

# GR551x Second Boot示例手册

版本: 2.0

发布日期: 2022-02-20

深圳市汇顶科技股份有限公司

#### 版权所有 © 2022 深圳市汇顶科技股份有限公司。保留一切权利。

非经本公司书面许可,任何单位和个人不得对本手册内的任何部分擅自摘抄、复制、修改、翻译、传播,或将其全部或部分用于商业用途。

#### 商标声明

*GODiX* 和其他汇顶商标均为深圳市汇顶科技股份有限公司的商标。本文档提及的其他所有商标或注册商标,由各自的所有人持有。

#### 免责声明

本文档中所述的器件应用信息及其他类似内容仅为您提供便利,它们可能由更新之信息所替代。确保应用符合技术规范,是您自身应负的责任。

深圳市汇顶科技股份有限公司(以下简称"GOODIX")对这些信息不作任何明示或暗示、书面或口 头、法定或其他形式的声明或担保,包括但不限于针对其使用情况、质量、性能、适销性或特定用途的 适用性的声明或担保。GOODIX对因这些信息及使用这些信息而引起的后果不承担任何责任。

未经GOODIX书面批准,不得将GOODIX的产品用作生命维持系统中的关键组件。在GOODIX知识产权保护下,不得暗中或以其他方式转让任何许可证。

深圳市汇顶科技股份有限公司

总部地址: 深圳市福田保税区腾飞工业大厦B座2层、13层

电话: +86-755-33338828 传真: +86-755-33338099

网址: <u>www.goodix.com</u>

# 前言

#### 编写目的

本文档介绍了如何使用和验证GR551x SDK中的Second Boot示例,旨在帮助用户快速进行二次开发。

#### 读者对象

本文适用于以下读者:

- GR551x用户
- GR551x开发人员
- GR551x测试人员
- 开发爱好者

#### 版本说明

本文档为第6次发布,对应的产品系列为GR551x。

#### 修订记录

版本	日期	修订内容
1.5	2020-08-30	首次发布
1.6	2020-11-25	<ul> <li>"下载固件"章节,新增下载ble_tem_dfu.bin固件前需进行的操作</li> <li>"Second Boot OTA验证"章节,新增使用Second Boot进行OTA升级时、升级后的操作/影响说明</li> <li>"应用固件校验跳转运行验证"章节,新增在Keil中使能Second Boot模式后重新编译 固件的操作步骤</li> </ul>
1.7	2020-12-25	<ul> <li>"下载固件"章节,新增下载second_boot.bin固件前需进行的操作</li> <li>"应用固件校验跳转运行"章节,新增说明</li> <li>新增章节"应用固件睡眠后无法被唤醒"</li> </ul>
1.8	2021-04-19	<ul><li>更新"下载固件"章节的user_config.h中的参数</li><li>新增"自定义固件拷贝升级、校验和跳转实现"章节</li></ul>
1.9	2021-12-29	<ul><li> 删除"硬件连接"章节</li><li> 更新"下载固件"为"固件烧录"</li></ul>
2.0	2022-02-20	基于SDK修改固件名称

# 目录

前言	I
1 简介	1
2 Flash布局	2
3 初次运行	3
3.1 准备工作	
3.2 固件烧录	
3.3 测试验证	5
3.3.1 Second Boot OTA验证	5
<b>3.3.2</b> 应用固件校验跳转运行验证	6
3.3.3 安全验签验证	9
4 应用详解	12
4.1 工程目录	12
4.2 交互流程及代码介绍	12
4.2.1 DFU固件拷贝升级	
<b>4.2.2</b> 应用固件校验跳转运行	15
4.2.3 自定义固件拷贝升级、校验和跳转实现	17
5 常见问题	
5.1 Second Boot安全OTA升级固件失败	
5.2 应用固件睡眠后无法被唤醒	

# 1 简介

Second Boot示例利用蓝牙无线传输、固件二次引导技术,演示了DFU(Device Firmware Update)固件拷贝升级、应用固件(Application Firmware)校验跳转运行、安全验签的功能,实现了灵活、可靠、安全的固件空中升级(OTA,Over The Air)。

- DFU固件拷贝升级:采用双区后台拷贝升级方式,通过蓝牙低功耗(BLE)无线传输实现固件空中升级。
- 应用固件校验跳转运行:对APP Image Info进行匹配,校验并跳转至应用固件(本文以*ble\_tem\_dfu.bin*固件为例)运行。
- 安全验签:为防止DFU固件被篡改和防否认,升级固件时可附加签名信息,Second Boot示例会对其进行验签,验签通过后即可升级。

在进行操作前,可参考以下文档。

名称	描述
GR551x开发者指南	介绍GR551x SDK以及基于SDK的应用开发和调试
GR551x固件升级指南	介绍GR551x的固件升级原理和应用
GR551x OTA示例手册	GR551x空中升级使用说明
GProgrammer用户手册	GProgrammer软件的使用说明,包括固件下载、加密加签等
GR551x固件加密及应用介绍	介绍GR551x的固件加密加签模式
J-Link用户指南	J-Link使用说明: <u>http://www.segger.com/downloads/jlink/UM08001_JLink.pdf</u>
Keil用户指南	Keil详细操作说明: http://www.keil.com/support/man/docs/uv4/

表 1-1 文档参考

### GODiX

# 2 Flash布局

适用于GR551x Second Boot示例的Flash布局如下图所示。



图 2-1 Second Boot的Flash布局

- SCA Info: SCA (System Configuration Area)系统配置区,主要用于存储系统信息和Second Boot固件的Boot Info。
- APP Image Info:应用固件信息区,用于存放应用固件运行的相关设置信息。
- DFU Image Info: DFU固件信息区,用于DFU固件拷贝之前的有效性检查。
- Second Boot: Second Boot示例存放和运行区。
- Bank0:应用固件存放和运行区。
- Bank1: DFU固件缓存区,通过有效性检查的DFU固件将被拷贝至Bank0。
- NVDS(Non-volatile Data Storage): 非易失性数据存储区。

# 3 初次运行

本章主要介绍如何运行和验证GR551x Second Boot示例。

#### 🛄 说明:

SDK\_Folder为GR551x SDK的根目录。

### 3.1 准备工作

运行Second Boot示例之前,请完成以下准备工作。

• 硬件准备

#### 表 3-1 硬件准备

名称	描述						
L Link 丁 目	SEGGER公司推出的JTAG仿真器,如需更多了解,请访问: <u>http://www.segger.com/products/</u>						
J-FIIIK工兴	debug-probes/j-link/						
开发板	GR5515 Starter Kit开发板(以下简称"开发板")						
数据线	Micro USB 2.0数据线						
Android手机	Android 5.0(KitKat)及以上版本						

软件准备

#### 表 3-2 软件准备

名称	描述
Windows	Windows 7/10
J-Link Driver	J-Link驱动程序,下载网址: <u>http://www.segger.com/downloads/jlink/</u>
Keil MDK5	IDE工具,支持MDK-ARM 5.20 及以上版本,下载网址: <u>www.keil.com/download/product/</u>
GProgrammer (Windows)	<b>Programming</b> 工具,位于SDK_Folder\tools\GProgrammer
GRUart (Windows)	串口调试工具,位于SDK_Folder\tools\GRUart
GRToolbox (Android)	BLE调试工具,位于SDK_Folder\tools\GRToolbox

### 3.2 固件烧录

本文涉及*second\_boot.bin*固件和*ble\_tem\_dfu.bin*固件的烧录,在烧录固件前需使用GProgrammer将GR551x SoC中的Flash全擦除。

使用GProgrammer烧录固件至开发板前,还需执行以下操作:

• *ble\_tem\_dfu.bin*固件:需在Keil中使能USE\_SECOND\_BOOT\_MODE(详见3.3.2应用固件校验跳转运行验证)后重新编译固件,再烧录到开发板。

 second\_boot.bin固件:需配置user\_config.h,该文件用于配置Second Boot工程的参数和公 钥Hash值,文件位于:SDK\_Folder\projects\ble\dfu\second\_boot\Src\config。配置 完成后需重新编译固件,再烧录到开发板。

宏	描述
BOOTLOADER_DEFAULT_STRATEGY_ENABLE	<ul> <li>是否使用默认的固件覆盖升级、校验、跳转策略。</li> <li>0:使用自定义的固件覆盖升级、校验、跳转策略</li> <li>1:使用默认的固件覆盖升级、校验、跳转策略</li> </ul>
BOOTLOADER_WDT_ENABLE	是否开启Second Boot看门狗。 • 0:关闭Second Boot看门狗 • 1:开启Second Boot看门狗
BOOTLOADER_OTA_ENABLE	是否开启Second Boot空中升级(OTA)功能 <ul> <li>O: 关闭Second Boot空中升级(OTA)功能</li> <li>I: 开启Second Boot空中升级(OTA)功能</li> </ul>
BOOTLOADER_SIGN_ENABLE	<ul> <li>是否开启Second Boot加签验签方</li> <li>案,当BOOTLOADER_DEFAULT_STRATEGY_ENABLE开启时有效。</li> <li>O:关闭Second Boot验签方案</li> <li>1:开启Second Boot验签方案</li> <li>说明:</li> <li>使用安全验签功能的具体操作请参考3.3.3 安全验签验证章节。</li> </ul>
USER_FW_COMMENTS	应用固件comments定义,当BOOTLOADER_DEFAULT_STRATEGY_ENABLE开启 (值为1)时有效。 查找应用固件Image Info时是通过应用固件comments进行匹配,该定义最 大长度为12 Bytes。 当前默认值为"ble_tem_dfu_"。
APP_FW_RUN_ADDRESS	应用固件运行地址,当BOOTLOADER_DEFAULT_STRATEGY_ENABLE(值为0) 关闭时有效。 说明: 请参考4.2.3 自定义固件拷贝升级、校验和跳转实现章节。

表 3-3 user\_config.h中的参数

使用GProgrammer的具体操作方法请参考《GProgrammer用户手册》。

🛄 说明:

- *second\_boot.bin*固件位于: SDK\_Folder\projects\ble\dfu\second\_boot\build\, 默认运行地 址为0x01004000。
- *ble\_tem\_dfu.bin*固件位于: SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_template\_d fu\build, 默认运行地址为0x01040000。
- 如果修改了*second\_boot.bin*固件和*ble\_tem\_dfu.bin*固件的运行地址,需确保两个固件存储空间不存在冲突。
- 如果使用自定义固件拷贝升级、校验和跳转策略,需要将BOOTLOADER\_DEFAULT\_STRATEGY\_ENABLE设置为0,然后自行实现vendor\_fw\_copy\_update()和vendor\_fw\_verify()。

### 3.3 测试验证

本章节将基于Second Boot OTA、应用固件校验跳转运行和安全验签功能来快速验证Second Boot示例。

### 3.3.1 Second Boot OTA验证

- 1. 在使用GProgrammer将Second Boot固件烧录至开发板前,先将GR551x SoC中的Flash全擦除以确保Flash中不存在任何OTA拷贝任务和应用固件。
- 2. 将Second Boot固件烧录至开发板后,则进入Second Boot OTA等待固件升级(原理参考4.2 交互流程及 代码介绍中的步骤3),此时GRUart串口界面如下图所示。

	Receive Data
Serial Port Setting	Format: <ul> <li>ASCII</li> <li>Hex</li> <li>Show Time</li> <li>Font Size</li> <li>10</li> </ul>
PortName COM121 JLink CDC -	Background:   White  Black  Search
BaudRate 115200 -	APP_D: APP_D: Bootloader info: APP_D: bin_size = 0x0000e810,
DataBits 8 👻	APP_D: check_sum = 0x005b3864, APP_D: load_addr = 0x01004000,
Parity None -	APP_D: run_addr = 0x01004000, APP_D:
StopBits 1	APP_D: There is no incomplete DFU copy task. APP_D: Not found APP img info on the third page, continue to search on the first page
Flow Control RTS DIR	APP_D: Not found APP FW image info. APP_D: Enter bootloader OTA.
	Save Pause Clear
Tx <u>R</u> x Data Size	Send data Single Multi
Tx Count 0 Bytes	Format: OASCII O Hex Loop Period 50 👘 ms 🕅 NewLine
Rx Count 516 Bytes	
Clear	
	file path Browse Send Pause Clear
Port Opened CTS=1 DSR=1 DCD=0	

#### 图 3-1 擦除Flash后进入Second Boot OTA



- 3. 开启手机端的蓝牙功能,打开GRToolbox,在扫描列表中发现"Goodix\_Boot",表明Second Boot固件运行正常。
- 4. Second Boot固件中集成了BLE OTA功能,可参考《GR551x OTA示例手册》"在ble\_app\_template\_dfu中进行固件升级"章节升级固件,升级完成之后会自动跳转至新的应用固件运行。此时GRUart串口界面如下图所示。

DOLLAR I	ort Setting	Format: • ASCII • Hex Show lime Font Size 10
PortName	COM121 JLink CDC	Background:  White Black Search
BaudRate	115200	APP_D: APP_D: Bootloader info: APP_D: bin_size = 0x0000e7f0,
DataBits	8	APP_D: check_sum = 0x005b2e55, APP_D: load_addr = 0x01004000,
Parity	None	APP_D: run_addr = 0x01004000, APP_D:
StopBits	1	APP_D: There is no incomplete DFU copy task. APP_D: Not found APP img info on the third page, continue to search on the first page APP D: Undate user image info
Flow Cont	rol 🗌 RIS 🗌 DII	APP_D: Found the APP firmware on the first page APP_D: Jump to APP FW.
•	Close Port	APP_I: Goodix GR551x SDK V1.6.01 (commit 2826) APP_I: Local Board EA:CB:3E:CF:00:0F. APP_I: Template application example started.
		Save Pause Clear
Tx <u>R</u> x Dat	a Size	Save Pause Clear Single Multi
Tx <u>R</u> x Dat Tx Count	a Size	Save Pause Clear Send data Single Multi Format: ASCII • Hex Loop Period 50 + ms NewLine
Tx <u>R</u> x Dat Tx Count Rx Count	a Size 0 Byte: 1791 Byte:	Save Pause Clear Single Multi Format: ASCII  Hex Loop Period  NewLine
Tx <u>E</u> x Dat Tx Count Rx Count	a Sixe 0 Byte: 1791 Byte: Clear	Save Pause Clear Single Multi Format: ASCII  Hex Loop Period 50  Rs NewLine

#### 图 3-2 升级完成后应用固件成功运行

#### 🛄 说明:

- 使用Second Boot模式进行OTA升级时,在GRToolbox的"固件升级"界面中需勾选"拷贝升级模式"。勾 选后, "Copy Address"指向区域的内容将被覆盖,若配置不当会造成该区域原有信息丢失。
- 使用Second Boot模式进行OTA升级后,在GProgrammer中将无法显示升级后的固件信息。

### 3.3.2 应用固件校验跳转运行验证

- 1. 使用GProgrammer工具先将GR551x SoC中的Flash全擦除以确保Flash中不存在任何OTA拷贝任务和应用 固件。
- 2. 在Keil中修改ble\_app\_template\_dfu示例工程配置后重新编译固件。具体操作如下:

- (1) 进入示例工程目录SDK\_Folder\projects\ble\ble\_peripheral\ble\_app\_template dfu\Keil 5,双击ble\_app\_template\_dfu.uvprojx,在Keil中打开示例工程。
- (3) 在 "Preprocessor Symbols" 面板的 "Define" 参数值中添加 "USE\_SECOND\_BOOT\_MODE",如下图所示。

#### 🛄 说明:

添加的 "USE\_SECOND\_BOOT\_MODE" 需要与前面的宏以 "," 分隔。

vice   Target   Uutput   Listing   User	r C/C++  Asm  Linker Debug	Utilities
Preprocessor Symbols		
Define: GR5515_SK,ENABLE_DFU_S	SPI_FLASH,USE_SECOND_BOOT_MODE	
Undefine:		
Language / Code Generation		Wamings
Execute-only Code	Strict ANSI C	warnings.
Optimization: Level 1 (-01) 💌	Enum Container always int	All Warnings
Optimize for Time	Plain Char is Signed	🔲 Thumb Mode
Split Load and Store Multiple	Read-Only Position Independent	No Auto Includes
One ELF Section per Function	Read-Write Position Independent	C99 Mode
Include Paths Misc Controls	Src\config;\\\\components\app_drive	ers\inc;\\\\compor
Compiler control string	_EVAL -li -g -O1 -apcs=interwork -split_sec ////components/app_drivers/inc -l	ctions -I/Src/platform -I 🔨

#### 图 3-3 使能Second Boot模式

- (4) 保存设置后,点击Keil工具栏中的 🕮 图标,编译示例工程,生成固件bin文件。
- 3. 使用GProgrammer将Second Boot固件和应用固件*ble\_tem\_dfu.bin*烧录至开发板,并设置Second Boot固件作为Startup固件。

	GProgrammer										- 🗆 ×
۲	Firmware										Ø
÷	0x010F FFFF				Firmv	vare File					
٩					User App	Firmware:				G	Export
					(i) Image	e Info					
11					Image Na	me:	second_bo	ot_	Run Address:	0x0100 4000	
					Version:		1		System Clock:		$\sim$
0					Size(Byte):		60480		XQSPI Speed:		$\sim$
-	Unused				SPI Access	Mode:			Boot Delay:	🖲 Yes 🔵 No	
E	NVDS				CheckSum	1:			Check Image:	🖲 Yes 🔵 No	
i	Download				Load Add	ress:	0x0100 400		Code Copy Mode:	🔵 QSPI 💿 XIP	
	🐥 Startup										Update
					🗐 Unfin	ished Ever	nts				
					No.	Action		Description	cocond boot fir hin		~
					2	add		Add and download	ble tem dfu fw.bin		x
					3	startup		Start up second_bo	ot_		×
	0x0100 2000	se	cond_boot_ 🍦								
		Refresh A	dd Delete	Startup							Commit

图 3-4 选中Second Boot固件作为Startup固件

4. GR551x SoC系统启动时检测到存在该应用固件*ble\_tem\_dfu.bin*,且通过有效性检查。会跳转至应用固件存储的起始地址,并开始运行程序,此时GRUart串口界面如下图所示。

			Receive Dat	a						
Serial H	Port Setting		Format:	ASCII	Hex	Show Time 📃	Font Si	ize 10		
PortName	COM121 JLink	CDC -	Background:	White	Black			Search		
BaudRate	115200	Ŧ	APP_D: APP_D: Bootlo APP D: bin si	ader info: ze =	0x0000e7f0.					
DataBits	8	Ŧ	APP_D: check_ APP_D: load_s	sum = ddr =	0x005b2e55, 0x01004000,					
Parity	None	Ŧ	APP_D: run_ad APP_D:	ldr =	0x01004000,					
StopBits	1	Ŧ	APP_D: There APP_D: Not fo APP_D: Update	is no inco ound APP im suser imag	mplete DFU cop g info on the e info.	y task. third page, contin	nue to se	arch on t	he first	page
Flow Cont	trol 🗌 RTS 🗌	DTR	APP_D: Found APP_D: Jump t	the APP fi o APP FW.	rmware on the	first page				
	Close Port		APP_I: Goodix APP_I: Local APP_I: Templa	: GR551x SD Board EA:C te applica	K V1.6.01 (com B:3E:CF:00:0F. tion example s	mit 2826) tarted.		Save	Pause	Clear
Tx <u>R</u> x Dat	ta Size		Send data							
Tx Count	0 1	Bytes	Single Multi Format: O AS	SCII 💿 Hea	K Loop 🕅	Period 50 🚔	ns 🕅 Ne	wLine		
Rx Count	1791 I	Bytes								
	Clear						-			
							Browse	Send	Pause	Clear
Port Opene	d CTS=1 DSR=1	1 DCD=0								

图 3-5 升级完成后应用固件成功运行

### 3.3.3 安全验签验证

Second Boot示例支持OTA固件安全验签功能,用户可根据自身使用场景,选择是否开启该验签功能。如开 启该验签功能,可在Second Boot示例工程目录下的*user\_config.h*中,设置 "BOOTLOADER\_SIGN\_ENABLE = 1"。

在验签前,用户可先使用GProgrammer对应用固件进行加签,完整的加签和验签过程如下:

1. 生成私钥和公钥Hash值

生成签名信息的具体操作请参考《GProgrammer用户手册》"加密加签"章节,原理可参 考《GR551x固件加密及应用介绍》"数字签名技术"章节。

通过GProgrammer生成的用于加密加签的文件如下:



#### 图 3-6 生成的用于加密加签的文件

#### 2. 加签固件

导入product.json文件和应用固件ble\_tem\_dfu.bin,再点击"Sign"按钮,如图 3-7:

Encrypt and Sign				
Product Info:	\product.json			B
Random Number:	<ul> <li>Using Random Number</li> </ul>	Select Number		B
Firmware:	\ble_tem_dfu_fw.bin			ß
			Encrypt Sig	gn

图 3-7 应用固件加签

选择加签文件路径后,即可生成加签应用固件(文件名带有\_sign, *ble\_tem\_dfu\_fw\_sign.bin*),如图 3-8所示:



图 3-8 生成的加签固件

**3.** 将*Public\_key\_hash.txt*中的公钥Hash值复制到Second Boot工程*user\_config.h*的public\_key\_hash数组 后, 重新编译Second Boot固件。



4. 对加签固件进行验证

利用GProgrammer将重新编译生成的Second Boot固件second\_boot.bin和加签

的*ble\_tem\_dfu\_fw\_sign.bin*下载至开发板,并设置Second Boot固件作为Startup固件并运行。Second Boot固件会对*ble\_tem\_dfu\_sign.bin*固件进行校验和验签,若验证成功,则跳转到该应用固件中运行,如下图所示:

	Receive Data	
Serial Port Setting	Format: 💿 ASCII 🔘 Hex	Show Time 📃 Font Size 10
PortName COM83 JLink CDC -	Background: 💿 White 🔘 Black	Search
BaudRate 115200 -	APP_D: found APP img info. APP_D: bin_size = 0x00011d60, APP D: check sum = 0x007007c0.	^ ^
DataBits 8 -	APP_D:         load_addr         = 0x01040000,           APP_D:         run_addr         = 0x01040000,	
Parity None -	APP_D: Signature verify check success. APP_D: check APP img valid.	
StopBits 1 -	APP_D: Jump to APP FW. APP_D:	
Flow Control RIS DIR	APP_I: Goodix GR551x SDK V1.6.02 (commit APP_I: Local Board EA:CB:3E:CF:00:10. APP_I: Template DFU example started.	5079) ⋿
		Save Pause Clear
-IxRx Data Size	Send data Single Multi	
Tx Count 0 Bytes	Format: 🔿 ASCII 💿 Hex Loop 📃 Pe	riod 50 🚔 ms 🔲 NewLine
Rx Count 2487 Bytes		
Clear	file nath	rowce Sand Pause Clear
Part Opened CTS=1 DSP=1 DCD=		cital
Port Opened CTS-I DSK-I DCD-0		

图 3-9 对加签固件进行验证

# 4 应用详解

本章主要介绍Second Boot示例的交互流程及相关代码。

## **4.1** 工程目录

Second Boot示例的源代码和工程文件位于: SDK\_Folder\projects\ble\dfu\second\_boot\Keil\_\_<sup>5</sup>。

双击打开*second\_boot.uvprojx*工程文件,在Keil中查看Second Boot示例工程目录结构,相关文件介绍如 所表 4-1 示。

Group	文件	描述	
gr_profiles	otas.c	OTA Service实现	
user_callback	user_gap_callback.c	GAP Callback实现,如连接、断连、GAP参数更新等	
	user_gatt_common_callback.c	GATT Common Callback实现,如MTU更新	
user_platform	user_periph_setup.c	APP Log和看门狗的配置	
user_app	main.c	main()入口函数	
	user_app.c	实现OTA Service初始化与BLE事件处理	
	user_dfu.c	实现DFU初始化	
	oc_board.h	实现固件有效性检查和固件跳转	
	sign_verify.lib	实现固件签名验证的静态库	
	user_config.h	看门狗、固件验签相关配置	

#### 表 4-1 Second Boot工程文件说明

### 4.2 交互流程及代码介绍

本节介绍Second Boot示例DFU固件拷贝升级、应用固件校验跳转运行的主要代码,便于用户深入了 解Second Boot示例的运行机制。

Second Boot示例运行流程如图 4-1所示。





#### 图 4-1 Second Boot示例运行流程

- 1. 读取DFU Image Info。当Bank1中存在需要拷贝至Bank0的DFU固件,且其有效性校验通过,则执行步骤2;如未通过,则执行步骤3。
- 2. 拷贝Bank1中的DFU固件至Bank0,更新APP Image Info和擦除DFU Image Info后,复位GR551x设备。
- 3. 读取APP Image Info。当BankO中存在应用固件,且其有效性校验通过,则跳转至应用固件运行;如未通过,则执行步骤4。
- 4. 进入Second Boot OTA。OTA结束后,更新APP Image Info,复位GR551x设备。

### 4.2.1 DFU固件拷贝升级

GR551x的应用固件进行OTA时采用双区后台升级方式。

- 1. 将DFU固件存放至Bank1区域,并更新DFU Image Info区域信息。
- 2. 复位GR551x设备后运行Second Boot固件,将Bank1区域中的DFU固件拷贝至Bank0区域运行。

DFU固件拷贝升级主要代码介绍如下。

**路径**:工程目录下的user\_app\user\_boot.c

名称: is\_fw\_need\_copy();

is\_fw\_need\_copy()函数用于读取DFU Image Info,判断是否存在DFU固件拷贝任务。

static bool is\_fw\_need\_copy(void)

#### 应用详解

### GODiX

```
copy load addr = 0;
   hal flash read judge security(IMG INFO DFU ADDR, (uint8 t*)&copy load addr, 4);
   memset((uint8_t*)&dfu_img_info, 0, sizeof(img_info_t));
   hal flash read judge security(IMG INFO DFU ADDR+4, (uint8 t*)&dfu img info,
                             sizeof(img info t));
   if (dfu img info.pattern != 0x4744 || 
       (memcmp(dfu img info.comments, USER FW COMMENTS, strlen(USER FW COMMENTS)) != 0))
   {
      APP LOG DEBUG("There is no incomplete DFU copy task.");
      return false;
   }
   APP LOG DEBUG("-----");
   APP_LOG_DEBUG("copy addr = 0x%08x", copy_load_addr);
   APP LOG DEBUG("DFU fw boot info:");
   log boot info(&dfu img info.boot info);
   APP LOG DEBUG("-----");
   APP LOG DEBUG("There is incomplete DFU copy task.");
   return true;
}
```

```
路径: 工程目录下的user_app\user_boot.c
```

名称: incplt\_dfu\_task\_continue();

incplt\_dfu\_task\_continue()函数用于检查DFU固件有效性,有效性通过之后将DFU固件从Bank1区域拷贝 至Bank0区域,并更新APP Image Info,擦除DFU Image Info,然后复位设备,程序将跳转到新的应用固件并运行,示例代码如下。

```
static void incplt dfu task continue (void)
{
    if (!boot_fw_valid_check(copy_load_addr, &dfu_img_info.boot_info))
    {
       APP LOG DEBUG("DFU FW image valid check fail.");
       return;
    }
    if (copy load addr != dfu img info.boot info.load addr)
        uint32_t copy_size = dfu_img_info.boot_info.bin_size + 48;
       APP LOG DEBUG("DFU FW image start copy.");
        if (sys security enable status check())
        {
           copy_size += 856;
        }
        else
        {
        #if BOOTLOADER SIGN ENABLE
```

```
copy_size += 856;
#endif
}
dfu_fw_copy(dfu_img_info.boot_info.load_addr, copy_load_addr, copy_size);
}
user_img_info_update(&dfu_img_info);
hal_flash_erase(IMG_INFO_DFU_ADDR, CODE_PAGE_SIZE);//clear copy info
hal_nvic_system_reset();
}
```

### 4.2.2 应用固件校验跳转运行

当不存在应用固件OTA拷贝升级任务时,Second Boot会对该应用固件进行有效性检查并跳转至该应用固件 中运行。

路径: 工程目录下的user\_app\user\_boot.c

名称: is\_jump\_user\_fw();

is\_jump\_user\_fw()函数用于跳转到应用固件运行前的检查。

在is\_jump\_user\_fw()函数中,通过读取APP Image Info的comments和实际应用固件的comments(USER\_FW\_COMMENTS)进行对比,判断两者comments是否相同,相同说明Bank0中存在应用固件,进一步校验APP Image Info有效性,有效性通过则可跳转到该应用固件运行。

如comments不相同,说明Bank0中不存在应用固件,此时可通过读取SCA区域索引到的应用固件Image Info的comments,和USER\_FW\_COMMENTS对比,如两者comments相同,再校验该SCA中应用固件Image Info有效性,有效性通过,会将SCA中的应用固件Image Info更新至APP Image Info;如两者comments不同,或者有效性不通过,则不能跳转。

```
static bool is jump user fw(void)
{
   memset((uint8 t*)&app img info, 0, sizeof(img info t));
   hal flash read judge security(IMG INFO APP ADDR, (uint8 t*)&app img info,
                                  sizeof(img info t));
    if ((app img info.pattern == 0x4744) &&\
        (0 == memcmp(app img info.comments, USER FW COMMENTS, strlen(USER FW COMMENTS))))
    {
       APP LOG DEBUG("found APP img info.");
        log_boot_info(&app_img_info.boot_info);
        if (boot fw valid check(app img info.boot info.load addr, &app img info.boot info))
        {
            APP LOG DEBUG("check APP img valid.");
            return true;
        }
    }
   APP LOG DEBUG("Not found APP img info on the third page, continue to search on the first
                  page");
    img_info_t img_info_main;
```

```
for (uint8 t i = 0; i < IMG INFO SAVE NUM MAX; i++)
    {
       fw img info get (BOOT INFO ADDR + 0x40, i, &img info main);
       if (0 == memcmp(img info main.comments, USER_FW_COMMENTS, strlen(USER_FW_COMMENTS)))
        {
            if (boot_fw_valid_check(img_info_main.boot_info.load_addr,
               &img info main.boot info))
            {
               user img info update(&img info main);
               memcpy(&app img info, &img info main, sizeof(img info t));
               APP LOG DEBUG("Found the APP firmware on the first page");
                return true;
            }
        }
    }
   APP LOG DEBUG("Not found APP FW image info.");
   return false;
}
   路径: 工程目录下的user_app\user_boot.c
```

名称1: jump\_user\_fw();

名称2: sec\_boot\_jump();

跳转前需更新热启动Boot Info、设置MSP、重定位向量表。

```
static void jump user fw(void)
{
   APP LOG DEBUG("Jump to APP FW.");
   APP_LOG DEBUG("------
                                           _____
                                                                  _____");
   sec boot jump(&app img info.boot info);
}
static void sec boot jump (boot info t *p boot info)
{
   extern void rom init(void);
   extern void jump_app(uint32_t addr);
   extern boot info t bl1 boot info;
   extern void bl xip dis(void);
   uint16 t enc mode = *(uint16 t*)0x30000020;
   bool mirror mode = false;
   if (p boot info->run addr != p boot info->load addr) //mirror mode
   {
       mirror mode = true;
       if(!enc_mode)
           SET CODE LOAD FLAG();
       memcpy((uint8 t*)p boot info->run addr, (uint8 t*)p boot info->load addr,
              p_boot_info->bin_size);
```

```
}
    if(enc_mode)
    {
        REG(0xA000C578UL) &= \sim0xFFFFFC00;
        REG(0xA000C578UL) |= (p boot info->run addr & 0xFFFFFC00);
    }
    rom init();
    disable irq();
    SysTick->CTRL = 0;
    enable_irq();
    memcpy(&bl1 boot info, p boot info, sizeof(boot info t));
    if(mirror_mode)
    {
        if(enc mode)
        {
            REG(0xa000d470) = ENCRY CTRL DISABLE;
        }
    }
    jump_app(p_boot_info->run_addr);
}
```

#### 🛄 说明:

为了在睡眠唤醒后热启动时,可直接跳转至应用固件,需先将应用固件boot info赋值给全局变量bl1\_boot\_info,即memcpy(&bl1\_boot\_info,p\_boot\_info,sizeof(boot\_info\_t));,此处切勿改动。

### 4.2.3 自定义固件拷贝升级、校验和跳转实现

当不采用默认的固件拷贝升级、校验和跳转策略,则可以将BOOTLOADER\_DEFAULT\_STRATEGY\_ENABLE设置为0,然后自行实现vendor\_fw\_copy\_update()和vendor\_fw\_verify()以完成固件拷贝升级和校验。另外vendor\_fw\_jump()已实现跳转功能,也可自行扩展功能。

以上三个函数位于工程目录下的user\_app\user\_boot.c。

# 5 常见问题

本章描述了在使用及验证Second Boot示例时,可能出现的问题、原因及处理方法。

### 5.1 Second Boot安全OTA升级固件失败

• 问题描述

使用Second Boot OTA升级时,验签失败。

• 问题分析

升级固件验签公钥获取失败。

• 处理方法

确保加签私钥和验签公钥为一对,并将Public\_key\_hash.txt中的公钥Hash值复制到Second Boot工程*user\_config.h*的public\_key\_hash数组。

### 5.2 应用固件睡眠后无法被唤醒

问题描述

应用固件配合Second Boot使用时,当应用固件睡眠后,无法被唤醒。

• 问题分析

Second Boot固件中校验和跳转流程代码被修改,当前应用固件的boot\_info未赋值给全局变量bl1\_boot\_info,导致睡眠唤醒后热启动时失败。

• 处理方法

确保在sec\_boot\_jump()中将应用固件的boot\_info赋值给全局变量bl1\_boot\_info。