MT793X IoT SDK for Bootloader

User Guide

Version: 1.0

Release date: 2021-07-01

Version History

Version	Date	Description
0.1	2021-04-06	Initial draft

Table of Contents

List of Figures

1 Terminology

Term or abbreviation	Meaning
BootROM	A piece of immutable software in ASIC.
BL	Bootloader is a piece of mutable software stored at the beginning of flash.
ECC/EC	Elliptic Curve Cryptography. In the MT793X, the curve used is P-256.
EFUSE	One Time Programmable Memory.
MT793X	The MT7931 and MT7933 family of chipset.
TLV	Tag-length-value is an encoding scheme for storing information used by imgtool.py.
VTOR	Vector table offset register.

2 Overview

In the MT793X, the bootloader is the only piece of software to be executed by BootROM in normal-boot. MediaTek provides the source code of a reference bootloader. The bootloader, abbreviated as BL, is signed during the image creation and stored at the beginning of external serial flash.

During a normal-boot, as the root-of-trust, BootROM verifies BL. If the verification has succeeded, BootROM passes the execution flow to BL. BL can then verify the firmware of the next stage and pass the execution flow to the firmware of the next stage.
As part of chain-of-trust. BL is signed with ECC algorithm and verified by BootROM accordingly. Similar to the way BootROM examines BL, BL can verify integrity and authenticity of the subsequent firmware to fulfill security requirements. This process forms the chain-of-trust.

In addition to normal-boot mode, BL supports resume-boot mode. A resume-boot is a power up procedure after the CPU core powers down in low power mode. It is different from a normal-boot where some hardware has been initialized in the previous normal-boot. In the MT793X resume-boot, the reference BL bypasses all security checks and passes the execution flow to the firmware of the next stage.

3 Boot Modes

The MT793X supports two boot modes:

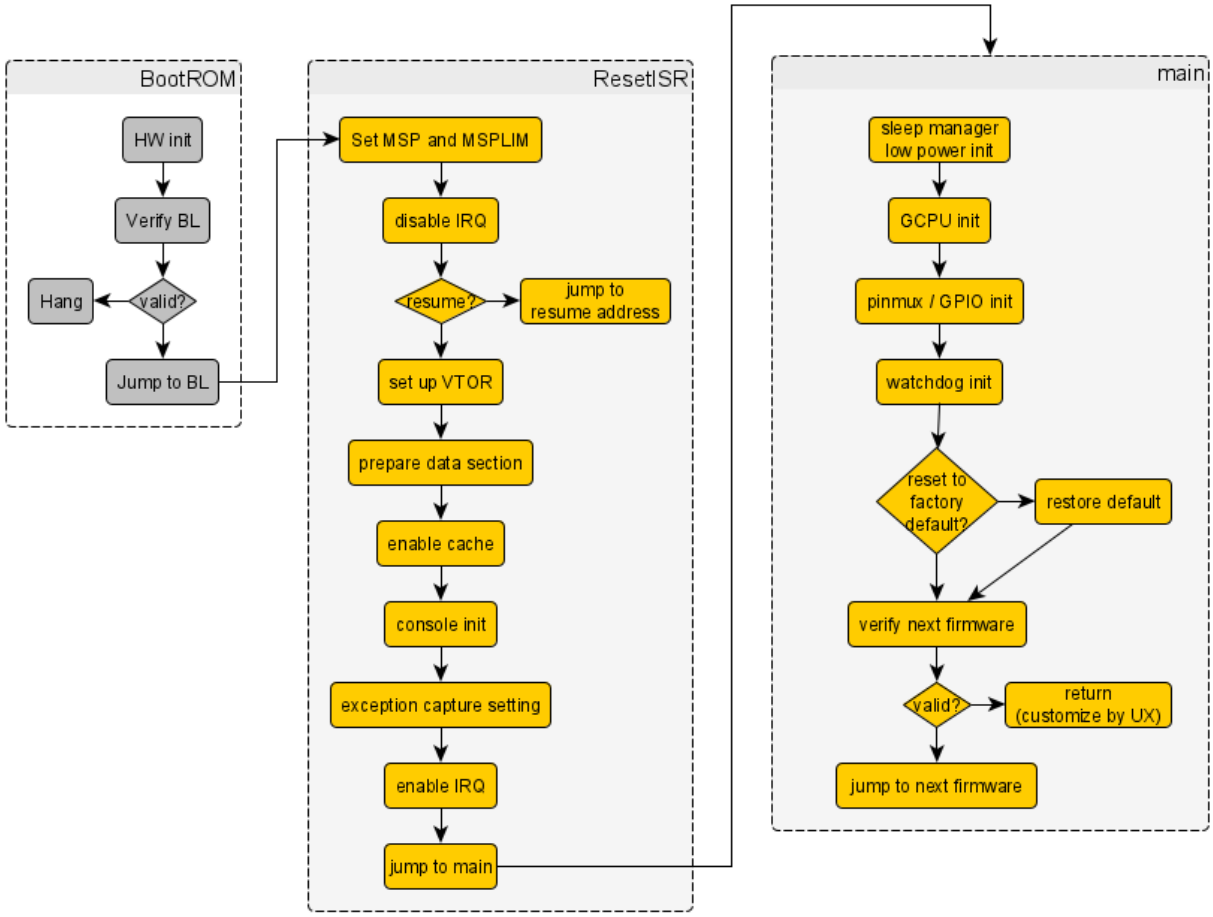
- Download mode
  - BootROM only
- Execution mode
  - Normal-boot
    - Normal-boot is executed when the MT793X powers on from the power off status or when the firmware triggers a system reboot.
  - Resume-boot
    - Resume-boot is executed when the MT793X powers on from some low power status with the Cortex-M33 processor off. Other peripherals may also be turned off, depending on the user scenario. For more information about low power status, please refer to sleep manager related documents.

The next chapter shows the difference between normal-boot and resume-boot in BL.

4 Bootloader Execution Flow

The complete execution of BL is shown in Figure 1. In this figure, BROM is added to denote the grand boot flow. The function ResetISR and main constitutes the major path of BL execution.

Figure 1 MediaTek Reference Bootloader Flow



4.1 Normal-Boot

In Figure 1, excluding the “jump to resume address” in ResetISR, the reset of ResetISR and main is the execution flow of normal-boot.

The first step is to set the stack pointers to allow the use of stack of function invocation and local variables. Then, disable IRQ because it is about setting VTOR.

4.2 Resume-Boot

During a resume-boot, the time to recover the system to running state is important and should be kept as short as possible. BL skips most of the procedures and at the 3rd step in ResetISR, BL jumps to the resume address, which is an address stored by the sleep manager before the MT793X enters low power mode.

5 Sign and Pack

When building the MediaTek SDK, the MT7933 BL is signed automatically with a default ECC private key.

5.1 Sign Tool - imgtool.py

The signing tool is called imgtool.py in MediaTek SDK (tools/mcuboot/scripts/imgtool.py).
It is based on the same tool, imgtool.py, from the open source project MCUBOOT (https://mcu-tools.github.io/mcuboot/) with MediaTek extensions.

5.2 Image Sign

Below is a simplified command line signing the BL. If you use another BL, you need to apply the same signing to it so that BootROM can verify it.

```
imgtool.py sign --pad-header --header-size 128 \
                --load-addr 0x18000080 -k $(SDK_PATH)/$(APP_PATH)/mtk-dev.pem -S 65536 \
                --align 4 -v 0.0.1 --pubkey \
                --no-bootrom-log \
                mt7933-bootloader.bin mt7933-bootloader.sgn
```

For complete command line, please search for 'IMGTOOL' in project/mt7933_hdk/apps/bootloader/GCC/Makefile.

5.2.1 ECC Key File

The previous section shows how to generate a signed BL binary (.sgn) file with the default ECC key. The default ECC key file is located at project/mt7933_hdk/apps/bootloader/mtk-dev.pem. Its content is shown below.

```
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgmyTNAJ4KL3dx15E2
7GdJ7+FeVnY29dXJVsA0dwHNngKhRANCAAQ5SQ5u+SgoOO8qaN2A2171kZYGL+cr
91/SzV7ln5lDWJJEYvn41lADOQb5mEYCzfBt8/q88ObveGoiVqVRTo8s
-----END PRIVATE KEY-----
```

The human readable form is shown below using OpenSSL to dump the content. You can see that this is an elliptic curve P-256 key pair.

```
$ openssl ec -in mtk-dev.pem -text -noout
read EC key
Private-Key: (256 bit)
priv:
    00:9b:24:cd:00:9e:0a:2f:77:71:d7:91:36:ec:67:
    49:ef:e1:5e:56:76:36:f5:d5:c9:56:c0:34:77:01:
    cd:9e:02
pub:
    04:39:49:0e:6e:f9:28:28:38:ef:2a:68:dd:80:db:
    5e:f5:91:96:06:2f:e7:2b:f7:5f:d2:cd:5e:e5:9f:
    99:43:58:92:44:62:f9:f8:d6:50:03:39:06:f9:98:
    46:02:cd:f0:6d:f3:fa:bc:f0:e6:ef:78:6a:22:56:
    a5:51:4e:8f:2c
ASN1 OID: prime256v1
```

The command to generate a customized key pair with OpenSSL is shown below. After a key is generated, update it to the SDK to generate BL with new EC P-256 key.

```
$ openssl ecparam -genkey -name prime256v1 -out cust-mp.pem
$ openssl ec -in cust-mp.pem -text -noout
read EC key
Private-Key: (256 bit)
priv:
    00:84:88:e4:dd:5b:d1:fb:95:9f:5a:1d:79:91:23:
    86:ec:9c:d7:15:5d:cd:f3:da:0b:9f:35:32:ba:78:
    e5:58:54
pub:
    04:0b:b2:8f:f9:c0:bd:7f:c3:21:ad:d9:50:15:ee:
    68:fc:50:0b:ca:5d:83:12:7a:6a:2d:0c:10:12:63:
    60:bc:73:61:22:e8:43:27:59:22:fd:0e:94:69:a6:
    31:c7:ef:f6:6b:29:79:03:e7:37:0e:45:b0:72:49:
    36:b4:e0:b6:d6
ASN1 OID: prime256v1
```

0.1.1. Public Key Hash

To generate the hash of public key, which shall be blown into eFuse, use the following command:

```
$ openssl pkey -inform PEM -in mtk-dev.pem -pubout -outform DER | sha256sum
0cd1705b518f10b74552626559d44ce3132552299dee6f81d3d541fcf544416e  -
```

The hex string above is 64 characters in ASCII format, equivalent to 32 bytes and 256 bits in actual data.

5.3 Anti-Rollback

Adding “--ar-ver <1~64>” to imgtool.py command adds the anti-rollback version TLV to the output BL image.

```
imgtool.py sign --pad-header --header-size 128 \
                --load-addr 0x18000080 -k $(SDK_PATH)/$(APP_PATH)/mtk-dev.pem -S 65536 \
                --align 4 -v 0.0.1 --ar-ver 1 --pubkey \
                --no-bootrom-log \
                mt7933-bootloader.bin mt7933-bootloader.sgn
```

5.4 Enable BootROM Log Messages

When booting from BROM to BL, BROM prints messages for debugging purpose. The logs are displayed whether the booting can or cannot proceed, but can be controlled via an option (TLV).

To expedite the boot up process, BROM log message can be turned off by adding an option (TLV) to BL. To add this option, the parameter "--no-bootrom-log" is applied by default in BL Makefile.

To turn on BROM log, remove this command parameter in Makefile.

To view the BROM log, set the console baud rate to 115200.

```
imgtool.py sign --pad-header --header-size 128 \
                --load-addr 0x18000080 -k $(SDK_PATH)/$(APP_PATH)/mtk-dev.pem -S 65536 \
                --align 4 -v 0.0.1 --ar-ver 1 --pubkey \
                --no-bootrom-log \
                mt7933-bootloader.bin mt7933-bootloader.sgn
```

6 Chain-of-Trust / Verified Boot

NOTE: Only the verification of BL by BROM has been implemented at the moment.

The term chain-of-trust refers to the trust relationship between firmware components, based on cryptographic algorithms.

Verified Boot works based on the chain-of-trust concept:

- A root of trust exists in eFuse.
  - In MT793x's case, the HASH of public key and the immutable BROM forms the foundation of subsequent trust relationship.
- BROM verifies BL before jumping into (executing) BL.
- BL verifies the rest of firmware components (RTOS firmware, TF-M firmware, Wi-Fi firmware, BT firmware, DSP firmware) as shown in Figure 2 and then starts to execute the firmware of the next stage as show in Figure 3.

Figure 2 Verify Sequence

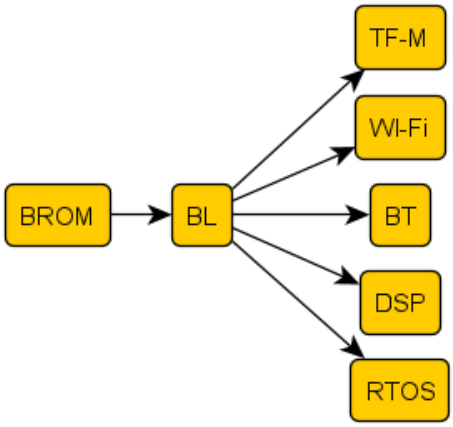
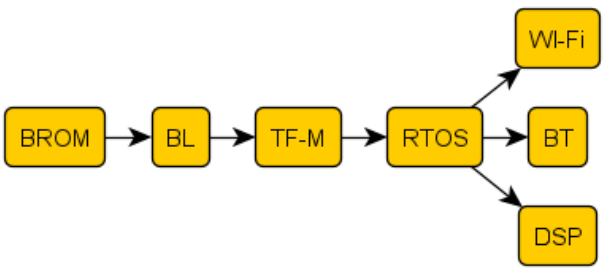


Figure 3 Execute Sequence



7 Bootloader Source Tree

```
project/mt7933_hdk/apps/bootloader/
|-- GCC
|    |-- bootloader.bin
|    |-- bootloader-ram.ld
|    |-- bootloader-xip.ld
|    |-- feature.mk
|    |-- hal_feature.mk
|    |-- Makefile
|    |-- mt7933_bootloader-ram.cmm
|    |-- mt7933_bootloader-xip.cmm
|    |-- security.mk
|    `-- syscalls.c
|-- inc
|    |-- bga
|    |    `-- ept_gpio_drv.h
|    |-- bl_cache_ops.h
|    |-- bl_fota.h
|    |-- bl_image.h
|    |-- bl_mperf.h
|    |-- cli_cmds.h
|    |-- crc16.h
|    |-- ept_eint_drv.h
|    |-- hal_feature_config.h
|    |-- hw_uart.h
|    |-- memory_map.h
|    |-- qfn
|    |    `-- ept_gpio_drv.h
|    |-- sys_init.h
|    `-- xmodem.h
|-- mtk-dev.pem
|-- readme.txt
`-- src
     |-- bl_fota.c
     |-- bl_image.c
     |-- bl_mperf.c
     |-- cli_cmds.c
     |-- cli_def.c
     |-- crc16.c
     |-- ept_eint_var.c
     |-- ept_gpio_var.c
     |-- exception_handler.c
     |-- exception.s
     |-- hw_uart.c
     |-- main.c
     |-- sys_init.c
     |-- system_mt7933.c
     `-- xmodem.c
```

Some of the most important files in the source tree of BL include:

main.c – most of the execution flows are in this file, including ResetISR() and main() functions.

bootloader-ram.ld contains the allocated memory region for in-RAM execution scenario.

bootloader-xip.ld contains the allocated memory region for XIP execution scenario.

feature.mk contains options that can be adjusted to enable/disable features. For the availability of supported options, check the file coming along with MediaTek SDK.

mtk-dev.pem contains the ECC keys. Observe the key with this command:

```
openssl ec -in mtk-dev.pem -text -noout
read EC key
Private-Key: (256 bit)
priv:
    00:9b:24:cd:00:9e:0a:2f:77:71:d7:91:36:ec:67:
    49:ef:e1:5e:56:76:36:f5:d5:c9:56:c0:34:77:01:
    cd:9e:02
pub:
    04:39:49:0e:6e:f9:28:28:38:ef:2a:68:dd:80:db:
    5e:f5:91:96:06:2f:e7:2b:f7:5f:d2:cd:5e:e5:9f:
    99:43:58:92:44:62:f9:f8:d6:50:03:39:06:f9:98:
    46:02:cd:f0:6d:f3:fa:bc:f0:e6:ef:78:6a:22:56:
    a5:51:4e:8f:2c
ASN1 OID: prime256v1
```

And create your own key with:

```
$ openssl version
OpenSSL 1.0.1f 6 Jan 2014
$ openssl ecparam -genkey -name prime256v1 -out k.pem
$ openssl ec -in k.pem -text -noout
read EC key
Private-Key: (256 bit)
priv:
    00:e1:de:9d:79:6d:4c:a4:51:ea:01:04:62:88:24:
    15:72:af:3c:76:b0:ca:c2:1b:f0:35:ad:af:02:78:
    34:91:f3
pub:
    04:52:29:a5:b4:85:b7:2a:58:d3:e5:60:58:7b:e5:
    c6:6f:81:eb:e5:f8:d9:f6:13:5e:61:97:f9:56:3d:
    ed:26:71:21:a4:e7:22:6a:44:c3:07:7b:09:65:06:
    76:0f:15:9e:a2:e2:e9:04:d7:30:c5:c6:2b:36:76:
    a5:0e:f4:f9:45
ASN1 OID: prime256v1
```

8 References

- MT793X IoT SDK for Secure Boot User Guide.
- MT793X IoT SDK for Low Power User Guide.
- MCUBOOT (https://mcu-tools.github.io/mcuboot/).
- OpenSSL (https://www.openssl.org/).

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.