**GigaDevice Semiconductor Inc.** 

GD32E230 系列移植到 GD32F3x0 系列

应用笔记 AN046



目录

目录	L C	2
图索	<b>玄引</b>	3
表索	<b>炙引</b>	5
1.	前言	6
2.	硬件差异介绍	7
3.	资源及外设地址对比介绍	9
4.	开发工具对比	11
5.	软件环境设置	12
5.	1. 使用 Keil 开发 GD32F3x0	
	5.1.1. 在 Keil4 中添加 GD32F3x0 MCU Device	12
	5.1.2. 在 Keil5 中添加 GD32F3x0 MCU Device	14
5.	2. 使用 GD-Link 工具开发 GD32F3x0	
5.	3. 使用 J-Link 工具开发 GD32F3x0	
5.4	4. 使用 IAR 开发 GD32F3x0	
	5.4.1. 在 IAR 中添加 GD32F3x0 MCU Device	
	5.4.2. 在 IAR 中编译调试 GD32F3x0	21
6.	GD32E23x 固件库适配 GD32F3x0 系列 MCU 步骤	25
7.	GD32E23x 项目底层 Library 替换成 GD32F3x0 Library 步骤	30
8.	版本历史	35
8.	版本历史	



图索引

图 2-1. GD32F3x0 系列及 GD32E230 系列 LQFP48 封装对比图	7
图 2-2. GD32F3x0 系列及 GD32E230 系列 QFN32 封装对比图	7
图 2-3. GD32F3x0 系列及 GD32E230 系列 QFN28 封装对比图	. 8
图 2-4. GD32F330/F310 系列及 GD32E230 系列 TSSOP20 封装对比图	. 8
图 2-5. GD32F310 系列及 GD32E230 系列 LQFP32 封装对比图	. 8
图 5-1. GD32F3x0 pack 包明细	12
图 5-2. GD32F3x0 系列 MCU Pack 包安装示意图(Keil4)	12
图 5-3. GD32F3x0 系列 MCU Pack 包成功安装示意图(Keil4)	13
图 5-4. GD32F3x0 系列 Flash 算法文件选择示意图(Keil4)	13
图 5-5. GD32F3x0 pack 包明细	14
图 5-6. GD32F3x0 系列 MCU Pack 包安装示意图(Keil5)	14
图 5-7. GD32F3x0 系列 MCU Pack 包安装成功示意图(Keil5)	15
图 5-8. GD32F3x0 系列 Flash 算法文件添加示意图(Keil5)	15
图 5-9. Debug 界面中选择"CMSIS-DAP Debugger"选项(Keil4)	16
图 5-10. Utilities 界面中选择"CMSIS-DAP Debugger"选项(Keil4)	17
图 5-11. GD-Link 成功连接目标芯片示意图(Keil4)	17
图 5-12. 添加 GD32F3x0 Flash 算法文件示意图(Keil4)	18
图 5-13. GD32F3x0 工程 GD-Link 仿真示意图(Keil4)	18
图 5-14. Debug 界面中选择"J-LINK/J-Trace Cortex"选项(Keil4)	19
图 5-15. Utilities 界面中选择"J-LINK/J-Trace Cortex"类型(Keil4)	19
图 5-16. J-Link 成功连接芯片示意图(Keil4)	19
图 5-17. 添加 GD32F3x0 Flash 算法文件示意图(Keil4)	20
图 5-18. GD32F3x0 工程 J-Link 仿真示意图(Keil4)	20
图 5-19. GD32F3x0 系列 MCU Pack 包安装示意图(IAR)	21
图 5-20. GD32F3x0 系列 MCU Pack 包安装完成示意图(IAR)	21
图 5-21. 在 IAR "Options"界面中选择芯片型号示意图	22
图 5-22. 在 IAR "Options"界面中添加 CMSIS 文件示意图	23
图 5-23. 在 IAR "Options"界面中添加 ICF 文件示意图	23
图 5-24. 在 IAR "Options"界面选择 Debugger 工具示意图	24
图 5-25. 在 IAR "Options"界面配置 flash loader 示意图	24
图 6-1. 打开 GD32E23x Keil 工程示意图	25
图 6-2. 在 GD32E23x 工程中选择 GD32F3x0 芯片型号示意图	25
图 6-3. 在 GD32E23x 工程中添加 GD32F3x0 的 Flash 算法示意图	26
图 6-4. 在 GD32E23x 固件库文件中添加 Cortex M4 内核文件示意图	26
图 6-5. 修改 GD32E23x 固件库中"gd32e23x.h"头文件的内容	26
图 7-1. 把 G32F3x0 固件库中 CMSIS 文件里的.h 文件复制到 GD32E23x 固件库中	30
图 7-2. 把 GD32F3x0 固件库 CMSIS 下的 Include 与 Source 文件复制替换到 GD32E23x 固件库中	法
	30
图 7-3. 把 GD32F3x0 固件库 standard_peripheral 下的文件复制替换到 GD32E23x 固件库中去	30
图 7-4. 把 GD32F3x0 固件库中的"gd32f3x0_libopt.h"文件复制到 GD32E23x 固件库中去	31



图 7-5.	打开 GD32E23x 固件库中 Template 文件下的 Keil 工程	31
图 7-6.	移除黄色标记文件并添加新文件	32
图 7-7.	修改"main.c"、"systick.c"文件中的内容	32
图 7-8.	重新选择 GD32F3x0 芯片型号	33
图 7 <b>-9</b> .	选择 GD32F3x0 Flash 算法	33



表索引

表 3-1. GD32F3x0 系列及 GD32E230 系列资源对比总览	9
表 3-2. GD32F3x0 系列及 GD32E230 系列外设地址对比总览	9
表 4-1. GD32F3x0 系列及 GD32E230 系列 IDE 环境对比	11
表 4-2. GD32F3x0 系列及 GD32E230 系列调试工具对比	11
表 6-1. 修改 GD32E23x 固件库中"gd32e23x.h"头文件的内容	26
表 6-2. 修改 GD32E23x 固件库中"gd32e23x_misc.h"头文件的内容	27
表 6-3. 修改 GD32E23x 固件库中"gd32e23x_misc.c"头文件的内容	27
表 6-4. 修改 GD32E23x 固件库中"gd32e23x_misc.h"头文件的内容	27
表 6-5. 修改 GD32E23x 固件库中"gd32e23x_misc.c"文件的内容	27
表 6-6. 去掉 GD32E23x 工程中插入等待周期的函数	28
表 6-7. 在 GD32E23x 固件库中"gd32e23x_fmc.h"文件中添加半字编程的内容	28
表 6-8. 在 GD32E23x 固件库中"gd32e23x_fmc.c"文件中添加半字编程的内容	29
表 7-1. nvic_priority_group_set 函数	33
表 7-2. nvic_irq_enable 函数	33
表 8-1. 版本历史	35



# 1. 前言

本应用笔记旨在帮助您快速将应用程序从 GD32E230xx 系列控制器移植到 GD32F3x0 系列控制器。

为了更好的利用本应用笔记中的信息,您需要从官网 www.GD32MCU.com 下载 GD32 各系 列微控制器资料,如 Datasheet、用户手册、官方例程及各种开发工具等。



# 2. 硬件差异介绍

GD32E230 系列的封装类型有: TSSOP20、LGA20、QFN28、QFN32、LQFP32、LQFP48; GD32F3x0 系列的封装类型有: TSSOP20 (仅 GD32F330/F310 系列)、QFN28、QFN32、 LQFP32 (仅 GD32F310 系列)、LQFP48、LQFP64 (仅 GD32F330/F350 系列),两个系列 相同封装的芯片引脚是兼容的,见<u>图 2-1. GD32F3x0 系列及 GD32E230 系列 LQFP48 封装对</u> 比图,图 2-2. GD32F3x0 系列及 GD32E230 系列 QFN32 封装对比图,图 2-3. GD32F3x0 系 列及 GD32E230 系列 QFN28 封装对比图,图 2-4. GD32F330/F310 系列及 GD32E230 系列 TSSOP20 封装对比图,图 2-5. GD32F310 系列及 GD32E230 系列 LQFP32 封装对比图。

### 注意**:**

- 1. TSSOP20 和 QFN28 的封装中, GD32E230 系列 PA9、PA10 可以映射为 PA11、PA12, GD32F3x0 系列不具备此功能。
- LQFP48 封装管脚 1 在 GD32E230 系列上面是 VDD,在 GD32F3x0 上是 VBAT,也就是 说 E230 不支持掉电运行 RTC。











图 2-3. GD32F3x0 系列及 GD32E230 系列 QFN28 封装对比图



### 图 2-4. GD32F330/F310 系列及 GD32E230 系列 TSSOP20 封装对比图



### 图 2-5. GD32F310 系列及 GD32E230 系列 LQFP32 封装对比图





# 3. 资源及外设地址对比介绍

GD32F3x0 与 GD32E230 的资源有细微的差别:

- 1. GD32F3x0 具有 TIMER1 外设,但没有 TIMER5 (GD32F350 保留有此外设)外设, GD32E230 不具有 TIMER1 外设,但有 TIMER5 外设;
- 2. GD32E230 系列有一路比较器, GD32F330/F310 没有该外设, GD32F350 具有此外设;
- 3. GD32E230 系列新增了 1K 的 OTP 区域, GD32F3x0 没有该资源。
- 4. GD32F350 具有 USBFS、HDMI-CEC、DAC、TSI 功能, GD32F330/F310、GD32E230 没有这些外设。

详情请查询<u>表 3-1. GD32F3x0 系列及 GD32E230 系列资源对比总览及表 3-2. GD32F3x0 系</u> 列及 GD32E230 系列外设地址对比总览。

Peripheral	GD32F310系列	GD32F330系列	GD32F350系列	GD32E230系列
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
主频	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
12S	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	<b>-40-85</b> ℃	<b>-40-85</b> ℃	<b>-40-85</b> ℃	<b>-40-85</b> ℃

### 表 3-1. GD32F3x0 系列及 GD32E230 系列资源对比总览

注: 以上"/"代表有多种情况,需要根据具体芯片型号区分。

表 3-2. GD32F3x0 系列及 GD32E230 系列外设地址对比总览

Peripheral	BUS	GD32F3x0系列	GD32E230系列
GPIOF		0X48001400	0X48001400
GPIOD	AHB2	0X48000C00	-
GPIOC		0X48000800	0x48000800



GPIOB		0X48000400	0X48000400
GPIOA		0X48000000	0X48000000
USBFS		0X5000000	-
TSI		0X40024400	-
CRC		0X40023000	0X40023000
FMC	AIDI	0X40022000	0X40022000
RCU		0X40021000	0X40021000
DMA		0X40020000	0X40020000
DBG		0xE0042000	0X40015800
TIMER16		0X40014800	0X40014800
TIMER15		0X40014400	0X40014400
TIMER14		0X40014000	0X40014000
USART0	4002	0X40013800	0X40013800
SPI0/I2S0	APD2	0X40013000	0X40013000
TIMER0		0X40012C00	0X40012C00
ADC		0X40012400	0X40012400
EXTI		0X40010400	0X40010400
SYSCFG+CMP		0X40010000	0X40010000
СТС		0X4000C800	-
CEC		0X40007800	-
DAC		0X40007800	-
PMU		0X40007000	0X40007000
I2C1		0X40005800	0X40005800
I2C0		0X40005400	0X40005400
USART1		0X40004400	0X40004400
SPI1		0X40003800	0X40003800
FWDGT	AFDI	0X40003000	0X40003000
WWDGT		0X40002C00	0X40002C00
RTC		0X40002800	0X40002800
TIMER13		0X40002000	0X40002000
TIMER5		0X40001000	0X40001000
TIMER2		0X40000400	0X40000400
TIMER1		0X4000000	-
SRAM		0x20000000	0x20000000
Option Byte		0x1FFFF800	0x1FFFF800
Main Flash		0x08000000	0x08000000
System Memory		0x1FFFEC00	0x1FFFEC00
OTP		-	0x1FFF7000



# 4. 开发工具对比

GD32F3x0可使用MDK for ARM的Keil 4及Keil 5进行开发,使用Keil 4建议安装4.74及以上; 使用Keil 5建议安装5.20以上版本。也可以使用IAR for ARM开发,建议安装IAR 6.3及以上版 本,如<u>表4-1. GD32F3x0系列及GD32E230系列IDE环境对比</u>介绍。

### 表 4-1. GD32F3x0 系列及 GD32E230 系列 IDE 环境对比

•		
芯片系列	GD32F3x0系列	GD32E230系列
KEIL	KEIL4或者Keil5均支持	KEIL 5.25及以上版本
IAR	IAR 6.3及以上版本	IAR 8.1及以上版本

GD32F3x0 可以使用 JLINK、ULINK、GDLINK 等调试工具进行开发,如<u>表 4-2. GD32F3x0 系</u> 列及 GD32E230 系列调试工具对比</u>介绍。

### 表 4-2. GD32F3x0 系列及 GD32E230 系列调试工具对比

芯片系列	GD32F3x0系列	GD32E230系列
JLINK JLINK OB、V8、V9等均支持		仅JLINK V9及以上版本支持
ULINK	支持	支持
GDLINK	支持	支持



# 5. 软件环境设置

## 5.1. 使用 Keil 开发 GD32F3x0

目前市面通用的MDK for ARM版本有Keil 4和Keil 5:使用Keil 4建议安装4.74及以上,使用Keil 5建议安装5.20以上版本。

### 5.1.1. 在 Keil4 中添加 GD32F3x0 MCU Device

1. 从GD32MCU官网下载相关的GD32F3x0系列插件。

### 图 5-1. GD32F3x0 pack 包明细

gd32mcu.com/cn/download/0?kw=GD32F3						
全部资料	Introduction: Windows系统下的串口ISP控制台 GD32F1/GD32F2/GD32F3/GD32F4/GD32E1/G	工具,用于串口烧录GD MCU Flash数 iD32E2/GD32VF1系列MCU	居。适用于			
GD32L2 MCU	GD32F3x0 Scatter Loading	:	2.0.0 🖀			
GD32F1 MCU	Introduction: GD32F3x0分散加載例程。					
GD32F2 MCU	GD32F3x0 AddOn		2.0.0			
GD32F3 MCU	Introduction: GD32F3X0蒸9归AR7.4、Kell MD	、4.74、Kell MDK 5.26 及更向版本支持	安装又件。			
📲 GigaDevice.GD32F	3x0_AddOn.2.0.0.exe KEIL4	2019/12/24 14:50	应用程序			
📲 GigaDevice.GD32F	GigaDevice.GD32F3x0_DFP.2.0.0.pack KEIL5 2019/12/24 14:50 uVision Softwar					
👼 IAR_GD32F3x0_AD	DON_2.0.0.exe IAR	2019/12/24 14:50	应用程序			

2. 双击安装文件,把插件安装至Keil 4的目录,一般都会默认选择,如若同时安装了Keil 4和 Keil 5才需要手动选择。

### 图 5-2. GD32F3x0 系列 MCU Pack 包安装示意图(Keil4)

Setup GigaDevice GD32F30x Device AddOn Package to Keil MDK-ARM X				
Folder Selection Select the folder where SETUP will install files.	<b>ARM°KEIL°</b> Microcontroller Tools			
This Add-On will install into the following product folder.				
To install to this folder, press 'Next'. To install to a different folder folder.	, press 'Browse' and select another			
Destination Folder				
C:\Keil	Browse			
— Keil MDK-ARM Setup	<< Back Next>> Cancel			

3. 安装成功后,重新打开Keil 4,则可以在"File->Device Database"中出现"Gigadevice"



的下拉选项,点击可以查看到相应的型号。

### 图 5-3. GD32F3x0 系列 MCU Pack 包成功安装示意图(Keil4)

🕅 Options for Target 'GD3	32F330'				×
Device Target Output I	isting   User	C/C++  Asm	Linker   Debug   V	tilities	
Database: GigaDev Vendor: GigaDevice	rice GD32F3x0 De	vices 💌			
Device: GD32F330G4 Toolset: ARM					
□	<ul> <li>▲ Core: ARI</li> <li>Memu- 16 - 4 K</li> <li>1 * - 4 * - 4 * - 1 * - 1 * - 1 * - 1 *</li> <li>- 4 K</li> <li>- 4 K</li> <li>- 4 K</li> <li>- 4 K</li> <li>- 1 *</li> <li></li> <li>- 1 *</li> <li></li> <li></li></ul>	M 32-bit Cortex-M4 Mic ories: Kbytes of Flash memor bytes of SRAM with H general timer (32bit) general timer (16bit) advance timer(16bit) watch dog timer C USART I2C SPI ADC (10 channels)	rocontroller, 120MHz y W parity checking	max)	×
	OK	Cancel	Defaults		Help

4. 为了后续debug工作的顺利进行,建议检查一下安装路径下是否有下载算法,可以通过如下方式查看:打开一个工程,将型号选为GD32F3x0的型号,然后 "Options for Target -> Debug ->Settings -> Flash Download-> Add",如果下拉选项中有GD32F3x0的Flash下载算法则表示安装成功。

### 图 5-4. GD32F3x0 系列 Flash 算法文件选择示意图(Keil4)

ownload Function	ase Full Chip 🔽 Program ase Sectors 🔽 Verify not Erase 🔽 Reset and Run	RAM for Algo Start: 0x20	nithm 10000000 Size	: 0x0800
rogramming Algori	Add Flash Programming Alg	orithm		×
Description	-			
	Description	Device Type	Device Size	^
	GD32F30x High-density FMC	On-chip Flash	512k	
	GD32F30x Extra-density FMC	On-chip Flash	3M	
	GD32F3x0 FMC	On-chip Flash	128k	
	GD32F403 FMC	Un-chip Hash	3M	
	GD32F4XX_IMB_FMC	On-chip Hash	11/1	
	GD32F4XX_ZIVIB FIVIC	On-chip Flash	∠IVI 2M	
	GD32F4XX_SIVID FIVIC	On-chip Flash	5126	
	UT22 Sories Flash	On-chip Flash	12K	
	HT32 Series Flash Ontions	On-chip Flash	ALC	
	K8P5615UQA Dual Elash	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	¥
		Canad	1	



### 5.1.2. 在 Keil5 中添加 GD32F3x0 MCU Device

1. 从GD32MCU官网下载相关的GD32F3x0系列插件。

### 图 5-5. GD32F3x0 pack 包明细

gd32mcu.com/cn/do	wnload/0?kw=GD32F3				
全部资料	Introduction: Windows系统下的串口ISP控制台 GD32F1/GD32F2/GD32F3/GD32F4/GD32E1/G	工具,用于串口烧录GD MCU Flash数排 D32E2/GD32VF1系列MCU	롢 <u>。</u> 适用于		
GD32L2 MCU	GD32F3x0 Scatter Loading	2	2.0.0 🖀		
GD32F1 MCU	Introduction: GD32F3x0分散加載例程。				
GD32F2 MCU	GD32F3x0 AddOn		2.0.0		
GD32F3 MCU	Introduction: GD32F3X0蒸9UAR7.4、Kell MD1	、4.74、Kell MUK 5.26	文友义件。		
GigaDevice.GD32F	3x0_AddOn.2.0.0.exe KEIL4	2019/12/24 14:50	应用程序		
📲 GigaDevice.GD32F	3x0_DFP.2.0.0.pack KEIL5	2019/12/24 14:50	uVision Software.		
👼 IAR_GD32F3x0_ADI	┃ AR_GD32F3x0_ADDON_2.0.0.exe				

2. 解压并安装至Keil 5的目录, 一般都会默认选择。

### 图 5-6. GD32F3x0 系列 MCU Pack 包安装示意图(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 Examples	
Destination Folder	
C:\Users\xian\AppData\Local\Arm\Packs\GigaDevice\GD32F30x_DFP\2.1.0	
Keil Pack Unzip	
Pack already installed.         << Back         Next >>         Cancel           Click "Next" to replace.         << Back	

3. 安装完后重新打开Keil 5工程,即可在Device中出现Gigadevice的型号。



### 图 5-7. GD32F3x0 系列 MCU Pack 包安装成功示意图(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: http://gd32mou.21ic.com/data/documer	
Search: GD32F30x Series GD32F3x0 Serie	*
GD32F330C4     GD32F330C6     GD32F330C6     GD32F330C6     GD32F330C8	
Image: GD32F330CB     GD32F330 - ARM Cortex-M4 Core       Image: GD32F330F4     Frequency up to 84 MHz       Image: GD32F330F6     Frequency up to 84 MHz       Image: GD32F330F6     Single-cycle multiplier and hardware divider	
▲ Memories	~
OK Cancel Defaults Help	

- **4**. 在 "Options for Target -> Debug ->Settings ->Flash Download" 中添加Flash算法,会出 现GD32F3x0的算法,即说明安装成功。根据相应的芯片选择合适的算法,即可下载仿真。
- 图 5-8. GD32F3x0 系列 Flash 算法文件添加示意图(Keil5)

Cortex JLink/JTra	ace Target Driver Setup						×
Debug Trace	Flash Download						
Download Fu	O Erase Full Chip I Pro ○ Erase Sectors I Ver	igram ify	RAM for A	Algorithm	Size: 0x10	000	
Programm	Add Flash Programming Alg	gorithm				×	
	Description	Flash Size	Device Type	Origin			
Descript	GD32Fx0 FMC AM25x128 Hash K8P5615UQA Dual Rash LPC18xx/43xx MX25V8035F LPC18xx/43xx S25FL032 SP LPC407x/43xx S25FL032 SPIFI LPC5460x MT250L128 SPIFI LPC5460x MT250L128 SPIFI MIMKRT105x EoxXP Rash MIXKRT105x EoxXP Rash S25FL128S_V2C S23GL064N Dual Rash S25UL034N Dual Rash S25UL032H_BOT Rash S29JL032H_TOP Rash	128k 16M 64M 8M 4M 4M 16M 8M 4M 16M 16M 16M 16M 4M	On-chip Flash Ext. Flash 16-bit Ext. Flash 32-bit Ext. Flash SPI Ext. Flash SPI Ext. Flash SPI Ext. Flash SPI Ext. Flash SPI Ext. Flash 32-bit Ext. Flash 32-bit Ext. Flash 16-bit Ext. Flash 16-bit Ext. Flash 16-bit	Device Family F MDK Core MDK Core	ackage		
		Add	Cance	1			
				确定	取消	¥	应用(A)

5. Keil 4打开Keil 5工程



如果没有安装Keil 5,也是能够使用Keil 4来编译Keil 5的工程,具体做法就是修改工程的后 缀名,将Keil 5工程的后缀名 "xxxx.uvprojx"修改为 "xxxx.uvproj",即可使用Keil 4来查 看编译了。

6. Keil 5打开Keil 4工程

如果使用Keil 5打开Keil 4工程,打开时会遇到找不到MCU器件的情况,这种可以直接将 Keil 4工程的后缀名"xxxx.uvproj修改为xxxx.uvprojx",即可正常使用Keil 5来查看编译了。

## 5.2. 使用 GD-Link 工具开发 GD32F3x0

使用GD-Link工具来debug GD MCU,硬件上需要用GD-Link工具连接开发板,软件上具体配置如下:

 打开一个GD32F3x0的工程,在 "Options for Target -> Debug"中选择 "CMSIS-DAP Debugger",部分客户反馈找不到这一驱动器选项,那是因为MDK版本过低,只有Keil4.74 以上的版本和Keil 5才支持 "CMSIS-DAP Debugger"选项。

### 图 5-9. Debug 界面中选择 "CMSIS-DAP Debugger" 选项(Keil4)

関 Options for Target 'GD32F330'	×
Device   Target   Output   Listing   User   C/C++   A	Asm Linker Debug Vtilities
C Use <u>S</u> imulator <u>Settings</u>	⊡se: CMSIS-DAP Debugger      Settings     Settin
Load Application at Startup Initialization File:	C Load Application at Startup 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 Can	cel Defaults Help

2. 在 "Options for Target -> Utilities",也需选择 "CMSIS-DAP Debugger"。



### 图 5-10. Utilities 界面中选择 "CMSIS-DAP Debugger"选项(Keil4)

📆 Options for Target 'GD3	2F330'			×
Device   Target   Output   L	isting   User   C/C	:++   Asm   Lin	aker Debug Utilitie	s
Configure Flash Menu Comm	and			
Use Target Driver for Flag	ash Programming			
CMSIS-DAP	Debugger	<ul> <li>Settings</li> </ul>	✓ Update Target befor	re Debugging
Init File:			Edit	
C Use External Tool for Fla	ash Programming			
Command:				
Arguments:				
🗖 Run Inde	pendent			
	ок	Cancel	Defaults	Help

3. 在 "Options for Target -> Debug ->Settings" 勾选SWJ、Port选择 SW。右框IDcode会出 现 "0xXBAXXXXX"。

Options for Target 'GD32F330'		×
Cortex-M Target Driver Setup		×
Debug Flash Download		
-CMSIS-DAP - JTAG/SW Adapter-	SW Device	
CMSIS-DAP	SWDIO	Move
Serial No: CMSIS-DAP	GUZZBAU14// ARIVI CoreSignt SW-DP	Down
Firmware Version: 2.0.0		Down
SWJ Port: SW 💌	Automatic Detection ID CODE:	
Max Clock: 1MHz 🗨	Add Delete Update	AP: 0x00
Connect & Reset Options	Cache Options Download Opt	ions
Reset after Connect	Autodetect	to Flash
	UK Lancel	Help
C	K Cancel Defaults	Help

图 5-11. GD-Link 成功连接目标芯片示意图(Keil4)

4. 在"Options for Target -> Debug ->Settings -> Flash Download"中添加GD32的Flash算法。



### 图 5-12. 添加 GD32F3x0 Flash 算法文件示意图(Keil4)

🕅 Options for Target 'GD32F3:	30'			×
Cortex-M Target Driver Setup				×
Debug Flash Download				
Download Function C Erase Full Chip C Erase Sectors C Do not Erase Programming Agorithm	<ul> <li>✓ Program</li> <li>✓ Verify</li> <li>✓ Reset and Run</li> </ul>	RAM for Alg	gorithm 20000000Size: 0x080	0
Description GD32F3x0 FMC	Device Type On-chip Flash	Device Size 128k	Address Range 08000000H - 0807FFFH	
		Start:	Size:	
	Add	Remove		
	OK	Cance	1	Help
	ОК	Cancel	Defaults	Help

5. 单击*图5-13. GD32F3x0工程GD-Link仿真示意图(Keil4)*的红框的快捷方式debug,即可 使用GD-Link进行仿真。

#### File Edit View Project Flash Debug Peripherals Tools SVCS Window Help 🗉 🗟 🦸 🔘 □ 😂 🚽 🗿 🕺 🖏 🖏 🥱 🖂 🔶 → 🕴 🎋 🎘 🎘 🎼 準 /// /// 🖄 word 🔗 🊓 🔳 🖣 \$\$ | 월 ② | 원 원 원 19 | 수 | 고려, 대 프랑, 유 · 프 · 공 · 프 · 명 · | 옷 · 🛛 🖬 Disassembly Registers 54: rcu\_configuration(); 0x080004A8 F000F832 BL.W rcu v Register figuration Core 55: led\_config(); < \_ 56: - R3 -- R4 -- R5 -- R5 -- R7 -- R8 -- R9 -- R10 -- R11 -- R12 -- R13 (SP) -- R14 (LR) main.c startup\_gd32f1x0.s 51 L \*/ 52 int main(void) 53 - { 53 54 € 55 0. 0. 0. rcu configuration(); led\_config(); 56 57 /\* setup SysTick Timer for 1ms interrupts \*/ 58 59 0. systick\_config(); while(1){ 60 📥 /\* PAO:Keyl is pressed \*/ if(0 == gpio\_input\_bit\_get(GPIOA, GPIO\_PIN\_0)){ /\* delay 100ms \*/ 61 62 = Banked System Internal 63 Mode Privilege 64 65 66 delay\_1ms(100); /\* LED1 on\*/ T. P. M. 6. gpio\_bit\_set(GPIOA, GPIO\_PIN\_1); delay\_ims(35); /\* LED1 off\*/ min\_bit Stack States Sec 67 68 gpio bit reset(GPIOA, GPIO PIN 1); 69 70 71 delay\_1ms(35); }else{ 🖻 Project 🛛 🚟 Registers

### 图 5-13. GD32F3x0 工程 GD-Link 仿真示意图(Keil4)

# 5.3. 使用 J-Link 工具开发 GD32F3x0

使用J-Link工具来debug GD MCU,硬件上需要用J-Link工具连接开发板,软件上具体配置如下:



1. 打开一个GD32F3x0的工程,在"Options for Target -> Debug中"选择"J-LINK/J-Trace Cortex"。

### 图 5-14. Debug 界面中选择 "J-LINK/J-Trace Cortex" 选项(Keil4)

🕎 Options for Target 'Target 1'	X
Device   Target   Output   Listing   User   C/C++   #	sm Linker Debug Vtilities
C Use Simulator Settings	Use: J-LINK / J-Trace Cortex     Settings
✓ Load Application at Startup ✓ Run to main() Initialization File:	Iv     Load Application at Startup     Iv     Run to main()       Initialization File:

2. 在Options for Target -> Debug ->Utilities,也要选择"J-LINK/J-Trace Cortex"。

### 图 5-15. Utilities 界面中选择"J-LINK/J-Trace Cortex" 类型(Keil4)

🕜 Options for Target 'Target 1'	×
Device   Target   Output   Listing   Vser   C/C++   Asm   Linker   Debug   Utilities	
Configure Flash Menu Command	
• Use Target Driver for Flash Programming	
J-LINK / J-Trace Cortex  Settings  Update Target before Debuggin	g
Init File: Edit	

3. 在"Options for Target -> Debug ->Settings", Port选择SW。右框IDCODE会出现 "0xXBAXXXX",表示成功连接目标芯片。

### 图 5-16. J-Link 成功连接芯片示意图(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 Ox2BA01477 ARM Core Sight SW-DP	Up
HW :   V9.20 dll :   V5.02c		Down
Five: 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:	

4. 在"Options for Target -> Debug ->Settings -> Flash Download"中添加GD32的Flash算法。



### 图 5-17. 添加 GD32F3x0 Flash 算法文件示意图(Keil4)

Cortex JLink/JTrace Target Driv	er Setup			×
Debug   Trace Flash Downlos	ad			
Download Function C Erase Full Chip C Erase Sectors C Do not Erase	<ul> <li>✓ Program</li> <li>✓ Verify</li> <li>✓ Reset and Run</li> </ul>	RAM for A	gorithm 20000000 Size: 0x0300	
Description	Device Type	Device Size	Address Range	
GD32F3x0 FMC	On-chip Flash	128k	08000000H - 0807FFFFH	
		Start:	Size:	
	Add	Remove		
	ОК	Cance	l Hel	.p

- 5. 单击如<u>图5-18. GD32F3x0工程J-Link仿真示意图(Keil4)</u>所示的快捷方式debug,即可使 用J-Link进行仿真。
- 图 5-18. GD32F3x0 工程 J-Link 仿真示意图(Keil4) File Edit View Project Flash Debug Peripherals Tools SVCS Window Help 🔹 🗟 🦸 🔯 🔗 🚷 🔳 🔍 
   V
   54:
   rcu\_configuration();

   0x080004A8 F000F832 BL.W rcu

   55:
   led\_config();

   0
   56:
   egisters Register - Core R3 - R4 - R5 - R6 - R7 - R6 - R9 - R10 - R11 - R12 - R13 (SP) - R14 (LR) main.c startup\_gd32f1x0.s 51 <sup>L</sup>\*/ 52 int main(void) 53 🖂 ( 
   >5
   rcu\_configuration();

   ●5
   led\_config();

   56
   /\* setup SysTick Timer for 1ms interrupts \*/
   58 systick\_config(); 59 60 61 62 63 64 65 56 ked System -System -Internal - Mode - Privilege - Stack - States - Sec T. P. H. 6. /\* LED1 on\*/
  gpio\_bit\_set(GFIOA, GFIO\_PIN\_1);
  delay\_lms(35);
  /\* LED1 off\*/
  gpio\_bit\_reset(GFIOA, GFIO\_PIN\_1);
  delay\_lms(35); 66 67 68 69 70 71 }else{ ect 🛛 🗮 Regi

#### 使用 IAR 开发 GD32F3x0 5.4.

IAR版本众多,版本之间的兼容性并不好,如果初次使用建议安装7.3以上的版本,安装好IAR以 后再根据该文档来添加GD的器件型号,进行相关的debug工作。

#### 5.4.1. 在 IAR 中添加 GD32F3x0 MCU Device



- 1. 从相关网站下载相应的GD32F3x0系列插件: IAR\_GD32F3x0\_ADDON\_2.0.0.exe:
- 2. 运行IAR\_GD32F3x0\_ADDON\_2.0.0.exe,单击Start开始安装插件。

### 图 5-19. GD32F3x0 系列 MCU Pack 包安装示意图(IAR)

a Setup GigaDevice GD32F3x0 Device AddOn Package to IAR v2.0.0	×
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 for $\_$ Destination Folder $\_$	lder.
C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.3 Browse	
Realtime Status	
0%	
Start Cancel	

3. 安装成功后单击Finish,结束插件安装。

### 图 5-20. GD32F3x0 系列 MCU Pack 包安装完成示意图(IAR)

羄 Setup GigaDevice GD32F3x0 Device AddOn Package to IAR v2.0.0	×
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 fo	der.
Destination Folder	
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. 在 IAR 中编译调试 GD32F3x0

在上一小节中我们已经添加了GD32F3x0系列的插件,这一小节我们介绍应如何使用它。

1. 使用IAR编译GD的型号,有两个办法,一种是使用现有的工程进行修改,另一种就是重新 建立工程,这里就不细说具体工程应该如何建立,GD的工程建立和别的平台都一致,建立 工程时选择GD的相应型号即可。



图 5-21. 在 IAR "Options	"界面中选择芯片型号示意图
------------------------	---------------

Options for node "Project"						$\times$
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	Library O Target Processor O Core () Device	ptions 2 Output variant GD	MISRA Library Co rtex-M4 GD32F330x8	-C:2004 nfiguration	MISRA-C:1998 Library Options 1	×
CADI CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	CMSIS- Endian mo © Little Big BE32 © BE32 © BE8	Pack No de F ension ed SIMD (N	ne Floating point FPU D registers EON)	settings VFPv4 singl 16  v TrustZc Mode Se	e precision v	
			[	OK	Cancel	

 6.1版本以后的IAR不需要添加CMSIS文件(core\_cm3.c和core\_cm3.h),但是需要勾选 "General Options->Library Configuration"的Use CMSIS,如果软件代码有使用到printf 函数,还需要修改Library为FULL。



图 5-22. 在 IAR "Options"界面中添加 CMSIS 文件示意图

Options for node "Project" Category:			×
General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Output Build	Library Options 2 Target Output	MISRA-C:2004 Library Configuration	MISRA-C:1998 Library Options 1
Build Actions Linker Debugger Simulator CADI CMSIS DAP GDB Server	Library: Full v	Description: Use the full configuration of runtime library. Full locale file descriptor support, mu and scanf, and hex floats in	of the C/C++ interface, C locale, iltibytes in printf n strtod.
I-jet J-Link/J-Trace TI Stellaris Nu-Link PE micro	Configuration file: \$TOOLKIT_DIR\$\inc\c\E	DLib_Config_Full.h rt in library	
ST-LINK Third-Party Driver TI MSP-FET TI XDS	O None     Semihosted     IAR breakpoint	stdout/stderr Via semihosting Via SWO	Use CMSIS
		OK	Cancel

3. 芯片的Link文件建立工程时会默认根据型号选定,但是编译前还是要有检查的习惯,检查 一下ICF文件是否有配置,是否正确。

图 5-23. 在 IAR "Options"界面中添加 ICF 文件示意图

Options for node "Project"						>
Category:					Factory	Settings
General Options Static Analysis Runtime Checking						
C/C++ Compiler	#define	Diagnostics	Checksum	Encodings	Extra C	ptions
Assembler	Config	Library Input	Optimizations	Advanced	Output	List
Custom Build Build Actions	Linker	configuration file				
Linker	✓ Ove	erride default				
Debugger Simulator	\$1	TOOLKIT_DIR\$\coi	nfig\linker\GD\G	iD32F330x8.ic	f	
CMSIS DAP GDB Server		Edit				
I-jet J-Link/J-Trace TI Stellaris	Configur	ration file symbol	definitions: (one	per line)		

4. 配置 "Debugger->Setup"选项,新建立的工程默认是Simulator模拟,如果需要调试那么 需要根据实际情况来选择:使用GD-Link选择CMSIS DAP(兼容性不好,不建议在IAR下 使用)或使用J-Link选择J-Link/J-Trace。



atenorur		
alegoly.		Factory Setting:
tatic Analysis		
untime Checking		
C/C++ Compiler	Setup Download Images Extra Options Multicore F	lugins
Assembler		
Output Converter		
Custom Build	Driver 🛛 Run to	
Build Actions	J-Link/J-Trace V main	
Linker	Simulator	
Simulator	CADI	
CADI	CMSIS DAP	
CMSIS DAP	GDB Server	
GDB Server	Liet	
I-jet	Link/LTrace	
J-Link/J-Trace	TI Stellaris	
TI Stellaris	Nu-Link	***
Nu-Link	PEmicro	
PE micro	ST-LINK	
ST-LINK	Third-Party Driver	
Third-Party Driver	TI MSP-FET	
TI MSP-FET	TI XDS (debugger\GD\GD32F330x8	.ddf
11 XDS		

图 5-24. 在 IAR "Options"界面选择 Debugger 工具示意图

5. 配置 "Debugger->Download" 选项,新建的工程有可能没有配置download选项,如果我 们需要调试代码那么务必要勾选User flash loader选项,且保证board file准确,否则程序 无法正常下载至芯片内部。

	×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Actions Linker Petugger Simulator CADJ GDB Server T Jett J-ink/J-Trace T Stellaris Nu-Link Perform mass erase before flashing ST-LINK Thrd-Party Driver TI MSP-FET TI XDS Factory Settings Factory Settings Factory Settings Factory Settings Factory Settings Factory Settings Factory Settings Factory Settings Factory Settings Factory Settings Public Actions Construction Setup Download Suppress download Suppress download Construction Setup Download Images Extra Options Multicore Plugins Override default .board file R\$\config\flashloader(s) Suppress download Construction Perform mass erase before flashing ST-LINK Thrd-Party Driver TI MSP-FET TI XDS	es Extra Options Multicore Plugins
OK	OK Cancel

图 5-25. 在 IAR "Options"界面配置 flash loader 示意图



6.

# GD32E23x 固件库适配 GD32F3x0 系列 MCU 步骤

本章将使用GD32E23x\_Firmware\_Library\_V1.1.1固件库文件Template里的工程做示例,介绍 如何适配GD32F3x0系列芯片。

1. 打开Keil工程

### 图 6-1. 打开 GD32E23x Keil 工程示意图

GD32E23x_Firmw	are_Library_V1.1.1 > Template >		
^	名称	~ 修改日期	类型
	IAR_project	2021/10/27 14:07	文件夹
	Keil_project	2021/10/28 13:35	文件夹
	gd32e23x_it.c	2021/5/18 14:21	C 文件
	gd32e23x_it.h	2021/5/18 14:21	H 文件
	gd32e23x_libopt.h	2021/5/18 14:21	H 文件
	main.c	2021/5/18 14:21	H 文件
	main.h	2021/5/18 14:21	H 文件
	📄 readme.txt	2021/5/18 14:21	文本文档
	🥮 systick.c	2021/5/18 14:21	C 文件
	🎒 systick.h	2021/5/18 14:21	H 文件

2. 打开工程后, "Options for Target -> Device", 选择对应的GD32F3x0型号。



图 6-2. 在 GD32E23x 工程中选择 GD32F3x0 芯片型号示意图

3. 在"Options for Target -> Debug ->Settings -> Flash Download"中添加GD32F3x0的Flash 算法。



### 图 6-3. 在 GD32E23x 工程中添加 GD32F3x0 的 Flash 算法示意图

Options for Target 'GD32E2	3x'	$\times$
CMSIS-DAP Cortex-M Target D	river Setup	×
Debug   Trace Flash Downloa	a	
Download Function C Erase Full Chip Erase Sectors Do not Erase Programming Algorithm	✓ Program       RAM for Algorithm         ✓ Verfy       Start: [0x20000000]       Size: [0x00001000]         Reset and Run       Start: [0x20000000]       Size: [0x00001000]	
GD32F3x0 FMC	Device Size Device Type Address Bange 128k On-chip Flash 08000000H - 0801FFFFH	
	Start: 0x08000000 Size: 0x00020000	
	Add Remove	
	OK Cancel F	lelp

4. 拷贝Cortex M4 内核支持文件至:

x:\GD32E23x\_Firmware\_Library\_V1.1.1\Firmware\CMSIS。

### 图 6-4. 在 GD32E23x 固件库文件中添加 Cortex M4 内核文件示意图

D:) > GD32E23x_Firmware_Library_V1.1.1 > Firmware > CMSIS >				
名称 ^ ~	修改日期	类型	大小	
GD	2021/10/27 14:07	文件夹		
📃 core_cm4.h	2020/11/23 11:45	H 文件	108 KB	
core_cm4_simd.h	2020/11/23 11:45	H 文件	23 KB	
core_cmFunc.h	2020/11/23 11:45	H 文件	16 KB	
core_cmInstr.h	2020/11/23 11:45	H 文件	17 KB	

5. 修改GD32E23x固件库中"gd32e23x.h"头文件的内容。

### 图 6-5. 修改 GD32E23x 固件库中 "gd32e23x.h" 头文件的内容

/* includes */	/* includes */		
<pre>#include "core cm4.h" #include "system_gd32e23x.h" #include <stdint.h></stdint.h></pre>	<pre>#include "core cm23.h" #include "system_gd32e23x.h" #include <stdint.h></stdint.h></pre>		
	/* configuration of the Cortex-M23 processor and core pe		

<pre>/* configuration of the Cortex-M4</pre>	processo	or and	core periphe	/* configuration of the Cortex-M2	3 processo:	c and core pe
#define CM4 REV	0x0001	/*!<	Core revisio	<pre>#defineCM23_REV</pre>	0x0100U	/*!< Core re
#define MPU PRESENT	011	1+12	GD32F3x0_do	<pre>#defineSAUREGION_PRESENT</pre>	0U	/*!< SAU rec
idefine NVIC PRIO BITS	411	1812	GD32F3x0 use	<pre>#define MPU PRESENT</pre>	00	/*!< MPU is
tdefine Vendor SysTickConfig	011	1.1.2	set to l if	#define VTOR PRESENT	10	/*!< VTOR is
idefine FDI DEFSENT	111	1812	FPU present	#define NVIC PRIO BITS	2U	/*!< Number
Factine	10	1.5	rio present	<pre>#define Vendor SysTickConfig</pre>	00	/*!< Set to

### 表 6-1. 修改 GD32E23x 固件库中 "gd32e23x.h" 头文件的内容

修改后		修改前		
#include "core_cm23.h"		#include "core_cm23.h"		
#defineCM4_REV	0x0001	#defineCM23_REV	0x0100U	



#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. GD32E230不支持中断分组,所以固件库中没有 "void nvic\_priority\_group\_set(uint32\_t nvic\_prigroup)"函数,需要在固件库添加相应的内容。

### 表 6-2. 修改 GD32E23x 固件库中 "gd32e23x\_misc.h" 头文件的内容

/* priority group - define the pre-emption priority and the subpriority */				
#define NVIC_PRIGROUP_PRE0_SUB4	((uint32_t)0x00000700)			
#define NVIC_PRIGROUP_PRE1_SUB3	((uint32_t)0x0000600)			
#define NVIC_PRIGROUP_PRE2_SUB2	((uint32_t)0x00000500)			
#define NVIC_PRIGROUP_PRE3_SUB1	((uint32_t)0x00000400)			
#define NVIC_PRIGROUP_PRE4_SUB0	((uint32_t)0x00000300)			

### /\* set the priority group \*/

void nvic\_priority\_group\_set(uint32\_t nvic\_prigroup);

### 表 6-3. 修改 GD32E23x 固件库中 "gd32e23x\_misc.c" 头文件的内容

void nvic\_priority\_group\_set(uint32\_t nvic\_prigroup)

/\* set the priority group value \*/

SCB->AIRCR = NVIC\_AIRCR\_VECTKEY\_MASK | nvic\_prigroup;

7. GD32E230仅支持4级抢占优先级,不支持子优先级,GD32F3x0既支持抢占优先级也支持 子优先级,需要在固件库里修改相应的内容。

### 表 6-4. 修改 GD32E23x 固件库中 "gd32e23x\_misc.h" 头文件的内容

/\* enable NVIC request \*/

void nvic\_irq\_enable(uint8\_t nvic\_irq, uint8\_t nvic\_irq\_pre\_priority, uint8\_t nvic\_irq\_sub\_priority);

### 表 6-5. 修改 GD32E23x 固件库中 "gd32e23x\_misc.c" 文件的内容

void r	nvic_irq_enable(uint8_t nvic_irq,
	uint8_t nvic_irq_pre_priority,
	uint8_t nvic_irq_sub_priority)
{	
U	uint32_t temp_priority = 0x00U, temp_pre = 0x00U, temp_sub = 0x00U;
/	* use the priority group value to get the temp_pre and the temp_sub */
s	switch ((SCB->AIRCR) & (uint32_t)0x700U) {
С	case NVIC_PRIGROUP_PRE0_SUB4:
	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\_irg\_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. GD32F3x0的Flash是零等待的,GD32E230系列需要配置Flash插入等待周期,因此可去 掉插入等待周期的函数。

### 表 6-6. 去掉 GD32E23x 工程中插入等待周期的函数

FMC\_WS = (FMC\_WS & (~FMC\_WS\_WSCNT)) | WS\_WSCNT\_2;

 GD32E230的Flash支持32位和64位编程,GD32F3x0的Flash支持32位字和半字编程。如 过应用代码中使用了64位编程需要修改成32位字或半字编程,GD32E230固件库中需要添 加半字编程的内容。

表 6-7. 在 GD32E23x 固件库中 "gd32e23x\_fmc.h" 文件中添加半字编程的内容

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

fmc\_state\_enum fmc\_halfword\_program(uint32\_t address, uint16\_t data);



}

表 6-8. 在 GD32E23x 固件库中 "gd32e23x\_fmc.c" 文件中添加半字编程的内容
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 the FMC state \*/
 return fmc\_state;

- 10. 如项目中使用到TIMER5定时器,由于GD32F3x0剪裁掉此定时器(GD32F350保留),则 相关TIMER5的代码需要更改为其它未使用的定时器。
- 11. 编译GD32E23x工程,至此,即可在GD32F3x0系列芯片中使用修改好的GD32E23x固件库进行软件开发。



7.

# GD32E23x 项目底层 Library 替换成 GD32F3x0 Library

## 步骤

本章将使用"GD32E23x\_Firmware\_Library\_V1.1.1\Template"里的工程以及 "GD32F3x0\_Firmware\_Library\_V2.2.0\Template"里的工程做示例。

- 1. 复制 "GD32F3x0\_Firmware\_Library\_V2.2.0\Firmware\CMSIS"下的.h文件替换到 "GD32E23x\_Firmware\_Library\_V1.1.1\Firmware\CMSIS"文件夹下。
- 图 7-1. 把 G32F3x0 固件库中 CMSIS 文件里的.h 文件复制到 GD32E23x 固件库中



 复制 "GD32F3x0\_Firmware\_Library\_ V2.2.0\Firmware\CMSIS\GD\GD32F3x0" 里的 Iclude和Source两个文件夹替换到

"GD32E23x\_Firmware\_Library\_V1.1.1\Firmware\CMSIS\GD\GD32E23x"文件夹下。

### 图 7-2. 把 GD32F3x0 固件库 CMSIS 下的 Include 与 Source 文件复制替换到 GD32E23x 固 件库中去

Firmware →	CMSIS > GD >	GD32F3x0	Firmware →	CMSIS → GD	GD32E23x
^	名称	^	^	名称	^
	Include			Include	
	Source			Source	

3. 复制

"GD32F3x0\_Firmware\_Library\_V2.2.0\Firmware\GD32F3x0\_standard\_peripheral" 里 的Iclude和Source两个文件夹替换到

"GD32E23x\_Firmware\_Library\_V1.1.1\Firmware\GD32E23x\_standard\_peripheral" 文 件夹下。

### 图 7-3. 把 GD32F3x0 固件库 standard\_peripheral 下的文件复制替换到 GD32E23x 固件库 中去

Firmware > GD32F3x0_standard_peripheral	Firmware > GD32E23x_standard_peripheral
<b>^</b> 名称 ^	<b>^</b> 名称 ^
Source	Include Source



 复制"GD32F3x0\_Firmware\_Library\_V2.2.0\Template"下的"gd32f3x0\_libopt.h"到 E230的相应文件夹路径下"GD32E23x\_Firmware\_Library\_V1.1.1\Template"。

### 图 7-4. 把 GD32F3x0 固件库中的 "gd32f3x0\_libopt.h" 文件复制到 GD32E23x 固件库中去



5. 至此,打开GD32E23x固件库中Template文件下的Keil工程。

### 图 7-5. 打开 GD32E23x 固件库中 Template 文件下的 Keil 工程



6. 工程界面上会看到左侧有黄色三角标记,表示原文件已经不存在,原因是前面的文件替换步骤已经把旧文件替换掉。此时只需要把黄色标记的文件全部移除,其中 "gd32e230c\_eval.c"为开发板配套配置,实际项目不使用,可以移除,然后再添加相应的GD32F3x0文件。



### 图 7-6. 移除黄色标记文件并添加新文件



7. 将项目应用的"main.c""systick.c"文件里包含的头文件"#include "gd32e23x.h""修改为"#include "gd32f3x0.h",并删除"#include "gd32e230c\_eval.h""。然后重新选择芯片型号,以及FLASH算法。

图 7-7. 修改 "main.c" 、"systick.c" 文件中的内容





### 图 7-8. 重新选择 GD32F3x0 芯片型号

🕅 Options for Target 'GD32E23x'	×
Device   Target   Output   Listing   User   C/C++ (AC6)   Asm   Linker   Debug   Utilities	
Software Packs	
Vendor: GigaDevice	- I
Device: GD32F330C8 Pack: GigaDevice.GD32F3x0_DFP.3.0.0	
Tooleet: APM URL: http://gd32mcu.com/data/documents/ps	
Search:	
GD32F330 GD32F330C4 GD32F330C4 GD32F330C4 GD32F330C6 GD32F330C6 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330C8 GD32F330F4 GD32F330F4 GD32F330F4 GD32F330F4 GD32F330-ARM Cotex-M4 Core Frequency up to 84 MHz GD32F330G4 Memories	~
OK Cancel Defaults Help	

### 图 7-9. 选择 GD32F3x0 Flash 算法

Options for Target 'GD32E23x'	1		×		
Device   Target   Output   Listing   User   C/C++	(AC6)   Asm   Linker Debu	5 Vtilitie	s 2		
C Use Simulator with restrictions Settings	Use: CMSIS-DAP Debug	ger _	Settings		
Limit Speed to Real-Time					
CMSIS-DAP Cortex-M Target Driver Setup			$\times$		
Debug   Trace   Flash Download	Add Elach Programming Als	orithm			
- Download Function	Add Hash Frogramming Alg	Jonann			~
Erase Full Chip 🔽 Program				La	
Erase Sectors Verify	Description	Hash Size	Device Type	Origin 🔾	<u>^</u>
C Do not Frase E Beset and Bun	GD32F3x0 FMC	128k	On-chip Flash	Device Family Packag	e
	AM29x128 Flash	16M	Ext. Flash 16-bit	MDK Core	
Programming Algorithm	K8P5615UQA Dual Flash	64M	Ext. Flash 32-bit	MDK Core	
	LPC18xx/43xx MX25V8035F	8M	Ext. Flash SPI	MDK Core	
Description Device Size	LPC18xx/43xx S25FL032 SP	4M	Ext. Hash SPI	MDK Core	
	LPC 18xx/43xx S25FL064 SP	8M	Ext. Flash SPI	MDK Core	
	LPC40/x/8x S25FL032 SPIFI	4M	Ext. Flash SPI	MDK Core	
	LPC5460X MT25QL128 SPIFT	16M	Ext. Flash SPI	MDK Core	
	M29W640FB Flash	8M	Ext. Hash 16-bit	MDK Core	
	MIMXR1105x EcoXiP Hash	4M	Ext. Flash SPI	MDK Core	
	RC28F640J3x Dual Flash	16M	Ext. Flash 32-bit	MDK Core	
	525FL1285_V2C	16M	Ext. Flash SPI	MDK Core	
	S29GL064N Dual Flash	16M	Ext. Flash 32-bit	MDK Core	
4	S29JLU32H_BUT Hash	414	Ext. Flash 16-bit	MDK Core	~
	S29JL032H TOP Hash	414	Ext. Hash T6-bit	MDK Core	
Add					
	C:\Users\xianiu.su\AnnData\Loca	al\Am\Packs\	GigaDevice\GD32E	3x0_DEP\3.0.0\Elash\G	D32E3v0 ELM
	10. Tobbie Marja.sa e ppedaa teos	6			0.021.01011.211
OK			_	-	
		Add	Cancel		
rief main function					

由于GD32E230不支持配置优先级组的位长度,在移植GD32F3x0库之后,当项目应用代码中有使用中断的配置时应用代码需要添加 "void nvic\_priority\_group\_set(uint32\_t nvic\_prigroup)"函数。

### 表 7-1. nvic\_priority\_group\_set 函数

/\* set the priority group \*/

void nvic\_priority\_group\_set(uint32\_t nvic\_prigroup);

而且GD32E230仅支持4级抢占优先级,不支持子优先级,所以移植之后,中断使能函数需更 改成 <u>表7-2. nvic irq enable 函数</u>所示的形式:

### 表 7-2. nvic\_irq\_enable 函数

/\* 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. 如项目中使用到TIMER5定时器,由于GD32F3x0剪裁掉此定时器(GD32F350保留),则 相关TIMER5的代码需要更改为其它未使用的定时器。
- 10. 编译项目,如有报错,则根据提示做修改,通常提示为项目上层应该逻辑代码的.C文件 里包含的#include "gd32e23x.h"没有修改为#include "gd32f3x0.h",根据提示修改即可。 至此,项目移植成功,可进行GD32F3x0系列MCU的开发。



8. 版本历史

## 表 8-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2022年3月15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 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