Airoha IoT SDK for RTOS Get Started Guide

Version: 4.11

Release date: 10 January 2019

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s). Airoha cannot grant you permission for any material that is owned by third parties. You may only use or reproduce this document if you have agreed to and been bound by the applicable license agreement with Airoha (“License Agreement”) and been granted explicit permission within the License Agreement (“Permitted User”). If you are not a Permitted User, please cease any access or use of this document immediately. Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited. THIS DOCUMENT IS PROVIDED ON AN “AS-IS” BASIS ONLY. AIROHA EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF ANY KIND AND SHALL IN NO EVENT BE LIABLE FOR ANY CLAIMS RELATING TO OR ARISING OUT OF THIS DOCUMENT OR ANY USE OR INABILITY TO USE THEREOF. Specifications contained herein are subject to change without notice.

Document Revision History

Revision	Date	Description
1.0	24 March 2016	Initial release
2.0	17 May 2016	• Flash tool and building the project content is moved to the corresponding documents. • Added the support for Keil IDE.
3.0	30 June 2016	• Added the support of IAR embedded workbench IDE. • More details on the SDK features and usage is included.
4.0	2 September 2016	• Updated Section 2 to describe how to use serial emulator software, such as TeraTerm with LinkIt 7687 HDK and LinkIt 2523 HDK.
4.1	4 November 2016	• Updated Section 2.4, building the project using the SDK.
4.2	13 January 2017	• Added information about 2533D • Added note for GCC/KEIL/IAR debugging limitation in sleep mode. • Updated file system additional descriptions. • Added support for MT76x7 Flash Tool in Linux.
4.3	5 May 2017	• Added support for MT7682/MT5932. • Updated OpenOCD configuration for debugging on LinkIt 7687 HDK. • Added support for MT7682 HDKs.
4.5	30 June 2017	• Added support for MT7686 HDK.
4.6	15 September 2017	• Added chapter for disabling automatic driver installation on Windows OS
4.7	26 October 2017	• Added jumper setting in section 2.4.4.2, "Using the MT7682 HDK as a removable storage device" and 2.5.4.2, "Using the LinkIt 7686 HDK as a removable storage device" • Corrected the mbed UART port number in section 2.4.3.1, "Install MK20 USB Driver" and 2.5.3.1, "Install MK20 USB Driver" • Added IAR environment configuration for MT7686/MT7682
4.8	18 May 2018	• Added debugging configuration for MT7682/MT7686 HDK.
4.9	18 November 2018	• Added support for AB155x EVK
4.10	11 December 2018	• Added co-build system description for AB155x EVK • Refined the “Building the project using the SDK” section
4.11	10 January 2019	• Refined the “Building the project using the SDK” section

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Table of Contents

Document Revision History i
Table of Contents ii
Lists of Tables and Figures iv
1. Overview 1
1.1. Architecture of the platform 1
1.2. Supported key components 7
1.2.1. Wi-Fi 7
1.2.2. Network 7
1.2.3. Bluetooth and Bluetooth Low Energy 8
1.2.4. Sensor subsystem 9
1.2.5. GNSS 9
1.2.6. FOTA 10
1.2.7. Peripheral drivers 10
1.2.8. Battery management 11
1.2.9. Advanced features and components 12
1.3. Folder structure 12
1.4. Project source structure 15
2. Getting Started Using GCC 16
2.1. Environment 16
2.2. Developing on LinkIt 7687 HDK 16
2.2.1. Configuring the LinkIt 7687 HDK 16
2.2.2. Installing the LinkIt 7687 HDK drivers on Microsoft Windows 18
2.2.3. Installing MT76x7 Flash Tool for LinkIt 7687 HDK 18
2.2.4. Flashing the image to LinkIt 7687 HDK 20
2.2.5. Running the project on LinkIt 7687 HDK 22
2.2.6. Debugging with the LinkIt 7687 HDK from Microsoft Windows 24
2.3. Developing on LinkIt 2523 HDK 27
2.3.1. Configuring the LinkIt 2523 HDK 27
2.3.2. Installing IoT Flash Tool for LinkIt 2523 HDK 27
2.3.3. Installing the LinkIt 2523 HDK drivers on Microsoft Windows 28
2.3.4. Flashing the image to LinkIt 2523 HDK 32
2.3.5. Running the project on LinkIt 2523 HDK 35
2.3.6. Debugging with the LinkIt 2523 HDK from Microsoft Windows 37
2.4. Developing on MT7682 HDK 39
2.4.1. Configuring the MT7682 HDK 39
2.4.2. Installing MT7682 Flash Tool for MT7682 HDK 40
2.4.3. Installing the MT7682 HDK drivers on Microsoft Windows 40
2.4.4. Flashing the image to MT7682 HDK 41
2.4.5. Running the project on MT7682 HDK 44
2.4.6. Debugging with the MT7682 HDK from Microsoft Windows 46
2.5. Developing on LinkIt 7686 HDK 48
2.5.1. Configuring the LinkIt 7686 HDK 48
2.5.2. Installing MT7686 Flash Tool for LinkIt 7686 HDK 49
2.5.3. Installing the LinkIt 7686 HDK drivers on Microsoft Windows 49
2.5.4. Flashing the image to LinkIt 7686 HDK 50
2.5.5. Running the project on LinkIt 7686 HDK 52
2.5.6. Debugging with the LinkIt 7686 HDK from Microsoft Windows 53
2.6. Developing on LinkIt 7697 HDK 55

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.7. Developing on AB155x EVK 56
2.7.1. Configuring the AB155x EVK 56
2.7.2. Installing AB155x Flash Tool for AB155x EVK 56
2.7.3. Installing the AB155x EVK drivers on Microsoft Windows 57
2.7.4. Flashing the image to AB155x EVK 57
2.7.5. Running the project on AB155x EVK 58
2.7.6. Debugging with the AB155x EVK from Microsoft Windows 60
2.8. Building the project using the SDK 61
2.8.1. Installing the SDK build environment on Linux 61
2.8.2. Methods to build a project 62
2.9. Create your own project 67
2.9.1. Using an existing project 67
2.9.2. Removing a module 68
2.9.3. Add the source and header files 68
3. Getting Started Using Keil µVision IDE 70
3.1. Environment 70
3.2. Installing the Microsoft Windows version of Keil 70
3.2.1. Installing the µVision IDE 70
3.2.2. Installing the Keil package for the SDK 70
3.3. Build the project 71
3.4. Download and run the project 73
3.4.1. Downloading the project binary using the Keil µVision IDE 74
3.4.2. Downloading the project binary using Flash Tool or using HDK as a removable storage device 77
3.4.3. Running the project 78
3.5. Debug configuration 79
4. Getting Started Using IAR 83
4.1. Environment 83
4.2. Installing the Microsoft Windows version of IAR 83
4.3. Build the project 83
4.4. Download and run the project 85
4.4.1. Download the project 86
4.4.2. Running the project 93
4.5. Debug configuration 94
5. Appendix A: Acronyms and Abbreviations 96
6. Appendix B: Disabling Automatic Driver Installation on Windows OS 97

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Lists of Tables and Figures

Table 1. HAL features on different chipsets 2
Table 2. Middleware features on different chipsets 4
Table 3. Wi-Fi station features 7
Table 4. Wi-Fi AP features 7
Table 5. Supported network protocols 7
Table 6. Bluetooth/Bluetooth Low Energy features 8
Table 7. Sensor subsystem features 9
Table 8. GNSS features 9
Table 9. FOTA features 10
Table 10. Supported peripheral drivers 10
Table 11. Battery management features 12
Table 12. Advanced features and components 12
Table 13. Recommended build environment 61
Table 14. Acronyms and Abbreviations 96

Figure 1. Architecture layout of the platform 1
Figure 2. Folder structure 13
Figure 3. Project folder structure 15
Figure 4. Top view of the LinkIt 7687 HDK 17
Figure 5. COM port associated with the LinkIt 7687 HDK 18
Figure 6. Executable file in the MT76x7 Flash Tool's destination folder 19
Figure 7. Setup the COM port and baud rate 20
Figure 8. Download an image to the device storage 21
Figure 9. Downloading the image is successfully complete 21
Figure 10. New removable storage detected 22
Figure 11. Serial port setting configuration 23
Figure 12. Connect to mbed Serial Port 24
Figure 13. USB connectors on the LinkIt 2523 development board 27
Figure 14. Unknown device on Windows 8 or later 29
Figure 15. Browse my computer for driver software 30
Figure 16. MTK USB Port on Windows 8 or later 30
Figure 17. Debug and modem ports for MTK USB port 31
Figure 18. MTK USB driver version in MTK USB Debug Port Properties 32
Figure 19. mbed serial port 32
Figure 20. Download the firmware to a target device using USB connection 34
Figure 21. LinkIt 2523 HDK connected as removable disk storage 35
Figure 22. Serial port setup 36
Figure 23. Connect to mbed Serial Port 37
Figure 24. Jumpers and connectors on the MT7682 HDK 39
Figure 25. Installing the UART driver 41

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Figure 26. Download the firmware to a target device using USB connection 43
Figure 27. MT7682 removable storage detected....44
Figure 28. Connect to mbed serial port 44
Figure 29. Front view of the LinkIt 7686 HDK 48
Figure 30. Installing the UART driver 50
Figure 31. Download the firmware to a target device using USB connection 51
Figure 32. MT7686 removable storage detected....52
Figure 33. Connecting to mbed serial port 53
Figure 34. Front view of the AB155x EVK....56
Figure 35. Download the firmware to a target device using USB connection 58
Figure 36. Serial Port Button....59
Figure 37. Serial Port Configuration Dialog....59
Figure 38. Start Button....59
Figure 39. Modify the `Makefile` under the GCC folder of `my_project` 67
Figure 40. Project source and header files under the project folder....69
Figure 41. Installing Airoha IoT SDK with Keil IDE 71
Figure 42. Importing the preconfigured project 71
Figure 43. Project file for Keil....72
Figure 44. Building the project....72
Figure 45. Project build result....73
Figure 46. Set download configuration on µVision IDE....74
Figure 47. Select configuration file 74
Figure 48. Select flash initialization file (flash.ini)....75
Figure 49. Select the CMSIS-DAP Debugger interface 75
Figure 50. Select device flash programming algorithm....76
Figure 51. Add device flash programming algorithm....76
Figure 52. Setup the parameters for downloading the project on LinkIt 2523 HDK....77
Figure 53. Download the project binary 77
Figure 54. User defined configurations....78
Figure 55. Configure the debug settings....79
Figure 56. Setting the debug parameters on LinkIt 7687 HDK....80
Figure 57. Setting the debug parameters on LinkIt 2523 HDK....81
Figure 58. Debugging a project on Keil IDE 82
Figure 59. Importing the preconfigured project 84
Figure 60. Adding an existing project....84
Figure 61. Rebuild the project....85
Figure 62. Project build result on IAR Embedded Workbench IDE 85
Figure 63. Set download configuration on IAR IDE 86
Figure 64. Select debug driver and `ddf` file 87
Figure 65. Open the MT2523.ddf file....88
Figure 66. Configure the download settings....89
Figure 67. Select the board file 90
Figure 68. Set extra download Image 91
Figure 69. JTAG/SWD settings....92

© 2015 - 2019 Airoha Technology Corp.
This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Figure 70. Download the project binary 93
Figure 71. Debugging the project without downloading 94
Figure 72. Debugging a project on IAR.....95
Figure 73. Disabling automatic driver updates on Windows 7 OS.....97
Figure 74. Disabling automatic driver updates on Windows 8 and 10 98

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

1. Overview

Airoha IoT SDK provides the software and tools for your application development on LinkIt 7687, 7697, 2523, 7682, 7686, 5932 HDKs and 155x EVK. The SDK includes drivers for hardware abstraction layer, peripherals, connectivity, such as Wi-Fi, Bluetooth/Bluetooth Low Energy, GNSS, sensor subsystem, lightweight IP (lwIP) and other third party features. It also provides battery management, Firmware update Over-The-Air (FOTA) and FreeRTOS.

This get started guide provides quick steps on how to use the SDK and its supported features on three different environments; default GCC, Keil µvision IDE and IAR embedded workbench IDE.

1.1. Architecture of the platform

The three-layer architecture of the platform including **BSP**, **Middleware** and **Application** with underlying components is shown in Figure 1.

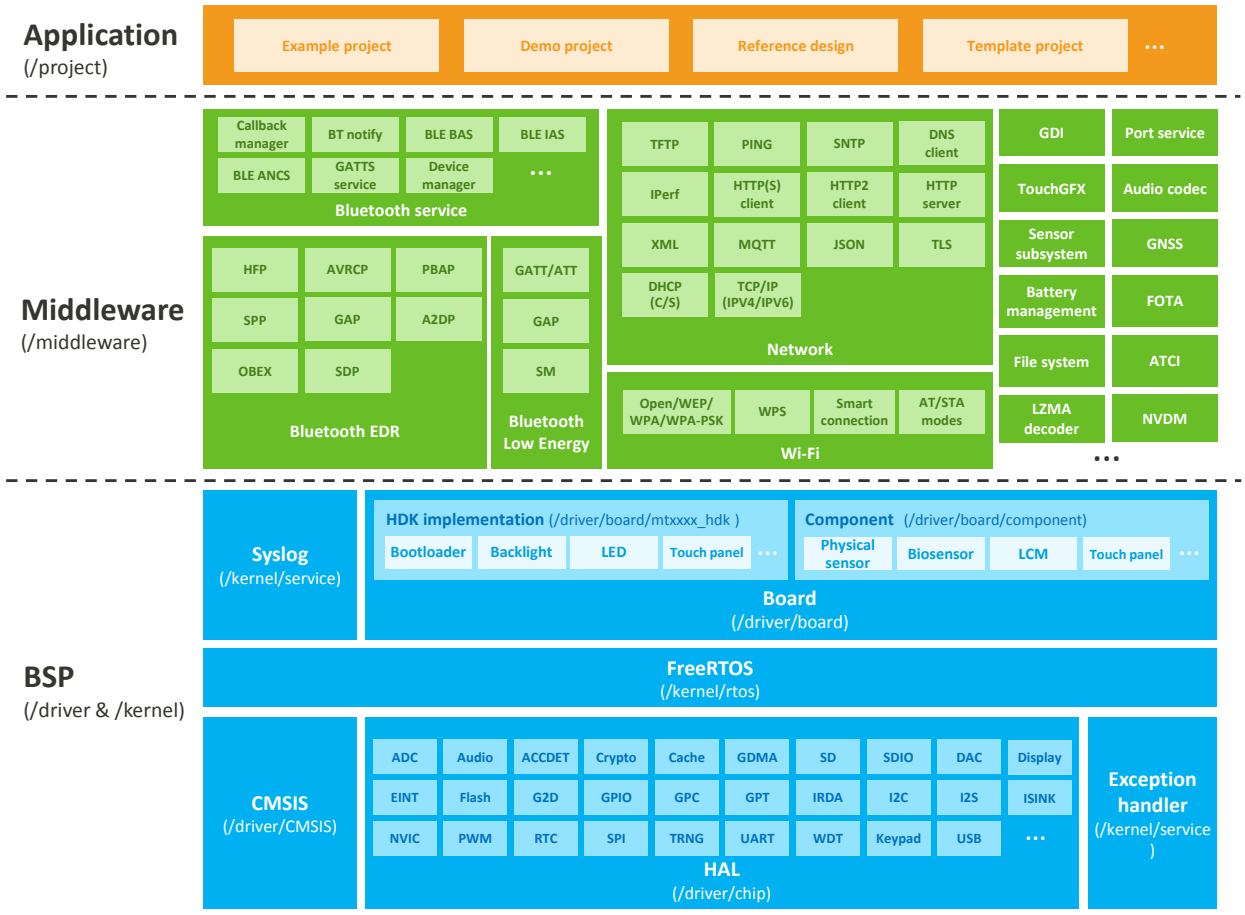

Figure 1. Architecture layout of the platform

A brief description of the layers is provided below:

- **BSP**
  - Hardware drivers. Provide peripheral drivers for the platform, such as ADC, I2S, I2C, SPI, RTC, GPIO, UART, Flash, Security Engine, TRNG, GDMA, PWM, WDT and IRDA TX/RX.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

- Hardware Abstraction Layer (HAL). Provides the driver Application Programming Interface (API) encapsulating the low-level functions of peripheral drivers for the operating system (OS), **Middleware** features and **Application**.
- The hardware components located at `<sdk_root>\driver\board\component` are used by the HDK (`<sdk_root>\driver\board\mtxxxx_hdk`). For example, the LCM drivers SH1107 and ST7789H2 located under `<sdk_root>\driver\board\component\lcm` folder will be available when you select the ST7789H2 LCM on LinkIt 2523 HDK but the GPIO pins need to be configured at `<sdk_root>\driver\board\mt2523_hdk\lcm` and also the source files should be included under the component folder.
- FreeRTOS. An OS with the open source software for **Middleware** components and **Application**.
- Syslog. This module implements system logging for development and debugging.

- **Middleware**
  - Wi-Fi. Provides OS dependent function calls, including Wi-Fi APIs that control the bridge supplicant and network processor messages.
  - Network. Provides OS dependent features, such as IPv4, Hyper-Text Transfer Protocol (HTTP) client and the Simple Network Time Protocol (SNTP).
  - Bluetooth/Bluetooth Low Energy. Provides stack and protocol-layer access profiles for data transferring and management control, such as Generic Access Profile (GAP), Serial Port Profile (SPP), Generic Attribute Profile (GATT) and Security Manager (SM).
  - Sensor subsystem. Software framework to interact with sensor drivers and fusion algorithms, including buffer and flow control.
  - FOTA. Provides a mechanism to update the firmware.
  - GNSS. Provides APIs to control the onboard GNSS system.
  - Battery management. Provides the charging flow control and precise information on battery.
  - File system. Provides APIs to control data storage and retrieval in a file system.
  - Other features. Non-Volatile Data Management (NVDM), Extensible Markup Language (XML), JavaScript Object Notation (JSON) and other features that are dependent on **HAL** and **FreeRTOS**. The Airoha IoT SDK also supports AT command interface (ATCI) as an advanced feature.
- **Application**
  - Pre-configured projects using **Middleware** components, such as Wi-Fi station and smart connection.

The application layer enables running the projects that are based on **Middleware**, **FreeRTOS** and **HAL** layers. These layers provide rich features for application development, such as **Middleware** provides the network features, Wi-Fi features, and the OS provides the underlying real-time operating system.

The supported HAL features on different chipsets are listed in Table 1.

To use the HAL features, enable the compile options of the corresponding modules.

1) Open the header file `hal_feature_config.h`, located under `inc` folder of each example project.
2) Edit and define the compile options as needed.
3) Include the corresponding module header files, located at `<sdk_root>\driver\chip\inc`, in the project source files.

Table 1. HAL features on different chipsets

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Feature	MT7687F	MT7697/ MT7697D	MT2523D/ MT2523G	MT2533D	MT7682/ MT7686/ MT5932	AB155x	Compile option
ADC	✓	✓	✓	✓	✓	✓	HAL_ADC_MODULE_ENABLED
ACCDET			✓	✓			HAL_ACCDET_MODULE_EANBLED
AUDIO			✓	✓		✓	HAL_AUDIO_MODULE_ENABLED
Cache	✓	✓	✓	✓	✓	✓	HAL_CACHE_MODULE_ENABLED
Crypto	✓	✓	✓	✓	✓	✓	HAL_AES_MODULE_ENABLED HAL_DES_MODULE_ENABLED HAL_MD5_MODULE_ENABLED HAL_SHA_MODULE_ENABLED
Clock			✓	✓	✓	✓	HAL_CLOCK_MODULE_ENABLED
CAP-TOUCH						✓	HAL_CAPTOUCH_MODULE_ENABLED
DAC			✓	✓			HAL_DAC_MODULE_ENABLED
DVFS			✓	✓			HAL_DVFS_MODULE_ENABLED
EINT	✓	✓	✓	✓	✓	✓	HAL_EINT_MODULE_ENABLED
Flash	✓	✓	✓	✓	✓	✓	HAL_FLASH_MODULE_ENABLED
GDMA	✓	✓	✓	✓	✓	✓	HAL_GDMA_MODULE_ENABLED
GPC	✓	✓	✓	✓			HAL_GPC_MODULE_ENABLED
GPIO	✓	✓	✓	✓	✓	✓	HAL_GPIO_MODULE_ENABLED
GPT	✓	✓	✓	✓	✓	✓	HAL_GPT_MODULE_ENABLED
I2C Master	✓	✓	✓	✓	✓	✓	HAL_I2C_MASTER_MODULE_ENABLED
I2S	✓	✓	✓	✓	✓		HAL_I2S_MODULE_ENABLED
ISINK			✓	✓			HAL_ISINK_MODULE_ENABLED
IRDA	✓	✓					HAL_IRRX_MODULE_ENABLED HAL_IRTX_MODULE_ENABLED
Keypad	✓	✓		✓			HAL_KEYPAD_MODULE_ENABLED

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Feature	MT7687F	MT7697/ MT7697D	MT2523D/ MT2523G	MT2533D	MT7682/ MT7686/ MT5932	AB155x	Compile option
DISPLAY_COLOR			✓	✓			HAL_DISPLAY_COLOR_MODULE_ENABLED
DISPLAY_DSI			✓	✓			HAL_DISPLAY_DSI_MODULE_ENABLED
DISPLAY_LCD			✓	✓			HAL_DISPLAY_LCD_MODULE_ENABLED
DISPLAY_PWM			✓	✓			HAL_DISPLAY_PWM_MODULE_ENABLED
MPU			✓	✓	✓		HAL_MPU_MODULE_ENABLED
NVIC	✓	✓	✓	✓	✓	✓	HAL_NVIC_MODULE_ENABLED
PWM	✓	✓	✓	✓	✓	✓	HAL_PWM_MODULE_ENABLED
RTC	✓	✓	✓	✓	✓	✓	HAL_RTC_MODULE_ENABLED
SPI Master	✓	✓	✓	✓	✓	✓	HAL_SPI_MASTER_MODULE_ENABLED
SPI Slave			✓	✓	✓	✓	HAL_SPI_SLAVE_MODULE_ENABLED
SD			✓	✓	✓	✓	HAL_SD_MODULE_ENABLED
SDIO			✓	✓	✓		HAL_SDIO_MODULE_ENABLED
SDIO Slave					✓		HAL_SDIO_SLAVE_MODULE_ENABLED
Sleep Manager			✓	✓	✓	✓	HAL_SLEEP_MANAGER_MODULE_ENABLED
TRNG	✓	✓	✓	✓	✓	✓	HAL_TRNG_MODULE_ENABLED
UART	✓	✓	✓	✓	✓	✓	HAL_UART_MODULE_ENABLED
USB			✓	✓		✓	HAL_USB_MODULE_ENABLED
WDT	✓	✓	✓	✓	✓	✓	HAL_WDT_MODULE_ENABLED

The supported middleware features on different chipsets are listed in Table 2. There is a `readme.txt` under the root directory of each middleware module. It contains the information about the module dependency, feature options, notes and brief introduction. To learn more about the usage of the middleware modules, refer to the `readme.txt` file under each module path.

Table 2. Middleware features on different chipsets

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Feature	MT7687F	MT7697/ MT7697D	MT2523D/ MT2523G	MT2533D	MT7682/ MT7686/ MT5932	AB155x	Module path
Bluetooth stack			✓	✓		✓	middleware\MTK\bluetooth
Bluetooth profile			✓	✓		✓	middleware\MTK\bluetooth
Bluetooth Low Energy stack		✓	✓	✓		✓	middleware\MTK\bluetooth
Bluetooth Low Energy profile		✓	✓	✓		✓	middleware\MTK\bluetooth
Wi-Fi	✓	✓			✓		middleware\MTK\minisupp driver\board\mt76x7_hdk\wifi
lwIP	✓	✓			✓		middleware\third_party\lwip
HTTP server	✓	✓			✓		middleware\third_party\httpd
HTTP(S) client	✓	✓			✓		middleware\third_party\httpclient
HTTP2 client	✓	✓			✓		middleware\third_party\nghttp2
IPerf	✓	✓			✓		middleware\third_party\iperf3
MQTT	✓	✓			✓		middleware\third_party\mqtt
JSON	✓	✓			✓		middleware\third_party\cjson
Xml	✓	✓			✓		middleware\third_party\xml
TLS	✓	✓			✓	✓	middleware\third_party\mbedtls
DHCP client\server	✓	✓			✓		middleware\third_party\dhcpd
DNS client	✓	✓			✓		middleware\third_party\lwip
SNTP	✓	✓			✓		middleware\third_party\sntp
PING	✓	✓			✓		middleware\third_party\ping
TFTP	✓	✓			✓		middleware\MTK\tftp
GNSS			✓				middleware\MTK\gnss

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Feature	MT7687F	MT7697/ MT7697D	MT2523D/ MT2523G	MT2533D	MT7682/ MT7686/ MT5932	AB155x	Module path
NVDM	✓	✓	✓	✓	✓	✓	middleware\MTK\nvdm
ATCI			✓	✓	✓	✓	middleware\MTK\atci
File system	✓	✓	✓	✓		✓	middleware\third_party\fatfs
Battery management			✓	✓		✓	middleware\MTK\battery_management
FOTA	✓	✓	✓	✓	✓	✓	middleware\MTK\fota
Sensor subsystem			✓			✓	middleware\MTK\sensor_subsys
LZMA decoder						✓	middleware\third_party\lzma_decoder

Note, the file system does not work without SD/eMMC.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

1.2. Supported key components

The platform offers rich connectivity options, such as Wi-Fi, network, Bluetooth, GNSS, peripheral drivers and other advanced components. This section introduces each of these components.

1.2.1. Wi-Fi

Wi-Fi is a key feature included in the platform. It supports both station and access point (AP) modes. More information on the Wi-Fi APIs can be found in 7687/7697 SDK API reference manual and Wi-Fi developer's guide under `<sdk_root>\doc`. More details about the module can be found in `<sdk_root>\middleware\MTK\minisupp\readme.txt`.

The detailed feature list of the station mode can be found in Table 3, where the items are released as a library.

Table 3. Wi-Fi station features

Item	Features
Standard	802.11 b/g/n Station (STA)
Channel	Channel 1 to 13
Personal Security	Open, WEP-Open, WPA, WPA2
Enterprise Security	N/A
WPS	Enrollee (PBC/PIN)
Advanced	AMPDU, RX-Filter, DTIM

The detailed feature list of the AP mode can be found in Table 4, where the items are released as a library.

Table 4. Wi-Fi AP features

Item	Features
Standard	802.11 b/g/n Soft AP
Channel	Channels 1 to 13
Personal Security	Open, WEP-Open, WPA, WPA2
Support Clients	• 9 STAs (AP only mode) on LinkIt 7687 or 7697 HDK • 3 STAs (AP only mode) on MT7686, MT7682, and MT5932 platform
WPS	Registrar (PBC/PIN) , Enrollee (PIN)
Enterprise Security	N/A

1.2.2. Network

The internet middleware APIs can be found in the Internet Middleware API Reference Manual and Airoha IoT SDK for RTOS Open Source Components Guide under `<sdk_root>\doc`. Supported network features of the platform are listed in Table 5. Learn how to include each supported protocol module from `<sdk_root>\middleware\third_party\xxx\readme.txt`.

Table 5. Supported network protocols

Item	Features
IP Stack	• IPv4 (LWIP)

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Item	Features
	• TCP, UDP • ICMP • DHCP Client/Server • DNS Client • NETCONN • SOCKET
SNTP	• Simple Network Time Protocol • RFC4330 • Support SNTP receive timeout • Support SNTP update delay • Support SNTP max server
HTTP	• HTTP 1.1 • Client (POST/GET)
HTTPS	• HTTP 1.1 • Client (POST/GET)
SSL/TLS	• mbed TLS • Client, Server (not tested) • SSL3.0, TLS1.0, 1.1, 1.2 • AES, 3DES, DES, ARC4 • MD5, SHA-1, SHA-256 • RSA/PKCS#1 v1.5

1.2.3. Bluetooth and Bluetooth Low Energy

Bluetooth with Enhanced Data Rate (EDR) and Bluetooth Low Energy (LE) are key features in the Airoha IoT SDK for RTOS. The details are listed in Table 6. The SDK API and module descriptions can be found in the API reference guide and Bluetooth developer's guides for LinkIt 2523 HDK and LinkIt 7697 HDK at `<sdk_root>\doc`. In addition, find more details on how to include the Bluetooth module in `<sdk_root>\middleware\MTK\bluetooth\readme.txt`.

Table 6. Bluetooth/Bluetooth Low Energy features

Item	Features
EDR-A2DP	Advanced Audio Distribution Profile
EDR-AVRCP	Audio/Video Remote Control Profile (CT:v1.3/TG:v1.0)
EDR-HFP/HSP	Hands-Free Profile v1.7 or Headset Profile
EDR-PBAP	• Phone Book Access Profile (PBAP) o Defines the procedures and protocols to exchange Phonebook objects between devices.
EDR-SPP	Serial Port Profile
EDR-GAP	Generic Access Profile

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Item	Features
BLE-GAP	Generic Access Profile
BLE-GATT/ATT	Generic Attribute Profile
BLE-SMP	Low Energy Security Manager Protocol
Multipoint Support	• Supports multipoint Bluetooth access in EDR. o Two HFP (HF) o Two A2DP (Sink) o Two AVRCP (CT) o Two SPP server/client • Supports multipoint Bluetooth access in Bluetooth Low Energy. o Four Bluetooth Low Energy links.

1.2.4. Sensor subsystem

Supported sensor subsystem features of the Airoha IoT SDK are listed in Table 7. More information on the sensor subsystem SDK APIs can be found in sensor subsystem section of the 2523 API Reference Manual under `<sdk_root>\doc`. In addition, find more details on how to include the sensor subsystem module in `<sdk_root>\middleware\MTK\sensor_subsys\readme.txt`.

Table 7. Sensor subsystem features

Item	Features
Physical sensor	• Accelerometer • Biosensors (PPG, EKG)
Sensor fusion	• Heart rate • Blood pressure

1.2.5. GNSS

The detailed list of GNSS features is provided in Table 8. The API and module descriptions can be found in API Reference Manual and Airoha IoT SDK for RTOS GNSS Developer's Guide under `<sdk_root>\doc`. In addition, find more details on how to use this module in `<sdk_root>\middleware\MTK\gnss\readme.txt`.

Table 8. GNSS features

Item	Features
GNSS	• GPS, BeiDou, GLONASS • Low Power Mode • Periodic mode • GLP mode • Time Aiding, Location Aiding
Extended Prediction Orbit (EPO)	• EPO host aiding • EPO download using Bluetooth

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

1.2.6. FOTA

The detailed list of FOTA features is provided in Table 9. The API and module descriptions can be found in Airoha IoT SDK API Reference Manual and Airoha IoT SDK for RTOS Firmware Update Developer's Guide under `<sdk_root>\doc`. In addition, find more information on how to include this module in `<sdk_root>\middleware\MTK\fota\readme.txt`.

Table 9. FOTA features

Item	Features
FOTA	• Full binary update mechanism • Package data compression • Integrity check • Power loss protection • FOTA packaging tool

1.2.7. Peripheral drivers

The detailed list of peripheral drivers is provided in Table 10. The APIs for the drivers can be found in the Airoha IoT SDK API Reference Manual under `<sdk_root>\doc`. To include HAL module, include `<sdk_root>\driver\chip\mt2523\module.mk` in project makefile for LinkIt 2523 HDK, or include `<sdk_root>\driver\chip\mt7687\module.mk` in project makefile for LinkIt 7687 HDK.

Table 10. Supported peripheral drivers

Item	Features
ACCDET	• Accessory Detector. • Detects plug-in/out of earphone based on the suggested circuit.
ADC (MT2523x)	• ADC module. • DAC module.
CACHE	• The maximum size of the cache is 32kB.
EINT	• External interrupt controller. • Processes the interrupt request from an external source or a peripheral device.
Flash	• Supports execute in place (XIP) and programming flash by software. • Default 2MB system in package (SiP) flash on LinkIt 7687 HDK • Default 4MB system in package (SiP) flash on LinkIt 2523 HDK. • Supports external flash up to 16MB on Airoha IoT SDK for RTOS platform HDKs
GPIO	• GPIO mode (in or out) • Set Pull Up/Down for GPIO IN mode
GPT	• General Purpose Timer. • Supports 32kHz and 1MHz clock sources, repeat and one-shot modes for timing events and delays

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Item	Features
	in µs or ms.
PWM	• Range is 256 duty cycles • 32kHz, 2MHz, XTAL clock for PWM frequency reference
UART	• Two full set (TX/RX) UART support on LinkIt 7687 HDK • Four UART ports, two of them featuring hardware flow control on LinkIt 2523 HDK • Baud rate of up to 921600
I2C Master	• Two I2C interfaces • Supports 50/100/200/400kHz transmission rate
I2S Master	• I2S master is capable of servicing an external codec component. • Supports 8/11.025/12/16/22.05/24/32/44.1/48 kHz audio sampling rates in stereo mode.
ISINK	• Current sink • Adjustable backlight current.
MPU	• Memory Protection Unit
IrDA	• TX (NEC, RC5, RC6, pulse width) • RX (RC5, pulse width)
GPC	• General Purpose Counter • Supports 1MHz pulse detection
WDT	• Supports hardware, software watchdog • Supports system reset
I2S-Slave	• Supports sample rates: 8,12, 16, 24, 32, 48 kbits • Supports mono and stereo mode
SPI-Master	• Serial Peripheral Interface
RTC	• Real-Time Clock
GDMA	• General Purpose DMA
Security	• SHA1, SHA2 (256, 384, 512), MD5, AES, 3DES
TRNG	• Truly Random Number Generator • Generates 32bit random number
Charger	• Supports single-cell Li-Ion battery charging.
Keypad	• Keypad scanner • Supports 3x3 single/double key mode

1.2.8. Battery management

The battery management features are listed in Table 11. The battery management APIs are found in 2523 API Reference Manual under `<sdk_root>\doc`. Find more information on how to include this module in `<sdk_root>\middleware\MTK\battery_management\readme.txt`.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Table 11. Battery management features

Item	Features
Battery management	• Charging flow control mechanism. • Algorithm for battery capacity measurement. • Precise information on the battery, including temperature and battery level.

1.2.9. Advanced features and components

The advanced features and components included in the platform are listed in Table 12.

Table 12. Advanced features and components

Item	Features
XML	• Mini-XML • Supports o Entity o GET/SET o Index o Search
JSON	• cJSON • JSON string parser
Smart Connection	• Airoha IoT Smart Connection
CLI command	• CLI command parser
ATCI	• AT command parser
File system	• Windows compatible • Platform independent • Very small footprint for code and work area • Multiple volumes • Multiple ANSI/OEM code pages including DBCS • Long file name support in ANSI/OEM or Unicode • FreeRTOS support for multitasking • Multiple sector size support up to 4kB • Read-only, minimized API, I/O buffer and more

1.3. Folder structure

The SDK is delivered as a single package organized in a folder structure, as shown in Figure 2.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

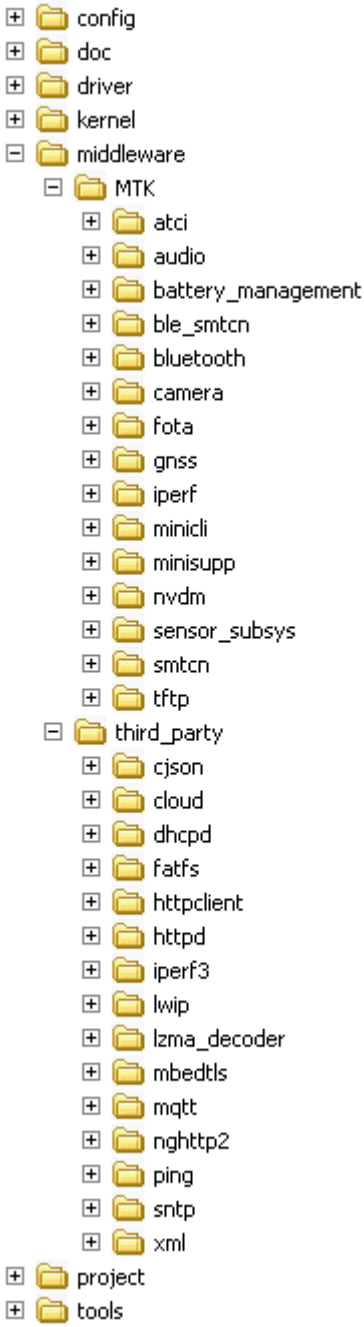

Figure 2. Folder structure

This package contains the source and library files of the major components, build configuration, related tools and documentation. A brief description on the layout of these files is provided below:

- `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.
- `project`. Includes pre-configured example and demo projects using Wi-Fi, HTTP, HAL, and more.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

- `tools`. Includes tools to compile, download and debug projects using the SDK.

The main components that belong to middleware are in the `middleware` folder:

- MTK
  - `minicli`. A Command Line Interface (CLI) that provides a framework for the upper layer to register a function executed by an input command. The input command and output message streaming is communicated through the UART.
  - `minisupp`. A supplicant is an entity at one end of a point-to-point LAN segment that seeks to be authenticated by an authenticator attached to the other end of that link. Mini-supplicant library supports this with minimum memory usage.
  - `nvdm`. NVDM is a type of memory mechanism that retains its contents when the system power is turned off.
  - `atci`. Provides the interface for a target communication using AT commands though UART.
  - `battery_management`. Includes battery monitor, charging flow control and battery capacity algorithms.
  - `bluetooth.` Bluetooth/Bluetooth Low Energy provides three profiles (GAP, GATT, SM) to discover and connect Bluetooth devices and to transfer and control data securely through a Bluetooth connection.
  - `fota`. FOTA provides firmware update functionality.
  - `gnss`. Provides APIs to receive GNSS data and control the on-board GNSS module.
  - `sensor_subsys`. Provides sensor drivers and fusion algorithms.
- `third_party`
  - `cjson`. JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. cJSON is a single file implementation in C.
  - `dhcpd`. DHCP daemon (DHCPD) is a program that operates as a daemon on a server to provide Dynamic Host Configuration Protocol (DHCP) service to a network. Devices in a network with a DHCP server can retrieve network parameter configuration from the server.
  - `httpclient`. The HTTP client is the client implementation for requesting data from HTTP servers.
  - `lwip`. A widely used open source TCP/IP stack designed for embedded systems. The focus of the lwIP TCP/IP implementation is to reduce resource usage while still having a full-scale TCP.
  - `mbedtls`. Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are cryptographic protocols designed to provide communications security over a computer network. mbed TLS is an open source implementation for developers to include cryptographic and SSL/TLS capabilities in embedded products with a minimal coding footprint.
  - `sntp`. SNTP is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.
  - `xml`. XML is a markup language defined by the W3C's XML 1.0 Specification that defines a set of rules for encoding documents in a human-readable and machine-readable format.
  - `fatfs`. FatFs is generic FAT file system for small embedded systems. It is used to control data storage and retrieval in a file system.
  - `lzma_decoder.`LZMA is the default and general compression method used to perform lossless data compression. LZMA is also suitable for embedded applications because it provides fast decompression and a high compression ratio.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

- `mqtt.` MQTT is a lightweight publish/subscribe messaging protocol, originally created by IBM and Arcom (later to become part of Eurotech) around 1998.

1.4. Project source structure

The SDK provides a set of reference applications. For example, projects with a single function showing how to use drivers or other module features and others with complex functionality demonstrating how to use the middleware components.

Example applications are located in the `<sdk_root>\project\<hdk>\apps` and `<sdk_root>\project\<hdk>\hal_examples` folder and they all have the same folder structure, as shown in Figure 3.

Figure 3. Project folder structure

1) `EWARM.` IAR related project configuration files.
2) `GCC.` GCC related project configuration files, such as a makefile.
3) `inc.` Project header files.
4) `MDK-ARM.` Keil related project configuration files.
5) `src.` Project source files.
6) `Readme.txt.` A brief introduction about project behavior and the required environment.

You can apply the relevant reference applications to further your development.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2. Getting Started Using GCC

This section provides a guide to getting started with the Airoha IoT development platform for RTOS and covers the following items:

- Supported environments for development.
- Configuring the LinkIt HDK.
- Building the project using the SDK.
- Downloading and running the project from Microsoft Windows.
- Debugging the project from Microsoft Windows.
- Creating your own project.

2.1. Environment

The SDK can be used on any edition of Microsoft Windows XP, Vista, 7 and 8, and on Linux. A GCC compiler is required to build the project.

- Download and extract the content of the SDK package on your local PC.
- Follow the instructions in the `readme.txt` file to download and extract the SDK toolchain package.
  - Copy the GCC compiler ("`gcc`" folder) to `<sdk_root>\tools\`. The compiler settings are in the `<sdk_root>\.config` configuration file.

The default GCC toolchain is supported for the following versions of the Linux 32 or 64 bit hosts.

- Ubuntu 8.x or later (tarball).
- Ubuntu LTS 10.04 or later (PPA).
- RHEL 4/5/6 (tarball).

2.2. Developing on LinkIt 7687 HDK

2.2.1. Configuring the LinkIt 7687 HDK

LinkIt 7687 HDK includes a main board and a MT7687F stamp module. The MT7687F stamp module is mounted on the main board. The top view of the main board is shown in Figure 4.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

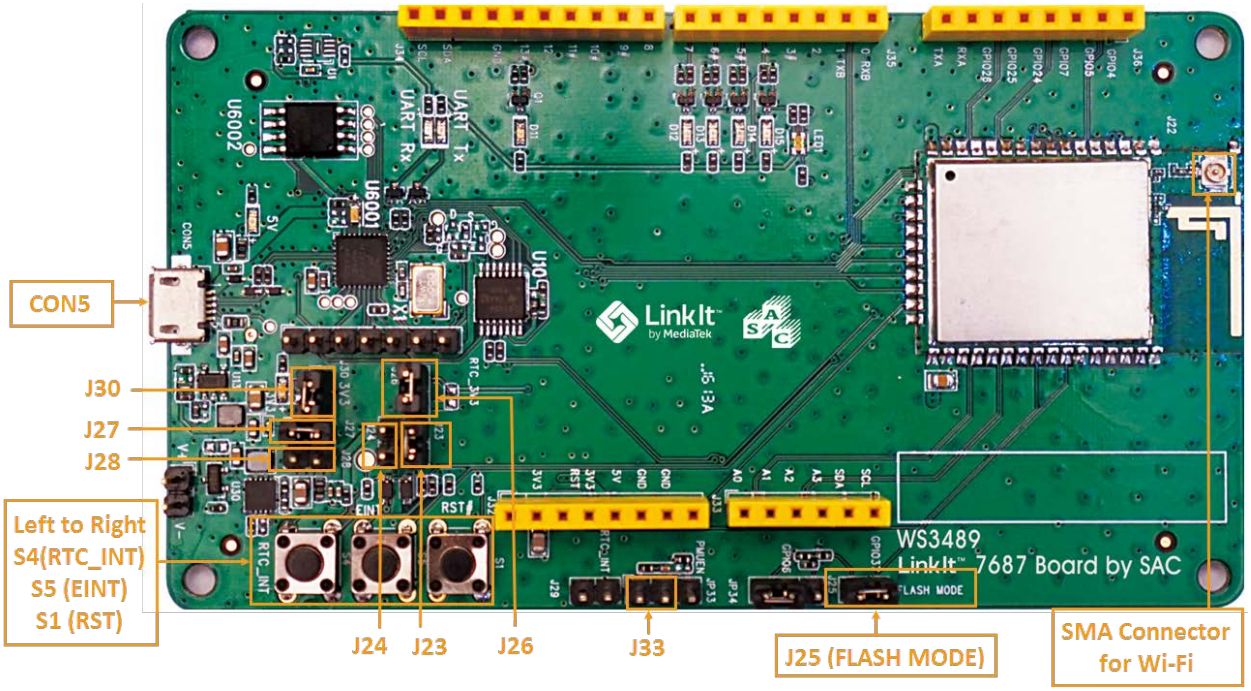

Figure 4. Top view of the LinkIt 7687 HDK

The description of pins and their functionality is provided below.

- **CON5** is a USB connector to debug through UART, transmit and receive a signal and supply power from the PC. The USB connectivity with the PC is supported by the on-board MK20DX128VFM5.
- Set the jumpers **J23**, **J26**, **J27**and **J30** on, if the board is powered up using the USB connector.
- Press **S4** to wake up the system from the RTC mode.
- Press **S1** to reset the system.
- **J25** sets the flash mode.
  - Remove the jumper **J25** to switch to **FLASH Normal** mode. In this mode, if the power is on, the board will load firmware from the flash and reboot.
  - Set the jumper **J25** on to switch to **FLASH Recovery** mode. In this mode, if the power is on, the board will load ROM code and wait for the MT76x7 Flash Tool to initiate firmware download process.
- **Wi-Fi Antenna** is a PCB antenna. MT7687F stamp module is by default connected to the PCB antenna to transmit and receive RF signals.

The default configuration of the LinkIt 7687 HDK support the following functionality:

- Power supply — attach a micro-USB connector to the **CON5.**
- Flash mode — **Recovery** mode.
- Supports RTC interrupt.
- Clock source — 32.768kHz source crystal clock for the RTC mode or external clock operating on 32.768kHz.
- XTAL — 40MHz.
- Supports RTC mode.

The hardware settings of the stamp module are shown below:

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

- XTAL — 40MHz.
- Clock source — 32.768kHz source crystal clock for the RTC mode or external clock operating on 32.768kHz.
- Supports RTC mode.
- Flash mode — **Normal** mode.

2.2.2. Installing the LinkIt 7687 HDK drivers on Microsoft Windows

To configure the LinkIt 7687 HDK:

- Connect the HDK to the computer using a micro-USB cable.
- Download and install mbed Windows serial port driver from here. Open Windows **Control Panel** then click **System** and:
  - On Windows 7 and 8, click **Device Manager**.
  - On Windows XP, click the **Hardware** tab and then **Device Manager**.
- In **Device Manager**, navigate to **Ports (COM & LPT)** (see Figure 5).
- A new **COM** device should appear under **Ports (COM & LPT)** in **Device Manager**, as shown in Figure 5. Note the **COMx** port number of the serial communication port, this information is needed to send command and receive logs from the COM port.

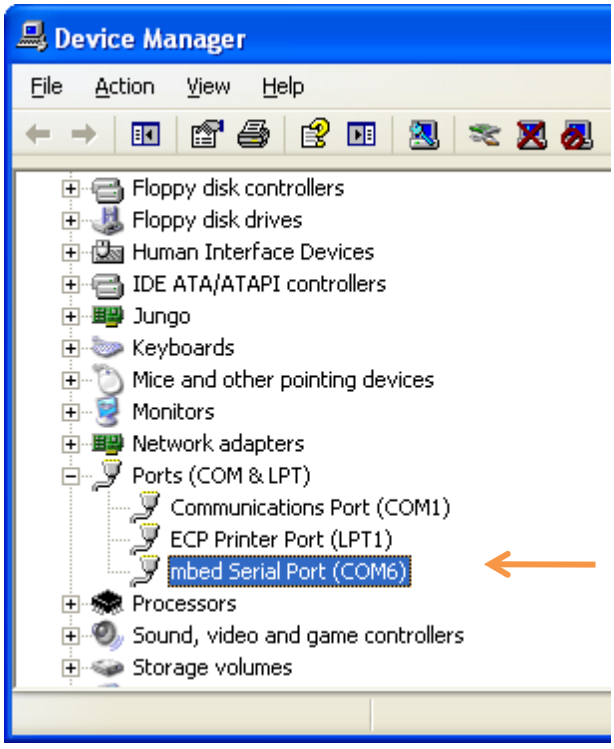

Figure 5. COM port associated with the LinkIt 7687 HDK

2.2.3. Installing MT76x7 Flash Tool for LinkIt 7687 HDK

The MT76x7 Flash Tool can be used on Microsoft Windows XP, Vista, 7, 8 and 10 with 32-bit and 64-bit operating systems. It's available for Linux with 32-bit and 64-bit operating system (Ubuntu 14.04 or higher).

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

To install the tool:

1) Download the Airoha IoT SDK v4 package from here.

On Linux OS environment:

2) Extract the content of the SDK and navigate to the MT76x7 Flash Tool's folder (`./linux32bit` or `./linux64bit`). The tool is a setup free package.

3) Run the following command to establish the library environment and launch the Flash Tool:

```
source env-setup.sh
./mt76x7-flash-tool.exe
```

On Microsoft Windows OS environment:

1) Extract the content of the SDK and navigate to the MT76x7 Flash Tool's folder (`./win`). The tool is a setup free package.

2) Execute the `mt76x7-flash-tool.exe` to launch the Flash Tool (see Figure 6).

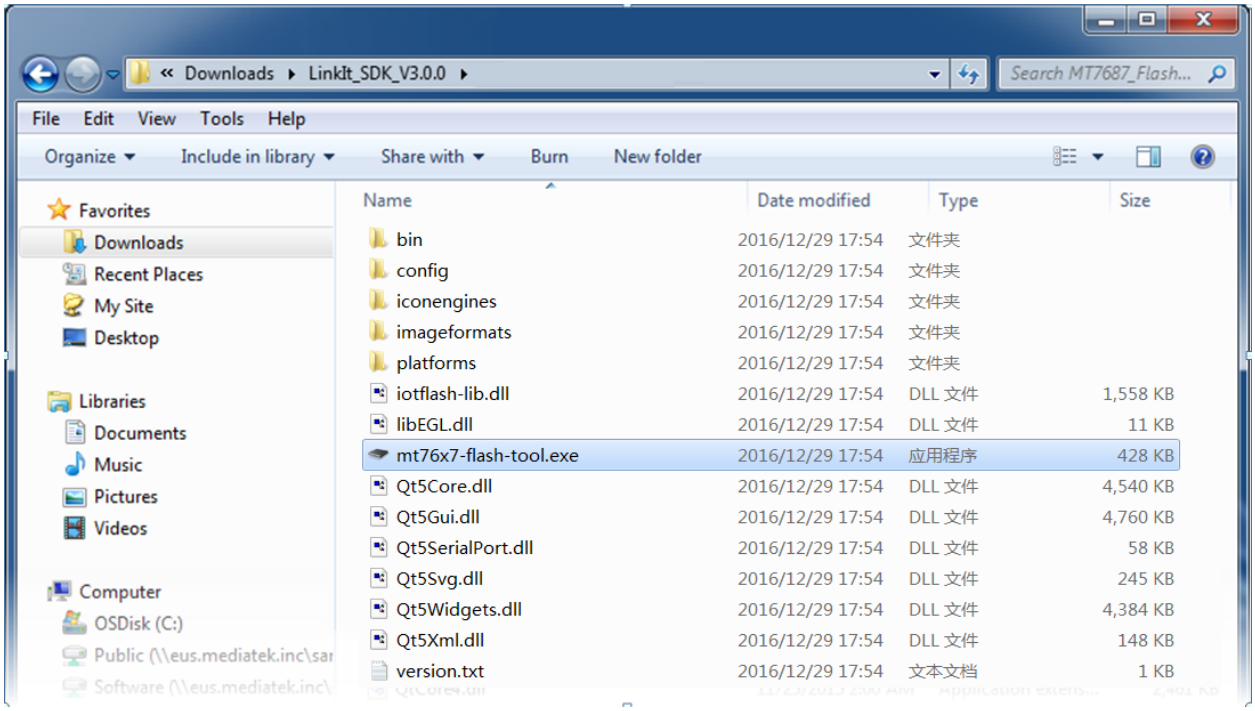

Figure 6. Executable file in the MT76x7 Flash Tool's destination folder

To configure the COM port and the baud rate:

- Launch the MT76x7 Flash Tool and connect the device to a PC with a micro-USB cable.
- Reset the development board or unplug and re-plug in the micro-USB cable.
- Click **Refresh** and configure the correct COM port (see section 2.2.2, "Installing the LinkIt 7687 HDK drivers on Microsoft Windows") and the baud rate, as shown in Figure 7.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

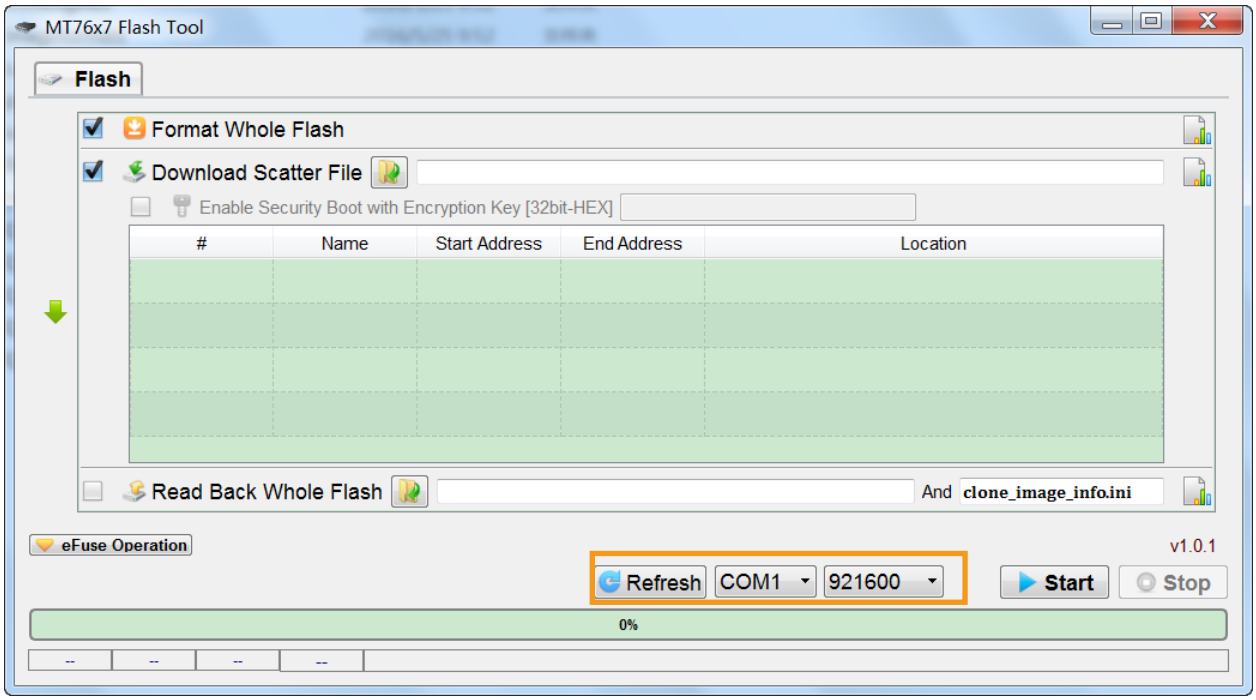

Figure 7. Setup the COM port and baud rate

2.2.4. Flashing the image to LinkIt 7687 HDK

There are two methods on how to flash the image to the LinkIt 7687 HDK.

2.2.4.1. Using the Flash Tool

The **Download** feature flashes the images to the device. The image data can be scrambled, if necessary. If scrambled, the data read back cannot be decoded.

Follow these steps to execute commands.

1) Enable **Download Scatter File** checkbox.
2) Click **open folder** to provide the scatter file. The scatter file, known as **Image Description File**, is located under the project binary folder, where all other binary files reside. It is usually named as `flash_download.ini`. The binary files could be either Airoha released or generated during build process. The **Image Description File** includes information about the file version, project name and partition layout details. Most projects have three partitions:
    a) **Loader** partition that stores bootloader.
    b) **TargetFW** partition that stores proprietary firmware for the Wi-Fi subsystem.
    c) **HostFW** partition that stores the project application binary, such as `mt7687_iot_sdk_demo.bin`.
3) The partition layout details including **Name**, **Start Address**, **End Address** and the location of the files are shown in Figure 8.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

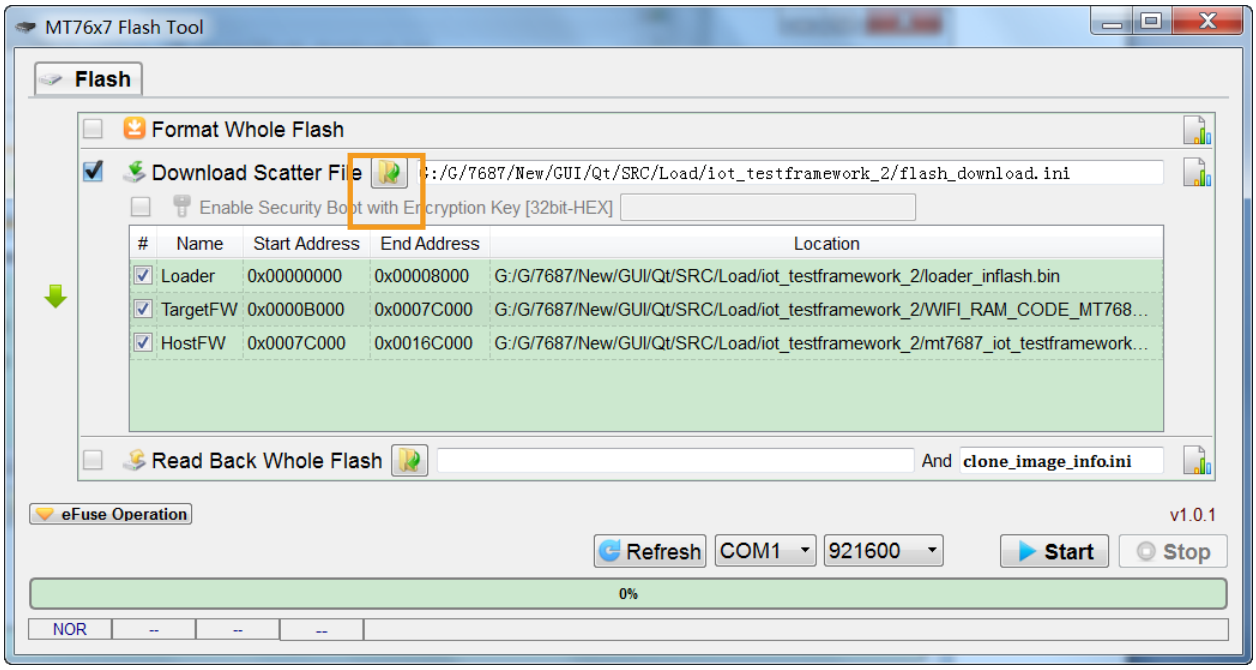

Figure 8. Download an image to the device storage

4) Click **Start** to execute the command. When downloading is complete, the result status is shown as in Figure 9.

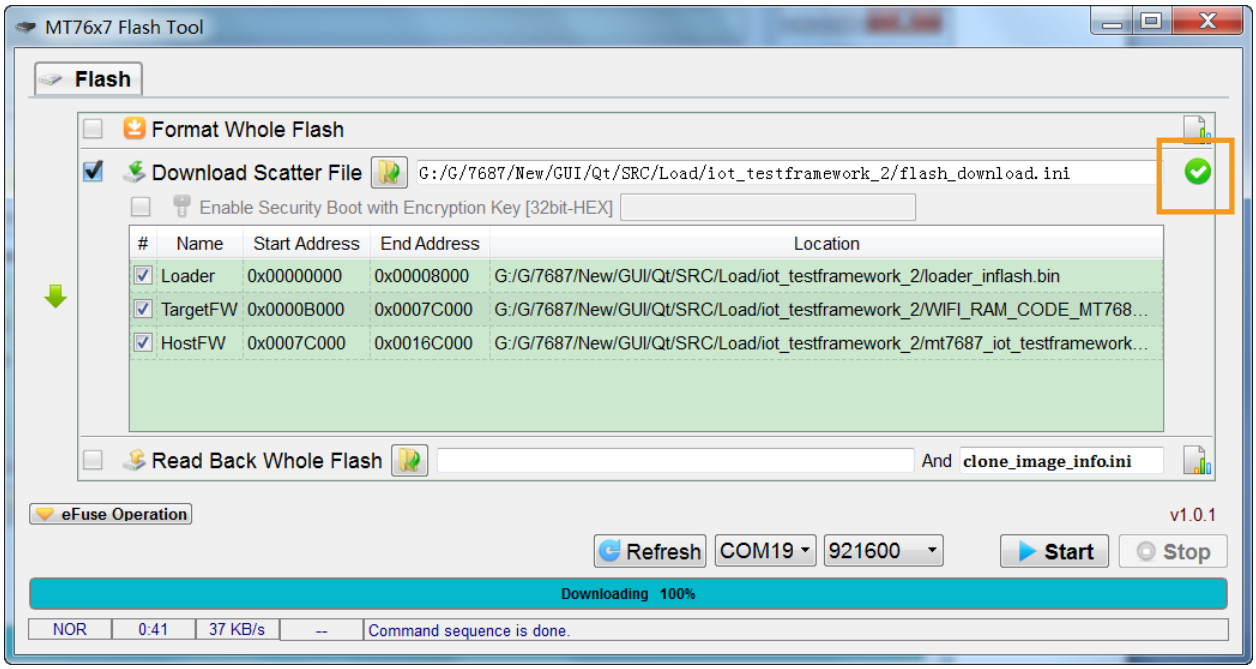

Figure 9. Downloading the image is successfully complete

2.2.4.2. Using the LinkIt 7687 HDK as a removable storage

To update the application binary only (example project binary: `mt7687_iot_sdk_demo.bin`), use the HDK as a mass storage device as follows:

1) Set the HDK to **FLASH Recovery** mode (see section 2.2.1, “Configuring the LinkIt 7687 HDK”).

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2) Power up the board with a micro-USB cable.

3) Navigate to **Computer** on your PC to check if a new mass storage named **MT76x7** is available under **Removable Disk**, as shown in Figure 10.

4) Open the **MT76x7** removable storage, then drag and drop the binary `mt7687_iot_sdk_demo.bin` to complete downloading the image.

5) Disconnect the board, set the jumper to **FLASH Normal** mode, and then reconnect the board to run your application.

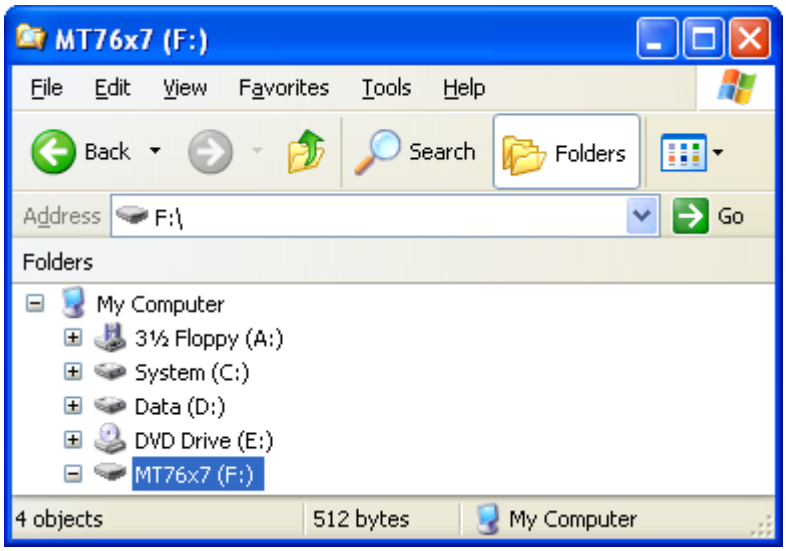

Figure 10. New removable storage detected

2.2.5. Running the project on LinkIt 7687 HDK

Most of the example projects for the LinkIt 7687 HDK facilitate the on-board mbed serial port to output logs and accept user inputs. Therefore, you'll need to setup a terminal application that connects to the serial port, such as TeraTerm terminal emulator software. You can use any other terminal application of your choice.

2.2.5.1. Terminal application setup

If you already have a terminal application installed, skip to the next section to configure the serial port.

To install TeraTerm, follow the steps below:

1) Download the latest version of TeraTerm installer from here.

2) Launch installer, such as `teraterm-4.91.exe`, and simply follow the steps using default options to complete the installation.

2.2.5.2. Serial port settings

All example projects use the same serial port configuration. Apply the following configuration settings in your terminal application:

If you are using TeraTerm:

3) Launch **TeraTerm** and then click **Setup** on the top menu of the command window.

4) Click to open **Serial Port...** setup.

5) Select the COM port number that maps to the mbed serial port shown in Figure 5.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

6) Set the **Baud rate**, **Data**, **Parity**, **Stop** and **Flow control** parameters (see Figure 11). Leave the **Transmit delay** fields with default values (0), and click **OK**.

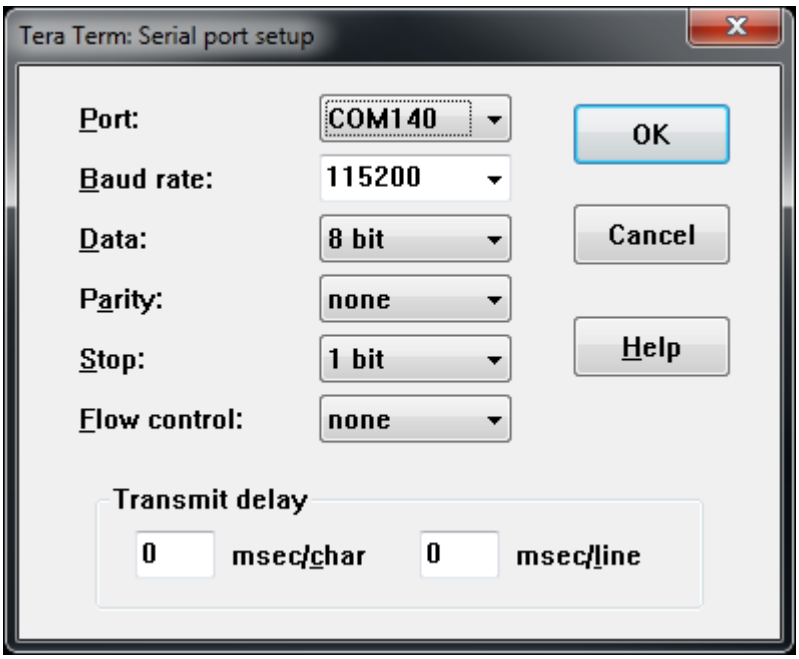

Figure 11. Serial port setting configuration

7) To apply this configuration to all future serial port connections, select **Setup** and then **Save Setup...** to save the `TERATERM.INI` file to the TeraTerm program folder. The default location is `C:\Program Files (x86)\teraterm`.

2.2.5.3. Run the project

To run the project on LinkIt 7687 HDK:

1) Disconnect the micro-USB cable to power off the board.
2) Switch the flash mode to **FLASH Normal** mode, see section 2.2.1, “Configuring the LinkIt 7687 HDK”, for more details.
3) Reconnect the USB cable to power on the board.
4) Open the terminal application and connect to the mbed serial port. If you are using TeraTerm, select **File** and then **New Connection...** from the menu bar, and then select **Serial**. Provide the **Port** corresponding to **mbed Serial Port**, as shown in Figure 12, and click **OK**.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

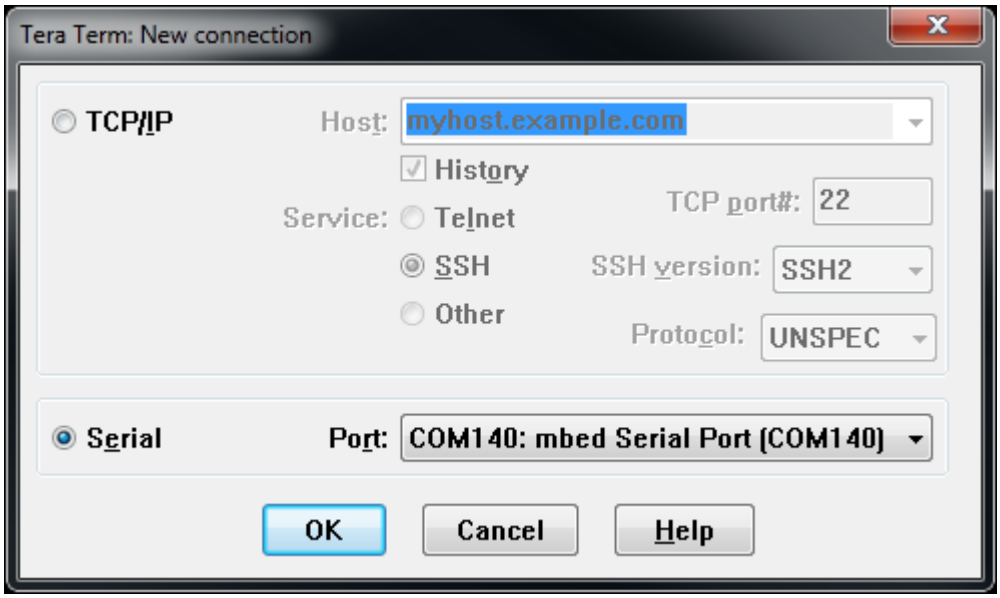

Figure 12. Connect to mbed Serial Port

5) Press the reset button **S1 (RST)** on the board.

6) Observe the output log written in project source files from UART port. A reference log example is shown below.

```
loader init [2016-04-28 15:03:10.924]
[2016-04-28 15:03:10.924]
fota: TMP is empty, skip upgrade[2016-04-28 15:03:10.939]
[2016-04-28 15:03:10.939]
jump to (0x1007c000) [2016-04-28 15:03:10.939]
total avail space = 13748
[2016-04-28 15:03:10.955]
nvdm init finished
[2016-04-28 15:03:10.955]
[T: 52 M: common C: INFO F: system_init L: 269]: FreeRTOS Running[2016-04-28
15:03:10.970]
```

2.2.6. Debugging with the LinkIt 7687 HDK from Microsoft Windows

This section describes how to debug a project built with the GCC compiler using openOCD debugger tool.

Before commencing project debugging, install supporting software on Windows OS.

1) Download openocd-0.9.0 from here and unzip it into `<openocd_root>` folder.

   a) Download GCC toolchain from here for your Windows version, and unzip it into `<gcc_root>` folder.

2) Install the mbed serial port driver, if the mbed serial port driver is not installed (see MT76x7 Flash Tool User's Guide).

You can debug a project once it's already downloaded on your HDK. More details can be found in Airoha IoT SDK GCC Build Environment Guide.

3) Create a board configuration file named `mt7687.cfg` and copy the following content to the file:

```
puts "load MT7687 configuration"

#source [find interface/jlink.cfg]
#transport select swd
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
source [find interface/cmsis-dap.cfg]
source [find target/swj-dp.tcl]

global _CHIPNAME
global _TARGETNAME
global _CPUTAPID
# Setup variables
set _CHIPNAME mt7687
set _TARGETNAME $_CHIPNAME.cm4
set _CPUTAPID 0x2ba01477

# Create DAP instance
swj_newdap $_CHIPNAME cpu -irlen 4 -expected-id $_CPUTAPID

# Create target instance
target create $_TARGETNAME cortex_m -endian little -chain-position
$_CHIPNAME.cpu

# Setup SWD frequence
adapter_khz 1000

reset_config srst_only
#adapter_nsrst_delay 1000

# Hook for reset, to clear init_done flag
$_TARGETNAME configure -event reset-start {
echo "reset start"
}
# Hook for reset & init, to execute initialization steps
$_TARGETNAME configure -event reset-init {
global _TARGETNAME
targets $_TARGETNAME
echo "reset init"
}

$_TARGETNAME configure -event reset-end {
global _TARGETNAME
targets $_TARGETNAME
unlock_swd
enable_debug
echo "reset end"
}

# Hook for GDB attach, to bring target into debug mode
$_TARGETNAME configure -event gdb-attach {
targets $_TARGETNAME
unlock_swd
halt
}
# Hook for GDB detach, to free target from debug mode
$_TARGETNAME configure -event gdb-detach {
targets $_TARGETNAME
resume
}

# MT7687 initialization, unlock SWD lock
$_TARGETNAME configure -event examine-end {
global _TARGETNAME
targets $_TARGETNAME
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
unlock_swd
}

#unlock swd
proc unlock_swd {} {
      global _TARGETNAME
      targets $_TARGETNAME

  mem2array ram_code 32 0x0 1
      if {$ram_code(0) == 0} {
            mww 0x8300F050 0x76371688
            mww 0x8300F050 0x76371688
            mww 0x8300F050 0x76371688
            echo "unlock swd success"
      }

      mem2array ram_code 32 0x0 1
      if {$ram_code(0) == 0} {
            echo "unlock swd failed, please do hw power-on reset"
      }
}

#enable debug
proc enable_debug {} {
      global _TARGETNAME
      targets $_TARGETNAME
      mww 0xE000EDF0 0xA05F0003
}

puts "Load MT7687(CMSIS-DAP) configuration done"
```

4) Place the board configuration file under `<openocd_root>\share\openocd\scripts\board.`

Start debugging with the LinkIt 7687 HDK:

1) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>`. For example, `<sdk_root>\out\mt7687_hdk\iot_sdk_demo\mt7687_iot_sdk_demo.elf`.
2) Open the Windows command window to run openOCD.
3) Change the directory of command window to openOCD tool folder, such as `<openocd_root>`\bin.
4) Disconnect the micro-USB cable to power off the board.
5) Set the board to **FLASH Normal** mode.
6) Reconnect the micro-USB cable and press the reset button to power on the board.
7) Run the command to start the openOCD.

```
openocd.exe –s ..\share\openocd\scripts –f board\mt7687.cfg
```

8) Open the Windows command window to run openOCD GNU project debugger (GDB). .
9) Change the directory of the command window to the tool folder, such as `<gcc_root>\bin`.
10) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\mt7687_iot_sdk_demo.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
(gdb) x/10i $pc
```

You now have an openOCD debugger running on your system.

Note, openOCD debugging cannot work if enter sleep mode, for more detail information, please refer to the document Airoha IoT SDK for RTOS Power Mode Developers Guide under <sdk_root>/doc.

2.3. Developing on LinkIt 2523 HDK

2.3.1. Configuring the LinkIt 2523 HDK

Before commencing the application development, you need to configure the HDK.

The front view of the LinkIt 2523 HDK is shown in Figure 13.

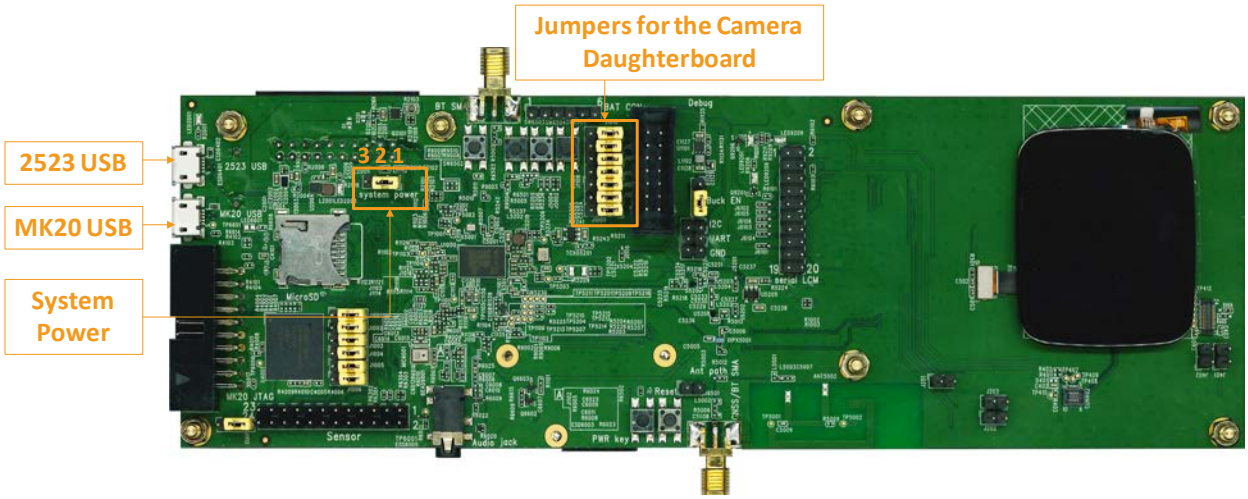

Figure 13. USB connectors on the LinkIt 2523 development board

The **2523 USB** can be used for powering up the board and downloading the binary using the IoT Flash Tool. The **MK20 USB** can be used for debugging with GDB and downloading the binary using Keil IDE.

To power up the board with **2523 USB**, connect the **system power** jumper pins 3 and 2 for USB power mode. Connect the pins 2 and 1 to power up with the battery, as shown in Figure 13.

2.3.2. Installing IoT Flash Tool for LinkIt 2523 HDK

IoT Flash Tool is a flexible device flashing tool for application development on LinkIt 2523 HDK.

To install the IoT Flash Tool:

1) If you don't have the SDK package yet, download the Airoha IoT SDK v4 package from here.
2) Extract the content of the SDK package and navigate to the **tools\pc_tool** folder, and extract the `PC_tool_Win.zip` package.

Furthermore, you also could find latest IoT Flash Tool from here by searching "IoT_Flash_Tool" in "Tool Name", and then download it.

In the extracted `PC_tool_Win` folder, navigate to **IOT_Flash_Tool** folder. The tool is a setup free package, and the `FlashTool.exe` inside the folder can be executed directly.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.3.3. Installing the LinkIt 2523 HDK drivers on Microsoft Windows

This section describes how to install LinkIt 2523 HDK drivers on PCs running Microsoft Windows. The two USB ports, **2523 USB** and **MK20 USB**, require different drivers. Follow the instructions below to install them.

2.3.3.1. Install 2523 USB Driver on Windows 7 or earlier

To install the MediaTek USB Port driver for **2523 USB** port on the HDK on Windows 7 or earlier versions of Windows:

1) Install the MediaTek USB Port driver from `MS_USB_ComPort_Driver` folder located in MT2523_FlashTool folder.
2) Execute `InstallDriver.exe` to install the driver.
3) Connect the **2523 USB** connector on the LinkIt 2523 HDK to your computer's USB port with a USB cable.

2.3.3.2. Install 2523 USB Driver on Windows 8 or later

If you are using Windows 8 or later, follow the instructions here:

1) Connect the **2523 USB** connector on the LinkIt 2523 HDK to your computer's USB port with a USB cable.
2) Open Windows **Control Panel** and click **System**, then click **Device Manager**.
3) An **Unknown device** will show under **Universal Serial Bus Controllers**. Double click the **Unknown device** and click **Update Driver...** (see Figure 14).

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

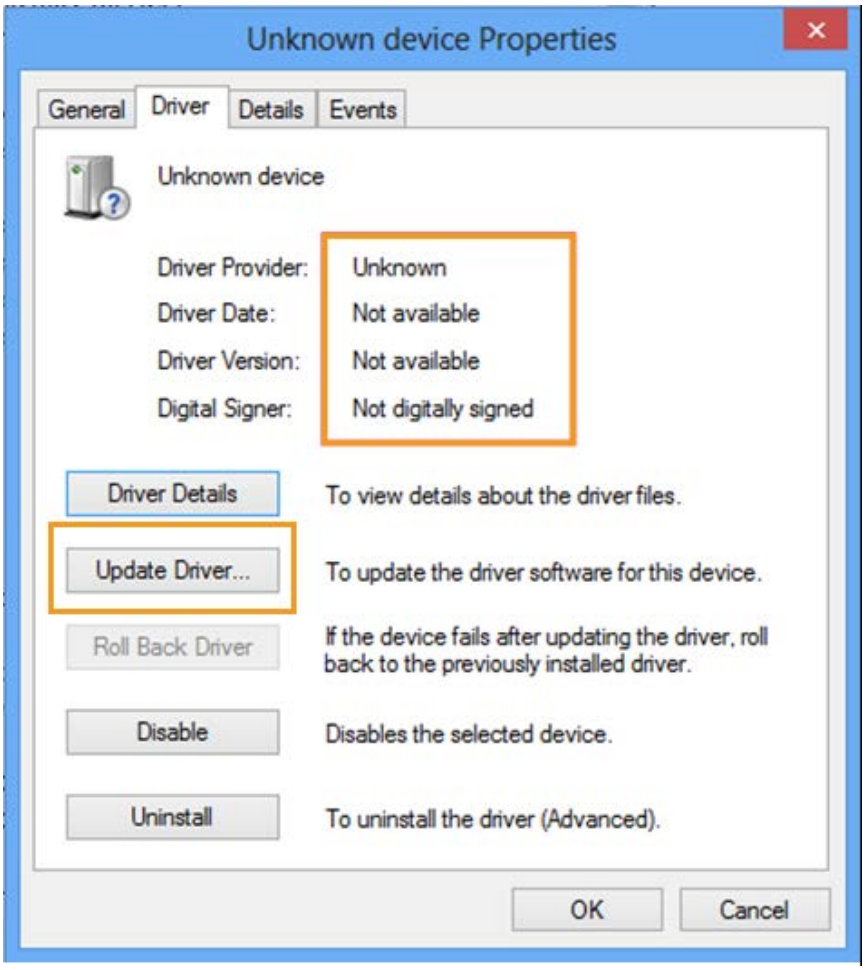

Figure 14. Unknown device on Windows 8 or later

4) Click **Browse my computer for driver software**, as shown in Figure 15.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

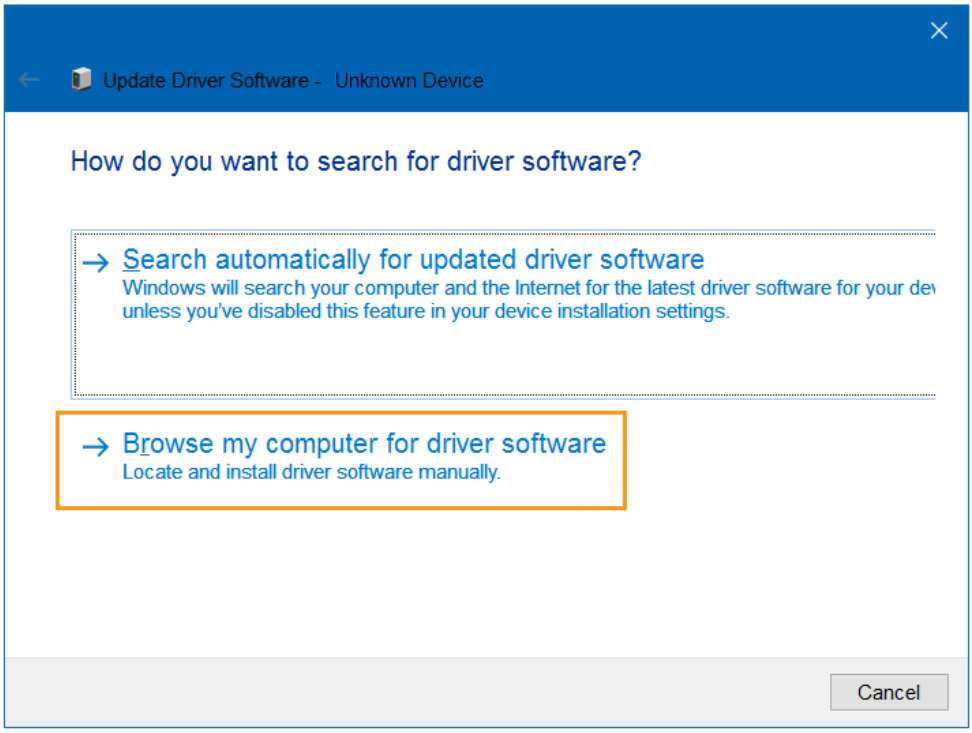

Figure 15. Browse my computer for driver software

5) Browse to the driver installation path, such as `MS_USB_ComPort_Driver\v1.1032.1\Win7`, then click **Next**. After a few seconds, the driver software and MTK USB Port device should be installed, as shown in Figure 16.

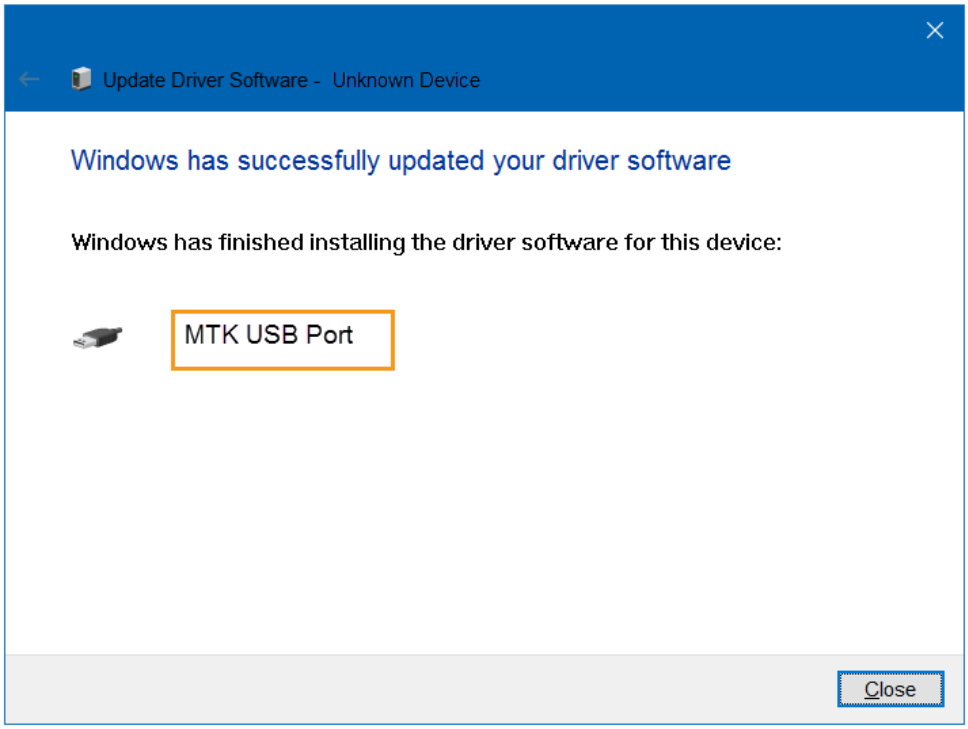

Figure 16. MTK USB Port on Windows 8 or later

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.3.3.3. Confirm COM port number of MTK USB Port

To determine the COM port number of the installed device:

1) Open Windows **Control Panel** and click **System** then
   a) On Windows 7 and 8, click **Device Manager**.
   b) On Windows XP, click the **Hardware** tab and then **Device Manager**.
2) In **Device Manager**, navigate to **Ports (COM & LPT)** and locate **MTK USB Port** (**COMx**), as shown in Figure 17.

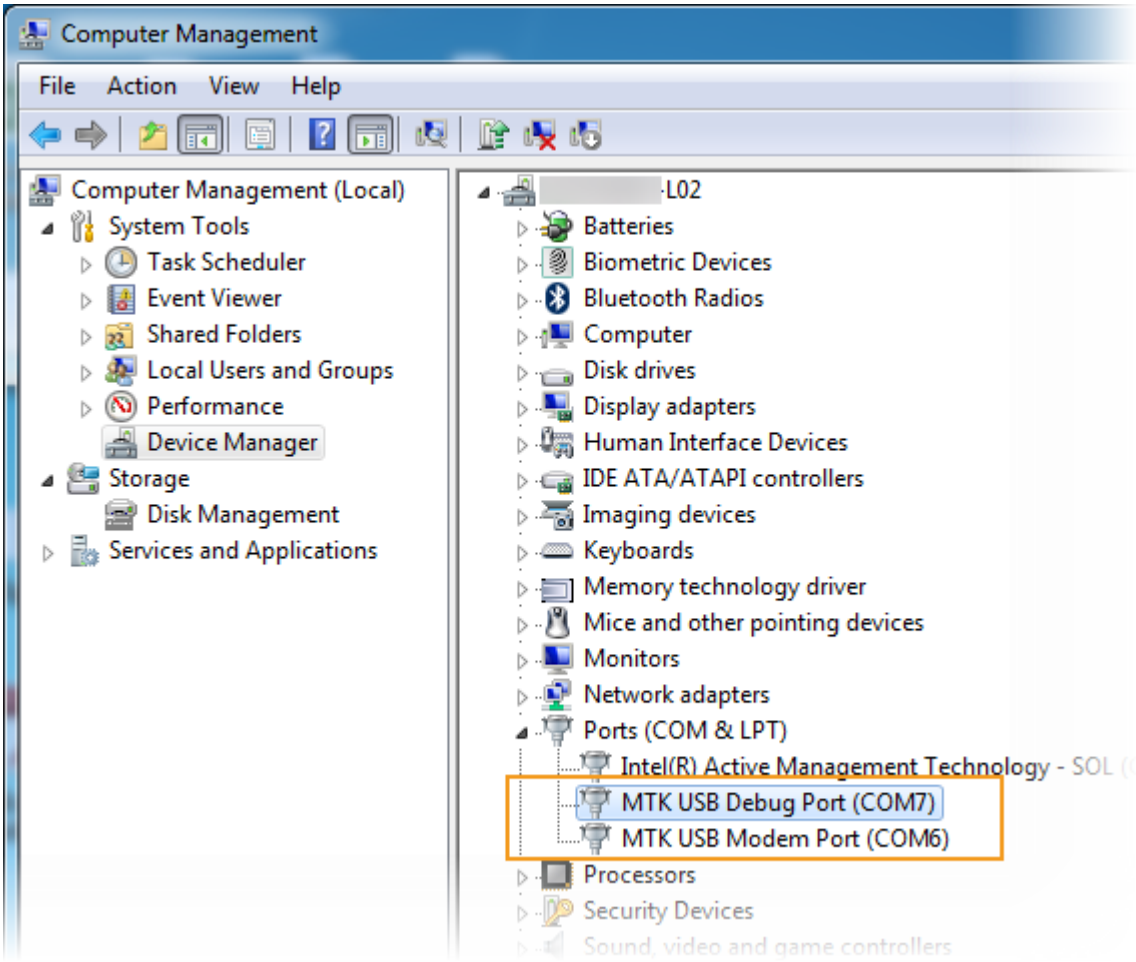

Figure 17. Debug and modem ports for MTK USB port

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

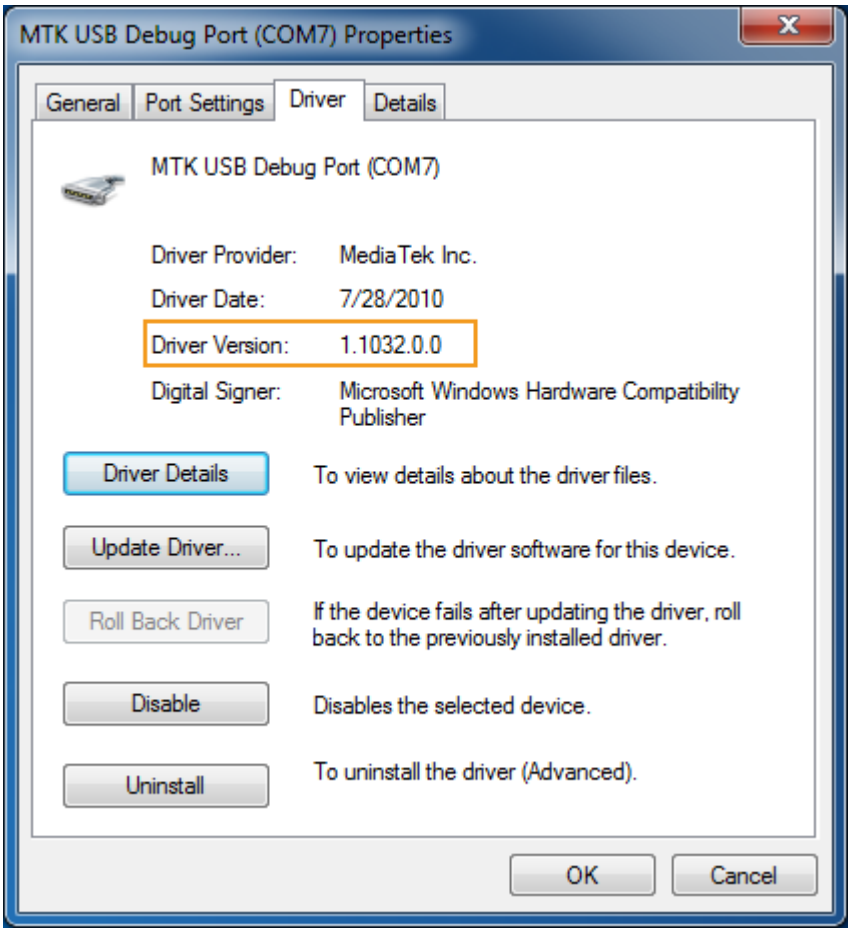

Figure 18. MTK USB driver version in MTK USB Debug Port Properties

Note, the driver version must be **1.1032.0** or later; an older driver doesn't guarantee successful download and operation. If the driver is before 1.1032.0, upgrade your SDK.

Figure 19. mbed serial port

2.3.4. Flashing the image to LinkIt 2523 HDK

6) Before using the IoT Flash Tool, it's required to have a pre-built project file (`.cfg`) or build your own project to get one (see 4), "Put the board configuration file under `<openocd_root>\share\openocd\scripts\board`.

Start debugging with AB155x EVK:

1) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>` (for example, `<sdk_root>\out\ab1552_evk\ earbuds_ref_design\debug\ earbuds_ref_design.elf`.)
2) Open the command window for openOCD.
3) Change the directory in the command window to the openOCD tool folder, such as `<openocd_root>\bin`.
4) Disconnect the micro-USB cable from the board to completely power off the board.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

5) Use a simulator (such as J-Link) to connect to JTAG pin (TMS, TCLK, VCC and GND) or SWD pin (SWDIO, SWCLK, VCC and GND) of AB155x.
6) Reconnect the micro-USB cable to the board to power-on the board.
7) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\ab155x.cfg
```

8) Open the command window for GNU project debugger (GDB).
9) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.
10) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\earbuds_ref_design.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have the openOCD debugging software running on your system.

Note: openOCD is a free third-party debugging tool (GPL license). To resolve any issues or perform troubleshooting, please refer to the openOCD official forum. In addition, openOCD debugging cannot work if the system goes into sleep mode. For more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/mcu/doc`.

Building the project using the SDK").

There are two methods to flash the image to LinkIt HDK: using the IoT Flash Tool or using the device as a removable storage device.

2.3.4.1. Using the Flash Tool

To download the firmware to the target device, use the **2523 USB** interface (see Figure 20):

1) Power off the target (USB cable must be unplugged).
2) Launch IoT Flash Tool, and click **Download** on the left panel of the main GUI.
3) Select **USB** from the **COM Port** drop down menu. If you don't have the adapter or battery, click the **Enable Download without Battery** option.

Click **Open** to provide the configuration file, which is usually named as `flash_download.cfg` and is generated after build process. If it loads successfully, **Download Information** will be displayed, including **Name**, **Length** and **File Path** of the firmware binary, as shown in Figure 20.

4) Click **Start** to start downloading.
5) Plug in the USB cable to power on the HDK through the **2523 USB** connector and then the process will start automatically.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

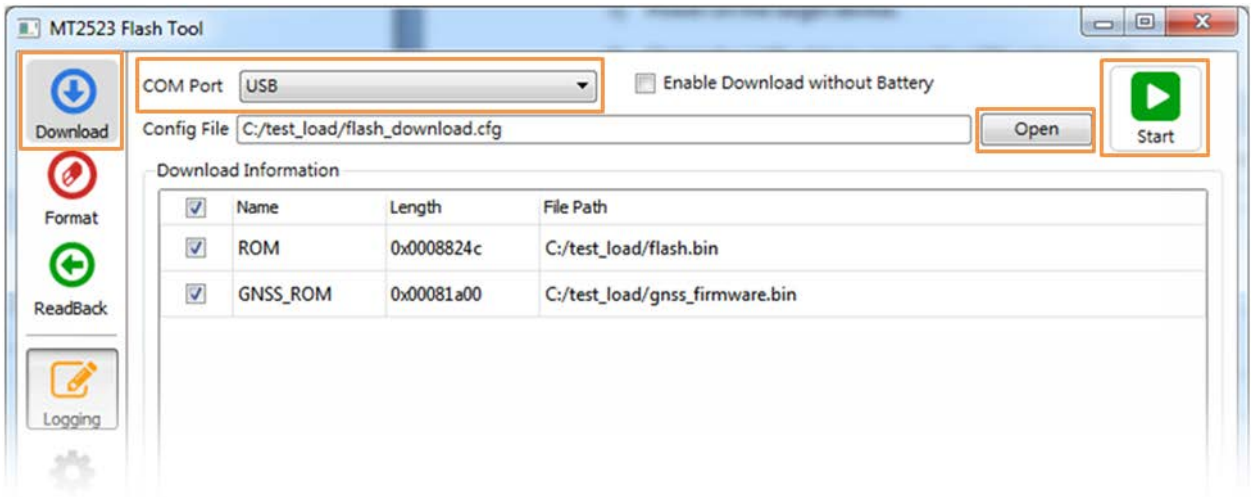

Figure 20. Download the firmware to a target device using USB connection

2.3.4.2. Using the LinkIt 2523 HDK as a removable storage device

6) To update the project image (example project image: iot_sdk_demo.bin (see section 4) Put the board configuration file under `<openocd_root>\share\openocd\scripts\board.`

Start debugging with AB155x EVK:

7) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>` (for example, `<sdk_root>\out\ab1552_evk\ earbuds_ref_design\debug\ earbuds_ref_design.elf`.)

8) Open the command window for openOCD.

9) Change the directory in the command window to the openOCD tool folder, such as `<openocd_root>\bin`.

10) Disconnect the micro-USB cable from the board to completely power off the board.

11) Use a simulator (such as J-Link) to connect to JTAG pin (TMS, TCLK, VCC and GND) or SWD pin (SWDIO, SWCLK, VCC and GND) of AB155x.

12) Reconnect the micro-USB cable to the board to power-on the board.

13) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\ab155x.cfg
```

14) Open the command window for GNU project debugger (GDB).

15) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.

16) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\earbuds_ref_design.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have the openOCD debugging software running on your system.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Note: openOCD is a free third-party debugging tool (GPL license). To resolve any issues or perform troubleshooting, please refer to the openOCD official forum. In addition, openOCD debugging cannot work if the system goes into sleep mode. For more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/mcu/doc`.

Building the project using the SDK")) use the LinkIt 2523 development board as a removable disk drive according to the following steps:

1) Connect the LinkIt 2523 HDK to your PC with a micro-USB cable to **MK20 USB** connector.
2) Navigate to **Computer** on your PC to check if a new mass storage named **MT2523** is available under **Devices with Removable Storage**, as shown in Figure 21.

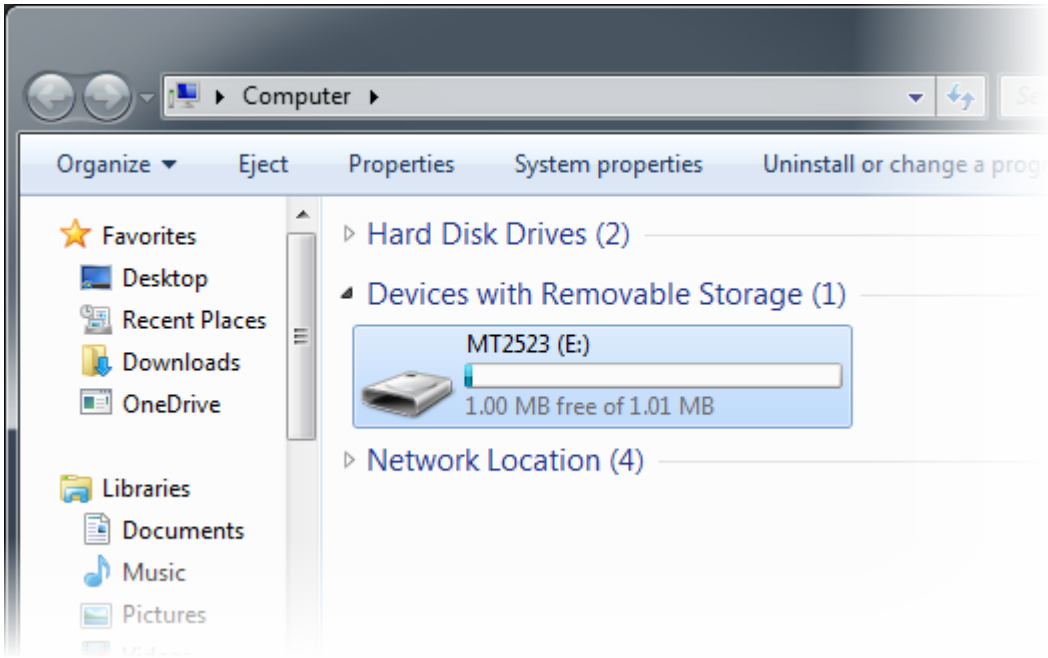

Figure 21. LinkIt 2523 HDK connected as removable disk storage

3) Open the **MT2523** removable storage, then drag and drop the project image, for example `iot_sdk_demo.bin`, from the original image folder to complete the update of the project image on the flash.
4) Wait for the mass storage device to disappear and re-appear again. After it reappears, check if there is no `fail.txt` in the 2523 mass storage. If there is no such file, the download has completed successfully. If `fail.txt` exists and its content is SWD ERROR, disconnect the MK20 USB port first, and make sure that you copy the `iot_sdk_demo.bin` file **within 15 seconds** after re-connecting to the **MK20 USB** port.

2.3.5. Running the project on LinkIt 2523 HDK

Most of the example projects for the LinkIt 2523 HDK facilitate the on-board mbed serial port to output logs and accept user inputs. Therefore, you'll need to setup a terminal application that connects to the serial port, such as TeraTerm terminal emulator software. You can use any other terminal application of your choice.

2.3.5.1. Terminal application setup

If you already have a terminal application installed, skip to the next section to configure the serial port.

To install TeraTerm, follow the steps below:

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

1) Download the latest version of TeraTerm installer from here.
2) Launch installer, such as `teraterm-4.91.exe`, and simply follow the steps using default options to complete the installation.

2.3.5.2. Serial port settings

All example projects use the same serial port configuration. Apply the following configuration settings in your terminal application:

If you are using TeraTerm:

1) Launch **TeraTerm** and then click **Setup** on the top menu of the command window.
2) Click to open **Serial Port...** setup.
3) Select the COM port number that maps to the mbed serial port shown in Figure 5.
4) Set the **Baud rate**, **Data**, **Parity**, **Stop** and **Flow control** parameters (see Figure 22). Leave the **Transmit delay** fields with default values (`0`), and click **OK**.

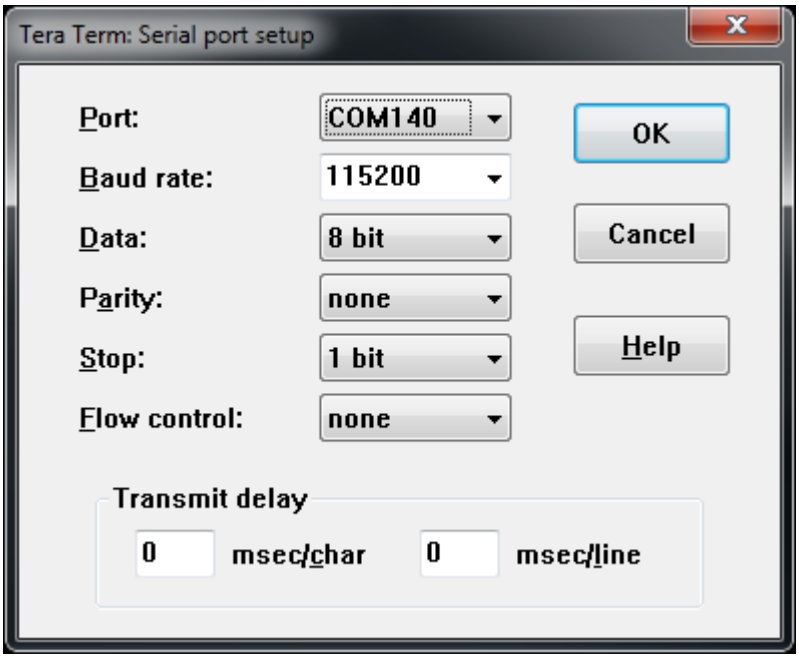

Figure 22. Serial port setup

5) To apply this setting to all future serial port connections, select **Setup** and then **Save Setup...**, and save the `TERATERM.INI` file to the TeraTerm program folder, the default location is at `C:\Program Files (x86)\teraterm`.

2.3.5.3. Run the project

To run the project on LinkIt 2523 HDK after downloading a project binary, reboot the board:

1) Disconnect the micro-USB cable to power off the LinkIt 2523 HDK.
2) Reconnect the USB cable to **MK20 USB** port to power on the LinkIt 2523 HDK.

Note, pressing the on-board **Reset** key won't reboot the board completely. After downloading a new firmware, you must un-plug the micro-USB cable and plug it back in to perform a complete reboot.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

3) Open the terminal application and connect to the mbed serial port. If you are using TeraTerm, select **File** and then **New Connection...** from the menu bar, and then select **Serial**. Provide the **Port** corresponding to **mbed Serial Port**, as shown in Figure 23, and click **OK**.

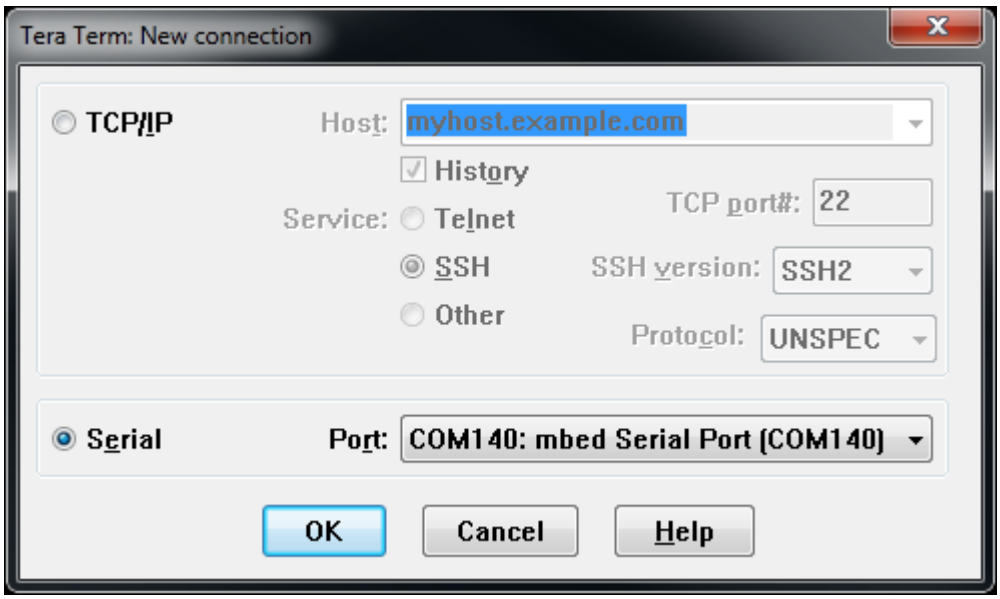

Figure 23. Connect to mbed Serial Port

4) Wait for 15 seconds and the board will boot up. Then observe the system log output from the terminal window, such as:

```
[2016-04-28 15:01:11.365]
PSRAM Memory Test Pass!!custom_setSFIExt
[2016-04-28 15:01:11.365]
NOR_init
[2016-04-28 15:01:11.365]
hal_flash_init
[2016-04-28 15:01:11.365]
gpdac_sram_power_down
[2016-04-28 15:01:11.365]
config bonding io register
```

2.3.6. Debugging with the LinkIt 2523 HDK from Microsoft Windows

This section describes how to debug a project built with the GCC compiler using openOCD debugger tool.

Before commencing project debugging on your LinkIt 2523 HDK, install supporting software on Windows OS.

1) Download openocd-0.9.0 from here and unzip it into `<openocd_root>` folder.
   a) Download the GCC toolchain from here for your Windows version and unzip it into `<gcc_root>` folder.
2) Install the mbed serial port driver, if the serial port driver is not installed (see MT2523 Flash Tool User's Guide).
3) Get a patched `openocd.exe`.
   a) Find minGW patch file named "mingw.tgz" located under `<sdk_root>/tools/config/mt2523_hdk/openocd_config` in SDK release package.
   b) Run the shell script on Linux OS to get the `mingw.tgz`.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

c) Copy the `mingw.tgz` to Windows OS and unzip it to `<mingw_root>` folder, then copy the files `<mingw_root>\bin to <openocd_root>\bin` folder.

d) Download a dynamic linked library (dll) file and copy it into `<openocd_root>\bin` folder.

You can debug a project once it's built and downloaded to your HDK.

4) Create a board configuration file named `mt2523.cfg` and copy the following content to the file.

```
puts "Load MT2523(CMSIS-DAP) configuration"

# Use CMSIS-DAP debug interface
source [find interface/cmsis-dap.cfg]

source [find target/swj-dp.tcl]

# Setup variables
set _CPUTAPID 0x3ba02477
set _CHIPNAME MT2523
set _TARGETNAME $_CHIPNAME.cm4

# Create DAP instance
swj_newdap $_CHIPNAME cpu -irlen 4 -expected-id $_CPUTAPID

# Create target instance
target create $_TARGETNAME cortex_m -endian little -chain-position
$_CHIPNAME.cpu

$_TARGETNAME configure -event gdb-attach {
      targets $_TARGETNAME
      halt
}

$_TARGETNAME configure -event gdb-detach {
      targets $_TARGETNAME
      resume
}

# Setup SWD frequence
adapter_khz 1000

puts "Load MT2523(CMSIS-DAP) configuration done."
```

5) Save the board configuration file into `<openocd_root>\share\openocd\scripts\board` folder.

Start debugging with LinkIt 2523 HDK:

1) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>`, for example, `<sdk_root>\out\mt2523_hdk\iot_sdk_demo\iot_sdk_demo.elf`.

2) Open the command window for openOCD.

3) Change the directory in the command window to openOCD tool folder, such as `<openocd_root>\bin`.

4) Remove the micro-USB cables to completely power off the board.

5) Reconnect the micro-USB cable to **MK20 USB** port to power on the board.

6) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\mt2523.cfg
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

7) Open the command window for GNU project debugger (GDB).

8) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.

9) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\iot_sdk_demo.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have an openOCD debugger running on your system.

Note, openOCD is a free third-party debugging tool (GPL license), for any issues or troubleshooting refer to openOCD official forum. In addition, openOCD debugging cannot work if the system enters into sleep mode, for more detail information, please refer to the document Airoha IoT SDK for RTOS Power Mode Developers Guide under `<sdk_root>/doc`.

2.4. Developing on MT7682 HDK

2.4.1. Configuring the MT7682 HDK

MT7682 HDK includes a main board and a MT7682 stamp module. The MT7682 stamp module is mounted on the main board. The top view of the main board is shown in Figure 24.

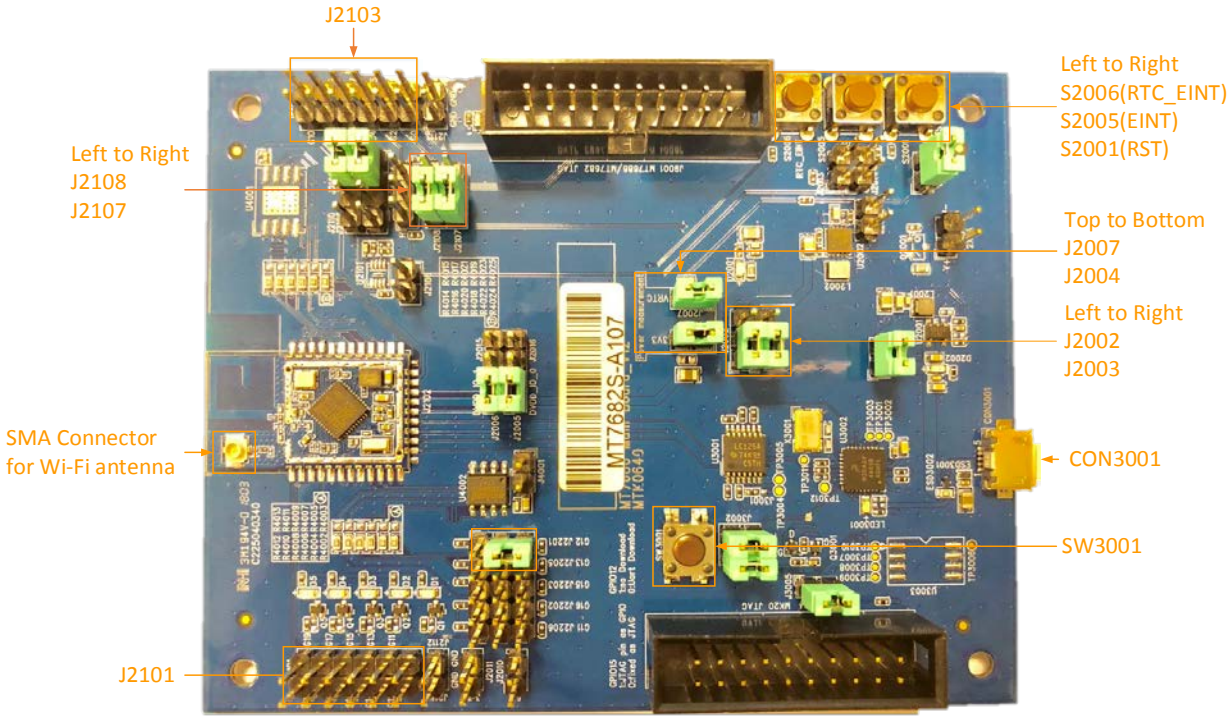

Figure 24. Jumpers and connectors on the MT7682 HDK

The description of pins (Figure 24) and their functionality is provided below.

1) **CON3001** is a USB connector to debug through UART, transmit and receive a signal and supply power from the PC. The USB connectivity with the PC is supported by the on-board MK20DX128VFM5.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

a) Set the jumpers **J2002**, **J2003**, **J2004** and **J2007** on, if the board is powered by a USB connector.

2) **S2005** enables the external interrupt (configured at **GPIO0**).

3) Press **S2001** to reset the system.

4) **Wi-Fi Antenna** is a PCB antenna. MT7682 stamp module is by default connected to the PCB antenna to transmit and receive RF signals.

The default configuration of the MT7682 HDK supports the following functionality:

1) Power supply. Attach a micro-USB connector to the **CON3001.**

2) Supports RTC interrupt.

3) Clock source — 32.768kHz source crystal clock for the RTC mode or external clock operating on 32.768 kHz.

4) XTAL at 40MHz.

5) Supports RTC mode.

The hardware settings of the stamp module are shown below:

1) XTAL at 40MHz.

2) Clock source — 32.768kHz source crystal clock for the RTC mode or external clock operating at 32.768kHz.

3) Supports RTC mode.

2.4.2. Installing MT7682 Flash Tool for MT7682 HDK

IoT Flash Tool is a flexible device flashing tool for application development on MT7682 HDK.

To install the IoT Flash Tool, navigate to MediaTek MOL website to download the IoT Flash Tool from here. You could find latest IoT Flash Tool by searching “IoT_Flash_Tool” in “Tool Name”. The tool is a setup free package, and the `FlashTool.exe` inside the folder can be executed directly.

Note, MT7682/MT7686 use the same IoT Flash Tool as MT2523.

2.4.3. Installing the MT7682 HDK drivers on Microsoft Windows

This section describes how to install MT7682 HDK drivers on PCs running Microsoft Windows. The **MK20 USB**, requires drivers for successful operation.

2.4.3.1. Install MK20 USB Driver

The MK20 USB port provides three different USB devices:

- CMSIS-DAP debug device. Keil and IAR IDEs use this device to download and debug programs.
- mbed serial port (COM port) that connects to the UART0 of the board. This serial port relays to the UART0 interface of the board and serves as a primary system log output interface for most of the SDK example projects.
- A virtual mass storage device to download a binary image to the board by simply copying the binary file to the storage device.

Only the mbed serial port requires additional driver installation. Install mbed serial port driver to use the USB serial port on Windows OS for debugging:

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

1) Download the IoT Flash Tool package from here. Install UART driver from "UART_Port_Driver" folder located under the Flash Tool's release folder.
2) Connect the board to the computer through **MK20 USB**.
3) Open Windows **Control Panel**, click **System** and:
   a) On Windows 7 and 8, click **Device Manager**.
   b) On Windows XP, click the **Hardware** tab and then **Device Manager**.
4) In **Device Manager**, navigate to **Ports (COM & LPT)** (see Figure 25).
5) A new COM device should appear under **Ports (COM & LPT)**, as shown in the figure below. Note the **COMx** port number of the **mbed Serial Port**. Use this port to receive system logs from the MT7682 HDK.

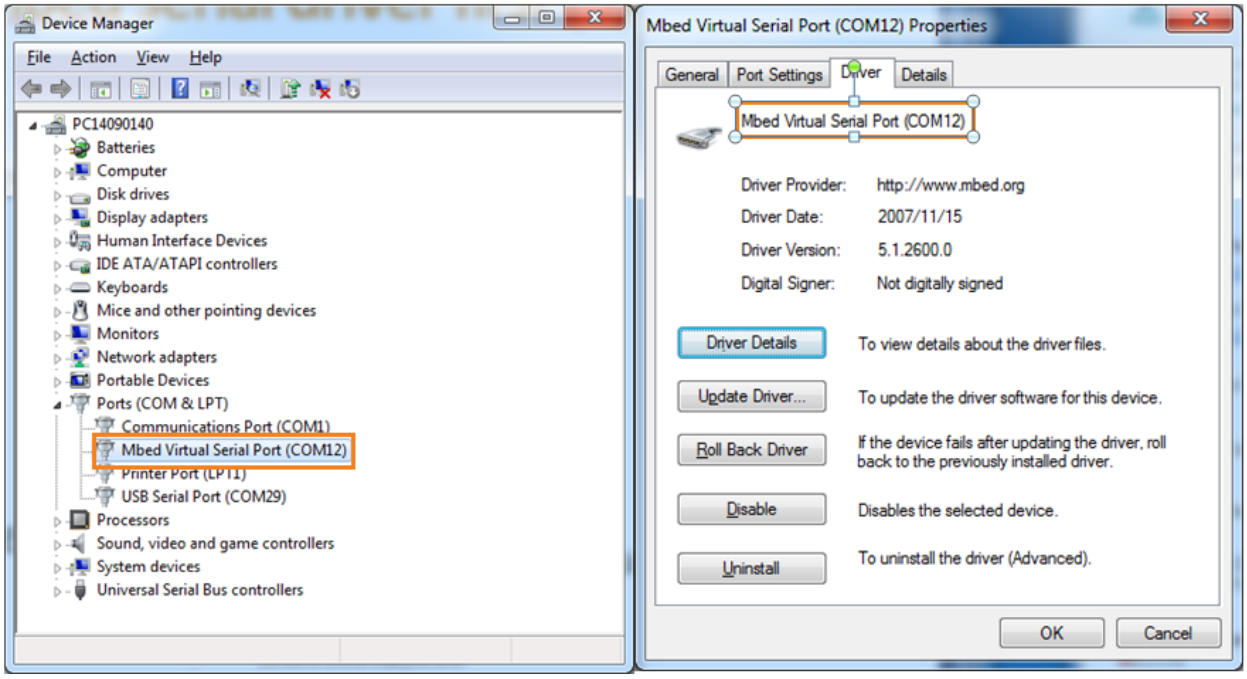

Figure 25. Installing the UART driver

2.4.4. Flashing the image to MT7682 HDK

6) Before using the IoT Flash Tool, make sure you have a pre-build project file (.cfg) or build your own project to get one (see 4), "Put the board configuration file under <openocd_root>\share\openocd\scripts\board.

Start debugging with AB155x EVK:

7) Copy the project .elf file from project Build folder to GCC tool path, such as <gcc_root> (for example, <sdk_root>\out\ab1552_evk\ earbuds_ref_design\debug\ earbuds_ref_design.elf.)
8) Open the command window for openOCD.
9) Change the directory in the command window to the openOCD tool folder, such as <openocd_root>\bin.
10) Disconnect the micro-USB cable from the board to completely power off the board.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

11) Use a simulator (such as J-Link) to connect to JTAG pin (TMS, TCLK, VCC and GND) or SWD pin (SWDIO, SWCLK, VCC and GND) of AB155x.

12) Reconnect the micro-USB cable to the board to power-on the board.

13) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\ab155x.cfg
```

14) Open the command window for GNU project debugger (GDB).

15) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.

16) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\earbuds_ref_design.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have the openOCD debugging software running on your system.

Note: openOCD is a free third-party debugging tool (GPL license). To resolve any issues or perform troubleshooting, please refer to the openOCD official forum. In addition, openOCD debugging cannot work if the system goes into sleep mode. For more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/mcu/doc`.

Building the project using the SDK").

There are two methods to flash the image to LinkIt HDK: using the IoT Flash Tool or using the device as a removable storage device.

2.4.4.1. Using the Flash Tool

To download the firmware to the target device, use the UART interface using **MK20 Port**:

1) Plug-in USB cable to **MK20 Port**.

2) Launch IoT Flash Tool, and click **Download** on the left panel of the main GUI.

3) Select the new port appeared after plug-in USB cable to **MK20 Port** from the **COM Port** drop down menu.

4) Click **Open** to provide the configuration file, which is usually named as `flash_download.cfg` and is generated after build process. If it loads successfully, **Download Information** will be displayed, including **Name**, **Length** and **File Path** of the firmware binary, as shown in Figure 26.

5) Click **Start** to download.

6) Click **Reset** key on board and then the process will start automatically.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

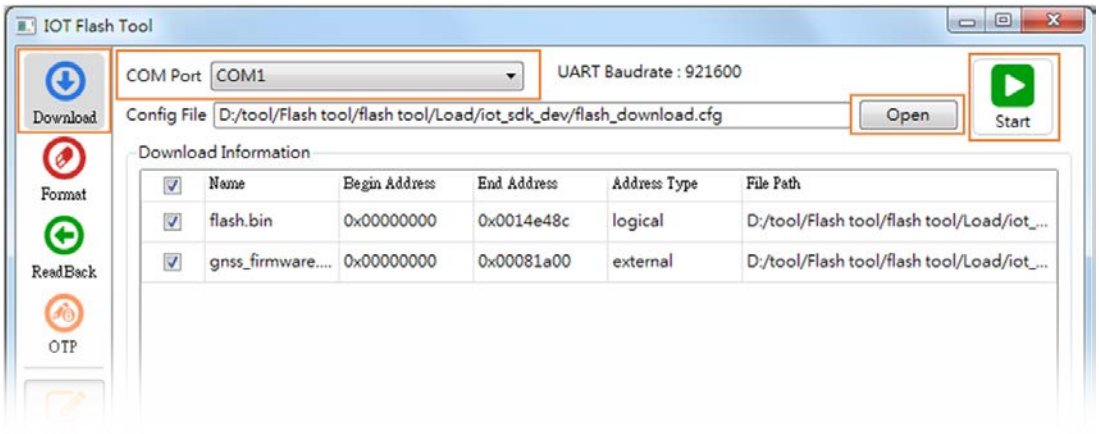

Figure 26. Download the firmware to a target device using USB connection

2.4.4.2. Using the MT7682 HDK as a removable storage device

To update the FreeRTOS binary only (example project binary: `mt7682_iot_sdk.bin`), use the HDK as a mass storage device according to the following steps:

1) Set the jumpers **J2107** pin 2 and pin 3, **J2108** pin2 and pin 3 on.
2) Power up the board with a micro-USB cable.
3) Navigate to **Computer** on your PC to check if a new mass storage named **MT7682** is available under **Removable Disk**, as shown in Figure 27.
4) Open the **MT7682** removable storage, then drag and drop the binary `mt7682_iot_sdk.bin` to complete downloading the image.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

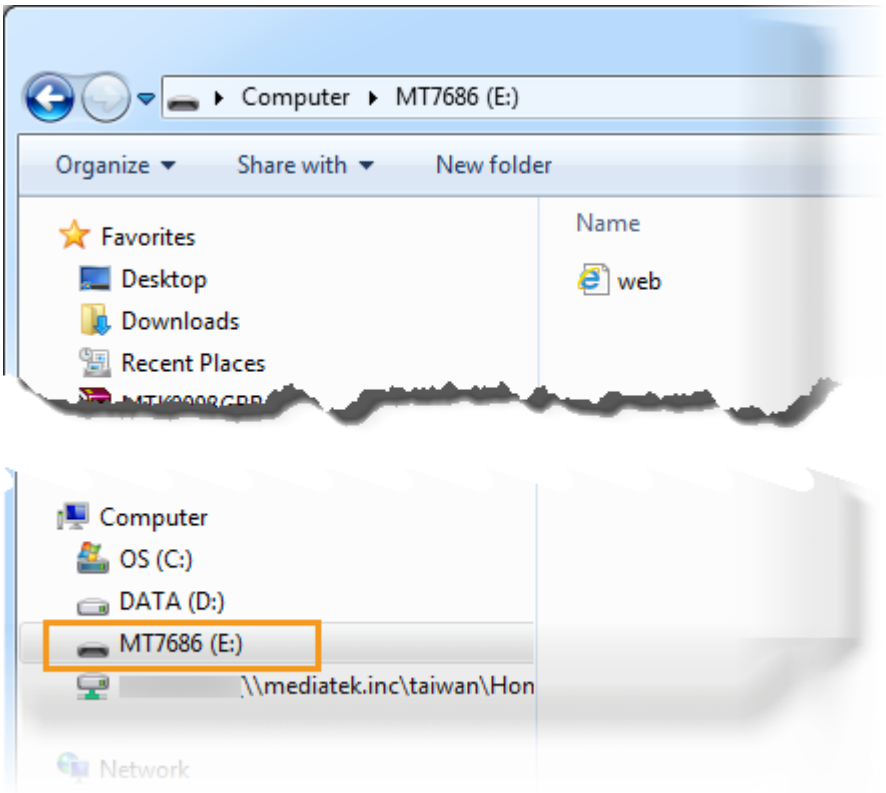

Figure 27. MT7682 removable storage detected

2.4.5. Running the project on MT7682 HDK

To run the project on MT7682 HDK after downloading a project binary, reboot the board:

1) Open the terminal application and connect to the mbed serial port. If you are using TeraTerm, select **File** and then **New Connection...** from the menu bar, and then select **Serial**. Provide the **Port** corresponding to **Mbed Virtual Serial Port**, as shown in Figure 28, and click **OK**.

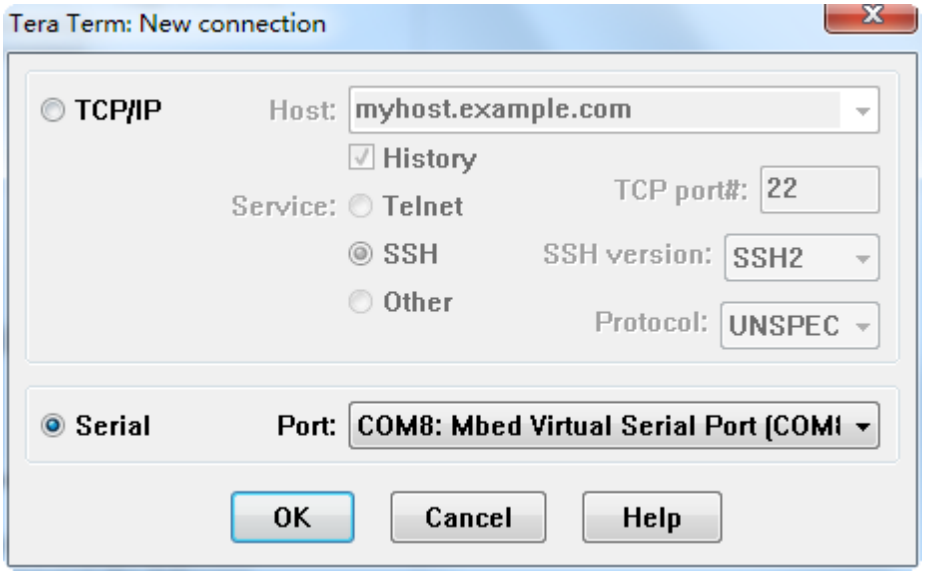

Figure 28. Connect to mbed serial port

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2) Press **Reset** key to reboot device.

3) The board will boot up. Then observe the system log output from the terminal window, such as:

```
[Sleep Management]sleep handle name : rx_eint

hal_sleep_manager_init start

manual load spm code
SPM Code loading Success

[Sleep Management]sleep handle name : CLI_Sleep

DCXO init

Xtal is 26M
DCXO efuse default value

rtc_dump_register[Just After Power On], RTC_POWERKEY1 = aeff, RTC_POWERKEY2
= ef9b, RTC_DIFF = 0, RTC_OSC32CON0 = 101

RTC_BBPU = f, RTC_LPD_CON = 10b, RTC_CON32CON1 = f07, RTC_CON32CON2 = 103,
RTC_WRTGR = 0, SRAMCON = ffff, CALI = 0

retention_flag = 0

normal bootup

Use 32k crystal

LPD occurred, rtc_wrtgr: 0, rtc_lpd_con: 10b

rtc_dump_register[Before Set Power Key], RTC_POWERKEY1 = aeff, RTC_POWERKEY2
= ef9b, RTC_DIFF = 0, RTC_OSC32CON0 = 101

RTC_BBPU = 0, RTC_LPD_CON = 100, RTC_CON32CON1 = f07, RTC_CON32CON2 = 3,
RTC_WRTGR = 0, SRAMCON = ff00, CALI = 0

rtc_dump_register[After Set Power Key], RTC_POWERKEY1 = a357, RTC_POWERKEY2
= 67d2, RTC_DIFF = 0, RTC_OSC32CON0 = 101

RTC_BBPU = 0, RTC_LPD_CON = 102, RTC_CON32CON1 = f07, RTC_CON32CON2 = 3,
RTC_WRTGR = 200, SRAMCON = ff00, CALI = 0

xosc frequency = 32767, RTC_CALI = 8

rtc_dump_register[hal_rtc_init done], RTC_POWERKEY1 = a357, RTC_POWERKEY2 =
67d2, RTC_DIFF = 0, RTC_OSC32CON0 = 106

RTC_BBPU = 0, RTC_LPD_CON = 2, RTC_CON32CON1 = fe7, RTC_CON32CON2 = 1,
RTC_WRTGR = 200, SRAMCON = ff00, CALI = 8

init_done, RTC_32K = 32767, EOSC = 0, DCXO = 0, XOSC = 32767

Move ROM data(0x8075acc) to (0x4259c00) 6614 bytes
connsys state 0, sys_config 0x00000000

[T: 0 M: common C: info F: system_init L: 333]: system initialize done.
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.4.6. Debugging with the MT7682 HDK from Microsoft Windows

This section describes how to debug a project built with the GCC compiler using openOCD debugger tool.

Before commencing project debugging, install supporting software on Windows OS.

1) Download openocd-0.9.0 from here and unzip it into <openocd_root> folder.
   a) Download GCC toolchain from here for your Windows version, and unzip it into <gcc_root> folder.
2) Install the mbed serial port driver, if the mbed serial port driver is not installed (see IoT Flash Tool Users Guide).
3) Create a board configuration file named mt7682.cfg and copy the following content to the file:

```
puts "Load MT7682 configuration"

source [find interface/jlink.cfg]
#source [find interface/cmsis-dap.cfg]

#source [find interface/ftdi/olimex-arm-usb-ocd-h.cfg]
transport select swd
source [find target/swj-dp.tcl]

set _CHIPNAME MT7682
set _TARGETNAME0 $_CHIPNAME.CM4
set _CPUTAPID 0x3ba02477

swj_newdap $_CHIPNAME DAP -irlen 4 -expected-id $_CPUTAPID

target create $_TARGETNAME0 cortex_m -chain-position $_CHIPNAME.DAP

adapter_khz 5000
reset_config srst_only


$_TARGETNAME0 configure -event reset-end {
global _TARGETNAME0
targets $_TARGETNAME0
enable_debug
echo "reset end"
}

$_TARGETNAME0 configure -event halted {
      targets $_TARGETNAME0
      mww 0xA2090000 0x11
}

$_TARGETNAME0 configure -event gdb-attach {
      targets $_TARGETNAME0
      mww 0xA2090000 0x11
      halt
}

$_TARGETNAME0 configure -event gdb-detach {
      targets $_TARGETNAME0
      resume
}

#enable debug
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
proc enable_debug {} {
      global _TARGETNAME0
      targets $_TARGETNAME0
      mww 0xE000EDF0 0xA05F0003
}

proc wdtreset {} {
      global _TARGETNAME0
      targets $_TARGETNAME0
      mww 0xA2090000 0x111
      mww 0xA2090010 0x156789ab
      mww 0xE000EDF0 0xA05F0003
      mww 0xA2090000 0x11
}

puts "MT7682 configuration done"
```

4) Place the board configuration file under `<openocd_root>\share\openocd\scripts\board.`

Start debugging with MT7682 HDK:

5) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>`, for example, `<sdk_root>\out\mt7682_hdk\iot_sdk_demo\iot_sdk_demo.elf`.
6) Open the command window for openOCD.
7) Change the directory in the command window to openOCD tool folder, such as `<openocd_root>\bin`.
8) Remove the micro-USB cables to completely power off the board.
9) Use Simulator(Such as J-Link) connect to JTAG pin(TMS, TCLK, VCC and GND) or SWD Pin(SWDIO, SWCLK, VCC and GND) of MT7682
10) Reconnect the micro-USB and power on the board.
11) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\mt7682.cfg
```

12) Open the command window for GNU project debugger (GDB).
13) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.
14) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\iot_sdk_demo.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have an openOCD debugger running on your system.

Note, openOCD is a free third-party debugging tool (GPL license), for any issues or troubleshooting refer to openOCD official forum. In addition, openOCD debugging cannot work if the system enters into sleep mode, for more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/doc`.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.5. Developing on LinkIt 7686 HDK

2.5.1. Configuring the LinkIt 7686 HDK

LinkIt 7686 HDK includes a main board and a MT7686 stamp module mounted on the main board. The top view of the main board is shown in Figure 29.

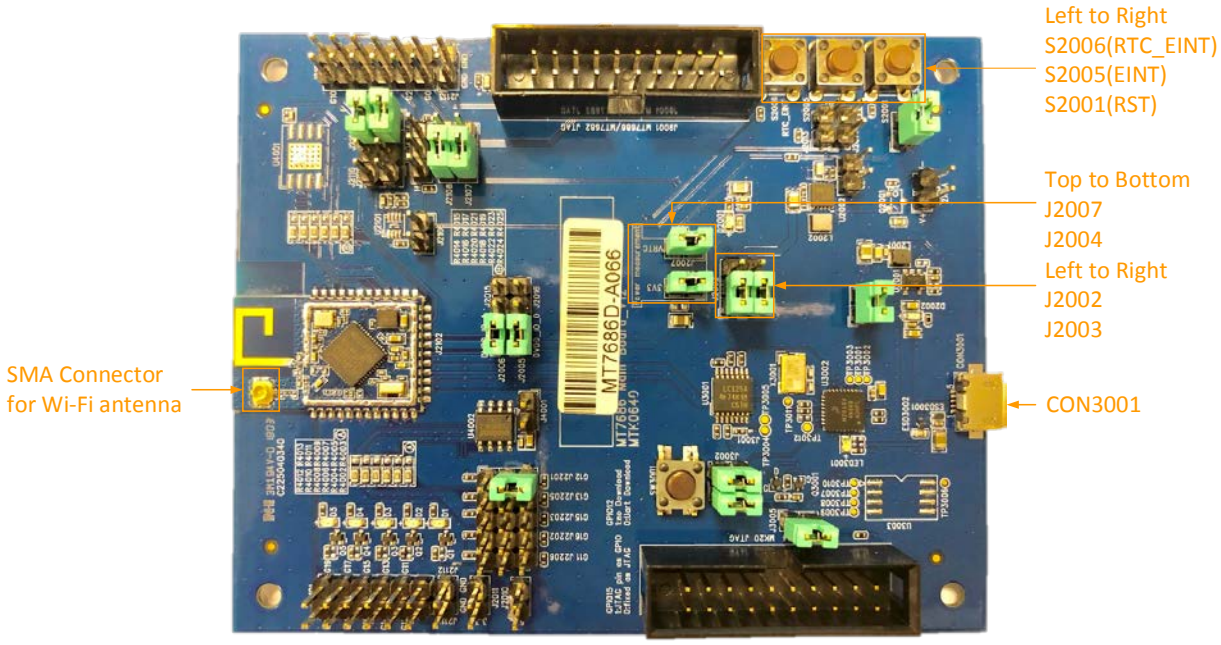

Figure 29. Front view of the LinkIt 7686 HDK

The description of pins (Figure 29) and their functionality is provided below.

1) **CON3001** is a USB connector to debug through UART, transmit and receive signals and supply power from the PC. The USB connectivity with the PC is supported by the on-board MK20DX128VFM5.
   a) Set the jumpers **J2002**, **J2003**, **J2004** and **J2007** on, if the board is powered by a USB connector.
2) **S2005** enables the external interrupt (configured at **GPIO0**).
3) Press **S2001** to reset the system.
4) **Wi-Fi Antenna** is a PCB antenna. MT7686 stamp module is by default connected to the PCB antenna to transmit and receive RF signals.

The default configuration of the LinkIt 7686 HDK supports the following functionality:

1) Power supply. Attach a micro-USB connector to the **CON3001.**
2) Supports RTC interrupt.
3) Clock source — 32.768kHz source crystal clock for the RTC mode or external clock operating on 32.768kHz.
4) XTAL at 40MHz.
5) Supports RTC mode.

The hardware settings of the stamp module are shown below:

1) XTAL at 40MHz.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2) Clock source — 32.768kHz source crystal clock for the RTC mode or external clock operating at 32.768kHz.

Supports RTC mode.

2.5.2. Installing MT7686 Flash Tool for LinkIt 7686 HDK

IoT Flash Tool is a flexible device flashing tool for application development on LinkIt 7686 HDK.

To install the IoT Flash Tool:

- Navigate to MediaTek MOL website to download the IoT Flash Tool from here.
- You could find latest IoT Flash Tool by searching “IoT_Flash_Tool” in “Tool Name”.

The tool is a setup free package, and the `FlashTool.exe` inside the folder can be executed directly.

Note, MT7682/MT7686 use the same IoT Flash Tool as MT2523.

2.5.3. Installing the LinkIt 7686 HDK drivers on Microsoft Windows

This section describes how to install LinkIt 7686 HDK drivers on PCs running Microsoft Windows. Follow the instructions below to install it.

2.5.3.1. Install MK20 USB Driver

The MK20 USB port provides three different USB devices:

- CMSIS-DAP debug device. Keil and IAR IDEs use this device to download and debug programs.
- mbed serial port (COM port) that connects the UART0 of the board. This serial port relays to the UART0 interface of the board and serves as the primary system log output interface for most of the SDK example projects.
- A virtual mass storage device to download a binary image to the board by simply copying the binary file to the storage device.

Only the mbed serial port requires additional driver installation. Install mbed serial port driver to use the USB serial port on Windows OS for debugging:

1) Download IoT Flash Tool package from here. Install UART driver from “`UART_Port_Driver`” folder located under the Flash Tool’s release folder.
2) Connect the board to the computer through **MK20 USB**.
3) Open Windows **Control Panel**, click **System** and:
   - On Windows 7 and 8, click **Device Manager**.
   - On Windows XP, click the **Hardware** tab and then **Device Manager**.
4) In **Device Manager**, navigate to **Ports (COM & LPT)** (see Figure 30).
5) A new COM device should appear under **Ports (COM & LPT)**, as shown in the figure below. Note the **COMx** port number of the **mbed Serial Port**. Use this port to receive system logs from the LinkIt 7686 HDK.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

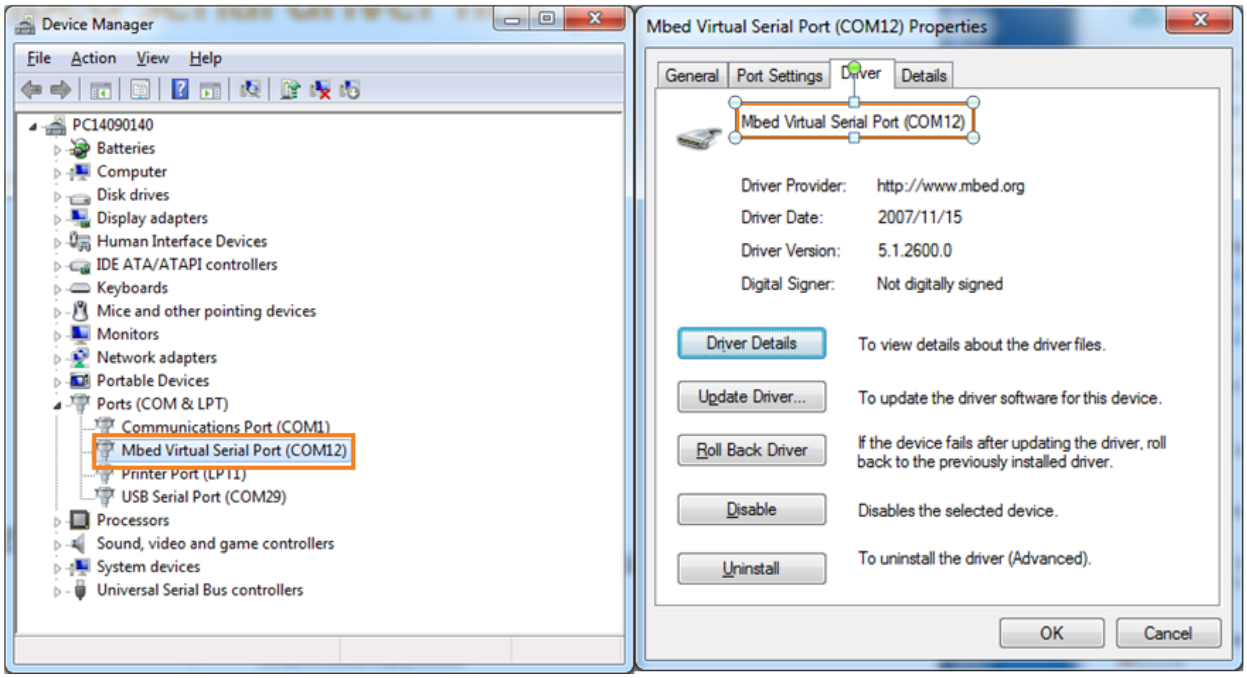

Figure 30. Installing the UART driver

2.5.4. Flashing the image to LinkIt 7686 HDK

3) Before using the IoT Flash Tool, make sure you have a pre-built project file (`.cfg`) or build your own project to get one (see section 4), "Put the board configuration file under `<openocd_root>\share\openocd\scripts\board`.

Start debugging with AB155x EVK:

4) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>` (for example, `<sdk_root>\out\ab1552_evk\ earbuds_ref_design\debug\ earbuds_ref_design.elf`.)
5) Open the command window for openOCD.
6) Change the directory in the command window to the openOCD tool folder, such as `<openocd_root>\bin`.
7) Disconnect the micro-USB cable from the board to completely power off the board.
8) Use a simulator (such as J-Link) to connect to JTAG pin (TMS, TCLK, VCC and GND) or SWD pin (SWDIO, SWCLK, VCC and GND) of AB155x.
9) Reconnect the micro-USB cable to the board to power-on the board.
10) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\ab155x.cfg
```

11) Open the command window for GNU project debugger (GDB).
12) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.
13) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\earbuds_ref_design.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have the openOCD debugging software running on your system.

Note: openOCD is a free third-party debugging tool (GPL license). To resolve any issues or perform troubleshooting, please refer to the openOCD official forum. In addition, openOCD debugging cannot work if the system goes into sleep mode. For more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/mcu/doc`.

Building the project using the SDK").

There are two methods to flash the image to LinkIt HDK: using the IoT Flash Tool or using the device as a removable storage.

2.5.4.1. Using the Flash Tool

To download the firmware to the target device, use the UART interface connected through **MK20 Port**, see Figure 29:

1) Plug-in a USB cable to **MK20 Port**.
2) Launch IoT Flash Tool, and click **Download** on the left panel of the main GUI.
3) Select the port corresponding to **MK20 Port** from the **COM Port** drop down menu.
4) Click **Open** to provide the configuration file, which is usually named as `flash_download.cfg` and is generated after build process. If it loads successfully, **Download Information** will be displayed, including **Name**, **Length** and **File Path** of the firmware binary, as shown in Figure 31.
5) Click **Start** to get ready for downloading.
6) Click **Reset** key on the board and then the process will start automatically.

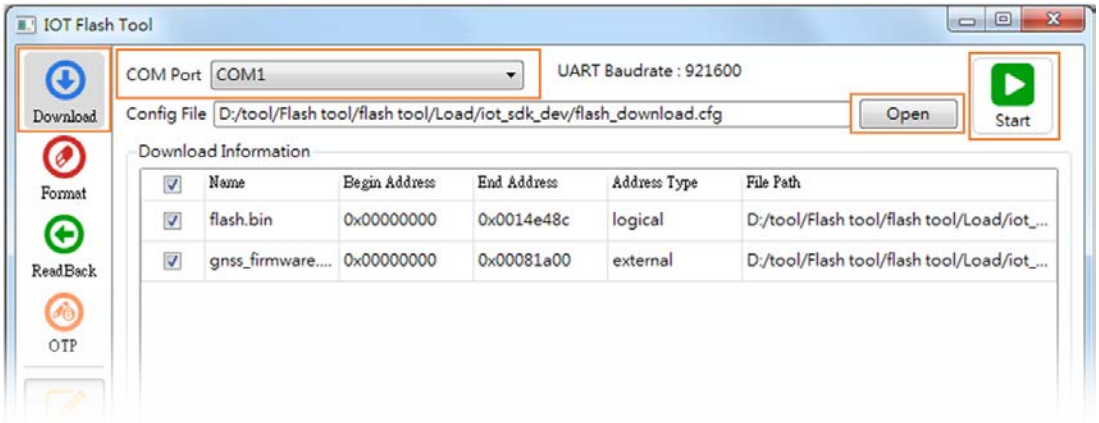

Figure 31. Download the firmware to a target device using USB connection

2.5.4.2. Using the LinkIt 7686 HDK as a removable storage device

To update the FreeRTOS binary only (example project binary: `mt7686_iot_sdk.bin`), use the HDK as a mass storage device:

1) Set the jumpers **J2107** pin 2 and pin 3, **J2108** pin2 and pin 3 on.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2) Power up the board with a micro-USB cable.

3) Navigate to **Computer** on your PC to check if a new mass storage named **MT7686** is available under **Removable Disk**, as shown in Figure 32.

4) Open the **MT7686** removable storage, then drag and drop the binary `mt7686_iot_sdk.bin` to complete downloading the image.

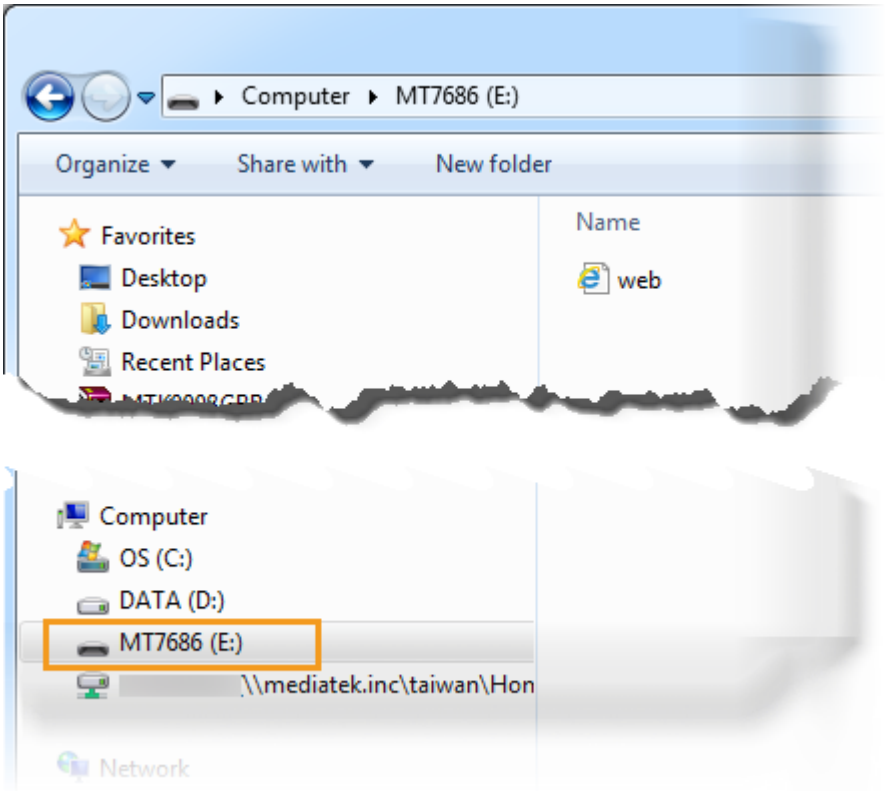

Figure 32. MT7686 removable storage detected

2.5.5. Running the project on LinkIt 7686 HDK

To run the project on LinkIt 7686 HDK after downloading a project binary:

1) Open the terminal application and connect to the mbed serial port. If you are using TeraTerm, select **File** and then **New Connection...** from the menu bar, and then select **Serial**. Provide the **Port** corresponding to **mbed Serial Port**, as shown in Figure 33 and click **OK**.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

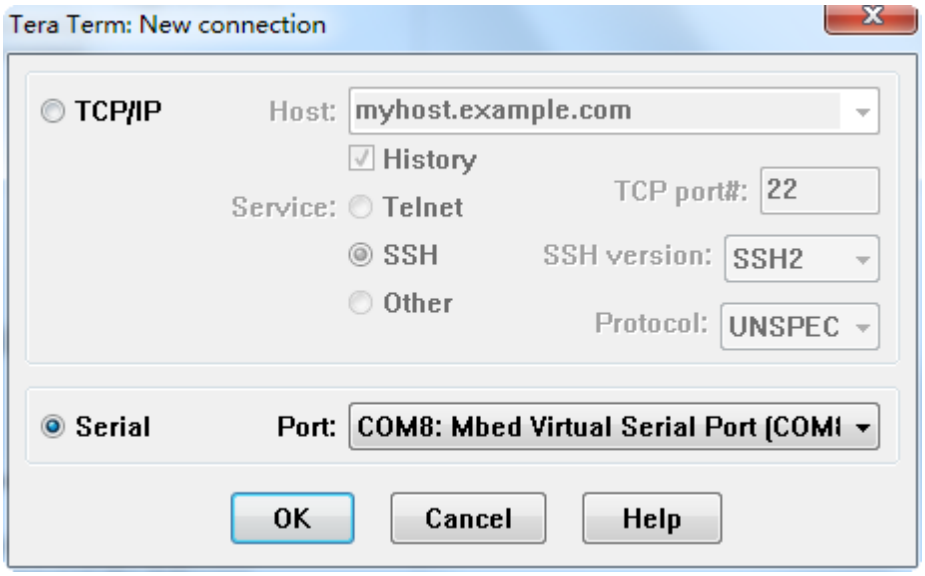

Figure 33. Connecting to mbed serial port

2) Press **Reset** key to reboot the board.

3) After boot up observe the system log output from the terminal window, such as:

```
rom init done
wifi fw init done
before iot init, iot_init=0x4417741
iot init done
        net pkt header pool : 68 entities, total 3808 bytes
        NET_BUF_128  : max used 128, free 4, total 4
        NET_BUF_256  : max used 256, free 10, total 10
        NET_BUF_512  : max used 512, free 12, total 12
        NET_BUF_1024  : max used 1024, free 0, total 0
        NET_BUF_2048  : max used 1600, free 32, total 32
        NET_BUF_4096  : max used 4096, free 0, total 0
net_pkt_show
wifi_firmware_init_task done
$ pAd1
MsgInwait=0, SpacAv=16
MsgInwait=0, SpacAv=16
macAddr: 0x42587e4, len = 6
0x0000 : ff ff ff ff ff ff
EntryLifeCheck=1024
        NORMAL MODE
        RMAC_RFCR = 0x1DE70A
Event 0x30 not handled
        Triplet[0] = [1, 14, 0]
[T: 96 M: common C: info F: wifi_init_done_handler L: 1053]: WiFi Init Done:
port = 0
```

2.5.6. Debugging with the LinkIt 7686 HDK from Microsoft Windows

This section describes how to debug a project built with the GCC compiler using openOCD debugger tool.

Before commencing project debugging, install supporting software on Windows OS.

1) Download openocd-0.9.0 from here and unzip it into <openocd_root> folder.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

a) Download GCC toolchain from here for your Windows version, and unzip it into <gcc_root> folder.

2) Install the mbed serial port driver, if the mbed serial port driver is not installed (see IoT Flash Tool Users Guide).

3) Create a board configuration file named `mt7686.cfg` and copy the following content to the file:

```
puts "Load MT7686 configuration"

source [find interface/jlink.cfg]
#source [find interface/cmsis-dap.cfg]

#source [find interface/ftdi/olimex-arm-usb-ocd-h.cfg]
transport select swd
source [find target/swj-dp.tcl]

set _CHIPNAME MT7686
set _TARGETNAME0 $_CHIPNAME.CM4
set _CPUTAPID 0x3ba02477

swj_newdap $_CHIPNAME DAP -irlen 4 -expected-id $_CPUTAPID

target create $_TARGETNAME0 cortex_m -chain-position $_CHIPNAME.DAP

adapter_khz 5000
reset_config srst_only


$_TARGETNAME0 configure -event reset-end {
global _TARGETNAME0
targets $_TARGETNAME0
enable_debug
echo "reset end"
}

$_TARGETNAME0 configure -event halted {
      targets $_TARGETNAME0
      mww 0xA2090000 0x11
}

$_TARGETNAME0 configure -event gdb-attach {
      targets $_TARGETNAME0
      mww 0xA2090000 0x11
      halt
}

$_TARGETNAME0 configure -event gdb-detach {
      targets $_TARGETNAME0
      resume
}

#enable debug
proc enable_debug {} {
      global _TARGETNAME0
      targets $_TARGETNAME0
      mww 0xE000EDF0 0xA05F0003
}

proc wdtreset {} {
      global _TARGETNAME0
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
      targets $_TARGETNAME0
      mww 0xA2090000 0x111
      mww 0xA2090010 0x156789ab
      mww 0xE000EDF0 0xA05F0003
      mww 0xA2090000 0x11
}

puts "MT7686 configuration done"
```

4) Place the board configuration file under `<openocd_root>\share\openocd\scripts\board.`

Start debugging with MT7686 HDK:

5) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>`, for example, `<sdk_root>\out\mt7686_hdk\iot_sdk_demo\iot_sdk_demo.elf`.
6) Open the command window for openOCD.
7) Change the directory in the command window to openOCD tool folder, such as `<openocd_root>\bin`.
8) Remove the micro-USB cables to completely power off the board.
9) Use Simulator(Such as J-Link) connect to JTAG pin(TMS, TCLK, VCC and GND) or SWD Pin(SWDIO, SWCLK, VCC and GND) of MT7686
10) Reconnect the micro-USB and power on the board.
11) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\mt7686.cfg
```

12) Open the command window for GNU project debugger (GDB).
13) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.
14) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\iot_sdk_demo.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have an openOCD debugger running on your system.

Note, openOCD is a free third-party debugging tool (GPL license), for any issues or troubleshooting refer to openOCD official forum. In addition, openOCD debugging cannot work if the system enters into sleep mode, for more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/doc`.

2.6. Developing on LinkIt 7697 HDK

Learn how to install board drivers and flash images to the LinkIt 7697 HDK at https://docs.labs.mediatek.com/resource/mt7687-mt7697/en/get-started.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.7. Developing on AB155x EVK

2.7.1. Configuring the AB155x EVK

The AB155x EVK has two separate major boards. One board is the EVK main board. The other board is an adaptor board for each chipset. There are two kinds of adaptors available: AB1558/6 and AB1555. The front view of the AB155x EVK with the AB1555 adaptor board is shown in Figure 34.

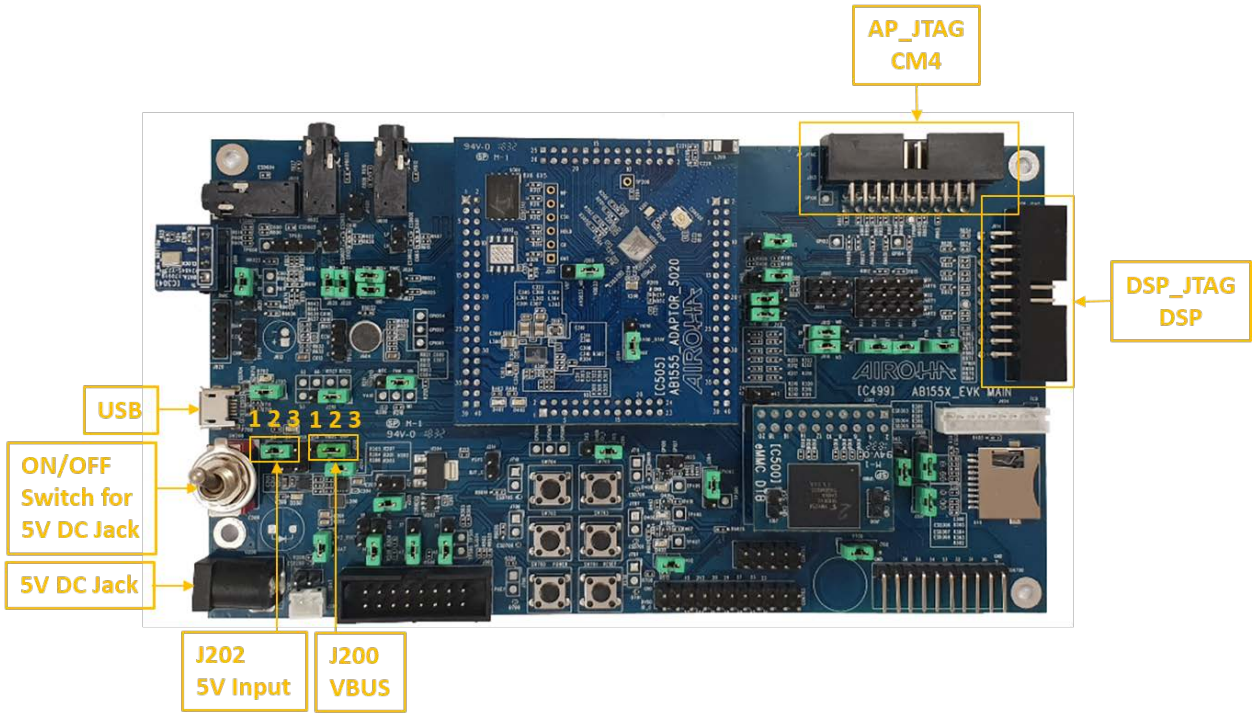

Figure 34. Front view of the AB155x EVK

The EVK can be powered by either USB VBUS or the 5V DC jack. Set the jumpers J202 2-3 on for USB VBUS (5V_USB).

A micro-USB connector is reserved on the left-side of the EVK, this port can also be used as the firmware download port. Set jumpers J200 1-2 on, the VBUS power source is from USB VBUS.

AB155x has reserved two JTAG debugging interfaces, AP_JTAG is used for embedded CM4 and DSP_JTAG is used for the embedded DSP. You can directly connect JTAG JIG with these connectors for debugging.

2.7.2. Installing AB155x Flash Tool for AB155x EVK

IoT Flash Tool is a flexible device flashing tool for application development on AB155x EVK.

To install the IoT Flash Tool:

- The IoT Flash Tool is located in <sdk_root>/tools/pc_tool/IOT_Flash_Tool
- The IoT Flash Tool is also available through MOL. Go here to download the IoT Flash Tool from the MediaTek MOL website.
  - Search for “IoT_Flash_Tool” in “Tool Name” to find the latest version of IoT Flash Tool.

The tool is a setup free package. You can start the Flash Tool by clicking the `FlashTool.exe` inside the folder.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Note, AB155x use the same IOT Flash Tool as MT2523/MT7682/MT7686.

2.7.3. Installing the AB155x EVK drivers on Microsoft Windows

This section describes how to install AB155x EVK USB drivers on PCs running Microsoft Windows. Complete the following procedure to install them.

To install the MediaTek USB Port driver AB155x **USB** port on the EVK on Windows 7 or other Windows:

1) Install the MediaTek USB Port driver from `MS_USB_ComPort_Driver/v3.16.46.1` folder located in AB155x _FlashTool folder.
2) Execute `InstallDriver.exe` to install the driver.
3) Use a USB cable to connect the AB155x **USB** connector on the AB155x EVK to your computer's USB port.

2.7.4. Flashing the image to AB155x EVK

15) Before using the IoT Flash Tool, it is necessary to use a pre-built project file (`.cfg`) or build your own project to get one (see 4), "Put the board configuration file under `<openocd_root>\share\openocd\scripts\board.`

Start debugging with AB155x EVK:

16) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>` (for example, `<sdk_root>\out\ab1552_evk\ earbuds_ref_design\debug\ earbuds_ref_design.elf`.)
17) Open the command window for openOCD.
18) Change the directory in the command window to the openOCD tool folder, such as `<openocd_root>\bin`.
19) Disconnect the micro-USB cable from the board to completely power off the board.
20) Use a simulator (such as J-Link) to connect to JTAG pin (TMS, TCLK, VCC and GND) or SWD pin (SWDIO, SWCLK, VCC and GND) of AB155x.
21) Reconnect the micro-USB cable to the board to power-on the board.
22) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\ab155x.cfg
```

23) Open the command window for GNU project debugger (GDB).
24) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.
25) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\earbuds_ref_design.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have the openOCD debugging software running on your system.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Note: openOCD is a free third-party debugging tool (GPL license). To resolve any issues or perform troubleshooting, please refer to the openOCD official forum. In addition, openOCD debugging cannot work if the system goes into sleep mode. For more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/mcu/doc`.

Building the project using the SDK").

To download the firmware to the target device, use the **AB155x USB** interface:

1) Power off the target (you must disconnect the USB cable).
2) Launch IoT Flash Tool, and click **Download** on the left panel of the main GUI.
3) Select **USB** from the **COM Port** drop down menu. If you do not have the adapter or battery, click the **Enable Download without Battery** option.

Click **Open** to provide the configuration file, which is usually named as `flash_download.cfg` and is generated after build process. If it loads successfully, **Download Information** is displayed, including **Name**, **Length** and **File Path** of the firmware binary.

4) Click **Start** to start downloading.
5) Connect the USB cable to power on the HDK through the **AB155x USB** connector. The process automatically starts.

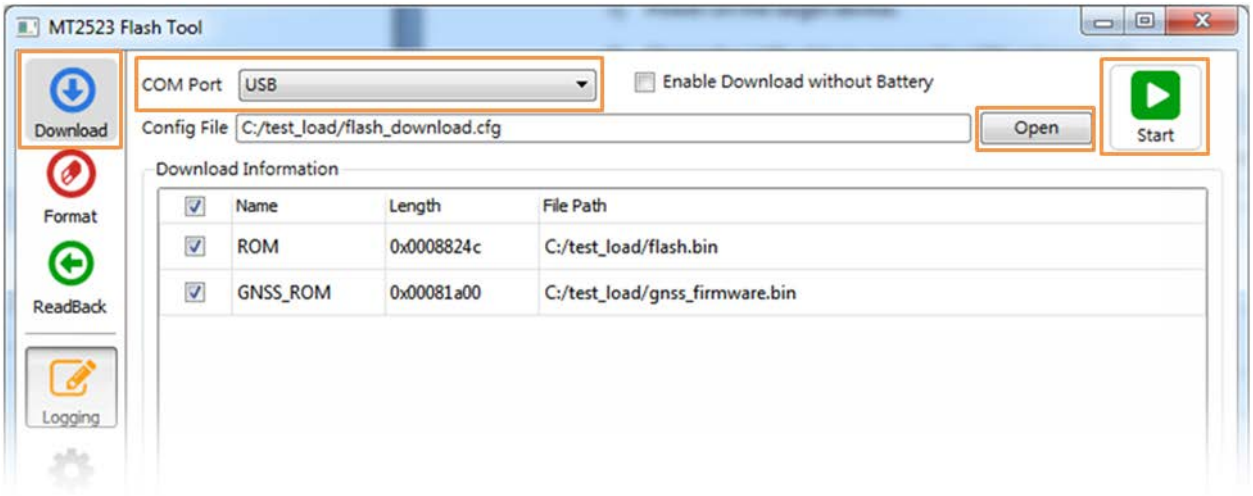

Figure 35. Download the firmware to a target device using USB connection

2.7.5. Running the project on AB155x EVK

You must complete the following procedure to use Logging tool:

1) Launch Logging tool.
2) Click "Serial port" as shown in Figure 36.
3) Set the Serial port & Baud Rate is shown in Figure 37.
4) Click "Start" as shown in Figure 38.
5) Reset the target. The log is shown in the Log window.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

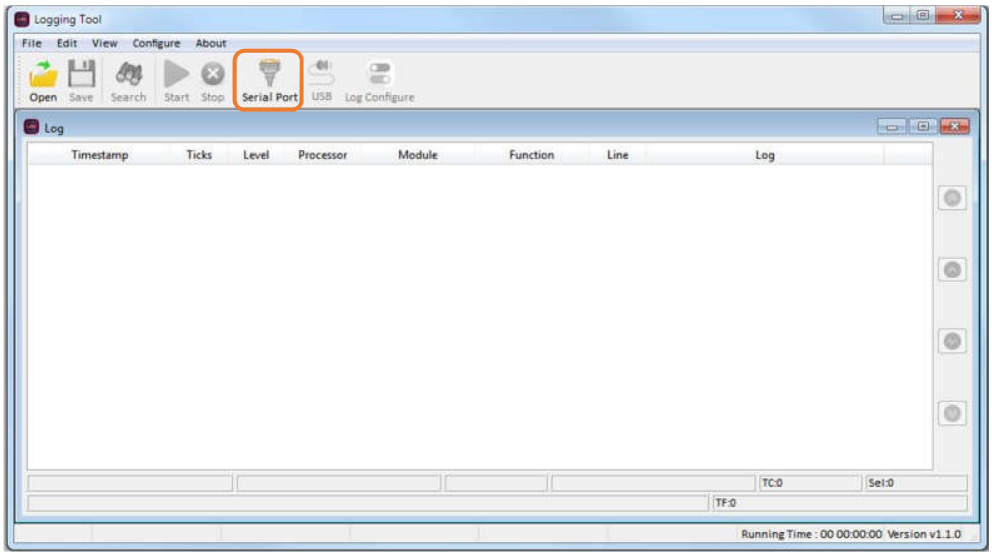

Figure 36. Serial Port Button

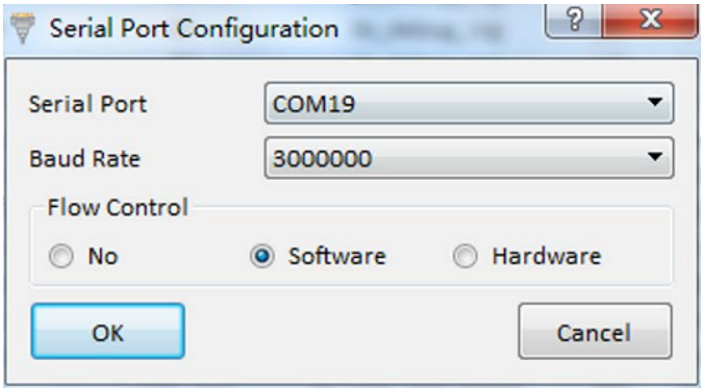

Figure 37. Serial Port Configuration Dialog

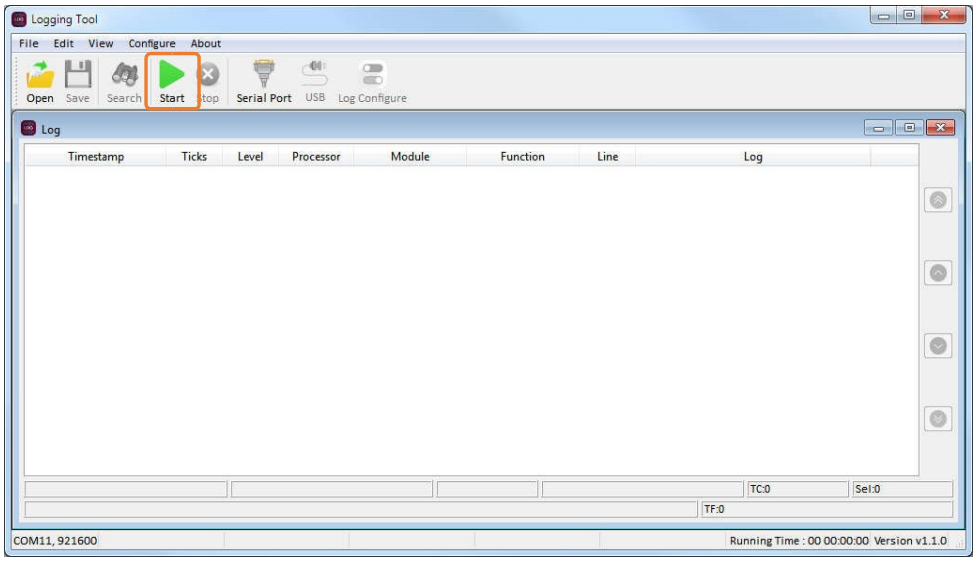

Figure 38. Start Button

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.7.6. Debugging with the AB155x EVK from Microsoft Windows

This section shows how to debug a project built with the GCC compiler using openOCD debugger tool.

Before commencing project debugging, install the supporting software on Windows OS.

1) Download openocd-0.9.0 from here and unzip it into the `<openocd_root>` folder.
   a) Download the GCC toolchain for your specific version of Windows from here, and unzip it into the `<gcc_root>` folder.
2) Install the mbed serial port driver, if the mbed serial port driver is not installed (see section 2.7.3 Installing the AB155x EVK drivers on Microsoft Windows).
3) Create a board configuration file named `ab155x.cfg` and copy the following content to the file:

```
puts "Load AB155x configuration"

#source [find interface/cmsis-dap.cfg]
source [find interface/jlink.cfg]
transport select swd
source [find target/swj-dp.tcl]

set _CHIPNAME AB155x
set _TARGETNAME $_CHIPNAME.CM4
set _CPUTAPID 0x3ba02477

swj_newdap $_CHIPNAME DAP -irlen 4 -expected-id $_CPUTAPID
target create $_TARGETNAME cortex_m -chain-position $_CHIPNAME.DAP

adapter_khz 1000
reset_config srst_only

$_TARGETNAME configure -event gdb-attach {
    global _TARGETNAME
      targets $_TARGETNAME
      halt
}

$_TARGETNAME configure -event gdb-detach {
    global _TARGETNAME
      targets $_TARGETNAME
      resume
}

puts "AB155x configuration done"
```

4) Put the board configuration file under `<openocd_root>\share\openocd\scripts\board.`

Start debugging with AB155x EVK:

5) Copy the project `.elf` file from project `Build` folder to GCC tool path, such as `<gcc_root>` (for example, `<sdk_root>\out\ab1552_evk\ earbuds_ref_design\debug\ earbuds_ref_design.elf`.)
6) Open the command window for openOCD.
7) Change the directory in the command window to the openOCD tool folder, such as `<openocd_root>\bin`.
8) Disconnect the micro-USB cable from the board to completely power off the board.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

9) Use a simulator (such as J-Link) to connect to JTAG pin (TMS, TCLK, VCC and GND) or SWD pin (SWDIO, SWCLK, VCC and GND) of AB155x.

10) Reconnect the micro-USB cable to the board to power-on the board.

11) Run the command to start the openOCD.

```
openocd.exe -s ..\share\openocd\scripts -f board\ab155x.cfg
```

12) Open the command window for GNU project debugger (GDB).

13) Change the directory in the command window to tool folder, such as `<gcc_root>\bin`.

14) Run the command to start the GDB.

```
arm-none-eabi-gdb.exe <gcc_root>\earbuds_ref_design.elf
(gdb) target remote localhost:3333
(gdb) monitor reset init
(gdb) load
(gdb) info registers
(gdb) x/10i $pc
```

You now have the openOCD debugging software running on your system.

Note: openOCD is a free third-party debugging tool (GPL license). To resolve any issues or perform troubleshooting, please refer to the openOCD official forum. In addition, openOCD debugging cannot work if the system goes into sleep mode. For more detail information, please refer to the document LinkIt for RTOS Power Mode Developers Guide under `<sdk_root>/mcu/doc`.

2.8. Building the project using the SDK

This section provides detailed information about how to set up the SDK build environment with the default GCC in the Linux OS. If you want to build the project on Microsoft Windows, you can set up a virtual machine (e.g., VMWare) with the Ubuntu Linux operating system installed or use the MinGW cross-compilation tool (Please refer to Airoha_IoT_SDK_GCC_Build_Environment_Guide.pdf for more information).

Note:

- Please note that the build performance in a virtual machine or MinGW environment is not as fast or efficient as the build performance in native Linux OS.
- There is currently no support for MinGW cross-compilation for AB155x series.

2.8.1. Installing the SDK build environment on Linux

The default GCC compiler provided in the SDK is required to run on Linux OS. The following description is based on the Ubuntu 14.04 LTS environment.

Note: The Airoha IoT SDK can be used on any edition of Linux OS. The default GCC compiler provided in the SDK is based on the 32-bit architecture.

Before building the project, verify that you've installed the required toolchain for your build environment, as shown in Table 13.

Table 13. Recommended build environment

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

Item	Description
OS	Linux OS
make	GNU make 3.81
Compiler	Linaro GCC Toolchain for ARM Embedded Processors 4.8.4

The following command downloads and installs the basic building tools on Ubuntu.

```
sudo apt-get install build-essential
```

Note, a compilation error occurs when building the Airoha IoT SDK with the default GCC cross compiler on a 64-bit system without installing the package to support the 32-bit executable binary, as shown below.

```
/bin/sh: 4: tools/gcc/gcc-arm-none-eabi/bin/arm-none-eabi-gcc: not found
```

The commands to install the basic build tools and the package for supporting 32-bit binary executable on the Ubuntu 14.04 are shown below.

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6-i386
```

The watch projects using TouchGFX framework require installing Ruby to generate resource files. Find more information on how to install Ruby in `<sdk_root>\project\mt2523_watch\apps\watch_demo\readme.txt`

Install the SDK package according to the instructions at `<sdk_root>\readme.txt`. The default installation path of the GCC compiler is `<sdk_root>\tools\gcc`, and the compiler settings are in the `<sdk_root>\.config` configuration file.

2.8.2. Methods to build a project

This section describes four methods for building a project:

- Build a project from the SDK root directory.
- Build AB155x series CM4 and DSP projects together from the SDK root directory.
- Build a project from the project GCC configuration directory.
- Build a DSP project from the SDK root directory. Please first refer to Airoha_IoT_SDK_for_BT_Audio_DSP_Get_Started_Guide under <sdk_root>/dsp/doc to set up the DSP toolchain environment.

Note: Build a DSP project from the SDK root directory only for AB155x series.

2.8.2.1. Building the project from the SDK root directory

Build the project using the script at `<sdk_root>/build.sh`. Please note that this does not apply to the AB155x series. For more information about the script, please navigate to the SDK's root directory and execute the following command:

```
cd <sdk_root>
./build.sh
```

For AB155x series, you must first navigate to `<sdk_root>/mcu` and execute the following command:

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
cd <sdk_root>/mcu
./build.sh
```

Note: The co-build script for AB155x series is in `<sdk_root>/build.sh`. It can build both the CM4 and DSP projects. Please refer to section 2.8.2.2, Building AB155x series CM4 and DSP projects together from the SDK root directory, for more information.

The outcome is:

```
============================================================
Build Project
============================================================
Usage: ./build.sh <board> <project> [bl|clean] <argument>

Example:
  ./build.sh mt7687_hdk iot_sdk_demo
  ./build.sh mt7687_hdk iot_sdk_demo bl     (build with bootloader)
  ./build.sh clean                    (clean folder: out)
  ./build.sh mt7687_hdk clean         (clean folder: out/mt7687_hdk)
  ./build.sh mt7687_hdk iot_sdk_demo clean  (clean folder:
out/mt7687_hdk/iot_sdk_demo)

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 options. 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
```

- List all available boards and projects.

Run the command to show all available boards and projects:

```
./build.sh list
```

The available boards and projects are listed based on the related configuration files under `<sdk_root>/config/project/<board>/<project>` folder. The console output is shown below.

```
================================================
Available Build Projects:
================================================
  mt2523_hdk
    accdet_detect_earphone_status
    aes_encrypt_decrypt_data
    atci_register_command
    audio_mp3_play
    audio_play_1k_tone
    ...
  mt2523_watch
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
    watch_demo
  mt7697_hdk
    bootloader
    freertos_initialize_main_features
    iot_sdk
```

- 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 on the LinkIt 2523 HDK, run the following build command.

```
./build.sh mt2523_hdk iot_sdk_demo
```

The standard output in the terminal window:

```
$ ./build.sh mt2523_hdk iot_sdk_demo
Build board:mt2523_hdk project:iot_sdk_demo
platform=MINGW32_NT-6.1
/usr/home/ user_name/SDK_vn.n.n
FEATURE=feature.mk
SENSOR_FUSION_ALGO
is ../../../../../middleware/MTK/sensor_subsys/fusion_algo/co
mmon/lib_core
make: Entering directory
`/usr/home/user_name/SDK_vn.n.n/project/mt2523_hdk/apps/
iot_sdk_demo/GCC'
  ...
```

The output files are then put in the `<sdk_root>/out/mt2523_hdk/iot_sdk_demo/` folder.

- 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
```

Note: The bootloader download mechanism is slightly different on LinkIt 2523 HDK from that on the LinkIt 76x7 HDK. On LinkIt 2523 HDK the bootloader image is combined with the project image file into a new image file named as `flash.bin`.

To build the project on the LinkIt 2523 HDK:

```
./build.sh mt2523_hdk iot_sdk_demo bl
```

The output image file of the project and the bootloader, along with the merged image file `flash.bin`, is put in the `<sdk_root>/out/mt2523_hdk/iot_sdk_demo` folder.

To build the project on the LinkIt 76x7 HDK:

```
./build.sh mt7687_hdk iot_sdk_demo bl
```

The output image file of the project and the bootloader are put in the `<sdk_root>/out/mt7687_hdk/iot_sdk_demo` folder.

- Clean the out folder

The build script `<sdk_root>/build.sh` provides options to remove the generated output files as follows.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

1) Clean the <sdk_root>/out folder.

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

2) Clean the <sdk_root>/out/<board> folder

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

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

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

2.8.2.2. Building AB155x series CM4 and DSP projects together from the SDK root directory

Note, the information in this section only applies to the AB155x series. Please refer to Airoha_IoT_SDK_for_BT_Audio_DSP_Get_Started_Guide under <sdk_root>/dsp/doc to first set up the DSP toolchain environment before building projects.

Build CM4 and DSP projects using the script in <sdk_root>/build.sh. To find more information about the script, navigate to the SDK root directory and execute the following command:

```
cd <sdk_root>
./build.sh
```

The outcome is:

```
============================================================
Build Project
============================================================
Usage: ./build.sh <board> <project> [clean] <argument>

Example:
       ./build.sh ab155x_evk earbuds_ref_design
       ./build.sh ab155x_evk earbuds_ref_design -fm=feature_ab1552_evk.mk
       ./build.sh clean
        (clean folder: out)
       ./build.sh ab155x_evk clean
        (clean folder: out/ab155x_evk)
       ./build.sh ab155x_evk earbuds_ref_design clean
        (clean folder: out/ab155x_evk/earbuds_ref_design)

Argument:
       -fm=<feature makefile>
           Replace feature.mk with other makefile for mcu. For example,
           the feature_example.mk is under project folder,
           -fm=feature_example.mk will replace feature.mk with
           feature_example.mk.

       -fd0=<feature makefile>
           Replace feature.mk with other makefile for dsp0. For example,
           the feature_example.mk is under project folder,
           -fd0=feature_example.mk will replace feature.mk with
           feature_example.mk.

       -fd1=<feature makefile>
           Replace feature.mk with other makefile for dsp1. For example,
           the feature_example.mk is under project folder,
           -fd1=feature_example.mk will replace feature.mk with
           feature_example.mk.

============================================================
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

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

- List all available boards and projects.

Run the command to show all available boards and projects:

```
./build.sh list
```

The available boards and projects are listed below.

```
============================================================
Available Build Projects:
============================================================
ab155x_evk
  earbuds_ref_design
      cm4: earbuds_ref_design
      dsp0: dsp0_headset_ref_design
```

- 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 AB155x EVK, run the following build command:

```
./build.sh ab155x_evk earbuds_ref_design
```

The standard output in the terminal window is as follows:

```
$./build.sh ab155x_evk earbuds_ref_design
cd /usr/home/user_name/SDK_vn.n.n /dsp
============================================================
Start DSP0 Build
============================================================
FEATURE = feature.mk
make -C project/ab155x_evk/apps/dsp0_headset_ref_design/XT-XCC OUTDIR=/
usr/home/user_name/SDK_vn.n.n/dsp/out/ab155x_evk/dsp0_headset_ref_design
2>>
/usr/home/user_name/SDK_vn.n.n/dsp/out/ab155x_evk/dsp0_headset_ref_design/lo
g/err.log
make: Entering directory
`/usr/home/user_name/SDK_vn.n.n/dsp/project/ab155x_evk/apps/dsp0_headset_ref
_design/XT-XCC'
  ...
```

The output files are then put in the `<sdk_root>/out/ab155x_evk/earbuds_ref_design/` folder.

- 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
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2.8.2.3. Building the project from the project GCC configuration directory

To build the project from the project GCC configuration directory, you must complete the following procedure:

1) Change the current directory to the project source directory where the SDK is located.
2) There are makefiles provided for the project build configuration. For example, the iot_sdk_demo is built by the project makefile under `<sdk_root>/project/mt7687_hdk/apps/iot_sdk_demo/GCC`.
   i) Navigate to the example project's location.

```
cd <sdk_root>/project/mt7687_hdk/apps/iot_sdk_demo/GCC
```

   ii) Run the `make` command.

```
Make
```

The output folder is defined under variable `BUILD_DIR` in the `Makefile` in `<sdk_root>/project/mt7687_hdk/apps/iot_sdk_demo/GCC`:

```
BUILD_DIR = $(PWD)/Build
PROJ_NAME = mt7687_iot_sdk_demo
```

A project image `mt7687_iot_sdk_demo.bin` is generated under `<sdk_root>/project/mt7687_hdk/apps/iot_sdk_demo/GCC/Build`.

2.9. Create your own project

This section provides details on how to use an existing project and create your own project named `my_project` on LinkIt 7687 development board using `iot_sdk_demo` project as a reference.

2.9.1. Using an existing project

Apply an existing project as a reference design for your own project development.

Copy the folder `<sdk_root>/project/mt7687_hdk/apps/iot_sdk_demo` to a new directory `<sdk_root>/project/mt7687_hdk/apps/my_project`.

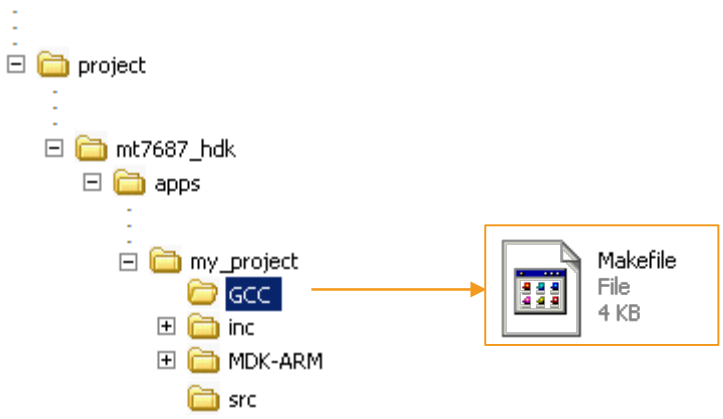

Figure 39. Modify the `Makefile` under the GCC folder of `my_project`

Modify the following settings defined in the `<sdk_root>/project/mt7687_hdk/apps/my_project/GCC/Makefile` under the GCC folder (see Figure 39) `my_project`, as shown below:

- `SOURCE_DIR`: the path to `<sdk_root>`.
- `PROJ_NAME`: project name.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

- APP_PATH: project path.

```
…
SOURCE_DIR  = ../../../../..
BINPATH     = ~/gcc-arm-none-eabi/bin
PWD         = $(shell pwd)
DATIME      = $(shell date --iso=seconds)
V          ?= 0
…
# Project name
PROJ_NAME       = mt7687_my_project
PROJ_PATH       = $(PWD)
OUTPATH         = $(PWD)/Build

APP_PATH        = project/mt7687_hdk/apps/my_project
APP_PATH_SRC    = $(APP_PATH)/src
…
```

2.9.2. Removing a module

The copied project has modules that could be removed in order to have a clean start for your project development. After the previous steps, a project with the same features has been created. It can be built to generate image file as the original project.

To remove a module:

1) Open the project `Makefile` from

   `<sdk_root>/project/mt7687_hdk/apps/my_project/GCC/Makefile.`

2) Locate the module include list of the project and remove any unwanted module by removing or commenting out the corresponding include statement.

```
…
################################################################################
##
#
# SDK source files
#
################################################################################
##
#include cjson
include $(SOURCE_DIR)/middleware/third_party/cjson/module.mk

#include xml
include $(SOURCE_DIR)/middleware/third_party/xml/module.mk

…
```

2.9.3. Add the source and header files

User defined project source and header files should be placed under the `src` and the `inc` folder respectively, as shown in Figure 40.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

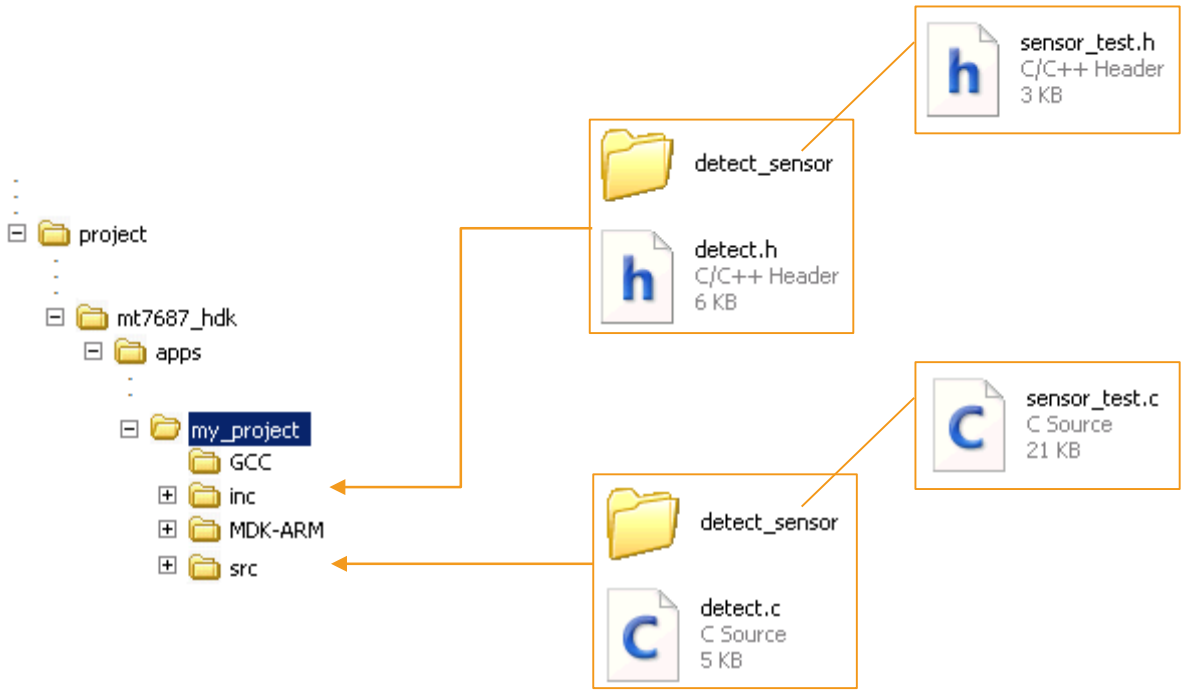

Figure 40. Project source and header files under the project folder

To compile the added source code, simply add the .c source files to variable "C_FILES" and the header search path to variable "CFLAGS" in the project Makefile, as shown below. The corresponding variables to support compiling the source files (.cpp) of the module are CXX_FILES and CXXFLAGS).

<sdk_root>/project/mt7687_hdk/apps/my_project/GCC/Makefile

```
…
C_FILES += $(APP_PATH_SRC)/main.c \
           $(APP_PATH)/GCC/syscalls.c \
           $(APP_PATH_SRC)/detect.c \
           $(APP_PATH_SRC)/detect_sensor/sensor_test.c

CXX_FILES +=
…
CFLAGS += -I$(SOURCE_DIR)/kernel/service/inc
CFLAGS += -I$(SOURCE_DIR)/$(APP_PATH)/inc/detect_sensor

CXXFLAGS +=
…
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

3. Getting Started Using Keil µVision IDE

This section provides a guide to getting started with the Airoha IoT SDK with Keil µVision IDE. It covers the following items:

- The supporting environments for development.
- Installing the Microsoft Windows version of Keil.
- Building the Keil project.
- Downloading and running the Keil project.
- Debugging configuration on Keil.
- Creating a new Keil project.

Note

1) MT7682 and MT7686 does not support Keil IDE.
2) There is currently no support Keil IDE for AB155x series.

3.1. Environment

The SDK supports µVision IDE to build the project on Microsoft Windows OS. The suggested version is version 5.15.

Configure the LinkIt 7687 development board as described in section 2.2.1, "Configuring the LinkIt 7687 HDK", and configure the LinkIt 2523 development board as described in section 2.3.1, "Configuring the LinkIt 2523 HDK".

LinkIt 7697 HDK does not have an on-board debugger, such as the on-board MK20DX128VFM5 on the LinkIt 2523 HDK. Therefore, you need to prepare a hardware debugger, such as a J-Link debug probe, that supports **SWD** interface and ARM Cortex-M4 MCU.

3.2. Installing the Microsoft Windows version of Keil

This section describes how to install the µVision IDE on PCs running Microsoft Windows.

3.2.1. Installing the µVision IDE

The SDK is released as a package for µVision IDE. If you've already installed one of the supported µVision versions, you can skip this step. If you don't have µVision installed:

1) Download the µVision IDE.
2) Install the µVision IDE.

3.2.2. Installing the Keil package for the SDK

Download the Keil package of the SDK from here. Double click on the file and install the package by following the on-screen instructions, as shown in Figure 41.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

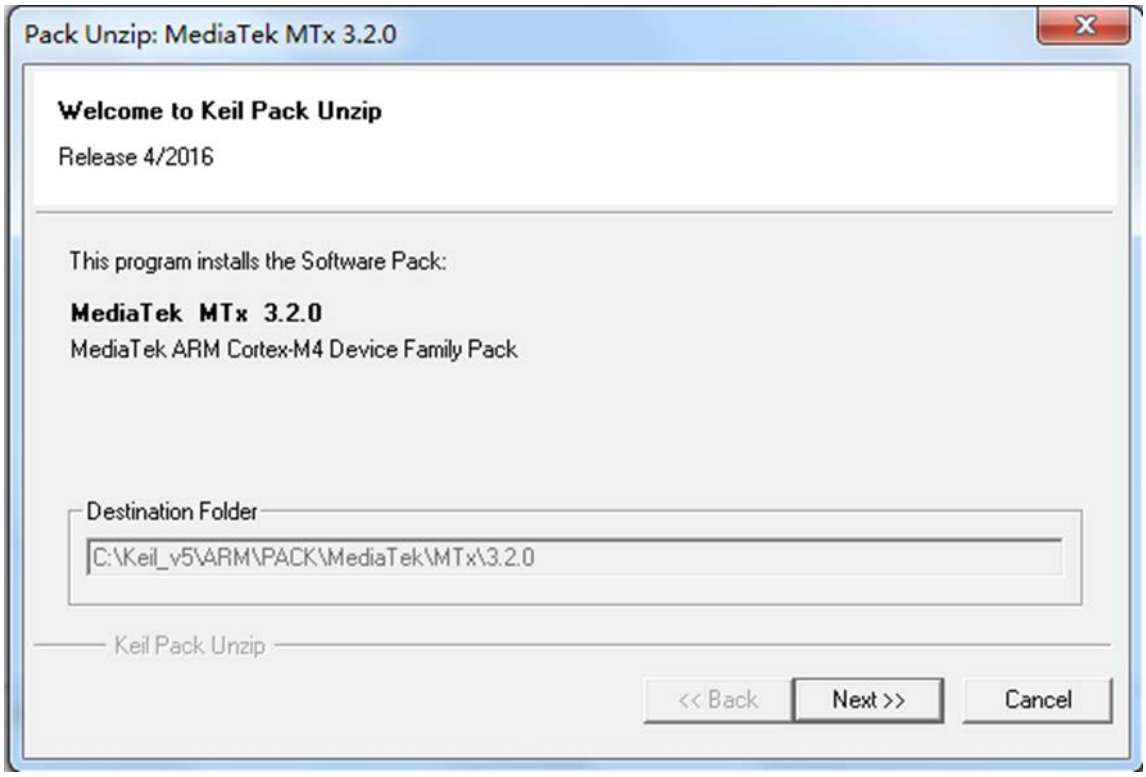

Figure 41. Installing Airoha IoT SDK with Keil IDE

The package directory can be browsed under **Destination Folder**, as shown in Figure 41 once the installation finishes.

3.3. Build the project

This section describes the required steps to build, download and run a project using the SDK with µVision IDE. The example projects are preconfigured for LinkIt 7687 HDK and LinkIt 2523 HDK, the example below is based on LinkIt 7687 HDK.

1) Launch the µVision IDE.

2) Import the preconfigured project.

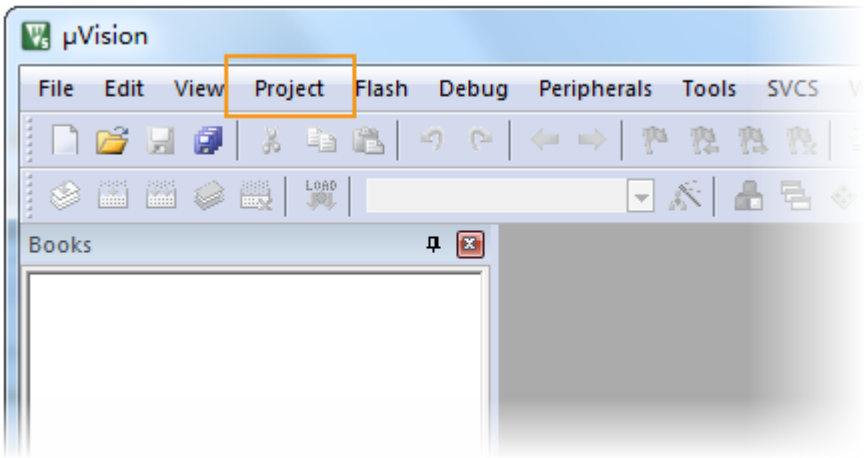

Figure 42. Importing the preconfigured project

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

a) Navigate to **Project** (see Figure 42) in the µVision IDE's main menu, then **Open Project**. Click **Open** to import the project. The project file is named as `<project>.uvprojx`, such as the one shown in Figure 43.

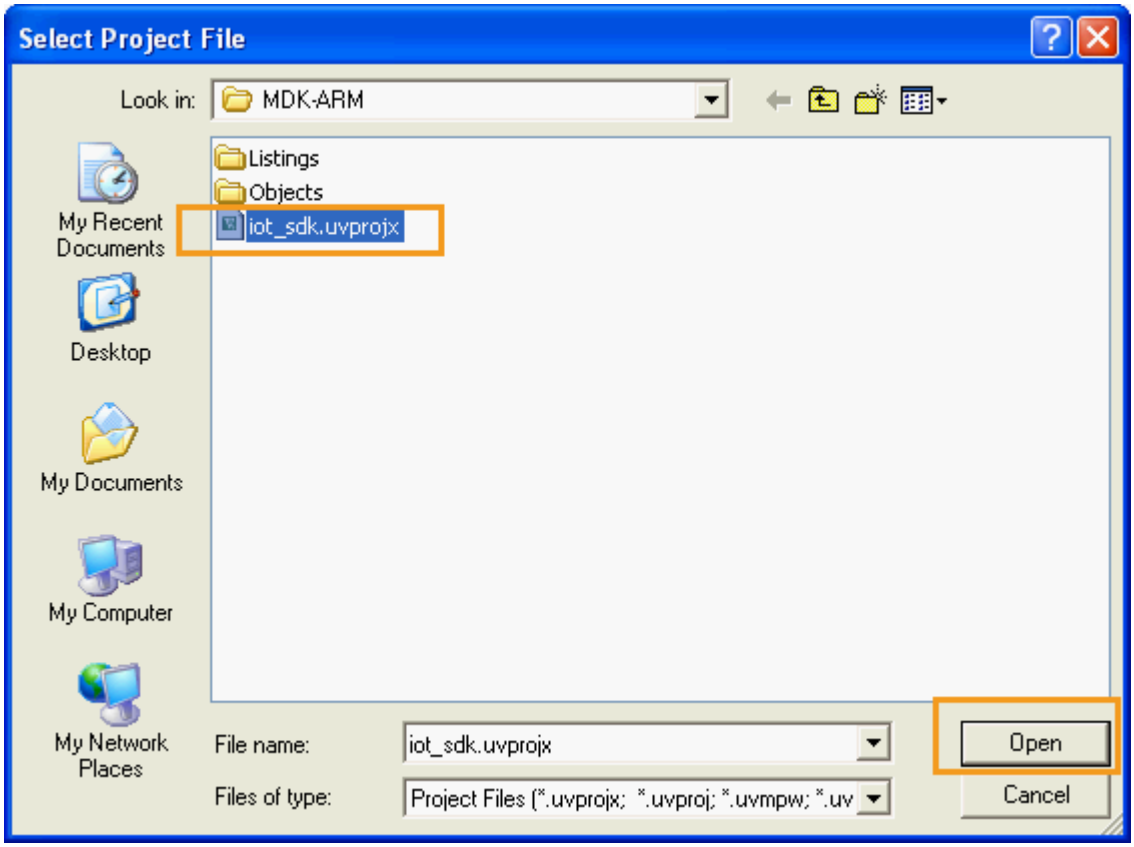

Figure 43. Project file for Keil

3) Click **Rebuild** to build the project, as shown in Figure 44.

Figure 44. Building the project

4) The **Build Output** window is updated with the result, as shown in Figure 45.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

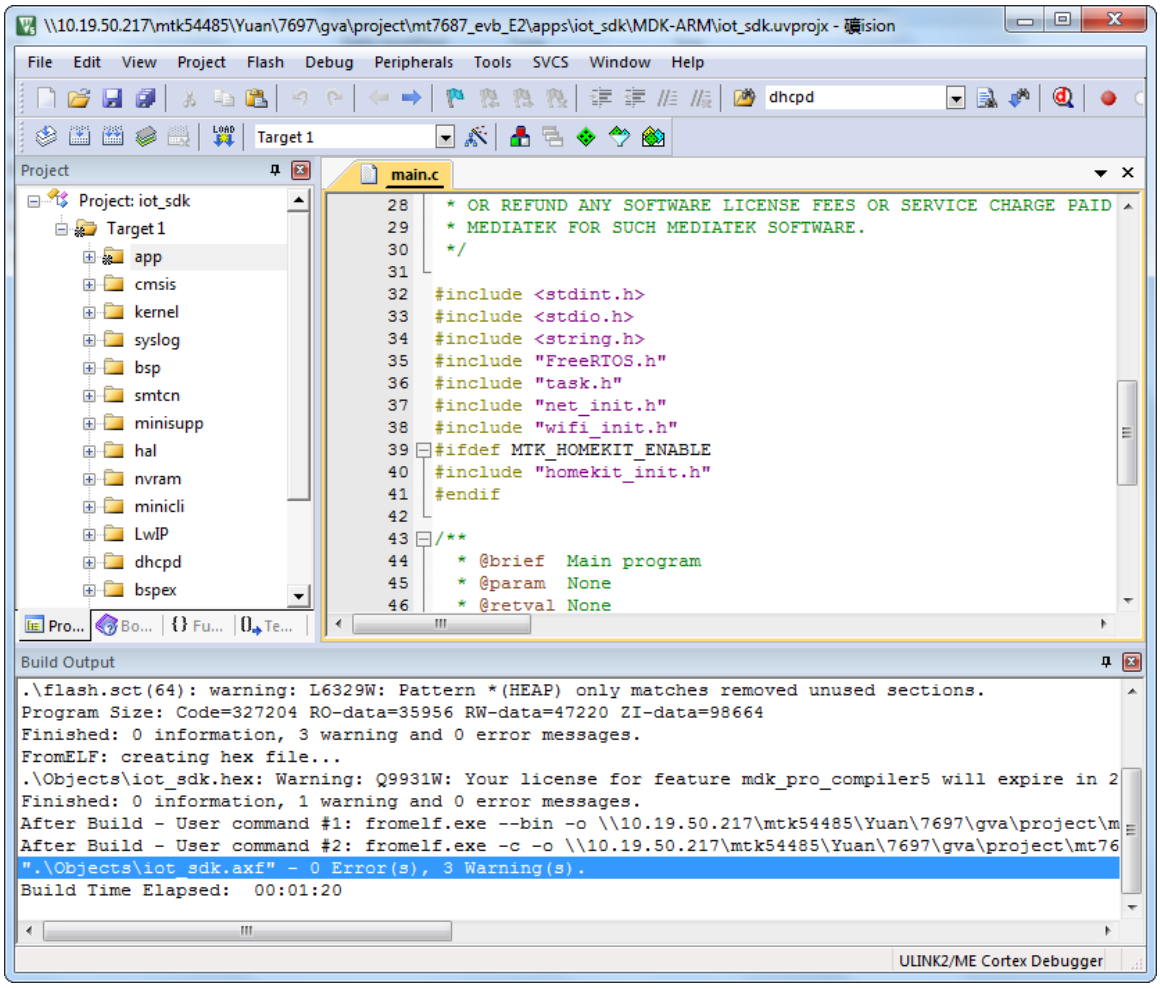

Figure 45. Project build result

5) The output binary `iot_sdk_demo.axf` is generated under project directory `<sdk_root>\out_keil\mt7687_hdk\iot_sdk_demo\iot_sdk_demo.axf`.

3.4. Download and run the project

This section describes how to download and run the project using the Keil µVision IDE on Airoha IoT development platform for RTOS including LinkIt 7687 HDK and LinkIt 2523 HDK. The example below is based on LinkIt 7687 HDK. The project is then built using the Keil µVision IDE.

There are three methods to download the project binary that is already built as described in Building Environment Developer's Guide.

1) Download the project binary using the Keil µVision IDE.

2) Download the project binary using MT76x7 Flash Tool. The MT2523 Flash Tool doesn't support to download multiple binaries.

3) Download the project binary using the LinkIt 7687 HDK as a removable storage.

To learn how to install board drivers and flash images to the LinkIt 7697 HDK, visit the online get started page at https://docs.labs.mediatek.com/resource/mt7687-mt7697/en/get-started.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

3.4.1. Downloading the project binary using the Keil µVision IDE

This method uses the µVision IDE integrated with CMSIS-DAP debugger.

1) Set the **FLASH MODE** (**J25**) jumper to **FLASH Recovery** mode.

2) Connect a micro-USB cable to power on the LinkIt 7687 HDK.

3) Click **Options for Target** or navigate to **Project**, then **Options for ...** and select **Target** on the main menu of the IDE (see Figure 46).

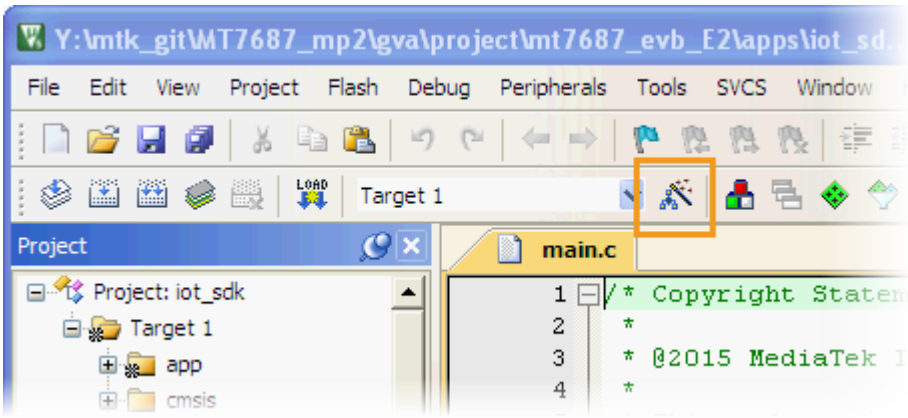

Figure 46. Set download configuration on µVision IDE

4) Select **Utilities** tab, then browse for the **Init File**. Select the file and press **Open**. The configuration file for the SDK is named as `flash.ini` and located under `<sdk_root>\tools\keil\mt7687`, as shown in Figure 47.

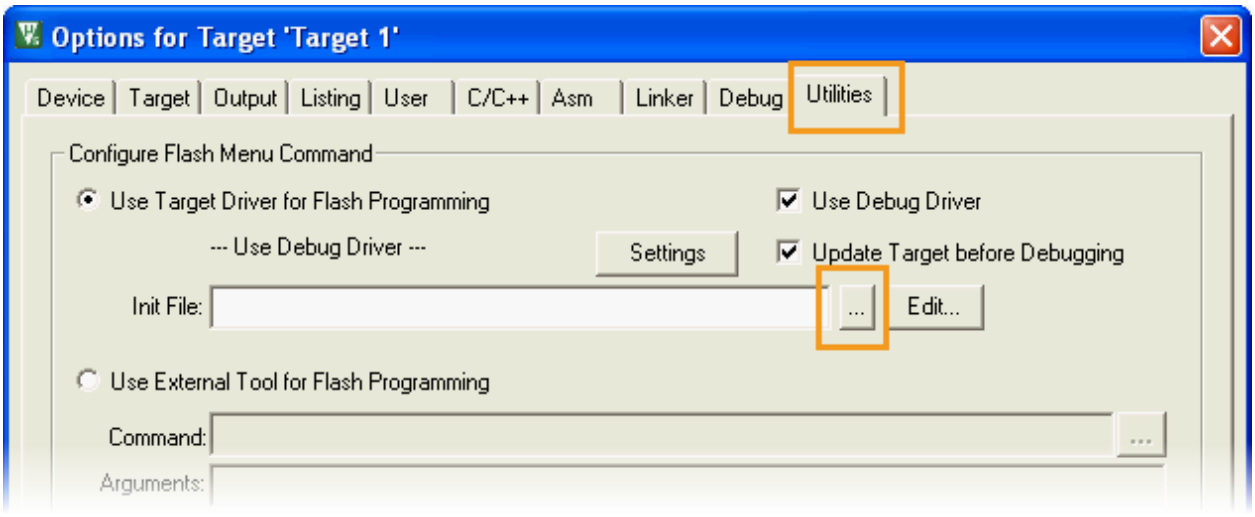

Figure 47. Select configuration file

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

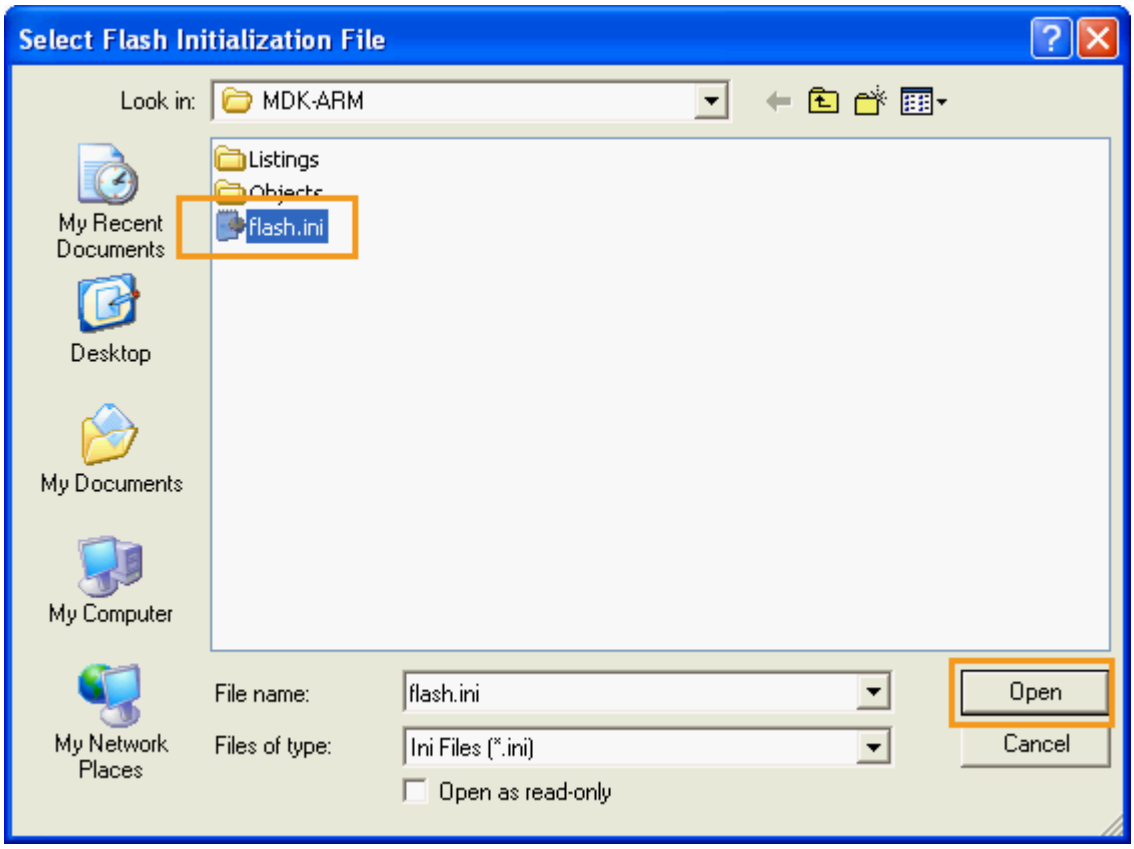

Figure 48. Select flash initialization file (flash.ini)

5) Navigate to the **Debug** tab to select the debug interface. The HDK supports the CMSIS-DAP interface, so here you can select **CMSIS-DAP Debugger** (see Figure 49).

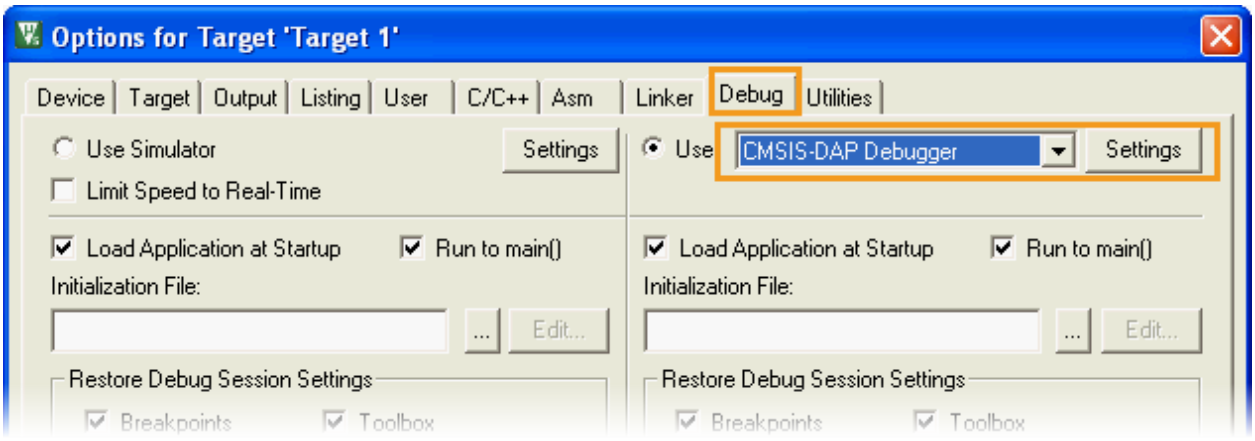

Figure 49. Select the CMSIS-DAP Debugger interface

6) Click **Settings** to switch to **Cortex-M Target Driver Setup** and select **Flash Download** to add device flash programming algorithm (see Figure 50).

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

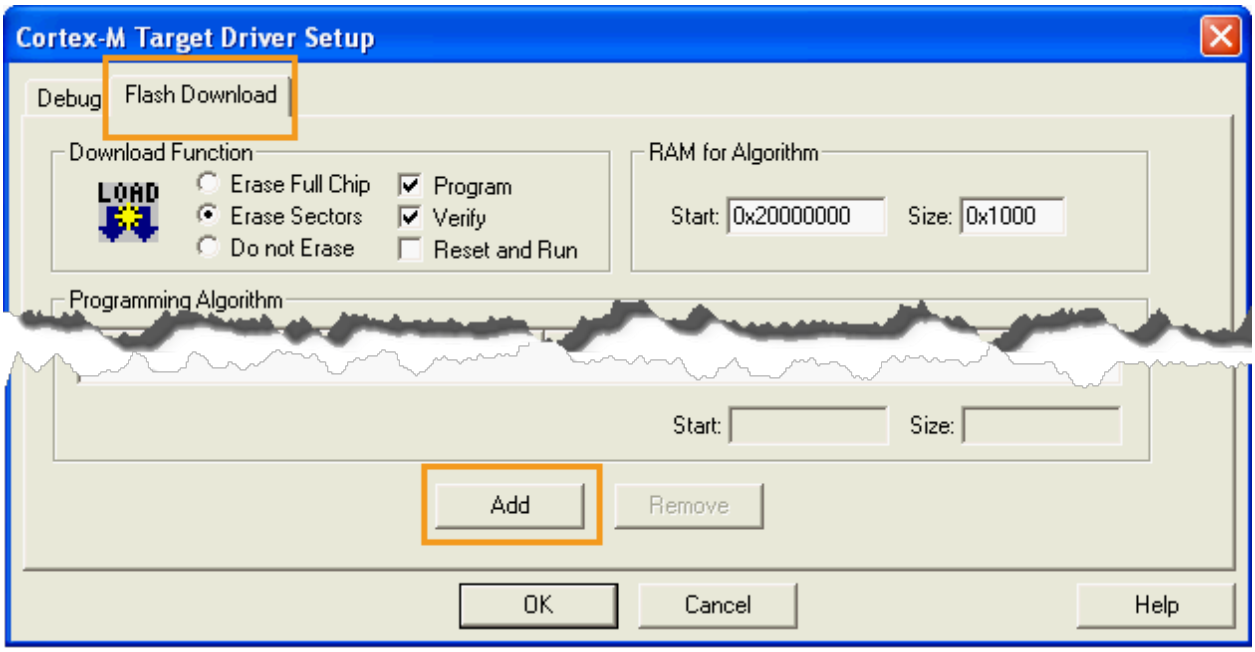

Figure 50. Select device flash programming algorithm

7) Click **Add** to switch to **Add Flash Programming Algorithm**, then select the **7687 32Mbits SIP Flash** and click **Add** (see Figure 51).

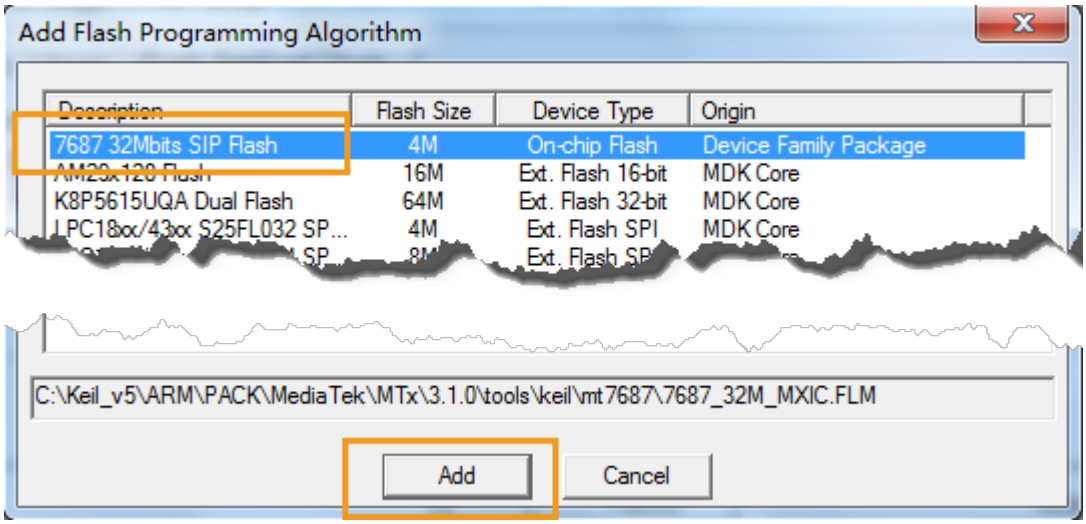

Figure 51. Add device flash programming algorithm

8) If downloading on LinkIt 2523 HDK, switch to **Debug**, and set **AP** as **0x03** then click **OK** (see Figure 52). Skip this step, if downloading on the LinkIt 7687 HDK.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

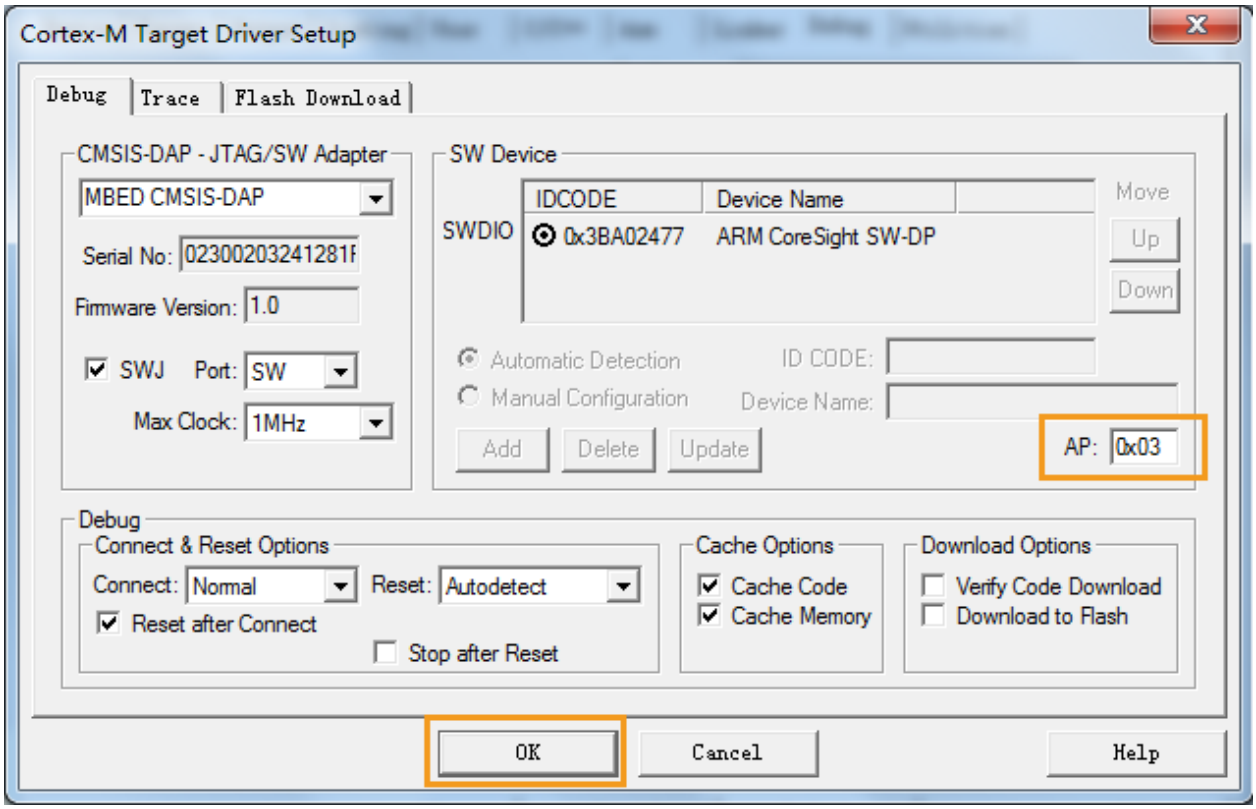

Figure 52. Setup the parameters for downloading the project on LinkIt 2523 HDK

9) Reset the board and then click **LOAD** to download the project binary to target (see Figure 53).

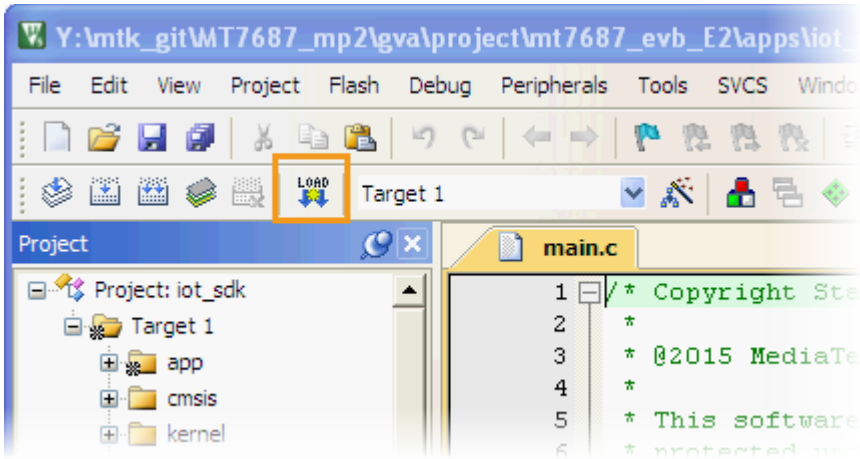

Figure 53. Download the project binary

3.4.2. Downloading the project binary using Flash Tool or using HDK as a removable storage device

Before you start to download the project binary, configure the settings and rebuild the project using Keil µVision IDE.

1) Click **Options for Target** or navigate to **Project**, then **Options for ...** and select **Target** on the main menu of µVision IDE, as shown in Figure 46.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

2) Navigate to **User** tab, select **Run #1** and **Run # 2** under **After Build/Rebuild**, fill **fromelf.exe --bin -o $L@L.bin #L**, and **fromelf.exe -c –o-$L@L.dis #L** in **User Command** for **Run #1** and **Run # 2**, then select **Beep When Complete**, as shown in Figure 54.

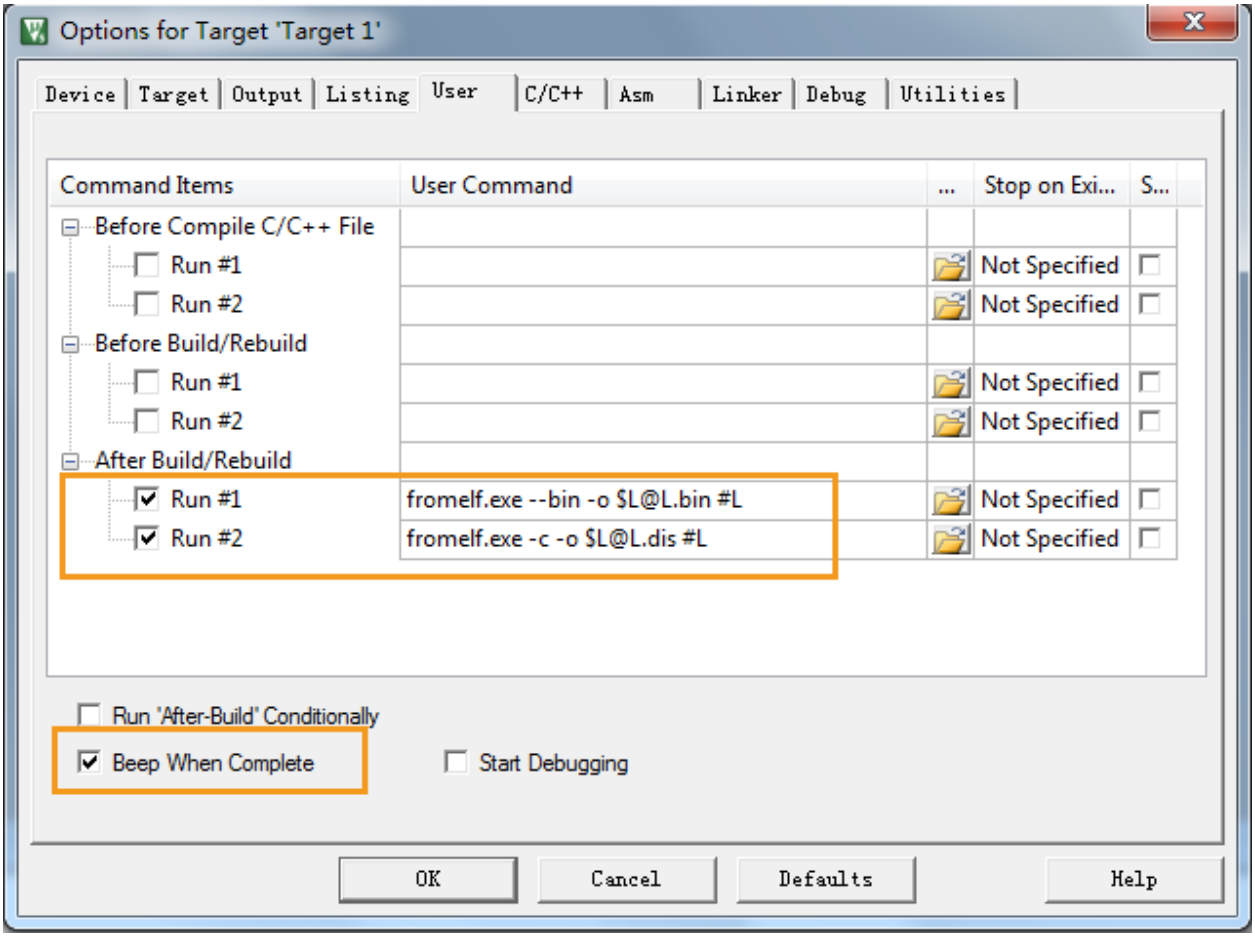

Figure 54. User defined configurations

3) Click **Rebuild** to build the project, as shown in Figure 44.

4) An output binary `mt7687_iot_sdk_demo.bin` will be generated in project directory `<sdk_root>\out_keil\mt7687_hdk\iot_sdk_demo\mt7687_iot_sdk_demo.bin`.

Download the project binary following the details in MT76x7 Flash Tool User's Guide or update the RTOS firmware through the removable storage based on the details in LinkIt 7687 HDK User's Guide.

3.4.3. Running the project

To run the project:

1) Disconnect the USB cable to power off the board.

2) Set **FLASH MODE** (**J25**) jumper to **FLASH Normal** mode.

3) Reconnect the USB cable or reset the board to power on the board.

4) Open **HyperTerminal** (the default: **HyperTerminal** on Windows OS) program, and configure it as described in section "Host UART Configuration" of Airoha IoT Development Platform for RTOS System Log Developer's Guide, then observe the output log written from the project source files to ARM Cortex-M4 UART port. The following is a reference log example.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

```
loader init [2016-04-28 15:03:10.924]
[2016-04-28 15:03:10.924]
fota: TMP is empty, skip upgrade[2016-04-28 15:03:10.939]
[2016-04-28 15:03:10.939]
jump to (0x1007c000) [2016-04-28 15:03:10.939]
total avail space = 13748
[2016-04-28 15:03:10.955]
nvdm init finished
[2016-04-28 15:03:10.955]
[T: 52 M: common C: INFO F: system_init L: 269]: FreeRTOS Running[2016-04-28
15:03:10.970]
```

3.5. Debug configuration

Keil µVision IDE uses the CMSIS-DAP/mbed debugger supported on LinkIt 7687 HDK and LinkIt 2523 HDK to debug the project. Here are the debug configurations based on LinkIt 7687 HDK after the project is built and downloaded with the IDE.

1) Disconnect the micro-USB cable to power off the board.
2) Set the **FLASH MODE** (**J25**) jumper to **FLASH Normal** mode.
3) Reconnect the USB cable or reset the board to power on the board.
4) Launch the µVision IDE.
5) Click **Options for Target** or navigate to **Project**, then **Options for ...** and select **Target** on the main menu of, as shown in Figure 46.
6) Navigate to **Debug** tab in the **Options for Target** to select the **CMSIS-DAP Debugger**, as shown in Figure 55.

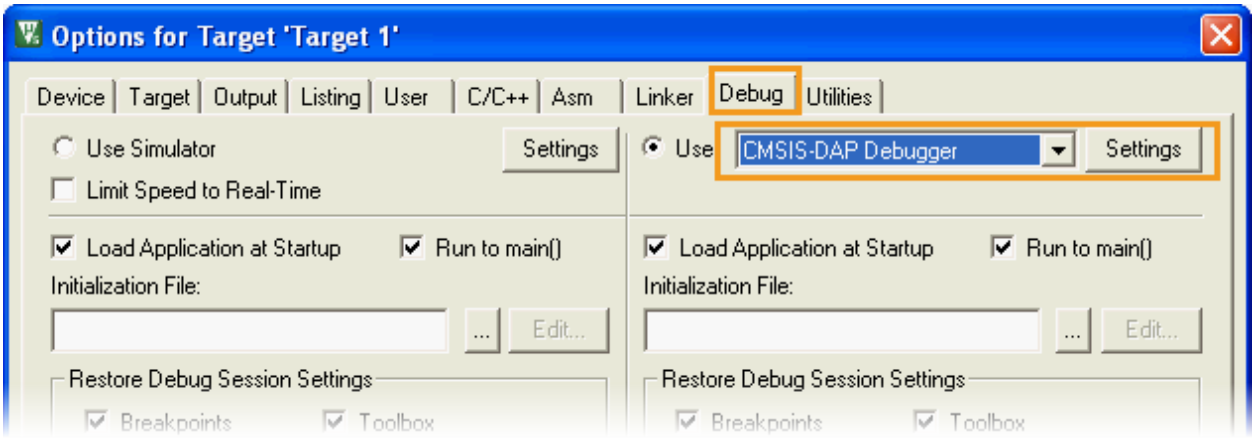

Figure 55. Configure the debug settings

7) Click **Settings** to switch to **Cortex-M Target Driver Setup**, then select **Debug** tab to set the configuration parameters and click **OK**, as shown in Figure 56.
    - **SWJ Port**: SW
    - **Max Clock**: 1MHz

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

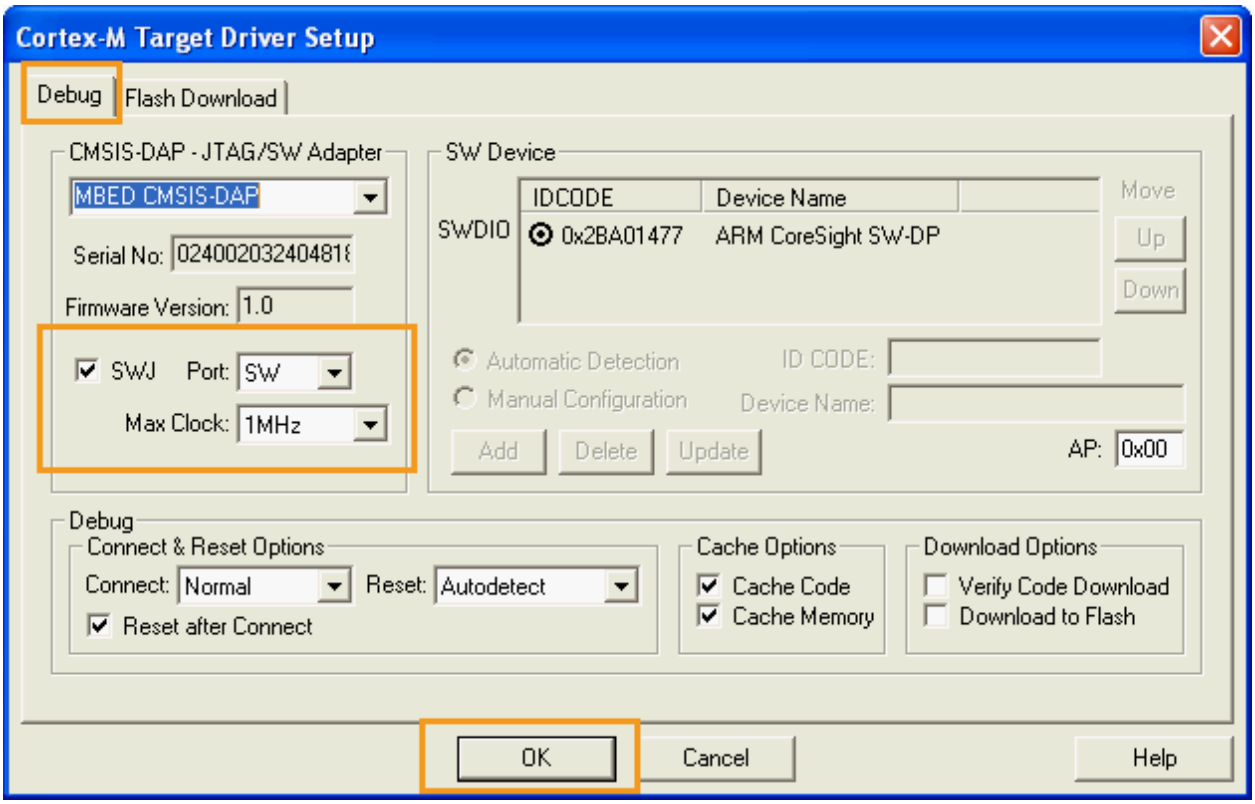

Figure 56. Setting the debug parameters on LinkIt 7687 HDK

8) If debugging on LinkIt 2523 HDK, set **AP** as 0x03, as shown in Figure 57.

- **AP:** 0x03

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

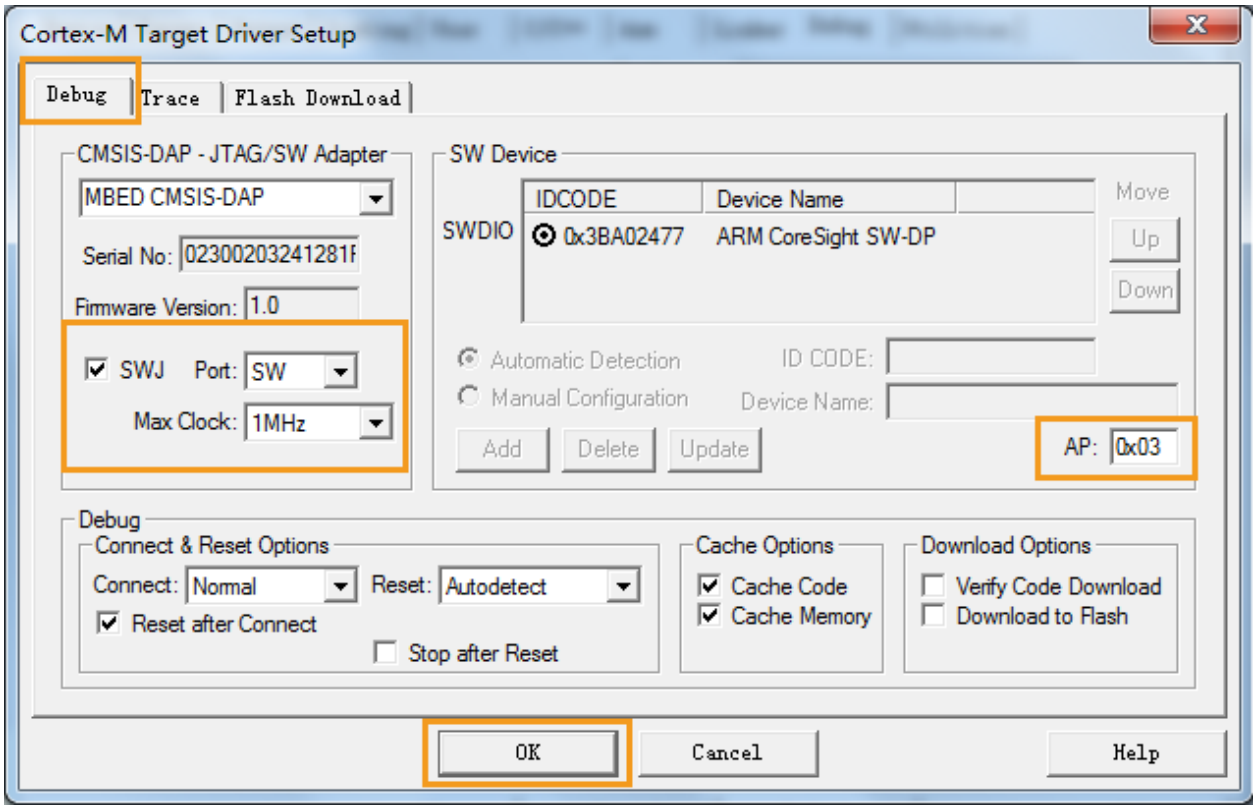

Figure 57. Setting the debug parameters on LinkIt 2523 HDK

9) Click **Debug** to start debugging the project, as shown in Figure 58.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

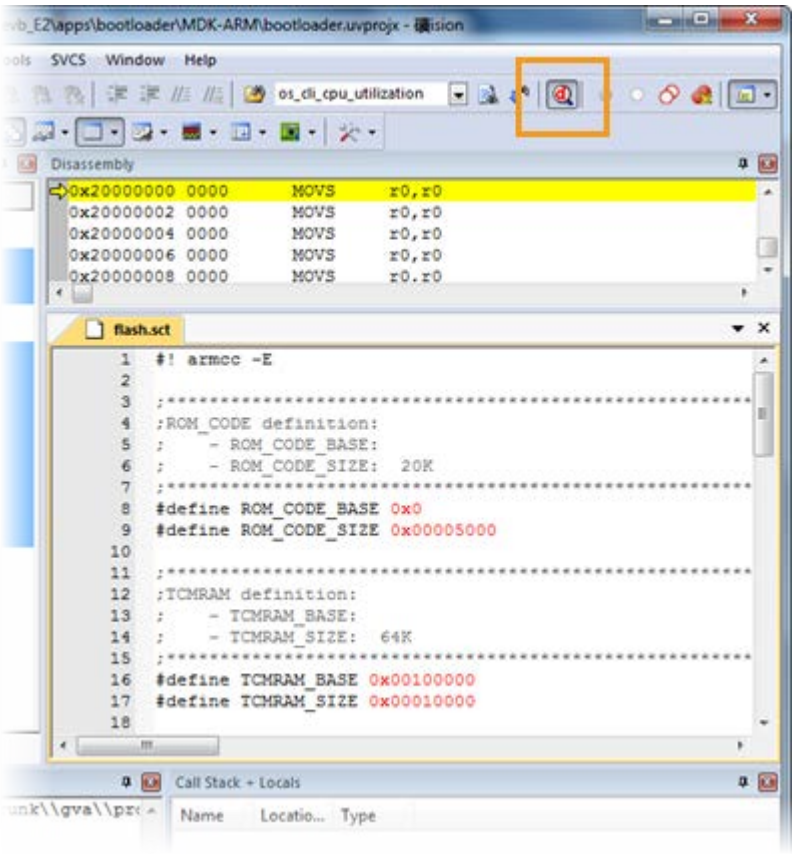

Figure 58. Debugging a project on Keil IDE

Note, Keil debugging is impossible in sleep mode. For more information, please refer to Airoha IoT SDK for RTOS Power Mode Developers Guide under `<sdk_root>/doc`.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

4. Getting Started Using IAR

This section provides a guide to getting started with the Airoha IoT SDK v4 with IAR embedded workbench. It covers the following items:
The supported environments for development.

Installing the Microsoft Windows version of IAR.

Building the IAR project.

Downloading and running the IAR project.

Configuring the debug settings on IAR.

Creating your own IAR project.

Note, there is currently no support IAR embedded workbench IDE for AB155x series.

4.1. Environment

The SDK supports IAR Embedded Workbench (suggested version is 7.50.1.10273) to build the project on Microsoft Window OS.

Configure the LinkIt 7687 development board as described in section 2.2.1, “Configuring the LinkIt 7687 HDK”, and configure the LinkIt 2523 development board as described in section 2.3.1, “Configuring the LinkIt 2523 HDK”.

To configure the LinkIt 7686 and 7682 HDK board, please refer to HDK Users Guide under `<sdk_root>/doc/board` (see section 2.5, "Enter IAR and GCC MDK development interface").

Note that LinkIt 7697 HDK does not have an on-board debugger, such as the on-board MK20DX128VFM5 on the LinkIt 2523 HDK. Therefore, you need to prepare a hardware debugger, such as a J-Link debug probe, that supports **SWD** interface and ARM Cortex-M4 MCU.

4.2. Installing the Microsoft Windows version of IAR

If you’ve already installed one of the supported IAR versions, you can skip this step. If you don’t have IAR installed:

1) Download the iar-embedded-workbench.

2) Install the IAR Embedded Workbench.

4.3. Build the project

The example projects are preconfigured for LinkIt 2523 HDK and LinkIt 7687 HDK, the example below is based on LinkIt 2523 HDK.

To build the project:

1) Launch the IAR Embedded Workbench IDE.

2) Import the pre-configured project.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

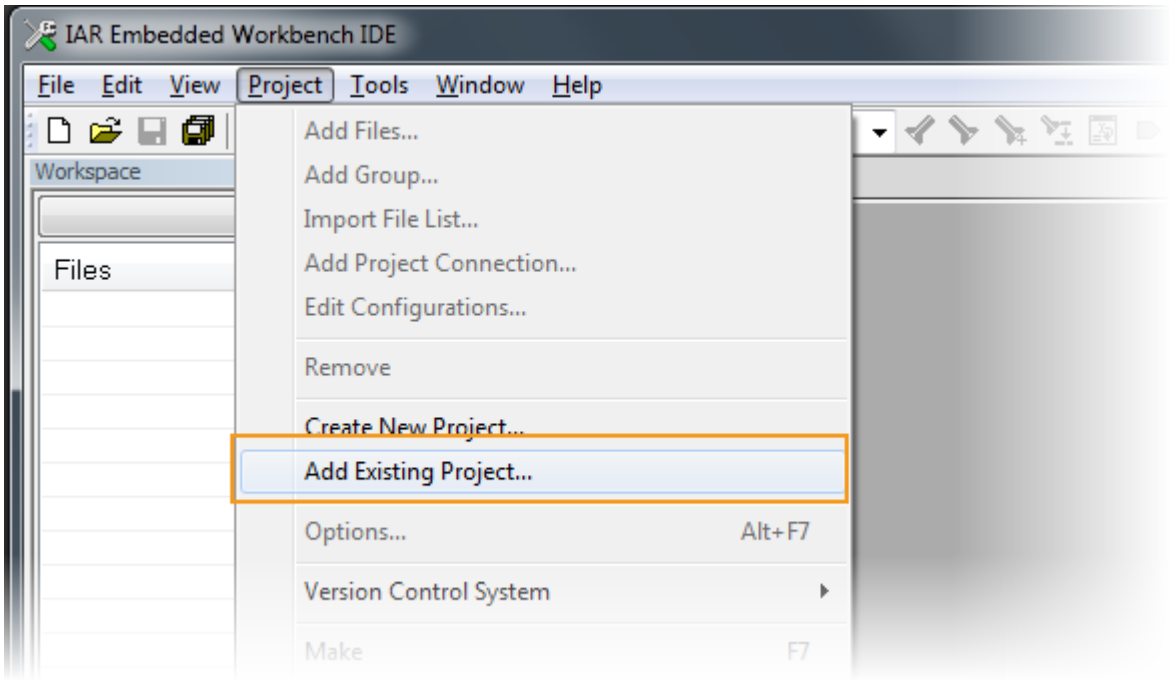

Figure 59. Importing the preconfigured project

a) Navigate to **Project** (see Figure 59) in the IDE's main menu, then **Open Project**. Click **Open** to import the project. The project file is named as `<project>.ewp`, such as the one shown in Figure 60.

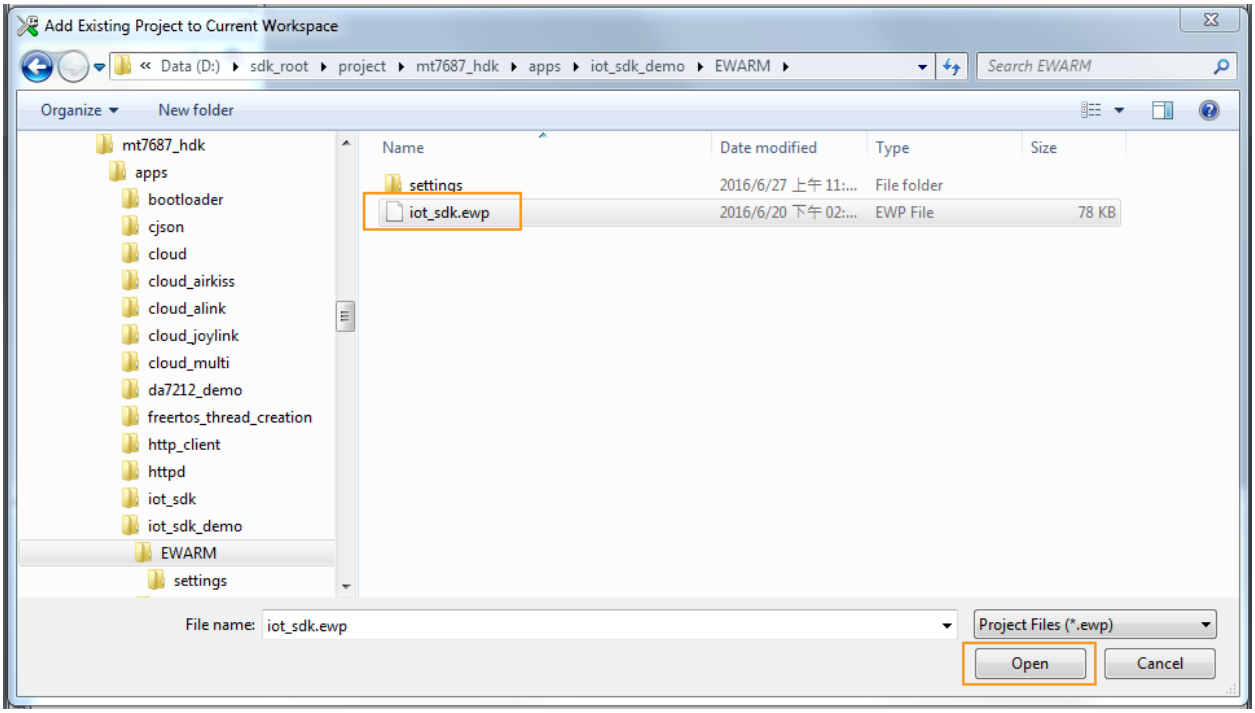

Figure 60. Adding an existing project

3) Click **Rebuild** to build the project, as shown in Figure 61.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

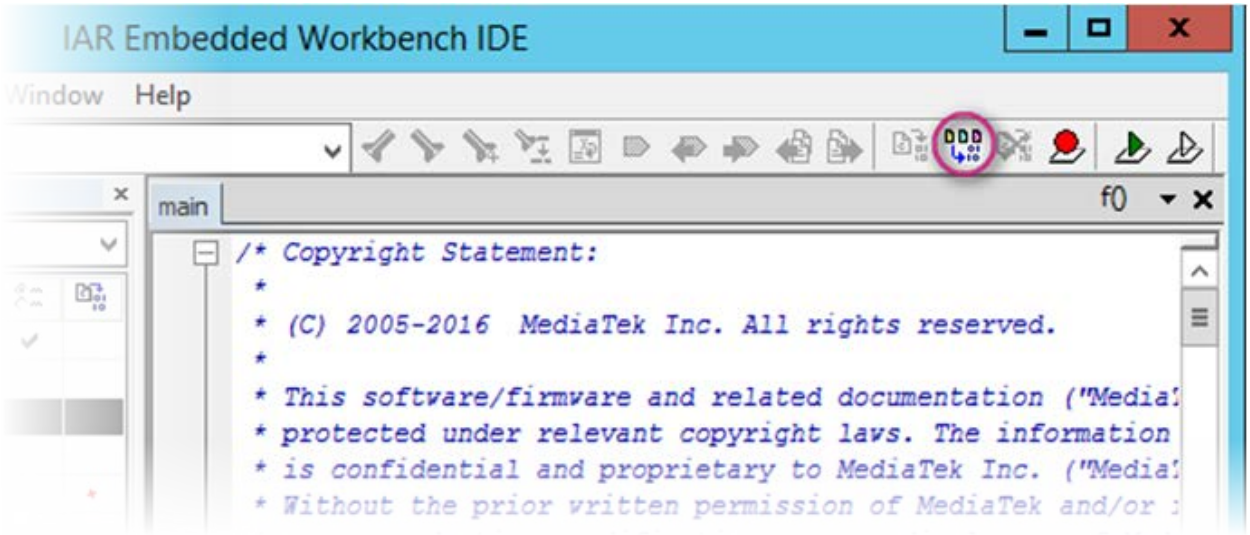

Figure 61. Rebuild the project

4) When the build operation is complete, the **Build output** window is updated with the result, as shown in Figure 62.

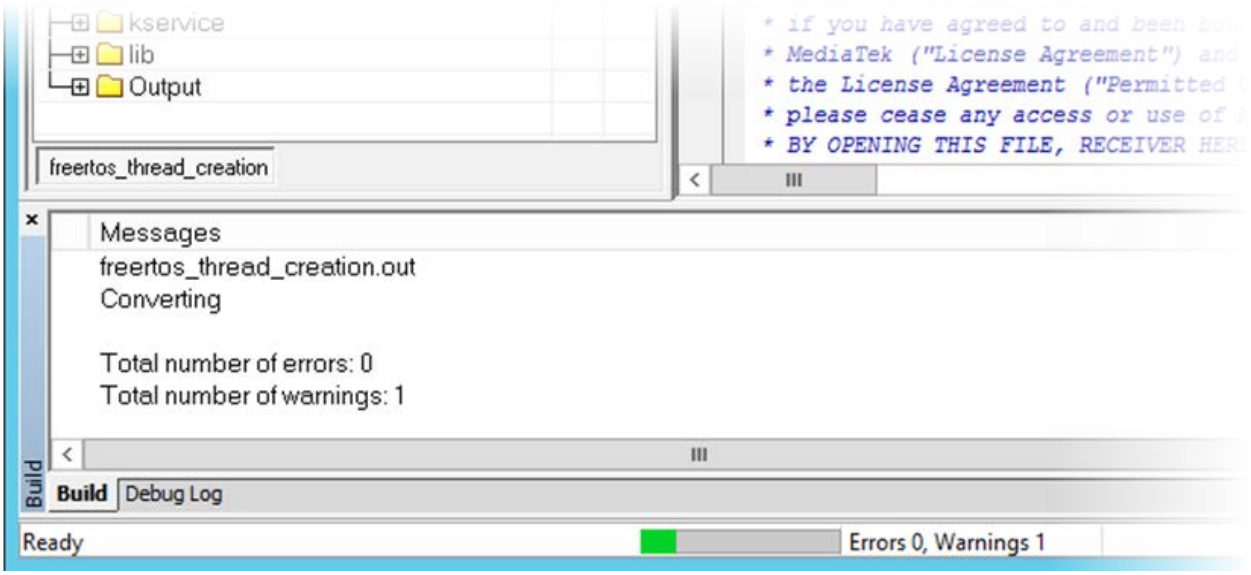

Figure 62. Project build result on IAR Embedded Workbench IDE

5) The output binary `freertos_thread_creation.out` is generated under project directory `<sdk_root>\out_iar\mt2523_hdk\freertos_thread_creation\freertos_thread_creation.out`.

4.4. Download and run the project

This section describes how to download and run the project using the IAR Embedded Workbench on Airoha IoT development platform for RTOS including LinkIt 2523 and LinkIt 7687. The example blew is based on LinkIt 2523 HDK. The project is then built using the IAR Embedded Workbench.

To learn how to install board drivers and flash images to the LinkIt 7697 HDK, visit the online get started page at https://docs.labs.mediatek.com/resource/mt7687-mt7697/en/get-started.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

4.4.1. Download the project

This method uses the IAR IDE integrated with CMSIS DAP debugger.

To download the project:

1) Connect micro-USB connector to power on the LinkIt 2523 HDK.

2) Click **Options** on IAR Embedded Workbench (see Figure 63).

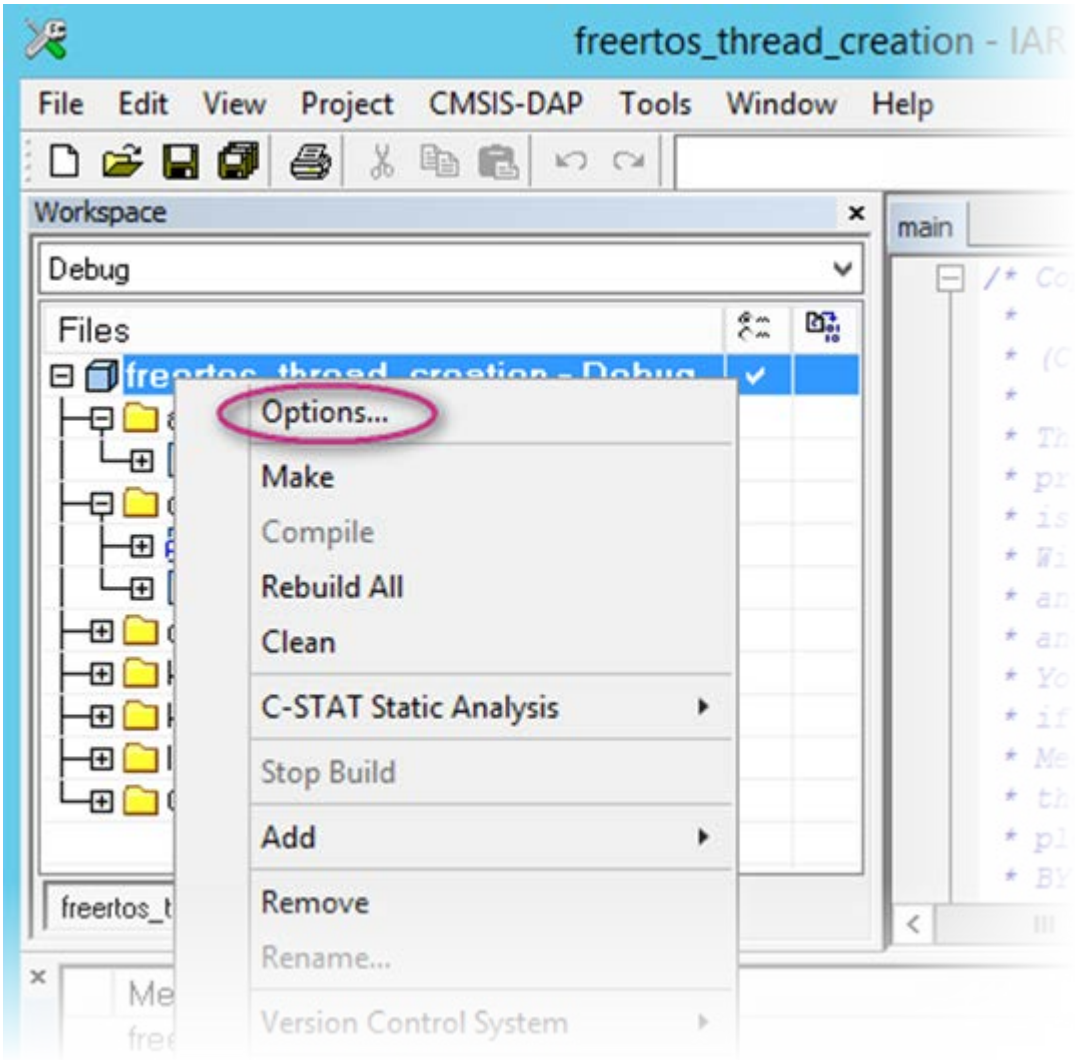

Figure 63. Set download configuration on IAR IDE

3) Choose **Debugger** item, then browse for the **Setup**.

   a) Set **Driver** to **CMSIS-DAP** debugger (see Figure 64).

   b) Set the **Device description file** to MT2523.ddf located under `<sdk_root>\tools\iar\mt2523`, as shown in Figure 65.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

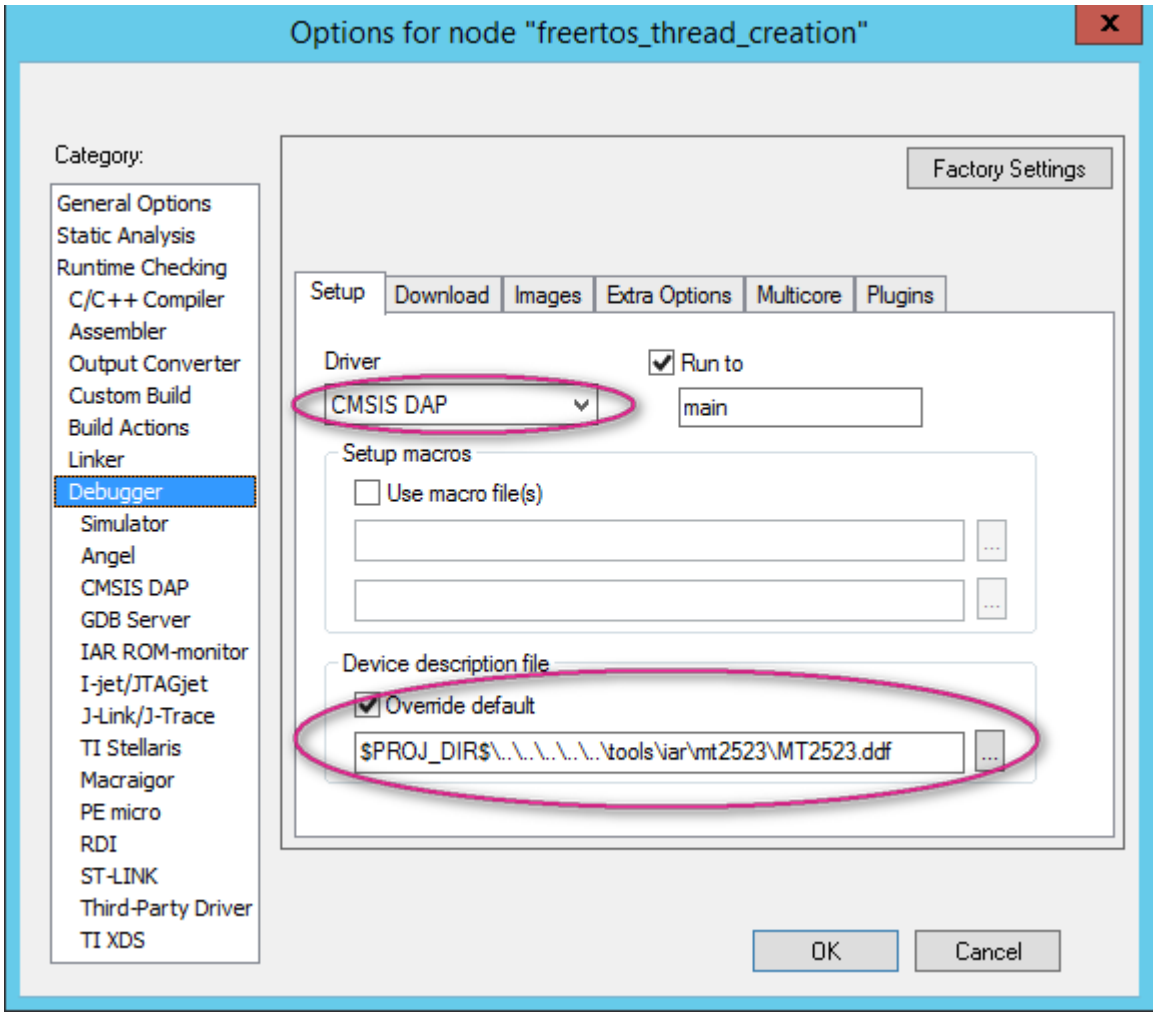

Figure 64. Select debug driver and ddf file

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

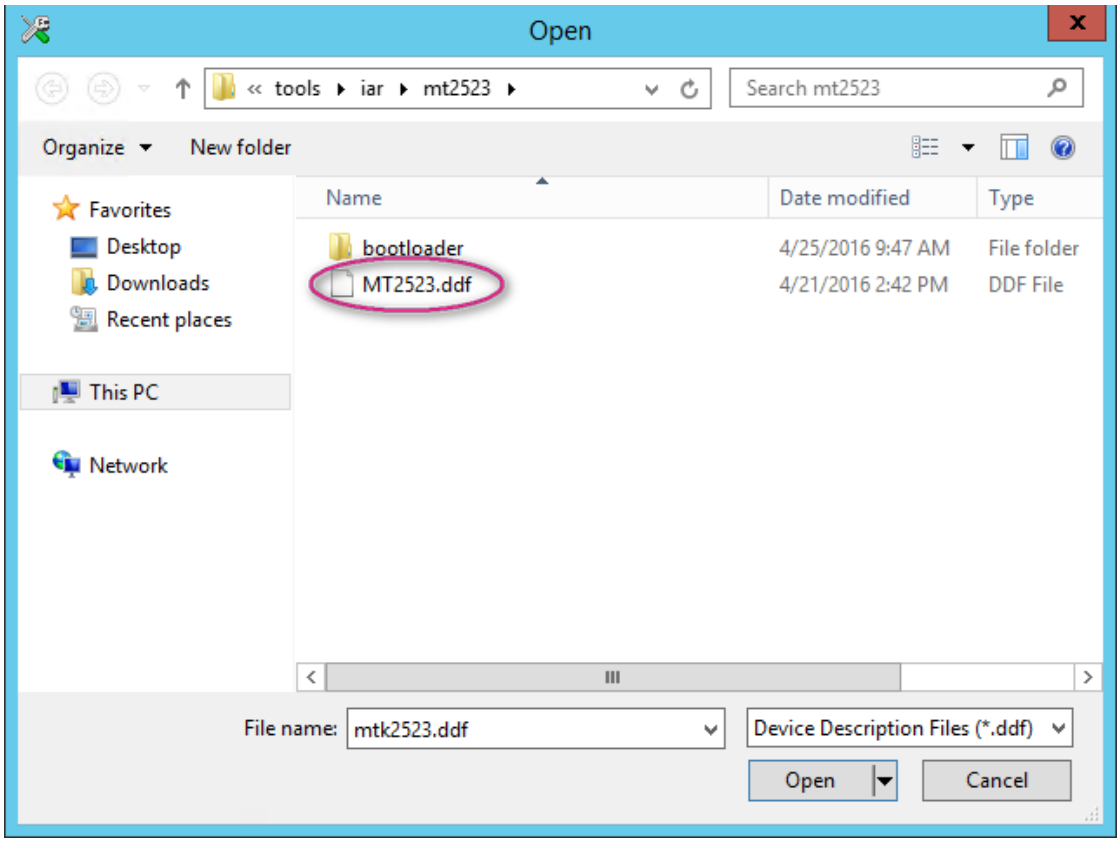

Figure 65. Open the MT2523.ddf file

4) Navigate to the **Download** tab and configure the settings as shown in Figure 66).

   a) The board file is located under `<sdk_root>\tools\iar\mt2523`, as shown in Figure 67.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

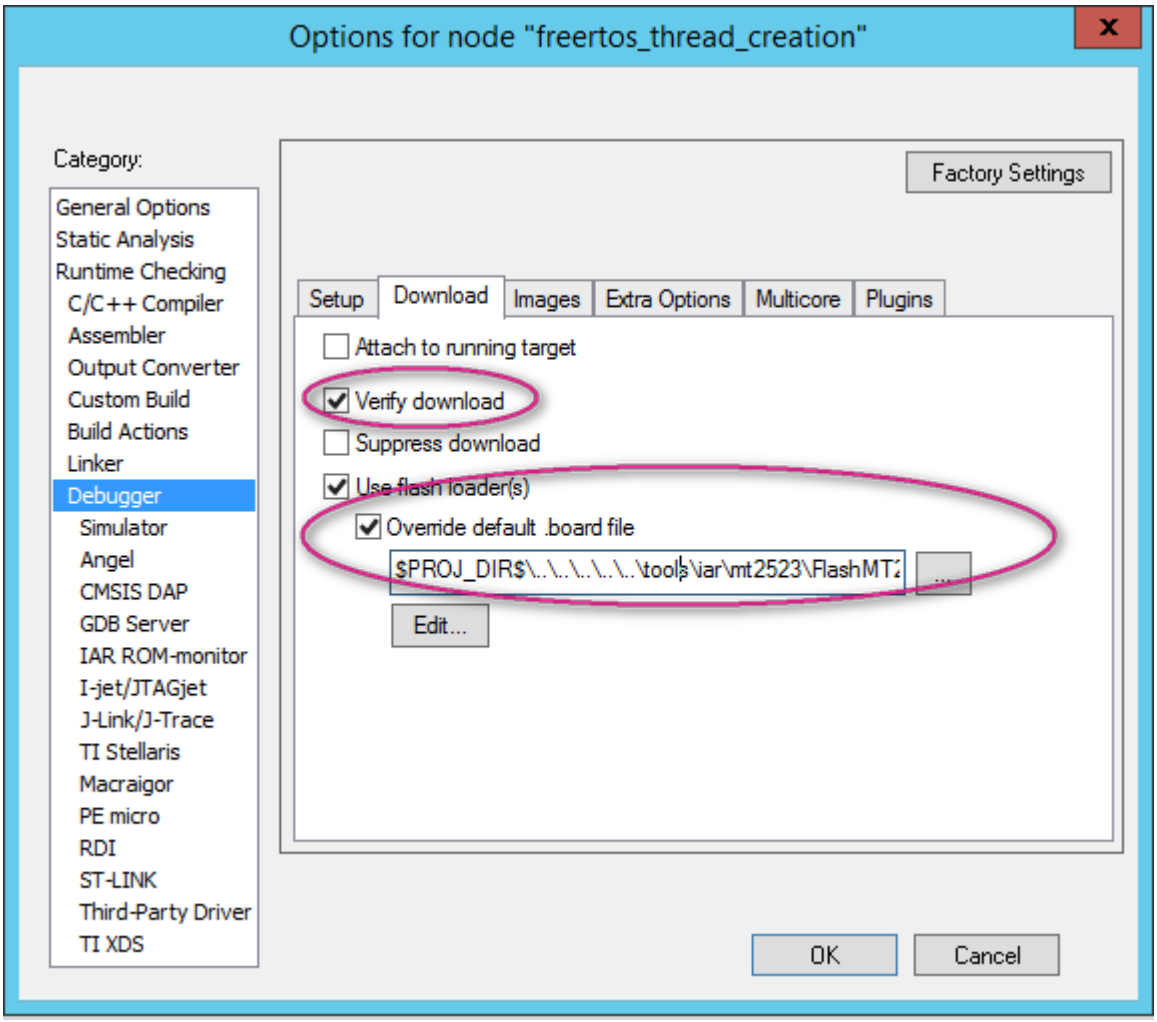

Figure 66. Configure the download settings

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

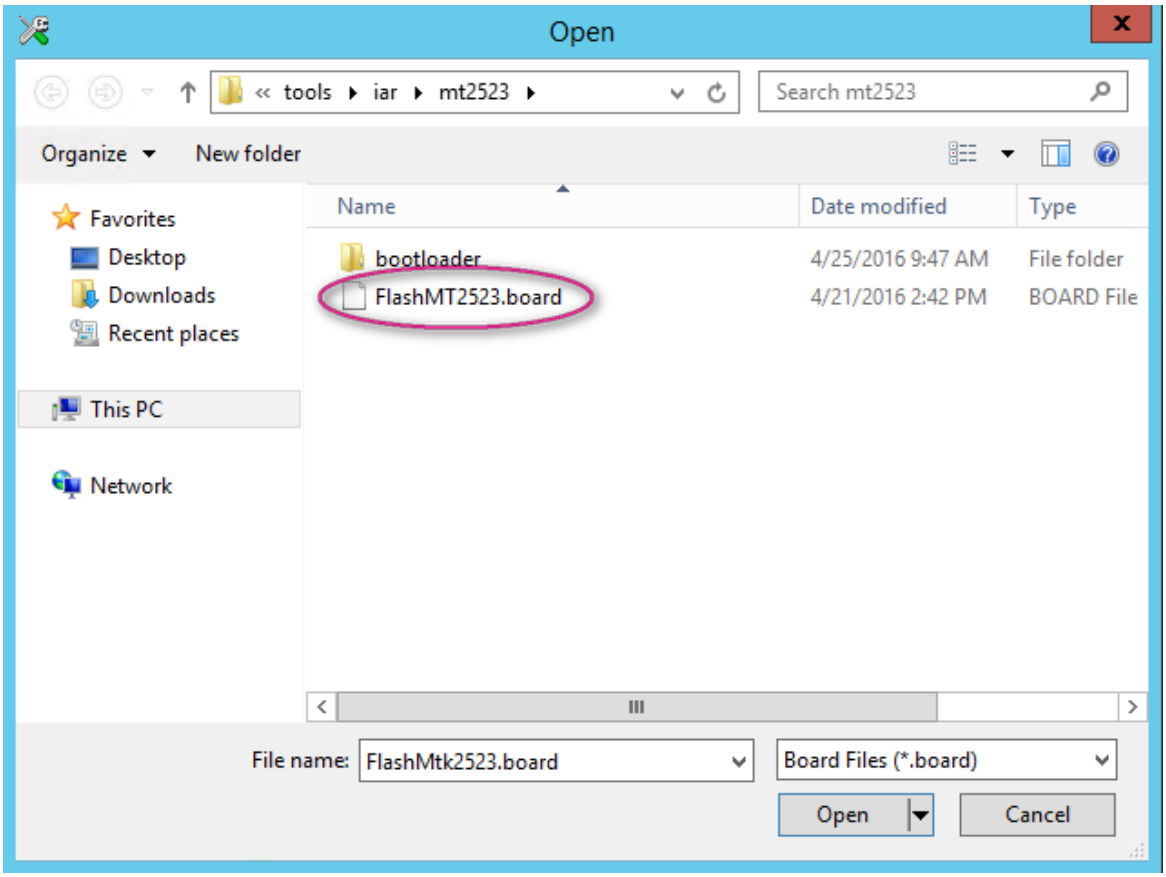

Figure 67. Select the board file

5) Choose Linker and Navigate to the input tab to modify setting as below, as shown in Figure 68.

- **Keep symbols:** `__bootloader`
- **File**: `$PROJ_DIR$\..\..\..\..\..\tools\iar\mt2523\bootloader\bootloader.bin`
- **Symbol**: `__bootloader`
- **Section**: `.bootloader`
- **Align**: 8

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

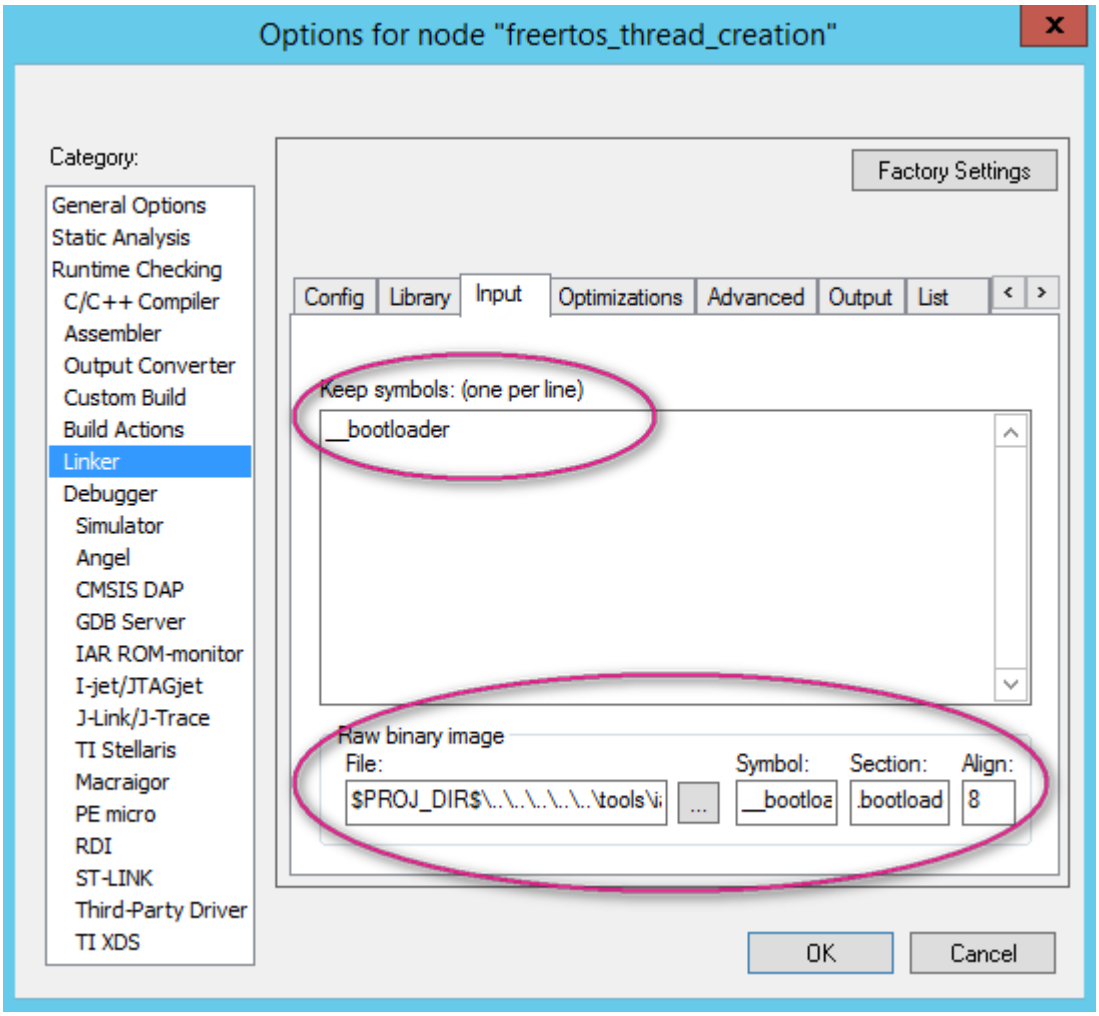

Figure 68. Set extra download Image

Next, configure the **CMSIS-DAP** settings under **Category** and then:

1) Navigate to **JTAG/SWD** tab and provide the interface (see Figure 69).

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

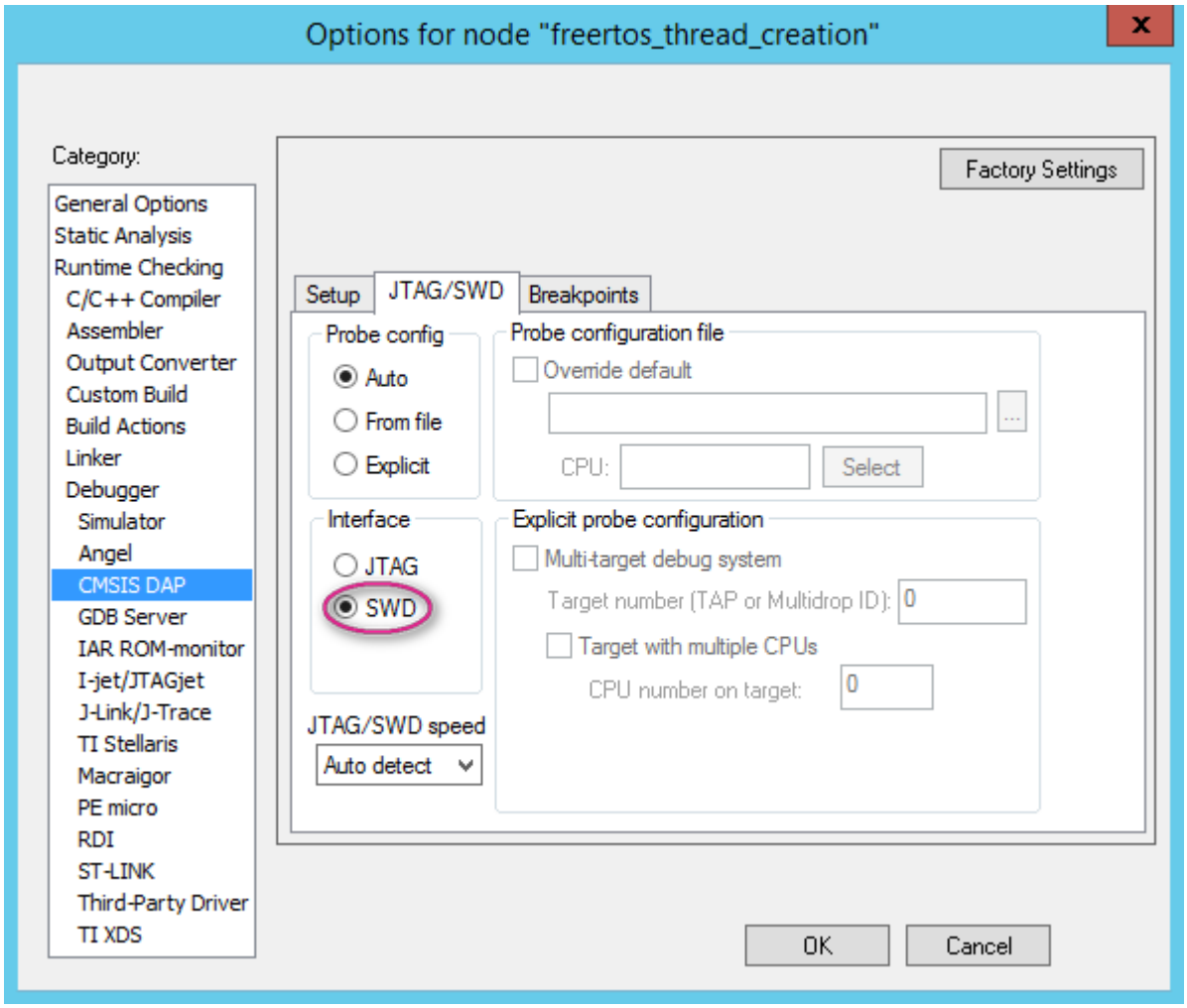

Figure 69. JTAG/SWD settings

2) Reset the board and then navigate to **Project**, then **Download** and click **Download file...** to download the project binary to target (see Figure 70).

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

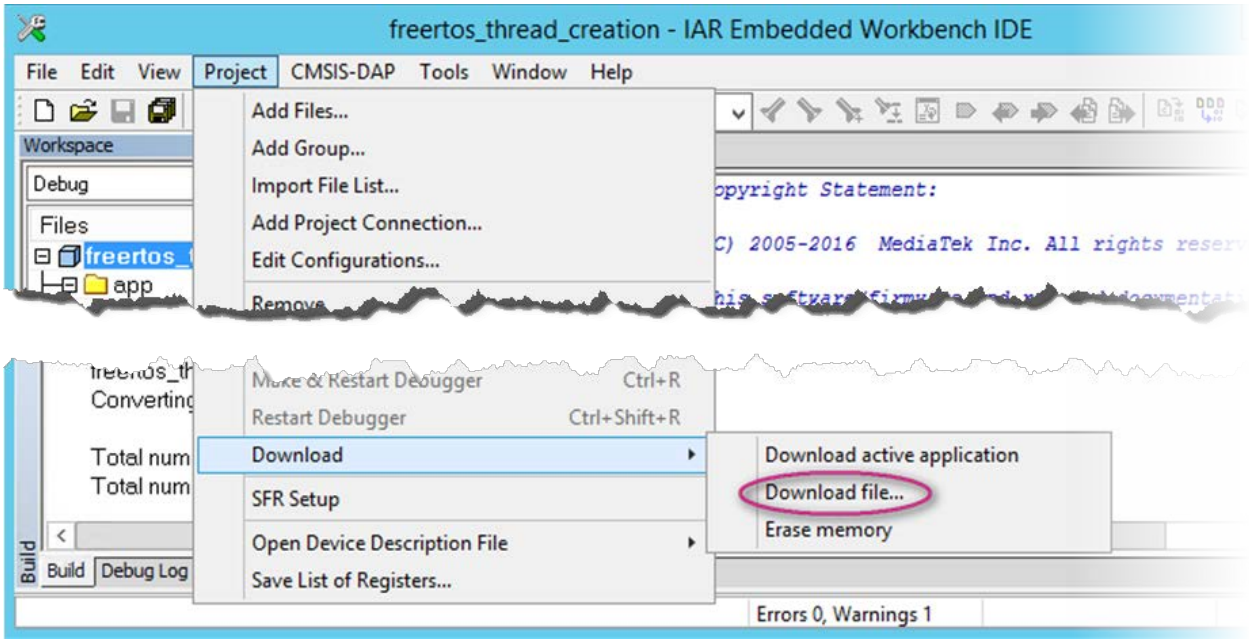

Figure 70. Download the project binary

Note, if you're using the LinkIt 7687 HDK, make sure the board is switched to **Flash Recovery** mode by setting jumper **J25** before resetting the board (see Figure 4). Set the LinkIt 7687 HDK back to **Flash Normal** mode after downloading the binary file to run the project.

4.4.2. Running the project

To run the project:

1) Remove the USB cable to power off the board.

2) Reconnect the USB cable or press the reset button to power on the board.

3) Open **HyperTerminal** (the default: **HyperTerminal** in Windows operation system) program, and configure it as in section "Host UART Configuration" of the Airoha IoT Development Platform for RTOS System Log Developer's Guide, then watch the output log written from the project source files to ARM Cortex-M4 UART port. The following is a reference log example.

```
loader init [2016-04-28 15:03:10.924]
[2016-04-28 15:03:10.924]
fota: TMP is empty, skip upgrade[2016-04-28 15:03:10.939]
[2016-04-28 15:03:10.939]
jump to (0x1007c000) [2016-04-28 15:03:10.939]
total avail space = 13748
[2016-04-28 15:03:10.955]
nvdm init finished
[2016-04-28 15:03:10.955]
[T: 52 M: common C: INFO F: system_init L: 269]: FreeRTOS Running[2016-04-28
15:03:10.970]
```

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

4.5. Debug configuration

The IAR Embedded Workbench IDE uses the CMSIS-DAP/mbed debugger supported on LinkIt 7687 HDK and LinkIt 2523 HDK to debug the project. Here are the debug configurations based on LinkIt 2523 HDK after the project is built and downloaded with IAR Embedded Workbench.

1) Remove the USB cable to power off the board.

2) Reconnect the **2523 USB** port using a micro-USB cable, as shown in Figure 13 and reset the board to power it on.

3) Reconnect the **MK20 port** using a micro-USB cable, as shown in Figure 13 to connect the CMSIS DAP debug interface.

4) Open the IAR Embedded Workbench tool.

5) Navigate to **Project**, and click **Debug without Downloading**, as shown in Figure 71.

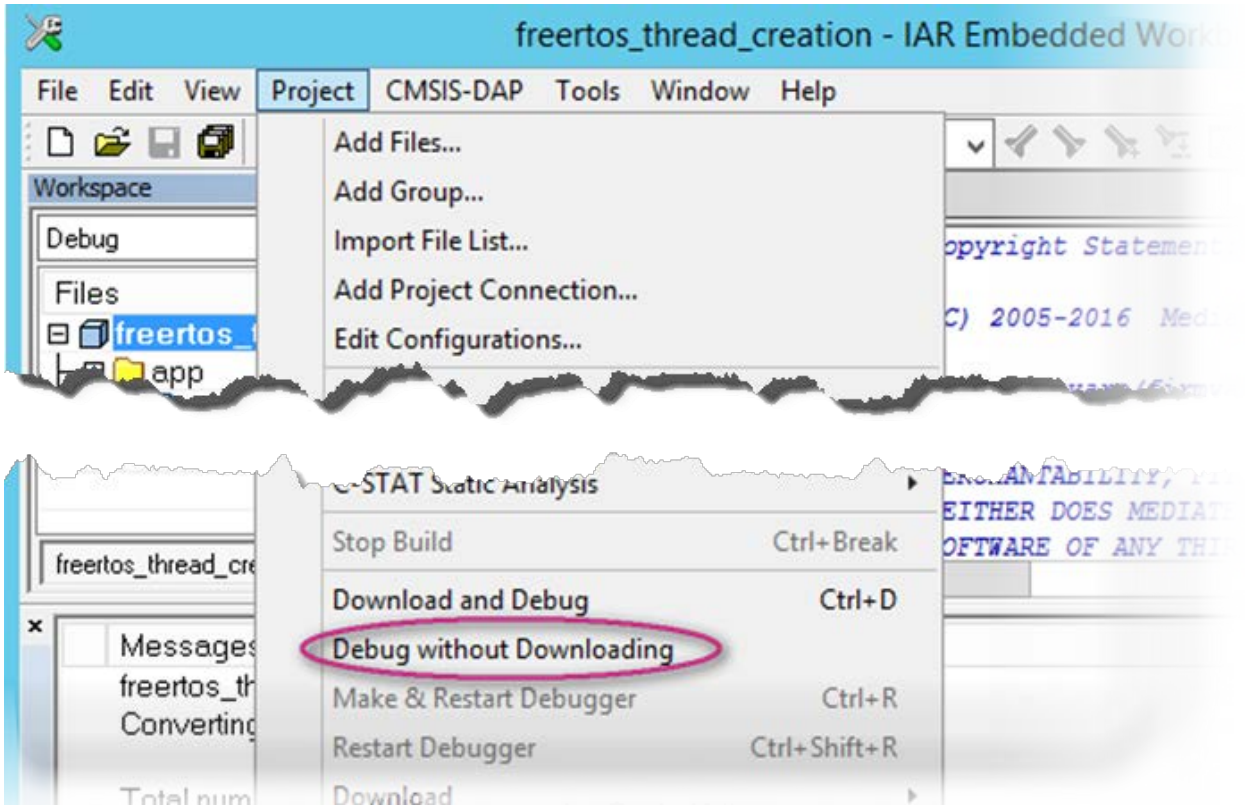

Figure 71. Debugging the project without downloading

6) Click **Debug** to start debugging, as shown in Figure 72.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

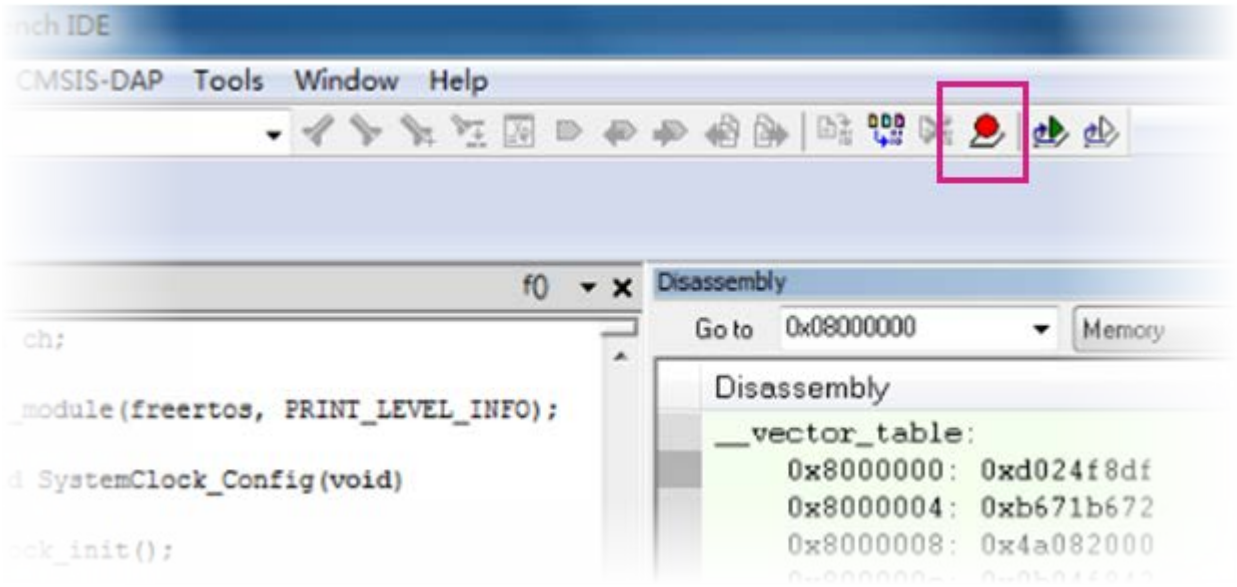

Figure 72. Debugging a project on IAR

Note, IAR debugging is impossible in sleep mode. For more information, please refer to Airoha IoT SDK for RTOS Power Mode Developers Guide under `<sdk_root>/doc`.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

5. Appendix A: Acronyms and Abbreviations

The acronyms and abbreviations used in this get started guide are listed in Table 14.

Table 14. Acronyms and Abbreviations

Abbreviation/Term	Expansion/Definition
ATCI	AT command interface (ATCI), usually AT commands are used to control modems to do their specified functions, they are also used to production line test.
ANSI	The American National Standards Institute
DBCS	A double-byte character set (DBCS) is a character encoding in which all characters (including control characters) are encoded in two bytes.
FOTA	Firmware over the air, a way to update firmware using wireless communication
GNSS	Global Navigation Satellite System
NEC	A kind of Infrared Transmission Protocol, The NEC IR transmission protocol uses pulse distance encoding of the message bits.
NVDM	Non-Volatile Data Management
OEM	Original Equipment Manufacturer (OEM) is a company that makes a part or subsystem that is used in another company's end product.
RC5	A type of infrared transmission protocol. The RC-5 protocol was developed by Philips in the late 1980s as a semi-proprietary consumer IR (infrared) remote control communication protocol for consumer electronics.
RC6	A type of infrared transmission protocol. RC-6 is, as may be expected, the successor of the RC-5 protocol. Like RC-5 the new RC-6 protocol was also defined by Philips. It is a very versatile and well-defined protocol.
TLS	Transport Layer Security (TLS) is cryptographic protocols that provide communications security over a computer network.
WEP	Wired Equivalent Privacy
WPS	Wi-Fi Protected Setup (WPS; originally Wi-Fi Simple Configuration) is a network security standard to create a secure wireless home network.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

6. Appendix B: Disabling Automatic Driver Installation on Windows OS

The automatic download and installation of device drivers can prevent proper installation of the USB COM port driver on Windows 7, 8 and 10 machines. If you've already disabled the automatic installation of device drivers, you can skip this step, otherwise:

1) Open **Control Panel**, search for and open **Change device installation settings**.
2) In **Device Installation Settings** select **No, let me choose what to do**, then click **Never install driver software from Windows Update**, as shown below.

- On Windows 7:

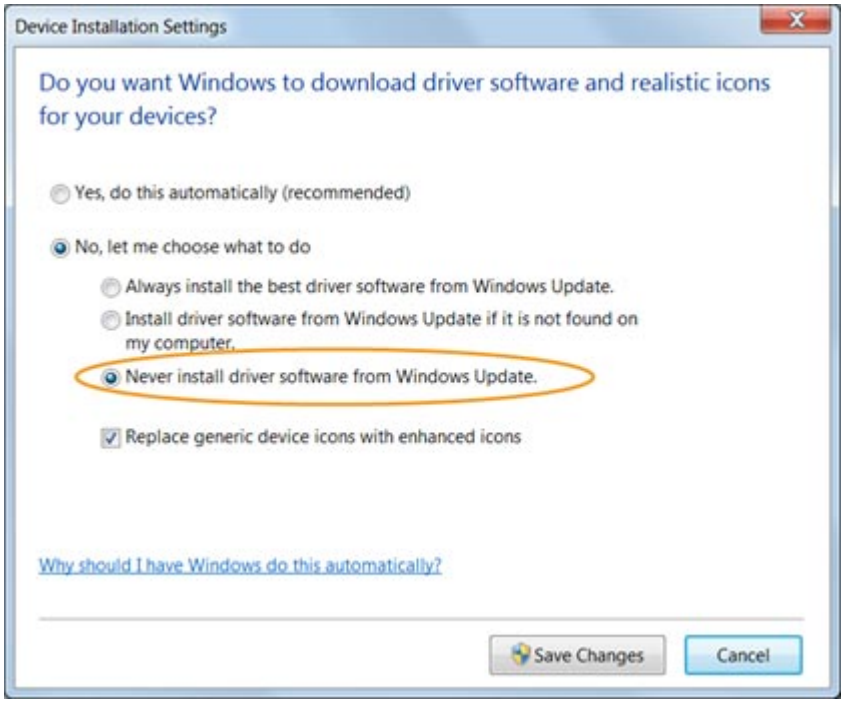

Figure 73. Disabling automatic driver updates on Windows 7 OS

- On Windows 8 and 10:

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.

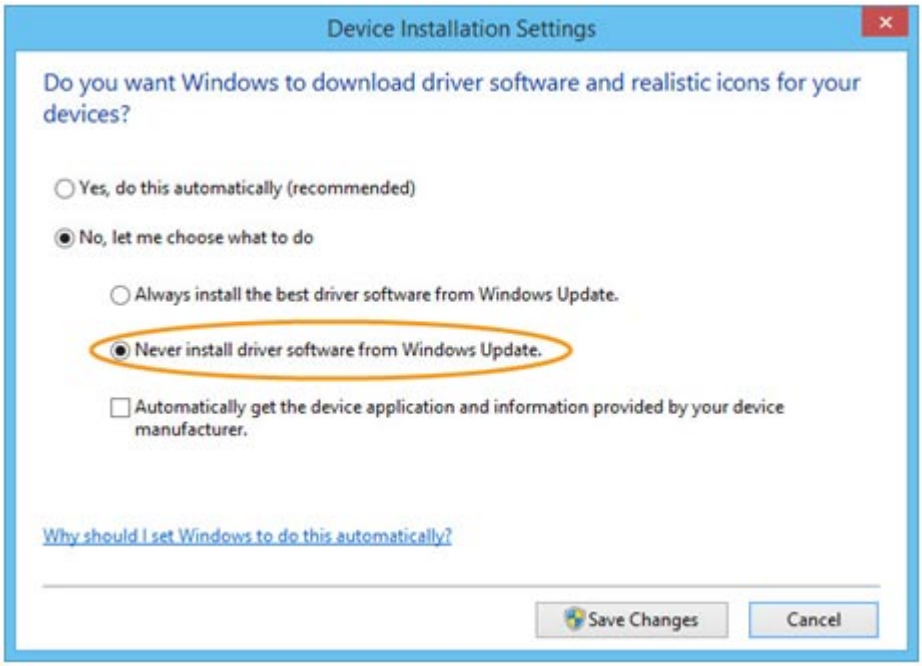

Figure 74. Disabling automatic driver updates on Windows 8 and 10

Also make sure to uncheck **Automatically get the device application and information provided by your device manufacturer**.

3) Click **Save** Changes

You can now install the dedicated USB Driver for your device.

© 2015 - 2019 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. (“Airoha”) and/or its licensor(s).
Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited.