

Introduction to Ameba-Z SDK

Compatible with Ameba-1



Content

- Ameba-Z
- SDK overview
- API of Components
- IDE Tool Demo
- MP Related



Ameba-Z Startup Document

- HW
 - UM0115 Realtek Ameba-Z Introduction.pdf
- Compare to Ameba-1
 - UM0116 Realtek Ameba-Z SDK change.pdf
- Datasheet
 - UM0114 Realtek Ameba-Z Data Sheet.pdf
- DEV board
 - UM0113 Realtek Ameba-Z DEV 1v0 User Manual.pdf
- IAR setup
 - UM0110 Realtek Ameba-Z build environment setup - iar.pdf
- Memory Layout
 - UM0111 Realtek Ameba-Z memory layout.pdf
- OTA
 - AN0110 Realtek Ameba-Z over the air firmware update.pdf



Ameba-Z New Software Feature

■ XIP

- About 140K SRAM free and 200K flash free
- RF Calibration data in EFUSE
- OTA mechanism

■ Security

- Hidden EFUSE key + RDP
- Polar SSL 1.3.8 in ROM (Most code)

■ High-speed Log UART

- For normal debug download or UART upgrade

■ Peripheral API

- Both Light weight and Mbed API

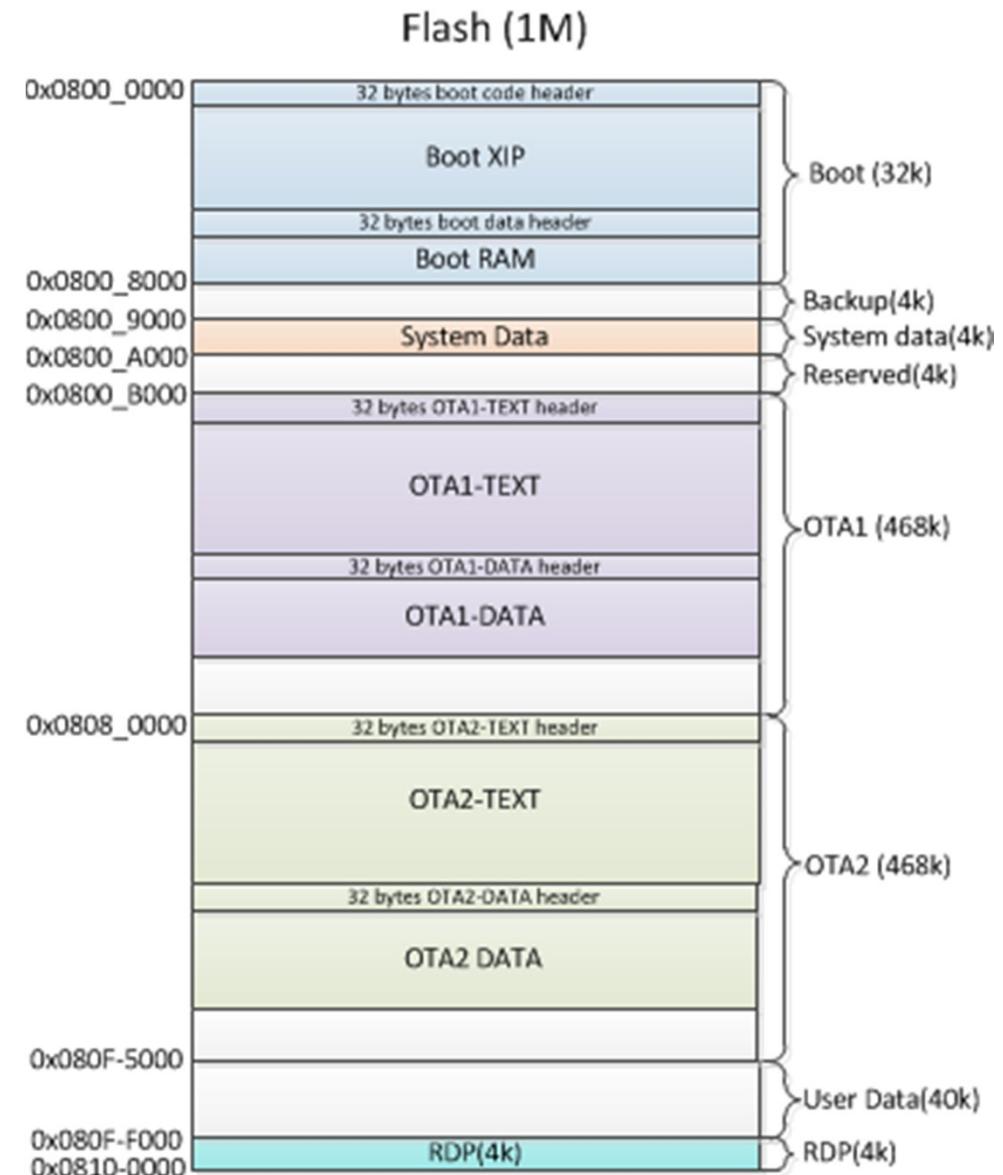
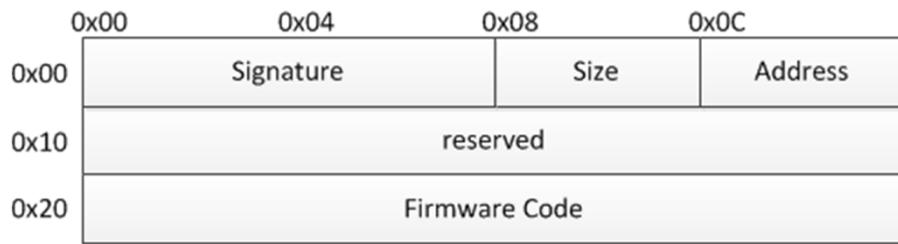
Flash Layout (Ref: UM0111)

Flash Layout

- Bootloader: Image1
- System data
- OTAx image: Image2

Image Header (32Bytes)

- Signature
 - Flash calibration data for image1
 - String "81958711" for image2
- Address
 - Code executes address after boot
 - 'BOOT RAM', 'OTA1 DATA' 'OTA2 DATA' is target RAM address
 - 'BOOT XIP'. 'OTA1 TEXT', 'OTA2 TEXT' is Flash XIP address



System Data (Ref: UM0111)

- Flash Offset: 0x9000
 - OTA2 Flash Address
 - Consistent with Image2
Flash offset selected at
compile time
 - RDP Flash Address
 - Any address from OTA1
 - Length less than 4K
 - Valid IMG2
 - Used for OTA1 and OTA
switch

	0x00	0x04	0x08	0x0C
0x00	OTA2 Flash Address	Valid IMG2	Forth OTA1 GPIO	
0x10	RDP Flash Address	RDP Len (no checksum 4B)		
0x20	WORD1: SPI Speed WORD0: SPI Mode	WORD1: Flash Size WORD0: Flash ID		
		reserved		
0x100 ~ 0x147		USB Parameter		
		reserved		
0x200		ADC Parameter		

Validate Image2 (32 Bits)																																	
																<input checked="" type="checkbox"/> 0		<input type="checkbox"/> 1															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
• • • •																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		

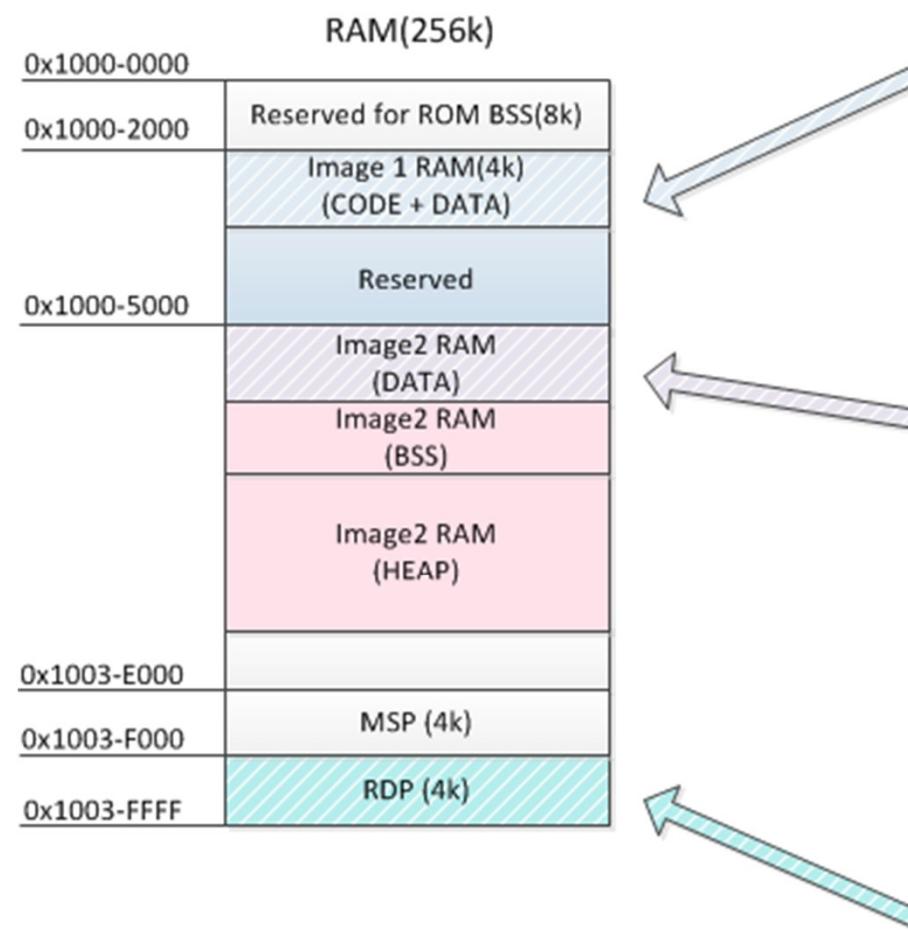
System data

- Flash speed
 - 0xFFFF: 100MHz
 - 0x7FFF: 83MHz
 - 0x3FFF: 71MHz
 - 0x1FFF: 62MHz
 - 0x0FFF: 55MHz
 - 0x07FF: 50MHz
 - 0x03FF: 45MHz
- Flash size
 - 0xFFFF: 2MB
 - 0x7FFF: 32M
 - 0x3FFF: 16M
 - 0x1FFF: 8MB
 - 0x0FFF: 4MB
 - 0x07FF: 2MB
 - 0x03FF: 1MB
- Flash SPI mode
 - 0xFFFF: Read quad IO, Address & Data 4 bits mode
 - 0x7FFF: Read quad O, Just data 4 bits mode
 - 0x3FFF: Read dual IO, Address & Data 2 bits mode
 - 0x1FFF: Read dual O, Just data 2 bits mode
 - 0x0FFF: 1 bit mode

RAM Layout (Ref: UM0111)

■ SRAM layout

- ROM BSS
- Image 1 used SRAM
- Image 2 used SRAM
- MSP
- RDP: Fixed location



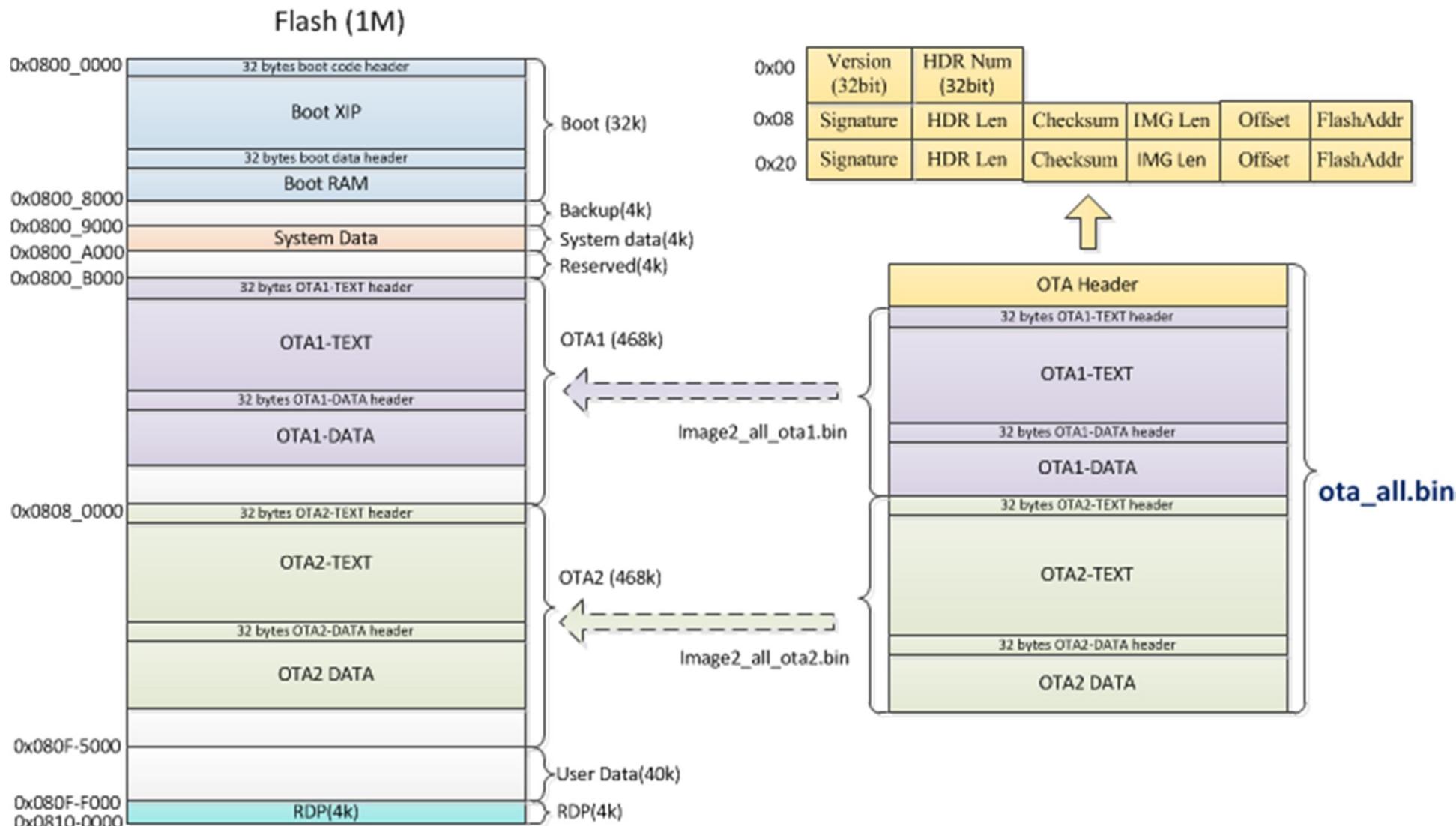
FLASH
32 bytes boot code header
Boot XIP
32 bytes boot data header
Boot RAM
System Data
32 bytes OTA1-TEXT header
OTA1-TEXT
32 bytes OTA1-DATA header
OTA1-DATA
32 bytes OTA2-TEXT header
OTA2-TEXT
32 bytes OTA2-DATA header
OTA2 DATA
User Data
RDP(4k)



Output Binary

- Boot loader
 - boot_all.bin
 - Boot loader code and initial value of data
- Application Image2
 - image2_all_ota1.bin
 - Application code and initial value of data
 - Run in flash **0x0800B000**
 - image2_all_ota2.bin
 - Application code and initial value of data
 - Default run in flash **0x08080000**
- OTA image
 - ota_all.bin
 - Integrated OTA header, image2_all_ota1.bin and image2_all_ota2.bin

OTA Update (Ref: AN0110)



OTA Header

- Version
 - The version of OTA image
- Header Number
 - The number of OTA Entry Header
- OTA Entry Header
 - Signature
 - "OTA1" for OTA1, and "OTA2" for OTA2
 - Header Length
 - The length of OTAx header
 - Checksum
 - The checksum of OTAx image
 - Image Length
 - The size of OTAx image
 - Offset
 - The start position of OTAx in current image
 - Flash Address
 - Address in flash where OTAx will be programmed

0x00	Version (32bit)	HDR Num (32bit)				
0x08	Signature	HDR Len	Checksum	IMG Len	Offset	FlashAddr
0x20	Signature	HDR Len	Checksum	IMG Len	Offset	FlashAddr



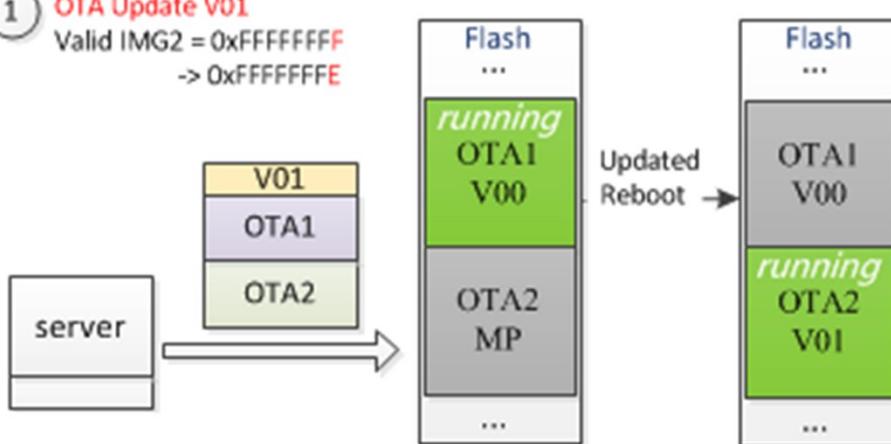
OTA Steps

1. Received upgrade command from the cloud server.
2. Find the target OTA area in flash according to “Valid IMG2” and erase.
3. Start receiving ota_all.bin from the server.
4. Find the correct OTA Entry header in OTA header.
5. Find the correct otax.bin start offset written in OTA Entry header .
6. Save otax.bin to the target flash area.
7. Calculate checksum of otax.bin in flash and compare to the checksum in OTA Entry header. If not equal, OTA fail.
8. If target is OTA2, verify OTA2 address in 0x9000 (flash system data) is the same as the flash offset in OTA Entry header. If not equal, OTA fail.
9. **Update Validate IMG2 in 0x9004 (flash system data).**
10. Reboot.

OTA Example

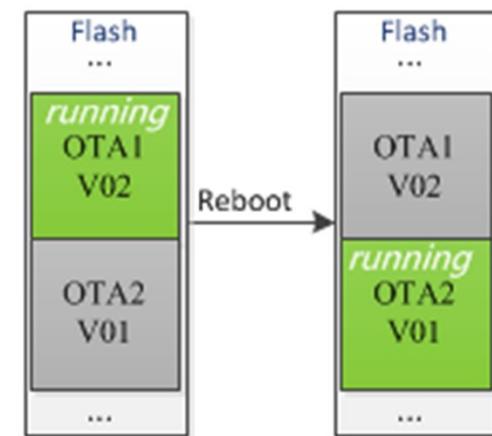
1 OTA Update V01

Valid IMG2 = 0xFFFFFFFF
-> 0xFFFFFFFFE



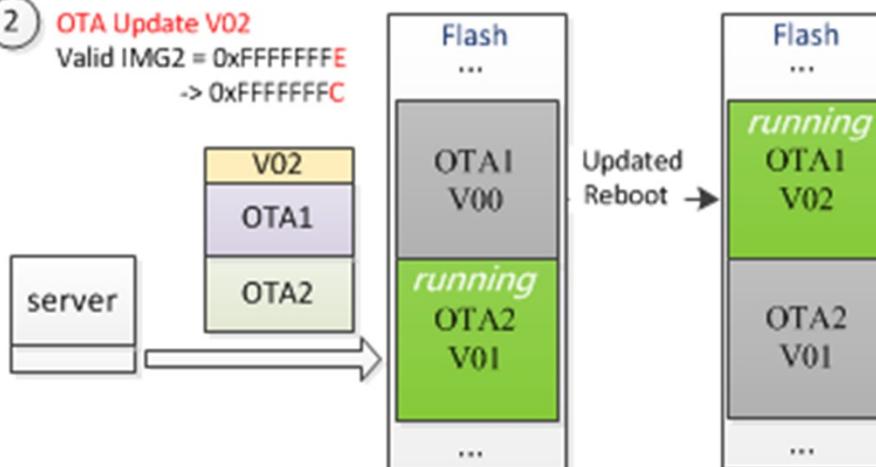
3 ATSC

Valid IMG2 = 0xFFFFFFFFC
-> 0xFFFFFFFF8



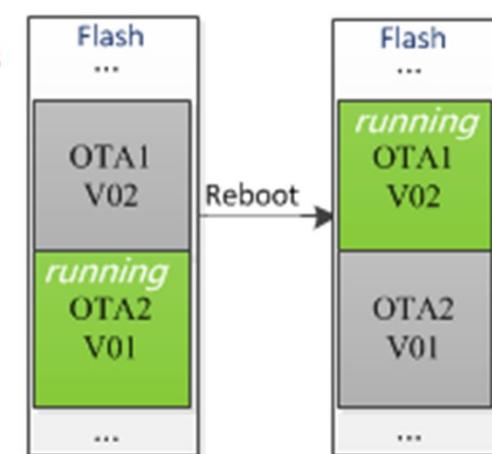
2 OTA Update V02

Valid IMG2 = 0xFFFFFFFFE
-> 0xFFFFFFFFC



4 ATSR

Valid IMG2 = 0xFFFFFFFF8
-> 0xFFFFFFFF0





FW protection – Read Protection

■ RDP

- Top 4k RAM can not be read.
- RDP Interrupt will happen when invalid access happen.
- RDP image should be encrypted use RDP KEY
- RDP image can only be decrypted and load to RDP RAM use IPSEC.

■ KEY

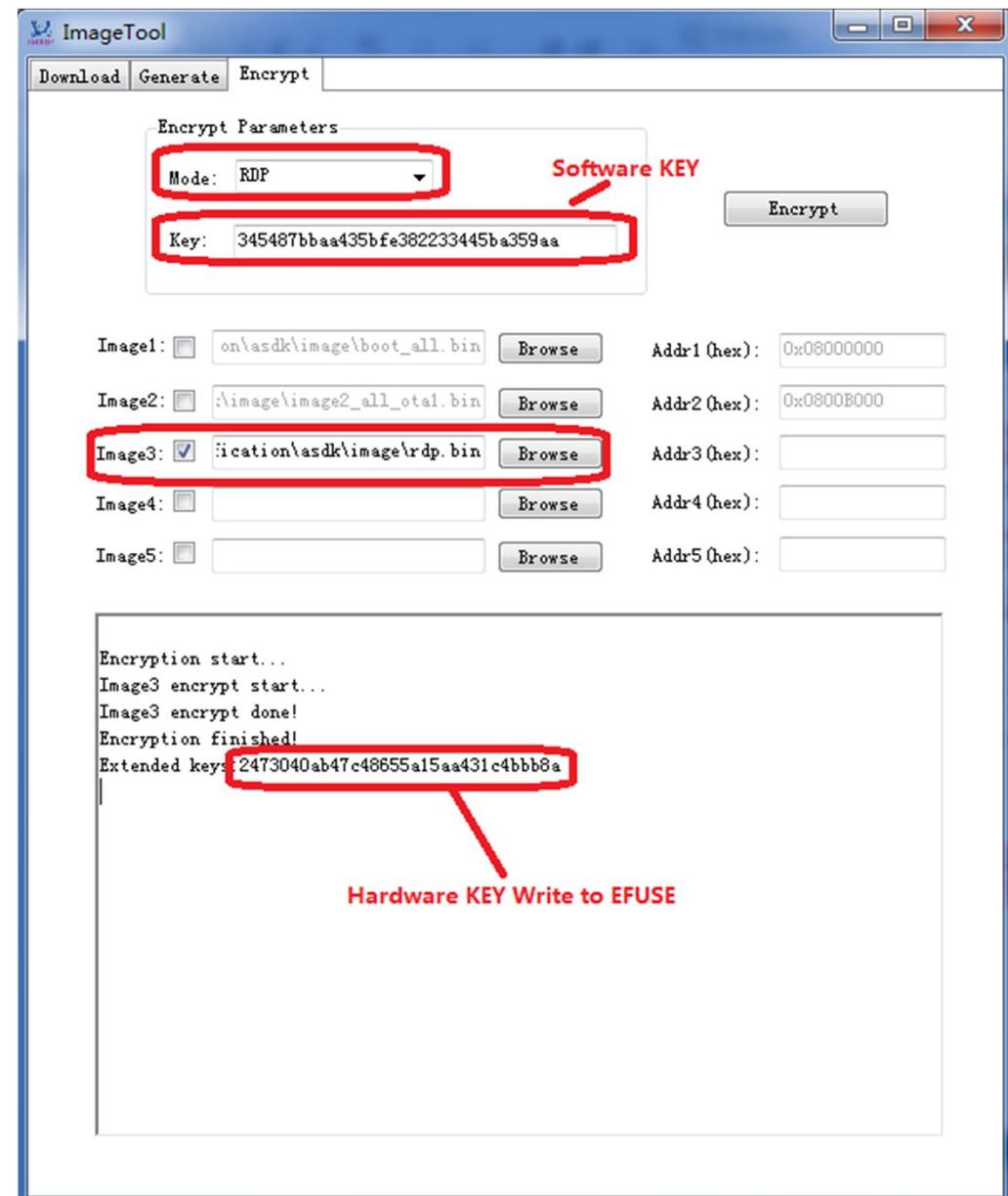
- 16B RDP key should be written to EFUSE RDP key area
 - Hidden EFUSE 0xB0~0xBF
- Can not read back again. (HW protect)
- Auto-load to IPSEC when boot.

■ Enable

- Hidden EFUSE 0xC0[0].
- Can not be closed after open.

How to use RDP

- Create IAR Project for RDP
 - Save algorithms or special datas need be protected to a c file
 - Link Address must be 0x1003F000
 - Output file: rdp.bin
- Enable RDP protection
 - efuse_rdp_enable
- Input **RDP software key** and rdp.bin to Image Tool Encrypt page, click Encrypt, then **rdp-en.bin** generated in the same directory as rdp.bin
- Set **RDP Hardware Key** to hidden EFUSE through RDP API
 - efuse_rdp_keyset
- Finally write rdp-en.bin to Flash

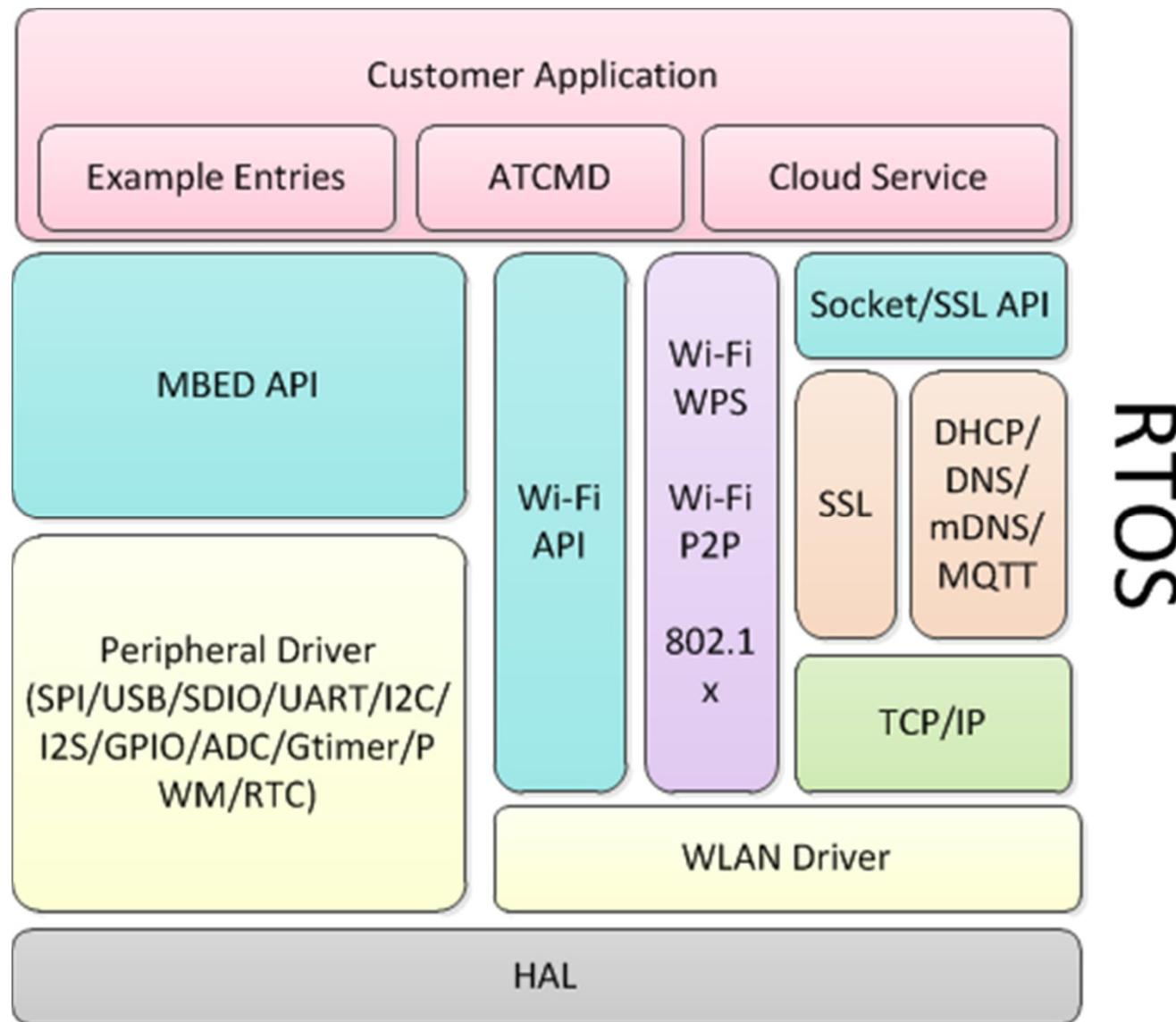




Content

- Introduction to Ameba-Z
- SDK overview
- API of Components
- IDE Tool Demo
- MP Related

Software Stack

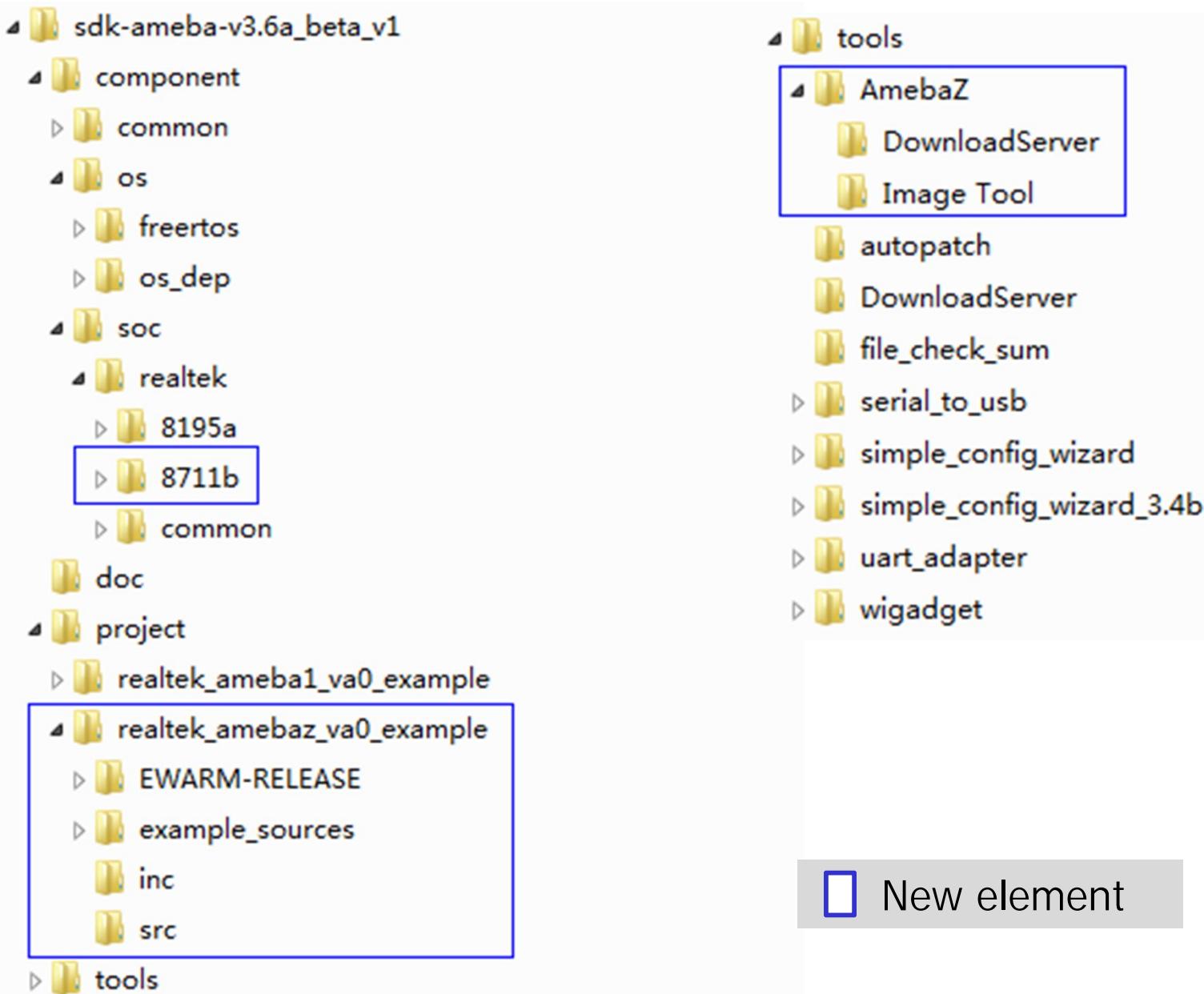




Software Features

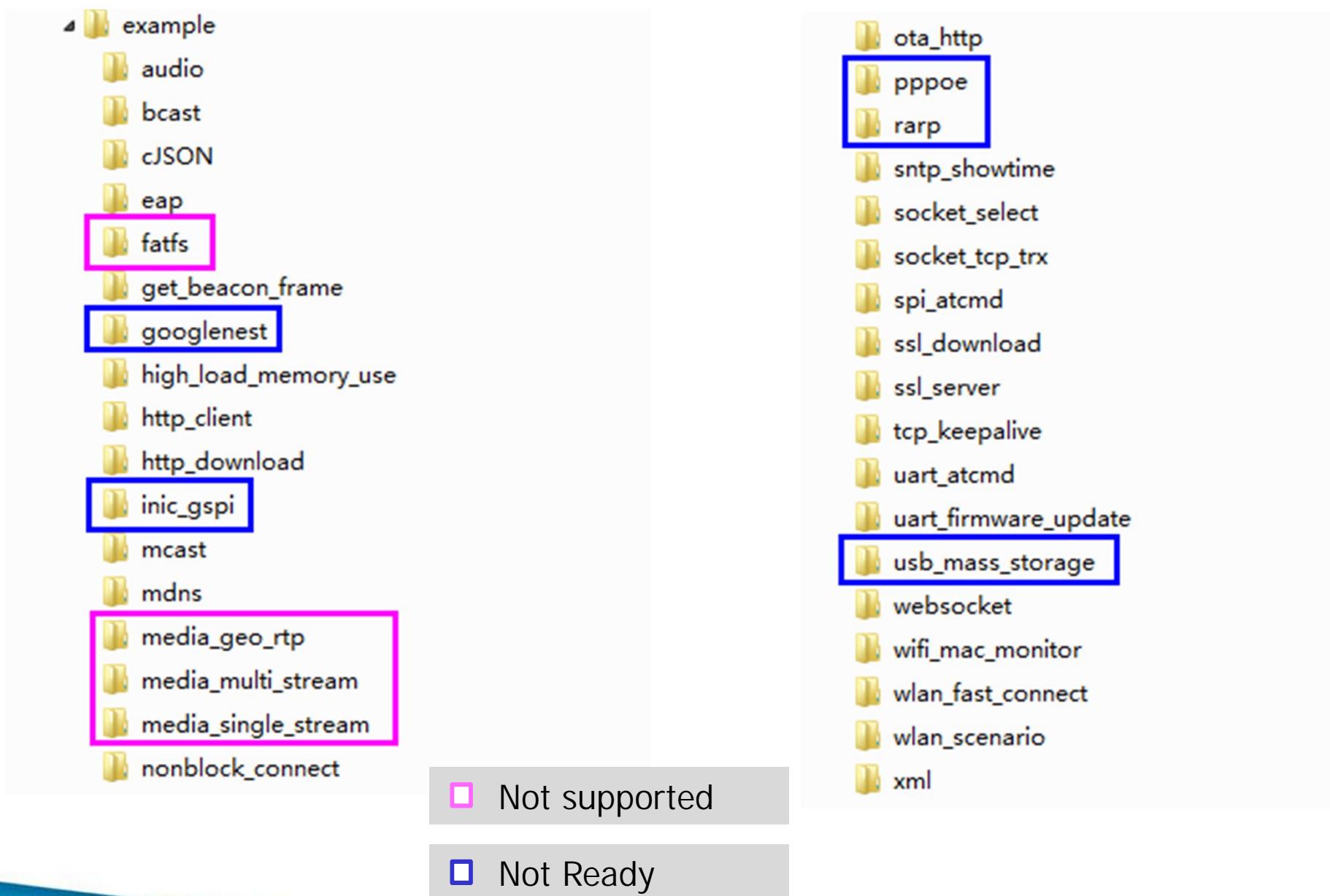
- Operation System
 - FreeRTOS
- Network Stack
 - LW/IP
- WLAN Security
 - Open/WEP/TKIP/AES PSK
- Architecture
 - STA mode
 - AP mode
 - STA+AP mode
 - Promiscuous mode
 - P2P mode
- Device Simple Config
 - SoftAP mode config
 - WPS
 - Realtek simple config
 - Customizable Promiscuous Mode
- Secure Sockets Layer
 - Polar SSL (Ref: AN0012)
 - Mbed TLS 2.4.0
- Instant messaging protocol
 - MQTT
 - Web Socket
- Application examples
- Peripheral operation examples
- Update Firmware
 - OTA update
 - UART upgrade
- Cloud
 - Homekit (Ref: AN0035)
 - Google Nest (Ref: AN0038)
 - Gitwits (Ref: UM0062)
 - Joylink (Ref: AN0052)
 - Wechat/Airkiss (Ref: AN0054)
 - QOLink (Ref: UM0074)
 - Ali Alink (Ref: UM0098)
 - Amazon
 - Hilink (Ref: UM0095)
- Application
 - Wi-Fi RS 232 (Ref: AN0046)

Directory Structure – Overview



New element

Directory Structure – Common Example



Directory Structure – Peripheral example

example_sources	gtimer	spi_multislave
analogin	i2c	spi_pl7223
analogin_voltage	i2c_dual	spi_stream_twoboard
crypto	i2c_epl2197_heartrate	spi_stream_twoboard_concurrent
crypto-customize	i2c_epl2590_light	spi_twoboard
efuse_mtp	i2c_epl2590_proximity	uart
efuse_otp	i2c_LPS25HB_pressure	uart_auto_flow_ctrl
flash	i2c-shtc1	uart_clock
flash_micron_block_protect	i2s	uart_irq
flash_setstatus	i2s_bypass	uart_stream_2_threads
gdma	i2s_tx_and_rx_only	uart_stream_4_threads
gpio	log_uart_char_loopback	uart_stream_dma
gpio_dht_temp_humidity	log_uart_loopback	uart_stream_irq
gpio_HC_SR04_ultrasonic	log_uart_stream_loopback	uart_stream_rx_timeout
gpio_irq	pm_deepsleep	uart_stream_rx_timeout_by_GTimer
gpio_jtag	pm_deepstandby	uart_stream_rx_timeout_by_semaphore_jar
gpio_level_irq	pm_sleep	uart_stream_rx_timeout_by_SoftTimer
gpio_light_weight	pm_tickless	uart_stream_rx_timeout_by_WaitSemaphore
gpio_port	pwm	uart_stream_tx_rx_concurrent_jar
gpio_pulse_measure	pwm-buzzer	usb_uvc
gspi	rtc	watchdog
gspi_fw_downloader	sdio_device	wlan
gspi_fw_loader	spi	wlan_inic
gspi_mp_downloader	spi_gpio_chipselect	
	spi_master_write_read_one_byte	

Freertos

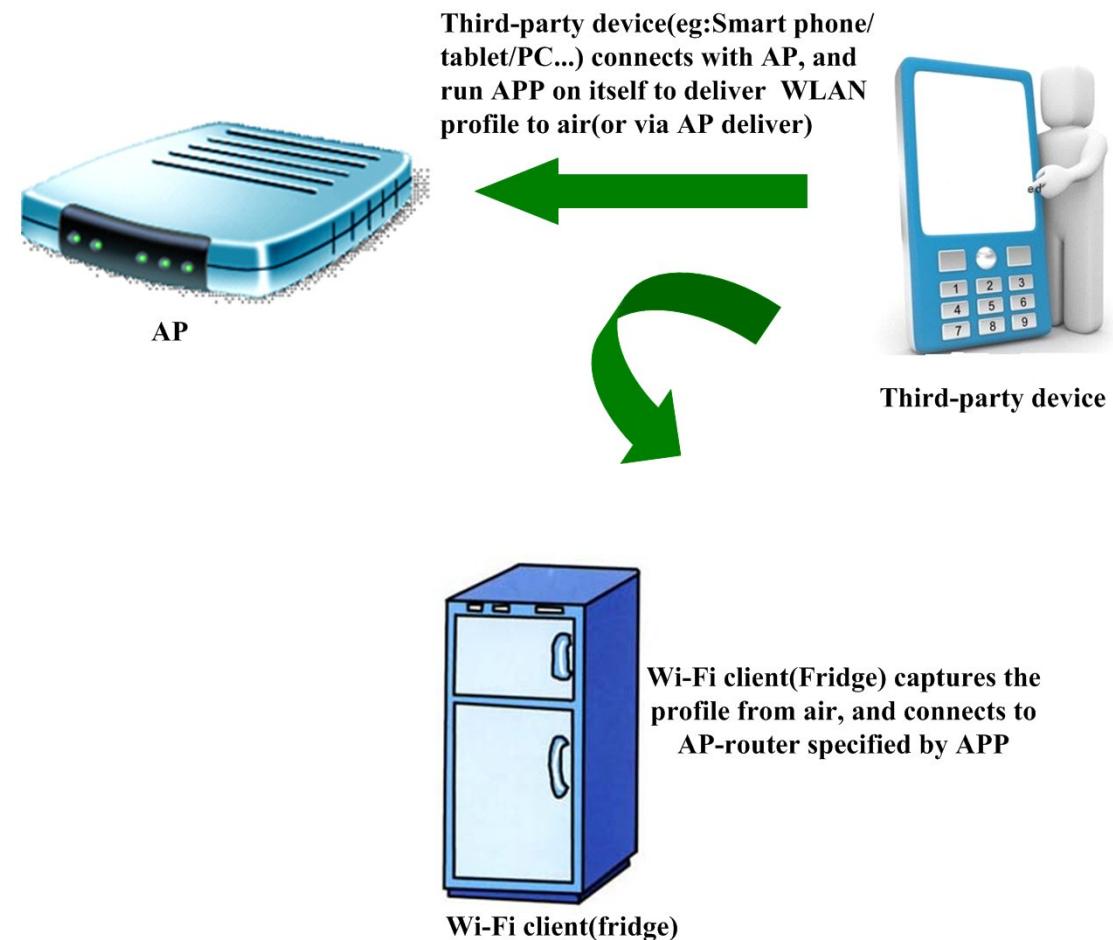
- Minimal RAM and processing overhead
 - Typically an RTOS kernel binary image will be in the region of 4K to 9K bytes.
 - The core of the FreeRTOS kernel is contained in only 4 C files.
 - Tasks.c
 - Queue.c
 - Heap_5.c
 - Timer.c
- Real time, reliable and scalable

LWIP

- Lightweight and open source TCP/IP stack
- Provide basic features of TCP Protocol with decreased system occupation
- Fit for small embedded applications , requires only 20K RAM and 40K ROM
- Support protocols
 - IP protocol
 - ARP protocol
 - ICMP protocol
 - UDP protocol
 - TCP protocol including Congestion Control, RTT Estimation and Fast Recovery/Fast Retransmit

Simple Config R3 (Ref: AN0011)

- SoftAP + Promiscuous mode
- As fast as old version
- High success rate
- Cover both Android and iOS
- Cost Flash 14K and SRAM 0.5K



Cloud Service

Cloud Service	Code	Data + BSS	Status
Homekit	-	-	Ready
Google	-	-	Ready
Amazon	-	-	Ready
Alink 1.1	60K+50K	68K	Ready
Joylink 1.3.3	61K	36K	Ready
QQlink 1.1.101	70K	44K	Ready
Hilink 0.5.4	73K	17K	Ready
Gagent + Airkiss	77K	22K	Ready
Wechat 3.1.0	89K	35K	Ready



Content

- Introduction to Ameba-Z
- SDK overview
- API of Components
- IDE Tool Demo
- MP Related



AT Command for Wi-Fi (Ref: AN0025)

- 'AT??' Print Log History
- 'AT--' Exit Log Service
- 'ATW0' Wlan Set Network SSID
- 'ATW1' Wlan set Network Passphrase
- 'ATW2' Wlan Set Key ID
- 'ATWC' Wlan Join a Network
- 'ATWD' Wlan Disconnect from Network
- 'ATW3' Wlan Set Access Point SSID
- 'ATW4' Wlan Set Access Point Security Key
- 'ATW5' Wlan Set Access Point Channel
- 'ATWA' Wlan Activate Access Point
- 'ATWB' Wlan Activate Access Point mode and Station mode
- 'ATW?' Wlan Show WiFi information
- 'ATWS' Wlan Scan for Network Access Point
- 'ATWR' Wlan Get RSSI of Associated Network Access Point
- 'ATWM' Wlan Wi-Fi promisc
- 'ATWE' Wlan Start Web Server
- 'ATWQ' Wlan Wi-Fi Simple Config
- 'ATWP' Wlan Power on/off wifi module
- 'ATWI' Wlan ping test
- 'ATWO' Wlan OTA update
- 'ATWT' Wlan TCP throughput test
- 'ATWU' Wlan UDP test
- 'ATWL' Wlan SSL client
- 'ATWW' Wlan Wi-Fi Protected Setup
- 'ATWZ' Wlan IWPRIV

\$sdk\component\common\api\at_cmd\atcmd_wifi.c



AT Command for system

- 'ATSD' Dump register
- 'ATSE' Edit register
- 'ATSC' Clear OTA signature
- 'ATSR' Recover OTA signature
- 'ATSA' MP ADC test
- 'ATSG' MP GPIO test
- 'ATSP' MP Power related test
- 'ATSB' OUT PIN setup
- 'ATSS' Show CPU stats
- 'ATSM' Apple CP test
- 'ATSJ' Turn off JTAG
- 'ATS@' Debug message setting
- 'ATS!' Debug configure setting
- 'ATS#' Test command
- 'ATS?' Help



WiFi common API (Ref:UM0006)

- Wifi enable/disable
 - wifi_on
 - wifi_off
 - wifi_is_up
 - wifi_is_ready_to_transceive
- Station Mode Connection
 - wifi_connect
 - wifi_disconnect
- AP Mode Startup
 - wifi_start_ap
 - wifi_restart_ap
 - wifi_get_ap_info
 - wifi_get_associated_client_list
- AP+STA Concurrent Mode
 - wifi_start_ap
 - wifi_connect
- Wifi Scan
 - wifi_scan_networks
 - wifi_set_pscan_chan
- Wlan Driver Indication
 - wifi_indication
- Wifi Promiscuous Mode
 - wifi_enter_promisc_mode
 - wifi_set_promisc
 - wifi_init_packet_filter
 - wifi_add_packet_filter
 - wifi_enable_packet_filter
 - wifi_disable_packet_filter
 - wifi_remove_packet_filter
- Wifi Setting Information
 - wifi_get_setting
 - wifi_show_setting



WiFi common API

- WiFi Mac Address
 - wifi_set_mac_address
 - wifi_get_mac_address
- WiFi Power save
 - wifi_enable_powersave
 - wifi_disable_powersave
- WiFi Tx Power
 - wifi_set_txpower
 - wifi_get_txpower
- WiFi Channel
 - wifi_set_channel
 - wifi_get_channel
- WiFi Multicast Address
 - wifi_register_multicast_address
 - wifi_unregister_multicast_address
- WiFi RF Control
 - wifi_rf_on
 - wifi_rf_off
- WiFi Auto Reconnection
 - wifi_set_autoreconnect
 - wifi_get_autoreconnect
- WiFi Custom IE
 - wifi_add_custom_ie
 - wifi_update_custom_ie
 - wifi_del_custom_ie
- WiFi RSSI Information
 - wifi_get_rssi
- Country Code Setup
 - wifi_set_country
- Network Mode Setup
 - wifi_set_network_mode

\$sdk\component\common\api\wifi\wifi_conf.c

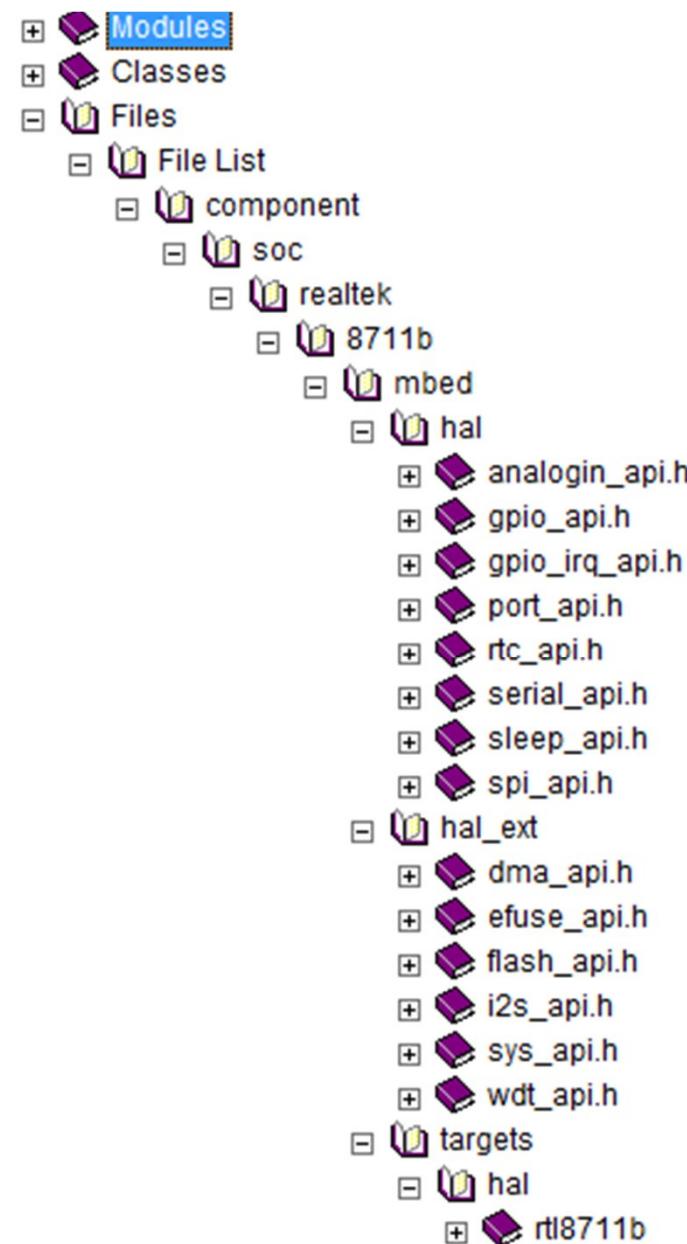
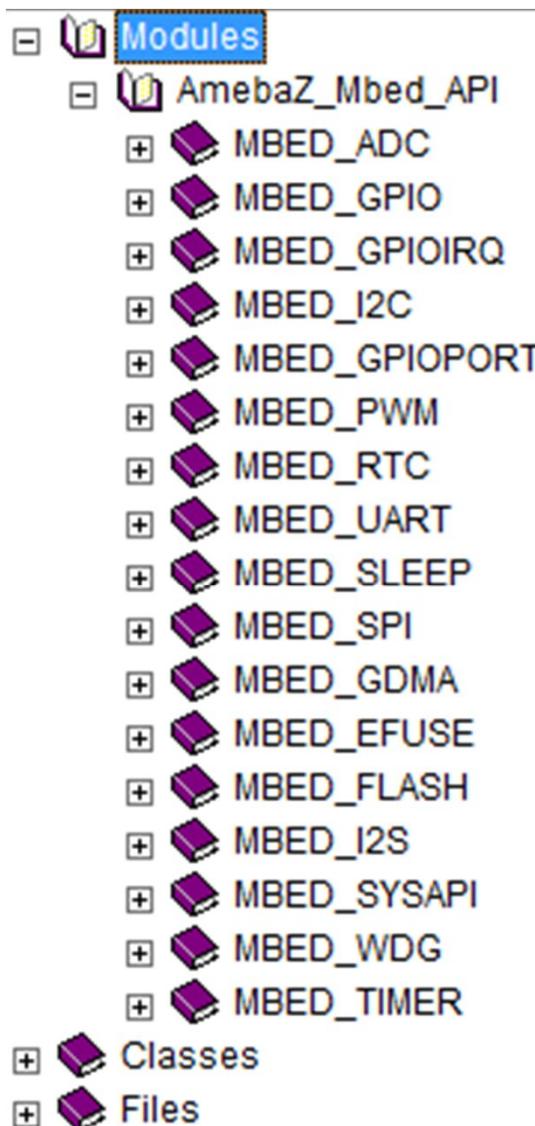


Peripheral API (Ref: UM0117)

- Modules
 - AmebaZ_Outline
 - AmebaZ Address Map
 - AmebaZ_Peripheral_Registers_Structures
 - + AMEBAZ_UART
 - + AMEBAZ_SPI
 - + AMEBAZ_SPIC
 - + AMEBAZ_ADC
 - + AMEBAZ_I2C
 - + AMEBAZ_I2S
 - + AMEBAZ_TIMER
 - + AMEBAZ_RTC
 - + AMEBAZ_PINMUX
 - + AMEBAZ_GPIO
 - + AMEBAZ_IPSEC
 - + AMEBAZ_USOC
 - + AMEBAZ_NCO32k
 - + AMEBAZ_NCO8M
 - + AMEBAZ_BACKUP_REG
 - + AMEBAZ_CACHE
 - AmebaZ_Peripheral_Declarations
 - AmebaZ_Platform
 - + BKUP_REG
 - + CLOCK
 - + DELAY
 - + OTA
 - + PIN
 - + PMC
 - + CACHE
 - + DIAG
 - + EFUSE
 - + PROTECTION
 - + RCC
 - + SYSCFG
 - + IRQ
 - + Debug
 - AmebaZPeriph_Driver
 - + CRYPTO
 - + DONGLE
 - + INIC
 - + SDIO
 - + USOC
 - + ADC
 - + FLASH
 - + GDMA
 - + GPIO
 - + I2S
 - + RTC
 - + SPI
 - + Timer
 - + UART
 - + WDG
 - + I2C



Mbed API (Ref: UM0118)



□ File Members



LWIP API

- Socket
- Shutdown
- Bind
- Listen
- Accept
- Connect
- Recv
- Recvfrom
- Send
- Sendto
- Select
- ioctlsocket
- Read
- Write
- Close
- tcp_new
- tcp_accept
- tcp_recv
- tcp_sent
- tcp_poll
- tcp_recved
- tcp_bind
- tcp_connect
- tcp_listen
- tcp_abort
- tcp_close
- tcp_write
- udp_new
- udp_remove
- udp_bind
- udp_connect
- udp_recv
- udp_send

\$sdk\component\common\network\lwip\lwip_v1.4.1\src\api\sockets.c

Freertos API

- Task
 - rtw_create_task
 - rtw_delete_task
 - rtw_wakeup_task
 - rtw_get_scheduler_state
- Queue
 - rtw_init_xqueue
 - rtw_push_to_xqueue
 - rtw_pop_from_xqueue
 - rtw_deinit_xqueue
- Semaphore
 - rtw_init_sema
 - rtw_free_sema
 - rtw_up_sema
 - rtw_up_sema_from_isr
 - rtw_down_timeout_sema
- Delay
 - rtw_mdelay_os
- Timer
 - rtw_timerCreate
 - rtw_timerDelete
 - rtw_timerIsTimerActive
 - rtw_timerStop
 - rtw_timerChangePeriod
- Memory
 - rtw_malloc
 - rtw_zmalloc
 - rtw_mfree
 - rtw_getFreeHeapSize
- Wake Lock
 - rtw_acquire_wakelock
 - rtw_release_wakelock
 - rtw_wakelock_timeout
- Time
 - rtw_get_current_time
- Random
 - rtw_get_random_bytes

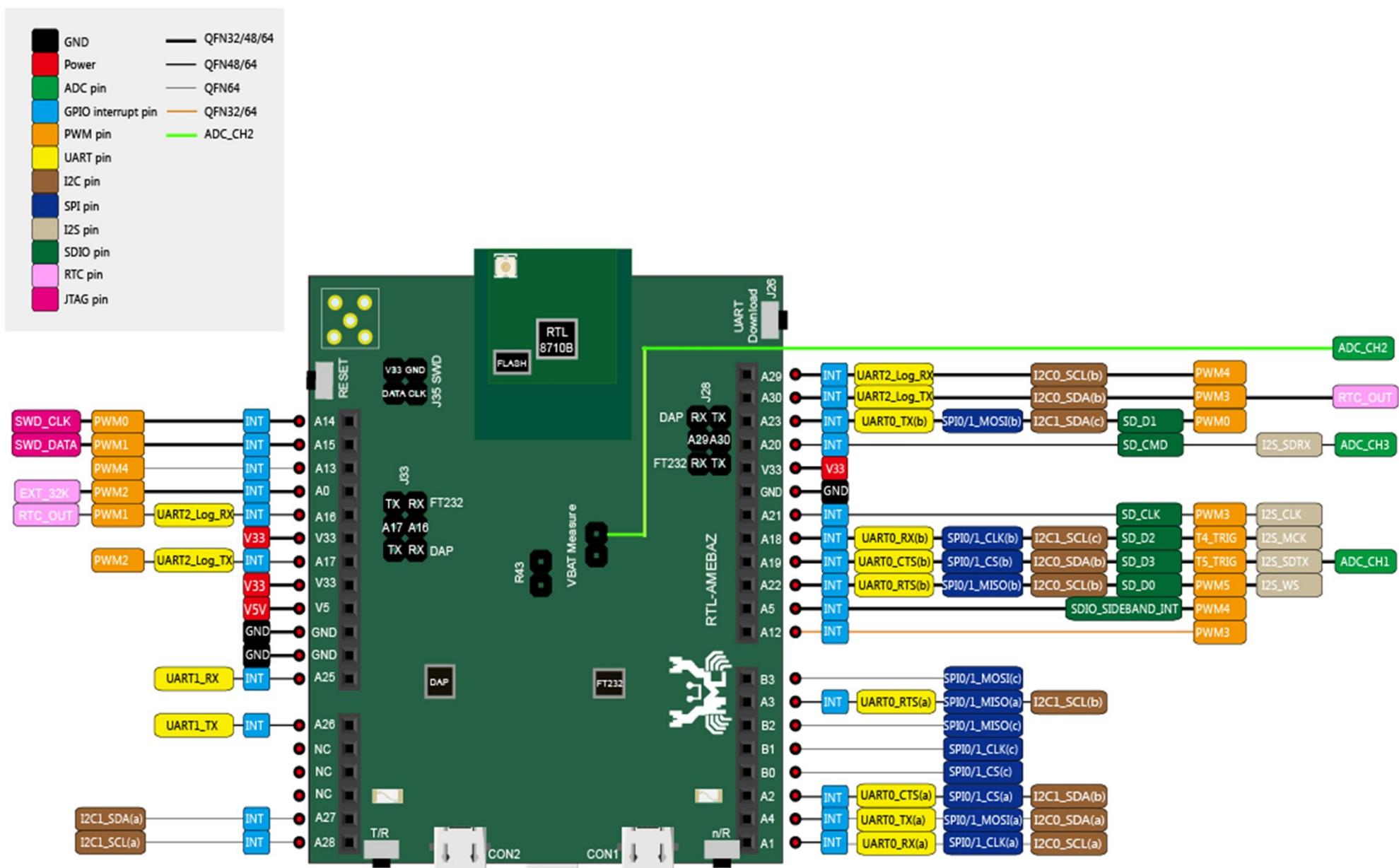
\$sdk\component\os\os_dep\osdep_service.c



Power Saving Related API

- CM4 power mode
 - sleep_ex
 - deepstandby_ex
 - deepsleep_ex
- Wi-Fi Power Save
 - wifi_enable_powersave
 - wifi_disable_powersave
 - wifi_set_power_mode
 - wifi_set_lps_dtim
 - wifi_get_lps_dtim
- Wakeup Event
 - DSLEEP_WAKEUP_BY_TIMER/GPIO
 - STANDBY_WAKEUP_BY_STIMER/GPI
O/RTC
 - SLEEP_WAKEUP_BY_STIMER/GTIME
R/GPIO_INT/WLANSADIO/USB/GPIO/
UART/I2C/RTC
- PM tickles
 - Suspend and Resume callback
 - pmu_register_sleep_callback
 - pmu_unregister_sleep_callback
 - Wake Lock
 - pmu_acquire_wakelock
 - pmu_release_wakelock
 - pmu_get_wakelock_status
 - Wakeup Event
 - add_wakeup_event
 - del_wakeup_event
 - PMU_DEVICE: nDeviceId
 - PMU_OS/USER_BASE/WLAN/LO
GUART/CONSOL/SDIO/UART0/U
ART1/RTC/I2C0/I2C1/ADC/USOC
/DONGLE

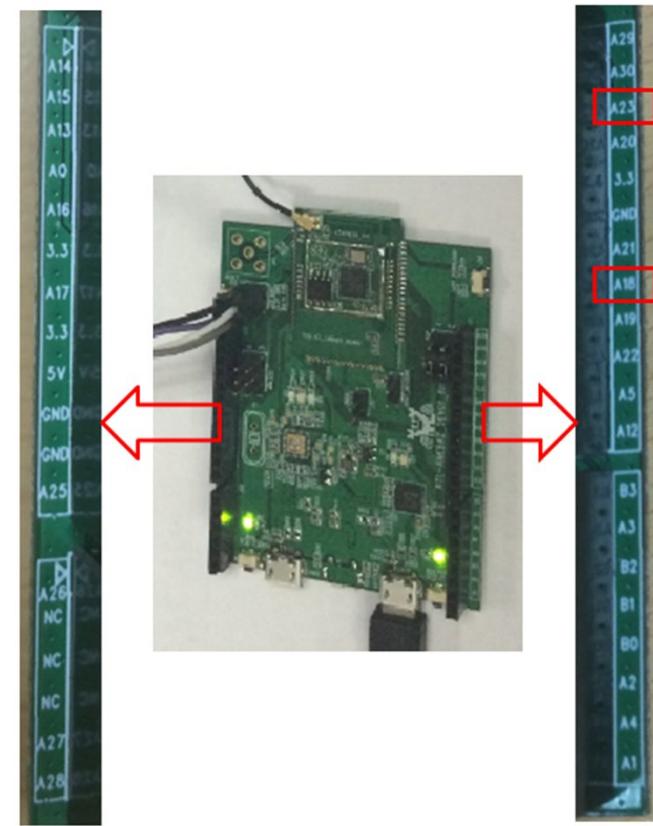
DEV Board PIN Multiplex



How to find UART0

- Firstly, find UART0 groups in Pin mux table defined in UM0113 Chapter 3 and shown in Blue and Red box in the lower left corner.
- Then pick the pin group which is enabled for your package. For example, PA_4/PA_1 is not available for QFN32.
- Finally, find the pin name in the right side of the dev board shown as below.

QFN68	QFN48	QFN32	GPIO	UART	SPI Master	SPI Slave
✓	✓	✓	PA_14			
✓	✓	✓	PA_15			
✓			PA_13			
✓	✓	✓	PA_0			
✓	✓		PA_16	UART2_log_RXD		
✓	✓		PA_17	UART2_log_TXD		
✓	✓		PA_25	UART1_RXD		
✓	✓		PA_26	UART1_TXD		
✓			PA_28			
✓			PA_27			
✓		✓	PA_12			
✓	✓		PA_4	UART0_TXD	SPI1_MOSI	SPI0_MOSI
✓	✓		PA_1	UART0_RXD	SPI1_CLK	SPI0_SCK
✓	✓		PA_2	UART0_CTS	SPI1_CS	SPI0_CS
✓	✓		PA_3	UART0_RTS	SPI1_MISO	SPI0_MISO
✓	✓	✓	PA_6			
✓	✓	✓	PA_7			
✓	✓	✓	PA_8			
✓	✓	✓	PA_9			
✓	✓	✓	PA_10			
✓	✓	✓	PA_11			
✓	✓	✓	PA_5			
✓	✓		PA_18	UART0_RXD	SPI1_CLK	SPI0_SCK
✓	✓		PA_19	UART0_CTS	SPI1_CS	SPI0_CS
✓	✓		PA_20			
✓	✓		PA_21			
✓	✓	✓	PA_22	UART0_RTS	SPI1_MISO	SPI0_MISO
✓	✓		PA_23	UART0_TXD	SPI1_MOSI	SPI0_MOSI
✓			PB_1		SPI1_CLK	SPI0_SCK





Content

- Introduction to Ameba-Z
- SDK overview
- API of Components
- IDE Tool Demo
- MP Related

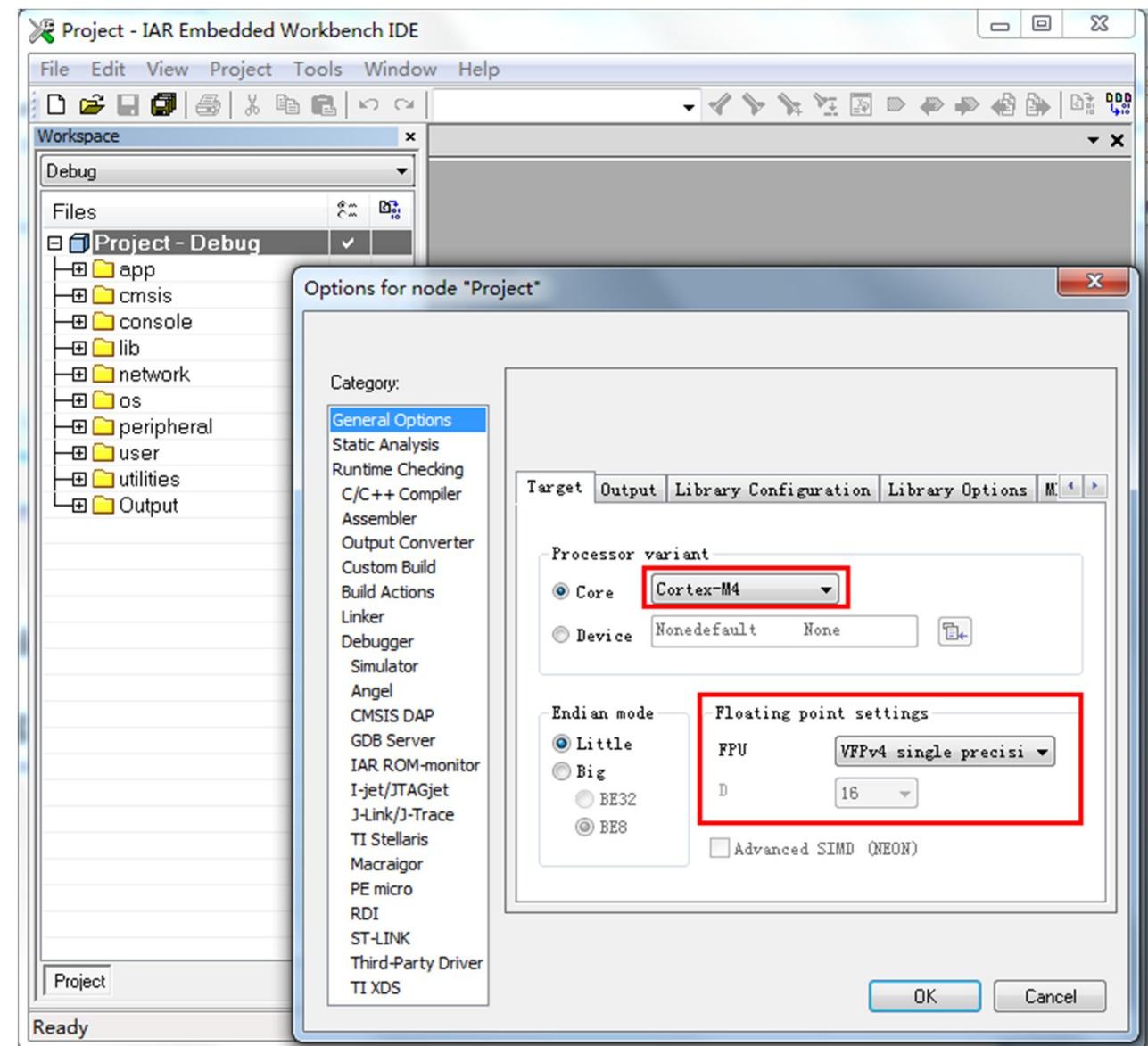
IAR build environment (Ref: UM0110)

■ IAR build code

- Open released IAR workspace
- Choose Cortex-M4F or Cortex-M4 at least
- Click Project -> Rebuild All

■ image2.icf

- Determines the location of sections on the Flash or SRAM
- Include external symbol of functions in ROM and Boot loader



IAR build

■ OTA2 binary

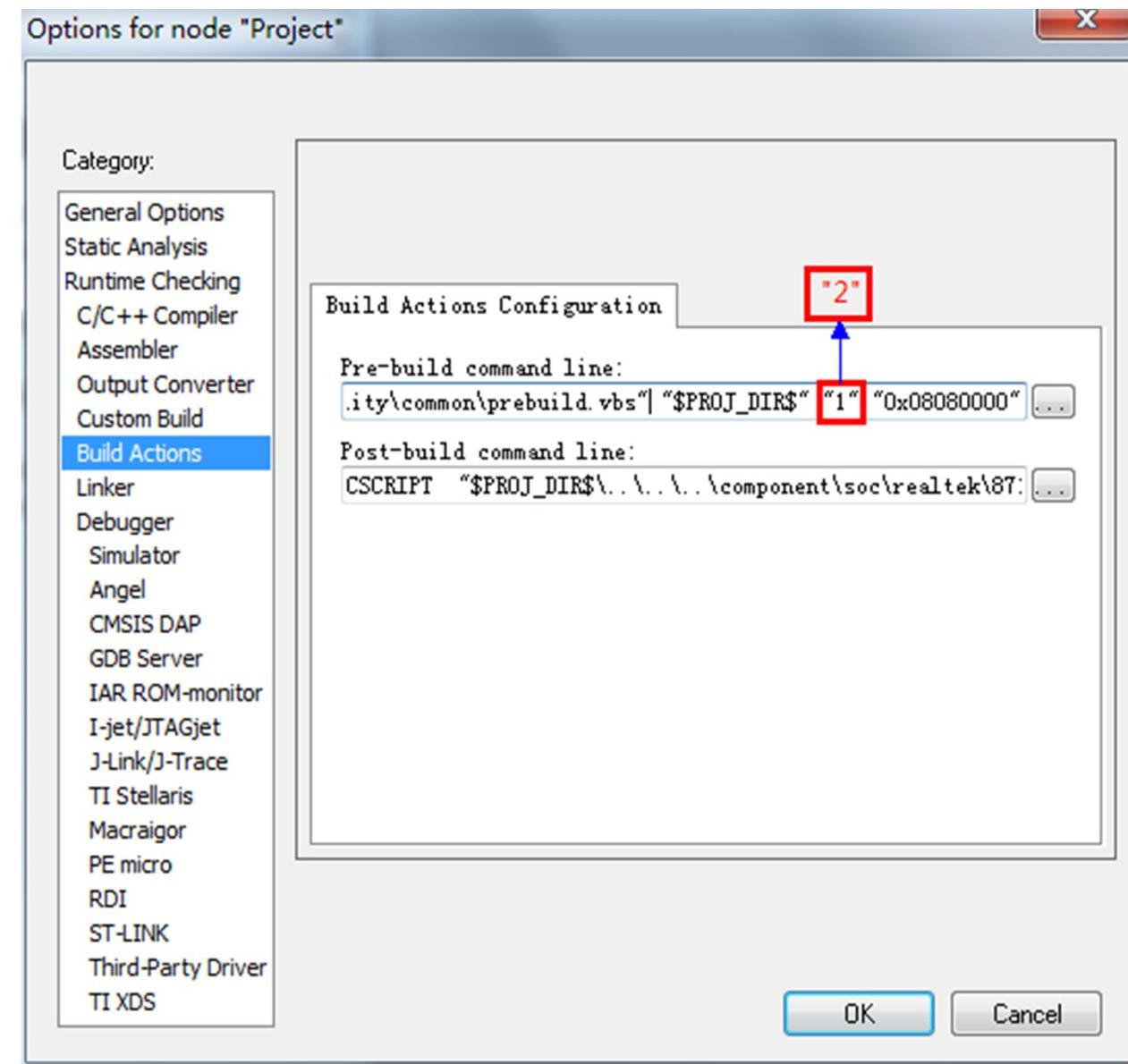
- Change the secondary input parameter of prebuild VB script from "1" to "2" to build

image2_all_ota2.bin
and generate
ota_all.bin

- Please Rebuild All

■ OTA2 flash offset

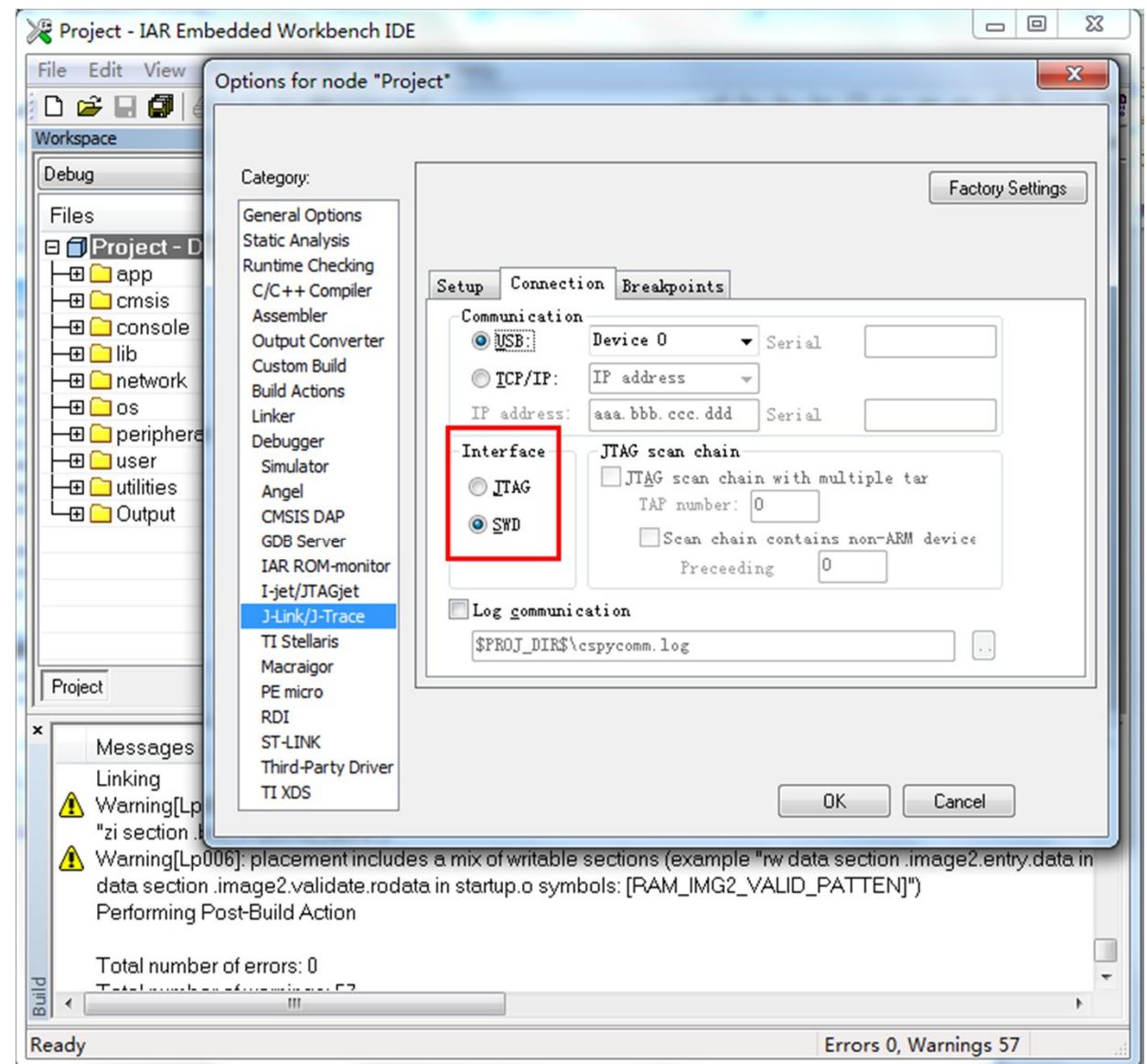
- The third input parameter of prebuild VB script is OTA2 flash offset, default set to 0x08080000





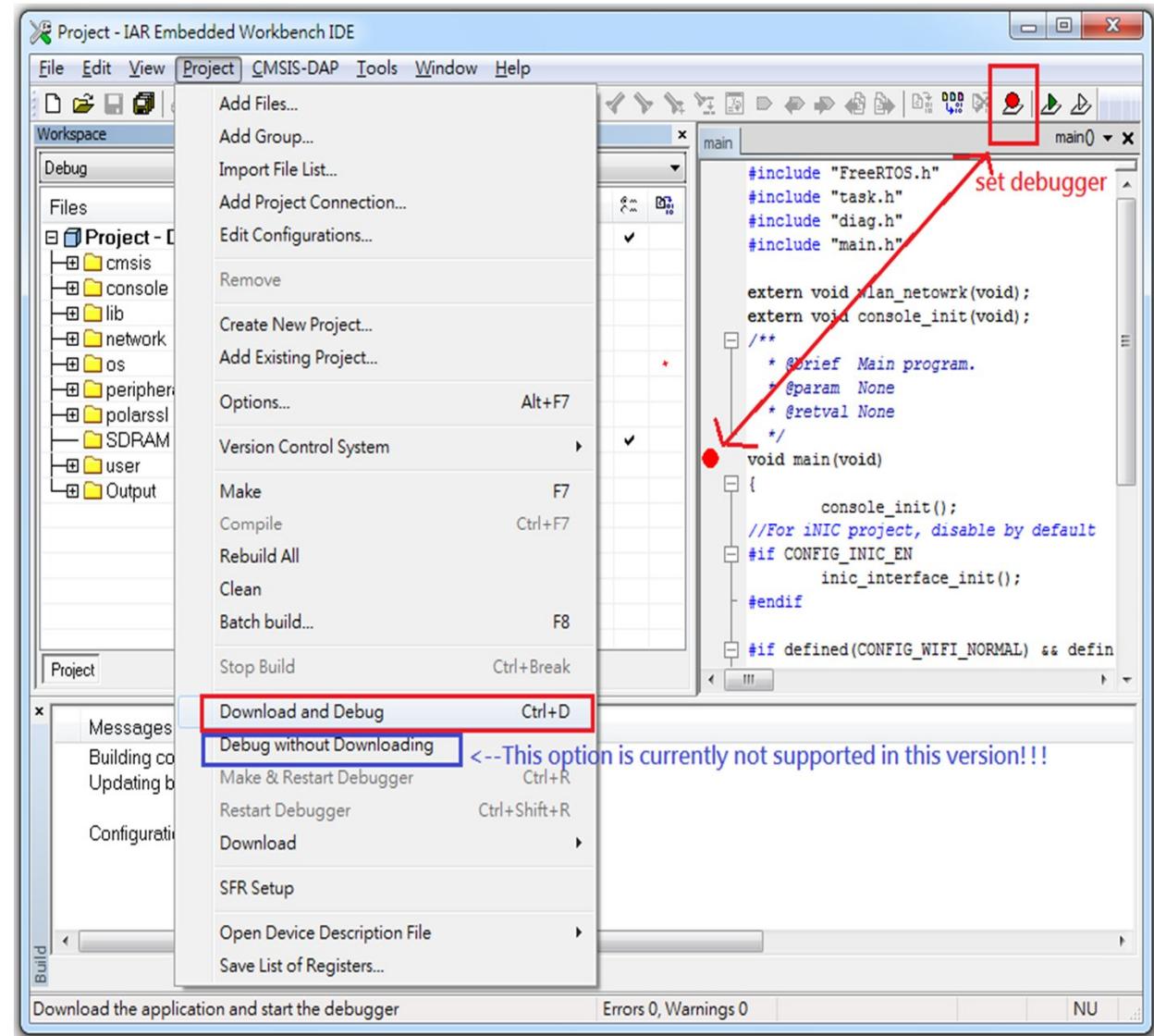
IAR Download

- Debugger
 - J-Link/J Trace
- J-Link Interface
 - SWD
- .board file
 - Decide what to write in output file and write to what position in Flash
 - tmp.board
 - Dynamically generated by postbuild VB script



IAR Debug

- Click Project->Download and Debug
- Set breakpoints
- View
 - Registers
 - Watch
 - Stack
 - Memory



Trouble shooting

- Project build fail
 - Check Core is set to CM4F or CM4.
 - Check if Flash and SRAM is enough.
- Download fail
 - Check SWD physical connection
 - Check the interface of J-Link settings is SWD
- UART log fail
 - Check jumper of J28
 - Check baud rate
- WLAN connect fail
 - Check log for connection status
 - Check security correctness
 - Check sniffer log

Trouble shooting – Hardfault

1. Record register PC and LR
2. Open application.asm and find out PC and LR in which function
3. The system crash caused by the instructions near PC
4. Check the PC's function and LR's function
5. If PC is not a valid memory address, may be memory overflow or catch flash instructions error
6. If PC is pointed to ROM or boot loader address, check the .icf symbol file to find the function.

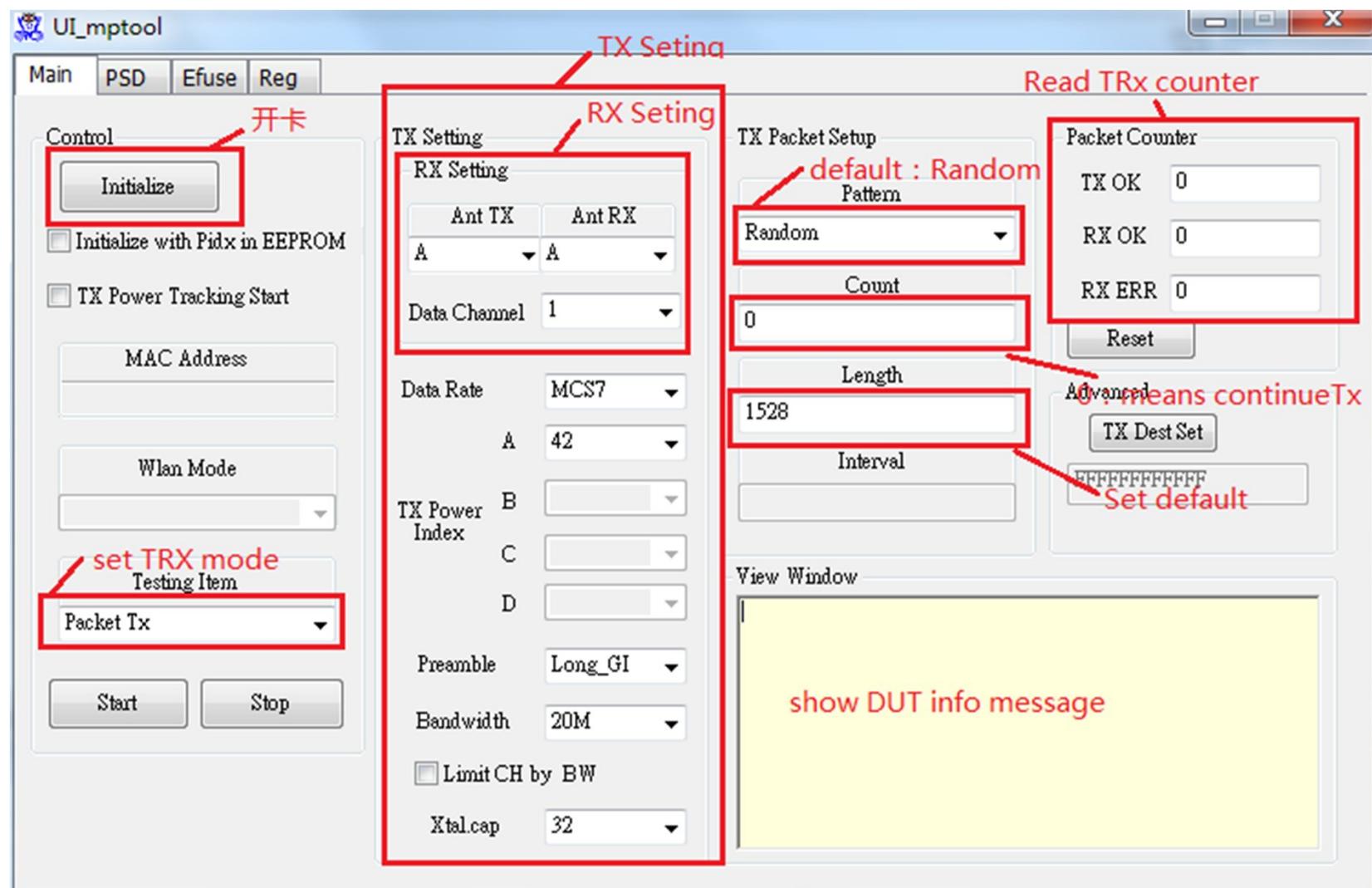
```
RTL8195A[HAL]: R0 = 0x0
RTL8195A[HAL]: R1 = 0x1005a37a
RTL8195A[HAL]: R2 = 0x6
RTL8195A[HAL]: R3 = 0x0
RTL8195A[HAL]: R12 = 0x1005f4a5
RTL8195A[HAL]: LR = 0x1004d0bf
RTL8195A[HAL]: PC = 0x1001ce00
RTL8195A[HAL]: PSR = 0x60000000
RTL8195A[HAL]: BFAR = 0x8
RTL8195A[HAL]: CFSR = 0x20000
RTL8195A[HAL]: HFSR = 0x40000000
RTL8195A[HAL]: DFSR = 0x0
RTL8195A[HAL]: AFSR = 0x0
RTL8195A[HAL]: PriMask 0x0
RTL8195A[HAL]: BasePri 0x0
RTL8195A[HAL]: SVC priority: 0x00
RTL8195A[HAL]: PendSVC priority: 0xf0
```

Content

- Introduction to Ameba-Z
- SDK overview
- API of Components
- IDE Tool Demo
- MP Related

MP tool – RF calibration (Ref: UM0119)

Realtek Ameba MP Flow (Ref: UM0059)





MP tool – EFUSE (Ref: UM0119)

UI_mptool

Main PSD Efuse Reg Mapfile

You must update the right data to the left mapfile before "Update" and "Program"!!

ID: 8195
MAC Address: FFFFFFFFFFFFFF
ThermalMeter: 1A
Channel plan: 20
XtalCapacity: 20

TX Power Index

	CCK	BW40
Ch 1.2	45	45
Ch 3.4.5	45	45
Ch 6.7.8	45	45
Ch 9.10.11	45	45
Ch 12.13	45	45
Ch 14	45	45

0x0~0x7 --> diff=0~7
0x8~0xF --> diff=-8~-1

Difference BW20 and BW40: 0
Difference OFDM and BW40: 2

Read IC Efuse data

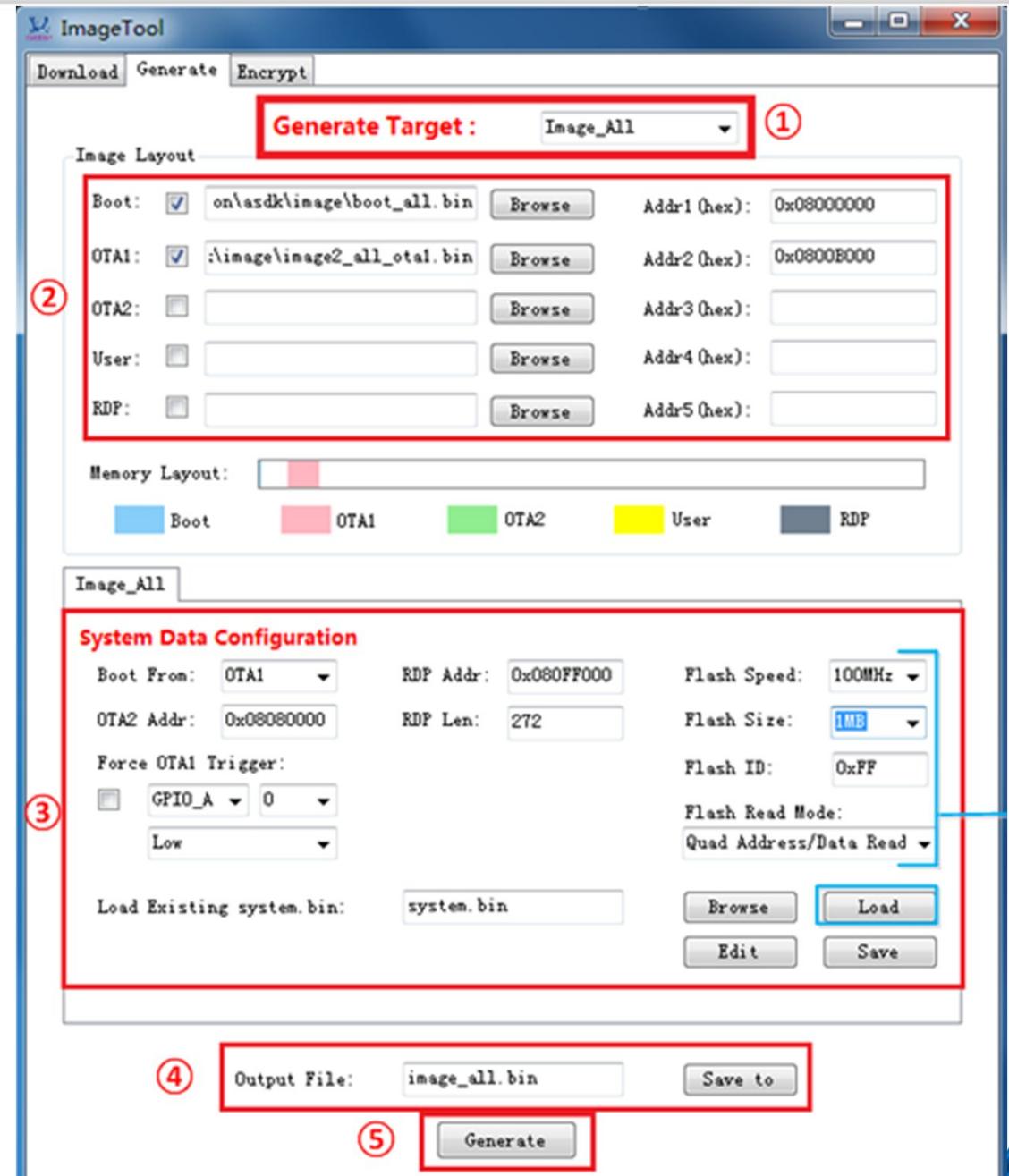
operate base on the mapfile data

Read mapfile Save mapfile Read from/Save to local

Read Update Program

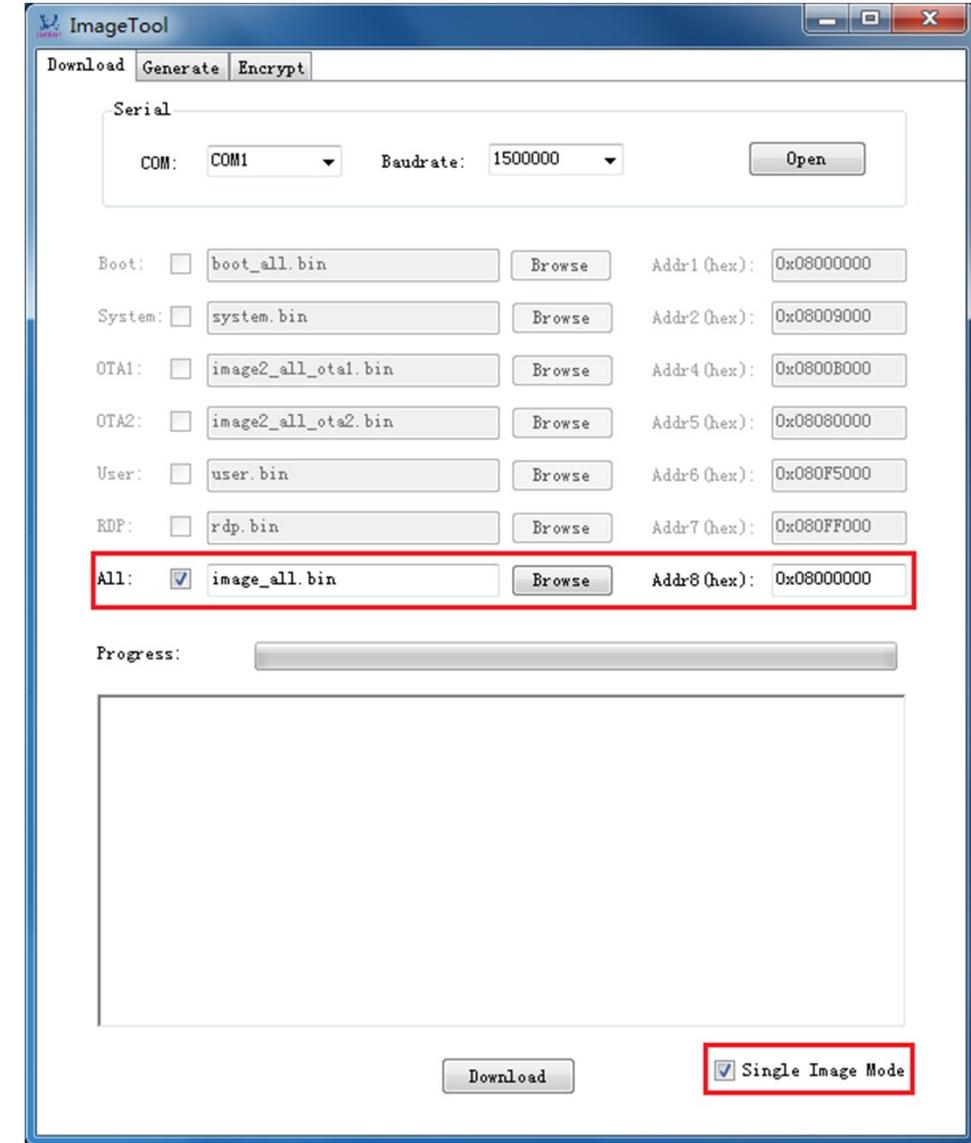
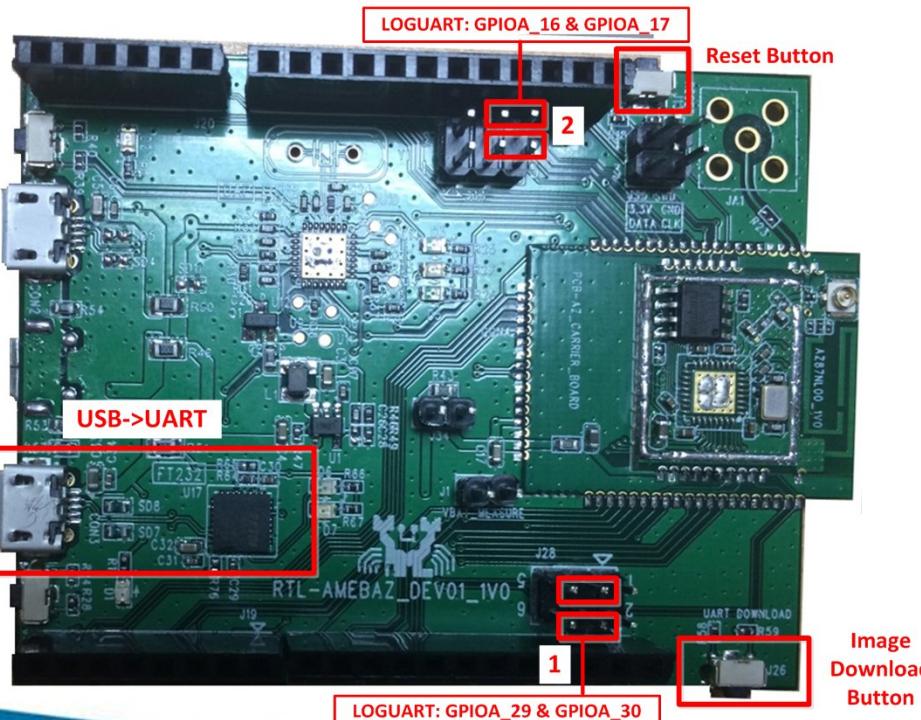
Image Generation (Ref: AN0112)

- Prepare binary files
 - boot_all.bin
 - image2_all_ota1.bin
 - image2_all_ota2.bin
- Config System Data
 - Boot From OTA1 or OTA2
 - OTA2 Address must be the same as Addr3 in Image Layout
 - Force OTA1 Trigger
 - RDP Addr and Len
 - Flash Parameters
 - Configure or Load system.bin
- Output file
 - image_all.bin



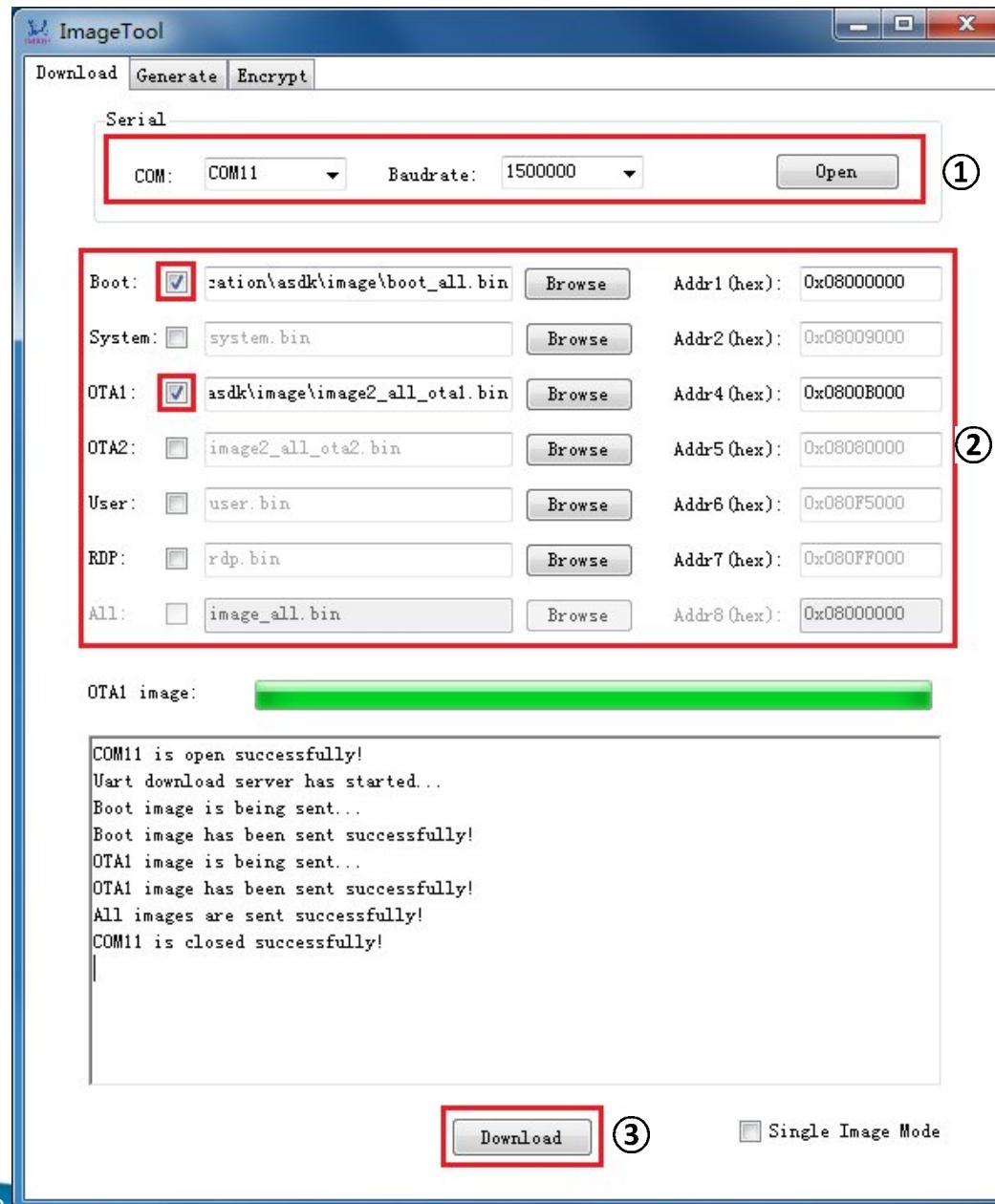
Flash Downloader for Factory (Ref: AN0112)

- Demo board: Enter download mode
 - Push the Image Download Button and keep it pressed
 - Power on the board or press the Reset Button
 - Release the Image Download Button
- Image tool
 - Choose COM and click Open
 - Choose image_all.bin
 - Click Download





Flash Downloader for RD (Ref: AN0112)



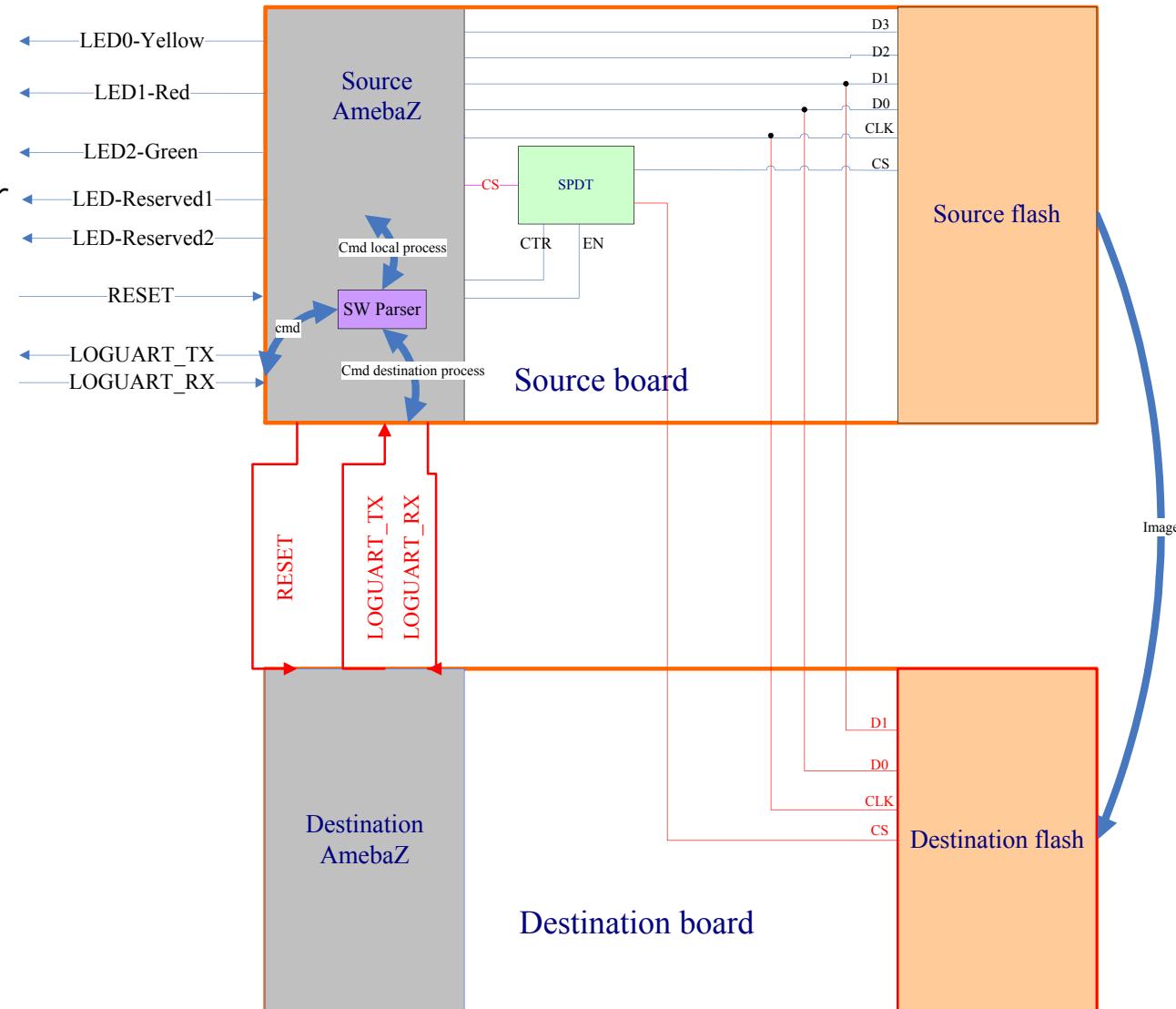
Flash To Flash Downloader (Ref: AN0115)

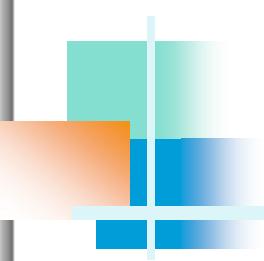
Source board

- On-board-flash download with **DEST CPU OFF**
- Control **SPDT** to select SRC or DEST flash use GPIO
- Control **reset pin** of DEST CPU, low for downloader, high for MP test through **log UART** of DEST Ameba-Z
- Flash loader + SW Parser + DEST flash image

LED

- Yellow: Power on when flash programing
- Red: Power on when flash program fail
- Green: Power on when flash program success





Thank you!