MT793X IoT SDK for Flash Burning Tool

Version: 2.84
Release date: 2022-10-5

Version History

Version	Date	Author	Description
2.77	2021-08-05	CY Chan	Create an external version
2.78	2021-12-28	CY Chan	Add customer eFuse read/write form. Add new flash ID. Modify new GUI and add supported Linux GUI
2.8	2022-1-12	CY Chan	Add customer eFuse read/write txt from. Modify pinmux pull up for new flash ID.
2.81	2022-2-9	CY Chan	Fixed get meid cmd. Change customer efuse table as mt7686 style.
2.82	2022-7-4	CY Chan	Fixed console mode path assign can't with "-". Add DA xip mode of Arduino. Console mode :add abs_path call fbtool Add buad rate 115200 for debug. Add show flash ID and size GUI mode add TLV parser Add query DA cmd for auto detect DA/BROM mode Fixed GUI readback struck error by textbox with yview. No support Mircro flash.
2.83	2022-8-31	CY Chan	Support GD25Q128E flash. GUI:add some teraterm functions and send CLI macro. Update libefuse.a for EDCSA -p256/384/521 Cmd_tab.txt:support customer command and it with any input arguments. Improve uart_start_cmd on data handshacking Support readback command with read small size for MP. Allow cmds_tab. txt within empty line. Implement flash auto unlock for winbond flash Support most common buad rate
2.84	2022-10-5	CY Chan	Fixed file naming with "." before endswitch .bin/.sgn Add log message when winbond flash be locked. Fixed end of cmd run console with error "armor" log Add section1.7: option of non-20.04 Ubuntu versions Unified executable file name from fbtool_v2p8x.exe to fbtool.exe and added -v and --version to check tool version

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

The Flash Burning Tool is an application that mainly updates the flash. The tool works on MediaTek MT793X IoT platforms. The Flash Burning Tool supports both UART and USB interfaces.

The Flash Burning Tool communicates with BROM, and BROM loads DA (download agent) code into SRAM, and then jumps into DA code for execution. DA is mainly responsible for downloading all image bin files to each partition zone. In addition, erase, read back, eFuse read/write functions are added to this application. Table 1 lists the items supported by the Flash Burning Tool .

The Flash Burning Tool can be used on Microsoft Windows 7 (64-bit) and Windows 10 (64-bit) PC that support USB interface communication.

MediaTek MT793X HDK has a USB connector for the Flash Burning Tool to operate through the USB interface. The HDK also provides a USB to UART connector for the Flash Burning Tool to operate through the UART.

Before using the Flash Burning Tool, install the corresponding driver. Refer to Sections 2.1 and 2.2.

Acronyms used throughout the document are defined in the following table.

Acronym	Definition
BROM	Boot ROM
DA	Download Agent
GUI	Graphical User Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
IoT	Internet of Things
DL	Download
FBTool	FlashBurningTool

1.2 Flash Burning Tool Feature List

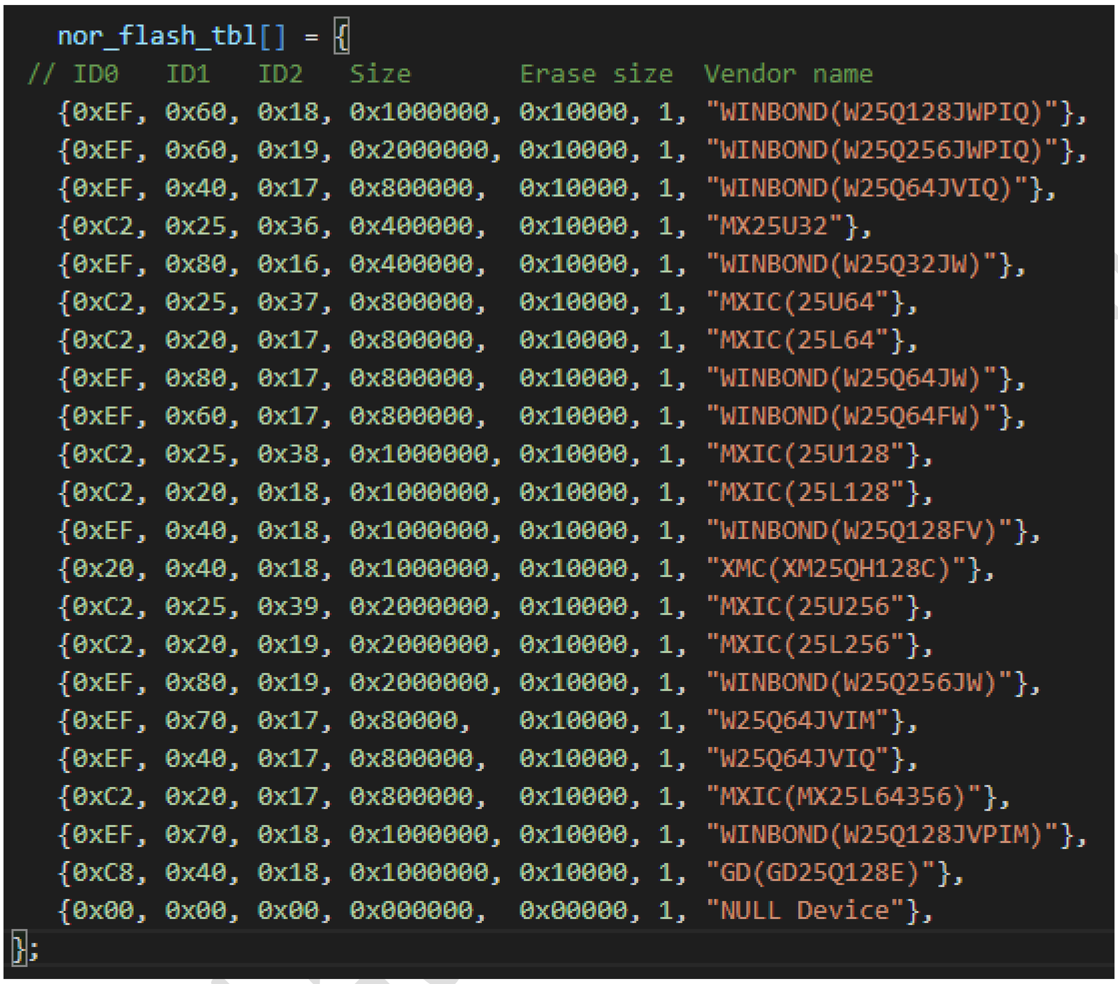
Table 1: Flash Burning Tool feature list

	Supported items		
	Item	**Feature**	**Output file**
	OS:		
V	Windows 64-bit (GUI mode)	FBTool_gui.exe, security by pyarmor tool	debug_log.txt
V	Windows 64-bit (Console mode)	fbtool.exe, security by pyarmor tool Whether to use an input argument is optional: fbtool.exe -e myefuse_tbl.txt -f my_scatterfile.ini -o order_command -s control_data -p uart_port	debug_log.txt DL_fa_log_append.txt Process.ini
V	Linux 64-bit (GUI mode)	Ubuntu 20.04 or py37 FBTool_gui, run_FBTool_gui.py , security by pyarmor tool	debug_log.txt
V	Linux 64-bit (Console mode)	Ubuntu 20.04 or py37 fbtool, run_fbtool_console.py ,security by pyarmor tool Whether to use an input argument is optional: ./fbtool -e myefuse_tbl.txt -f my_scatterfile.ini -o order_command -s control_data -p uart_port	debug_log.txt DL_fa_log_append.txt Process.ini
	Connect Path:		
V	UART	CM33 UART port, baud rate: 921600. (reference test: 1.8M bin file, burning time ~= 42sec)	
V	USB	USB2.0 (reference test: 1.8M bin file, burning time ~= 22sec)	
	Command:		
V	Format	Quick erase of 64/32/4K alignment	
V	Download	Check sum, 32K buffering Support erasing before download or only download	
V	Read Back	Check sum + File compare UART: 32K buffering, USB: 32K buffering Support read full size or small size:4k form header	“file name”+_readback.bin
V	Read eFuse	1) Input by control data to read eFuse, CR, and flash data. Support reads of GRP1/2/3, bin data sizes of which are 0x200, 0x200, 0x400 Support logical and physical reads of GRP2/3. 2) Support reads of GRP1 in customer mode.	working folder: 1) eFuse.bin 2.1) Customer_ef_tbl.txt 2.2) user define
V	Write eFuse	1) Input by control data to write eFuse, CR, and flash. eFuse only supports GRP1. 2) Support writes of GRP1 in customer mode.	Same as above.
	HW PCB:		
V	MT793X_DRQFN_EVB	UART: MT8127 debug board or FTDI debug board	
V	MT793X_BGA_EVB	UART: MT8127 debug board or FTDI debug board / USB download	
V	MT793X_QFN_RFB	UART: FTDI debug board	
V	MT793X_BGA_RFB	UART: FTDI debug board /USB download	

1.3 Flash Burning Tool supported Nor Flash list

Table2 show supported lists of nor flash.

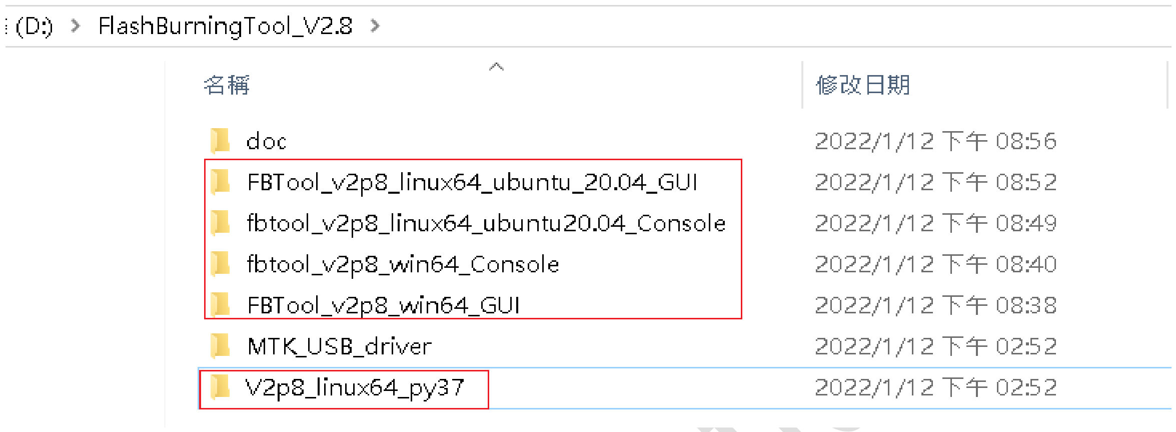
Table 2: Support nor flash list



```
  nor_flash_tbl[] = {
// ID0   ID1   ID2   Size         Erase size  Vendor name
  {0xEF, 0x60, 0x18, 0x1000000, 0x10000, 1, "WINBOND(W25Q128JWPIQ)"},
  {0xEF, 0x60, 0x19, 0x2000000, 0x10000, 1, "WINBOND(W25Q256JWPIQ)"},
  {0xEF, 0x40, 0x17, 0x800000,  0x10000, 1, "WINBOND(W25Q64JVIQ)"},
  {0xC2, 0x25, 0x36, 0x400000,  0x10000, 1, "MX25U32"},
  {0xEF, 0x80, 0x16, 0x400000,  0x10000, 1, "WINBOND(W25Q32JW)"},
  {0xC2, 0x25, 0x37, 0x800000,  0x10000, 1, "MXIC(25U64"},
  {0xC2, 0x20, 0x17, 0x800000,  0x10000, 1, "MXIC(25L64"},
  {0xEF, 0x80, 0x17, 0x800000,  0x10000, 1, "WINBOND(W25Q64JW)"},
  {0xEF, 0x60, 0x17, 0x800000,  0x10000, 1, "WINBOND(W25Q64FW)"},
  {0xC2, 0x25, 0x38, 0x1000000, 0x10000, 1, "MXIC(25U128"},
  {0xC2, 0x20, 0x18, 0x1000000, 0x10000, 1, "MXIC(25L128"},
  {0xEF, 0x40, 0x18, 0x1000000, 0x10000, 1, "WINBOND(W25Q128FV)"},
  {0x20, 0x40, 0x18, 0x1000000, 0x10000, 1, "XMC(XM25QH128C)"},
  {0xC2, 0x25, 0x39, 0x2000000, 0x10000, 1, "MXIC(25U256"},
  {0xC2, 0x20, 0x19, 0x2000000, 0x10000, 1, "MXIC(25L256"},
  {0xEF, 0x80, 0x19, 0x2000000, 0x10000, 1, "WINBOND(W25Q256JW)"},
  {0xEF, 0x70, 0x17, 0x80000,   0x10000, 1, "W25Q64JVIM"},
  {0xEF, 0x40, 0x17, 0x800000,  0x10000, 1, "W25Q64JVIQ"},
  {0xC2, 0x20, 0x17, 0x800000,  0x10000, 1, "MXIC(MX25L64356)"},
  {0xEF, 0x70, 0x18, 0x1000000, 0x10000, 1, "WINBOND(W25Q128JVPIM)"},
  {0xC8, 0x40, 0x18, 0x1000000, 0x10000, 1, "GD(GD25Q128E)"},
  {0x00, 0x00, 0x00, 0x000000,  0x00000, 1, "NULL Device"},
};
```

1.4 Environment and Supported Versions

The IoT FlashBuringTool with GUI interface and command line can operate on Windows and Linux Ubuntu 20.04 both; the tool can be used on Microsoft Windows Win 7/10 (64-bis) and Linux (64-bit).
MediaTek MT793X IoT HDK has a USB connector for the Flash Burning Tool to operate through the USB interface. The HDK also provides a USB to UART connector for the Flash Burning Tool to operate through the UART. Before using the Flash Burning Tool, install the corresponding driver. Refer to Sections 2.1 and 2.2.
Unzip the FlashBurningTool package. The folder name shows the supported Windows or Linus version.



(D:) > FlashBurningTool_V2.8 >

名稱	修改日期
doc	2022/1/12 下午 08:56
FBTool_v2p8_linux64_ubuntu_20.04_GUI	2022/1/12 下午 08:52
fbtool_v2p8_linux64_ubuntu20.04_Console	2022/1/12 下午 08:49
fbtool_v2p8_win64_Console	2022/1/12 下午 08:40
FBTool_v2p8_win64_GUI	2022/1/12 下午 08:38
MTK_USB_driver	2022/1/12 下午 02:52
V2p8_linux64_py37	2022/1/12 下午 02:52

Figure 1: OS versions supported by Flash Burning Tool

1.5 Installing the Flash Burning Tool on Windows

The tool supports both GUI and console modes as shown in Figure 1. To install the Flash Burning Tool, simply unzip the tool package to a folder on your Windows computer. No further steps are required.

1.6 Installing the Flash Burning Tool on Linux

The tool supports both GUI and console modes on Ubuntu 20.04 as Figure 1 shows. You can also choose to install Python 3.7 to use the tool package of V2p8_linux64_py37 for different Ubuntu versions.
Reference commands for installing Python 3.7:

sudo apt install python3.7
sudo apt-get install python3.7-dev
sudo apt-get install python3-pip

install modules:

sudo python3.7 -m pip install xxx --trusted-host pypi.python.org --trusted-host files.pythonhosted.org --trusted-host pypi.org
(xxx: modules name, which can be imported to the header on python script. e.g. pyserial, configparser, optparse-pretty …)
install tkinter module on python3.7 : sudo apt-get install python3.7-tk

Check version: python3 –version
Add list table to version: sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 x (x=digital number).
Choose version: sudo update-alternatives --config python3

1.7 Option of install tools on non-20.04 Ubuntu version

Another method for use tools on non-20.04 Ubuntu version that can base on Ubuntu 20.04 released package and overwrite releated libraried(*.so files) , we giving an example of use Ubuntu 18.04 OS and showing how to base on released package of Ubuntu 20.04_console to the different Ubuntu distribution.

Step1:
Install python3.7 and related modules on Ubuntu 18.04, Please reference section1.6

Step2:
Copy the installed Python3.7 libraries to the tool's directory with the following commands:

```
cp /usr/lib/x86_64-linux-gnu/libpython3.7m.so.1.0 fbtool_v2p8_linux64_ubuntu20.04_Console
cp /usr/lib/x86_64-linux-gnu/libpng16.so.16 fbtool_v2p8_linux64_ubuntu20.04_Console
cp /usr/lib/x86_64-linux-gnu/libtcl8.6.so fbtool_v2p8_linux64_ubuntu20.04_Console
cp /usr/lib/x86_64-linux-gnu/libtk8.6.so fbtool_v2p8_linux64_ubuntu20.04_Console
cp /usr/lib/x86_64-linux-gnu/libtk8.6.so.0 fbtool_v2p8_linux64_ubuntu20.04_Console
```

Step3:
Renaming folder :
"fbtool_v2p8_linux64_ubuntu20.04_Console" to "fbtool_v2p8_linux64_ubuntu18.04_Console"

2 MT793X HDK Application Note

2.1 UART Download Path: FDTI Debug Board Driver Installation

2.1.1 Install Windows UART Driver

You can install the FTDI driver from this reference link:
https://www.ftdichip.com/Drivers/VCP.htm
The FlashBurningTool uses the CM33_UART COM port to communicate.

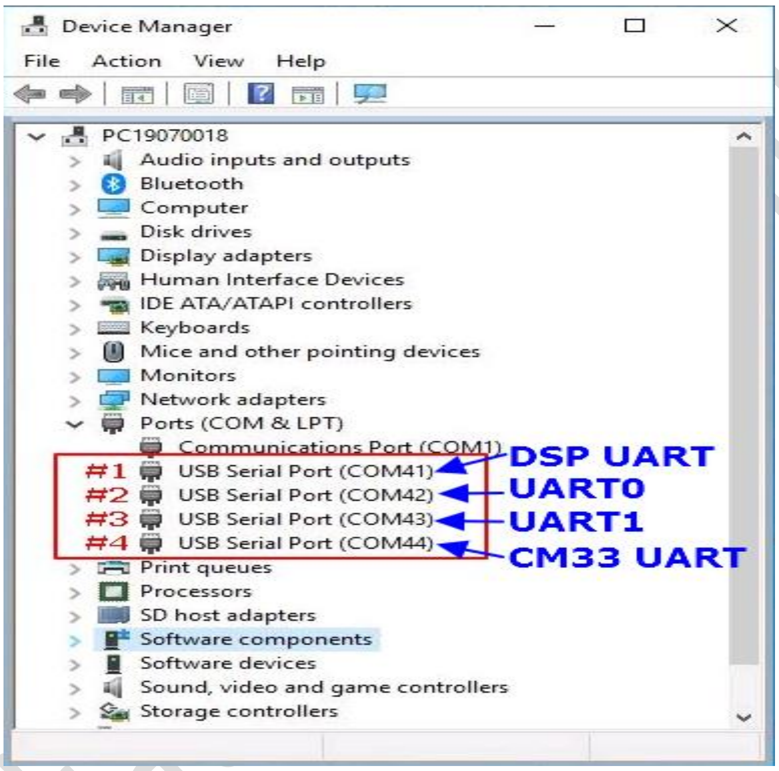
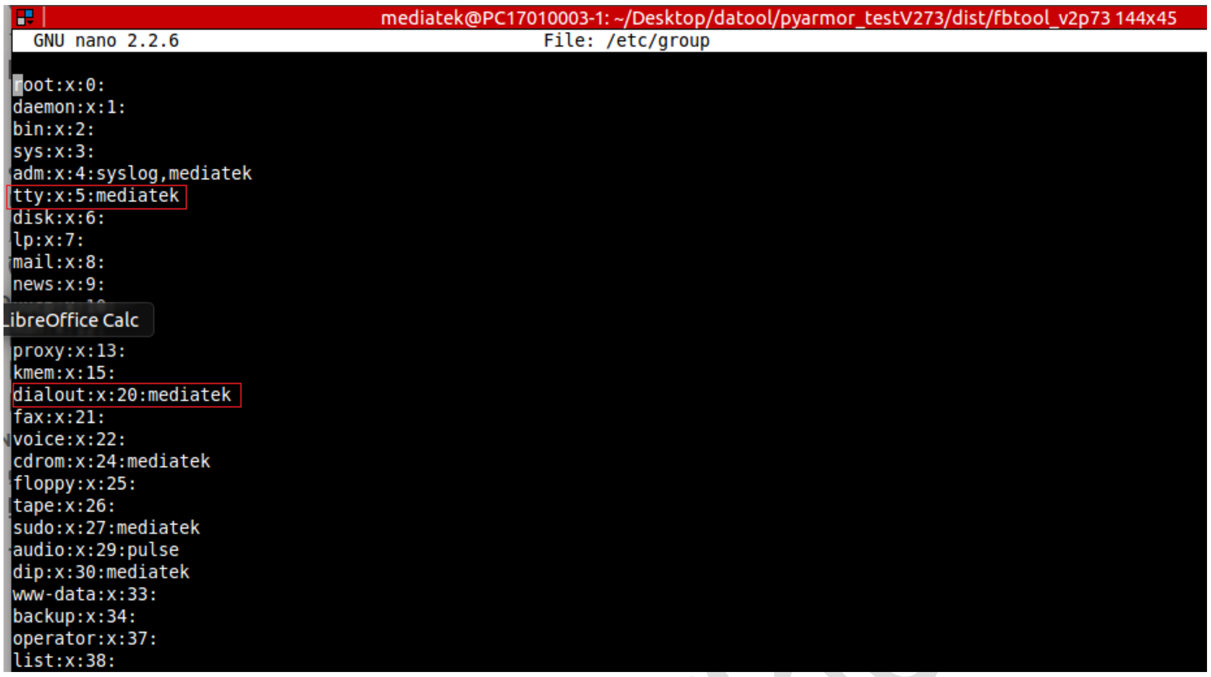


Figure 2: Select CM33 UART port on FTDI debug board

2.1.2 Install Linux UART Driver

(1) Set up Linux environment:
Copy mtk-usb.rules to /etc/udev/rules.d/; mtk-usb.rules is located in the folder under the Linux tool folder of this tool. Assign “mediatek” after tty:x:5 and dialout:x:20
Linux command:

```
$cp -f mtk-usb.rules /etc/udev/rules.d/mtk-usb.rules
$sudo nano /etc/group
```



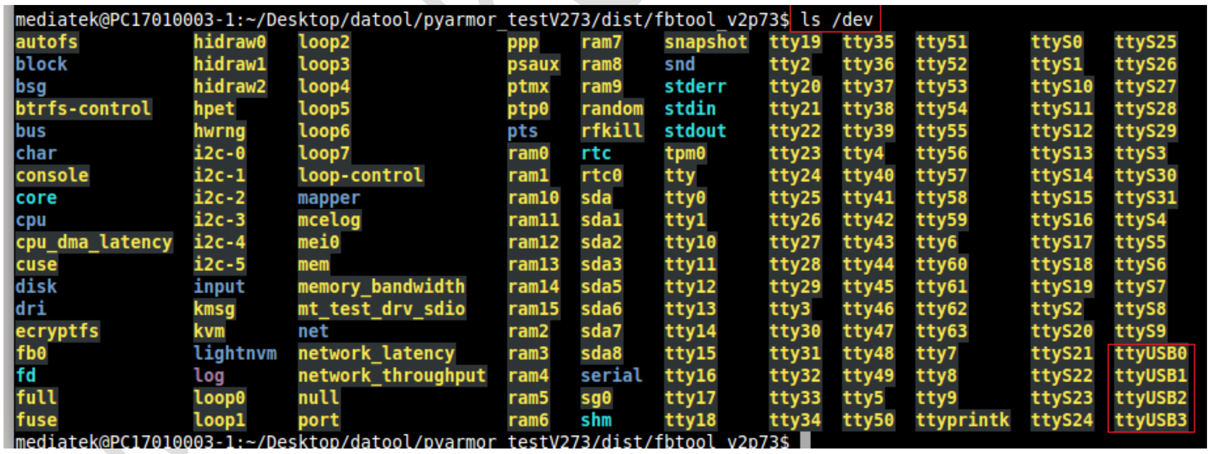
```
mediatek@PC17010003-1: ~/Desktop/datool/pyarmor_testV273/dist/fbtool_v2p73 144x45
  GNU nano 2.2.6                            File: /etc/group

root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,mediatek
tty:x:5:mediatek
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
ibreOffice Calc
proxy:x:13:
kmem:x:15:
dialout:x:20:mediatek
fax:x:21:
voice:x:22:
cdrom:x:24:mediatek
floppy:x:25:
tape:x:26:
sudo:x:27:mediatek
audio:x:29:pulse
dip:x:30:mediatek
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
```

Figure 3: Set up Linux environment

Reboot the computer to enable the **Udev** rules

(2) Normally you do not need to install FTDI or PL2303 driver for Linux. You can check by plugging in the USB to UART cable, and then enter “ls /dev”; the information similar to the following appears, and you can find ttyUSB0 is added to PL2303 debug board. Or, you can use FTDI, and ttyUSB0 to ttyUSB03 are displayed.



```
mediatek@PC17010003-1:~/Desktop/datool/pyarmor_testV273/dist/fbtool_v2p73$ ls /dev
autofs           hidraw0  loop2              ppp    ram7    snapshot  tty19  tty35  tty51      ttyS0   ttyS25
block            hidraw1  loop3              psaux  ram8    snd       tty2   tty36  tty52      ttyS1   ttyS26
bsg              hidraw2  loop4              ptmx   ram9    stderr    tty20  tty37  tty53      ttyS10  ttyS27
btrfs-control    hpet     loop5              ptp0   random  stdin     tty21  tty38  tty54      ttyS11  ttyS28
bus              hwrng    loop6              pts    rfkill  stdout    tty22  tty39  tty55      ttyS12  ttyS29
char             i2c-0    loop7              ram0   rtc     tpm0      tty23  tty4   tty56      ttyS13  ttyS3
console          i2c-1    loop-control       ram1   rtc0    tty       tty24  tty40  tty57      ttyS14  ttyS30
core             i2c-2    mapper             ram10  sda     tty0      tty25  tty41  tty58      ttyS15  ttyS31
cpu              i2c-3    mcelog             ram11  sda1    tty1      tty26  tty42  tty59      ttyS16  ttyS4
cpu_dma_latency  i2c-4    mei0               ram12  sda2    tty10     tty27  tty43  tty6       ttyS17  ttyS5
cuse             i2c-5    mem                ram13  sda3    tty11     tty28  tty44  tty60      ttyS18  ttyS6
disk             input    memory_bandwidth   ram14  sda5    tty12     tty29  tty45  tty61      ttyS19  ttyS7
dri              kmsg     mt_test_drv_sdio   ram15  sda6    tty13     tty3   tty46  tty62      ttyS2   ttyS8
ecryptfs         kvm      net                ram2   sda7    tty14     tty30  tty47  tty63      ttyS20  ttyS9
fb0              lightnvm network_latency    ram3   sda8    tty15     tty31  tty48  tty7       ttyS21  ttyUSB0
fd               log      network_throughput ram4   serial  tty16     tty32  tty49  tty8       ttyS22  ttyUSB1
full             loop0    null               ram5   sg0     tty17     tty33  tty5   tty9       ttyS23  ttyUSB2
fuse             loop1    port               ram6   shm     tty18     tty34  tty50  ttyprintk  ttyS24  ttyUSB3
mediatek@PC17010003-1:~/Desktop/datool/pyarmor_testV273/dist/fbtool_v2p73$
```

Figure 4: UART port of debug board. (Linux)

(3) If you cannot find ttyUSB*, it means the driver installation is not successful. You can find the path of the file: “pl2303.ko” on PC and copy the file to /lib/modules/$(uname -r)/kernel/drivers/

Linux command:

$find / -type f –name pl2303.ko

$cp path/of/pl2303.ko /lib/modules/$(uname -r)/kernel/drivers/

$sudo depmod

Then, reboot the PC.

2.2 Install USB Download Path Driver

2.2.1 Write eFuse to Enable USB DL Function

For USB DL function, if the chip is MT793X E1, you must be careful with two things:
(1) Rework the BGA_RFB PCB for USB DL mode. You can refer to Section 2.3.2.
(2) Write the eFuse using FlashBurningTool UART DL mode:
You need to write the eFuse to enable USB DL. You can use FlashBurningTool version 2.7 (or newer versions).
(a) Copy the attached bin file doc\GRP1_usb_en.bin to the working folder.
(b) Assign file_name GRP1_usb_en.bin in the scatter.ini in "EFUSE" partition.
(c) Assign control_data = 0x01ff0000, and run the "Write Efuse" command.

2.2.2 Install Windows USB Driver

When you unzip this tool, you can see the folders mtk_usb_driver _win7 and mtk_usb_driver_win10. Choose your PC's Windows version and install the tool. When the installation process finishes, you can open Device Manager and expand "Ports(COM$LPT)". A virtual COM port: "Mediatek USB Port(COMx)" should appear on your PC when your PCB is in USB DL mode.

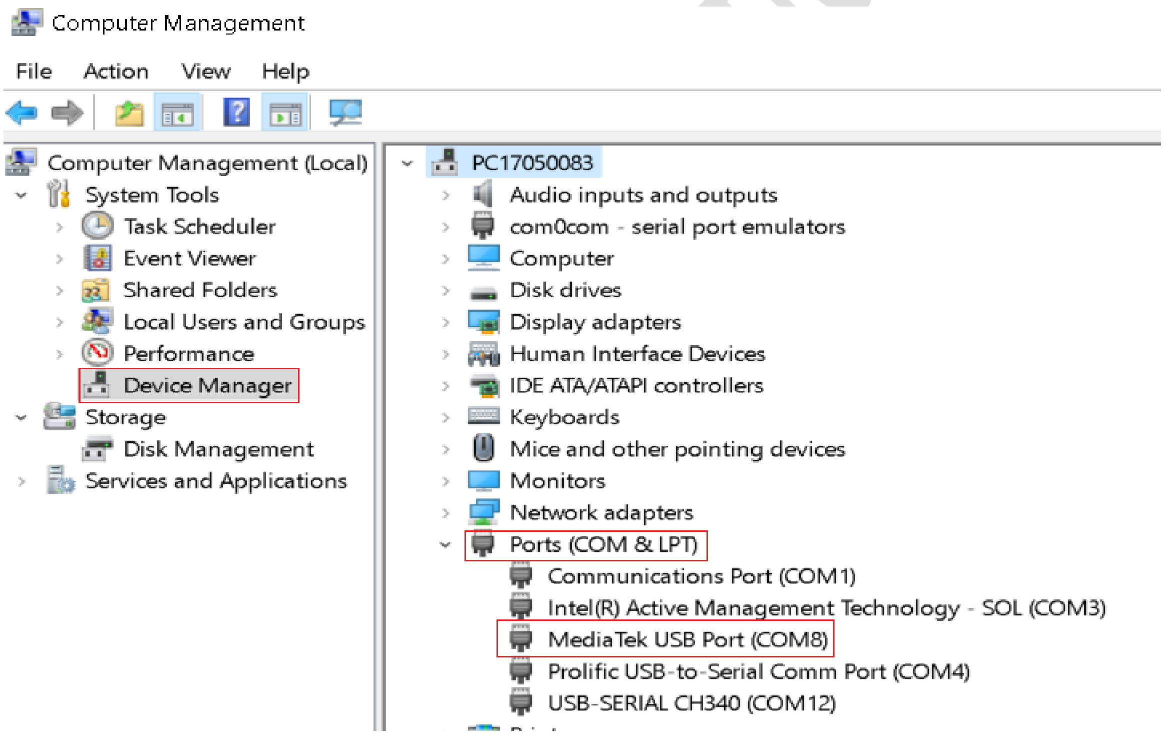
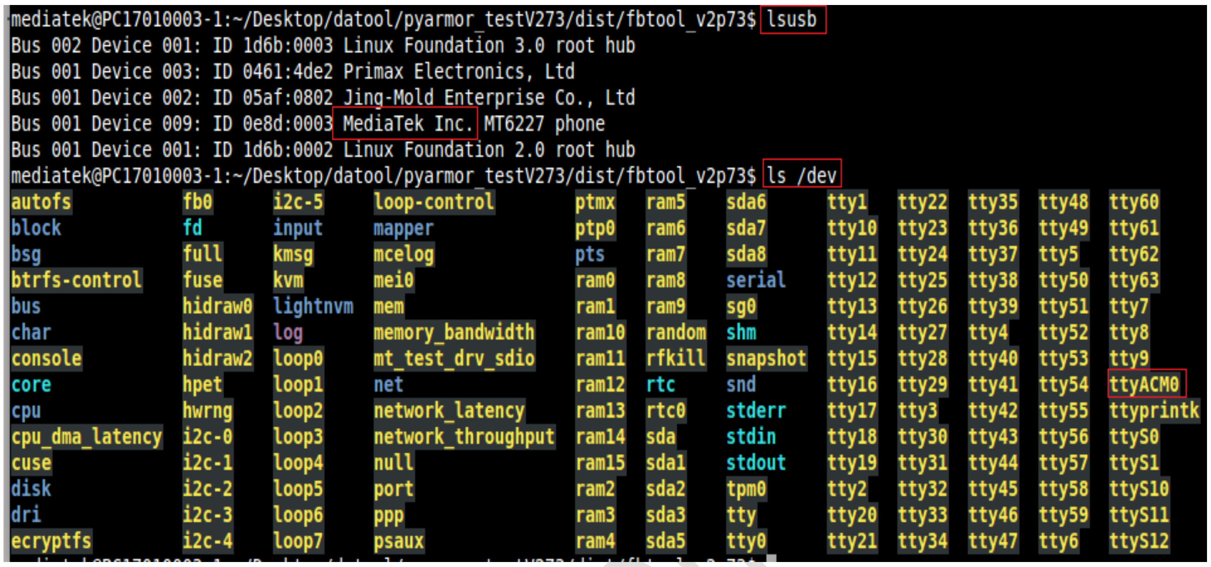


Figure 5: Install Windows USB driver via virtual COM port

2.2.3 Install Linux USB Driver

Normally you do not need to install a USB driver for Linux. To check whether you need to install a driver, plug in the USB cable of BGA-EVB after the platform powers on, or plug in the USB cable and then plug in the USB power cable of BGA-RFB. Then, ener the “lsusb” or “ls /dev”command, and “Mediatek Inc.“ or “dev/ttyACM0” should appear.



```
mediatek@PC17010003-1:~/Desktop/datool/pyarmor_testV273/dist/fbtool_v2p73$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0461:4de2 Primax Electronics, Ltd
Bus 001 Device 002: ID 05af:0802 Jing-Mold Enterprise Co., Ltd
Bus 001 Device 009: ID 0e8d:0003 MediaTek Inc. MT6227 phone
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
mediatek@PC17010003-1:~/Desktop/datool/pyarmor_testV273/dist/fbtool_v2p73$ ls /dev
autofs           fb0      i2c-5     loop-control        ptmx   ram5    sda6      tty1   tty22  tty35  tty48  tty60
block            fd       input     mapper              ptp0   ram6    sda7      tty10  tty23  tty36  tty49  tty61
bsg              full     kmsg      mcelog              pts    ram7    sda8      tty11  tty24  tty37  tty5   tty62
btrfs-control    fuse     kvm       mei0                ram0   ram8    serial    tty12  tty25  tty38  tty50  tty63
bus              hidraw0  lightnvm  mem                 ram1   ram9    sg0       tty13  tty26  tty39  tty51  tty7
char             hidraw1  log       memory_bandwidth    ram10  random  shm       tty14  tty27  tty4   tty52  tty8
console          hidraw2  loop0     mt_test_drv_sdio    ram11  rfkill  snapshot  tty15  tty28  tty40  tty53  tty9
core             hpet     loop1     net                 ram12  rtc     snd       tty16  tty29  tty41  tty54  ttyACM0
cpu              hwrng    loop2     network_latency     ram13  rtc0    stderr    tty17  tty3   tty42  tty55  ttyprintk
cpu_dma_latency  i2c-0    loop3     network_throughput  ram14  sda     stdin     tty18  tty30  tty43  tty56  ttyS0
cuse             i2c-1    loop4     null                ram15  sda1    stdout    tty19  tty31  tty44  tty57  ttyS1
disk             i2c-2    loop5     port                ram2   sda2    tpm0      tty2   tty32  tty45  tty58  ttyS10
dri              i2c-3    loop6     ppp                 ram3   sda3    tty       tty20  tty33  tty46  tty59  ttyS11
ecryptfs         i2c-4    loop7     psaux               ram4   sda5    tty0      tty21  tty34  tty47  tty6   ttyS12
```

Figure 6: Linux USB driver DL via virtual COM port

2.3 MT793X_QFN_RFB / MT793X_BGA_RFB Download Operation

2.3.1 MT793X_QFN_RFB UART Download Mode

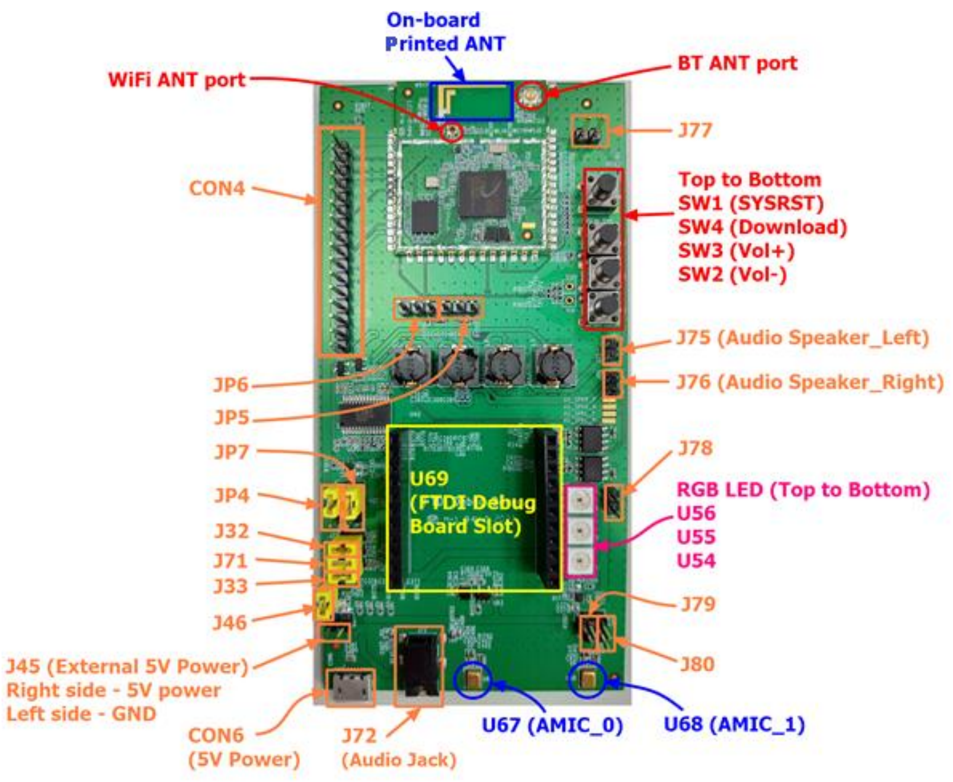


Figure 7: MT793X QFN_RFB debug board UART DL operation

For download key and SYSRST key usage, please also refer to Section 5.3.

1) **Keep pressing "SW1 and SW4" and release "SW4" only** and then wait for FlashBurningTool's feedback.
2) If the FlashBurningTool does not feedback (the progress bar does not turn yellow) after about 2 to 3 seconds, repeat Step 1.
3) If the progress bar becomes yellow, it means the PC and platform handshaking is successful. You can release both SW1 and SW4 keys. Then, the progress bar turns green to indicate the progress.

2.3.2 MT793X_BGA_RFB UART/USB Download Mode Selection

MT793X_BGA_RFB_V20 has USB and UART download paths. If you select USB download mode, the UART path is disabled.

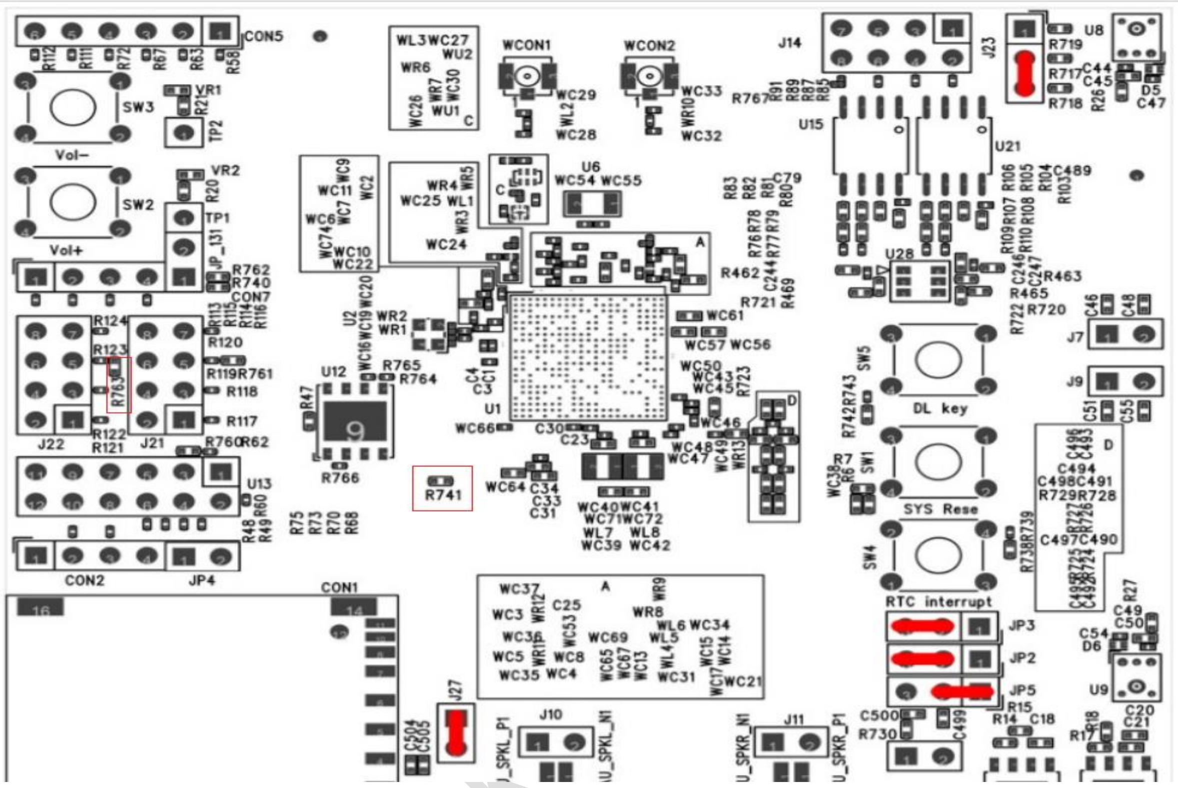


Figure 8: MT793X_BGA_RFB, UART/USB DL mode selection

(1) UART DL mode: Remove R763, and R741 = 10K.(default)

(2) USB DL mode: Remove R741, and R763 = 10K

For details of USB DL mode and normal mode, please refer to Section 2.2.3.

2.3.3 MT793X_BGA_RFB UART Download Mode

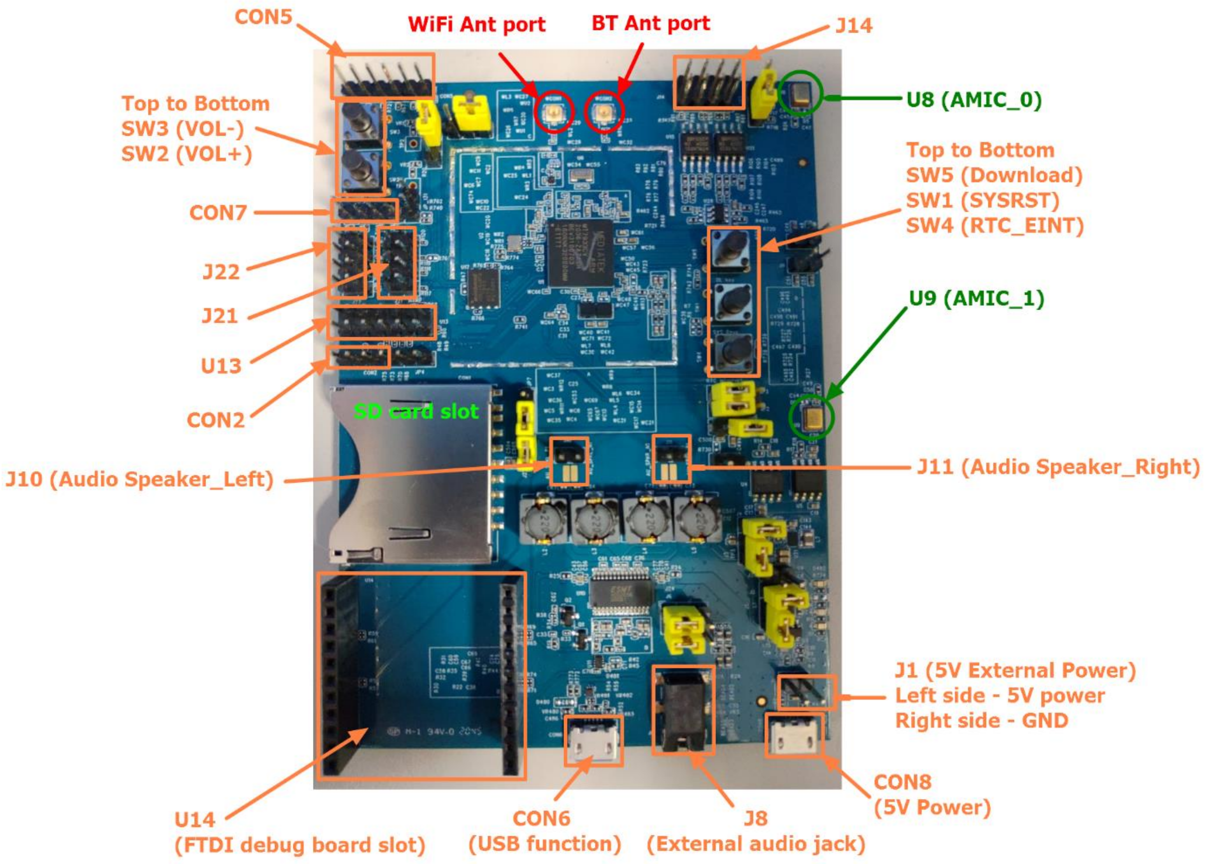
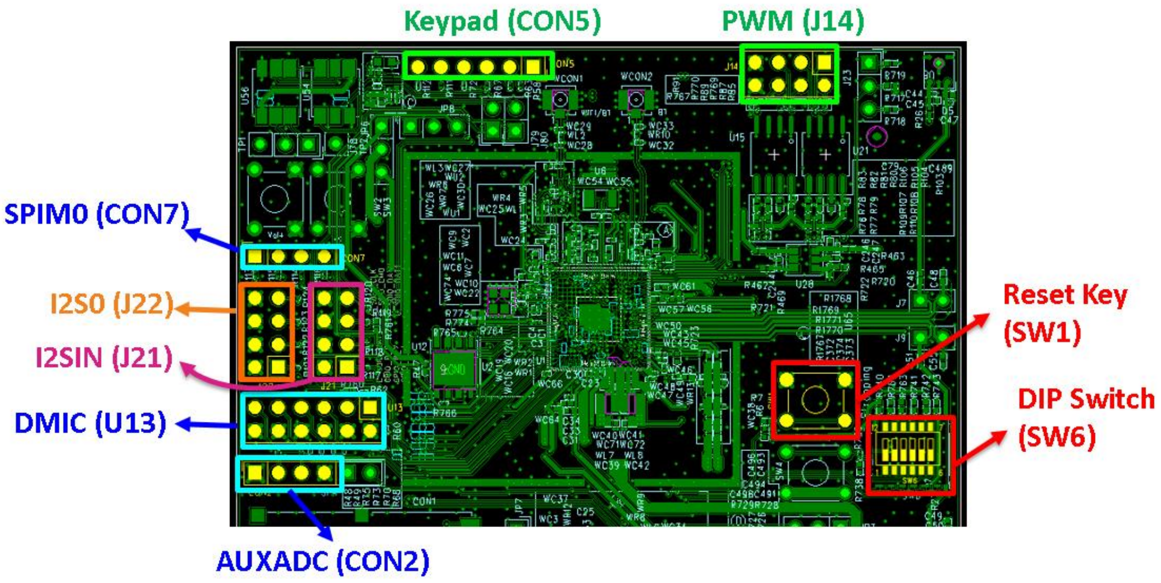


Figure 9: MT793X BGA_RFB Debug board (V20) UART DL operation



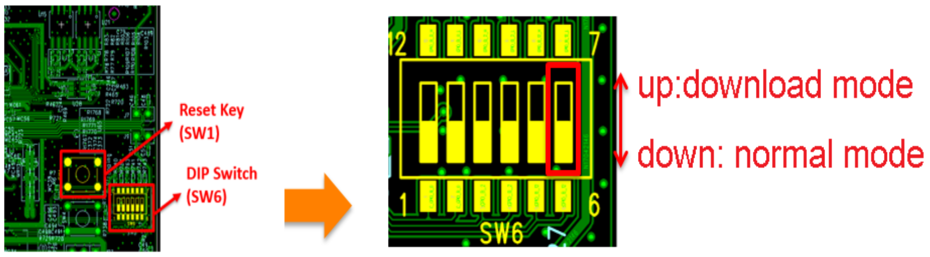


Figure 10: MT793X BGA_RFB Debug board (V21/V22) UART DL operation

2.3.4 MT793X_BGA_RFB USB Download Mode

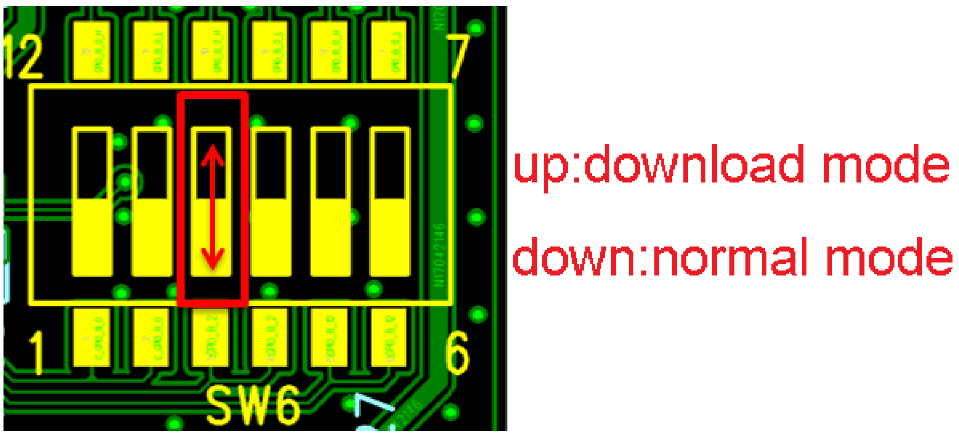
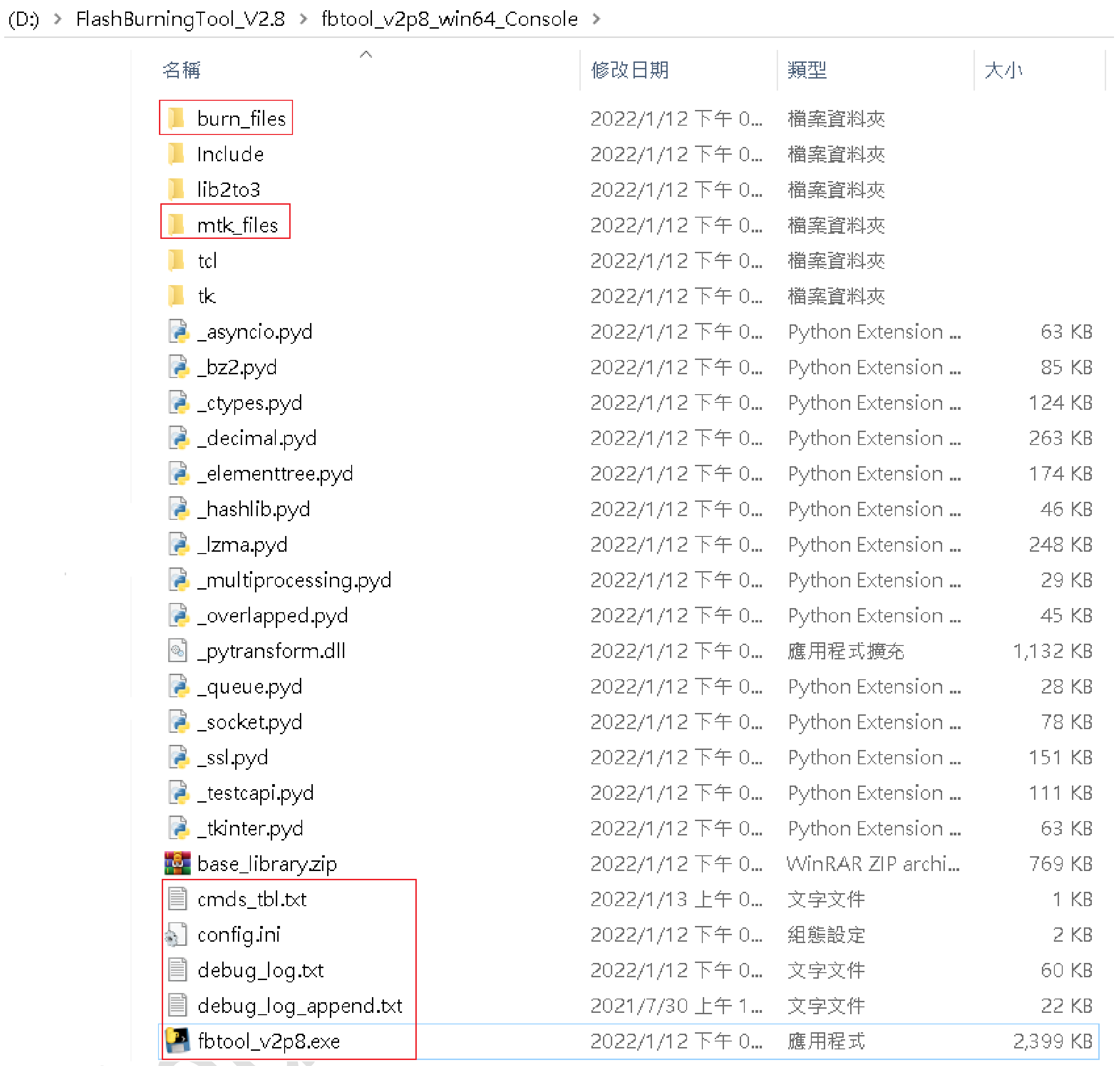


Figure 11: MT793X BGA_RFB Debug board (V21/V22) USB DL operation

MT793X_BGA_RFB_V20: You must power off the platform by unplugging the USB power cable first, and then run the FlashBurningTool command. After you see the log of wait for PC/DA handshaking, you can plug in the USB power cable to power on the platform to finish the command.
Please refer to Chapter 5 for details.
MT793X_BGA_RFB_V21/V22: You must power on the platform in normal mode. Then, change DIP to download mode, and then run the FlashBurningTool command. After you see the log of wait for PC/DA handshaking, you can press the reset key to finish the command.

3 Software Folder Structure and File Description



(D:) > FlashBurningTool_V2.8 > fbtool_v2p8_win64_Console >

名稱	修改日期	類型	大小
burn_files	2022/1/12 下午 0...	檔案資料夾	
Include	2022/1/12 下午 0...	檔案資料夾	
lib2to3	2022/1/12 下午 0...	檔案資料夾	
mtk_files	2022/1/12 下午 0...	檔案資料夾	
tcl	2022/1/12 下午 0...	檔案資料夾	
tk	2022/1/12 下午 0...	檔案資料夾	
_asyncio.pyd	2022/1/12 下午 0...	Python Extension ...	63 KB
_bz2.pyd	2022/1/12 下午 0...	Python Extension ...	85 KB
_ctypes.pyd	2022/1/12 下午 0...	Python Extension ...	124 KB
_decimal.pyd	2022/1/12 下午 0...	Python Extension ...	263 KB
_elementtree.pyd	2022/1/12 下午 0...	Python Extension ...	174 KB
_hashlib.pyd	2022/1/12 下午 0...	Python Extension ...	46 KB
_lzma.pyd	2022/1/12 下午 0...	Python Extension ...	248 KB
_multiprocessing.pyd	2022/1/12 下午 0...	Python Extension ...	29 KB
_overlapped.pyd	2022/1/12 下午 0...	Python Extension ...	45 KB
_pytransform.dll	2022/1/12 下午 0...	應用程式擴充	1,132 KB
_queue.pyd	2022/1/12 下午 0...	Python Extension ...	28 KB
_socket.pyd	2022/1/12 下午 0...	Python Extension ...	78 KB
_ssl.pyd	2022/1/12 下午 0...	Python Extension ...	151 KB
_testcapi.pyd	2022/1/12 下午 0...	Python Extension ...	111 KB
_tkinter.pyd	2022/1/12 下午 0...	Python Extension ...	63 KB
base_library.zip	2022/1/12 下午 0...	WinRAR ZIP archi...	769 KB
cmds_tbl.txt	2022/1/13 上午 0...	文字文件	1 KB
config.ini	2022/1/12 下午 0...	組態設定	2 KB
debug_log.txt	2022/1/12 下午 0...	文字文件	60 KB
debug_log_append.txt	2021/7/30 上午 1...	文字文件	22 KB
fbtool_v2p8.exe	2022/1/12 下午 0...	應用程式	2,399 KB

Figure 12: Software folder structure and file description

3.1 executable files

The root directory contains: executable files:

FBTool_gui.exe (run in win64-GUI mode)

fbtool.exe (run in win64-console mode)

FBTool_gui (run in Linux64-GUI mode)

fbtool (run in Linux64-console mode)

run_FBTool_gui.py (run in Linux64-python37-GUI mode)

run_fbtool_console.py (run in Linux64--python37-console mode)

3.2 Config.ini File

Config.ini: Configure this tool to set up the environment.

3.3 Folders

The "burn_files" folder is an optional folder for you to put necessary image files that you want to burn. It also With some necessary system environment files of the application, the folder can be defined as the default working folder.

The "mtk_files" folder contains some security files such as mt7933_auth.auth, mt7933_cert.cert and DA version file.

3.4 Scatter File

Scatter file: the file configures flash partitions and parameters. The scatter file can be built and output to the "out" folder of the project after SDK code built:
..\out\MT793X_hdk\bootloader\ mt7931an_bootloader_scatter.ini
..\out\MT793X_hdk\iot_sdk_demo\ iot_sdk_demo_scatter.ini
The scatter file must be put in the same folder where images files are located, which is defined as the working folder.
The default working folder is "burn_files" and the default scatter file is "scatter.ini"

3.5 Debug Log

Debug_log.txt is the recorded log message of the execution process. The log file is cleaned when FlashBurningTool is opened. DL_fa_log_append.txt is the Debug_log.txt appended when an error occurs. You can analyze this file to debug.

3.6 Commands Table File

Cmds_tbl.txt is the command lines list. The commands can be run in console mode of the product line. The commands are run in bypass BROM mode to reduce time. So, there is no need to power on/off between commands. Please also refer to Section 6.9.

4 Supported Commands

4.1 Format Command

Erase some partitions or erase all partitions:
Click "select all partitions" to erase the whole flash. Or, select some partitions, and the Flash Tool erases the selected partitions of the flash accordingly.
Console mode: it can order command:
-o erase

4.2 Download Command

When the bin file is burned to each partition, the flash is erased before data is written to the flash.
After the image bin files are burned to the target device, if the burn operation completes successfully, the burning ok message appears.
Console mode: it can order command:
-o burn
For reduce time on product line, user can enable burn only feature by order command on console mode:
-o burn_o
burn_o is a special command that can be applied after the whole flash is erased fully, and the command only burns images bin.

4.3 Read Back Command

To read back the flash, click the "ReadBack" button on the GUI. The ReadBack command outputs the read back file; the output path is the same as assigned in each partition. The name of the output file is the original file name +_readback
E.g. Original file name: bootloader.bin
Read back output file name: bootloader_readback.bin
The original file and the read back file are compared; if the contents are not exactly the same, an error occurs.
Console mode: it can order command:
-o rdback
For reduce time on product line, user can enable read small size feature by order command on console mode:
-o rdback_s
if GUI mode want to enable, it must assign it on config.ini:[GUI_CONFIG]:
rdback_small_confirm_gui=y

when read back small size feature be enable, the readback size can be assigned on config.ini:[MP_Feature]:
rdback_small_size=xxxx ,

the suggested rdback_small_size is 4096. If the image file size is smaller than rdback_small_size, rdback_small_size will change to image file size.

4.4 Read eFuse Command

The command supports logical/physical eFuse reads of GRP1, GRP2, and GRP3 by using specific control data in GUI mode or by argument input in console mode. You can assign eFuse bin file name in the scatter file. This command can read full group data, the size of which is 0x200 (GRP1 and GRP2) or 0x400 (GRP3). The command can also read offset address (aligned with 0x10) data of each GRP, the size of which is 16 bytes.
In customer mode, to read GRP1 eFuse, you can run the read eFuse command directly on GUI. The command outputs the result in the file customer_ef_tbl.txt under the working folder.
In console mode, you can assign eFuse file name and file path by input argument.

```
-o rdefuse -e path_of_efuse_tbl.txt
```

The tool supports another command “rdefuse_c” for reading back eFuse data to confirm the correctness after the write eFuse command.
Example:

```
-o rdefuse_c   -e Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/my_ef_tbl.txt
```

The command reads back eFuse data to another file as: my_ef_tbl.txt(readback) and compares the file with my_ef_tbl.txt,
After the files are compared, if there is any error, you see an error message.

4.5 Write eFuse Command

The command supports eFuse writes of GRP1 only, by using specific control data on GUI or by argument input in console mode. You can assign eFuse bin file name in the scatter file. This command can write full group data, the size of which is 0x200 (GRP1). The command can also read offset address (aligned with 0x10) data of GRP1, the size of which is 16 bytes.
In GUI mode, the write eFuse command can be opened by selecting the checkbox after clicking GUI’s Settings button.
In customer mode, to write GRP1 eFuse, you can run the write eFuse command directly on GUI. The command writes the eFuse according to the file customer_ef_tbl.txt located under the working folder.
In console mode, you can assign eFuse file name and file path by input argument.

```
-o wrefuse -e path_of_efuse_tbl.txt
```

To enable writing eFuse content to the platform, you must assign: “blow enable=y” in your eFuse_table.txt.

4.6 Customer Command on cmds_tab.txt

Customer can input any command and it’s inupt argument on cmds_tab.txt ,the command line format as:

```
-o customer -s [customer command and input arguments ]
```

For windows executable file example:

```
-o customer -s  customer_cmd.exe -k c:/data.txt
```

For linux executable file example:

```
-o customer -s  ./customer_cmd  argument1 argument2 ….
```

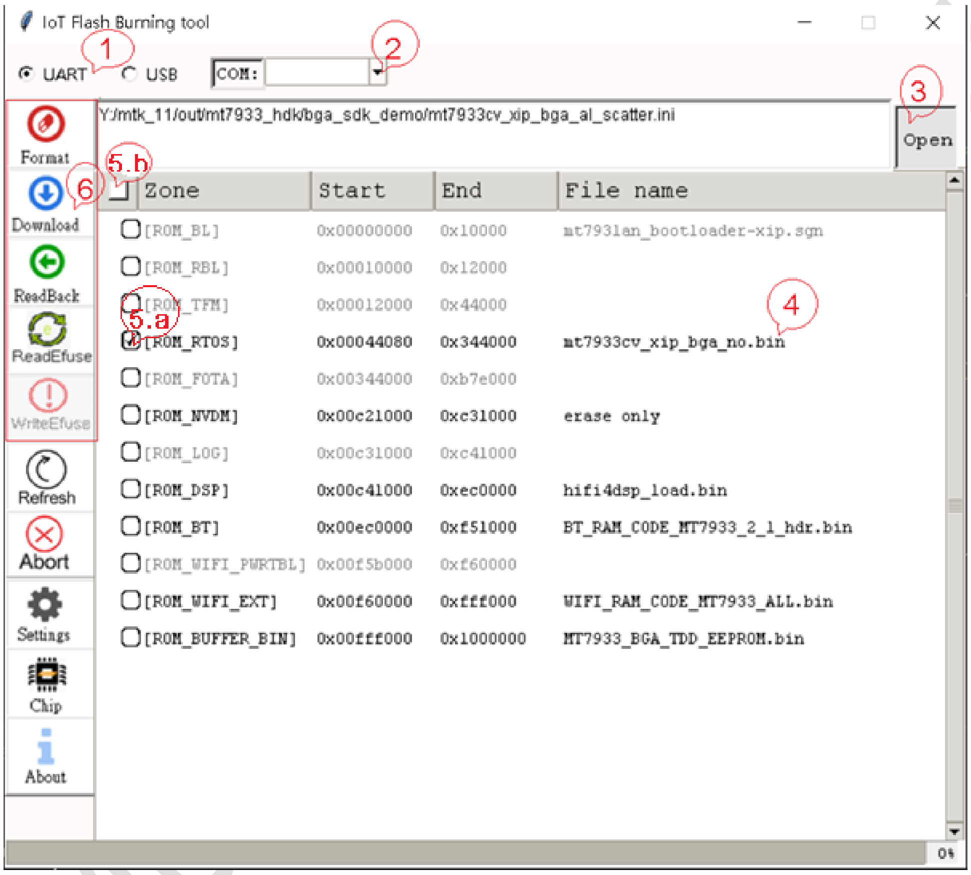
Customer command must locate on root of fbtool ,and this mode only valid on cmds_tab.txt.

5 GUI Mode Operation Sequence

5.1 General Description

You can execute FlashBurningTool_Vx_GUI.exe first. Then, set the connection path and file location. Select partitions and choose one command to run. The tool waits to perform handshaking with the platform. Then, power on the platform with the hardware jumper in download mode. In order to start the protocol of each other. When handshaking successfully ,it start to run command.

5.2 Operation Sequence



Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini

	Zone	Start	End	File name
☐	[ROM_BL]	0x00000000	0x10000	mt7931an_bootloader-xip.sgn
☐	[ROM_RBL]	0x00010000	0x12000	
☐	[ROM_TFM]	0x00012000	0x44000	
☑	[ROM_RTOS]	0x00044080	0x344000	mt7933cv_xip_bga_no.bin
☐	[ROM_FOTA]	0x00344000	0xb7e000	
☐	[ROM_NVDM]	0x00c21000	0xc31000	erase only
☐	[ROM_LOG]	0x00c31000	0xc41000	
☐	[ROM_DSP]	0x00c41000	0xec0000	hifi4dsp_load.bin
☐	[ROM_BT]	0x00ec0000	0xf51000	BT_RAM_CODE_MT7933_2_1_hdr.bin
☐	[ROM_WIFI_PWRTBL]	0x00f5b000	0xf60000	
☐	[ROM_WIFI_EXT]	0x00f60000	0xfff000	WIFI_RAM_CODE_MT7933_ALL.bin
☐	[ROM_BUFFER_BIN]	0x00fff000	0x1000000	MT7933_BGA_TDD_EEPROM.bin

Figure 13: GUI mode operation sequence

General operating steps:

(Step 0) Power off the platform.

QFN_RFB/BGA_RFB in UART DL mode: Download key + SYSRST key: refer to Section 2.3.1.

BGA_RFB in USB DL mode: unplug the USB power cable.

(Step 1) Select the connection path. (UART or USB)

(Step 2) Select the COM port if you use the UART path.

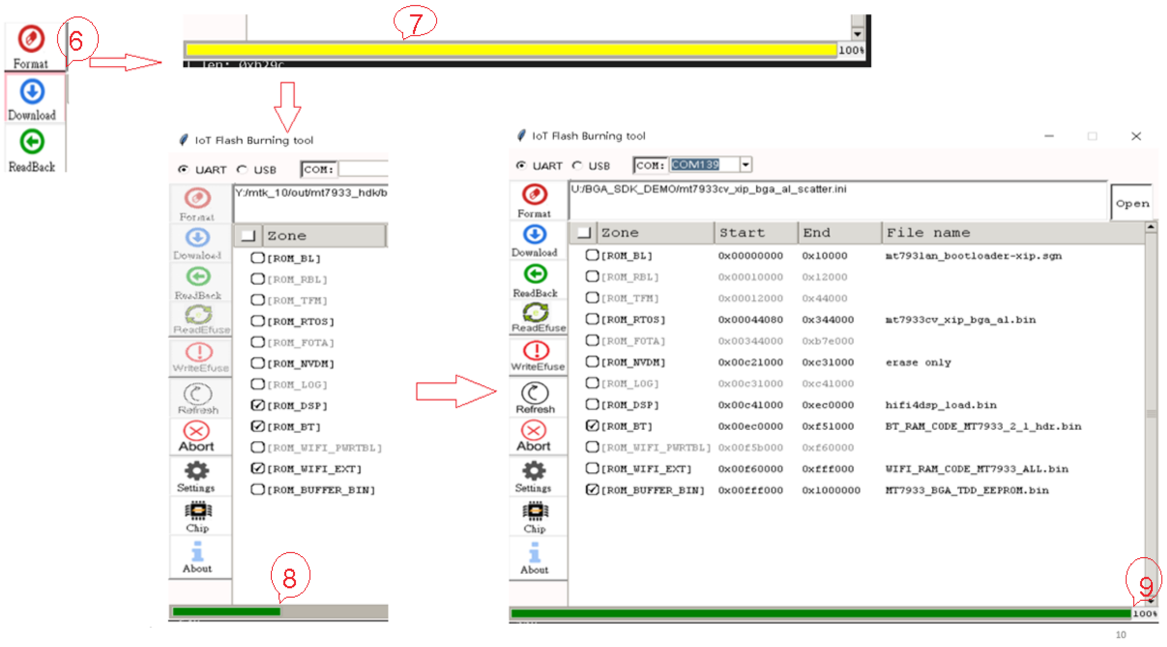
(Step 3) Select the working folder to store the burned images and scatter file.

(Step 4) Assign file names in the scatter file (where the files are located) and then you can select the corresponding checkbox.

(Step 5.a) Select zones.(multiple zones to choose from; only valid for Format, Download, ReadBack commands)

(Step 5.b) You can select or clear all zones. (Optional)

(Step 6.a) Select one command to run.



IoT Flash Burning tool

UART ● USB ○ COM: COM139

U:/BGA_SDK_DEMO/mt7933cv_xip_bga_al_scatter.ini

	Zone	Start	End	File name
☐	[ROM_BL]	0x00000000	0x10000	mt7931an_bootloader-xip.sgn
☐	[ROM_RBL]	0x00010000	0x12000	
☐	[ROM_TFM]	0x00012000	0x44000	
☐	[ROM_RTOS]	0x00044080	0x344000	mt7933cv_xip_bga_al.bin
☐	[ROM_FOTA]	0x00344000	0xb7e000	
☐	[ROM_NVDM]	0x00c21000	0xc31000	erase only
☐	[ROM_LOG]	0x00c31000	0xc41000	
☐	[ROM_DSP]	0x00c41000	0xec0000	hifi4dsp_load.bin
✓	[ROM_BT]	0x00ec0000	0xf51000	BT_RAM_CODE_MT7933_2_1_hdr.bin
☐	[ROM_WIFI_PWRTBL]	0x00f5b000	0xf60000	
☐	[ROM_WIFI_EXT]	0x00f60000	0xfff000	WIFI_RAM_CODE_MT7933_ALL.bin
✓	[ROM_BUFFER_BIN]	0x00fff000	0x1000000	MT7933_BGA_TDD_EEPROM.bin

Figure 14: Process of running a command

(Step 6.b)

QFN_RFB/BGA/RFB in UART DL mode: Download key + SYSRST key: refer to Section 2.3.1.

BGA_RFB in USB DL mode: plug in the USB power cable.

(Step 7) If the progress bar turns yellow, it means handshaking is successful.

(Step 8) Wait for the progress bar to turn green.

(Step 9) Wait for the progress bar to turn completely green to indicate the task is finished.

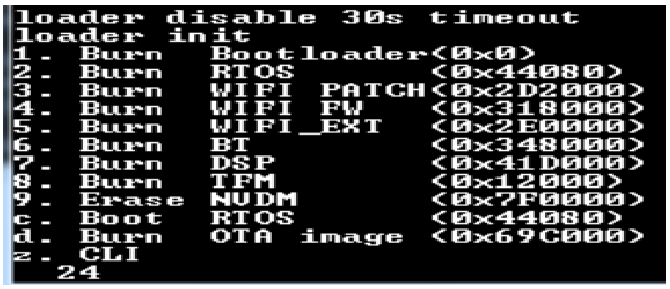
(Step 10) Do not power off the platform if you want to run the command again. Repeat Steps 5 to 9.

(Step 11) Change to normal mode:

QFN_RFB in UART DL mode: press the SYSRST key: refer to Section 2.3.1.

BGA_RFB: unplug and then plug in the USB power cable.

(step 12) Run FlashBurningTool again: re-start flashburning.exe if you already exit the GUI window. If not, press “Refresh” and then repeat Step 0.

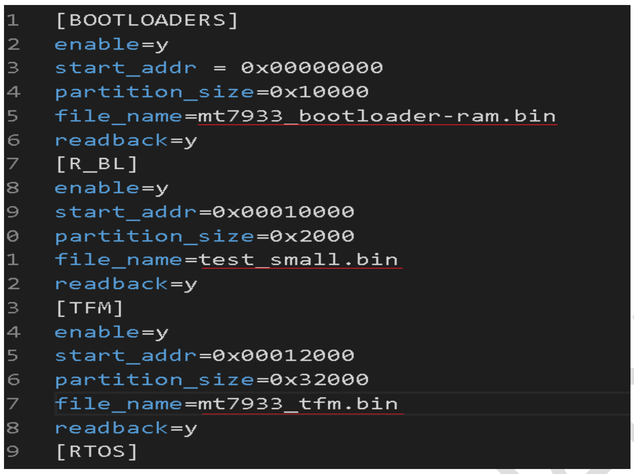


```
loader disable 30s timeout
loader init
1. Burn    Bootloader(0x0)
2. Burn    RTOS       (0x44080)
3. Burn    WIFI PATCH(0x2D2000)
4. Burn    WIFI FW    (0x318000)
5. Burn    WIFI_EXT   (0x2E0000)
6. Burn    BT         (0x348000)
7. Burn    DSP        (0x41D000)
8. Burn    TFM        (0x12000)
9. Erase   NVDM       (0x7F0000)
c. Boot    RTOS       (0x44080)
d. Burn    OTA image  (0x69C000)
z. CLI
 24
```

Figure 15: Normal mode UART log of strap to bootloader

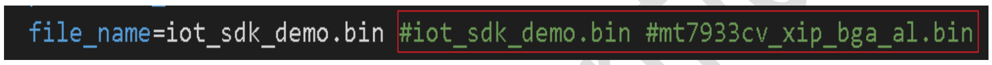
Step 4 can be further elaborated as follows.

(Step 4.a) Assign bin file names in scatter.ini:



```
[BOOTLOADERS]
enable=y
start_addr = 0x00000000
partition_size=0x10000
file_name=mt7933_bootloader-ram.bin
readback=y
[R_BL]
enable=y
start_addr=0x00010000
partition_size=0x2000
file_name=test_small.bin
readback=y
[TFM]
enable=y
start_addr=0x00012000
partition_size=0x32000
file_name=mt7933_tfm.bin
readback=y
[RTOS]
```

Figure 16: Assign file names in scatter file

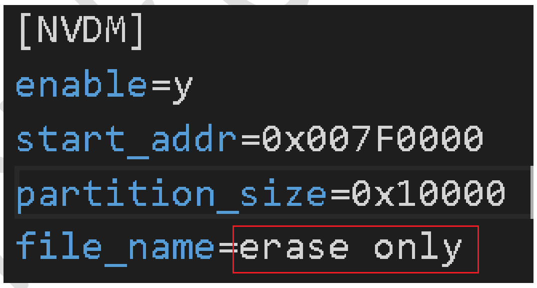


```
file_name=iot_sdk_demo.bin #iot_sdk_demo.bin #mt7933cv_xip_bga_al.bin
```

You can add a temp string after normal file name as shown above.

(Step 4.b) NVDM zone:

For NVDM, you do not need to assign the file name, which is named as "erase only" because it is an erase only area.



```
[NVDM]
enable=y
start_addr=0x007F0000
partition_size=0x10000
file_name=erase only
```

Figure 17: Scatter file of NVDM Zone

5.2.1 Bypass BROM

If you run a command for the first time, after the command is finished and you want to run the command again, you can run the command directly and do not need to power off the platform. However, if you have clicked the "Refresh" button to release bypass BROM mode, you need to power off the platform for the next command.

Or, if you power off the platform but do not close the GUI window, click the "Refresh" button to release bypass BROM mode.

5.3 Some GUI Element Descriptions

5.3.1 Refresh Button

The Refresh button can

(1) Reset the tool to the initial state.
(2) Reload scatter file content.
(3) Reset bypass BROM mode.

If you do not assign the bin file or the working folder is empty, or if you have the wrong file name in "file_name=", you cannot select the checkbox of the partition zone. The tool shows a warning message to remind you. If you assign the bin file correctly, you can click the "Refresh" button to update scatter content.

5.3.2 Abort Button

If you are running a command, but you want to abort the command, you can click the "Abort" button. Note that the command does not always respond immediately.

6 Console Mode Operation Sequence

6.1 General Description

The UI is mainly controlled by config.ini and scatter.ini. After you plan ini, just execute fbtooll.exe directly. Then, power on the platform with hardware jumper in download mode.

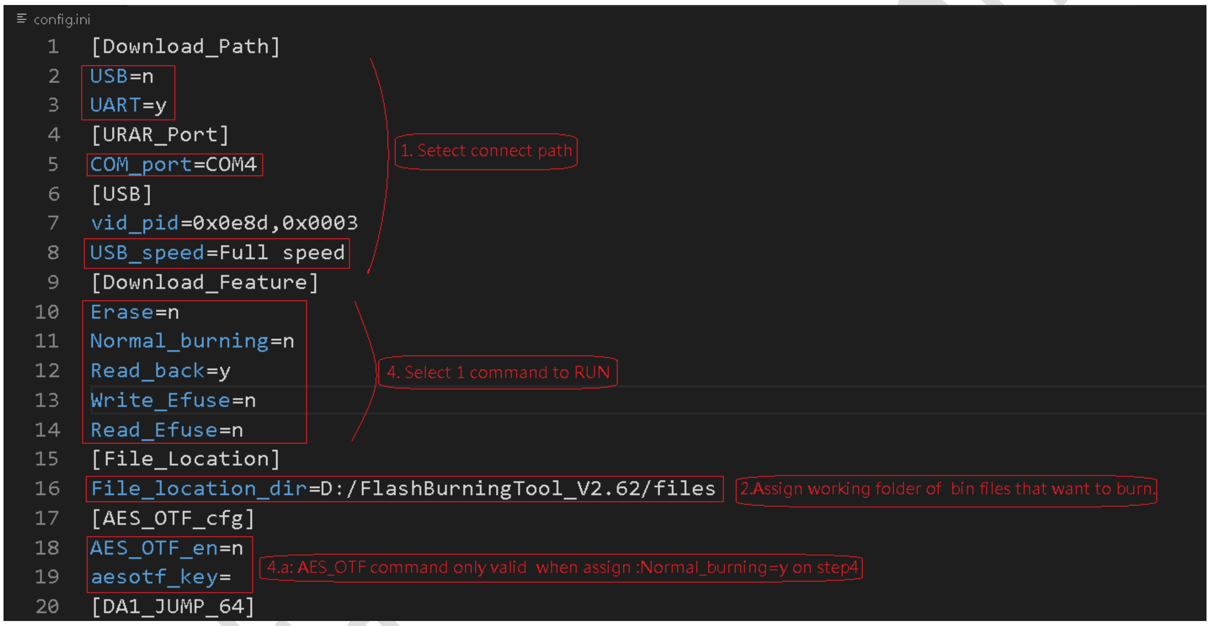
6.2 General Operation Sequence

Step 0: Power off the platform.

QFN_RFB/BGA/RFB in UART DL mode: Download key+SYSRST key: refer to Section 2.3.1.

BGA_RFB in USB DL mode: unplug the USB power cable.

Step 1: Select a connection path:

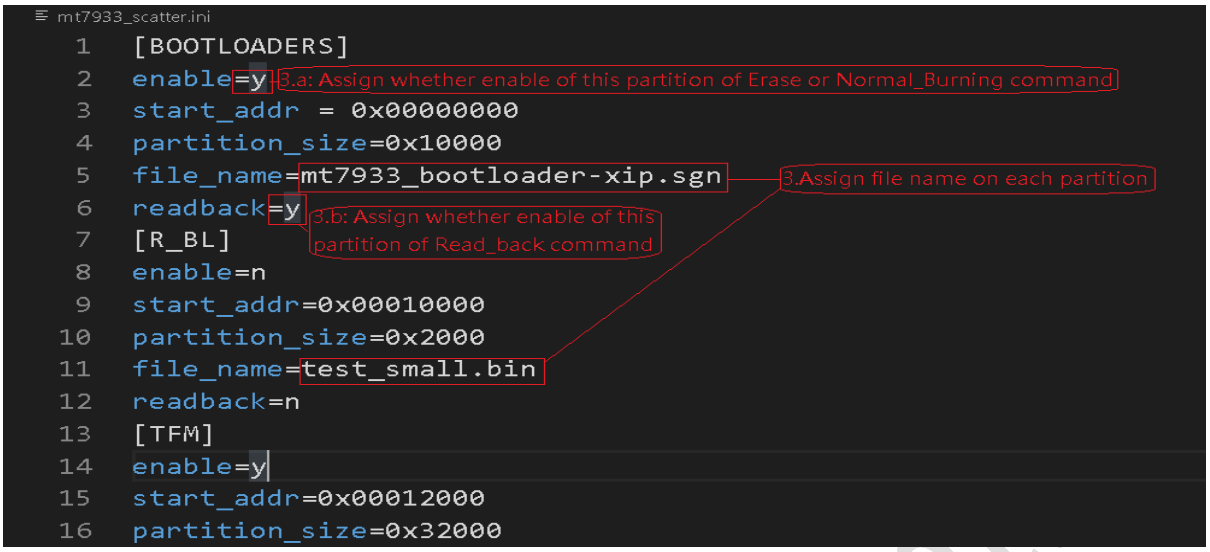


config.ini

```
[Download_Path]
USB=n
UART=y
[URAR_Port]
COM_port=COM4
[USB]
vid_pid=0x0e8d,0x0003
USB_speed=Full speed
[Download_Feature]
Erase=n
Normal_burning=n
Read_back=y
Write_Efuse=n
Read_Efuse=n
[File_Location]
File_location_dir=D:/FlashBurningTool_V2.62/files
[AES_OTF_cfg]
AES_OTF_en=n
aesotf_key=
[DA1_JUMP_64]
```

1. Select connect path
4. Select 1 command to RUN
2.Assign working folder of bin files that want to burn.
4.a: AES_OTF command only valid when assign :Normal_burning=y on step4

Figure 18: Console mode operating step: Config.ini



```
mt7933_scatter.ini
[BOOTLOADERS]
enable=y
start_addr = 0x00000000
partition_size=0x10000
file_name=mt7933_bootloader-xip.sgn
readback=y
[R_BL]
enable=n
start_addr=0x00010000
partition_size=0x2000
file_name=test_small.bin
readback=n
[TFM]
enable=y
start_addr=0x00012000
partition_size=0x32000
```

3.a: Assign whether enable of this partition of Erase or Normal_Burning command
3.Assign file name on each partition
3.b: Assign whether enable of this partition of Read_back command

Figure 19: Console mode operating step: assign data in scatter file

(1.a) Select the UART path to connect:

In config.ini, when UART = “y” and “COM_port” is specified, the UART download path can be selected first.

(1.b) Select the USB path to connect:

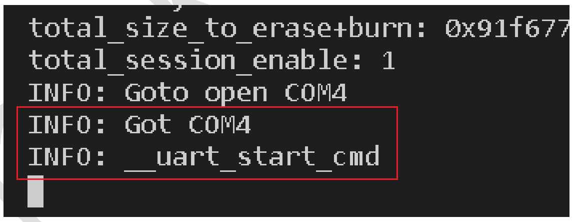
If the above conditions are not met, and if you set USB = “y”, and have correct “Vid” and “Pid” values, you can select the USB path to connect.

Step 2: Assign the working folder of bin files that you want to burn.

Step 3: Assign file name of each partition, and choose whether to enable commands.

Step 4: Select the first command to run. Only assign “y” to the first command; assign “n” to other commands.

Step 5: Run fbtool_vx.exe.



```
total_size_to_erase+burn: 0x91f677
total_session_enable: 1
INFO: Goto open COM4
INFO: Got COM4
INFO: __uart_start_cmd
```

Figure 20: UART log of wait for user handshaking

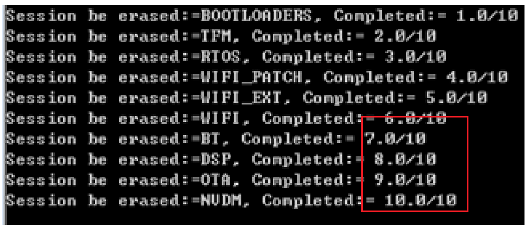
If you select UART connection, you see the log, “__uart_start_cmd”; otherwise, you must check your COM port setting.

Step 6:

QFN_RFB/BGA/RFB in UART DL mode: Download key + SYSRST key: refer to Section 2.3.1.

BGA_RFB in USB DL mode: plug-in the USB power cable.

Step 7: Wait for the command to finish (check the log message.)



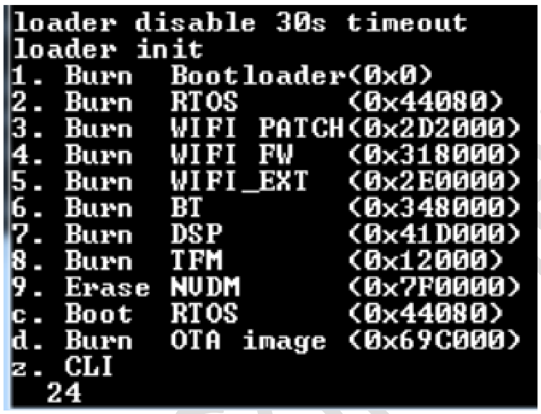
```
Session be erased:=BOOTLOADERS, Completed:= 1.0/10
Session be erased:=TFM, Completed:= 2.0/10
Session be erased:=RTOS, Completed:= 3.0/10
Session be erased:=WIFI_PATCH, Completed:= 4.0/10
Session be erased:=WIFI_EXT, Completed:= 5.0/10
Session be erased:=WIFI, Completed:= 6.0/10
Session be erased:=BT, Completed:= 7.0/10
Session be erased:=DSP, Completed:= 8.0/10
Session be erased:=OTA, Completed:= 9.0/10
Session be erased:=NVDM, Completed:= 10.0/10
```

Figure 21: UART log of command running process

Step 8: Power off the platform.

Step 9a: Repeat Step 3 to Step 8, if you want to run other commands.

Step 9b: The hardware jumper changes to normal mode and then powers on the platform; then, MT793X-bootloader-xip.sgn starts to run.



```
loader disable 30s timeout
loader init
1. Burn  Bootloader(0x0)
2. Burn  RTOS       (0x44080)
3. Burn  WIFI PATCH(0x2D2000)
4. Burn  WIFI FW    (0x318000)
5. Burn  WIFI_EXT   (0x2E0000)
6. Burn  BT         (0x348000)
7. Burn  DSP        (0x41D000)
8. Burn  TFM        (0x12000)
9. Erase NVDM       (0x7F0000)
c. Boot  RTOS       (0x44080)
d. Burn  OTA image  (0x69C000)
z. CLI
  24
```

Figure 22: Normal mode UART log of run bootloader

6.3 Scatter File Input Argument

You can select different scatter files for some specific applications.

fbtool.exe –f path_of_my_scatter.ini

Example:

fbtool.exe -f U:/BGA_SDK_DEMO/mt7933cv_xip_bga_al_scatter.ini

If there is no input argument, the default scatter file is burn_files/scatter.ini

The scatter file must be put in the working folder. In this case, the working folder is: U:/BGA_SDK_DEMO/

All images files are put in the working folder.

6.4 eFuse File Input Argument

You can select different eFuse file names for where the files are located for different applications.

fbtool.exe –e path_of_my_efuse_tbl.txt

Example:

fbtool.exe -e Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/my_ef_tbl.txt

fbtool.exe -e Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/my_ef_tbl_1.txt

If there is no input argument, the default file is the working folder/customer_ef_tbl.txt

6.5 Control Data Input Argument

You can input 4-byte control data to read or write eFuse. This argument is only valid for reading or writing eFuse commands.

```
fbtool.exe –s control_data
```

Example:

```
fbtool.exe –s 0x020001a0
fbtool.exe –s 0x34060000
```

P.S. You can read or write the eFuse without needing to input control data.
For more information, see the next chapter.

another application is customer command,please reference section 4.6.

6.6 Order Command Input Argument

You can select different commands by input argument.

```
fbtool.exe –o command_data
```

Example:

```
fbtool.exe.exe -o erase
fbtool.exe.exe -o burn
fbtool.exe.exe -o burn_o
fbtool.exe.exe -o rdback
fbtool.exe.exe -o rdback_s
fbtool.exe.exe -o wrefuse –s 0x01ff0000
fbtool.exe.exe -o rdefuse  –s 0x01ff0000
fbtool.exe.exe -o wrefuse –e  path_of_efuse_table.txt
fbtool.exe.exe -o rdefuse  –e  path_of_efuse_table.txt
fbtool.exe.exe -o rdefuse_c    path_of_efuse_table.txt
fbtool.exe.exe -o cmds_tbl
```

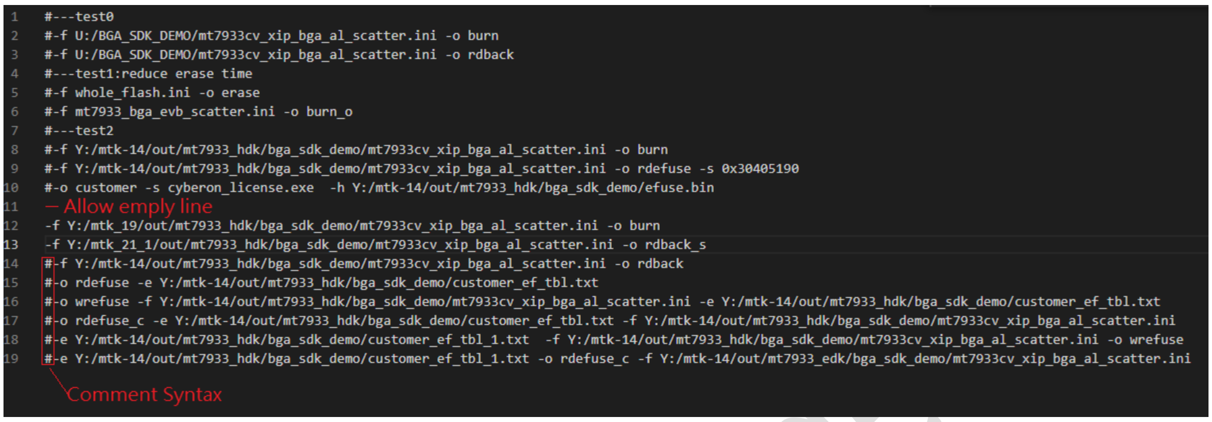
-o customer command only valid on cmds_tab.txt:
-o customer my_cmd my_cmd_input_arguments

6.7 Run Command Table Input Argument

You can run commands in cmds_tbl.txt. The commands are run in bypass BROM mode to reduce time, so there is no need to power on/off between commands.

fbtool.exe –o cmds_table

Example:



```
#---test0
#-f U:/BGA_SDK_DEMO/mt7933cv_xip_bga_al_scatter.ini -o burn
#-f U:/BGA_SDK_DEMO/mt7933cv_xip_bga_al_scatter.ini -o rdback
#---test1:reduce erase time
#-f whole_flash.ini -o erase
#-f mt7933_bga_evb_scatter.ini -o burn_o
#---test2
#-f Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -o burn
#-f Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -o rdefuse -s 0x30405190
#-o customer -s cyberon_license.exe  -h Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/efuse.bin

-f Y:/mtk_19/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -o burn
-f Y:/mtk_21_1/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -o rdback_s
#-f Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -o rdback
#-o rdefuse -e Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl.txt
#-o wrefuse -f Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -e Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl.txt
#-o rdefuse_c -e Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl.txt -f Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini
#-e Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl_1.txt  -f Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini -o wrefuse
#-e Y:/mtk-14/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl_1.txt -o rdefuse_c -f Y:/mtk-14/out/mt7933_edk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini
```

Figure 23: Commands table

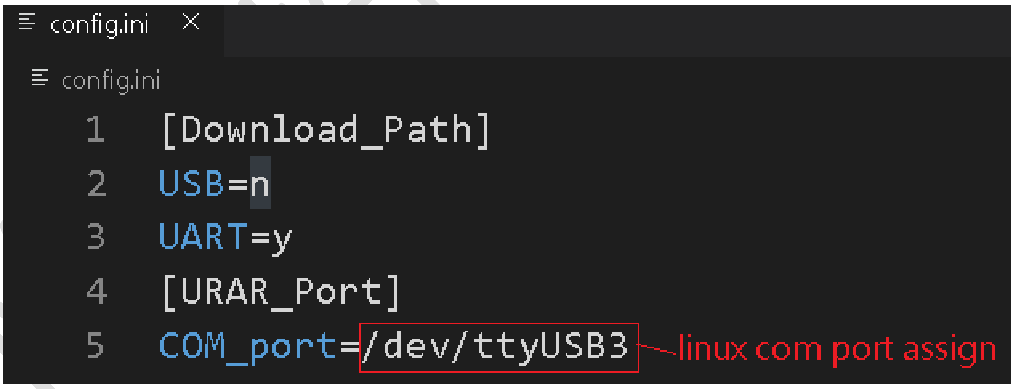
6.8 Run in Linux Console Mode

Running in Linux console mode is almost the same as running in Windows console mode. There are only some differences.

(1) You may need to change the permission the first time:

$chmod 777 ./fbtool_vx

(2) Assign the COM port in config.ini:



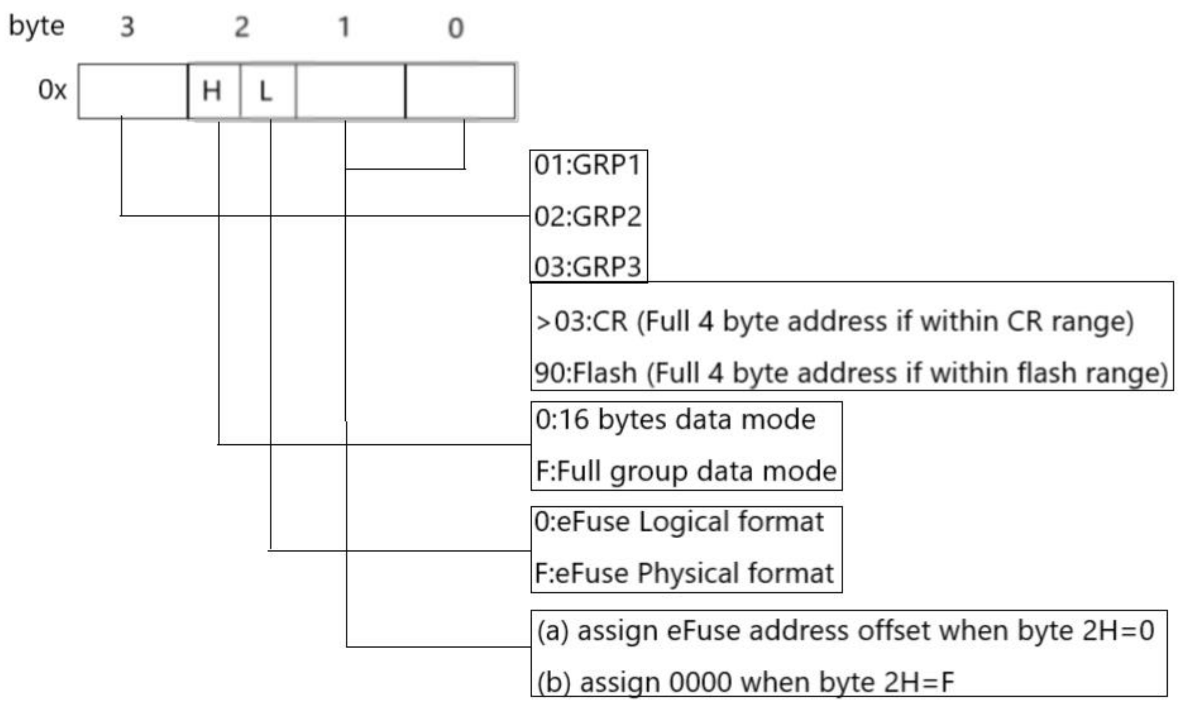
```
[Download_Path]
USB=n
UART=y
[URAR_Port]
COM_port=/dev/ttyUSB3
```

Figure 24: Assign Linux COM port

7 Read/Write eFuse

7.1 Control Data Format.

The control data is used to control the eFuse read/write group, choose whether to get a full dump or partial dump, and assign the offset address of each group. The control data is 4 bytes long.
To read or write the eFuse, you don't need to input control data.



byte 3 2 1 0
0x
H
L
01:GRP1
02:GRP2
03:GRP3
>03:CR (Full 4 byte address if within CR range)
90:Flash (Full 4 byte address if within flash range)
0:16 bytes data mode
F:Full group data mode
0:eFuse Logical format
F:eFuse Physical format
(a) assign eFuse address offset when byte 2H=0
(b) assign 0000 when byte 2H=F

Figure 25: eFuse control data format

You can read GRP1, GRP2, GRP3 of eFuse data by using specific control data in GUI or console modes. The control data can be used to read full data, the size of which is 0x200 (GRP1, GRP2) and 0x400 (GRP3), or assign offset address (aligned with 0x10) to read/write a part of the data of each GRP, the size of which is 16 bytes.
For example:

(1) Ctl-data=0x010001A0: Read/Write GRP1 in 16-byte data mode with offset address = 0x01A0.
(2) Ctl-data=0x02FF0000: Physical reads of GRP2 in full data mode and related efuse.bin.
(3) Ctl-data=0x02F00000: Logical reads of GRP2 in full data mode and related efuse.bin.
(4) Ctl-data=0x02000060: Logical reads of GRP2 of address = 0x60 in 16-byte data mode.
(5) Ctl-data=0x34040030: Read/Write CR in 16-byte data mode, with the address aligned with 0x10.
(6) Ctl-data=0x90000010: Read/Write flash in 16-byte data mode.

7.2 Customer Mode: eFuse Table

In GUI mode, the tool contains a default template file, custumer_ef_tbl.txt, in the “burn_files” folder, and the content data is filled with zeroes. When you select a new working folder, the file is copied to the new working folder. You can edit the file under the working folder. Do not edit the file under the “burn_files” folder.
When you run a read/ write eFuse command, the tool bases the operations on the file under the working folder. You can press the readEfuse or writeEfuse button directly in GUI mode.
For safety, the write eFuse button is disabled by default. You must enable eFuse writing by selecting the checkbox after clicking the Settings button, and then the write eFuse button will change to enabled.
To enable writing eFuse content to the platform, you must assign: “blow enable=y” in your eFuse_table.txt
In console mode, you can select the location of the eFuse file by an input argument. If you don’t select a locaton, the default file location is working folder/customer_ef_tbl.txt
For example:

```
fbtool.exe -o rdefuse  -e Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/my_ef_tbl.txt
```

Or

```
fbtool.exe -o rdefuse  -f  Y:/mtk_10/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini
```

In this case, because there is no input -e argument, so the eFuse file is under:
Y:/mtk_10/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl.txt

Or

```
fbtool.exe -o rdefuse  -e Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/customer_ef_tbl.txt
-f  Y:/mtk_10/out/mt7933_hdk/bga_sdk_demo/mt7933cv_xip_bga_al_scatter.ini
```

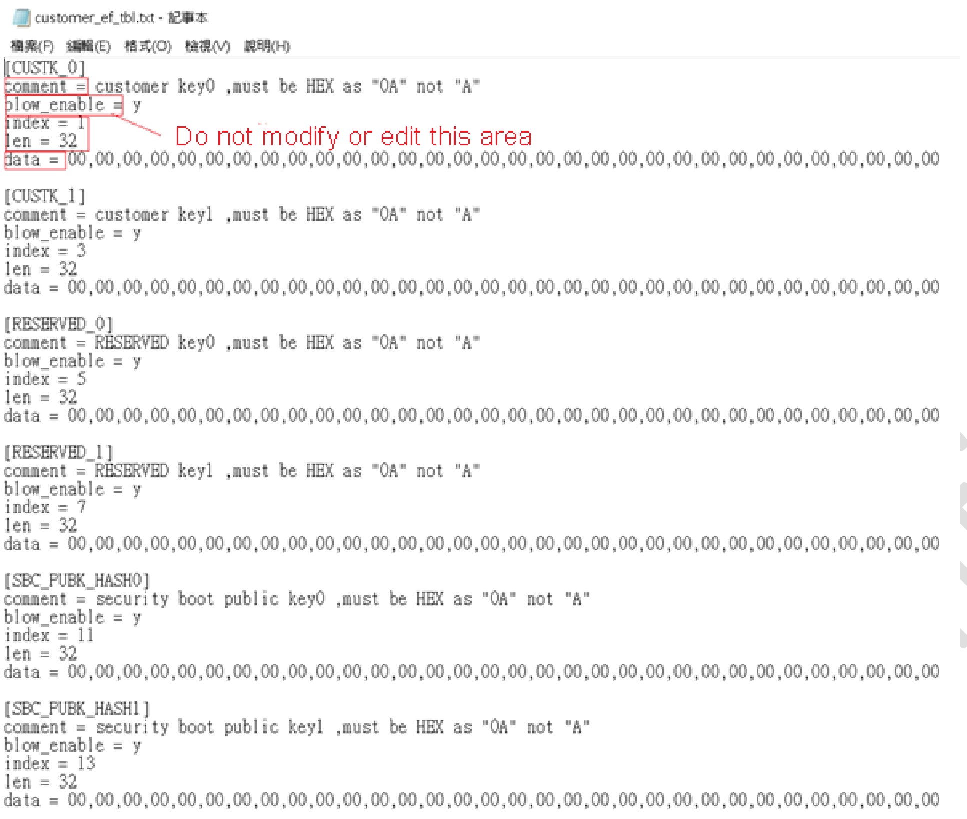
In this case, the -f input argument has no effect for read eFuse function.

Or

```
fbtool.exe -o rdefuse
```

In this case, because there is no -f argument, the default scatter is burn_files/scatter.ini, and eFuse file is: burn_files/ customer_ef_tbl.txt. But this file is the default template file with full data = 0 for the tool. The file can’t be edited or modified. If the file is modified, the tool shows error messages.

Please also refer Sections 6.3, 6.4 and 6.6.


customer_ef_tbl.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Do not modify or edit this area

```
[CUSTK_0]
comment = customer key0 ,must be HEX as "0A" not "A"
blow_enable = y
index = 1
len = 32
data = 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[CUSTK_1]
comment = customer key1 ,must be HEX as "0A" not "A"
blow_enable = y
index = 3
len = 32
data = 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[RESERVED_0]
comment = RESERVED key0 ,must be HEX as "0A" not "A"
blow_enable = y
index = 5
len = 32
data = 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[RESERVED_1]
comment = RESERVED key1 ,must be HEX as "0A" not "A"
blow_enable = y
index = 7
len = 32
data = 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[SBC_PUBK_HASH0]
comment = security boot public key0 ,must be HEX as "0A" not "A"
blow_enable = y
index = 11
len = 32
data = 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[SBC_PUBK_HASH1]
comment = security boot public key1 ,must be HEX as "0A" not "A"
blow_enable = y
index = 13
len = 32
data = 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
```

Figure 26: eFuse table (customer mode)

Figure 26 shows example of eFuse table. Please note the red blocks are the area that can't be edited or modified.

The tool supports another command “rdefuse_c” for reading back eFuse data to confirm the correctness after the write eFuse command.

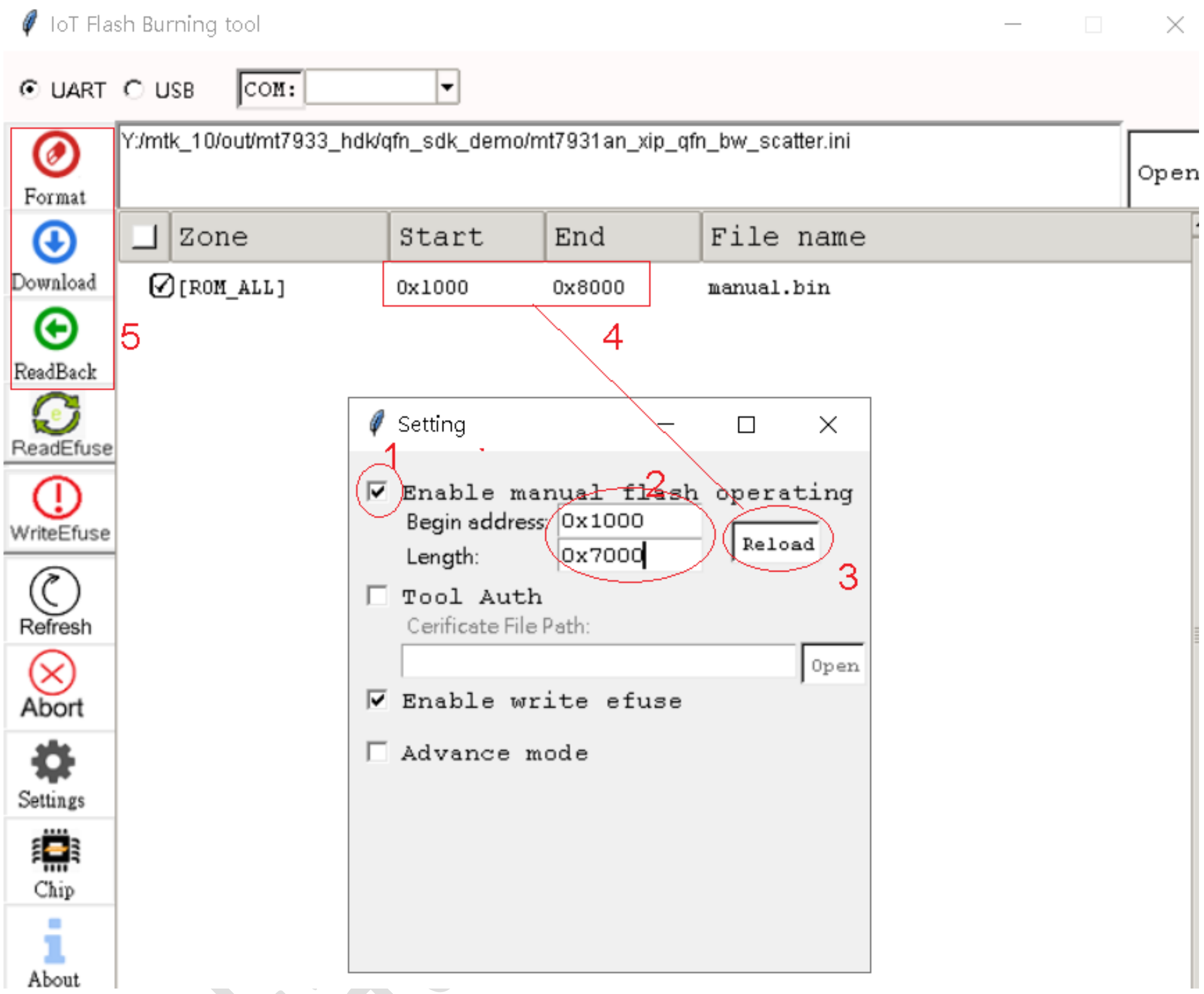
Example:

fbtool.exe.exe -o rdefuse_c   -e Y:/mtk_11/out/mt7933_hdk/bga_sdk_demo/my_ef_tbl.txt

The command reads back eFuse data to another file as: my_ef_tbl.txt(readback) and compares the file with my_ef_tbl.txt,

After the files are compared, if there is any error, you see an error message.

8 Manual Flash Operation



Y:/mtk_10/out/mt7933_hdk/qfn_sdk_demo/mt7931an_xip_qfn_bw_scatter.ini

	Zone	Start	End	File name
✓	[ROM_ALL]	0x1000	0x8000	manual.bin

[✓] Enable manual flash operating
Begin address: 0x1000
Length: 0x7000
[ ] Tool Auth
Cerificate File Path:
[✓] Enable write efuse
[ ] Advance mode

Figure 27: Manual flash operating

General operating steps:
(Step 1) Select the checkbox.
(Step 2) Enter the Begin address and Length
(Step 3) Click the Reload button.
(Step 4) Data will be updated to the scatter file.
(Step 5) Choose one command to run.
(Step 6) Repeat Steps 2 to 5.
If you run Read back command, it will outputs manual_readback.bin file to working folder.
If you run Download command, tool will burn manual.bin to flash.

Exhibit 1 Terms and Conditions

Your access to and use of this document and the information contained herein (collectively this "Document") is subject to your (including the corporation or other legal entity you represent, collectively "You") acceptance of the terms and conditions set forth below ("T&C"). By using, accessing or downloading this Document, You are accepting the T&C and agree to be bound by the T&C. If You don't agree to the T&C, You may not use this Document and shall immediately destroy any copy thereof.

This Document contains information that is confidential and proprietary to MediaTek Inc. and/or its affiliates (collectively "MediaTek") or its licensors and is provided solely for Your internal use with MediaTek's chipset(s) described in this Document and shall not be used for any other purposes (including but not limited to identifying or providing evidence to support any potential patent infringement claim against MediaTek or any of MediaTek's suppliers and/or direct or indirect customers). Unauthorized use or disclosure of the information contained herein is prohibited. You agree to indemnify MediaTek for any loss or damages suffered by MediaTek for Your unauthorized use or disclosure of this Document, in whole or in part.

MediaTek and its licensors retain titles and all ownership rights in and to this Document and no license (express or implied, by estoppels or otherwise) to any intellectual propriety rights is granted hereunder. This Document is subject to change without further notification. MediaTek does not assume any responsibility arising out of or in connection with any use of, or reliance on, this Document, and specifically disclaims any and all liability, including, without limitation, consequential or incidental damages.

THIS DOCUMENT AND ANY OTHER MATERIALS OR TECHNICAL SUPPORT PROVIDED BY MEDIATEK IN CONNECTION WITH THIS DOCUMENT, IF ANY, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. MEDIATEK SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, COMPLETENESS OR ACCURACY AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MEDIATEK SHALL NOT BE RESPONSIBLE FOR ANY MEDIATEK DELIVERABLES MADE TO MEET YOUR SPECIFICATIONS OR TO CONFORM TO A PARTICULAR STANDARD OR OPEN FORUM.

Without limiting the generality of the foregoing, MediaTek makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does MediaTek assume any liability arising out of the application or use of any product, circuit or software. You agree that You are solely responsible for the designing, validating and testing Your product incorporating MediaTek's product and ensure such product meets applicable standards and any safety, security or other requirements.

The above T&C and all acts in connection with the T&C or this Document shall be governed, construed and interpreted in accordance with the laws of Taiwan, without giving effect to the principles of conflicts of law.