

Fibocom Upgrade_Tool 升级指南 _Linux&Android

V1.3

版权声明

版权所有©2022 深圳市广和通无线股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

商标申明

 为深圳市广和通无线股份有限公司的注册商标，由所有人拥有。

联系方式

公司网址：<https://www.fibocom.com/>

总部地址：深圳市南山区西丽街道西丽社区打石一路深圳国际创新谷六栋 A 座 10-14 层

总机：+86 755-26733555

目录

目录

修订记录	2
1 引言	3
1.1 适用型号	3
2 编译	4
2.1 工具包介绍	4
2.2 编译	4
2.2.1 Linux 版本 Upgrade_tool	4
2.2.2 Android 版本 Upgrade_tool	5
3 升级	7
3.1 本地升级	7
3.2 远程升级	7
3.3 查看升级结果	8
4 升级参数说明	9
5 还原 NV	10
5.1 MDM9205 TX 模块还原 NV	10
5.2 MDM9x07 LE 模块还原 NV	10

修订记录

V1.3 (2022-03-27)	支持不带 erase 字段的 rawprogram_nand_XX.xml 添加-r 支持 mdm9x07 LE 系列 NV 自动还原功能 9205 TX 模块 NV 还原说明 下载成功删除下载日志。
V1.2 (2021-12-27)	新增 PCIE 端口说明 支持指定固件包路径进行升级 新增升级结果查看
V1.1 (2021-11-01)	添加适用型号 修改附录 A 章节
V1.0 (2021-09-14)	初始版本

1 引言

Upgrade_tool 用于 Linux 和 Android 主机对高通平台模块进行固件升级。

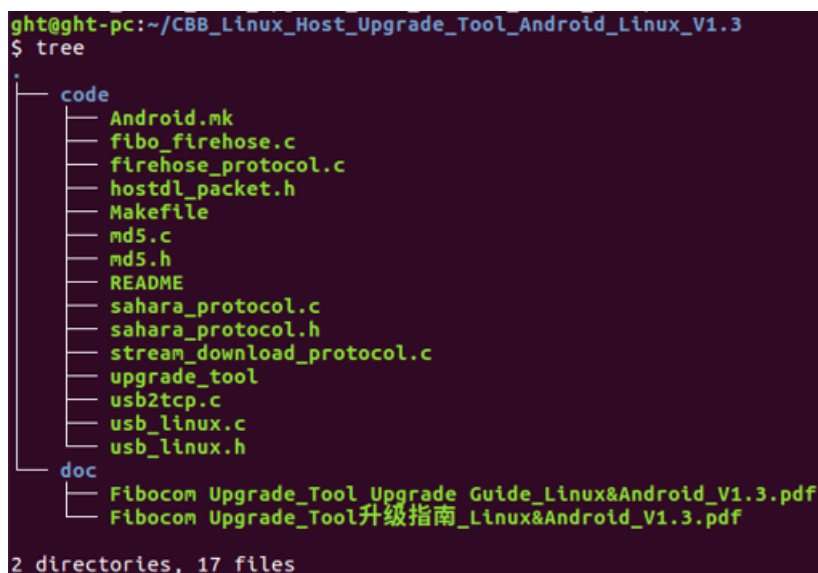
1.1 适用型号

序号	模块型号	说明
1	FM150/ FG150	SDX55 LE 平台
2	FM160	SDX65 LE 平台
3	FG10x/FM10x	SDX12 LE 平台
4	NL668/ MC116/LC116	MDM9x07 LE 平台
5	MA510/ MC109/MC100E	MDM9205 TX 平台

2 编译

2.1 工具包介绍

- code: 源码
- doc: 指导文档
- code/Makefile: Linux 环境编译配置文件
- code/Android.mk: Android 环境编译配置文件
- doc: 中英文指导文档



```
ght@ght-pc:~/CBB_Linux_Host_Upgrade_Tool_Android_Linux_V1.3
$ tree
.
├── code
│   ├── Android.mk
│   ├── fibo_firehose.c
│   ├── firehose_protocol.c
│   ├── hostdl_packet.h
│   ├── Makefile
│   ├── md5.c
│   ├── md5.h
│   ├── README
│   ├── sahara_protocol.c
│   ├── sahara_protocol.h
│   ├── stream_download_protocol.c
│   ├── upgrade_tool
│   ├── usb2tcp.c
│   ├── usb_linux.c
│   └── usb_linux.h
└── doc
    ├── Fibocom Upgrade_Tool Upgrade Guide_Linux&Android_V1.3.pdf
    └── Fibocom Upgrade_Tool升级指南_Linux&Android_V1.3.pdf
2 directories, 17 files
```

2.2 编译

2.2.1 Linux 版本 Upgrade_tool

- 设置交叉编译工具

Makefile 默认使用系统 gcc，如果需要 arm gcc，需在 Makefile 设置 CROSS_COMPILE 变量。

如下图所示：

```

ght@ght-pc: ~/CBB_Linux_Host_Upgrade_Tool_Android_Linux_V1.3/code
#arm gcc tool
#CROSS_COMPILE = ~/ARM_Linux_GCC/bin/arm-none-linux-gnueabi-

CC = ${CROSS_COMPILE}gcc

CFLAGS = -g -Wall
CFLAGS += -Wno-format-overflow
LD_LIBRARY = -lpthread -ldl

INCLUDE =
SOURCES = \
    fibo_firehose.c \
    firehose_protocol.c \
    sahara_protocol.c \
    usb_linux.c \
    md5.c \
    usb2tcp.c \
    stream_download_protocol.c

linux: clean
    ${CC} ${CFLAGS} -s ${SOURCES} -o upgrade_tool ${LD_LIBRARY}

debug: clean
    ${CC} ${CFLAGS} -DFH_DEBUG -s ${SOURCES} -o upgrade_tool ${LD_LIBRARY}

clean:
    rm -f upgrade_tool

~
"Makefile" [dos] 29L, 557C                                29,0-1    All

```

- 编译

将下载工具代码放到 Linux 主机，cd 进入代码目录，执行 make，编译成功会生成 upgrade_tool。

如下图所示：

```

ght@ght-pc:~/CBB_Linux_Host_Upgrade_Tool_Android_Linux_V1.3/code
$ make
rm -f upgrade_tool
gcc -g -Wall -Wno-format-overflow -s fibo_firehose.c firehose_protocol.c sahara_protocol.c usb_linux.c md5.c usb2tcp.c stream_download_protocol.c -o upgrade_tool -lpthread -ldl
ght@ght-pc:~/CBB_Linux_Host_Upgrade_Tool_Android_Linux_V1.3/code
$ ls
Android.mk      hostdl_packet.h  md5.h           sahara_protocol.h  usb2tcp.c
fibo_firehose.c  Makefile         README          stream_download_protocol.c  usb_linux.c
firehose_protocol.c  md5.c           sahara_protocol.c  upgrade_tool       usb_linux.h
ght@ght-pc:~/CBB_Linux_Host_Upgrade_Tool_Android_Linux_V1.3/code

```

2.2.2 Android 版本 Upgrade_tool

1. 将解压后的下载工具代码放到 Android 代码目录。
2. 执行 `source build/envsetup.sh`
3. 执行 `Lunch` 选择对应产品的编译选项。
4. 执行 `mmm CBB_Linux_Upgrade_Tool_V1.0`
5. 编译成功，会生成 upgrade_tool，编译 log 会打印 upgrade_tool 生成路径。

例如: out/target/product/msm8953_64/system/bin/upgrade_tool。

```
ght@ubuntu-230:~/android
$ source build/envsetup.sh
=====BSP_SELECT_ID=====
including device/qcom/common/vendorsetup.sh
including vendor/qcom/proprietary/common/vendorsetup.sh
including sdk/bash_completion/adb.bash
ght@ubuntu-230:~/android
$ lunch

You're building on Linux

Lunch menu... pick a combo:
  1. msm8953_64-userdebug
  2. msm8953_64-user

Which would you like? [aosp_arm-eng] 2

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=7.1.2
TARGET_PRODUCT=msm8953_64
TARGET_BUILD_VARIANT=user
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=generic
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv7-a-neon
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-4.4.0-31-generic-x86_64-with-Ubuntu-14.04-trusty
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=N2G47H
OUT_DIR=out
=====
ght@ubuntu-230:~/android
$ mmm CBB_Linux_Host_Upgrade_Tool_Android_Linux_V1.3/code/

=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=7.1.2
TARGET_PRODUCT=msm8953_64
TARGET_BUILD_VARIANT=user
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm64
TARGET_ARCH_VARIANT=armv8-a
TARGET_CPU_VARIANT=generic
TARGET_2ND_ARCH=arm
TARGET_2ND_ARCH_VARIANT=armv7-a-neon
TARGET_2ND_CPU_VARIANT=cortex-a53
HOST_ARCH=x86_64
HOST_2ND_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-4.4.0-31-generic-x86_64-with-Ubuntu-14.04-trusty
HOST_CROSS_OS=windows
HOST_CROSS_ARCH=x86
HOST_CROSS_2ND_ARCH=x86_64
HOST_BUILD_TYPE=release
BUILD_ID=N2G47H
OUT_DIR=out
=====
Fatal: Not a git repository (or any parent up to mount point /fwork3)
Stopping at filesystem boundary (GIT_DISCOVERY_ACROSS_FILESYSTEM not set).
make: Entering directory `/fwork3/ght/android'
Running kati to generate build-ba5cb121c5911a0d1757241257c87132.ninja...
No need to regenerate ninja file
Starting build with ninja
ninja: Entering directory `.'
[100% 14/14] Install: out/target/product/msm8953_64/system/bin/upgrade_tool
make: Leaving directory `/fwork3/ght/android'

#### make completed successfully (3 seconds) ####
```


3 升级

3.1 本地升级

1. 查看 USB 连接正常。

如果模块在正常模式，需要先发送 `at+disk=0,0,0` 解锁 diag 端口。

2. 将解压后的固件包和 `upgrade_tool` 工具拷贝到主机上面。
3. 进入 `upgrade_tool` 所在目录，执行下载命令，如果只有一个模块，通常情况不需要指定下载端口

```
./upgrade_tool -f <firmware image dir> <-r 1>
```

例如：

```
./upgrade_tool -f 19010.1000.00.02.73.15/Maincode <-r 1>
```

4. 指定下载端口下载

- UART 和 USB 端口：-p /dev/ttyUSBx

```
./upgrade_tool -f <firmware image dir> <-r 1> -p /dev/ttyUSB0
```

- PCIE 端口：-M /dev/mhi_DUN

```
./upgrade_tool -f <firmware image dir> -p /dev/mhi_DUN
```

3.2 远程升级

如果设备主机没有足够的存储空间，`upgrade_tool` 可以通过网络用远程主机升级固件。

操作步骤如下：

1. 将 `upgrade_tool` 工具放到设备主机上。
2. 将固件包和 `upgrade_tool` 放到远程主机上。
3. 在设备主机运行 `upgrade_tool` 启动 USB TCP 服务。

```
./upgrade_tool -p 9008
```

4. 查看设备主机 IP 地址，通过远程主机可以 ping 通设备主机。

5. 在远端主机执行下载命令：

```
./upgrade_tool -f <firmware image dir> -p <The client IP:9008> <-r 1>
```

3.3 查看升级结果

- 升级成功会打印

```
Upgrade module successfully
```

- 升级失败后，可以从下载日志获取升级失败原因。
- 如果客户查看 log 不方便，upgrade_tool 工具生成 update_result.txt 文件保存升级结果。

升级成功：OK

升级失败：ERROR

升级中：UPGRADING

4 升级参数说明

序号	参数	是否必须	描述
1	-f <firmware image dir>	是	升级固件包名称或者固件包目录
2	-r <0/1>	否	此参数只针对 NL668 项目。当设置-r 1，下载后开机会自动还原 NV。不设置-r 1，下载后需要手动执行 AT 命令还原 NV。
3	-s </sys/bus/usb/devices/xx>	否	当多个模块连接到主机时，该参数用于指定需要升级的模块。当只有一个模块连接主机时，无需设置该参数。
4	-p <port>	否	下载端口 (/dev/ttyUSBX) 或者 TCP USB 远程升级 <-p 9008 and -p IP:9008>
5	-z <0/1>	否	发送 0 包，默认为 0，没有特别说明无需设置参数。
6	-e <0/1>	否	升级前全擦分区，默认为 0，没有特别说明无需设置参数
7	-l <log dir>	否	设置 log 路径，保存下载失败 log

5 还原 NV

5.1 MDM9205 TX 模块还原 NV

MA510模块不能自动还原NV，下载完成后需要执行AT命令还原NV。

操作步骤如下：

1. 发送如下 at 命令。

```
at+efserrfatal
```

2. 发送如下命令，重启模块。

```
at+cfun=15
```

3. 重启完成后，查看 IMEI 和 SN 是否正常。

5.2 MDM9x07 LE 模块还原 NV

MDM9x07 模块升级固件的时候添加参数<-r 1>可以实现自动还原 NV。

如果升级固件的时候未添加<-r 1>，可以执行 AT 命令还原 NV。

操作步骤如下：

1. 发送 at 命令: at+efserrfatal
2. 发送 at 命令: at+cfun=15
3. 模块重启，并自动进行 NV 还原。
4. 等待开机完成后，查看 IMEI 和 SN 是否正常。