



GR551x Fast DFU示例手册

版本： 1.6

发布日期： 2020-12-15

版权所有 © 2020 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

GOODIX 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：www.goodix.com

前言

编写目的

本文档介绍了如何基于Bluetooth Low Energy（BLE）技术，通过GRToolbox（Android）App对GR551x进行固件快速升级，帮助用户深入了解GR551x固件快速升级功能。

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
- 文档工程师

版本说明

本文档为第2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	2020-12-15	更新GRToolbox软件界面截图

目录

前言.....	I
1 简介.....	1
2 Fast DFU概述.....	2
2.1 设备角色.....	2
2.2 交互流程.....	2
3 启用Fast DFU功能.....	4
3.1 ble_dfu_fast工程介绍.....	4
3.2 操作步骤.....	4
4 Fast DFU测试验证.....	7
4.1 准备工作.....	7
4.2 下载Fast DFU固件到开发板.....	7
4.3 手机端快速升级固件.....	8
5 常见问题.....	11

1 简介

设备固件快速升级（Fast Device Firmware Update, Fast DFU）是指目标设备通过蓝牙无线传输的方式下载目标固件后进行自行校验并更新升级。

相较于同样采用无线传输方式下载目标固件并更新的固件空中升级方式（OTA），GR551x Fast DFU提供Fast OTA Service，用于快速升级功能的控制与交互，实现快速升级特性，同时优化数据读写方式，极大地缩短了固件升级所需时间。

Fast DFU是一种升级速度更快、操作体验更佳的更优无线传输固件升级方案。本文档主要介绍Fast DFU的实现原理与使用方式。

在进行操作前，建议参考以下文档：

表 1-1 文档参考

名称	描述
GR551x开发者指南	GR551x软硬件介绍、快速使用及资源总览
Bluetooth Core Spec v5.1	Bluetooth官方标准核心规范5.1: https://www.bluetooth.com/specifications/bluetooth-core-specification/
J-Link用户指南	J-Link使用说明: www.segger.com/downloads/jlink/UM08001_JLink.pdf
Keil用户指南	Keil详细操作说明: www.keil.com/support/man/docs/uv4/
GR551x BLE Stack用户指南	GR551x低功耗蓝牙协议栈各层的基本功能介绍

2 Fast DFU概述

本章介绍了Fast DFU的基础概念。

2.1 设备角色

Fast DFU设备角色包括：

- 控制设备（主机端）：向目标设备发送升级数据，比如手机。
- 目标设备（设备端）：接收来自控制设备的升级数据，比如手环。

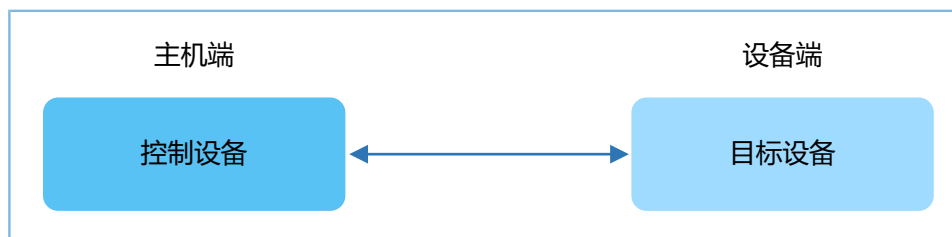


图 2-1 设备角色

2.2 交互流程

为加快数据传输速率，主机端和设备端之间仅支持固件信息交互、固件数据交互、固件校验和交互。

交互流程如[图 2-2](#)所示。

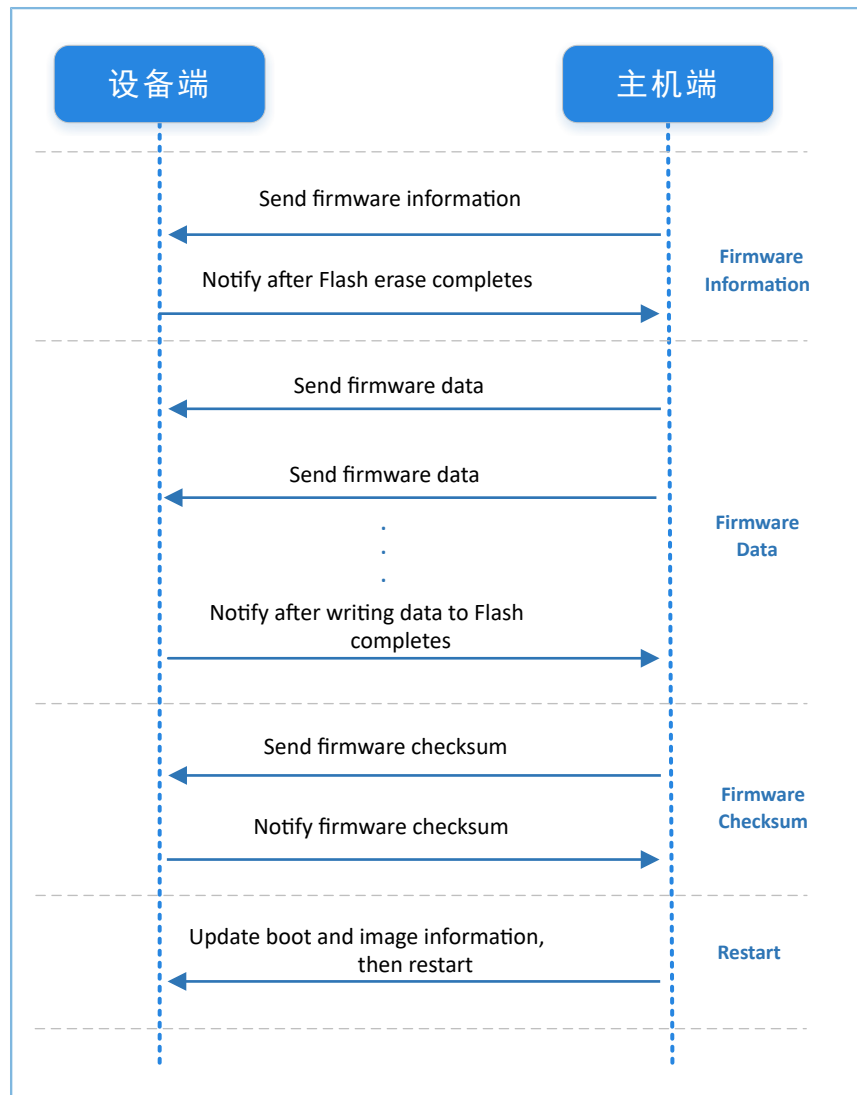


图 2-2 快速升级交互流程

1. 固件信息交互：主机端向设备端发送目标固件信息，设备端接收到固件信息后，一次性擦除待升级固件需使用的Flash空间后再通知主机端。
2. 固件数据交互：主机端连续将目标固件数据发送给设备端，设备端收到固件数据后会先缓存在本地Ring Buffer中。当Ring Buffer中的数据达到设定长度时，设备端会将其一次写入到Flash中。当全部固件数据都写入到Flash后，会通知主机端Flash写入已完成。
3. 固件校验和交互：主机端和设备端互相发送各自计算的升级固件校验和，并进行对比检查。

若主机端与设备端的校验和对比结果一致，则更新Boot和Image Information，并重启设备端。

3 启用Fast DFU功能

本章介绍如何启用GR551x芯片的Fast DFU功能。

Fast DFU功能封装在SDK_Folder\components\libraries\fast_dfu模块中。用户只需调用相应的API接口，就能在应用中添加Fast DFU功能。下面以ble_dfu_fast工程为例进行说明。

说明:

SDK_Folder为GR551x SDK的根目录。

3.1 ble_dfu_fast工程介绍

ble_dfu_fast示例工程位于：SDK_Folder\projects\ble\dfu\ble_dfu_fast，其中工程文件位于Keil_5文件夹。

双击打开ble_dfu_fast.uvprojx工程文件，在Keil中查看ble_dfu_fast示例的工程目录结构，主要文件说明如表3-1所示。

表 3-1 ble_dfu_fast文件说明

Group	文件	描述
gr_libraries	fast_dfu.c	Fast DFU Service初始化，包含快速升级协议控制与调度
gr_profiles	ble_prf_utils.c	Profile相关操作工具
	fast_otas.c	Fast 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交换
user_platform	user_periph_setup.c	串口、设备地址、电源管理模式、Fast DFU的配置
user_app	main.c	main()入口函数
	user_app.c	Fast OTA Profile注册及逻辑处理

3.2 操作步骤

在应用工程中执行以下操作，添加Fast DFU功能。

1. 初始化Fast DFU功能。

路径：工程目录下user_platform\user_periph_setup.c

名称：app_periph_init();

在该函数中调用fast_dfu_init()接口，初始化Fast DFU功能。

```
void app_periph_init(void)
{
    SET_BD_ADDR(BD_ADDR_NVDS_TAG, BD_ADDR_LENGTH, s_bd_addr);
    bsp_uart_init();
    user_log_init();
}
```



```
fast_dfu_init(NULL, &dfu_pro_call);
pwr_mgmt_mode_set(PMR_MGMT_ACTIVE_MODE);
APP_LOG_DEBUG("DFU FAST DEMO START");
}
```

表 3-2 fast_dfu_init()接口的入参说明

参数	说明	取值
p_dfu_func	是否需要操作外部Flash。 如需操作外部Flash，则需传入实现的外部Flash操作（读、写和擦除）接口指针。	<ul style="list-style-type: none">是：将该参数赋值为用户实现的外部Flash操作接口否：将该参数赋值为“NULL”
p_state_callback	是否需要在应用层监测升级状态。	<ul style="list-style-type: none">是：将该参数赋值为用户实现的Fast DFU状态处理Callback否：将该参数赋值为“NULL”

说明:

- 示例代码中，将fast_dfu_init()接口的入参“p_dfu_func”赋值为“NULL”、“p_state_callback”赋值为自定义的Fast DFU状态处理回调函数“dfu_pro_call”。
- 若需操作外部Flash，则需实现fast_dfu.h中的“fast_dfu_func_t”结构体，并将该结构体实例的指针赋值给“p_dfu_func”，其中fast_dfu.h位于SDK_Folder\components\libraries\fast_dfu中。
当使用SPI操作外部Flash时，fast_dfu.h中的ONCE_WRITE_DATA_LEN应小于4096 Bytes。

2. 初始化Fast DFU服务。

路径：工程目录下user_app\user_app.c

名称：services_init();

在该函数中调用fast_dfu_service_init()接口，实现Fast DFU服务注册。

```
static void services_init(void)
{
    fast_dfu_service_init();
}
```

3. 在main()函数的while(1)中调用fast_dfu_schedule()接口，实现Fast DFU功能调度。

路径：工程目录下user_app\main.c

名称：main();

```
int main(void)
{
    // Initialize user peripherals.
    app_periph_init();

    // Initialize BLE Stack.
```

```
ble_stack_init(&s_app_ble_callback, &heaps_table);

// loop
While (1)
{
    fast_dfu_schedule();
    pwr_mgmt_schedule();
}
}
```

4 Fast DFU测试验证

GR551x SDK提供了Fast DFU固件升级示例。本章将介绍如何使用GR5515 Starter Kit开发板（以下简称GR5515 SK开发板）与Android手机进行交互，测试并验证Fast DFU功能。

4.1 准备工作

进行Fast DFU功能测试前，需要完成以下准备工作。

- 硬件准备

表 4-1 硬件准备

名称	描述
J-Link工具	SEGGER公司推出的JTAG仿真器，如需更多了解，请访问 www.segger.com/products/debug-probes/j-link/
开发板	GR5515 Starter Kit开发板
Android手机	用于安装GRToolbox App，进行Fast DFU功能测试

- 软件准备

表 4-2 软件准备

名称	描述
Windows	Windows 7/Windows 10
J-Link Driver	J-Link驱动程序，下载网址： www.segger.com/downloads/jlink/
Keil MDK5	IDE工具，下载网址： www.keil.com/download/product/
GRToolbox（Android）	GR551x BLE调试工具，位于SDK_Folder\tools\GRToolbox
GProgrammer（Windows）	GR551x Programming工具，位于SDK_Folder\tools\GProgrammer

4.2 下载Fast DFU固件到开发板

GR551x SDK中包含Fast DFU升级示例及预编译固件。进行Fast DFU功能测试时，需先下载Fast DFU固件到GR5515 SK开发板，再使用手机端的GRToolbox App连接开发板进行目标固件快速更新升级。

下载Fast DFU固件`ble_dfu_fast_fw.bin`至开发板。具体操作方法，请参考《GProgrammer用户手册》。

说明:

- `ble_dfu_fast_fw.bin`的Load Address和Run Address为0x01002000。
- `ble_dfu_fast_fw.bin`位于SDK_Folder\projects\ble\dfu\ble_dfu_fast\build\。

开启手机端的蓝牙功能，打开GRToolbox，若在设备扫描列表中存在设备名“Goodix_Fast_DFU”，表明固件Demo运行正常。

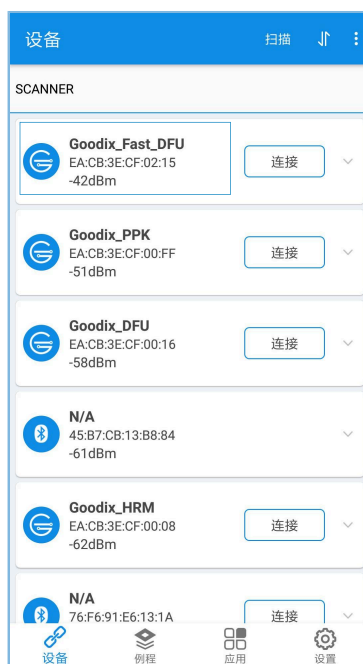


图 4-1 发现Goodix_Fast_DFU

说明:

“Goodix_Fast_DFU” 为在Fast DFU固件`ble_dfu_fast_fw.bin`中定义的设备广播名称。

4.3 手机端快速升级固件

本章介绍如何在Android手机端使用目标固件`ble_tem_dfu_fw.bin`进行设备快速升级。

说明:

- `ble_tem_dfu_fw.bin`的Load Address和Run Address为0x01040000（须避免与Fast DFU固件`ble_dfu_fast_fw.bin`的Load Address和Run Address发生冲突）。
- `ble_tem_dfu_fw.bin`位于：SDK_Folder\projects\ble\ble_peripheral\ble_app_template_dfu\build\。

1. 拷贝目标固件。

将目标固件`ble_tem_dfu_fw.bin`拷贝到手机根目录下的\Goodix\SaveData目录。

2. 载入目标固件。

(1) 打开GRTtoolbox App后选择“应用”，点击应用列表中的“Fast DFU”，进入Fast DFU操作界面。

(2) 点击“升级固件”区域的“选择文件”按钮，选择步骤1中拷贝的`ble_tem_dfu_fw.bin`文件。

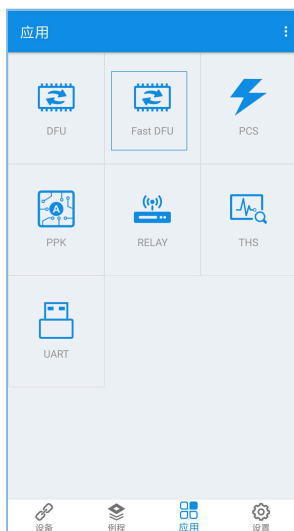


图 4-2 选择“Fast DFU”应用



图 4-3 选择目标固件文件



图 4-4 查看固件Info

文件载入成功后，将在界面显示升级文件`ble_tem_dfu_fw.bin`的固件Info信息（如图 4-4）。

3. 通过Bluetooth连接手机与GR5515 SK开发板。

说明:

进行该操作前，请先确保手机蓝牙功能已开启。

点击“连接”按钮，在扫描到的设备列表中选择“Goodix_Fast_DFU”（下载了Fast DFU固件的GR551x SK开发板的广播名，详见[4.2 下载Fast DFU固件到开发板](#)）进行设备连接，如图 4-5所示。

若界面提示“连接成功”，表明手机与GR5515 SK开发板连接成功（如图 4-6所示）。



图 4-5 选择连接设备

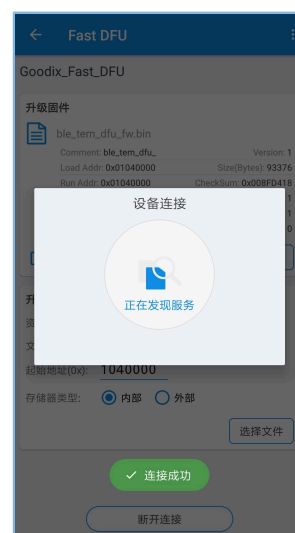


图 4-6 连接成功

4. 点击“升级”按钮（如图 4-7所示），开始升级固件。

当进度显示为100%时，升级完成。



图 4-7 执行升级操作



图 4-8 升级中

5. 验证固件快速升级是否成功。

升级完成后，设备端会自动启动并运行目标固件。目标固件（即升级后的固件）成功运行后会发起名为“Goodix_Tem_DFU”的广播。因此，进入GRTtoolbox的“设备”界面，下拉刷新设备列表（或点击右上角的“扫描”按钮），查看扫描到的设备列表中是否存在广播设备“Goodix_Tem_DFU”。

若存在广播设备“Goodix_Tem_DFU”（如图 4-9所示），则表明固件快速升级成功。

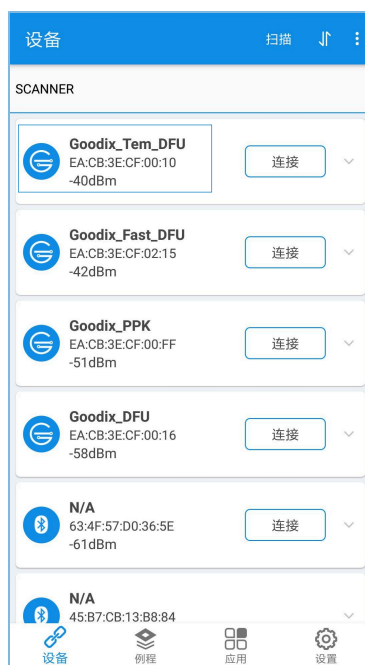


图 4-9 发现广播设备

5 常见问题

本章介绍Fast DFU过程中可能出现的问题、原因及处理方法。

- 问题描述
 - 目标固件升级失败。
- 问题分析
 1. 如果在2.4 GHz干扰较严重的环境中使用Fast DFU功能升级固件，蓝牙连接可能断开而导致升级失败。
 2. 目标固件运行地址与Fast DFU固件运行地址存在冲突，升级过程中出现运行错误导致升级失败。
- 处理方法
 1. 确保Fast DFU功能运行环境中不存在严重干扰。
 2. 确保目标固件运行地址与Fast DFU运行地址无冲突。