Fibocom
PERFECT WIRELESS EXPERIENCE

Firmware Upgrade Guide_Linux

V1.5

Disclaimer

Any action you take in the course of using this document is at your own risk, and Fibocom shall not be liable for any damages or losses under any circumstances. Due to product version upgrade or other reasons, Fibocom reserves the right to modify any information in this document at any time without prior notice and any responsibility. Unless otherwise agreed, all statements, information and suggestions in this document do not constitute any express or implied guarantee.

This document may include the third-party information covering products, services, software, data, and so on. Fibocom does not control and assumes no responsibility for the third-party content, including but not limited to the accuracy, compatibility, reliability, availability, legitimacy, appropriateness, performance, non-infringement, and status update, unless otherwise specified in this document. Fibocom does not provide any guarantee or authorization for the third-party content mentioned or referenced in this document. If you need a third-party license, obtain it in an authorized or legal way, unless otherwise specified in this document.

Copyright Notice

Unless specially authorized by Fibocom, the recipient of the documents shall keep the documents and information received confidential, and shall not use them for any purpose other than the implementation and development of this project. Without the written permission of Fibocom, no unit or individual shall extract or copy part or all of the contents of this document without authorization, or transmit them in any form. Fibocom has the right to investigate legal liabilities for any offense and tort in connection with violation of confidentiality obligations, or unauthorized use or malicious use of the said documents and information in other illegal forms.

Trademark Statement

Fibocom The trademark is registered and owned by Fibocom Wireless Inc.

Other trademarks, product names, service names and company names appearing in this document are owned by their respective owners.

Contact Information

Website: https://www.fibocom.com

Address: 10/F-14/F, Block A, Building 6, Shenzhen International Innovation Valley, Dashi First Road, Xili Community, Xili Subdistrict, Nanshan District, Shenzhen

Tel: 0755-26733555

Contents

Applicable Models

No.	Applicable Model	Description
1	L716/L718	ZTE V3E/T
2	L61x/LC61x/MC61x/LG61x	UNISOC 8910
3	MC66x/MG66x	UNISOC 8850
4	FG650/FG652/FM650	UNISOC UDX710
5	NL668/MC116/LC116	QCOM MDM9x07
6	MA510/MC109/MC100E	QCOM MDM9205
7	FG10x/FM10x	QCOM X12
8	FM150/FG150	QCOM X55
9	FM160/FG160	QCOM X62
10	FG621	UNISOC SL8563
11	LE270	EC718
12	FG132	QCOM X35

Change History

V1.5 (2023-11-23)	1.Modify the FG621 chip signal 2. Added core shifting platform and related description of the platform
V1.4 (2023-11-07)	Improved the tool adaptation model and -r1 parameter description
V1.3 (2023-10-23)	The power-off message cannot be displayed during the upgrade
V1.2 (2023-08-23)	Modifying Parameters
V1.1 (2023-05-20)	Added the description of upgrading pcie ports on a Qualcomm platform
V1.0 (2022-08-28)	Initial version.

1 Introduction

Upgrade_ tool is used for Linux and Android hosts to upgrade the firmware of Fibocom modules.

2 Compilation

2.1 Upgrade Tool Package Introduction

- doc: Chinese and English guidance documents
- misc_code: common code
- qcom_code: Qcom modules upgrade code
- zte_code: ZTE modules upgrade code
- unisoc_code: Unisco modules upgrade code
- eigencomm _code: eigencomm modules upgrade code
- main.c: The program code
- Makefile: Compiling configuration file for Linux Environment
- Android.mk: Compiling configuration file for Android Environment

The detailed directory structure is as shown in the figure 1.

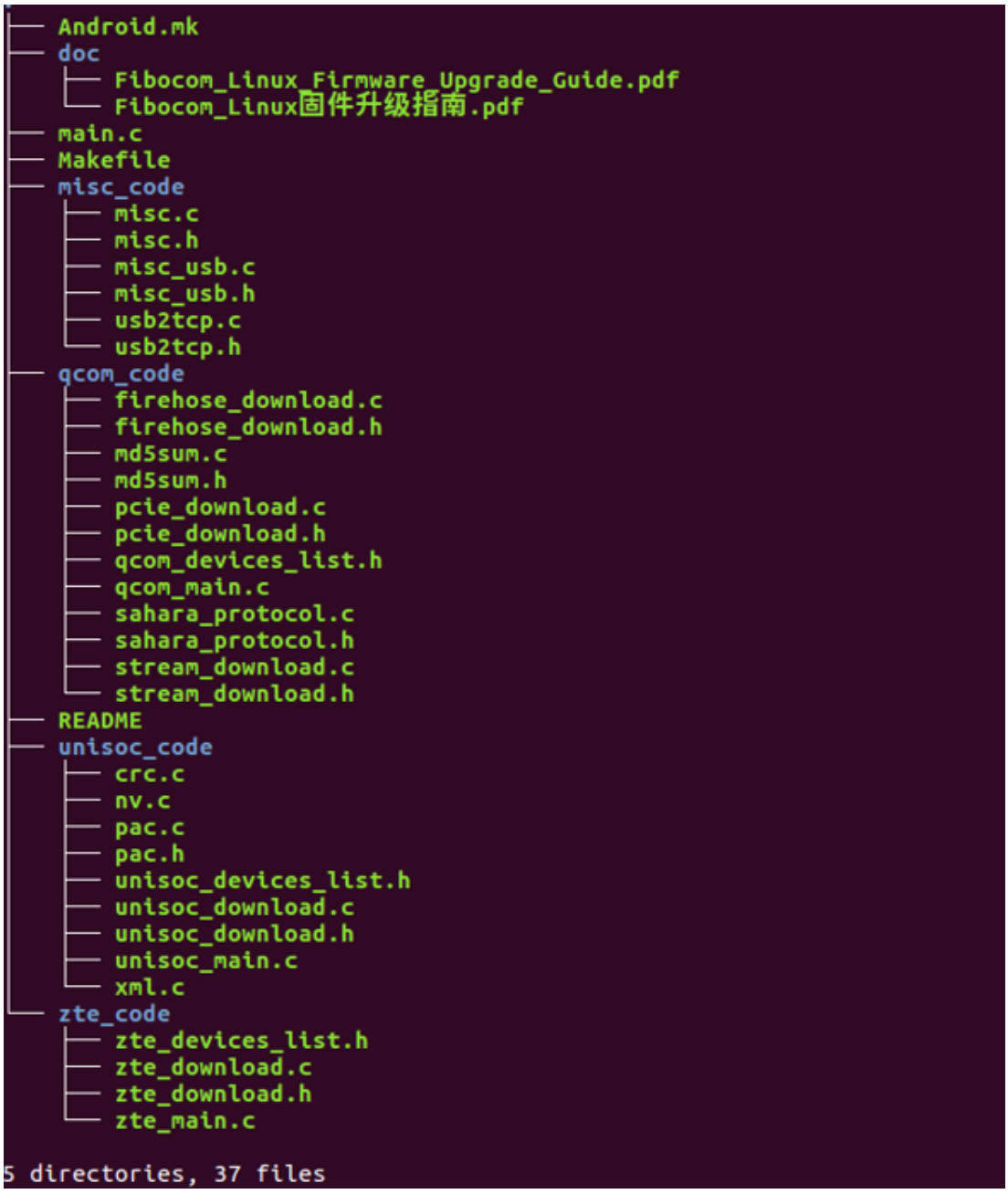

Figure 1. Directory structure

2.2 Compile Upgrade Tool

2.2.1 Linux Upgrade Tool

- Configure cross compilation tool

GCC is used by default. If arm GCC is required, set CROSS_COMPILE variable in Makefile.

As blow:

```
# If use the arm gcc, please set the arm gcc path.
# ARM_GCC = ~/ARM_Linux_GCC/bin/arm-none-linux-gnueabi-

CC = $(ARM_GCC)gcc

CFLAGS = -g -Wall

LD_LIBRARY = -lpthread

INCLUDE = -I./zte_code -I./qcom_code -I./unisoc_code -I./misc_code
SOURCES = $(wildcard ./zte_code/*.c ./qcom_code/*.c ./unisoc_code/*.c ./misc_cod
e/*.c ./*.c)

BIN_TARGET = upgrade_tool

linux: clean
        ${CC} ${CFLAGS} -s ${SOURCES} ${INCLUDE} -o ${BIN_TARGET} $(LD_LIBRARY)

.PHONY: clean

clean:
        @rm -rvf $(BIN_TARGET)
~
~
~
"Makefile" [dos] 22L, 515C                                    22,0-1        All
```

Figure 2. Makefile

- Compilation

Put the upgrade tool code on the Linux host, then in the code directory, execute make. The upgrade_ tool will be generated if the compilation is OK.

As shown in the following figure.

```
ght@ght-pc:~/Fibocom_Linux_Firmware_Upgrade_V1.1.0.0
$ ls
Android.mk  main.c    misc_code  README       zte_code
doc         Makefile  qcom_code  unisoc_code
ght@ght-pc:~/Fibocom_Linux_Firmware_Upgrade_V1.1.0.0
$ make
gcc -g -Wall -s ./zte_code/zte_download.c ./zte_code/zte_main.c ./qcom_code/saha
ra_protocol.c ./qcom_code/firehose_download.c ./qcom_code/pcie_download.c ./qcom
_code/stream_download.c ./qcom_code/qcom_main.c ./qcom_code/md5sum.c ./unisoc_co
de/xml.c ./unisoc_code/unisoc_download.c ./unisoc_code/crc.c ./unisoc_code/uniso
c_main.c ./unisoc_code/nv.c ./unisoc_code/pac.c ./misc_code/usb2tcp.c ./misc_cod
e/misc_usb.c ./misc_code/misc.c ./main.c -I./zte_code -I./qcom_code -I./unisoc_c
ode -I./misc_code -o upgrade_tool -lpthread
ght@ght-pc:~/Fibocom_Linux_Firmware_Upgrade_V1.1.0.0
$ ls
Android.mk  main.c    misc_code  README       upgrade_tool
doc         Makefile  qcom_code  unisoc_code  zte_code
ght@ght-pc:~/Fibocom_Linux_Firmware_Upgrade_V1.1.0.0
$
```

Figure 3. Compile generated result

The eigencomm platform currently only supports 32-bit compilation, which includes - m32 in the compilation option. If an error is reported during the compilation process: sys/cdefs. h: No such file or directory, it indicates that the

> Linux host needs to install 32-bit library files. In Ubuntu, the sudo apt-get install libc6-dev-i386 command to install online.

2.2.2 Android Upgrade Tool

1. Put the upgrade tool code into the Android code directory.
2. Run source build/envsetup.sh.
3. Run lunch, and then select the build option.
4. Run mmm Fibocom_Linux_Firmware_Upgrade_XXX
5. If the compilation is successful, the upgrade_tool will be generated

The path of upgrade_tool will be displayed in the compilation log.

e.g.

out/target/product/msm8953_64/system/bin/upgrade_tool

> Android system tools for Android 8 and higher are not currently compatible.

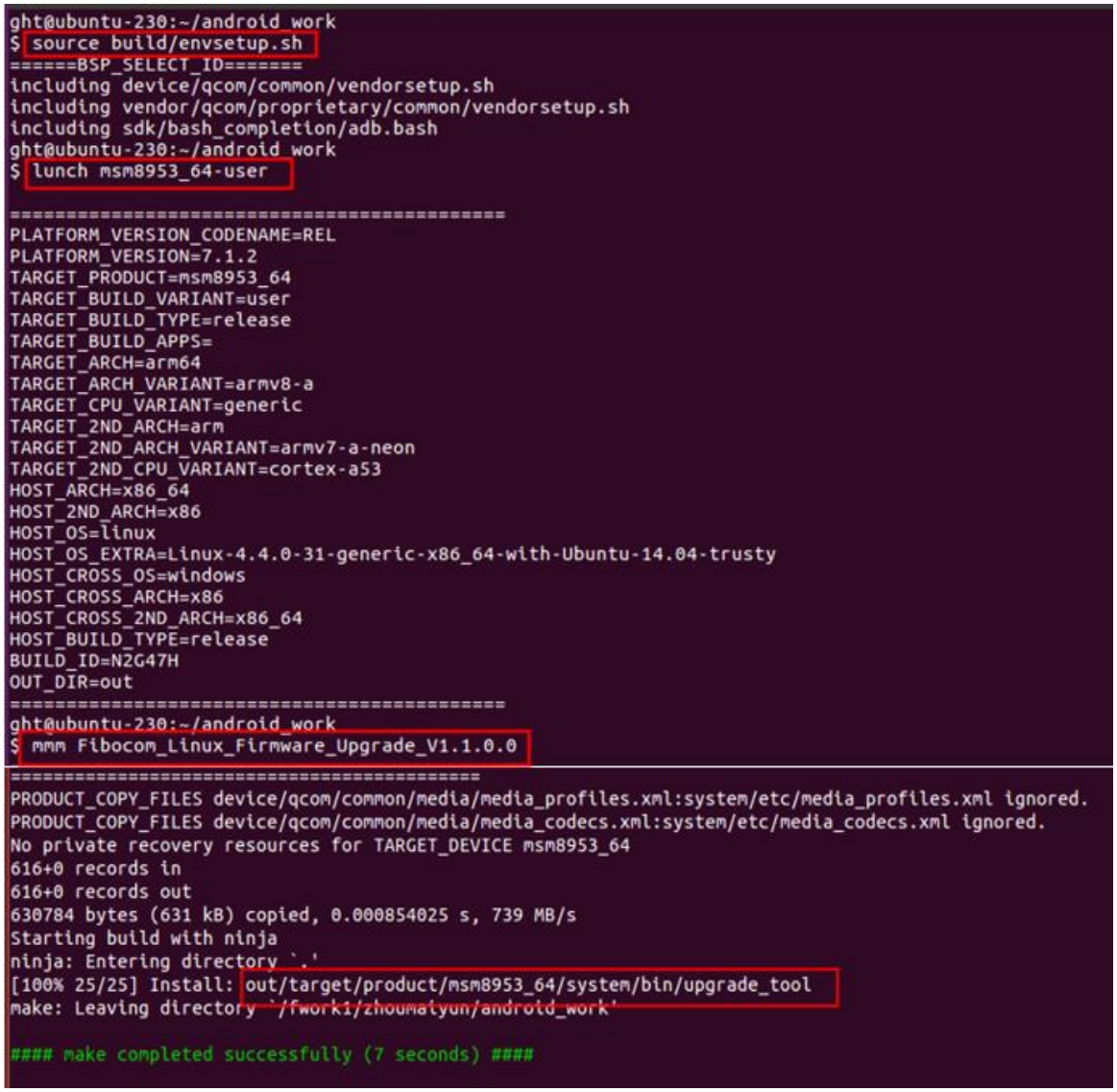

Figure 4. Android environment compilation procedure

3 Upgrade

3.1 Local Upgrade

1. Check whether the USB connection of the module is normal.

Qcom module:

If the qcom module is in normal mode, please send at+disk=0,0,0 to unlock diag port first.

During the upgrade, ensure that the module cannot be powered off.

Eigencomm module:

1. At present, it only supports pressing the boot button when powered on, and the module enters forced download mode, using USB for firmware upgrade.

2. The upgrade process relies on files in the config folder, so upgrade programs in the same level directory must exist in the config folder.

2. Copy the firmware file and upgrade_tool to the host.
3. Enter the directory where the upgrade_tool is located, and then execute the upgrade command.
4. ./upgrade_tool -f firmware image or dir

e.g.

```
qcom:    ./upgrade_tool -f 19010.1000.00.02.73.15/Maincode -r 1
unisoc:  ./upgrade_tool  -f  16000.1000.00.06.01.05.pac
zte:           ./upgrade_tool  -f  17016.1000.00.38.01.21.bin
eigencomm：./upgrade_tool  -f  12007.6000.00.02.02.07-718.binpkg
```

There is a difference between the software version directory of Qualcomm SDX35 and the general Qualcomm platform, and the following command needs to be used:

Qcom-sdx35:

1、open diag：AT+GTDIAGEN=1

```
2、./upgrade_tool   -f   19003.1000.00.01.01.07_80000.00.0000/Maincode/.cust/
```

3.2 Viewing Upgrade Result

The following log will be printed after the upgrade succeeds:

```
    Upgrade module successfully
```

- If the upgrade fails, you can get the reason for the upgrade failure from the download log.
- If you run the download tool with the - l parameter, the download tool will save the upgrade log file to the corresponding directory, with the log file name fibo_download_xxx.log.

4 Upgrade Parameter Description

No.	Parameter	Necessary	Description
1	-f firmware image or dir	Yes	Upgrade firmware image dir
2	-l <log dir>	No	set log dir and save the upgrade log.
3	-r 0/1	No	This parameter is only applicable to MDM9x07 and MDM9205 platform. When -r 1 is set, NV will be restored automatically after upgrade. If -r 1 is not set, you need to manually execute at command to restore NV after upgrade
4	-d <port>	No	Download port (/dev/ttyUSBX)
5	-z <0/1>	No	Send Zero-length package. It is 0 by default. No special instructions, no need to set parameters.
6	-e	No	Erase ALL partitions before upgrading. Defaults is 0. No special instructions, no need to set parameters.

The PCIE port on the Qualcomm platform needs to be upgraded using -d /dev/mhi. Otherwise, the upgrade cannot be performed.

The eigencomm platform currently only supports the - f parameter

5 NV Restoration Commands

5.1 NV Restoration of MDM9X07 and MDM9205 Modules

- Execute upgrade_tool with the -r 1 parameter, the NV can be restored automatically when the module is powered on for the first time.
- e.g.

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

- If the parameter -r 1 is not set when upgrading firmware, you can execute the at command to restore NV.

The operation steps:

5. Send at command: at+efserrfatal
6. If the module does not restart automatically, please send reboot command: at+cfun=15
7. The module will restart and auto restore NV.
8. Wait until the module is started, and check whether IMEI and SN is OK.

MDM9205 TX modules cannot upgrade EFS. If static NV in the EFS need to update, you need to use Windows Tool Fibocom_ MDM_ Multiupdater to upgrade.

9.

5.2 NV Restoration of Other Modules

There is no need to add the -r parameter when downloading. The module will automatically restore NV after downloading.