

Fibocom_Linux_Firmware_Upgrade_Guide

V1.0

Copyright

Copyright ©2022 Fibocom Wireless Inc. All rights reserved.

Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or transmit the document in any form.

Notice

The document is subject to update from time to time owing to the product version upgrade or other reasons. Unless otherwise specified, the document only serves as the user guide. All the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Trademark

 The trademark is registered and owned by Fibocom Wireless Inc.

Contact

Website: <https://www.fibocom.com/en/>

Address: Floor 10, Building A, Shenzhen International Innovation Valley, First Stone Road, Xili Community, Xili Street, Nanshan District, Shenzhen

Tel: +86 755-26733555

Contents

Change History	3
1 Forward.....	4
1.1 Applicable Models.....	4
2 Compilation.....	5
2.1 Upgrade tool package Introduction	5
2.2 Compile Upgrade Tool.....	7
2.2.1 Linux Upgrade Tool.....	7
2.2.2 Android Upgrade Tool.....	8
3 Upgrade.....	10
3.1 Local Upgrade	10
3.2 Remote Upgrade.....	10
3.3 Viewing Upgrade Result.....	11
4 Upgrade Parameter Description	12
5 NV Restore Instructions	13
5.1 MDM9X07 and MDM9205 Modules NV Restore.....	13
5.2 Other Modules NV Restore	13

Change History

V1.0 (2022-08-28) Initial version

1 Forward

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

1.1 Applicable Models

Num	Modules	Description
1	L716/L718	ZTE V3E/T
2	L610	UNISOC 8910
3	MC661-CN-19	UNISOC 8850
4	FG650,FG652,FG621	UNISOC UDX710
5	NL668/ MC116/LC116	QCOM MDM9x07
6	MA510/ MC109/MC100E	QCOM MDM9205
7	FG10x/FM10x	QCOM SDX12

Table 1

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
- 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.

```
├── Android.mk
├── doc
│   ├── Fibocom_Linux_Firmware_Upgrade_Guide.pdf
│   └── Fibocom_Linux固件升级指南.pdf
├── main.c
├── Makefile
├── misc_code
│   ├── misc.c
│   ├── misc.h
│   ├── misc_usb.c
│   ├── misc_usb.h
│   ├── usb2tcp.c
│   └── usb2tcp.h
├── qcom_code
│   ├── firehose_download.c
│   ├── firehose_download.h
│   ├── md5sum.c
│   ├── md5sum.h
│   ├── pcie_download.c
│   ├── pcie_download.h
│   ├── qcom_devices_list.h
│   ├── qcom_main.c
│   ├── sahara_protocol.c
│   ├── sahara_protocol.h
│   ├── stream_download.c
│   └── stream_download.h
├── README
├── unisoc_code
│   ├── crc.c
│   ├── nv.c
│   ├── pac.c
│   ├── pac.h
│   ├── unisoc_devices_list.h
│   ├── unisoc_download.c
│   ├── unisoc_download.h
│   ├── unisoc_main.c
│   └── xml.c
├── zte_code
│   ├── zte_devices_list.h
│   ├── zte_download.c
│   ├── zte_download.h
│   └── zte_main.c
└── 5 directories, 37 files
```

Figure 1

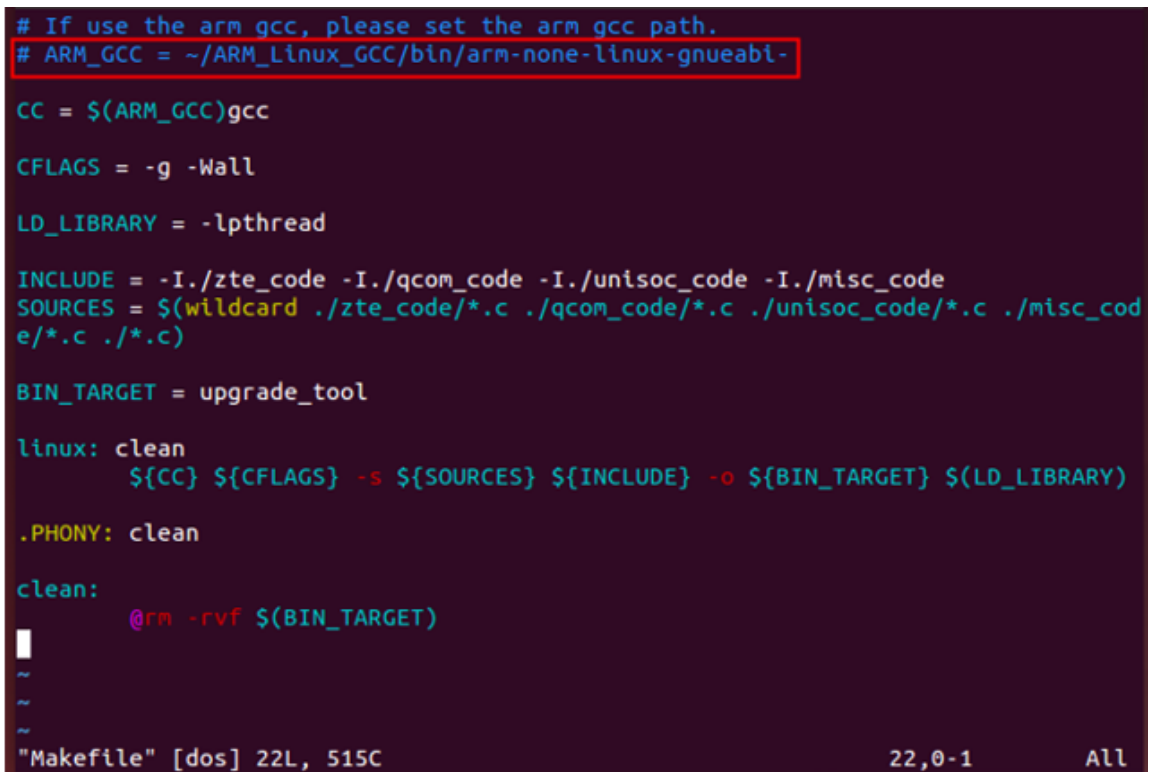
2.2 Compile Upgrade Tool

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_code/*.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}
```

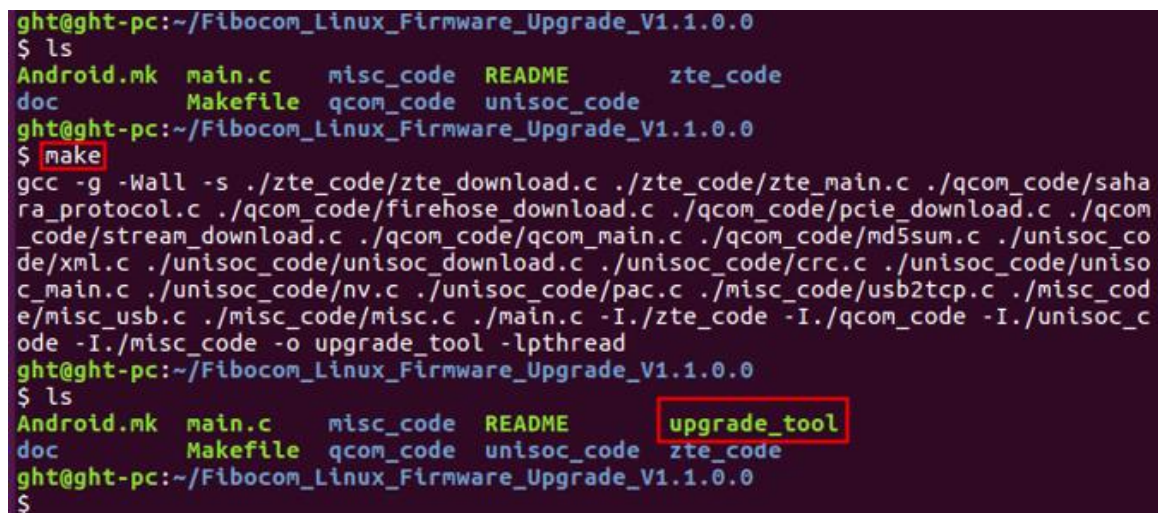
"Makefile" [dos] 22L, 515C 22,0-1 All

Figure 2

- Compile

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 figure 3.



```
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/sahara_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_code/xml.c ./unisoc_code/unisoc_download.c ./unisoc_code/crc.c ./unisoc_code/unisoc_main.c ./unisoc_code/nv.c ./unisoc_code/pac.c ./misc_code/usb2tcp.c ./misc_code/misc_usb.c ./misc_code/misc.c ./main.c -I./zte_code -I./qcom_code -I./unisoc_code -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

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_MultiPlatform_Upgrade_Tool_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

```
ght@ubuntu-230:~/android_work
$ 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_work
$ lunch msm8953_64-user

=====
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_work
$ mmm Fibocom_Linux_Firmware_Upgrade_V1.1.0.0

=====
PRODUCT_COPY_FILES device/qcom/common/media/media_profiles.xml:system/etc/media_profiles.xml ignored.
PRODUCT_COPY_FILES device/qcom/common/media/media_codecs.xml:system/etc/media_codecs.xml ignored.
No private recovery resources for TARGET_DEVICE msm8953_64
616+0 records in
616+0 records out
630784 bytes (631 kB) copied, 0.000854025 s, 739 MB/s
Starting build with ninja
ninja: Entering directory `.'
[100% 25/25] Install: out/target/product/msm8953_64/system/bin/upgrade_tool
make: Leaving directory `/fwork1/zhoumatiyun/android_work'

#### make completed successfully (7 seconds) ####
```

Figure 4

3 Upgrade

3.1 Local Upgrade

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

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

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.

`./upgrade_tool -f firmware image or dir`

e.g.

```
qcom: ./upgrade_tool -f 19010.1000.00.02.73.15/Maincode
unisoc: ./upgrade_tool -f 16000.1000.00.06.01.05.pac
zte:    ./upgrade_tool -f 17016.1000.00.38.01.21.bin
```

3.2 Remote Upgrade

If the device host does not have enough storage space, upgrade_Tool can upgrade firmware with remote host through network.

Operation steps:

1. Put the upgrade_tool on the device host

2. Put the firmware package and upgrade_tool on the remote host

3. Run upgrade_tool on the device host to start the USB TCP service

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

4. Check the IP address of the device host. You can ping the device host through the remote host

5. Execute the upgrade tool on the remote host

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

3.3 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 upgrade result will be save to update_result.txt that in the same directory of the upgrade_tool.
- Upgrade result in the update_result.txt

```
Upgrading: UPGRADING
```

```
Upgrade failed : ERROR
```

```
Upgrade succeeded: OK
```

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 failed log.
3	-r 0/1	No	This parameter is only applicable to nl668 projects. 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 /sys/bus/usb/devices/xx	No	When multiple modules are connected to the host, this parameter is used to specify the module that need to be upgraded. When only one module is connected to the host, this parameter does not need to be set.
5	-d <port>	No	Net Remote Upgrade, upgrade_tool with -d 9008 in the Modules host. upgrade_tool with -d IP:9008 in the remote host.
5	-z <0/1>	No	Send Zero-length package.Defaults is 0. 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.

5 NV Restore Instructions

5.1 MDM9X07 and MDM9205 Modules NV Restore

- 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:

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

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

5.2 Other Modules NV Restore

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