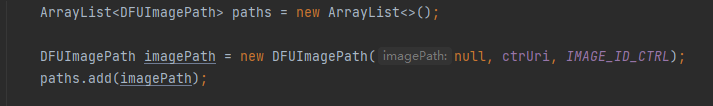
To add the dependency library:

1. Copy the SifliDFU-release.aar to the “app/libs” directory in the project
2. Add the following dependencies to the app’s build.gradle:

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

Usage:

1. Create a new DFUImagePath arraylist to upload the upgrade file path



1. DFUImagePath

Path and Uri, just specify either one.

File type, the type of the current file.

The type is defined under Protocol.java, where NAND\_RES is a zip file and all other types are bin files.

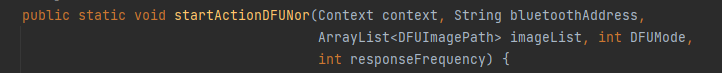
1. Start the upgrade

NOR FLASH type watches, use startActionDFUNor or startActionDFUNorExt to upgrade, of which the watch version solution 1.0.4 patch and later versions, it is recommended to use startActionDFUNorExt, which optimizes the process of starting the upgrade and is easier to use. Earlier versions can only use startActionDFUNor.

For NAND FLASH type watches, use startActionDFUNand.

1. Parameters：

startActionDFUNor

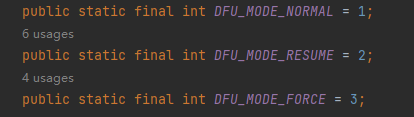


context: The caller context

bluetootthAddress: The Bluetooth address of the device to be upgraded

imageList: The list of upgrade file paths

DFUMode: Defined in protocol.java



Normal: Normal upgrade

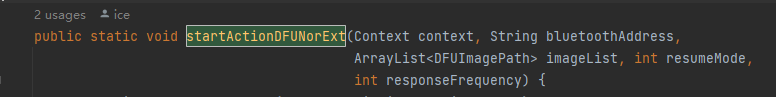
Resume: Use resumes

Force: Force upgrade

If the device did not finish upgrading last time, or the device is currently in OTA mode, it can only use resume or force mode to upgrade; If normal is used, it will return error 10, OTA\_ONGOING

responseFrequency: The response frequency in the download, that is, the watch reply times for packets transferred, which will affect the download speed. In Android, it is recommended to set it to 0, no reply, and can reach the fastest speed

startActionDFUNorExt



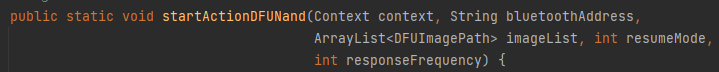
The other parameters are the same as startActionDFUNor

ResumeMode:

When set to 0, the OTA will always restart the transfer

When set to 1, the SDK will automatically determine the resume conditions and will try to resume when it can, and will restart the transfer when it cannot be renewed.

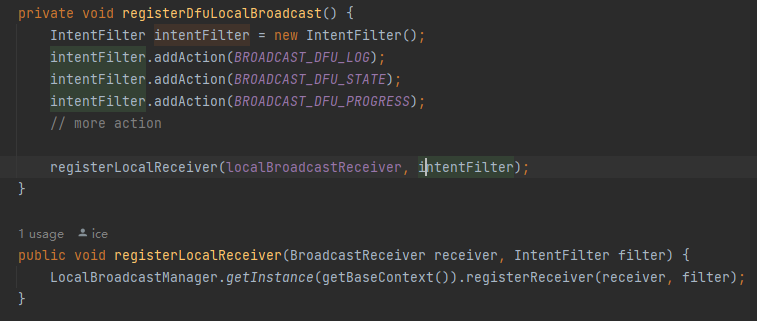
startActionDFUNand



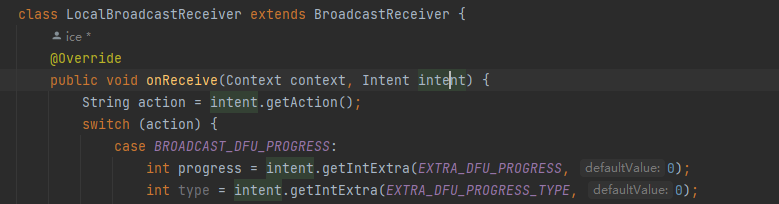
The same as startActionDFUNorExt

Get Results

Register local broadcast, get results, progress



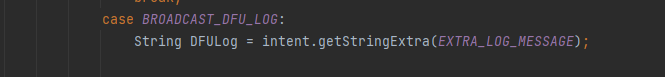
Progress:



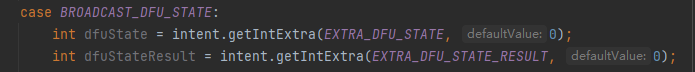
Progress is an integer from 0 to 100

In the case of NAND upgrades, the progress of resources (zip) and hcpu (bin) are separated, distinguished by PROGRESS\_TYPE\_IMAGE and PROGRESS\_TYPE\_FILE.

LOG



State:



Determine the end of the upgrade

The value of dfuState is DFU\_SERVICE\_EXIT

The value of dfuStaeResult is 0 for successful upgrade, other values are failure