3. Add the flash algorithm of GD32F3x0

Options for Target 'GD32E	23x'	×
CMSIS-DAP Cortex-M Target I	Driver Setup	×
Debug Trace Flash Downlo	ad	
Download Function C Erase Full Chip C Erase Sectors C Do not Erase Programming Algorithm	Iv Program Iv Verify Iv Reset and Run	
Description	Device Size Device Type Address Bange	
GD32F3KU FMC		
	Start: 0x08000000 Size: 0x00020000	
	Add Remove	
	OK Cancel	lelp

4. Copy Cortex M4 kernel files to:

x:\GD32E23x_Firmware_Library_V1.1.1\Firmware\CMSIS.

Figure 6-4. Add Cortex M4 kernel files to GD32E23x firmware library

SD32E23x_Firmware_Library_V1.1.1 → Firmware → CMSIS v 0 Search C						
^	Name	Date modified	Туре	Size		
	GD	3/10/2022 10:33 AM	File folder			
	📃 core_cm4.h	1/6/2022 4:58 PM	H File	108 KB		
	core_cm4_simd.h	1/6/2022 4:58 PM	H File	23 KB		
	core_cmFunc.h	1/6/2022 4:58 PM	H File	16 KB		
	core_cmlnstr.h	1/6/2022 4:58 PM	H File	17 KB		

5. Modify the contents of the " gd32e23x.h" in GD32E23x firmware library.

Figure 6-5. Modify the contents of "gd32e23x.h"

/* includes */ #include "core cm4 #include "system_g #include <stdint.h< th=""><th>.h" d32e23x.h"</th><th>/* includes */ #include "core cm #include "system_ #include <stdint.< th=""><th>23.h" gd32e23x.h h></th><th>."</th></stdint.<></th></stdint.h<>	.h" d32e23x.h"	/* includes */ #include "core cm #include "system_ #include <stdint.< th=""><th>23.h" gd32e23x.h h></th><th>."</th></stdint.<>	23.h" gd32e23x.h h>	."
/* configuration of the Cortex-M4 fdefine _CM4_REV fdefine _MUD_PRESENT fdefine _NVIC_PRIO_BITS fdefine _Vendor_SysTickConfig fdefine _FPU_PRESENT	processor and core periphe 0x0001 /*!< Core revisic 0U /*!< GD32F3x0 do 4U /*!< GD32F3x0 use 0U /*!< GD32F3x0 use 1U /*!< FPU present	<pre>/* configuration of the Cortex-M #defineCM23_REV #defineSAUREGION_PRESENT #defineMPU_PRESENT #defineVTOR_PRESENT #defineVTOR_PRESENT #defineVTOR_PRESENT</pre>	23 processor and co 0x0100U /*!< Co 0U /*!< SI 0U /*!< Mi 1U /*!< Ni 2U /*!< Ni 0U /*!< SI	ore pe ore re AU reç PU is FOR is umber et to

Table 6-1. Modify the contents of "gd32e23x.h"

After modification		Before modification	
#include "core_cm23.h"		#include "core_cm23.h"	
#defineCM4_REV	0x0001	#defineCM23_REV	0x0100U



AN046 Migration from GD32E230 series to GD32F3x0 series

#defineMPU_PRESENT	0U	#defineSAUREGION_PRESENT	0U
#defineNVIC_PRIO_BITS	4U	#defineMPU_PRESENT	0U
#defineVendor_SysTickConfig	0U	#defineVTOR_PRESENT	1U
#defineFPU_PRESENT	1U	#defineNVIC_PRIO_BITS	2U
		#defineVendor_SysTickConfig	0U

6. GD32E230xx does not support interrupt grouping, so there is no "void nvic_priority_group_set (uint32_t nvic_prigroup)" function in the firmware library. We need to add corresponding content in the firmware library.

Table 6-2. Modify the contents of "gd32e23x_misc.h"

/* priority group - define the pre-emption prior	ority and the subpriority */
#define NVIC_PRIGROUP_PRE0_SUB4	((uint32_t)0x0000700)
#define NVIC_PRIGROUP_PRE1_SUB3	((uint32_t)0x0000600)
#define NVIC_PRIGROUP_PRE2_SUB2	((uint32_t)0x0000500)
#define NVIC_PRIGROUP_PRE3_SUB1	((uint32_t)0x0000400)
#define NVIC_PRIGROUP_PRE4_SUB0	((uint32_t)0x0000300)
/* set the priority group */	

void nvic_priority_group_set(uint32_t nvic_prigroup);

Table 6-3. Modify the contents of "gd32e23x_misc.c"

void nvic_priority_group_set(uint32_t nvic_prigroup)

```
/* set the priority group value */
```

SCB->AIRCR = NVIC_AIRCR_VECTKEY_MASK | nvic_prigroup;

 GD32E230xx only supports level 4 priority, not sub priority. GD32F3x0 supports both priority and sub priority. The corresponding contents need to be modified in the firmware library.

Table 6-4. Modify the contents of "gd32e23x_misc.h"

/* enable NVIC request */
void nvic_irq_enable(uint8_t nvic_irq, uint8_t nvic_irq_pre_priority, uint8_t
nvic_irq_sub_priority);

Table 6-5. Modify the contents of "gd32e23x_misc.c"



temp_pre = 0U; temp_sub = 0x4U; break; case NVIC_PRIGROUP_PRE1_SUB3: temp_pre = 1U; temp_sub = 0x3U; break: case NVIC_PRIGROUP_PRE2_SUB2: temp_pre = 2U; temp_sub = 0x2U; break: case NVIC_PRIGROUP_PRE3_SUB1: temp_pre = 3U; temp_sub = 0x1U; break; case NVIC_PRIGROUP_PRE4_SUB0: temp_pre = 4U; temp_sub = 0x0U; break; default: nvic_priority_group_set(NVIC_PRIGROUP_PRE2_SUB2); temp_pre = 2U; temp_sub = 0x2U; break; } /* get the temp_priority to fill the NVIC->IP register */ temp_priority = (uint32_t)nvic_irq_pre_priority << (0x4U - temp_pre);</pre> temp_priority |= nvic_irq_sub_priority &(0x0FU >> (0x4U - temp_sub)); temp_priority = temp_priority << 0x04U; NVIC->IP[nvic_irq] = (uint8_t)temp_priority; /* enable the selected IRQ */ NVIC->ISER[nvic_irq >> 0x05U] = (uint32_t)0x01U << (nvic_irq & (uint8_t)0x1FU);

8. The flash of GD32F3x0 is zero waiting. GD32E230xx series needs to configure the waiting cycle, so the function of waiting cycle can be removed

Table 6-6. Remove the function of waiting period in GD32E23x project

FMC_WS = (FMC_WS & (~FMC_WS_WSCNT)) | WS_WSCNT_2;

9. The flash of GD32E230xx supports 32-bit and 64-bit programming, and the flash of GD32F3x0 supports 32-bit word and half word programming. If 64-bit programming is used in the application code, it needs to be modified to 32-bit word or half word programming, and half word programming needs to be added to the GD32E230xx firmware library



Table 6-7. Add half word programming to "gd32e23x_fmc.h" of GD32E23x project

/* FMC program a half word at the corresponding address */

fmc_state_enum fmc_halfword_program(uint32_t address, uint16_t data);

Table 6-8. Add half word programming to "gd32e23x_fmc.c" of GD32E23x project

```
fmc_state_enum fmc_halfword_program(uint32_t address, uint16_t data)
{
    fmc_state_enum fmc_state = fmc_ready_wait(FMC_TIMEOUT_COUNT);

    if(FMC_READY == fmc_state){
        /* set the PG bit to start program */
        FMC_CTL |= FMC_CTL_PG;
        REG16(address) = data;
        /* wait for the FMC ready */
        fmc_state = fmc_ready_wait(FMC_TIMEOUT_COUNT);
        /* reset the PG bit */
        FMC_CTL &= ~FMC_CTL_PG;
    }
    /* return the FMC state */
    return fmc_state;
}
```

- 10. If TIMER5 is used in the project, because GD32F3x0 remove this TIMER5(Except GD32F350xx), the code of TIMER5 needs to be changed to other timer.
- 11. Compile GD32E23x project, so far, you can use the modified GD32E23x firmware library for software development in GD32F3x0 series MCU.



7. Steps to replace GD32E23x project library with GD32F3x0 Library

This chapter will use the projects in "GD32E23x_Firmware_Library_V1.1.1\Template" and "GD32F3x0 Firmware Library V2.2.0\Template" as examples.

1. Copy the files in "GD32F3x0_Firmware_Library_V2.2.0\Firmware\CMSIS" to the "GD32E23x_Firmware_Library_V1.1.1\Firmware\CMSIS" folder.

Figure 7-1. Copy h file in CMSIS of GD32F3x0 firmware library to GD32E23x

GD32F3x0_Firmware_Libra	ry_V2.2.0 > Firmware > 0	CMSIS	GD32E23x_Firmware_Lib	orary_V1.1.1 > Firmware > CMSIS >
Name	Date modified	Туре	Name	Date modified
GD	3/10/2022 10:32 AM	File folder	GD	3/10/2022 10:33 AM F
//// core_cm4.h	1/6/2022 4:58 PM	H File		
core_cm4_simd.h	1/6/2022 4:58 PM	LL EX		
core_cmFunc.h	1/6/2022 4:58 PM	H File		
core_cmInstr.h	1/6/2022 4:58 PM	H File		

 Copy the Iclude and Source folders in "GD32F3x0_Firmware_Library_ V2.2.0\Firmware\CMSIS\GD\GD32F3x0" and replace them to the "GD32E23x_Firmware_Library_V1.1.1\Firmware\CMSIS\GD\GD32E23x" folder.

Figure 7-2. Copy and replace the Include and Source files in CMSIS under GD32F3x0 firmware library into GD32E23x firmware library

GD32F3x0_Firmware	_Library_V2.2.0 > Firmware >	$CMSIS \rightarrow GD \rightarrow GD$	32F3x0 →	GD32E23x_Firmware	_Library_V1.1.1 → Fi	rmware >	CMSIS > GD > GD	32E23x
Name	^	Date modified	Туре	Name	^		Date modified	Туре
Include		3/10/2022 10:32 AM	File folder	Include			3/10/2022 11:00 AM	File folde
Source		3/10/2022 10:32 AM	File folder	Source			3/10/2022 11:00 AM	File folde

3. Copy the Iclude and Source folders in

"GD32F3x0_Firmware_Library_V2.2.0\Firmware\GD32F3x0_standard_peripheral" and replace them to the

"GD32E23x_Firmware_Library_V1.1.1\Firmware\GD32E23x_standard_peripheral" folder.

Figure 7-3. Copy and replace the Include and Source files in standard_peripheral under GD32F3x0 firmware library into GD32E23x firmware library

GD32F3x0_Firm	ware_Library_V2.2.0 > Firmware >	GD32F3x0_standard_p	peripheral	GD32E23x_Firmware	Library_V1.1.1 > Firmware	> GD32E23x_standard_	peripheral
Name	^	Date modified	Туре	Name	^	Date modified	Туре
Include		2/10/2022 10:22 111	PH 4 11	🔶 📙 Include		3/10/2022 12:25 PM	File folder
Source		3/10/2022 10:32 AM	File folde	Source		3/10/2022 12:25 PM	File folder

4. Copy the "gd32f3x0_libopt.h" file in "GD32F3x0_Firmware_Library_V2.2.0\Template" into "GD32E23x_Firmware_Library_V1.1.1\Template".

Figure 7-4. Copy the "gd32f3x0_libopt.h" file in GD32F3x0 firmware library into



GD32E23x firmware library

GD32F3x0_Firmware_Library_V2.2.0 > Template		GD32E23x_Firmware_Library_V1.1.1 > Template >	
^ Name	Date modified	Name	Date modified
IAR_project	3/10/2022 10:32 AM	IAR_project	3/10/2022 10:33 AM
Keil_project	3/10/2022 10:32 AM	Keil_project	3/10/2022 10:11 PM
gd32f3x0_it.c	1/6/2022 4:58 PM	gd32e23x_it.c	5/18/2021 2:21 PM
gd32f3x0_it.h	1/6/2022 4:58 PM	//////////////////////////////////////	3/10/2022 1:18 PM
gd32f3x0_libopt.h	1/6/2022 4:58 PM	gd32e23x_libopt.h	5/18/2021 2:21 PM
main.c	1/6/2022 4:58 PM	🔶 🧐 gd32f3x0_libopt.h	1/6/2022 4:58 PM
🧾 main.h	1/6/2022 4:58 PM	🥮 main.c	3/10/2022 1:19 PM
readme.txt	1/6/2022 4:58 PM	🥮 main.h	5/18/2021 2:21 PM
systick.c	1/6/2022 4:58 PM	readme.txt	5/18/2021 2:21 PM
systick.h	1/6/2022 4:58 PM	systick.c	3/10/2022 1:26 PM
		systick.h	5/18/2021 2:21 PM

5. Open the Keil project under the template file in the GD32E23x firmware library.

Figure 7-5. Open the Keil project under the template file in the GD32E23x firmware library

GD32E23x_Firmware_Libra	ary_V1.1.1 > Template >	Keil_project >
Name	^	Date modified
📙 list		3/10/2022 1:14 PM
output		3/10/2022 1:27 PM
RTE		3/10/2022 11:26 AM
🔣 Project.uvprojx		3/10/2022 1:26 PM

6. A yellow triangle mark on the left side of the engineering interface indicates that the original file no longer exists because the old file has been replaced in the previous file replacement steps. At this time, you only need to remove all the files marked in yellow. Among them, "gd32e230c_eval. c" is the supporting configuration of the development board. If it is not used in the actual project, it can be transplanted, and then add the corresponding GD32F3x0 files.



Project	д 🗙	Project 🛛 🗘 🔀
🔅 🚞 Application	•	system_gd32f3x0.c
🖭 🥅 CMSIS		🖃 🦢 Peripherals
Peripherals		gd32f3x0_adc.c
⊕ 1 gd32e23x ad	c.c	gd32f3x0_cec.c
ad32e23x cm	ip.c	gd32f3x0_cmp.c
ad32e23x crc		gd32f3x0_crc.c
ad32e23y db		gd32f3x0_ctc.c
ad32e23v da	gic .	gd32f3x0_dac.c
gd32e23x_un	i.c.	gd32f3x0_dbg.c
guszezsz_ext	inc inc	gd32f3x0_dma.c
⊕ 1 gds2e2sx_fm	c.c	gd32f3x0_exti.c
	ugt.c	gds2f3x0_fmc.c
gd32e23x_gpt	10.C	
⊞ <u>∎</u> gd32e23x_i2c	.c	gus215x0_gpio.c
	sc.c	gds2f3x0_tec.c
🕀 🔟 gd32e23x_pm	nu.c	ad32f3v0_msc.c
⊞ 1 gd32e23x_rcu	i.c	d32f3x0_pmu.c
⊕ 🚺 gd32e23x_rtc	.c	ad32f3x0_ttc.c
🕀 👔 gd32e23x_spi	.c	
⊕ <u>1</u> gd32e23x_sys	cfg.c	ad32f3x0_syscfa.c
🕀 👔 gd32e23x_tim	ner.c	gd32f3x0 timer.c
🖽 📶 gd32e23x_usa	art.c	gd32f3x0 tsi.c
⊕ 👔 gd32e23x_ww	vdgt.c	gd32f3x0_usart.c
😑 🦾 Startup		gd32f3x0_wwdgt.c
startup_gd32	e23x.s	🖃 🦢 Startup
🖃 🗁 Utilities		startup_gd32f3x0.s
gd32e230c er	val.c	🖃 🗁 Utilities
	22 Wang Wei - 123 - 124 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 129 - 1	<u>↓</u>

Figure 7-6. Remove the Yellow marked files and add new files

 Modify the "#include "gd32e23x.h" "statement contained in the" main. c" and "systick. c" files in the project to "#include "gd32f3x0.h"" statement, and delete the "#include"gd32e230c_eval. h"" statement. Then reselect the MCU device and flash algorithm.

Figure 7-7. Modify the contents of "main.c", "systick.c" files





Figure 7-8. Reselect GD32F3x0 MCU device

🕅 Options for Target 'GD32E23x'	×
Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities	
Software Packs	
Pavine: GD2222000 Pack: GigaDevice.GD32F3x0 DFP.3.0.0	
Toolset: ARM URL: http://gd32mcu.com/data/documents/pg	
Search:	
GD32F330 GD32F330 GD32F330C4 GD32F330C4 GD32F330C6 GD32F330C6 GD32F330C6 GD32F330C6 GD32F330C8 GD32F	^
GD32F330F4 GD32F330F6 GD32F330F6 GD32F330F8 GD32F330F8 GD32F330F8 GD32F330- ARM Cortex-M4 Core Frequency up to 84 MHz Flash access zero wait state Single-cycle multiplier and hardware divider	
▲ Memories	~
OK Cancel Defaults Help	

Figure 7-9. Reselect GD32F3x0 Flash algorithm

🕅 Options for Target 'GD32E23x'	1		\times				
Device Target Output Listing User C/C++ (ACG) Asm Linker Debug Utilities 2							
C Use Simulator with restrictions Settings	Use: CMSIS-DAP Debug	ger 🗨	Settings				
Limit Speed to Beal-Time							
CMSIS-DAP Cortex-M Target Driver Setup			\times				
Debug Trace Flash Download		2.4			~		
- Download Euroction	Add Flash Programming Alg	gorithm			~		
C Frase Full Chip IV Program							
Frase Sectors Vorte	Description	Flash Size	Device Type	Origin 5	^		
C Do not Erroro Do not And Dura	GD32F3x0 FMC	128k	On-chip Flash	Device Family Pack	age		
C Do not Erase Reset and Run	AM29x128 Flash	16M	Ext. Flash 16-bit	MDK Core			
- Programming Algorithm	K8P5615UQA Dual Flash	64M	Ext. Flash 32-bit	MDK Core			
- rogramming - rgonam	LPC18xx/43xx MX25V8035F	8M	Ext. Flash SPI	MDK Core			
Description Device Size	LPC18xx/43xx S25FL032 SP	4M	Ext. Flash SPI	MDK Core			
	LPC18xx/43xx S25FL064 SP	8M	Ext. Flash SPI	MDK Core			
	LPC407x/8x S25FL032 SPIFI	4M	Ext. Flash SPI	MDK Core			
	LPC5460x MT25QL128 SPIFI	16M	Ext. Flash SPI	MDK Core			
	M29W640FB Flash	8M	Ext. Flash 16-bit	MDK Core			
	MIMXRT105x EcoXiP Flash	4M	Ext. Flash SPI	MDK Core			
	RC28F640J3x Dual Flash	16M	Ext. Flash 32-bit	MDK Core			
,	S25FL128S_V2C	16M	Ext. Flash SPI	MDK Core			
	S29GL064N Dual Flash	16M	Ext. Flash 32-bit	MDK Core			
Λ	S29JL032H_BOT Flash	4M	Ext. Flash 16-bit	MDK Core			
	S29JL032H TOP Flash	4M	Ext. Flash 16-bit	MDK Core	×		
Add	<				>		
	C:\Users\xianju.su\AppData\Loc	al\Arm\Packs\	GigaDevice\GD32F	3x0_DFP\3.0.0\Flash	GD32F3x0.FLM		
ОК		6	_	1			
		Add	Cancel				

8. Since GD32E230xx does not support the bit length of the configuration priority group, after transplanting the GD32F3x0 library, when there is a configuration of using interrupt in the application code, the application code needs to add the "void nvic_priority_group_set (uint32_t nvic_prigroup)" function.

Table 7-1. nvic_priority_group_set function



Moreover, GD32E230xx only supports level 4 preemption priority and does not support sub priority. Therefore, after transplantation, the interrupt enabling function needs to be changed to the function shown in <u>Table 7-2. nvic irg enable function</u>.



Table 7-2. nvic_irq_enable function

/* set the priority group */

void nvic_irq_enable(uint8_t nvic_irq, uint8_t nvic_irq_pre_priority, uint8_t nvic_irq_sub_priority);

- 9. If TIMER5 is used in the project, because GD32F3x0 remove this TIMER5(Except GD32F350xx), the code of TIMER5 needs to be changed to other timer.
- 10. Compile the project. If there is an error, modify it according to the prompt. Usually, the prompt is that "#include "gd32e23x.h"" in the code is not modified to "#include "gd32f3x0.h"", and modify it according to the prompt. So far, the project has been transplanted successfully, and the development of GD32F3x0 Series MCU can be carried out.



8. Revision history

Table 8-1. Revision history

Revision No.	Description	Date		
1.0	Initial Release	Mar.15 2022		



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.

© 2022 GigaDevice – All rights reserved