MT793x Hands on Training

2022/10/06

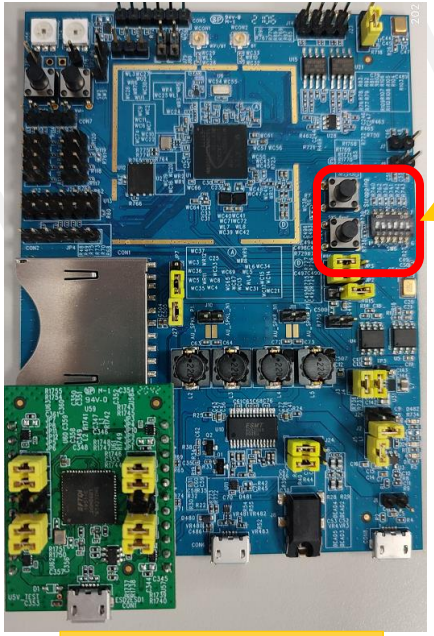
Agenda

- Image Flashing
- CLI Commands
- Fetch SDK
- SDK Architecture
- Build Environments

MT793x HDK Boards

Top to Bottom
- SYSRST
- Download
- Vol+
- Vol-

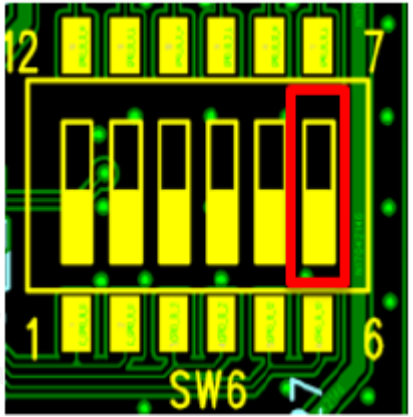
MT7931AN RFB



- DIP Switch
- SYSRST
- RTC_EINT

MT7933CT RFB v2

DIP Switch

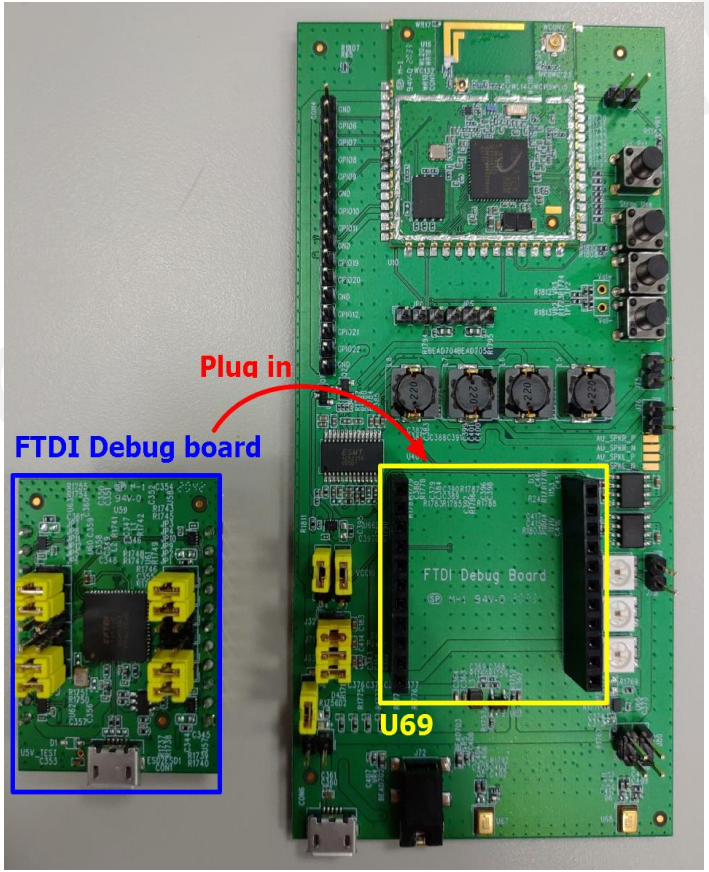


Up: Download Mode

Down: Normal Mode

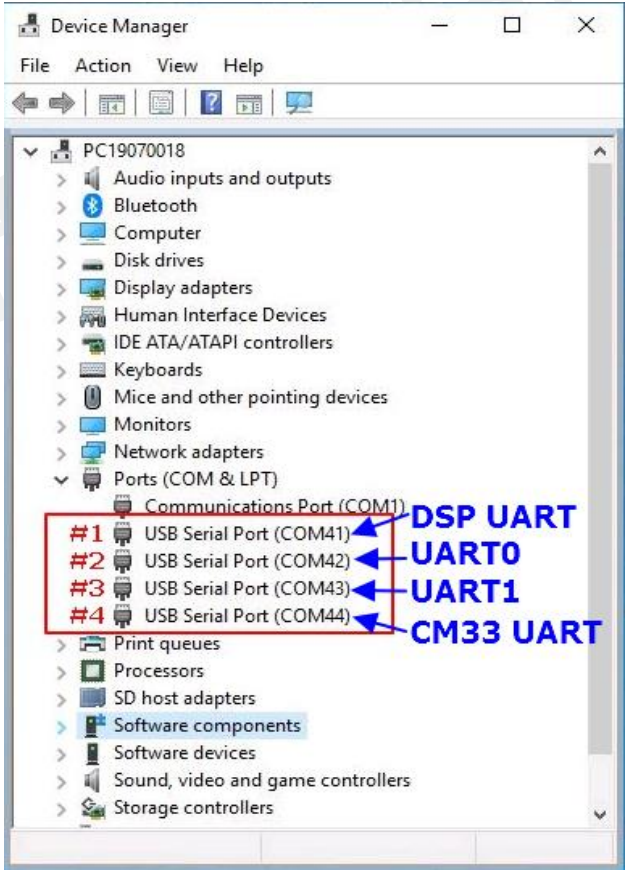
FTDI Debug Board

FTDI debug board make UART signal transfer to USB signal and provide a Micro-USB connector to link to with your PC with USB cable.



COM Port Associated w/ RFB Board

- Will generate 4 COM ports
- Need to install FTDI driver one by one
- CM33 UART is the main working COM port for debugging and entering CLI commands

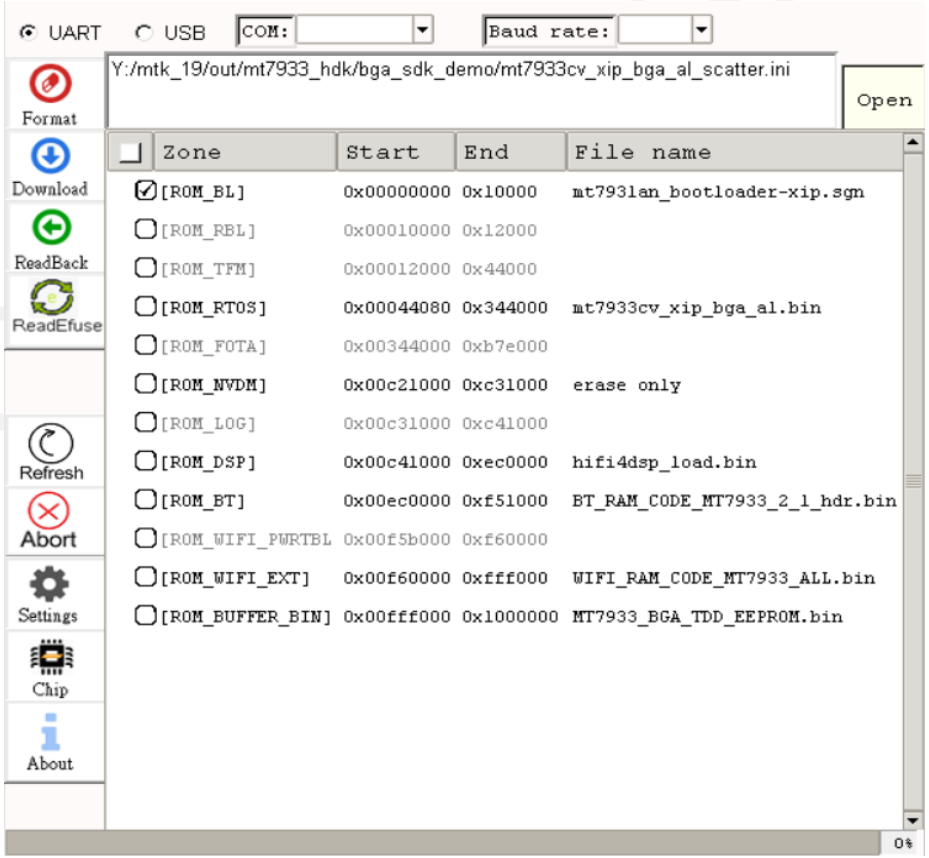


https://ftdichip.com/drivers/vcp-drivers/

FlashBurningTool Operation Step

- Step 1: Press ”**SYSRST”** and ”**Download”** buttons on RFB at the same time and hold them.
- Step 2: Press “**Download”**on FlashBurningTool and release ”**SYSRST”** button immediately.

PS: Only MT7931 needs to press “Download” button.



Agenda

- ☐ Image Flashing
- ☐ CLI Commands
- ☐ Fetch SDK
- ☐ SDK Architecture
- ☐ Build Environments

Normal Mode and Test Mode

- ‘en’ to Test Mode
- ‘back’ to Normal Mode

```
back   - back to normal mode
rr     - read addr
wr     - write addr
os     - os info
reboot - reboot
ble    - bluetooth ble related cmd
picus  - bt picus command
iwpriv - WiFi iw command
wifi   - WiFi Init CLI
lp     - sleep manager cli
lp_dvt - Low Power DVT
iperf  - iperf
ip     - ip config
stat   - show statistics
wifitest - Wifi Test Tool
wpa_cli  - wpa_cli for wpa_supp
```

Utilize ‘?’ and ‘??’ to Check All Available Commands in Normal Mode

- **Help: ‘?’ - list commands**

```
$ ?
mem    - show memory type of <addr>
s      - search <addr> <len> <pat>
d      - dump memory <addr> <len>
f      - fill memory
rr     - read reg
wr     - write reg
wifi   - wifi commands
iwpriv - WiFi iw command
iperf  - iperf
bt     - BT commands
en     - enter test mode
reboot - reboot
ver    - f/w ver
log    - log control
config - user config read/write/reset/show
ble    - bluetooth ble related cmd
mesh   - bluetooth mesh related cmd
picus  - bt picus command
iwpriv - WiFi iw command
wifi   - WiFi Init CLI
ping   - ping <addr> <count> <pkt_len>
iperf  - iperf
ip     - ip config
stat   - show statistics
fota   - FOTA commands
wifitest - Wifi Test Tool
wpa_cli  - wpa_cli for wpa_supp
thermal  - thermal test
os       - os info
```

Utilize ‘?’ and ‘??’ to Check All Available Commands in Normal Mode (Con’t)

- **Help: ‘??’ - list commands**

```
$ ??
mem - show memory type of <addr>
s - search <addr> <len> <pat>
d - dump memory <addr> <len>
f - fill memory
rr - read reg
wr - write reg
wifi - wifi commands
wifi on - Wifi init
wifi info - Wifi info
wifi set_dbg - set init dbg level
wifi get_dbg - get init dbg level
wifi check_lock - check semaphore status
wifi config - wifi config
wifi config set - wifi config set
wifi config set opmode - STA/AP
wifi config set ssid - SSID
wifi config set bssid - BSSID
wifi config set sec - Security
wifi config set msec - multiple security
wifi config set ieee80211w - ieee80211w
wifi config set proto - proto
wifi config set psk - wpa psk key
wifi config set pmk - pmk key
wifi config set wep - wep key
wifi config set bw - bandwidth
wifi config set channel - channel
wifi config set wirelessmode - wireless mode
wifi config set country_code - country code
wifi config set bss_pref - BSS preference
wifi config set radio - OFF/ON
wifi config set ps_mode - PS mode
wifi config set listen - listen interval
wifi config set pretbtt - set pretbtt value in psmode
wifi config set bcn_lost - clear beacon lost count
wifi config set ps_log - powersave log on/off
wifi config set reload - reload
wifi config set dtim - dtim
wifi config set bcn_int - bcn_int
wifi config set retry_limit - retry limit
wifi config set tx_rate - tx rate
wifi config set arp_offload - arp offload
wifi config set rx_filter - set RX Filter
wifi config set mc_address - set mc address
```

Utilize ‘?’ and ‘??’ to Check All Available Commands in Test Mode

- **Help: ‘?’ - list commands**

```
$ ?
back   - back to normal mode
rr     - read addr
wr     - write addr
os     - os info
reboot - reboot
ble    - bluetooth ble related cmd
picus  - bt picus command
iwpriv - WiFi iw command
wifi   - WiFi Init CLI
lp     - sleep manager cli
lp_dvt - Low Power DVT
iperf  - iperf
ip     - ip config
stat   - show statistics
wifitest - Wifi Test Tool
wpa_cli  - wpa_cli for wpa_supp
```

Utilize ‘?’ and ‘??’ to Check All Available Commands in Test Mode (Con’t)

- **Help: ‘??’ - list commands**

```
$ en
$ ??
back - back to normal mode
rr - read addr
wr - write addr
os - os info
os ver - show os version
os cpu - show cpu utilization
os task - show FreeRtos task
os mem - show heap status
os crash - force system crash
os swla - enable/disable swla
os exc - exception handler config
reboot - reboot
ble - bluetooth ble related cmd
picus - bt picus command
iwpriv - WiFi iw command
wifi - WiFi Init CLI
wifi on - Wifi init
wifi info - Wifi info
wifi set_dbg - set init dbg level
wifi get_dbg - get init dbg level
wifi check_lock - check semaphore status
wifi config - wifi config
wifi config set - wifi config set
wifi config set opmode - STA/AP
wifi config set ssid - SSID
wifi config set bssid - BSSID
wifi config set sec - Security
wifi config set msec - multiple security
wifi config set ieee80211w - ieee80211w
wifi config set proto - proto
wifi config set psk - wpa psk key
wifi config set pmk - pmk key
wifi config set wep - wep key
wifi config set bw - bandwidth
wifi config set channel - channel
wifi config set wirelessmode - wireless mode
wifi config set country_code - country code
```

Useful Wifi CLI Commands

- wifi config set ssid 0 <value>
  - Configure AP's SSID
- wifi config set sec 0 <Auth> <Encrypt>
  - Configure Security Setting including auth and encrypt type
- wifi config set psk 0 <password>
  - Configure PSK key
- wifi config set wep 0 <key_id> <key_string>
  - Configure wep key
- wifi config set reload
  - Important: Make above setting to take effect and re-start the connection

sec <Auth> <Encrypt>

- Open (0, 1)
- WPA + TKIP (4, 4)
- WPA2 + CCMP (7, 6)
- WPA & WPA2 + TKIP & AES (9, 8)
- WPA2 & WPA3 + Only CCMP (13, 6)
- WPA3 + only CCMP (11, 6)

Useful Wifi CLI Commands (Con't)

Commands	Usage
wifi info	Wifi connection information
wifi init	Initial Wifi module
wifi deinit	Deinit Wifi module
wifi connect set sched_scan <value>	Start/stop scheduled scan (1: start; 0: stop)
wifi connect get rssi	Get RSSI value
wifi config get wlanstat	Get wlan statistic
Iwpriv wlan driver version	Get Wifi F/W version

iperf Commands

- Start TCP Server
  - iperf -s
- Start UDP Server
  - iperf -s -u
- Start TCP Client
  - iperf -c <server IP> -t <time>
- Start UDP Client
  - iperf -u -c <server IP> -t <time>

Agenda

- ☐ Image Flashing
- ☐ CLI Commands
- ☐ Fetch SDK
- ☐ SDK Architecture
- ☐ Build Environments

Fetch SDK

Download from Git01 by refer to "**SOP_of_Git_Clone_MT793x_SDK_v1p6.txt**"

```
==============================================================================
{Common Information}
==============================================================================
Version: v1.6
Release Date: 2021/02/24
==============================================================================
{Procedures of Git Clone MT793x SDK}
==============================================================================
1. git lfs install

2. Configure your '~/.netrc' as below:
    machine git01.mediatek.com login hadron-mcuiot-ring-git01-user password [redacted]

3. Init & sync 'general' repos:
    a. repo init -u https://git01.mediatek.com/hadron-mcuiot/manifest -b release-hadron.mcuiot.mt7933 -m 2021_02_24_13.xml --no-repo-verify
    b. repo sync -c -j8 --no-tags

4. Execute lfs pull
    repo forall -c 'git lfs pull'

5. Create 'gcc' folder:
    mkdir tools/gcc
    cd tools/gcc

6. Clone 'license' repo:
    a.  git clone https://git01.mediatek.com/hadron-mcuiot/license/tools/gcc/linux
    b.  cd linux
    c.  git checkout origin/release-hadron.mcuiot.mt7933 -b master
```

Need to support Git Large File Storage

Only change .xml number when there's a new release

Agenda

- Image Flashing
- CLI Commands
- Fetch SDK
- Build Environments
- SDK Architecture

Build Environment-1

- **Recommended build environment:**
  - **OS: Linux OS 18.10**
  - **Distribution: Ubuntu 18.10 64bit**
  - **Make: GNU make 3.81**

Build Environment-2

- MCUBOOT comes with an image packing/signing tool called imgtool.py. It is the tool of choice for signing MT7933 images.
- imgtool.py, is written using Python3 and has some Python3 package dependencies that may not be installed in your development environment by default.
- Refer to: Virtual_Environment_Installation_v1.1.docx

Agenda

- Image Flashing
- CLI Commands
- Fetch SDK
- SDK Architecture
- Build Environments

Source Folder Structure-1

- config
- doc
- driver
- kernel
- middleware
- out
- prebuilt
- project
- tools

FreeRTOS: V8.2.0 => V10.2.1
Toolchain: V4.8.4 => V9.2.1

- `config` — includes make and compile configuration files for compiling a binary project.
- `doc` — includes SDK related documentation, such as developer and SDK API reference guides.
- `driver` — includes common driver files, such as board drivers, peripheral and CMSIS-CORE interface drivers.
- `kernel` — includes the underlying RTOS and system services for exception handling and error logging.
- `middleware` — includes software features for HAL and OS, such as network and advanced features.
- `out` — contains binary files, libraries, objects and build logs.
- `prebuilt` — contains binary files, libraries, header files, makefiles and other pre-built files.
- `project` — includes pre-configured example and demo projects using Wi-Fi, HTTP, HAL, and more.
- `tools` — includes tools to compile, download and debug projects using the SDK.

Source Folder Structure-2

```
$ll driver/chip/

21:44 ./
21:43 ../
21:44 inc/
21:44 mt7933/
21:44 mt8512/
```

```
$ll driver/chip/inc

21:44 ./
21:44 ../
21:48 .git/
21:44 .gitattributes
21:44 hal_accdet.h*
21:44 hal_adc.h*
21:44 hal_aes.h*
21:44 hal_audio.h*
21:44 hal_cache.h*
21:44 hal_clock.h*
21:44 hal_dac.h*
21:44 hal_define.h*
21:44 hal_des.h*
21:44 hal_display_color.h*
21:44 hal_display_dsi.h*
21:44 hal_display_lcd.h*
21:44 hal_display_pwm.h*
21:44 hal_dvfs.h*
21:44 hal_ecc.h
21:44 hal_eint.h*
21:44 hal_flash.h*
21:44 hal_g2d.h*
21:44 hal_gdma.h*
```

```
$ll driver/chip/mt7933/inc/

21:44 ./
21:44 ../
21:44 common.h*
21:44 driver_api.h*
21:44 gdma.h
21:48 .git/
21:44 hal_asic_mpu.h
21:44 hal_asic_mpu_internal.h
21:44 hal_boot.h
21:44 hal_cache_internal.h*
21:44 hal_clk.h*
21:44 hal_dwt.h*
21:44 hal_ecc_internal.h*
21:44 hal_efuse_get.h
21:44 hal_eint_internal.h
21:44 hal_flash_cmd_macro.h
21:44 hal_flash_reg.h
21:44 hal_gcpu_internal.h
21:44 hal_gdma_internal.h
21:44 hal_gpio_internal.h
21:44 hal_gpt_internal.h
21:44 hal_i2c_master_internal.h
```

```
$ll driver/chip/mt7933/src

24 21:44 ./
24 21:44 ../
24 21:44 adc/
24 21:44 common/
24 21:44 efuse/
24 21:44 eint/
24 21:44 GCC/
24 21:48 .git/
24 21:44 hal_aes.c
24 21:44 hal_asic_mpu.c
24 21:44 hal_cache.c*
24 21:44 hal_cache_internal.c*
24 21:44 hal_clk.c*
24 21:44 hal_des.c
24 21:44 hal_dwt.c*
24 21:44 hal_ecc_api.c
24 21:44 hal_flash.c
24 21:44 hal_gdma.c
24 21:44 hal_gdma_internal.c
24 21:44 hal_gpio.c
24 21:44 hal_gpt.c
24 21:44 hal_gpt_internal.c
```

Source Folder Structure-3

- Directories
  - **doc/** : documents
  - **driver/** :source code of drivers
  - **kernel/** : source code of RTOS and system services
  - **middleware/** : source code of middleware
  - **project/** : user projects
    - <board> /apps/<project>/GCC
    - <board> /apps/<project>/GCC/Makefile
    - <board> /apps/<project>/GCC/feature.mk
  - **config/** : config files of chips, boards, and projects
    - chip/<ic_name>/chip.mk
    - board/<board_name>/board.mk
    - project/<board>/<project script>
- Files
  - **build.sh** : build command see the next page

Project makefile, the main file that trigger other makefile listed in chip.mk to generate libs and form the final bin file

Project's feature option are defined in this file

Compiler, CFLAGS, Middleware Module Path are defined in this file

Extra CFLAGS used for each board are defined in this file

Source Folder Structure-4

project/mt7933_hdk/apps/qfn_sdk_demo/GCC/feature.mk

```
IC_CONFIG                          = mt7933
BOARD_CONFIG                       = mt7933_hdk
# debug level: none, error, warning, and info
MTK_DEBUG_LEVEL                      = info
# 3 options with psram/flash or not, only 1 option is y,
MTK_MEMORY_WITH_PSRAM_FLASH        = y
MTK_MEMORY_WITHOUT_PSRAM           = n
MTK_MEMORY_WITHOUT_PSRAM_FLASH     = n
# System service debug feature for internal use
MTK_SUPPORT_HEAP_DEBUG             = n
MTK_HEAP_SIZE_GUARD_ENABLE         = n
MTK_OS_CPU_UTILIZATION_ENABLE      = y
MTK_XIP_ENABLE                     = y

#NVDM
MTK_NVDM_ENABLE                    = y
MTK_NVDM_NO_FLASH_ENABLE           = n

#CONSYS
MTK_MT7933_CONSYS_ENABLE           = y

#CONNSYS WF
MTK_MT7933_CONSYS_WIFI_ENABLE      = y

#WIFI features
MTK_WIFI_TGN_VERIFY_ENABLE         = n
MTK_WIFI_DIRECT_ENABLE             = n
MTK_WIFI_PROFILE_ENABLE            = y
MTK_SMTCN_V5_ENABLE                = n
MTK_CM4_WIFI_TASK_ENABLE           = n
MTK_WIFI_ROM_ENABLE                = n
MTK_WLAN_SERVICE_ENABLE            = y
MTK_ATED_ENABLE                    = n
```

project/mt7933_hdk/apps/qfn_sdk_demo/GCC/Makefile

```
ifeq ($(MTK_ATED_ENABLE), y)
ATED_DIR = middleware/MTK/connectivity/wlan_daemon/ated_ext
include $(SOURCE_DIR)/$(ATED_DIR)/GCC/module.mk
CFLAGS += -DMTK_ATED_ENABLE
endif

ifeq ($(MTK_WIFI_TEST_TOOL_ENABLE), y)
TEST_TOOL_DIR = middleware/MTK/connectivity/wlan_tool/wifi_tes
$(info TEST_TOOL_DIR $(TEST_TOOL_DIR))
include $(SOURCE_DIR)/$(TEST_TOOL_DIR)/module.mk
CFLAGS += -DMTK_WIFI_TEST_TOOL_ENABLE
endif

# HAL driver files
include $(SOURCE_DIR)/driver/chip/mt7933/module.mk
# EPT Config
#include $(SOURCE_DIR)/driver/board/mt7933_hdk/ept/module.mk

# RTOS source files
include $(SOURCE_DIR)/kernel/rtos/FreeRTOS/module.mk
# kernel service files
include $(SOURCE_DIR)/kernel/service/module.mk

# RTOS POSIX files
ifeq ($(MTK_POSIX_SUPPORT_ENABLE),y)
CFLAGS += -DMTK_POSIX_SUPPORT_ENABLE
include $(SOURCE_DIR)/kernel/rtos/FreeRTOS-ext/FreeRTOS-Labs/S
endif

# NVDM files
ifeq ($(MTK_NVDM_ENABLE),y)
include $(SOURCE_DIR)/middleware/MTK/nvdm/module.mk
endif
```

Source Folder Structure-5

config/chip/mt7933/chip.mk*

```
##
## MTK_NVDM_ENABLE
## Brief:        This option is to enable NVDM feature.
## Usage:        Enable the feature by configuring it as y.
## Path:         middleware/MTK/nvdm
## Dependency:   Flash driver must be enabled.
## Notice:       None
## Relative doc:None
##
ifeq ($(MTK_NVDM_ENABLE),y)
  CFLAGS += -DMTK_NVDM_ENABLE
endif


##
## MTK_SECURE_BOOT_ENABLE
## Brief:        This option is to enable secure boot feature in bootloader.
## Usage:        Enable the feature by configuring it as y.
## Path:         middleware/MTK/sboot
## Dependency:   libmbedtls
## Notice:       None
## Relative doc:None
##
ifeq ($(MTK_SECURE_BOOT_ENABLE),y)
  CFLAGS += -DMTK_SECURE_BOOT_ENABLE
  CFLAGS += -I$(SDK_PATH)/tools/mcuboot/boot/bootutil/include/bootutil
endif
```

Source Folder Structure-6

```
project/mt7686_hdk/apps/iot_sdk_demo/inc/hal_feature_config.h*
```

```
project/mt7933_hdk/apps/qfn_sdk_demo/GCC/hal_feature.mk
```

```
#HAL Module Features

MTK_HAL_ADC_MODULE_ENABLE              = y
MTK_HAL_AES_MODULE_ENABLE              = y
MTK_HAL_CACHE_MODULE_ENABLE            = y
MTK_HAL_CLOCK_MODULE_ENABLE            = y
MTK_HAL_DES_MODULE_ENABLE              = y
MTK_HAL_EFUSE_MODULE_ENABLE            = y
MTK_HAL_EINT_MODULE_ENABLE             = y
MTK_HAL_FLASH_MODULE_ENABLE            = y
MTK_HAL_GDMA_MODULE_ENABLE             = y
MTK_HAL_GPIO_MODULE_ENABLE             = y
MTK_HAL_GPT_MODULE_ENABLE              = y
MTK_HAL_GCPU_MODULE_ENABLE             = y
MTK_HAL_I2C_MASTER_MODULE_ENABLE       = y
MTK_HAL_NVIC_MODULE_ENABLE             = y
MTK_HAL_IRRX_MODULE_ENABLE             = n
MTK_HAL_KEYPAD_MODULE_ENABLE           = n
MTK_HAL_MD5_MODULE_ENABLE              = y
MTK_HAL_MPU_MODULE_ENABLE              = y
MTK_HAL_ASIC_MPU_MODULE_ENABLE         = y
MTK_HAL_PMU_MODULE_ENABLE              = y
MTK_HAL_PSRAM_MODULE_ENABLE            = y
MTK_PSRAM_TYPE_VALUE                   = y
MTK_HAL_PWM_MODULE_ENABLE              = y
MTK_HAL_RTC_MODULE_ENABLE              = y
MTK_HAL_SPI_MASTER_MODULE_ENABLE       = y
MTK_HAL_SPI_SLAVE_MODULE_ENABLE        = y
MTK_HAL_SDIO_MODULE_ENABLE             = y
MTK_HAL_SDIO_SLAVE_MODULE_ENABLE       = y
MTK_HAL_SHA_MODULE_ENABLE              = y
MTK_HAL_SLEEP_MANAGER_MODULE_ENABLE    = y
MTK_HAL_TRNG_MODULE_ENABLE             = y
```

Build Image-1

For BGA MT7933CT

```
$ll project/mt7933_hdk/apps/

19:50 ./
21:45 ../
21:45 bga_sdk_demo/
21:45 bootloader/
21:45 qfn_sdk_demo/
```

For QFN MT7931AN

```
==========================================================
Build Project
==========================================================
Usage: ./build.sh <board> <project> [bl|clean] <argument>
...
Argument:
        -f=<feature makefile> or --feature=<feature makefile>
            Replace feature.mk with other makefile. For example,
            the feature example.mk is under project folder, -
f=feature example.mk
            will replace feature.mk with feature_example.mk.

        -o=<make option> or --option=<make option>
            Assign additional make option. For example,
            to compile module sequentially, use -o=-j1.
            to turn on specific feature in feature makefile, use -
o=<feature_name>=y
            to assign more than one options, use -o=<option_1> -
o=<option 2>.

==========================================================
List Available Example Projects
==========================================================
Usage: ./build.sh list
```

Build Image-2

- **3.1.1. Build the project.**

To build a specific project, simply run the following command.

```
./build.sh <board> <project>
```

The output files are then put in the <sdk_root>/out/<board>/<project> folder.

For example, to build a project in the MT7933 HDK, run the following build command:

```
./build.sh mt7933_hdk qfn_sdk_demo
```

The standard output in the terminal window is as follows:

```
$./build.sh mt7933_hdk qfn_sdk_demo
UE BUILD BOARD: mt7933_hdk
UE BUILD PROJECT: qfn_sdk_demo
platform=MSYS_NT-10.0-16299
FEATURE = feature.mk
BL_FEATURE = bl_feature_XXXX.mk

Build bootloader...
...
```

The output files are then put in the <sdk_root>/out/mt7933_hdk/qfn_sdk_demo/ folder.

Build Image-3

3.1.3. Build the project with the "bl" option

By default, the pre-built bootloader image file is copied to the <sdk_root>/out/<board>/<project>/ folder after the project is built. The main purpose for the bootloader image is to download the Flash Tool.

Apply the "bl" option to rebuild the bootloader and use the generated bootloader image file instead of the pre-built one, as shown below.

```
./build.sh <board> <project> bl
```

To build the project on the MT7933 HDK:

```
cd <sdk_root>
./build.sh mt7933_hdk qfn_sdk_demo bl
```

The output image file of the project and the bootloader, along with the merged image file `flash.bin`, will be placed under <sdk_root>/out/mt7933_hdk/iot_sdk_demo folder.

Build Image-4

- Binary files
  - `project image` — mt7931an_xip_qfn_bw.bin.
  - `bootloader image` — mt7931an_bootloader-xip.sgn.
  - other Wifi or BT images copied from prebuild folder.
- `elf file` — contains information about the executable, object code, shared libraries and core dumps.
- `map file` — contains the link information of the project libraries.
- `lib folder` — contains module libraries.
- `log folder` — contains build log including build information, timestamp and error messages.
- `obj folder` — contains object and dependency files.

Build Image-5

```
./build.sh mt7933_hdk qfn_sdk_demo
```

```
out/mt7933_hdk/qfn_sdk_demo/

20:07 ./
11:48 ../
21:59 autogen/
20:07 BT_RAM_CODE_MT7933_1_1_hdr.bin*
20:07 BT_RAM_CODE_MT7933_2_1_hdr.bin*
20:07 lib/
20:07 log/
21:59 mt7931an_bootloader-ram.bin*
21:59 mt7931an_bootloader-ram.elf*
21:59 mt7931an_bootloader-ram.hex
21:59 mt7931an_bootloader-ram.lnk
21:59 mt7931an_bootloader-ram.map
21:59 mt7931an_bootloader-ram.sgn
21:59 mt7931an_bootloader_scatter.ini
21:59 mt7931an_bootloader-xip.bin*
21:59 mt7931an_bootloader-xip.elf*
21:59 mt7931an_bootloader-xip.hex
21:59 mt7931an_bootloader-xip.lnk
21:59 mt7931an_bootloader-xip.map
21:59 mt7931an_bootloader-xip.sgn
20:07 mt7931an_xip_qfn_bw.bin*
21:59 mt7931an_xip_qfn_bw.cmm
20:07 mt7931an_xip_qfn_bw.elf*
20:07 mt7931an_xip_qfn_bw.elf.map
20:07 mt7931an_xip_qfn_bw.elf.opts
20:07 mt7931an_xip_qfn_bw_scatter.ini
21:59 mt7933_bootloader-ram.cmm
21:59 mt7933_bootloader-xip.cmm
20:07 mt7933_patch_e1_hdr.bin*
20:07 obj/
20:07 WIFI_RAM_CODE_iemi.bin*
20:07 WIFI_RAM_CODE_log.bin*
20:07 WIFI_RAM_CODE_MT7933_APSOC.bin*
```

```
ll out/mt7933_hdk/qfn_sdk_demo/log/

20:07 ./
20:07 ../
20:07 build.log
20:07 build_time.log
20:07 copy_firmware_opts.log
20:07 err.log
```

Bootloader image

RTOS image

Scatter file

Wifi images

Build Image-6

3.1.2. Clean the out folder.

The build script <sdk_root>/build.sh provides options for removing the generated output files, as shown below.

Clean the <sdk_root>/out folder.

```
./build.sh clean
```

Clean the <sdk_root>/out/<board> folder.

```
./build.sh <board> clean
```

Clean the <sdk_root>/out/<board>/<projecct> folder.

```
./build.sh <board> <project> clean
```

The output folder is defined under variable BUILD_DIR in the Makefile in <sdk_root>/project/mt7933_hdk/apps/qfn_sdk_demo/GCC:

```
BUILD_DIR = $(PWD)/Build
PROJ_NAME = $(shell basename $(dir $(PWD)))
```

A project image earbuds_ref_design.bin is generated under <sdk_root>/project/mt7933_hdk/apps/iot_sdk_demo/GCC/Build.

Creating a Project

```
cp -r project/mt7933_hdk/apps/qfn_sdk_demo/ project/mt7933_hdk/apps/my_qfn_sdk
```

```
                    hadron-mcuiot-ring-3rd]$./build.sh mt7933_hdk my_qfn_sdk
UE BUILD BOARD: mt7933_hdk
UE BUILD PROJECT: my_qfn_sdk
platform=Linux
FEATURE = feature.mk
TEST_TOOL_DIR middleware/MTK/connectivity/wlan_tool/wifi_test_tool/wifitesttool
build WIFI TEST TOOL in middleware/MTK/connectivity/wlan_tool/wifi_test_tool/wifitesttool
build minisupp
make: Entering directory `/proj/        /projects/hadron-mcuiot-ring-3rd/project/mt7933_hdk/apps/my
trigger by build.sh, skip cleanlog
CC      mt7931an_xip_qfn_bw - bsp_gpio_ept_config.o
CC      mt7931an_xip_qfn_bw - debug.o
CC      mt7931an_xip_qfn_bw - wlan_lib.o
CC      mt7931an_xip_qfn_bw - dump.o
CC      mt7931an_xip_qfn_bw - wlan_oid.o
CC      mt7931an_xip_qfn_bw - assoc.o
CC      mt7931an_xip_qfn_bw - ais_fsm.o
CC      mt7931an_xip_qfn_bw - auth.o
CC      mt7931an_xip_qfn_bw - bss.o
CC      mt7931an_xip_qfn_bw - cnm.o
CC      mt7931an_xip_qfn_bw - cnm_mem.o
CC      mt7931an_xip_qfn_bw - cnm_timer.o
CC      mt7931an_xip_qfn_bw - hem_mbox.o
CC      mt7931an_xip_qfn_bw - he_ie.o
CC      mt7931an_xip_qfn_bw - he_rlm.o
CC      mt7931an_xip_qfn_bw - mib.o
CC      mt7931an_xip_qfn_bw - privacy.o
CC      mt7931an_xip_qfn_bw - gosmap.o
```

```
ll out/mt7933_hdk/my_qfn_sdk/

25 20:49 ./
25 20:46 ../
25 20:46 autogen/
25 20:49 BT_RAM_CODE_MT7933_1_1_hdr.bin*
25 20:49 BT_RAM_CODE_MT7933_2_1_hdr.bin*
25 20:49 lib/
25 20:49 log/
25 20:49 mt7931an_xip_qfn_bw.bin*
25 20:49 mt7931an_xip_qfn_bw.cmm
25 20:49 mt7931an_xip_qfn_bw.elf*
25 20:49 mt7931an_xip_qfn_bw.elf.map
25 20:49 mt7931an_xip_qfn_bw.elf.opts
25 20:49 mt7931an_xip_qfn_bw_scatter.ini
25 20:49 mt7933_patch_e1_hdr.bin*
25 20:49 obj/
25 20:49 WIFI_RAM_CODE_iemi.bin*
25 20:49 WIFI_RAM_CODE_log.bin*
25 20:49 WIFI_RAM_CODE_MT7933_APSOC.bin*
```

Adding a Module to Middleware

```
$tree middleware/third_party/myModule/
middleware/third_party/myModule/
├── inc
│   └── myModule.h
├── Makefile
├── module.mk
└── src
    └── myModule.c
```

Makefile

```
120
121 #include mymodule
122 include $(SOURCE_DIR)/middleware/third_party/myModule/module.mk
```

myModule.c

```
1  #include <string.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <ctype.h>
5
6  #include "myModule.h"
7
8  void myMoulde_Test(void)
9  {
10     printf("\n******************* MYMODULE TEST *******************\n\n");
11 }
```

```
[T: 148 M: dhcpd C: info F: dhcpd_start L: 671]: DHCPD dhcpd_start [0][0]
[T: 148 M: dhcpd C: info F: dhcpd_start L: 682]: DHCPD preparing

******************* NNNNNNNNNNNNNNNN: 2222 *******************


******************* NNNNNNNNNNNNNNNN: 3333 *******************


******************* MYMODULE TEST *******************
```

MEDIATEK

everyday genius