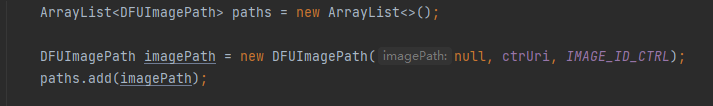
添加依赖库：

1. 把SifliDFU-release.aar复制到项目中的“app/libs”目录下
2. 在app的build.gradle中的依赖项添加

implementation files('libs/SifliDFU-release.aar')

使用：

1. 新建 DFUImagePath arraylist来传升级文件路径



1. DFUImagePath

Path和Uri，只需要指定任意一个即可，

文件类型，当前文件的类型，

类型定义在Protocol.java下，其中NAND\_RES为一个zip文件，其他类型的文件都是bin文件。

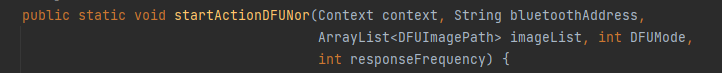
1. 开始升级

NOR FLASH类型的手表，使用startActionDFUNor或者startActionDFUNorExt来升级，其中手表版本solution 1.0.4 patch以后的版本，推荐使用startActionDFUNorExt，优化了开始升级的流程，使用更简单。稍早之前的版本只能使用startActionDFUNor。

NAND FLASH类型的手表，使用startActionDFUNand。

1. 参数：

startActionDFUNor

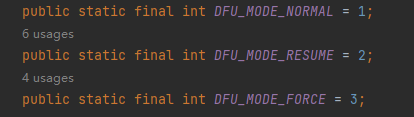


context: 调用者context

bluetootthAddress: 待升级设备的蓝牙地址

imageList：升级文件路径列表

DFUMode：在protocol.java中定义



Normal: 常规升级

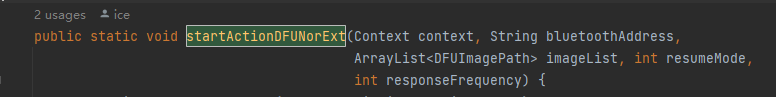
Resume：使用续传

Force：强制升级

当设备上一次没有升级完成，或者设备当前处在OTA模式，只能使用resume或者force模式升级；如果使用normal将返回错误10，OTA\_ONGOING

responseFrequency：下载中的回复频率，即每传输多少包手表回复一包，会影响到下载速度。Android中推荐设置为0，不回复，可以达到最快速度

startActionDFUNorExt



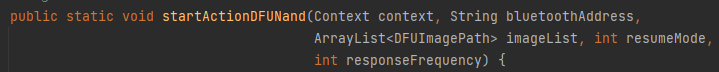
其他参数同startActionDFUNor

ResumeMode:

设置为0时，OTA将始终重新开始传输

设置为1时，SDK会自动判断续传条件，会在可以续传的时候尝试续传，不能续传的时候也会重新开始传输。

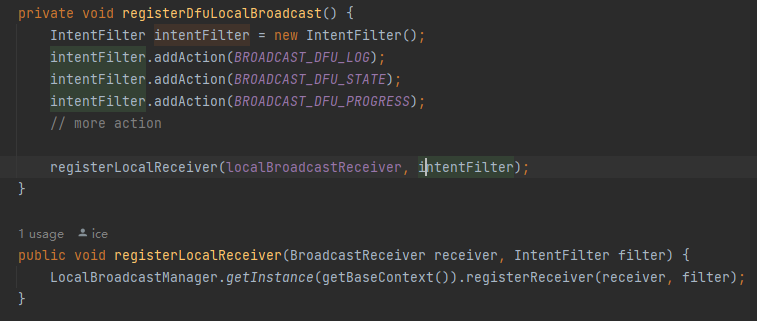
startActionDFUNand



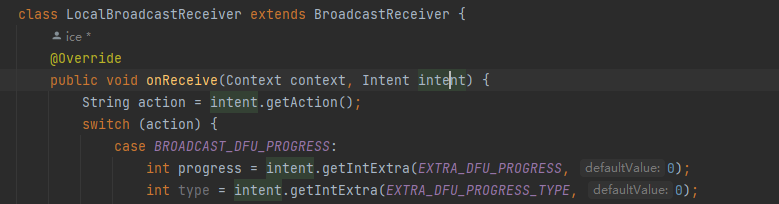
同startActionDFUNorExt

获取结果

注册本地广播，获取结果，进度



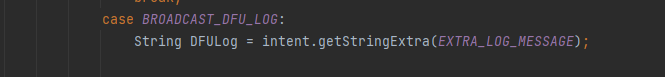
进度：



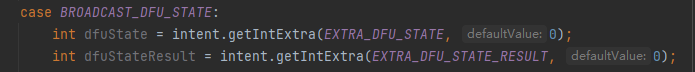
Progress是0~100的整数

如果是nand升级，资源(zip)和hcpu(bin)的进度是分开的，用PROGRESS\_TYPE\_IMAGE和PROGRESS\_TYPE\_FILE区分。

LOG



状态：



判断升级结束

dfuState 的值为 DFU\_SERVICE\_EXIT

dfuStaeResult的值为0代表升级成功，其他值为失败