



Secure boot Hands-on

Fill efuse name and Read secure boot check

- CM33 log will notify user that the secure boot verify RTOS fail if not enable secure boot enable bit **EFUSE_SBC_EN** and Fill hash of key in efuse.

```
loader init
0
Your choose c
secure boot: checking
invalid key 0x3c3c
secure boot: rtos verify fail, status 7 addr 0x18044000 size 0x208000
secure boot: permissive, ignore
jump pc 0x180a0c4d, sp 0x111000
hal_psrainit
```

[EFUSE]

enable=n

start_addr=

partition_size=

file_name=test.bin

readback=n

- Fill in Efuse file name in scatter file, which path is as below:

{Top}/out/mt7933_hdk/XXX_scatter.ini

- Read secure boot check

- Set Control data
- Index: 0x01000100

FlashBurningTool_v2.77

```
KG_TYPE/PSRAM_TYPE: QFN_8M(APM)
flash query data: 0x60020000 0x20000
flash init:
```

```
flash query data: 0x60020000 0x20000
A valid
```

```
est_DA=>start to set bypass brom
efuse protect
```

```
efuse GRP: 1
```

```
full_page_enable: 0
addr_offset: 0x100
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 0x0
```

```
receive:byte: 0xb : 0x0
```

```
receive:byte: 0xc : 0x0
```

```
receive:byte: 0xd : 0x0
```

```
receive:byte: 0xe : 0x0
```

```
receive:byte: 0xf : 0x0
```

```
receive:byte: 0x0 : 0x0
```

```
receive:byte: 0x1 : 0x0
```

```
receive:byte: 0x2 : 0x0
```

```
receive:byte: 0x3 : 0x0
```

```
receive:byte: 0x4 : 0x0
```

```
receive:byte: 0x5 : 0x0
```

```
receive:byte: 0x6 : 0x0
```

```
receive:byte: 0x7 : 0x0
```

```
receive:byte: 0x8 : 0x0
```

```
receive:byte: 0x9 : 0x0
```

```
receive:byte: 0xa : 
```

Enable secure boot check and Generate key pair

4. Enable secure boot check

- Set EFUSE_SBC_EN 1 as below
- Control data : 0x01000100
- Address: 0x00
- Index: 0x01

5. Use OpenSSL to generate an ECC key pair

- The command generate a file, called *my.pem*, with an ECC key pair (private and public keys) in it.
- `openssl ecparam -genkey -name secp256r1 -out my.pem`

The screenshot displays the FlashBurningTool v2.77 interface. The top window shows the file 'GRP1.bin' with the first byte of the address (0) highlighted. The main window shows the command 'WRITE EFUSE CMD:' and the resulting data '0x01'. The right panel shows the 'Write Efuse' button highlighted.

1. Set first byte [0: 0x01]

2. Click Write Efuse

3. Shows Write efuse data

Replace pem file and Generate Hash

6. Replace the mtk-dev.pem and build project

- {Top}/project/mt7933_hdk/apps/Bootloader/mtk-dev.pem

7. Generate the Hash of the public Key in mtk-dev.pem

- openssl pkey -inform PEM -in my.pem -pubout -outform DER | sha256sum

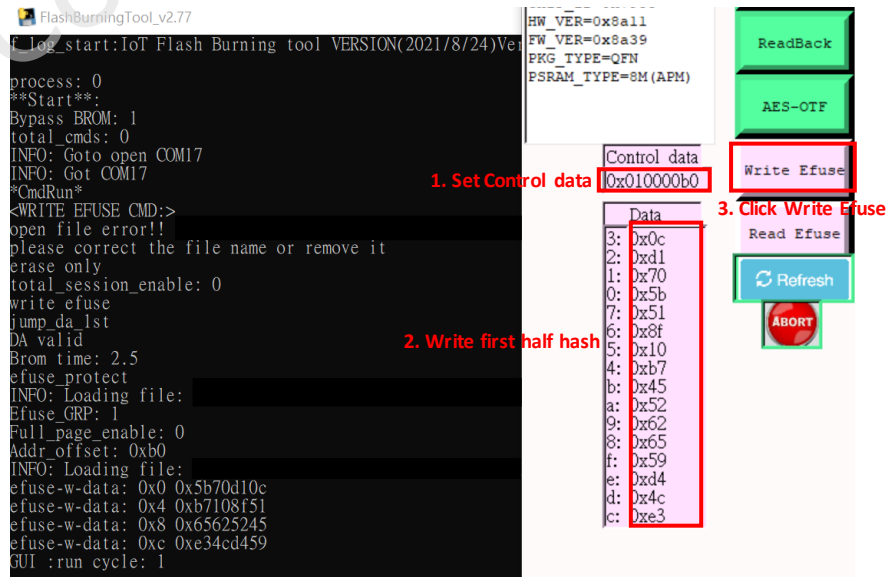
8. Blow efuse in the MT793x with the Hash

- Set control data: 0x010000**b0**
- Index: first half of hash
- For example:

0x0c	0x45
0xd1	0x52
0x70	0x62
0x5b	0x65
0x51	0x59
0x8f	0xd4
0x10	0x4c
0xb7	0xe3

Use the following OpenSSL commands to calculate the SHA-256 hash of the public key derived from the private key file my.pem.

```
$ openssl pkey -inform PEM -in my.pem -pubout -outform DER | sha256sum
0cd1705b518f10b74552626559d44ce31132552299dee6f81d3d541cf544416e -
```



Blow eFuse in the MT793X with the Hash

9. Blow efuse in the MT793x with the Hash

- Set control data: 0x010000c0
- Index: the rest of hash

0x13	0xd3
0x25	0xd5
0x52	0x41
0x29	0xfc
0x9d	0xf5
0xee	0x44
0x6f	0x41
0x81	0x6e

10. Check bootloader verify RTOS procedure.

```
loader init
0
Your choose c
secure boot: checking
jump pc 0x180a0c4d, sp 0x111000
hal_psram_init
```

Use the following OpenSSL commands to calculate the SHA-256 hash of the public key derived from the private key file my.pem.

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

FlashBurningTool v2.77

process: 0
 Start:
 Bypass_BROM: 1
 total_cmds: 0
 INFO: Goto open COM17
 INFO: Got COM17
 CmdRun
 <WRITE EFUSE CMD:>
 open file error!! Z:/SDK_1.1.2_promis_official/SDK_1.1.2
 please correct the file name or remove it
 erase only
 total_session_enable: 0
 write efuse
 jump_da_lst
 DA valid
 Brom time: 2.49
 efuse_protect
 INFO: Loading file: Z:/SDK_1.1.2_promis_official/SDK_1.1.2
 Efuse_GRP: 1
 Full_page_enable: 0
 Addr_offset: 0xc0
 INFO: Loading file: Z:/SDK_1.1.2_promis_official/SDK_1.1.2
 efuse-w-data: 0x0 0x29522513
 efuse-w-data: 0x4 0x816fee9d
 efuse-w-data: 0x8 0xfc41d5d3
 efuse-w-data: 0xc 0x6e4144f5
 GUI :run cycle: 1
 download_path_uart: y
 uart_com_port: COM17
 f log start: IoT Flash Burning tool VERSION(2021/8/24)Ver

HW_VER=0x8a11
 FW_VER=0x8a39
 PKG_TYPE=QFN
 PSRAM_TYPE=8M (APM)

Control data: 0x010000c0

Data:

3:	0x13
2:	0x25
1:	0x52
0:	0x29
7:	0x9d
6:	0xee
5:	0x6f
4:	0x81
b:	0xd3
a:	0xd5
9:	0x41
8:	0xfc
f:	0xf5
e:	0x44
d:	0x41
c:	0x6e

1. Set Control data

2. Write first half hash

3. Click Write Efuse

Buttons: ReadBack, AES-OTF, Write Efuse, Read Efuse, Refresh, ABORT