GigaDevice Semiconductor Inc.

GD32W51x 系列 TrustZone 开发指南

应用笔记 AN103



目录			2
图索	JI		3
表索引	;]		4
1.	前言.		5
2.	Trust	·Zone 简介	6
3.	软件ヲ	开发	8
3.1.	Kei	I下开发 TrustZone	8
3.	1.1.	安全工程	8
3.	1.2.	非安全工程	11
3.	1.3.	编译工程	13
3.	1.4.	下载工程	14
3.2.	IAR	2下开发 TrustZone1	4
3.	2.1.	安全工程	15
3.	2.2.	非安全工程	17
3.	2.3.	编译工程	20
3.	2.4.	下载工程	20
4.	代码角	释读	22
4.1.	安全	全工程2	22
4.2.	非多	安全工程	24
5.	版本质	历史2	26



图索引

图	2-1.	内存映射安全属性与 SAU 配置区域的示例	6
图	2-2.	通过 GD-Link Programmer 使能 TrustZone	7
图	3-1.	Keil 工程文件结构	8
图	3-2.	选择安全工程	8
图	3-3.	选择安全模式	9
图	3-4.	选择分散加载文件和 NSC 输出库	9
图	3-5.	安全工程设置调试器1	0
图	3-6.	安全工程设置下载算法和功能	1
图	3-7.	选择非安全工程	1
图	3-8.	选择非安全模式	1
图	3-9.	选择分散加载文件和导入 NSC 库1	2
图	3-10).非安全工程设置调试器1	3
图	3-11	. 非安全工程设置下载算法和功能1	3
图	3-12	2. 工程编译	4
图	3-13	8. BOOT 选项1	4
图	3-14	ト工程下载1	4
图	3-15	5. IAR 工程文件结构1	5
图	3-16	3. 选择安全工程	5
图	3-17	1. 选择安全模式1	6
图	3-18	9. 选择分散加载文件和 NSC 输出库1	6
图	3-19)安全工程设置调试器	7
图	3-20). 选择非安全工程1	8
图	3-21	上选择非安全模式1	8
图	3-22	2. 选择分散加载文件和导入 NSC 库1	9
图	3-23	1. 非安全工程设置调试器	20
图	3-24	L 工程编译2	20
图	3-25	5. BOOT 选项	!1
图	3-26	3. 工程下载	!1
图	4-1.	SAU 配置	22



表索引

表 3-1	I. Project_S.sct 代码	9
表 3-2	2. Project_NS.sct 代码	12
表 3-3	3. gd32w51x_flash_s.icf 代码	16
表 3-4	4. gd32w51x_flash_ns.icf 代码	19
表 4-1	I. 安全 main 代码	22
表 4-2	2. FMC 配置	23
表 4-3	3. SRAM 配置	23
表 4-4	4. 安全代码调用非安全函数	24
表 4-5	5. 安全代码调用非安全函数	24
表 5-1	I. 版本历史	26



1. 前言

本文介绍在 GD32W51x 系列上开发 TrustZone 程序。TrustZone 安全属性是 ARMv8-M 主扩展内容,提供硬件级安全分离和保护,示例代码也分为安全和非安全两部分,将分别介绍 Keil 和 IAR 两种环境下的开发流程和要点。

本文使用 GD32W515P-EVAL 开发板,主控芯片型号为 GD32W515PIQ6,采用带有 TrustZone 功能的 Cortex-M33 内核,主频高达 180MHz。TrustZone 安全功能由 EFUSE_TZCTL 寄存器 的 TZEN 位或者选项字节中的 TZEN 位激活。示例代码在'GD32W515P_EVAL_Demo_Suites / 23_Trustzone'中提供,由 GD32MCU.COM 软件包提供。

适用产品: GD32W51x 系列



2. TrustZone 简介

带有 TrustZone 功能的 M33 内核具有安全和非安全两种状态,两种状态分别使用一组独立的 内核寄存器和 Systick。存储区域分为安全(S)、非安全可调用(NSC)、非安全(NS)区域, 安全区域可访问非安全区域,非安全区域只可访问非安全可调用区域(安全代码中),访问时 提供硬件级安全保护。当 TrustZone 使能时,SAU(安全属性单元)和 IDAU(实施定义属性 单元)用来共同设定内存地址的安全属性。最终安全等级为 SAU 和 IDAU 中定义的较高安全 等级的属性(S>NSC>NS),见<u>图 2-1. 内存映射安全属性与 SAU 配置区域的示例</u>。

IDAU 提供一个硬件的非安全(NS)和非安全可调用(NSC)安全属性划分,软件无法更改,内存映射分区参考GD32W51x_用户手册的1.4存储器映射章节。

SAU 可软件配置最多8个安全属性区域,可以将这些区域的安全(S)、非安全(NS)或非安全可调用(NSC)。



图 2-1. 内存映射安全属性与 SAU 配置区域的示例

GD32W51x 系列使用 TrustZone 保护控制器组(TZPCU)管理外设、SRAM、闪存的安全属性,并提供非法访问控制,配合内核实现 TrustZone 全功能。

在使用这些功能之前,请确保 EFUSE_TZCTL 寄存器的 TZEN 位或者选项字节中的 TZEN 位 已使能。可使用 GD-Link Programmer 使能选项字节的 TZEN 位,见<u>图 2-2. 通过 GD-Link</u> <u>Programmer 使能 TrustZone</u>, GD-Link Programmer 软件可通过 <u>GD32MCU.COM</u>获取。



图 2-2. 通过 GD-Link Programmer 使能 TrustZone

📸 GI	D-Link Progr	ammer 4.	5.10.137	69						
E File(F) Target(T)	GD-Link	Refresh	Edit(E)	View(V)	Help(H)				
GD-Lir	nk Option By	tes Config	juration							Х
F	OntionByt	es 0x400	22040						^	
E	FMC OBR			0x000080	AA					
	TZEN			~	Tru	st zone ei	nable bit			
	SRAM1_R	ST			SRA	M1 reset	enable bit		-	
	SPC			0xAA	Flas	h security	protection v	/alue		
E	FMC_OBUS	SER		0xFFFFFFF	F					
	USER			0xFFFFFFF	F Opt	tion byte I	JSER value			
E	FMC_SECM	ICFG0		0x003F00	00					
	SECM0_E	PAGE		0x3F	End	page of	secure mark	area O		
	SECM0_S	PAGE		0x0	Sta	rt page of	secure mark	c area O		
E	FMC_DMP0)		0x000000	00					
	DMP0EN				DM	P area 0 e	nable			
	DMP0_EP	AGE		0x0	End	l page of I	DMP area 0		~	
T	u plazas rafer	to the chird	leer Marrie	ما الم المراجد	the option	buton				
nps	s, picase refer	to the chip t	iser Manu	ar to modify	uie opuor	noyues.	-	_	~	
						Reset	OK		Car	ncel



3. 软件开发

本文使用 MDK-ARM v5.28.0.0 和 IAR v8.50.9 两种开发环境。请确保已正确安装对应软件和 器件支持包,开发环境软件可通过 KEIL 和 IAR 官网下载,器件支持包(GD32W51x_AddOn) 可通过 GD32MCU.COM 下载。

示例工程中包含安全和非安全两部分,两部分分别使用不同的 Flash 和 SRAM。安全代码使用 FMC 中的前 256KB 空间(起始地址为 0x0C000000,大小 0x40000),内存使用 SRAM0 块 (起始地址为 0x30000000,大小为 0x10000)。非安全代码使用 FMC 中的后 1792KB 空间 (起始地址为 0x08040000,大小 0x1C0000),内存使用 SRAM1 块(起始地址为 0x20010000,大小为 0x10000)。

3.1. Keil 下开发 TrustZone

使用多工程方式可同时开发两个工程,文件结构见<u>图 3-1. Keil 工程文件结构</u>,工程路径为: GD32W51x_Demo_Suites\GD32W515P_EVAL_Demo_Suites\Projects\23_Trustzone\MDK-ARM\GD32W515P_EVAL_TRUSTZONE.uvmpw

图 3-1. Keil 工程文件结构



3.1.1. 安全工程

1、选择安全工程为当前工程,见图3-2.选择安全工程。

图 3-2. 选择安全工程

WorkSpace	
	407
E Project: GD32W515P_EVAL CECUPE	408
GD32W515P_EVAL_S Set as Active Pro	ject ig
Application	410

2、在 Project / Options for Target / Target 选项卡中,选择 Code Generation / Software Model 为 Secure Mode,见图 3-3. 选择安全模式。



图 3-3. 选择安全模式

Options for Target 'GD32W515P_EVAL_SECURE'											
Device Target Output Listing User C/C++ (ACG) Asm Linker Debug Utilities											
GigaDevice GD32W515PIQ6	Code Generation										
Xtal (MHz): 12.0	Software Model: Secure Mode										
Operating system: None											
System Viewer File:	Use MicroLIB 🔲 Big Endian										
GD32W515Px.svd	Floating Point Hardware: Single Precision										
Use Custom File											
Read/Only Memory Areas	Read/Write Memory Areas										
default off-chip Start Size Start	Ip default off-chip Start Size NoInit										
□ ROM1: □ C	□ RAM1: □ □										
□ ROM2: □ C	□ RAM2: □ □										
ROM3:	□ RAM3: □										
on-chip	on-chip										
□ IROM1: 0x8040000 0x1C0000 C	□ IRAM1: 0x20010000 0x10000 □										
IROM2: 0xC000000 0x40000 €	IRAM2: 0x30000000 0x100000 □										
OK Cancel Defaults Help											

3、安全代码和非安全代码使用不同的 Flash 和 SRAM,使用分散加载文件为安全代码分配正确的 Flash 和 SRAM。在 Project / Options for Target / Linker 选项卡中,选择不使用 Target 界面分配地址,选择分散加载文件,并为 NSC 函数设置输出库。见图3-4.选择分散加载文件和NSC 输出库。

输出命令为--import_cmse_lib_out=.\secure_nsclib.o,编译器将在编译时将代码中 __attribute((cmse_nonsecure_entry))标识代码编译到.\secure_nsclib.o文件中,供非安全代码 访问。



Options for Target 'GD32W515P_EVAL_SECURE'	\times
Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities	
Use Memory Layout from Target Dialog X/O Base: Make RW Sections Position Independent R/O Base: Make RO Sections Position Independent R/O Base: Make RO Sections Position Independent R/W Base Dopt Sarder Standard Litracies R/W Base	
✓ Report 'might fail' Conditions as Errors	
Scatter Secure_Code \Project_S.sct	
Misc -import_comse_lib_out=.\secure_nsclib.o	
Linker -cpu=Cottex-M33 *o control -library_type=microlib -strict -scatter "\.\Secure_Code\Project_S.sct"	
OK Cancel Defaults Help	

4、安全代码使用 Flash 地址为 0x0C000000,大小为 0x40000,SRAM 地址为 0x3000000, 大小为 0x10000。其中 NSC 函数分配到 0x0C03E000 地址,大小为 0x00002000。分散加载 文件 Project_S.sct 代码见<u>表 3-1. Project S.sct 代码</u>。

表 3-1. Project_S.sct 代码

LR_IROM1 0x0C000000 0x00040000 {		; load region size_region
ER_IROM1 0x0C000000 0x0003E000	{	; load address = execution address



```
*.o (RESET, +First)
*(InRoot$$Sections)
.ANY (+RO)
.ANY (+RO)
}
RW_IRAM2 0x3000000 0x00010000 { ; RW data
.ANY (+RW +ZI)
}
LR_IROM2 0x0C03E000 0x00002000 {
ER_IROM2 0x0C03E000 0x00002000 { ; load address = execution address
*(Veneer$$CMSE) ; check with partition.h
}
```

5、使用 GD-Link 进行调试和下载,将开发板的 GD-Link 接口与 PC 的 USB 口直接相连,在 Project / Options for Target / Debug 下选择 CMSIS-DAP ARMv8-M Debugger 选项,在 Settings 中识别能到调试器 ID,使用 SW 接口并设置最大时钟为 1MHz。见图 3-5. 安全工程设置调试器。

图 3-5. 安全工程设置调试器



6、在 Flash Download 选项卡下选择 Erase Sectors 并取消选中 Reset and Run,添加针对安全区域的下载算法 GD32W515_Secure_Flash,该下载算法由所安装的器件包提供。见图 36. *安全工程设置下载算法和功能*。



图 3-6. 安全工程设置下载算法和功能

ISIS-DAP ARMv8-M Target [Driver Setup				
bug Trace Flash Downlos	.d				
Download Function LOAD C Erase Full Chip C Erase Sectors C Do not Erase	 ✓ Program ✓ Verify ☐ Reset and Run 	RAM for /	Algorithm 0x20000000	Size: 0x00002000	
Programming Algorithm Description Device Size Device Type Address Range GD32W515_Secure_Flash 2M On-chip Flash 0C000000H - 0C1FFFFFH					
		Start:	0x0C000000	Size: 0x00200000	
	Add	Remove			

3.1.2. 非安全工程

1、选择非安全工程为当前工程,见图3-7.选择非安全工程。

图 3-7. 选择非安全工程



2、在 Project / Options for Target / Target 选项卡中,选择 Code Generation / Software Model 为 Non-Secure Mode,见图 3-8.选择非安全模式。

图 3-8. 选择非安全模式

Options for Target 'GD32W515P_EVAL_NON_SECURE' X											
Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities											
GigaDevice GD32W515PIQ6	Code Generation ARM Compiler: Use default compiler version 6 🔻										
Xtal (MHz): 12.0	Software Model: Non-Secure Mode										
Operating system: None											
System Viewer File:	▼ Use MicroLIB □ Big Endian										
GD32W515Px svd	Floating Point Hardware: Single Precision										
) Use Custom File											
Read/Only Memory Areas	Read/Write Memory Areas										
default off-chip Start Size Startup	default off-chip Start Size NoInit										
E ROM1: 0	□ RAM1: □ □										
□ ROM2: ○	□ RAM2: □ □										
□ ROM3: □ ○	□ RAM3: □ □										
on-chip	on-chip										
IROM1: 0x8000000 0x200000 €	IRAM1: 0x20000000 0x70000 □										
□ IROM2: 0xC000000 0x200000 C	□ IRAM2: 0x3000000 0x70000 □										
OK Cancel Defaults Help											

3、安全代码和非安全代码使用不同的 Flash 和 SRAM,使用分散加载文件为安全代码分配正确的 Flash 和 SRAM。在 Project / Options for Target / Linker 选项卡中,选择不使用 Target 界



面分配地址,选择分散加载文件,并使用安全工程导出的 NSC 函数库。见图 3-9. 选择分散加载文件和导入 NSC 库。

函数库导入命令.../Secure_Project/secure_nsclib.o,可直接调用安全工程中___attribute((cmse_nonsecure_entry))标识代码,这部分代码包含在.\secure_nsclib.o文件中。

图 3-9. 选择分散加载文件和导入 NSC 库

Options for Target 'GD32W515P_EVAL_NON_SECURE'	×
Device Target Output Listing User C/C++ (AC6) Asm Linker Debug Utilities Use Memory Layout from Target Dialog X/O Base: X/O Base:	
Make RN Sections Position Independent R/O Base: Mu0000000 Make RO Sections Position Independent R/W Base 0x2000000 Don't Search Standard Libraries disable Warnings: Image: Conditions as Errors	
Scatter\.\NSecure_Code\Project_NS.sctEdit	
Misc/Secure_Project/secure_nsclib.o	
Linker control string -library_type=microlib -strict -scatter "\\NSecure_Code\Project_NS.sct"	
OK Cancel Defaults Help	

4、非安全代码使用 Flash 地址为 0x08040000,大小为 0x001C0000,SRAM 地址为 0x20010000,大小为 0x10000。分散加载文件 Project_NS.sct 代码见<u>表 3-2. Project_NS.sct</u> <u>代码</u>。

表 3-2. Project_NS.sct 代码

LR_IROM1 0x08040000 0x001C0000 {		; load region size_region
ER_IROM1 0x08040000 0x001C0000	{	; load address = execution address
*.o (RESET, +First)		
*(InRoot\$\$Sections)		
.ANY (+RO)		
.ANY (+XO)		
}		
RW_IRAM1 0x20010000 0x00010000	{	; RW data
.ANY (+RW +ZI)		
}		
}		

5、使用 GD-Link 进行调试和下载,将开发板的 GD-Link 接口与 PC 的 USB 口直接相连,在 Project / Options for Target / Debug 下选择 CMSIS-DAP ARMv8-M Debugger 选项,在 Settings 中识别能到调试器 ID,使用 SW 接口并设置最大时钟为 1MHz。见图 3-10. 非安全工程设置 调试器。



图 3-10. 非安全工程设置调试器

Options for Target 'GD32W515P_EVAL_SECURE'	\times
Device Target Output Listing User C/C++ (ACG) Asm Linker Debug Utilities C Use Simulator with restrictions Settings C Use: [CMSIS-DAP ARMv8-M Debugg Setting] CMSIS-DAP ARMv8-M Target Driver Setup Debug Target Target Driver Setup	ngs X
CMSIS-DAP JTAG/SW Adapter CMSIS-DAP Serial No: Firmware Version: [2.0.0] V Swurden IDCODE Device Name Mundic Detection ID CODE: C Automatic Detection ID CODE: C	love Up Jown
Max Clock: 1MHz Image: Connect in the local connegation of the local connegative of the local connegati	oad
UN Cancel H	stb

6、在 Flash Download 选项卡下选择 Erase Sectors 并取消选中 Reset and Run,添加针对非 安全区域的下载算法 GD32W515_Nsecure_Flash,该下载算法由所安装的器件包提供。见图 3-11. 非安全工程设置下载算法和功能。

图 3-11. 非安全工程设置下载算法和功能

CMSIS-DAP ARMv8-M Target Driver Setup	×
Debug Trace Flash Download	
Download Function C Erase Full Chip ✓ Program C Erase Sectors ✓ Verify C Do not Erase Reset and Run	
Programming Algorithm	
Description Device Size Device Type Address Range GD32W515_Nsecure_Rash 2M On-chip Rash 08000000H - 081FFFFH	
Start: 0x08000000 Size: 0x00200000	
Add Remove	
OK Cancel Help	

3.1.3. 编译工程

由于非安全工程需要使用安全工程生成 secure_nsclib.o 库,因此需要先编译安全工程生成该 库,后编译非安全工程。可先设置安全工程为当前工程,编译安全工程,再设置非安全工程为 当前工程,编译非安全工程。

也可设置 batch setup 选项依次编译两个工程,见图 3-12. 工程编译, 勾选两个工程并点击 Build,编译顺序由工程目录顺序决定,可通过 Project / Manage / Multi-Project Workspace 选项修改。



图 3-12. 工程编译

🍪 🗸 🔛 🙀 GD32W51	Batch Setup	×
👹 Batch Build	Select Project Targets:	
Batch Rebuild	GD32W515P_EVAL_SECURE	Build
Batch Setup	GD32W515P_EVAL_SECURE	Rebuild
	GD32W515P_EVAL_NON_SECURE	Clean

3.1.4. 下载工程

下载代码前需要先正确设置开发板,JP4 的 BOOT0/BOOT1 连接到 L,设置硬件 BOOT选项 从安全代码启动,JP3 连接到 SWD。见图 3-13. BOOT选项。JP21 连接到 USART 用于串口 打印。连接开发板 GD-Link 和 USART 到 PC 机上,并确保软件驱动已正确安装。





含有 TrustZone 功能的工程中,代码总是从安全代码启动,并跳转到非安全代码。因此建议先下载非安全代码,后下载安全代码。见图 3-14. 工程下载。



🛯 🕮 🗼 🕶 🧱 🙀 GD32W515I	e 🖹 🛗 🍪 ▾ 🗮 🙀 GD32W51
ect 📮 🗵	ect 🛛 🗸 🛛
🖥 WorkSpace	WorkSpace
Project: GD32W515P_EVAL_SECU	Project: GD32W515P_EVAL_SECU
🗄 쓚 Project: GD32W515P_EVAL_NON	🖮 🍄 Project: GD32W515P_EVAL_NON

由于 Flash Download 选项中没有勾选 Reset and Run,因此需要手动按下 Reset 按键重启 MCU。重启后可观察到安全和非安全工程分别打印一条信息,并分别控制两个 LED 灯闪烁。

3.2. IAR 下开发 TrustZone

使用多工程方式可同时开发两个工程,文件结构见<u>图 3-15. IAR 工程文件结构</u>,工程路径为: GD32W51x_Demo_Suites\GD32W515P_EVAL_Demo_Suites\Projects\23_Trustzone\EWA RM\GD32W515P_EVAL_TRUSTZONE.eww



图 3-15. IAR 工程文件结构



3.2.1. 安全工程

1、选择安全工程为当前工程,见图3-16.选择安全工程。

图 3-16. 选择安全工程

GD32W515P_EVAL_TRUS	TZONE
-	Options
	Make
H GD32W51x EVA	Compile
–⊕ 🛋 GD32W51x_Peri	Rebuild All
	Clean
- Application	
	Stop Build
He Doc	Add >
GD32W51x_EVA	
F-⊞ ■ GD32W51x_Perip	Remove
Here Cutout	Rename
	Version Control System >
	Open Containing Folder
	File Properties
	Set as Active

2、在 Project/Options/General Options/Target 选项卡中,选择TrustZone/Mode为Secure。 见<u>图 3-17.</u> 选择安全模式。



图 3-17. 选择安全模式

Options for node "GD32W	/515P_EVAL_SECURE"			×
Category: General Options Static Analysis Runtime Checking C/C++ Compiler Assembler Output Converter Custom Build Build Actions	Library Options Target Outpu Processor variant	2 MISR. ut Library Co	A-C:2004 onfiguration	MISRA-C:1998 Library Options 1
Linker Debugger Simulator CADI CMSIS DAP GDB Server I-jet	 Core Device CMSIS-Pack 	Cortex-M33 GD GD32W515 None	×I_TZ	Ē.
J-Link/J-Trace TI Stellaris Nu-Link PE micro ST-LINK Third-Party Driver TI MSP-FET TI XDS	Endian mode © Little Dig 0 BE32 © BE8	Floating poin FPU D registers	None	~
	DSP Extension	d (NEON)	Mode Se	cure V

3、安全代码和非安全代码使用不同的 Flash 和 SRAM,使用分散加载文件为安全代码分配正确的 Flash 和 SRAM。在 Project/Options/Linker/Config 选项卡中,选择分散加载文件,并为 NSC 函数设置输出库。见图 3-18. 选择分散加载文件和 NSC 输出库。

输 出 库 文 件 为 secure_nsclib.o , 编 译 器 将 在 编 译 时 将 代 码 中 __attribute((cmse_nonsecure_entry))标识代码编译到 secure_nsclib.o 文件中,供非安全代码 访问。

|--|

图 3-18. 选择分散加载文件和 NSC 输出库

4、安全代码使用 Flash 地址为 0x0C000000,大小为 0x40000,SRAM 地址为 0x3000000, 大小为 0x10000。其中 NSC 函数分配到 0x0C03E000 地址,大小为 0x00002000。分散加载 文件 gd32w51x_flash_s.icf 代码见<u>表 3-3. gd32w51x_flash_s.icf 代码</u>。

表 3-3. gd32w51x_flash_s.icf 代码

/*-Specials-*/
define symbolICFEDIT_intvec_start = 0x0C000000;
/*-Memory Regions-*/



AN103 GD32W51x 系列 TrustZone 开发指南

define symbol __ICFEDIT_region_ROM_start_ = 0x0C000000; define symbol __ICFEDIT_region_ROM_end__ = 0x0C03FFFF;define symbol __ICFEDIT_region_RAM_start__ = 0x30000000; define symbol __ICFEDIT_region_RAM_end__ = 0x3000FFFF;..... define symbol __region_ROM_NSC_start__ $= 0 \times 0 C 0 3 E 0 0 0;$ define symbol __region_ROM_NSC_end__ $= 0 \times 0 C 0 3 F F F F;$ define region ROM_region = mem:[from __ICFEDIT_region_ROM_start__ to CFEDIT_region_ROM_end__]; define region ROM_NSC_region = mem:[from __region_ROM_NSC_start__ to _region_ROM_NSC_end__]; define region RAM_region = mem:[from __ICFEDIT_region_RAM_start__ to __ICFEDIT_region_RA M_end__]; place in ROM_region { readonly }; place in ROM_NSC_region { section Veneer\$\$CMSE }; place in RAM_region { readw rite, block HEAP, block CSTACK };

5、使用 GD-Link 进行调试和下载,在 Project/ Options/ Debugger 下选择 CMSIS-DAP 选项。 见<u>图 3-19. 安全工程设置调试器</u>。

图 3-19. 安全工程设置调试器

.ategory:	Factory Settings
General Options	
static Analysis	
C/C++ Compiler	Setup Download Images Multicore Extra Ontions Plugins
Assembler	Download Images Manucole Extra Options Plugins
Output Converter	
Custom Build	Driver Kun to
Build Actions	CMSIS DAP v main
Linker	
Simulator	Setup macros
CADI	Lise macro file(s)
CMSIS DAP	
GDB Server	C:\Program Files (x86)\IAR Systems\Embedded Workbench
I-jet	
J-Link/J-Trace	
11 Stellaris Nu Link	
PE micro	Device description file
ST-LINK	
Third-Party Driver	✓ Override default
TI MSP-FET	\$TOOLKIT_DIR\$\CONFIG\debugger\GD\GD32W515vLddf
TI XDS	anoocian_privateora no la conggen (db (db 32103 15x).ddi
	OK Creat

3.2.2. 非安全工程

1、选择非安全工程为当前工程,见图3-20.选择非安全工程。



图 3-20. 选择非安全工程

🗉 💼 GD32W515P EVAL N	ION SECURE-G 🗸 📔 🍡
- I Application	Options
	Make Compile Rebuild All
Le 🖬 🖬 Output	Clean
	C-STAT Static Analysis >
	Stop Build
	Add >
	Remove
	Rename
	Version Control System >
	Open Containing Folder File Properties
	Set as Active

2、在 Project / Options / General Options / Target 选项卡中,选择 TrustZone / Mode 为 Non-Secure。见<u>图 3-21. 选择非安全模式</u>。

ategory:					
eneral Options					
tatic Analysis					
C/C++ Compiler	Library O	intions 2	MISR	A-C:2004	MISRA-C:1998
Assembler	Target	Output	Library C	onfiguration	Library Options 1
Output Converter Custom Build	Processor	variant			
Linker Debunger	Core	C	ortex-M33	\sim	
Ceougger Simulator CADI CMSIS DAP GDB Server I-jet J-Link/J-Trace TI Stellaris	Device	C	GD GD32W515xI_TZ		
	OCMSIS	O CMSIS-Pack			
	Endian mo	de	Floating poir	nt settings	
Nu-Link	Little		EDII	None	
PE micro ST-I INK			TFG .	None	
Third-Party Driver	O RE2	,	D registers	- ~	
TI MSP-FET	0 000	2			
11,05	I BES	() BE8			
	DSP Ext	ension	NEON)	Mode No	ne m-secure V

3、安全代码和非安全代码使用不同的 Flash 和 SRAM,使用分散加载文件为安全代码分配正 确的 Flash 和 SRAM。在 Project/Options/Linker/Config 选项卡中,选择分散加载文件,并 使用安全工程导出的 NSC 函数库。见图 3-22. 选择分散加载文件和导入 NSC 库。

函数库导入\$PROJ_DIR\$\..\Secure_Project\GD32W51x\Exe\secure_nsclib.o,可直接调用安 全工程中__attribute((cmse_nonsecure_entry))标识代码,这部分代码包含在 secure_nsclib.o 文件中。



AN103 GD32W51x 系列 TrustZone 开发指南

图 3-22. 选择分散加载文件和导入 NSC 库

Options for node "GD32W515P_EVAL_NON_SECURE"	×	Options for node 'GD32W	/515P_EVAL_NON_SECURE' X
Caregory: General Options Static Analysis Ruther Checking CC++ Complex Assertion Options Option	Factory Settings Extra Options Output List Insicf III Cancel	Calegory: General Options Static Analysis Runtime Checking C/C++ Complex Assemble: some nam Outputs Build Actions Debugger Smulator CADI CADI CADI CADI CADI CADI CADI CADI	Factory Setting: #define Diagnostics Checksum Encodings Extra Options Config Lbrary Input Optimizations Advanced Output List Automatic runtime library selectic Additional libraries: (one per SPR01_DIRS\-\Secure_Project\(GD32W51x\Exe\secure_nsclib.o Image: Constraint of the secure and the secu

4、安全代码使用 Flash 地址为 0x08040000,大小为 0x001C0000, SRAM 地址为 0x20010000, 大小为 0x10000。分散加载文件 Project_NS.sct 代码见<u>表 3-4.gd32w51x_flash_ns.icf 代码</u>。

表 3-4. gd32w51x_flash_ns.icf 代码

/*-Specials-*/		
define symbolICFEDIT_intvec_start = 0x08040000;		
/*-Memory Regions-*/		
define symbolICFEDIT_region_ROM_start = 0x08040000;		
define symbolICFEDIT_region_ROM_end = 0x081FFFFF;		
define symbolICFEDIT_region_RAM_start = 0x20010000;		
define symbolICFEDIT_region_RAM_end = 0x2001FFFF;		
define region ROM_region = mem:[fromICFEDIT_region_ROM_start to		
ICFEDIT_region_ROM_end];		
define region RAM_region = mem:[fromICFEDIT_region_RAM_start to		
ICFEDIT_region_RA M_end];		
place in ROM_region { readonly };		
place in RAM_region { readw rite, block HEAP, block CSTACK};		

5、使用 GD-Link 进行调试和下载,在 Project/Options/Debugger 下选择 CMSIS-DAP 选项。 见 *图* 3-23. 非安全工程设置调试器。



图	3-23.	非安全工程设置调试器
---	-------	------------

Category:	Factory Settings		
General Options Static Analysis			
Runtime Checking			
C/C++ Compiler Assembler	Setup Download Images Multicore Extra Options Plugins		
Output Converter			
Custom Build	Driver 🖉 Run to		
Build Actions	CMSIS DAP V main		
Debugger			
Simulator	Setup macros		
CADI	Use macro file(s)		
GDB Server	C:\Program Files (x86)\IAR Systems\Embedded Workbench		
I-jet			
J-Link/J-Trace			
11 Stellaris Nu-Link			
PE micro	Device description file		
ST-LINK			
TI MSP-FET			
TI XDS	\$TOOLKIT_DIR\$\CONFIG\debugger\GD\GD32W515xl.ddf		

3.2.3. 编译工程

由于非安全工程需要使用安全工程生成 secure_nsclib.o 库,因此需要先编译安全工程生成该 库,后编译非安全工程。可先设置安全工程为当前工程,编译安全工程,再设置非安全工程为 当前工程,编译非安全工程。

也可设置 Project / Batch build 选项依次编译两个工程,见<u>图 3-24. 工程编译</u>,分别按顺序添加安全和非安全工程,完成后进行 Make 依次编译两个工程。

	Edit Batch Build	×
Batch Build ×	Name	
Batches:	GD32W515_T2	
Edit Delete Close Cancel	Available configurations Configurations to build GD32W515P_EVAL_SECURE - GD32W GD32W515P_EVAL_NON_SECURE - I	
Build Make Clean Rebuild All	(Drag to order)	
	OK Cancel	

图 3-24. 工程编译

3.2.4. 下载工程

下载代码前需要先正确设置开发板,JP4 的 BOOT0/BOOT1 连接到 L,设置硬件 BOOT选项 从安全代码启动,JP3 连接到 SWD。见图 3-25. BOOT 选项。JP21 连接到 USART 用于串口 打印。连接开发板 GD-Link 和 USART 到 PC 机上,并确保软件驱动已正确安装。



图 3-25. BOOT 选项



含有 TrustZone 功能的工程中,代码总是从安全代码启动,并跳转到非安全代码。因此先下载 非安全代码,后下载安全代码。设置非安全工程为当前工程,使用 Project / Download / Download active application 下载代码,见<u>图 3-26. 工程下载</u>。

图 3-26. 工程下载



分别下载完成后需要手动按下 Reset 按键重启 MCU。重启后可观察到安全和非安全工程分别 打印一条信息,并分别控制两个 LED 灯闪烁。



4. 代码解读

4.1. 安全工程

当在工程中 Project / Options for Target / Target 选项卡中,选择 Code Generation / Software Model 为 Secure Mode,则定义__ARM_FEATURE_CMSE 为 3,上电复位后执行 system_gd32w51x.c / SystemInit() / sau_region_config() 函 数,该函数将根据 partition_gd32w51x.h 内容配置 SAU。partition_gd32w51x.h 文件可通过 Configuration Wizard 界面配置并生成 Text Editor 代码。如<u>图 4-1. SAU 配置</u>所示,使能 SAU,并配置四块区域, Region0 配置 NSC 函数地址为非安全可调用,与 Project_S.sct 配置对应。Region1 配置非安全代码的 Flash 地址为非安全,Region2 配置非安全代码的 SRAM 地址为非安全,与 Project_NS.sct 配置对应。Region3 配置外设地址为非安全,安全和非安全代码均可访问。

图 4-1. SAU 配置

partition_gd32w51x.h	
Expand All Collapse All Help Show Grid	
Option	Value
□−Configure Security Attribution Unit (SAU)	v
Enable SAU	V
When SAU is disabled, the atrributes of all memory	Non-Secure
Configure Security Attribution Unit (SAU) Address Regions Attributes	
😑 Configure SAU Region 0	▼
Start Address	0x0C03 E000
End Address	0x0C03 FFFF
Region is	Secure, Non-Secure Callable
😑 Configure SAU Region 1	v
Start Address	0x0804 0000
End Address	0x081F FFFF
Region is	Non-Secure
🖻 Configure SAU Region 2	v
Start Address	0x2001 0000
End Address	0x2001 FFFF
Region is	Non-Secure
🖻 Configure SAU Region 3	v
Start Address	0x4000 0000
End Address	0x4FFF FFFF
Region is	Non-Secure
😟 Configure SAU Region 4	
i	
🗈 Configure SAU Region 6	
Configure SAU Region 7	
Configure behaviour of Floating Point Unit	
Text Editor Configuration Wizard	

安全工程 main 函数中进行安全域的中断使能、Systick 配置、FMC 配置、TZBMPC 配置、LED 配置、USART 配置,最后跳转非安全代码。见表 4-1. 安全 main 代码。

表 4-1. 安全 main 代码





AN103 GD32W51x 系列 TrustZone 开发指南

```
if (SECM_SPAGE != (FMC_SECMCFG0 & 0x3FF) || SECM_EPAGE != ((FMC_SECMCFG0 >>
16) & 0x3FF)){
        fmc_secmark_config();
    }
    if (OU == (FMC_OBR & FMC_OBR_TZEN)){
        /* enable trustzone */
        fmc_trustzone_enable();
    }
    /* configure TZBMPC */
    tzbmpc_config();
    led_config();
    com_config();
    /* setup and jump to non-secure */
    nonsecure_init();
    w hile(1){
    }
}
```

fmc_secmark_config()函数配置 **FMC** 的 0-63 页,即前 256KB 为安全,该区域为安全代码使用。

表 4-2. FMC 配置

```
void fmc_secmark_config(void)
{
    fmc_unlock();
    ob_unlock();
    /* configure mark secure pages */
    ob_secmark_config(SECM_SPAGE, SECM_EPAGE, SECM_INDEX0);
    ob_start();
    w hile(0U ⊨ (FMC_SECSTAT & (FMC_SECSTAT_SECBUSY))){
    }
    ob_reload();
    ob_lock();
    fmc_lock();
}
```

tzbmpc_config()函数配置 SRAM1 为非安全,该区域为非安全代码使用。

表 4-3. SRAM 配置

```
void tzbmpc_config(void)
{
    uint16_t block_number = 0U;
    /* enable TZPCU clock */
    rcu_periph_clock_enable(RCU_TZPCU);
    /* SRAM1 is used to nonsecure code, so all blocks of SRAM1 should set to nonsecure */
```



AN103 GD32W51x 系列 TrustZone 开发指南

for(block_number = 0U; block_number <= TZBMPC1_BLOCK_NUMBER; block_number++){
 tzpcu_tzbmpc_block_secure_access_mode_config(TZBMPC1, block_number,
BLOCK_SECURE_ACCESS_MODE_NSEC);
}</pre>

led_config()函数将 LED1 和 LED2 端口配置为非安全,由非安全代码控制。com_config()函数 将 USART2(串口) 配置为安全,由安全代码控制。

使用 __attribute((cmse_nonsecure_entry)) 标记函数 entry_cb_func_register() 和 non_secure_print()为非安全可调用,非安全代码可直接调用,用于传递非安全函数地址和信息 打印。

若安全代码需要调用非安全函数时,先使用 cmse_nsfptr_create(func_addr)函数注册该地址, 后使用__attribute((cmse_nonsecure_call))转化为非安全函数,最后调用该函数。非安全代码 先调用安全域的 entry_cb_func_register()函数传入函数 toggle_led1 地址,后安全代码调用 nonsecure_func()实现 LED1 翻转,主要代码见<u>表4-4.安全代码调用非安全函数</u>。

表 4-4. 安全代码调用非安全函数

```
#define CMSE_NS_ENTRY __attribute((cmse_nonsecure_entry))
#define CMSE_NS_CALL __attribute((cmse_nonsecure_call))
typedef void CMSE_NS_CALL (*ns_fptr)(void);
ns_fptr nonsecure_func = (ns_fptr)NULL;
......
CMSE_NS_ENTRY void entry_cb_func_register(void *callback)
{
    if(callback != NULL){
    nonsecure_func = (ns_fptr)cmse_nsfptr_create(callback);
    }
}
```

4.2. 非安全工程

非安全代码导入 secure_nsclib.o 库,后可直接调用 NSC 函数。main()函数通过调用 entry_cb_func_register()函数将 toggle_led1 函数地址传给安全代码,配置非安全域 Systick、 LED 端口初始化, while 循环中周期翻转 LED2 并调用 NSC 函数 non_secure_print 打印信息。

表 4-5. 安全代码调用非安全函数

```
extern void entry_cb_func_register(void *callback);
extern void non_secure_print(const char * str);
.....
int main(void)
{
    entry_cb_func_register((void *)toggle_led1);
    /* configure systick*/
```



systick_config(); gd_eval_led_init(LED1); gd_eval_led_init(LED2); w hile(1){ /* toggle LED2 */ gd_eval_led_toggle(LED2); non_secure_print("non-secure code toggle LED2.\r\n"); delay_1ms(1000); }



5.

版本历史

表 5-1. 版本历史

版本号.	说明	日期
1.0	首次发布	2023 年 3 月 3 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 in struments, 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.

© 2023 GigaDevice - All rights reserved