MT793X IoT SDK for TF-M User Guide

Version: 1.0

Release date: 2022-08-08

Use of this document and any information contained therein is subject to the terms and conditions set forth in Exhibit 1. This document is subject to change without notice.

Version History

Version	Date	Description
0.1	2021-09-10	Initial draft
1.0	2022-08-08	Modify section 3 platform isolation part

Table of Contents

List of Figures

List of Tables

1 Overview

As embedded closed systems become more and more complicated, platform security becomes more and more important. The MT793X adopts the open source project, TF-M (Trusted Firmware-M), as the solution to ensure platform security. This user guide does not introduce TF-M in detail. For background information, refer to the TF-M official website https://www.trustedfirmware.org/projects/tf-m/. This document only describes the additional porting functions that aim to enhance platform security of the MT793X, and these functions include boot flow, platform isolation, and deep sleep.

2 Boot Flow

Boot flow is the root of trust in platform security. The MT793X applies the secure boot to protect the system against malicious code by ensuring only the authenticated software runs on the device. This boot flow is a little different after the TF-M firmware is integrated.

2.1 Boot Flow of Non-TF-M Projects

In the secure boot flow, the system starts from BROM and then jumps to the bootloader if BROM verifies the integration and validation of the bootloader is successful. The bootloader then verifies RTOS load the way similar to the BROM verification procedure, and the system jumps to RTOS to finish the whole system boot process as Figure 2-1 shows. The Cortex-M33 (CM33) processor is always in the secure state during the entire boot process, even after the boot flow is over and the system starts to execute in RTOS.

Figure 2-1. Non-TF-M project boot flow

2.2 Boot Flow of TF-M Projects

In the TF-M project, the TF-M initialization flow is added to the whole boot process. In the TF-M initialization flow, there are secure and non-secure environments. To ensure the bus access is under control, TF-M builds up the mechanism of access permission control of the whole system. So, the TF-M initialization flow should be done before the system jumps to RTOS. In the secure boot flow, the previous boot stage needs to verify the next one; therefore, TF-M is verified by the bootloader. In contrast with the non-TF-M project, by enabling platform isolation, the secure state of CM33 changes to non-secure after the system jumps to RTOS.

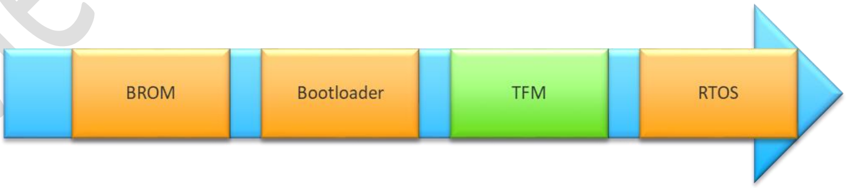

Figure 2-2. TF-M project boot flow

3 Platform Isolation Setup

To prevent malicious software from accessing confidential data, TF-M establishes an isolated execution environment, and the platform is divided into two environments, SPE (Secure Processing Environment) and NSPE (Non-Secure Processing Environment). The secure firmware runs in SPE and sensitive data can also be stored in SPE. Non-secure tasks in NSPE cannot access the data and services in SPE directly; they are only allowed to request the data and services in SPE by limited veneers. This chapter describes how to set up platform isolation for the MT793X TF-M projects.

3.1 Hardware Isolation Module

The MT793X is equipped with DAPC and ASIC_MPU, and cooperates with SAU to build the whole system, in which access to all modules is under permission control.

Table 3-1. MT793X hardware isolation modules

Hardware module	Descriptions
SAU	One of the functions of ARM TZ; it partitions memory regions into secure, non-secure, and NSC (Non-Secure Callable) regions to CM33
DAPC	Bus protection module, peripherals APC to bus master domain
ASIC_MPU	Bus protection module, memory regions APC to bus master domain

3.2 Platform Isolation Build up Flow

As Figure 3-1 shows, each step in this flow is dependent on the configurations of the hardware isolation module, and all of the steps construct the whole environment isolation.

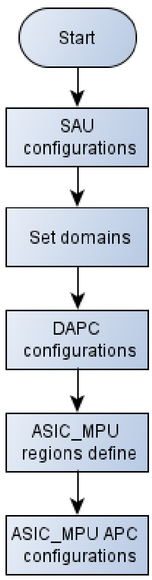

Figure 3-1. Platform isolation build up flow

3.3 SAU Configurations

The MT793X contains 8 configurable SAU regions; each configuration includes the start and limit addresses of the region, and the NSC attribute. Please note that the region set by SAU is "non-secure" and NSC; regions other than these 8 configurations are all secure regions.

```
struct sau_cfg_t {
    uint32_t RNR;
    uint32_t RBAR;
    uint32_t RLAR;
};
```

Table 3-2. MT793X hardware isolation modules

Field	Descriptions
RNR	Number of the configuration region
RBAR	Start address of the configuration region
RLAR	Limit address of the configuration region

3.4 Bus Master Domain Configurations

In an SoC, not all the IPs or modules support TrustZone. Each of them could be partially secure-aware or non-secure-aware. To integrate all the designs into the system and provide security and data protection without modifying the original IP design, a configurable bus protection module is implemented. The bus protection mechanism is used to verify the access permission of the bus master to the bus slave. The bus master requests a read or write transaction to the slave with two sideband signals - domain ID and secure state. Under the bus protection mechanism, the permission check target changes from the bus master to the domain ID. All masters' domains are 0 by default. TF-M groups all masters into 8 domains; the slave APC affects all masters with the same domain ID. Table 3-3 shows the configurations of all bus masters on the MT793X and the corresponding domains.

Table 3-3. MT793X bus masters domain

Bus master	Domain
CPUM	0
SPI TEST	0
CM33	1
SDIO SLAVE	2
SDIO MASTER	2
SPIM0	2
SPIM1	2
SPIS	2
USB HOST	2
USB DEV	2
CONNAC_CONN2AP	3
CONNAC_WFDMA	4
AP DMA	6

Bus master	Domain
CQ DMA	6
GCPU	5
DSP	7
AFE	7

3.4.1 DAPC

Under the bus protection mechanism, a slave may be a peripheral module or a memory region. DAPC checks whether the master has valid permission to the peripheral. The details of DAPC can refer the document MT7933AT(BT)_DAPC_datasheet.docx.

3.4.2 ASIC_MPU

ASIC_MPU is another bus protection module on the MT793X. For ASIC_MPU, the slave object is a memory region. Similarly, the ASIC_MPU also checks whether the master has valid access permission to the corresponding memory region. The details of ASIC_MPU can refer the document MT7933AT(BT)_ASIC_MPU_datasheet.docx.

3.5 Platform Isolation Configurations

As the descriptions in Section 2.2, the CPU state changes to non-secure after the system jumps to FreeRTOS. To ensure the whole platform is in a secure and controllable environment, platform isolation configurations need to be done before the CPU changes to non-secure state. Table 3-4 shows the corresponding configurations file paths and functions, user can change the configurations in these files. MTK recommends the settings of SAU and ASIC_MPU align the settings of the regions defined in the linker script of FreeRTOS.

Table 3-4. Platform isolation configurations file

File	Function	Descriptions	MT793X configuration
middleware/third_party/tfm/trusted-firmware-m/platform/ext/target/mt7933/mt7933_hdk/target_cfg.c	sau_and_idau_cfg	Set the non-secure and NSC region by the SAU register.	The MT793X SDK default sets the regions based on the sections defined in the linker script of RTOS.
middleware/third_party/tfm/trusted-firmware-m/platform/ext/target/mt7933/mt7933_hdk/drivers/platform_isolation/Domain_config.h	master_domain[BUS_MASTER_MAX]	set the domain to each bus master	Refer to the table3-3
middleware/third_party/tfm/trusted-firmware-m/platform/ext/target/mt7933/mt7933_hdk/drivers/platform_isolation/DAPC_config.h	INFRA_Devices[], AUD_Devices[]	Set APC of each domain to peripherals	Refer to the configuration header file Domain_config.h

File	Function	Descriptions	MT793X configuration
middleware/third_party/tfm/trusted-firmware-m/platform/ext/target/mt7933/mt7933_hdk/drivers/platform_isolation/ASIC_MPU_config.h	ASIC_MPU_Devices[][]	1. 1. Set start address of the regions. Length of the region depends on the start address of the next region, so note that the start address set to the table must guarantee the sequence is ascending 2. Set APC of the regions to each domain	1. Refer to the configuration header file ASIC_MPU_config.h 2. Note the start address marked to the tag TMP_DATA_SECTION_START_ADDR will be replaced with the start address of the data section of RTOS during the process of platform isolation, so keep remaining the tag in the configuration table to ensure settings are correct.
middleware/third_party/tfm/trusted-firmware-m/platform/ext/target/mt7933/mt7933_hdk/drivers/platform_isolation/pltfm_iso.h	enum bus_master	enumeration definition of the bus master	
driver/chip/mt7933/inc/hal_devapc.h		1. Bus type 2. Domain value definitions 3. APC value definitions of the DAPC	
driver/chip/mt7933/inc/hal_asic_mpu.h		1. Memory type definitions 2. APC value definitions of the ASIC_MPU	

3.6 Bus Protection Violations

After TF-M enables platform isolation, DAPC and ASIC_MPU check the validation of every bus transaction requested from the bus master to the slave. DAPC and ASIC_MPU identify the sideband signal issued from the master. If they find the domain or secure status does not match the APC of the slave, a violation message shows on the console.

3.6.1 DAPC Violation

As 錯誤! 找不到參照來源。 shows, once the DAPC detects the invalid access request from the bus master, a violation message appears. The message includes violation address, domain ID, slave index, violation type (read or write), return value, etc.

```
$ rr 0x30300000
rr 0x30300000
0x30300000[DEVAPC] INFRA vio_sta found: 70, shift_bit: 7
[DEVAPC] INFRA Violation (R) - Vio Addr: 0x30300000, High: 0x0, Bus ID: 0x0, Domain ID: 0x1
: 0x0
```

Figure 3-2. DAPC violation message

3.6.2 ASIC_MPU Violation

Similarly, as 錯誤! 找不到參照來源。 shows, once the ASIC_MPU detects the invalid access request from the bus master, a violation message appears. The message includes the violation address, domain ID, region number, violation type (read or write), access types, return value, etc.

```
$ rr 0x90000000
rr 0x90000000
0x90000[ASIC_MPU] IRQ_STA: 0x2
[ASIC_MPU] FLASH MPU Violation!!
[ASIC_MPU] Dumping Vio Info...
[ASIC_MPU] (R Violation) Permission: 0x5, Domain: 0x1, Region: 0x1, Addr: 0x90000000
[ASIC_MPU] Access type: Privileged, Non-secure, Data
[ASIC_MPU] ABN ID: 0x0
000: 0x0
```

Figure 3-3. ASIC_MPU violation message

4 Deep Sleep with TF-M

To ensure system security, the warm boot flow must be the same as the cold boot flow. The boot procedure also begins with the BROM and ends with the RTOS as Figure 2-2 shows. The platform isolation configurations must be restored to the settings set before deep sleep after platform wakeup.

4.1 Deep Sleep Flow

The MT793X deep sleep and wakeup flow can be roughly divided into two parts - the backup and restore flows of the normal modules and of the system modules.

4.1.1 Deep Sleep on Non-TF-M Projects

As Figure 4-1 shows, elements in the pink box are normal modules, elements in the blue box are system modules, and the dotted arrow in yellow is the sequence of the whole system sleep and wakeup flow. The procedure starts from the normal modules backup on the upper left-hand corner of Figure 4-1 and ends with normal modules restore on the lower left-hand corner of Figure 4-1. Between system modules backup and restore, the platform enters deep sleep status by the WFI command. The platform wakes up from WFI by an interrupt and starts with BROM, then the bootloader, and finally goes back to the restore flow of the system modules.

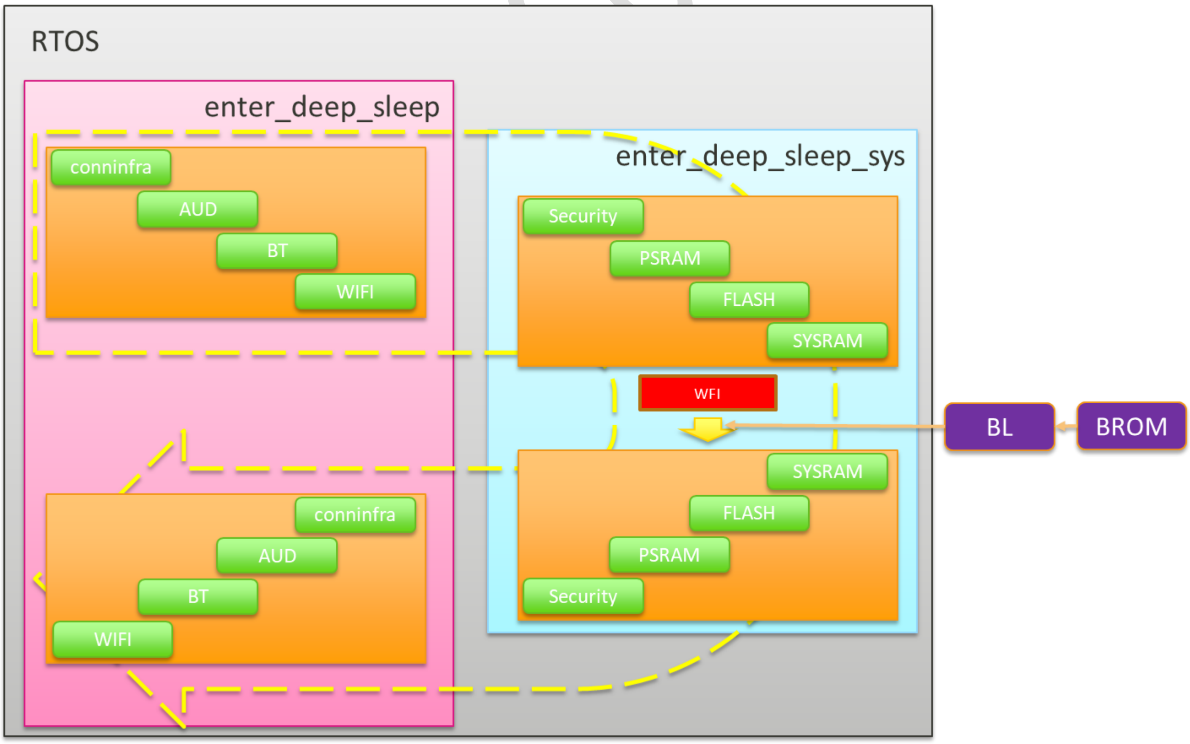

Figure 4-1. Sleep and wakeup on non-TF-M projects

4.1.2 Deep Sleep on TF-M Projects

As Figure 4-2 shows, the only difference in the deep sleep flow between the TF-M project and non-TF-M project is that the backup and restore of the system modules execute in TF-M. This figure also shows the platform wakeup procedure begins with the BROM, then the bootloader, TF-M, and finally ends with RTOS. This process matches the cold boot procedure.

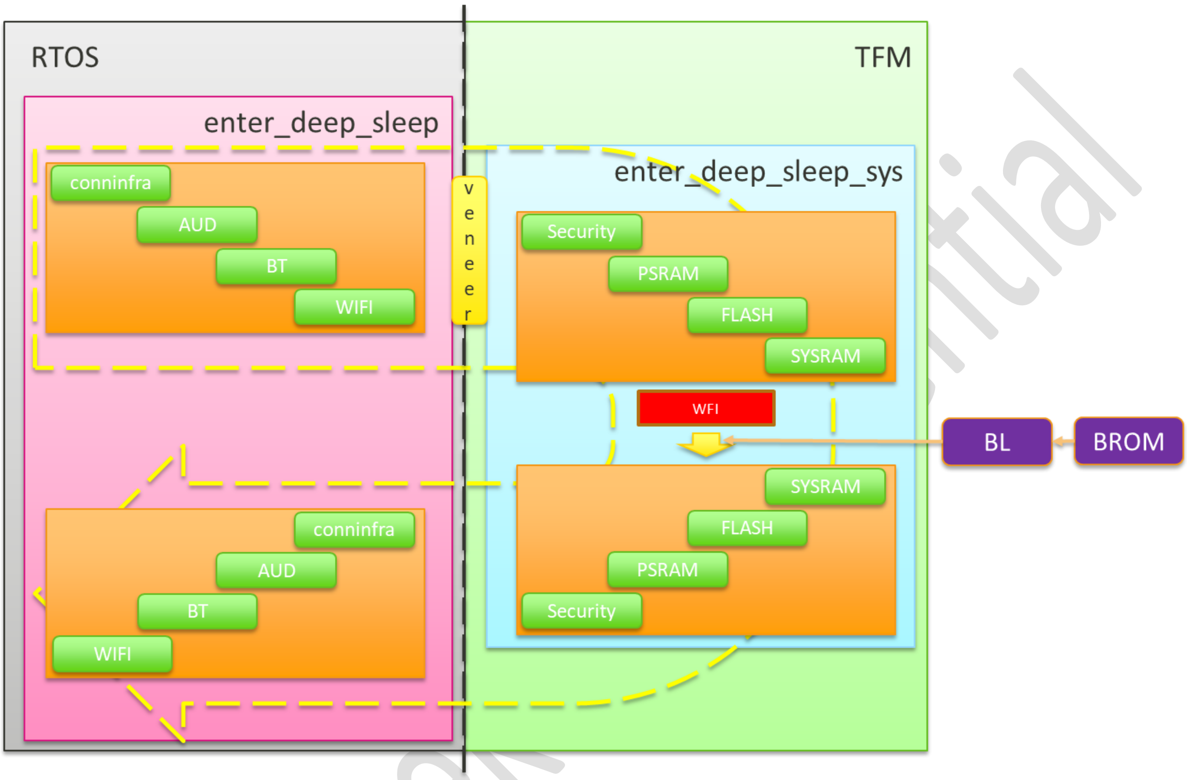

Figure 4-2. Sleep and wakeup on TF-M projects

5 TF-M Functions Test

TF-M provides many services such as crypto, ITS, and PS to NSPE tasks to access by veneers. To check whether the functionalities of the services work as expected, TF-M provides TF-M test suites. The MT793X integrates TF-M internal test suites into CLI for the user to test TF-M service functions. The MT793X CLI includes two test commands – tfm test_s and tfm test_ns.

5.1 tfm test_s

As Figure 5-1 shows, when you input “tfm test_s” to CLI, the TF-M test suite starts to execute, and the console outputs each result of the sub-test case. After all test cases are complete, the TF-M test suite shows a summary report on the console. You can use the report to check TF-M function status easily.

```
$ tfm test_s
TFM test service partition in secure world start.
#### Execute test suites for the Secure area ####

**WARNING** The SST regression tests reduce the life of the flash memory as they write/erase multiple times the memory.
Please, set the SST_RAM_FS flag to use RAM instead of flash.

Running Test Suite PSA protected storage S interface tests (TFM_SST_TEST_2XXX)...
> Executing 'TFM_SST_TEST_2001'
  Description: 'Set interface'
  TEST PASSED!
> Executing 'TFM_SST_TEST_2002'
  Description: 'Set interface with create flags'
  TEST PASSED!
```

Figure 5-1. TF-M test suite tfm test_s

```
*** Secure test suites summary ***
Test suite 'PSA protected storage S interface tests (TFM_SST_TEST_2XXX)' has  PASSED
Test suite 'SST reliability tests (TFM_SST_TEST_3XXX)' has  PASSED
Test suite 'SST rollback protection tests (TFM_SST_TEST_4XXX)' has  PASSED
Test suite 'PSA internal trusted storage S interface tests (TFM_ITS_TEST_2XXX)' has  PASSED
Test suite 'ITS reliability tests (TFM_ITS_TEST_3XXX)' has  PASSED
Test suite 'Crypto secure interface tests (TFM_CRYPTO_TEST_5XXX)' has  PASSED
Test suite 'Initial Attestation Service secure interface tests(TFM_ATTEST_TEST_1XXX)' has  PASSED
Test suite 'Platform Service Secure interface tests(TFM_PLATFORM_TEST_1XXX)' has  PASSED
Test suite 'Audit Logging secure interface test (TFM_AUDIT_TEST_1XXX)' has  PASSED
```

Figure 5-2. TF-M test suite summary report

5.2 tfm test_ns

If one sub-test case fails, the test case is determined as failed. As Figure 5-3 shows, the sub-test case “TFM_ITS_TEST_1002” fails, and “TFM_ITS_TEST_1XXX” is determined as failed in the test summary report.

```
Running Test Suite PSA internal trusted storage NS interface tests (TFM_ITS_TEST_1XXX)...
> Executing 'TFM_ITS_TEST_1001'
  Description: 'Set interface'
  TEST PASSED!
> Executing 'TFM_ITS_TEST_1002'
  Description: 'Set interface with create flags'
  Set should not fail with no flags (Failed at /mfs/mcpslt0034/mtk18285/Hadron/hadron_082607_vio158_test/middleware/third_party/tfm/trusted-firmware-m/test/suites/its/its_tests_common.c:75
)
  TEST FAILED!
> Executing 'TFM_ITS_TEST_1003'
  Description: 'Set interface with NULL data pointer'
  TEST PASSED!
```

Figure 5-3. Failed TF-M test case

Please note that the test case TFM_AUDIT_TEST_1XXX in tfm test_ns is dependent on the result of tfm test_s, so before you run the test case tfm test_ns, you need to run tfm test_s first, or TFM_AUDIT_TEST_1XXX in tfm test_ns will fail as Figure 5-4 shows.

```
*** Non-secure test suites summary ***
Test suite 'PSA protected storage NS interface tests (TFM_SST_TEST_1XXX)' has  PASSED
Test suite 'PSA internal trusted storage NS interface tests (TFM_ITS_TEST_1XXX)' has  FAILED
Test suite 'Crypto non-secure interface test (TFM_CRYPTO_TEST_6XXX)' has  PASSED
Test suite 'Initial Attestation Service non-secure interface tests(TFM_ATTEST_TEST_2XXX)' has  PASSED
Test suite 'Platform Service Non-Secure interface tests(TFM_PLATFORM_TEST_2XXX)' has  PASSED
Test suite 'QCBOR regression test(TFM_QCBOR_TEST_7XXX)' has  PASSED
Test suite 'T_COSE regression test(TFM_T_COSE_TEST_8XXX)' has  PASSED
Test suite 'AuditLog non-secure interface test (TFM_AUDIT_TEST_1XXX)' has  FAILED
Test suite 'Core non-secure positive tests (TFM_CORE_TEST_1XXX)' has  PASSED
```

Figure 5-4. TF-M test suite summary report with TFM_AUDIT_TEST_1XXX failed

6 Appendix A: Acronyms and Abbreviations

The acronyms and abbreviations used in this user guide are listed in the following table.

Table 6-1. Acronyms and abbreviations

Acronym/Abbreviation	Definition
TF-M	Trusted Firmware-M
ITS	Internal Trusted Storage
PS	Protected Storage
CM33	Cortex-M33
TZ	TrustZone
TEE	Trusted Execution Environment
SPE	Secure Processing Environment
NSPE	Non-Secure Processing Environment
DAPC	Device Access Permission Control
SAU	Secure Attribution Unit
NSC	Non-Secure Callable
APC	Access Permission Control
CPU	Central Processing Unit
RTOS	Real Time Operating System
FW	Firmware
CLI	Command-Line Interface

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.