Airoha IoT SDK for RTOS Wi-Fi Profile API Migration Guide

Version: 1.3

Release date: 15 September 2017

Document Revision History

Revision	Date	Description
1.0	4 November 2016	Initial release
1.1	5 May 2017	Added migration guide for version 4.2.0 to 4.3.0
1.2	6 July 2017	• Updated the configuration paths and files in section 3.1.1, “Files under EWARM, GCC and MDK-ARM folders”. • Updated error description in sections 3.2.1.3, “Other files” and 3.2.2.2, “`task_def.h` modification”.
1.3	15 September 2017	• Updated the migration method for default configuration in the `wifi_default_config.h`, see section 4, “Migration Guide from SDK v4.3.0 to v4.5.0”. • Removed `wifi_connection_inform_ip_ready()` function and migration method in section 5, “Migration Guide from SDK v4.5.0 to v4.6.0”. • Added sections to modify IAR and Keil project configurations, see section 5.1.3, “Modify IAR project configuration file (`*.ewp`)”, 5.1.4, “Modify Keil project configuration file (`*.uvprojx`”

Table of Contents

Lists of Tables and Figures

1. Overview

This document provides details on how to migrate Wi-Fi module changes between different SDK versions, including Wi-Fi APIs and example projects.

2. Migration Guide from SDK v4.0.0 to v4.1.0

The Wi-Fi profile APIs are deprecated starting from Airoha IoT SDK v4.1.0. This guide offers a smooth transition from old deprecated APIs to the new APIs.

This chapter guides you through an example to replace the deprecated APIs.

2.1. Wi-Fi profile API Migration

All Wi-Fi profile APIs are deprecated starting from Airoha IoT SDK v4.1.0. The deprecated profile APIs are listed in Table 1.

Table 1. Deprecated Wi-Fi APIs in v4.1.0

Deprecated Profile API	Deprecated Profile API
`wifi_profile_set_opmode()`	`wifi_profile_get_opmode()`
`wifi_profile_set_channel()`	`wifi_profile_get_channel()`
`wifi_profile_set_bandwidth()`	`wifi_profile_get_bandwidth()`
`wifi_profile_set_mac_address()`	`wifi_profile_get_mac_address()`
`wifi_profile_set_ssid()`	`wifi_profile_get_ssid()`
`wifi_profile_set_wireless_mode()`	`wifi_profile_get_wireless_mode()`
`wifi_profile_set_security_mode()`	`wifi_profile_get_security_mode()`
`wifi_profile_set_wpa_psk_key()`	`wifi_profile_get_wpa_psk_key()`
`wifi_profile_set_pmk()`	`wifi_profile_get_pmk()`
`wifi_profile_set_wep_key()`	`wifi_profile_get_wep_key()`
`wifi_profile_set_country_region()`	`wifi_profile_get_country_region()`
`wifi_profile_set_dtim_interval()`	`wifi_profile_get_dtim_interval()`
`wifi_profile_set_listen_interval()`	`wifi_profile_get_listen_interval()`
`wifi_profile_set_power_save_mode()`	`wifi_profile_get_power_save_mode()`
`wifi_profile_commit_setting()`	`wifi_profile_get_profile()`

2.2. Using wifi_init() API

The list of deprecated Wi-Fi profile APIs that can be replaced with `wifi_init()` API is shown in Table 2:

Table 2. `wifi_init()` to replace the Wi-Fi deprecated APIs

Deprecated Profile API	Deprecated Profile API
`wifi_profile_set_opmode();`	`wifi_profile_set_channel();`
`wifi_profile_set_bandwidth();`	`wifi_profile_set_ssid();`
`wifi_profile_set_wireless_mode();`	`wifi_profile_set_security_mode();`
`wifi_profile_set_wpa_psk_key();`	`wifi_profile_set_wep_key();`
`wifi_profile_set_dtim_interval();`	`wifi_profile_set_listen_interval();`
`wifi_profile_set_power_save_mode()`	

The `wifi_init()` API initializes the Wi-Fi module at boot up. Before calling this API, configure the profile settings in `wifi_config_t` and `wifi_config_ext_t` structures. Initializing the `wifi_config_t` settings is mandatory, while initializing `wifi_config_ext_t` is optional. More details on `wifi_init()` can be found in Wi-Fi API reference.

The structures `wifi_config_t` and `wifi_config_ext_t` have a set of parameters to map the deprecated Wi-Fi profile APIs, as shown in Table 3.

Table 3. The mapping between deprecated Wi-Fi profile API and `wifi_init()`

Deprecated Profile APIs	wifi_config_t	wifi_config_ext_t
wifi_profile_set_opmode()	opmode	–
wifi_profile_set_channel()	ap_config.channel	–
wifi_profile_set_bandwidth()	ap_config.bandwidth	–
wifi_profile_set_mac_address()	–	–
wifi_profile_set_ssid()	sta_config.ssid sta_config.ssid_length ap_config.ssid ap_config.ssid_length	–
wifi_profile_set_wireless_mode()	–	sta_wireless_mode ap_wireless_mode
wifi_profile_set_security_mode()	ap_config.auth_mode ap_config.encrypt_type	–
wifi_profile_set_wpa_psk_key()	sta_config.password sta_config.password_length ap_config.password ap_config.password_length	–
wifi_profile_set_pmk()	–	–
wifi_profile_set_wep_key()	sta_config.password sta_config.password_length ap_config.password ap_config.password_length	–
wifi_profile_set_country_region()	–	–
wifi_profile_set_dtim_interval()	–	ap_dtim_interval
wifi_profile_set_listen_interval()	–	sta_listen_interval
wifi_profile_set_power_save_mode()	–	sta_power_save_mode
wifi_profile_commit_setting()	–	–
wifi_profile_get_profile()	–	–

2.2.1. Migration steps from deprecated Wi-Fi profile APIs

Before migrating the Wi-Fi profile APIs:

1) Profile APIs are used to get or set the Wi-Fi profile in NVDM. The profile settings can be read by Wi-Fi profile get APIs.
2) At system boot up, the Wi-Fi driver configurations were initialized automatically based on the profile settings.

After migration:

1) Store the Wi-Fi settings in NVDM, flash, macro or another storage supported by the HDK, instead of using `wifi_profile_set_xxxx()` APIs.

2) At system boot up, the Wi-Fi driver configurations cannot be initialized automatically. Call the function `wifi_init()` to initialize the Wi-Fi driver. User can load the Wi-Fi settings from the storage to initialize the structure of `wifi_init()`.

The Wi-Fi initialization flow before and after migration is shown in Figure 1.

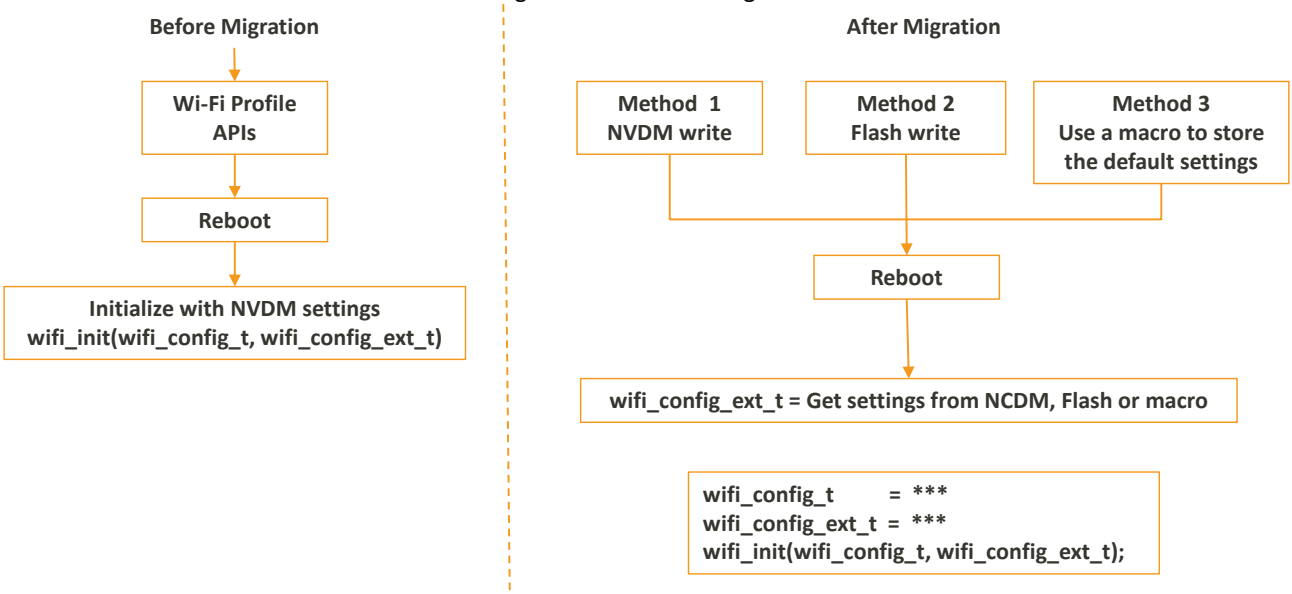

Figure 1. Initialization flow before and after migration

An example implementation of using NVDM (Method 1) to initialize the opmode, SSID, password and wireless mode to initialize the Wi-Fi settings, is shown below. Note that, you can also use Method 2 and Method 3 (see Figure 1).

1) Write settings in NVDM.

```
char opmode = '1';
char ssid[] = "AP1";
char ssid_len[] = "3";
char password[] = "12345678";
char password_len = '8';
char wireless_mode = '9';
if (NVDM_STATUS_OK != nvdm_write_data_item("common", "OpMode",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)opmode, os_strlen(opmode))) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "Ssid",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)ssid, os_strlen(ssid))) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "SsidLen",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)ssid_len, os_strlen(ssid_len))) {
    return -1;
}
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "PassWord",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)password, os_strlen(password))) {
    return -1;
}
```

```
if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "PassWordLen",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)password_len, os_strlen(password_len))) {
    return -1;
}

if (NVDM_STATUS_OK != nvdm_write_data_item("STA", "WirelessMode",
        NVDM_DATA_ITEM_TYPE_STRING,
        (uint8_t *)wireless_mode, os_strlen(wireless_mode))) {
    return -1;
}
```

2) Load Wi-Fi settings from storage medium when boot up.

```
wifi_config_t config = {0};
wifi_config_ext_t config_ext = {0};

uint8_t buff[128];
uint32_t len = sizeof(buff);
nvdm_read_data_item("common", "OpMode", buff, &len);
config.opmode = (uint8_t)atoi((char *)buff);

len = sizeof(buff);
nvdm_read_data_item("STA", "SsidLen", buff, &len);
config.sta_config.ssid_length = (uint8_t)atoi((char *)buff);

len = sizeof(buff);
nvdm_read_data_item("STA", "Ssid", buff, &len);
memcpy(config.sta_config.ssid, buff, config.sta_config.ssid_length);

len = sizeof(buff);
nvdm_read_data_item("STA", "PassWordLen", buff, &len);
config.sta_config.password_length = (uint8_t)atoi((char *)buff);

len = sizeof(buff);
nvdm_read_data_item("STA", "PassWord", buff, &len);
memcpy(config.sta_config.password, buff,
config.sta_config.password_length);

len = sizeof(buff);
nvdm_read_data_item("STA", "WirelessMode", buff, &len);
config_ext->sta_wireless_mode_present = 1;
config_ext->sta_wireless_mode = (uint8_t)atoi((char *)buff);
```

3) Call the function `wifi_init()` to initialize the Wi-Fi profile.

```
wifi_init(&config, &config_ext);
```

2.3. Other deprecated Wi-Fi profile APIs

2.3.1. `wifi_profile_set_mac_address()`

This API should no longer be used and can be removed from the source code, as eFuse is used to load and set the MAC address.

2.3.2. `wifi_profile_set_pmk()`

This API should no longer be used and can be removed from the source code.

2.3.3. `wifi_profile_set_country_region()`

This API should be removed from the source code, as the country code is set through `wifi_init()`.

2.3.4. `wifi_profile_commit_setting()` and `wifi_profile_get_profile()`

These two APIs should no longer be used and can be removed from the source code.

2.3.5. `wifi_profile_get_xxxx()` APIs

The deprecated `wifi_profile_get_xxxx()` APIs were used to read settings from the NVDM. However, with current APIs the settings can be read from NVDM, flash or macro sees section 2.2.1, “Migration steps from deprecated Wi-Fi profile APIs”.

3. Migration Guide from SDK v4.2.0 to v4.3.0

In SDK v4.3.0, Wi-Fi folder path and naming are adjusted and additional chipsets are supported including MT7682 and MT7686.

This section provides two examples; one is to migrate MT7687 based project from v4.2.0 to v4.3.0, the other is to migrate the MT7687 based project to MT7682 chipset using v4.3.0.

3.1. Migrating MT7687 based project from v4.2.0 to v4.3.0

The path to the MT7687 project is `<sdk_root>\project\mt7687_hdk\apps\iot_sdk_demo`.

Only the files descibed below require modification, others remain the same.

3.1.1. Files under EWARM, GCC and MDK-ARM folders

The Wi-Fi folder structure is different compared to SDK_V4.2.0 that results in project configuration change. Run the script "update_420proj_to_430.pl" in the path '`<sdk_root>/tools/scripts/migration`' to automatically modify the folder and file paths in EWARM, GCC and MDK-ARM configuration files. For the script usage, please refer to section 2.3 in 'Airoha_IoT_SDK_Release_Notes.pdf' under `<sdk_root>/doc` folder. In addition, the following file paths in EWARM, GCC and MDK-ARM configuration files need to be modified manually.

1) Modified `EWARM/iot_sdk_demo.ewp` configuration files:

- Replace

```
<state>MTK_SMTCN_ENABLE</state>
```

With

```
<state>MTK_FLASH_DIRECT_DL</state>
<state>MTK_PATCH_DL_ENABLE</state>
<state>MTK_SMTCN_V5_ENABLE</state>
```

- Replace

```
<file>
<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\smtcn\src\elian.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\smtcn\lib\libsmtcn_CM4_IAR.
a</name>
</file>
```

With

```
<file>
<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\smtcn\src\bsmtcn_ops.c</nam
e>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\smtcn\src\ops_config.c</nam
e>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\prebuilt\middleware\MTK\smtcn\lib\libbcsmt
cn_CM4_IAR.a</name>
</file>
```

- Remove

```
<file>
<name>$PROJ_DIR$\..\..\..\..\common\bsp_ex\src\wifi_ex_cli.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\common\bsp_ex\src\wifi_ex_config.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\common\bsp_ex\src\wifi_ex_connect.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\common\bsp_ex\src\wifi_ex_profile.c</name>
</file>
```

- Remove

```
<file>
<name>$PROJ_DIR$\..\..\..\..\..\kernel\service\lib\libkservice_CM4_MT7687_IAR.a</name>
</file>
```

- Add

```
<state>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\smtcn\inc\internal</state>
<state>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\connsys\inc</state>
```

- Add

```
<file>
<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\wifi_service\combo\src\
inband_queue_option.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\driver\chip\mt7687\src\hal_misc.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\kernel\service\src\context_info_save.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\kernel\service\src\exception_handler.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\kernel\service\src\syslog.c</name>
</file>
<file>
<name>$PROJ_DIR$\..\..\..\..\..\kernel\service\src\toi.c</name>
</file>
```

2) Modify `MDK-ARM\iot_sdk_demo.uvprojx` configuration files, as shown below:

- Replace

```
 <Define>PCFG_OS=2 _REENT_SMALL MTK_MINISUPP_ENABLE MTK_MINICLI_ENABLE
MTK_BSPEXT_ENABLE MTK_HAL_LOWPOWER_ENABLE MTK_LWIP_ENABLE MTK_IPERF_ENABLE
PRODUCT_VERSION=7687 MTK_SMTCN_ENABLE
MTK_WIFI_REPEATER_ENABLE  CONFIG_REPEATER MTK_DEBUG_LEVEL_INFO
MTK_DEBUG_LEVEL_WARNING MTK_DEBUG_LEVEL_ERROR  CFG_SUPPORT_SMNT_PROTO=2
MTK_PING_OUT_ENABLE MTK_WIFI_WPS_ENABLE MTK_NVDM_ENABLE</Define>
```

With

```
<Define>PCFG_OS=2 _REENT_SMALL MTK_MINISUPP_ENABLE MTK_MINICLI_ENABLE
MTK_BSPEXT_ENABLE MTK_HAL_LOWPOWER_ENABLE MTK_LWIP_ENABLE MTK_IPERF_ENABLE
PRODUCT_VERSION=7687 MTK_FLASH_DIRECT_DL MTK_PATCH_DL_ENABLE
MTK_SMTCN_V5_ENABLE MTK_WIFI_REPEATER_ENABLE  CONFIG_REPEATER
MTK_DEBUG_LEVEL_INFO MTK_DEBUG_LEVEL_WARNING
```

```
MTK_DEBUG_LEVEL_ERROR  CFG_SUPPORT_SMNT_PROTO=2 MTK_PING_OUT_ENABLE
MTK_WIFI_WPS_ENABLE MTK_NVDM_ENABLE</Define>
```

- Replace

```
<FileName>elian.c</FileName>
   <FileType>1</FileType>
<FilePath>..\..\..\..\..\middleware\MTK\smtcn\src\elian.c</FilePath>
```

With

```
<File>
        <FileName>bsmtcn_ops.c</FileName>
   <FileType>1</FileType>
   <FilePath>..\..\..\..\..\middleware\MTK\smtcn\src\bsmtcn_ops.c</FilePat
h>
</File>
<File>
   <FileName>ops_config.c</FileName>
   <FileType>1</FileType>
   <FilePath>..\..\..\..\..\middleware\MTK\smtcn\src\ops_config.c</FilePat
h>
</File>
```

- Replace

```
 <FileName>libsmtcn_CM4_Keil.lib</FileName>
```

With

```
<FileName>libbcsmtcn_CM4_Keil.lib</FileName>
```

- Replace

```
<FilePath>..\..\..\..\..\middleware\MTK\smtcn\lib\libsmtcn_CM4_Keil.lib</F
ilePath>
```

With

```
<FilePath>..\..\..\..\..\prebuilt\middleware\MTK\smtcn\lib\libbcsmtcn_CM4_
Keil.lib</FilePath>
```

- Remove

```
    <FileName>wifi_ex_cli.c</FileName>
  <FileType>1</FileType>
  <FilePath>..\..\..\..\common\bsp_ex\src\wifi_ex_cli.c</FilePath>
</File>
<File>
   <FileName>wifi_ex_config.c</FileName>
   <FileType>1</FileType>
   <FilePath>..\..\..\..\common\bsp_ex\src\wifi_ex_config.c</FilePath>
</File>
<File>
   <FileName>wifi_ex_connect.c</FileName>
   <FileType>1</FileType>
   <FilePath>..\..\..\..\common\bsp_ex\src\wifi_ex_connect.c</FilePath>
</File>
<File>
   <FileName>wifi_ex_profile.c</FileName>
   <FileType>1</FileType>
   <FilePath>..\..\..\..\common\bsp_ex\src\wifi_ex_profile.c</FilePath>
</File>
<File>
```

- Add

```
<File>
<FileName>inband_queue_option.c</FileName>
<FileType>1</FileType>
<FilePath>..\..\..\..\..\middleware\MTK\wifi_service\combo\src\
inband_queue_option.c</FilePath>
</File>
```

- Add

```
<File>
 <FileName>hal_misc.c</FileName>
 <FileType>1</FileType>
 <FilePath>..\..\..\..\..\driver\chip\mt7687\src\hal_misc.c</FilePath>
</File>
```

3) Modified GCC\feature.mk:

a) Feature options to change:

Original	Change to
MTK_SMTCN_ENABLE = y	MTK_SMTCN_V5_ENABLE = y

b) Makefile to change:

- Replace

```
ifeq ($(MTK_SMTCN_ENABLE),y)
```

With

```
ifeq ($(findstring y,$(MTK_SMTCN_V4_ENABLE)$(MTK_SMTCN_V5_ENABLE)),y)
```

- Remove

```
ifeq ($(MTK_MINICLI_ENABLE),y)
APP_FILES       += driver/board/mt76x7_hdk/util/src/io_def.c
endif
```

- Remove

```
ifneq ($(MTK_DEBUG_LEVEL), none)
LDFLAGS += -Wl,-wrap=printf
endif
```

- Add

```
#connsys_module
include $(SOURCE_DIR)/middleware/MTK/connsys/module.mk
```

- Add

```
include $(SOURCE_DIR)/driver/board/mt76x7_hdk/util/module.mk
```

3.1.2. Files under src folder

Copy the files main.c, system_mt7687.c, wifi_lwip_helper.c from <sdk_root>\project\mt7687_hdk\apps\iot_sdk_demo\src of SDK V4.3.0 to target src folder.

3.2. Migrating MT7687 based project to MT7682

The path to the MT7687 project is `<sdk_root>\project\mt7687_hdk\apps\iot_sdk_demo`.

The path to the MT7682 project is `<sdk_root>\project\mt7682_hdk\apps\iot_sdk_demo`.

Only the files descibed below require modification, others remain the same.

3.2.1. Files under GCC folder

3.2.1.1. `feature.mk`

Feature options to change:

Original		Change to	
IC_CONFIG	= mt7687	IC_CONFIG	= mt7682
BOARD_CONFIG	= mt7687_hdk	BOARD_CONFIG	= mt7682_hdk
MTK_FW_VERSION	= mt7687_fw	MTK_FW_VERSION	= mt7682_fw

Feature options to remove		Feature options to add	
MTK_BSPEXT_ENABLE	= y	MTK_CM4_WIFI_TASK_ENABLE	= y
MTK_MINISUPP_ENABLE	= y	MTK_WIFI_ROM_ENABLE	= y
MTK_WIFI_WPS_ENABLE	= y	MTK_NO_PSRAM_ENABLE	= y
MTK_WIFI_DIRECT_ENABLE	= n	MTK_MEMORY_WITH_PSRAM_FLASH	= n
MTK_WIFI_REPEATER_ENABLE	= y	MTK_MEMORY_WITHOUT_PSRAM	= y
		MTK_MEMORY_WITHOUT_PSRAM_FLASH	= n

3.2.1.2. `Makefile`

- Replace

```
# HAL driver files
include $(SOURCE_DIR)/driver/chip/mt7687/module.mk
```

With

```
# HAL driver files
include $(SOURCE_DIR)/driver/chip/mt7686/module.mk
```

- Replace

```
# EPT Config
-include $(SOURCE_DIR)/driver/board/mt76x7_hdk/ept/module.mk
```

With

```
# EPT Config
-include $(SOURCE_DIR)/driver/board/mt7686_hdk/ept/module.mk
```

- Replace

```
# Minisupp Config
ifneq ($(wildcard $(strip $(SOURCE_DIR))/middleware/MTK/minisupp/),)
include $(SOURCE_DIR)/middleware/MTK/minisupp/module.mk
else
include $(SOURCE_DIR)/prebuilt/middleware/MTK/minisupp/module.mk
endif
```

With

```
# WiFi  driver files
ifeq ($(MTK_CM4_WIFI_TASK_ENABLE),y)
ifeq ($(MTK_WIFI_ROM_ENABLE), y)
ifneq ($(wildcard $(strip $(SOURCE_DIR))/middleware/MTK/wifi_stack),)
include $(SOURCE_DIR)/middleware/MTK/wifi_stack/module.mk
include $(SOURCE_DIR)/middleware/MTK/connsys/module.mk
else
include $(SOURCE_DIR)/prebuilt/middleware/MTK/wifi_stack/module.mk
endif
endif
endif
```

- Replace

```
SYS_FILES       = $(APP_PATH_SRC)/system_mt7687.c
```

With

```
SYS_FILES       = $(APP_PATH_SRC)/system_mt7682.c
```

- Replace

```
S_FILES         += $(APP_PATH)/GCC/startup_mt7687.s
```

With

```
S_FILES         += $(APP_PATH)/GCC/startup_mt7682.s
```

- Replace

```
CFLAGS += -I$(SOURCE_DIR)/driver/chip/$(IC_CONFIG)/inc
```

With

```
CFLAGS += -I$(SOURCE_DIR)/driver/chip/mt7686/inc
```

- Replace

```
CFLAGS += -I$(SOURCE_DIR)/driver/board/mt76x7_hdk/ept/inc
```

With

```
CFLAGS += -I$(SOURCE_DIR)/driver/board/mt7686_hdk/ept/inc
```

- Replace

```
CFLAGS += -I$(SOURCE_DIR)/driver/board/mt76x7_hdk/ept/inc
```

With

```
CFLAGS += -I$(SOURCE_DIR)/driver/board/mt7686_hdk/ept/inc
```

- Replace

```
ifeq ($(RAM_BOOTING), 1)
LDFLAGS += -Wl,-Tmt7687_sram.ld -Wl,--gc-sections
else
LDFLAGS += -Wl,-Tmt7687_flash.ld -Wl,--gc-sections
endif
```

With

```
ifeq ($(RAM_BOOTING), 1)
LDFLAGS += -Wl,-Tmt7682_sram.ld -Wl,--gc-sections
else
LDFLAGS += -Wl,-Tmt7682_flash.ld -Wl,--gc-sections
endif
```

- Replace

```
include $(SOURCE_DIR)/driver/board/mt76x7_hdk/util/module.mk
```

With

```
include $(SOURCE_DIR)/driver/board/mt7682_hdk/util/module.mk
```

- Replace

```
$(OUTPATH)/$(PROJ_NAME).elf: $(C_OBJS) $(CXX_OBJS) $(S_OBJS) $(LIBS)
```

With

```
$(OUTPATH)/$(PROJ_NAME).elf: $(C_OBJS) $(CXX_OBJS) $(S_OBJS) $(LIBS)
$(WIFI_ROM_SYM)
```

- Remove

```
#connsys_module
include $(SOURCE_DIR)/middleware/MTK/connsys/module.mk
include $(SOURCE_DIR)/project/common/bsp_ex/module.mk
```

- Add

```
APP_FILES       += $(APP_PATH_SRC)/mem_layout_info.c
```

3.2.1.3. Other files

- Delete the files `mt7687_flash.ld`, `mt7687_hdk.cmm`, `mt7687_sram.ld`, `startup_mt7687.s` and `syscalls.c`.
- Copy the files `mt7682_flash.ld`, `mt7682_hdk.cmm`, `startup_mt7682.s` and `syscalls.c` from `<sdk_root>\project\mt7682_hdk\apps\iot_sdk_demo\GCC` to target GCC folder.

3.2.2. Header files

3.2.2.1. `FreeRTOSConfig.h` modification

- Replace

```
#define configTOTAL_HEAP_SIZE           ( ( size_t ) ( 68 * 1024 ) )
```

With

```
#define configTOTAL_HEAP_SIZE           ( ( size_t ) ( 100 * 1024 ) )
```

- Replace

```
#define configPRIO_BITS             __NVIC_PRIO_BITS
#else
#define configPRIO_BITS             3       /* 7 priority levels */
#endif
```

With

```
#define configPRIO_BITS             __NVIC_PRIO_BITS
#else
#define configPRIO_BITS             5     /* 32 priority levels */
#endif
```

- Replace

```
#define configLIBRARY_LOWEST_INTERRUPT_PRIORITY         0xf
```

With

```
#define configLIBRARY_LOWEST_INTERRUPT_PRIORITY 0xff
```

- Add

```
#define configUSE_QUEUE_SETS      1
```

3.2.2.2. `task_def.h` modification

- Replace

```
#define UNIFY_SMTCN_TASK_STACKSIZE          (1024 * 4) /*unit byte!*/
```

With

```
#define UNIFY_SMTCN_TASK_STACKSIZE          (512 * 4) /*unit byte!*/
```

- Replace

```
#if (PRODUCT_VERSION == 7687) || (PRODUCT_VERSION == 7697) ||
defined(MTK_NO_PSRAM_ENABLE)
#define SYSLOG_QUEUE_LENGTH 8
#elif (PRODUCT_VERSION == 2523)
#define SYSLOG_QUEUE_LENGTH 512
#endif
```

With

```
#if (PRODUCT_VERSION == 7687) || (PRODUCT_VERSION == 7697) ||
(PRODUCT_VERSION == 7686) || (PRODUCT_VERSION == 7682) || (PRODUCT_VERSION
== 5932) || defined(MTK_NO_PSRAM_ENABLE)
#define SYSLOG_QUEUE_LENGTH 8
#elif (PRODUCT_VERSION == 2523)
#define SYSLOG_QUEUE_LENGTH 512
#endif
```

- Remove

```
/* for wifi supplicant task */
#define UNIFY_WPA_SUPPLICANT_TASK_NAME         "wpa_supplicant"
#define UNIFY_WPA_SUPPLICANT_TASK_STACKSIZE    (2048*4) /*unit byte!*/
#define UNIFY_WPA_SUPPLICANT_TASK_PRIO         TASK_PRIORITY_ABOVE_NORMAL
```

3.2.2.3. Other files

- Delete the files `ept_gpio_drv.h`, `flash_map.h`, `hal_feature_config.h`.
- Copy the files `ept_gpio_drv.h`, `hal_feature_config.h`, `mem_layout_info.h`, `memory_map.h` and `msdc_custom_config.h` from `<sdk_root>\project\mt7682_hdk\apps\iot_sdk_demo\inc` to target `inc` folder.

3.2.3. `src` folder files

3.2.3.1. `Main.c`

- Remove

```
#ifndef MTK_DEBUG_LEVEL_NONE
log_create_module(main, PRINT_LEVEL_ERROR);

LOG_CONTROL_BLOCK_DECLARE(main);
LOG_CONTROL_BLOCK_DECLARE(common);
LOG_CONTROL_BLOCK_DECLARE(hal);
LOG_CONTROL_BLOCK_DECLARE(lwip);
LOG_CONTROL_BLOCK_DECLARE(minisupp);
LOG_CONTROL_BLOCK_DECLARE(inband);
```

```
LOG_CONTROL_BLOCK_DECLARE(wifi);

log_control_block_t *syslog_control_blocks[] = {
    &LOG_CONTROL_BLOCK_SYMBOL(main),
    &LOG_CONTROL_BLOCK_SYMBOL(common),
    &LOG_CONTROL_BLOCK_SYMBOL(hal),
    &LOG_CONTROL_BLOCK_SYMBOL(lwip),
    &LOG_CONTROL_BLOCK_SYMBOL(minisupp),
    &LOG_CONTROL_BLOCK_SYMBOL(inband),
    &LOG_CONTROL_BLOCK_SYMBOL(wifi),
    NULL
};

static void syslog_config_save(const syslog_config_t *config)
{
    char *syslog_filter_buf;

    syslog_filter_buf = (char*)pvPortMalloc(SYSLOG_FILTER_LEN);
    configASSERT(syslog_filter_buf != NULL);
    syslog_convert_filter_val2str((const log_control_block_t **)config-
>filters, syslog_filter_buf);
    nvdm_write_data_item("common", "syslog_filters", \
                         NVDM_DATA_ITEM_TYPE_STRING, (const uint8_t
*)syslog_filter_buf, strlen(syslog_filter_buf));
    vPortFree(syslog_filter_buf);
}

static uint32_t syslog_config_load(syslog_config_t *config)
{
    uint32_t sz = SYSLOG_FILTER_LEN;
    char *syslog_filter_buf;

    syslog_filter_buf = (char*)pvPortMalloc(SYSLOG_FILTER_LEN);
    configASSERT(syslog_filter_buf != NULL);
    nvdm_read_data_item("common", "syslog_filters", (uint8_t
*)syslog_filter_buf, &sz);
    syslog_convert_filter_str2val(config->filters, syslog_filter_buf);
    vPortFree(syslog_filter_buf);

    return 0;
}
#endif
```

- Remove

```
#ifndef MTK_DEBUG_LEVEL_NONE
    log_init(syslog_config_save, syslog_config_load,
syslog_control_blocks);
#endif
```

3.2.3.2. sys_init.c modification

Apply the following steps to modify the file:

1) Delete this file in `src` folder;
2) Copy the file `<sdk_root>\project\mt7682_hdk\apps\iot_sdk_demo\src\sys_init.c` to `src` folder;

3) Remove the funtion `user_check_default_value()` in `system_init()`

3.2.3.3. `Other files`

- Delete the files `ept_eint_var.c, ept_gpio_var.c, system_mt7687.c.`
- Copy the files `ept_eint_var.c, ept_gpio_var.c, mem_layout_info.c` and `system_mt7682.c` from `<sdk_root>\project\mt7682_hdk\apps\iot_sdk_demo\src` to target `src` folder.

4. Migration Guide from SDK v4.3.0 to v4.5.0

In SDK v4.5.0, Wi-Fi initialization is separated into four sections, and this change will affect developing projects on MT7682, MT7686, MT7687, MT7697 and MT5932 chipsets.

Follow this guide, to migrate projects from SDK v4.3.0 to SDK v4.5.0.

4.1. Migrating default Wi-Fi parameters

Default configurations for Wi-Fi support are defined as macros in the `wifi_default_config.h` header file at `<sdk_root>\middleware\MTK\wifi_service\inc\wifi_default_config.h`. Table 4 provides the list of changes, besides the change in value, the type needs to be changed from String to Integer.

Table 4. Different definitions between SDK v4.3.0 and v4.5.0

Parameter	SDK v4.3.0	SDK v4.5.0
WIFI_DEFAULT_OPMODE	1	"1"
WIFI_DEFAULT_COUNTRY_REGION	5	"5"
WIFI_DEFAULT_COUNTRY_REGION_A_BAND	3	"3"
WIFI_DEFAULT_RADIO_ONOFF	0	"0"
WIFI_DEFAULT_N9_DEBUG_LEVEL	3	"3"
WIFI_DEFAULT_RTS_THRESHOLD	2347	"2347"
WIFI_DEFAULT_FRAGMENT_THRESHOLD	2346	"2346"
WIFI_DEFAULT_WIFI_PRIVILEGE_ENABLE	0	"0"
WIFI_DEFAULT_STA_FAST_LINK	0	"0"
WIFI_DEFAULT_STA_LOCAL_ADMIN_MAC	1	"1"
WIFI_DEFAULT_STA_SSID_LEN	11	"11"
WIFI_DEFAULT_STA_BSS_TYPE	1	"1"
WIFI_DEFAULT_STA_CHANNEL	1	"1"
WIFI_DEFAULT_STA_BANDWIDTH	0	"0"
WIFI_DEFAULT_STA_WIRELESS_MODE	9	"9"
WIFI_DEFAULT_STA_BA_DECLINE	0	"0"
WIFI_DEFAULT_STA_AUTO_BA	1	"1"
WIFI_DEFAULT_STA_HT_MCS	33	"33"
WIFI_DEFAULT_STA_HT_BA_WINDOW_SIZE	64	"64"
WIFI_DEFAULT_STA_HT_GI	1	"1"
WIFI_DEFAULT_STA_HT_PROTECT	1	"1"
WIFI_DEFAULT_STA_HT_EXT_CHANNEL	1	"1"
WIFI_DEFAULT_STA_WMM_CAPABLE	1	"1"
WIFI_DEFAULT_STA_LISTEN_INTERVAL	1	"1"
WIFI_DEFAULT_STA_AUTH_MODE	0	"0"
WIFI_DEFAULT_STA_ENCRYPT_TYPE	1	"1"
WIFI_DEFAULT_STA_WPA_PSK_LEN	8	"8"

Parameter	SDK v4.3.0	SDK v4.5.0
WIFI_DEFAULT_STA_PAIR_CIPHER	0	"0"
WIFI_DEFAULT_STA_GROUP_CIPHER	0	"0"
WIFI_DEFAULT_STA_DEFAULT_KEY_ID	0	"0"
WIFI_DEFAULT_STA_POWER_SAVE_MODE	0	"0"
WIFI_DEFAULT_STA_KEEP_ALIVE_PERIOD	10	"10"
WIFI_DEFAULT_STA_BEACON_LOST_TIME	2	"2"
WIFI_DEFAULT_APCLI_BW_AUTO_UP_BELOW	1	"1"
WIFI_DEFAULT_STA_KEEP_ALIVE_PACKET	1	"1"
WIFI_DEFAULT_AP_LOCAL_ADMIN_MAC	1	"1"
WIFI_DEFAULT_AP_SSID_LEN	11	"11"
WIFI_DEFAULT_AP_CHANNEL	1	"1"
WIFI_DEFAULT_AP_BANDWIDTH	0	"0"
WIFI_DEFAULT_AP_WIRELESS_MODE	9	"9"
WIFI_DEFAULT_AP_AUTO_BA	1	"1"
WIFI_DEFAULT_AP_HT_MCS	33	"33"
WIFI_DEFAULT_AP_HT_BA_WINDOW_SIZE	64	"64"
WIFI_DEFAULT_AP_HT_GI	1	"1"
WIFI_DEFAULT_AP_HT_PROTECT	1	"1"
WIFI_DEFAULT_AP_HT_EXT_CHANNEL	1	"1"
WIFI_DEFAULT_AP_WMM_CAPABLE	1	"1"
WIFI_DEFAULT_AP_DTIM_PERIOD	1	"1"
WIFI_DEFAULT_AP_AUTH_MODE	0	"0"
WIFI_DEFAULT_AP_ENCRYPT_TYPE	1	"1"
WIFI_DEFAULT_AP_WPA_PSK_LEN	8	"8"
WIFI_DEFAULT_AP_PAIR_CIPHER	0	"0"
WIFI_DEFAULT_AP_GROUP_CIPHER	0	"0"
WIFI_DEFAULT_AP_DEFAULT_KEY_ID	0	"0"
WIFI_DEFAULT_AP_HIDDEN_SSID	0	"0"
WIFI_DEFAULT_AP_REKEY_INTERVAL	3600	"3600"
WIFI_DEFAULT_AP_AUTO_CHANNEL_SELECT	0	"0"
WIFI_DEFAULT_AP_BEACON_DISABLE	0	"0"
WIFI_DEFAULT_MBSS_ENABLE	0	"0"
WIFI_DEFAULT_CONFIG_FREE_READY	0	"0"
WIFI_DEFAULT_CONFIG_FREE_ENABLE	0	"0"

If the projects in SDK v4.3.0 use the macros from Table 4, convert the string type to integer. An example is shown below:

```
uint8_t opmode = WIFI_DEFAULT_OPMODE;
```

Replace with

```
uint8_t opmode = (uint8_t)atoi(WIFI_DEFAULT_OPMODE);
```

5. Migration Guide from SDK v4.5.0 to v4.6.0

In SDK v4.6.0, the `wifi_connection_inform_ip_ready()` API is removed from projects, and this change will affect MT7682, MT7686, MT7687, MT7697 and MT5932 chipsets.

Follow this guide, to migrate projects from SDK v4.5.0 to SDK v4.6.0.

5.1. Migrating project files

5.1.1. Modify `wifi_lwip_helper.c`

Remove `wifi_connection_inform_ip_ready()` function from `ip_ready_callback()` and `wifi_station_port_secure_event_handler()` functions.

- Replace

```
lwip_tcpip_config_t tcpip_config = {{0}, {0}, {0}, {0}, {0}, {0}};
```

With

```
lwip_tcpip_config_t tcpip_config = {0, {0}, {0}, {0}, {0}, {0}, {0}};
```

in `lwip_network_init()` function.

- Replace

```
dhcpd_settings_t dhcpd_settings = {{0}, {0}};
lwip_tcpip_config_t tcpip_config = {{0}, {0}, {0}, {0}, {0}, {0}};
```

With

```
dhcpd_settings_t dhcpd_settings = {{0}, {0}, {0}, {0}, {0}, {0}, {0}};
lwip_tcpip_config_t tcpip_config = {0, {0}, {0}, {0}, {0}, {0}, {0}};
```

in `lwip_net_start()` function.

5.1.2. Modify `network_default_config.c`

- Add into `tcpip_config_init()` function:

```
tcpip_config->ip_mode = dhcp_config_init();
```

5.1.3. Modify IAR project configuration file (*.ewp)

- Open the project file ending with `.ewp`, find the group that includes "`wifi_init.c`" source file, then add the following code into that group:

```
<file>

<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\wifi_service\combo\src\wifi
_cm4_scan.c</name>
</file>
<file>

<name>$PROJ_DIR$\..\..\..\..\..\middleware\MTK\wifi_service\combo\src\wifi
_repeater.c</name>
</file>
```

5.1.4. Modify Keil project configuration file (*.uvprojx)

- Open the project file ending with `.uvprojx`, find the group that includes "`wifi_init.c`" source file, then add the following code into that group:

```
            <File>
               <FileName>wifi_cm4_scan.c</FileName>
               <FileType>1</FileType>

<FilePath>..\..\..\..\..\middleware\MTK\wifi_service\combo\src\wifi_cm4_sc
an.c</FilePath>
             </File>
             <File>
               <FileName>wifi_repeater.c</FileName>
               <FileType>1</FileType>

<FilePath>..\..\..\..\..\middleware\MTK\wifi_service\combo\src\wifi_repeat
er.c</FilePath>
             </File>
```

6. Migration Guide from SDK v4.6.0 to v4.7.0

In SDK v4.7.0, the architecture of MT5932 project was adjusted, the host project wifi5932_ref_design of MT2523 was changed and this change will affect MT5932 and MT2523 Wi-Fi host relate chipset.

Follow this guide, to migrate MT2523 Wi-Fi relate project from SDK v4.6.0 to SDK v4.7.0.

6.1. Migrating MT2523 wifi5932_ref_design project

6.1.1. Files under GCC folder

6.1.1.1. `Makefile`

<sdk_root>/project/mt2523_hdk/apps/wifi5932_ref_design/GCC/Makefile

- Remove

```
CFLAGS      += -DMTK_ATCI_ENABLE
CFLAGS      += -DMTK_WIFI_STUB_CONF_ENABLE
include $(SOURCE_DIR)/middleware/MTK/wifi_service/stub_conf/module.mk
include $(SOURCE_DIR)/middleware/MTK/wifi_service/wifi_host/module.mk
```

- Remove

```
ifeq ($(MTK_WIFI_STUB_CONF_SPIM_ENABLE),y)
CFLAGS      += -DMTK_WIFI_STUB_CONF_SPIM_ENABLE
Endif
```

- Remove

```
ifeq ($(MTK_WIFI_STUB_CONF_SDIO_MSDC_ENABLE),y)
CFLAGS      += -DMTK_WIFI_STUB_CONF_SDIO_MSDC_ENABLE
Endif
```

- Remove

```
ifeq ($(MTK_WFC_WITH_LWIP_NO_WIFI_ENABLE),y)
CFLAGS      += -DMTK_WFC_WITH_LWIP_NO_WIFI_ENABLE
else
CFLAGS      += -DMTK_WFC_WITH_WIFI_NO_LWIP_ENABLE
Endif
```

- Add

```
include $(SOURCE_DIR)/middleware/MTK/wifi_host/wfcm/module.mk
include $(SOURCE_DIR)/middleware/MTK/wifi_host/xboot/module.mk
include $(SOURCE_DIR)/middleware/MTK/wifi_host/bwcs/module.mk
include $(SOURCE_DIR)/middleware/MTK/wifi_host/common/module.mk
include $(SOURCE_DIR)/middleware/MTK/wifi_host/platform/freertos/kal/module.mk
include $(SOURCE_DIR)/middleware/MTK/wifi_host/platform/freertos/hif/sdio/module.mk
include $(SOURCE_DIR)/driver/board/mt25x3_hdk/keypad/module.mk
```

6.1.2. Header files

6.1.2.1. `FreeRTOSConfig.h` modification

<sdk_root>/project/mt2523_hdk/apps/wifi5932_ref_design/inc/FreeRTOSConfig.h

- Add

```
extern void tickless_handler( uint32_t xExpectedIdleTime );\
```

- Add

```
#if configUSE_TICKLESS_IDLE == 1
#define portSUPPRESS_TICKS_AND_SLEEP( xExpectedIdleTime )
vPortSuppressTicksAndSleep( xExpectedIdleTime )
#elif configUSE_TICKLESS_IDLE == 2
#if defined(__ICCARM__) || defined(__CC_ARM) || defined(__GNUC__)
extern void tickless_handler( uint32_t xExpectedIdleTime );
#endif /*#if defined(__ICCARM__) || defined(__CC_ARM) || defined(__GNUC__)*/
#define portSUPPRESS_TICKS_AND_SLEEP( xExpectedIdleTime ) tickless_handler( xExpectedIdleTime )
#endif
```

- Add

```
extern void tickless_handler( uint32_t xExpectedIdleTime );
```

6.1.2.2. `hal_feature_config.h` modification

`<sdk_root>/project/mt2523_hdk/apps/wifi5932_ref_design/inc/hal_feature_config.h`

- Add

```
#define HAL_KEYPAD_MODULE_ENABLED
```

6.1.2.3. lwipopts.h modification

`<sdk_root>`/project/mt2523_hdk/apps/wifi5932_ref_design/inc/lwipopts.h

- Remove

```
#define MEMP_MEM_MALLOC             1
```

- Replace

```
#define PBUF_POOL_SIZE    10
```

With

```
#define PBUF_POOL_SIZE    20
```

6.1.2.4. `msdc_custom_config.h` modification

- Replace

```
#define WIFI_USE_MSDC_PORT_NUMBER (1)
```

With

```
#define WIFI_USE_MSDC_PORT_NUMBER (0)
```

- Replace

```
#define WIFI_MSDC_BUS_WIDTH  (2)
```

With

```
#define WIFI_MSDC_BUS_WIDTH  (1)
```

- Replace

```
#define WIFI_MSDC_BUS_CLOCK  (22000)
```

With

```
#define WIFI_MSDC_BUS_CLOCK  (46000)
```

6.1.3. src folder files

6.1.3.1. main.c modification

<sdk_root>/project/mt2523_hdk/apps/wifi5932_ref_design/src/main.c

- Replace

```
static void wifi_host_main_init(void);
```

With

```
void wifi_host_main_init(void);
```

- Add

```
void wfcm_set_pinmux(void)
{
#if (PRODUCT_VERSION == 2523)
      /* Step1: Call hal_pinmux_set_function() to set GPIO pinmux, if EPT tool was not used to configure the related pinmux.*/
      hal_pinmux_set_function(HAL_GPIO_30, HAL_GPIO_30_MC0_CK);// MC0_CK
      hal_pinmux_set_function(HAL_GPIO_31, HAL_GPIO_31_MC0_CM0);// MC0_CM0
      hal_pinmux_set_function(HAL_GPIO_32, HAL_GPIO_32_MC0_DA0);// MC0_DA0
      hal_pinmux_set_function(HAL_GPIO_33, HAL_GPIO_33_MC0_DA1);// MC0_DA1
      hal_pinmux_set_function(HAL_GPIO_34, HAL_GPIO_34_MC0_DA2);// MC0_DA2
      hal_pinmux_set_function(HAL_GPIO_35, HAL_GPIO_35_MC0_DA3);// MC0_DA3

   #ifndef WFC_HRX_POLLING
      hal_gpio_init(HAL_GPIO_6);
      hal_pinmux_set_function(HAL_GPIO_6, 1);
      hal_gpio_set_direction(HAL_GPIO_6, HAL_GPIO_DIRECTION_INPUT);
      hal_gpio_disable_pull(HAL_GPIO_6);
   #endif
#else
      hal_pinmux_set_function(11, 4);
      hal_pinmux_set_function(12, 4);
      hal_pinmux_set_function(13, 4);
      hal_pinmux_set_function(14, 4);
      hal_pinmux_set_function(15, 4);
      hal_pinmux_set_function(16, 4);

      hal_gpio_set_pupd_register(11, 0, 0, 1);
      hal_gpio_set_pupd_register(12, 0, 0, 1);
      hal_gpio_set_pupd_register(13, 0, 0, 1);
      hal_gpio_set_pupd_register(14, 0, 0, 1);
      hal_gpio_set_pupd_register(15, 0, 0, 1);
      hal_gpio_set_pupd_register(16, 0, 0, 1);

   #ifndef WFC_HRX_POLLING
      hal_gpio_init(HAL_GPIO_17);
      hal_pinmux_set_function(HAL_GPIO_17, 8);
      hal_gpio_set_direction(HAL_GPIO_17, HAL_GPIO_DIRECTION_INPUT);
```

```
        hal_gpio_disable_pull(HAL_GPIO_17);
    #endif
#endif

#ifdef HAL_SLEEP_MANAGER_ENABLED
        hal_gpio_init(HAL_GPIO_24);
        hal_pinmux_set_function(HAL_GPIO_24, 0);
        hal_gpio_set_direction(HAL_GPIO_24, HAL_GPIO_DIRECTION_OUTPUT);
        hal_gpio_set_output(HAL_GPIO_24, HAL_GPIO_DATA_HIGH);
#endif
}
```

- Add in main() function

```
wfcm_set_pinmux();
```

6.1.3.2. `sys_init.c` modification

`<sdk_root>/project/mt2523_hdk/apps/wifi5932_ref_design/src/sys_init.c`

- Add in prvSetupHardware(void) function

```
bsp_ept_gpio_setting_init();
hal_rtc_init();
hal_dcxo_init();
hal_sleep_manager_init();
hal_keypad_powerkey_init(DEVICE_KEY_NONE);
hal_display_dsi_init(false);
hal_display_dsi_deinit();
```