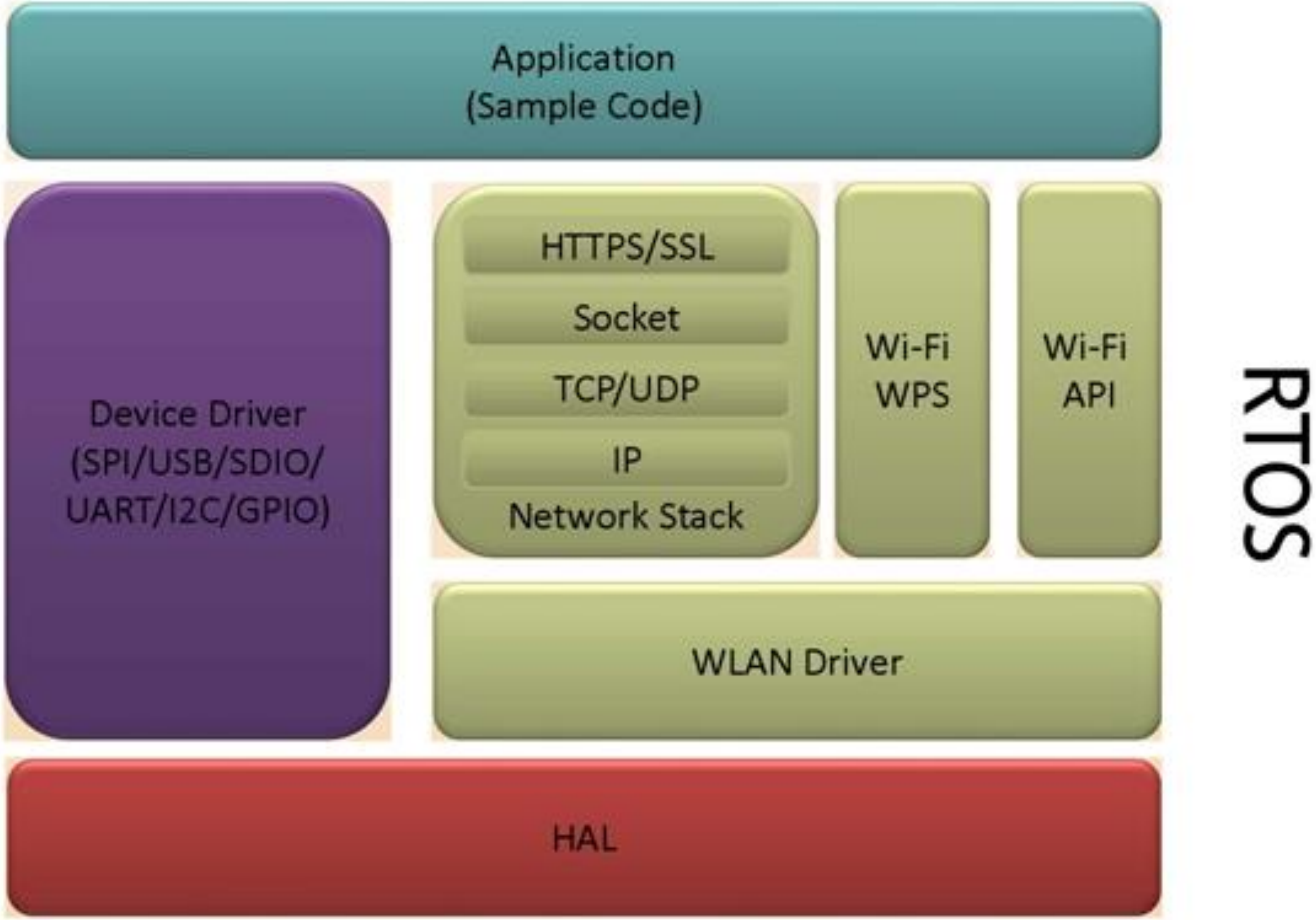
Introduction to Ameba SDK

2015.Q1

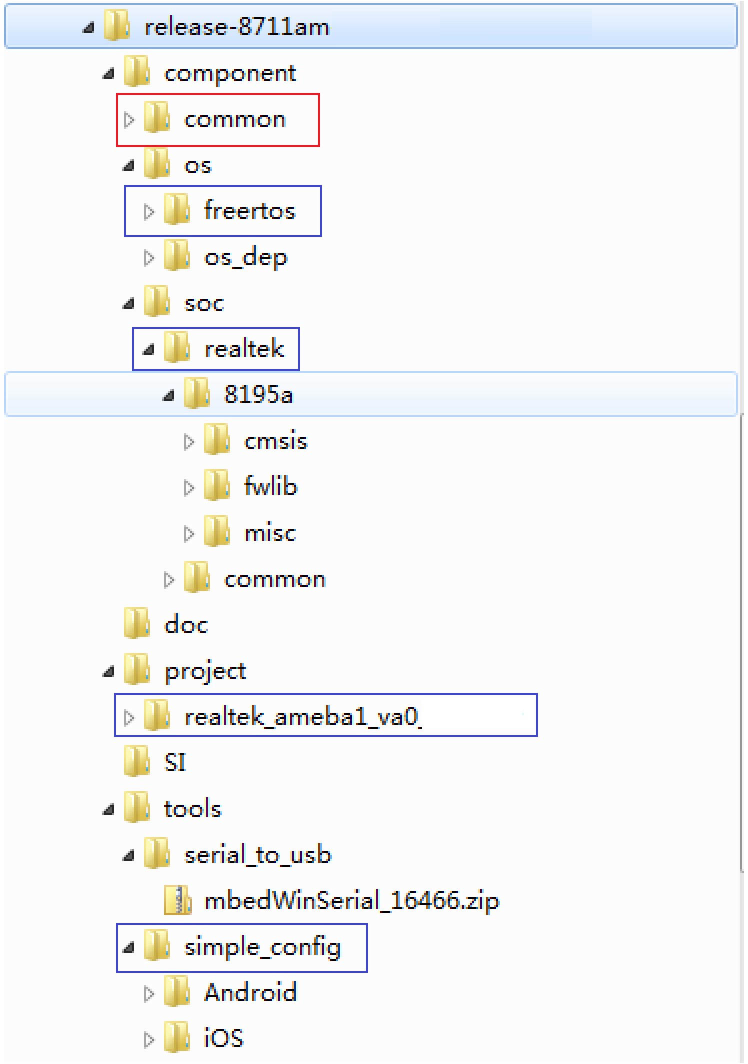
Content

- **Introduction to SDK**
- Network Stack and OS
- API of Components
- IDE Tool Demo

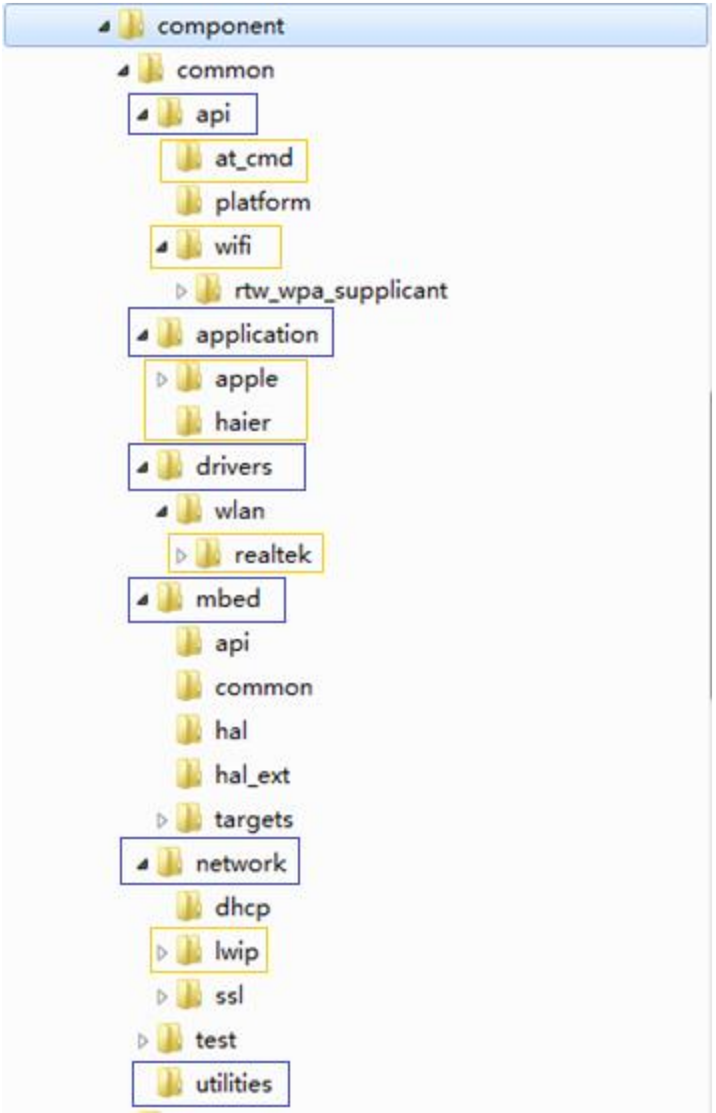
SDK Software Stack



Directory Structure



```
release-8711am
  component
    common
    os
      freertos
      os_dep
    soc
      realtek
        8195a
          cmsis
          fwlib
          misc
        common
  doc
  project
    realtek_ameba1_va0_
  SI
  tools
    serial_to_usb
      mbedWinSerial_16466.zip
    simple_config
      Android
      iOS
```



```
component
  common
    api
      at_cmd
      platform
      wifi
        rtw_wpa_supplicant
    application
      apple
      haier
    drivers
      wlan
        realtek
    mbed
      api
      common
      hal
      hal_ext
      targets
    network
      dhcp
      lwip
      ssl
    test
    utilities
```

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
- Device Simple Config
  - SoftAP mode config
  - WPS
  - Realtek simple config
  - Customizable Promiscuous Mode
- Secure Sockets Layer
  - Polar SSL (Ref: AN0012)
- OTA update (Ref: AN0033)

Simple Config (Ref: AN0011)

- How to get IoT device link to AP
  - AP mode -> STA mode
    - Most reliable but more complicated
    - User experience is more complicated for iPhone user
  - WPS
    - Easy
    - Has more interoperability issue, but user may have enough WPS experience
  - Simple Connection
    - Easiest way
    - Realtek provide Android/iPhone API
    - Average configure time less than 10 seconds
  - Customizable Promiscuous Mode
    - Design individual algorithm

Device Discovery

- mDNS (ref:AN0043)

Cloud Access

- Google Nest Cloud API (ref:AN0038)

Ameba Memory Layout (Ref: UM0034)

Feature	RTL8195AM	RTL8711AM	RTL8711AF
Package	TFBGA98	QFN56	QFN48
Package Dimension	6x6mm	7x7mm	6x6mm
CPU	ARM Cortex M3 166MHz		
ROM	1MB	1MB	1MB
Flash	selectable	selectable	1MB
RAM	2MB + 512KB	2MB + 512KB	512KB

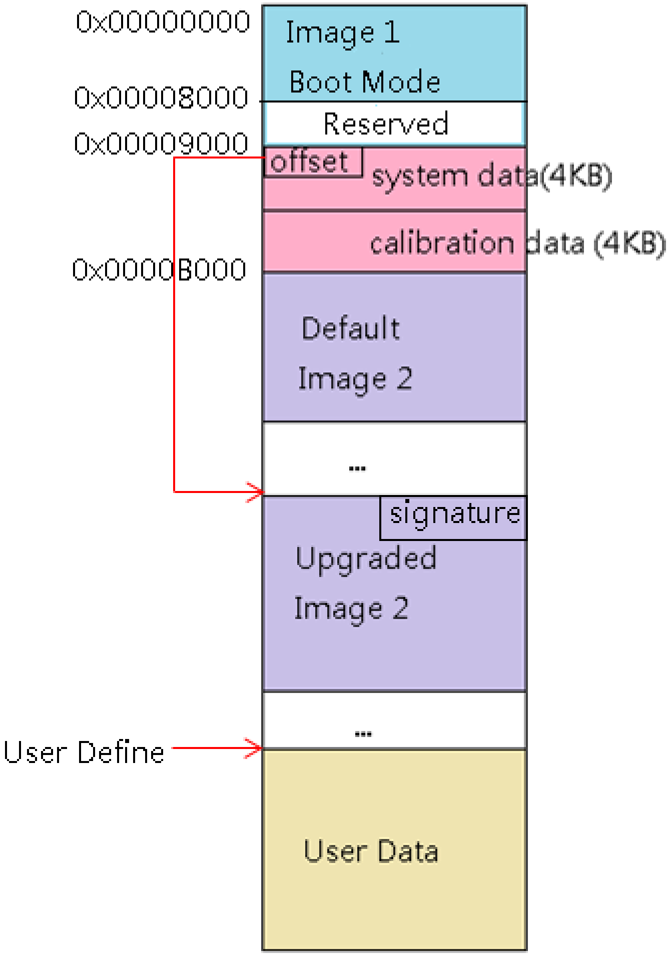
Ameba Flash Layout (Ref: UM0034)

- Bootloader
  - Hardware initialization
  - Image 2 loading

	0	4	8	10	16
0x10	Signature				
0x20	Size	Address	Image 2 Addr	Dummy	
	Firmware Code				

- Upgraded Image 2

	0	4	8	16
0x10	Size	Address	signature	
	Firmware Code			



Ameba Crypto Engine (UM0027)

- Polar SSL can be used with crypto engine.
- Crypto engine is the hardware which can help CPU to do the encryption, decryption and authentication.
- Authentication
  - Md5
  - Sha1
  - Sha2
  - suggests keep using software authentication
- Encryption and Decryption
  - AES (cbc, ecb, ctr)
  - DES (cbc, ecb)
  - 3DES (cbc, ecb).

Content

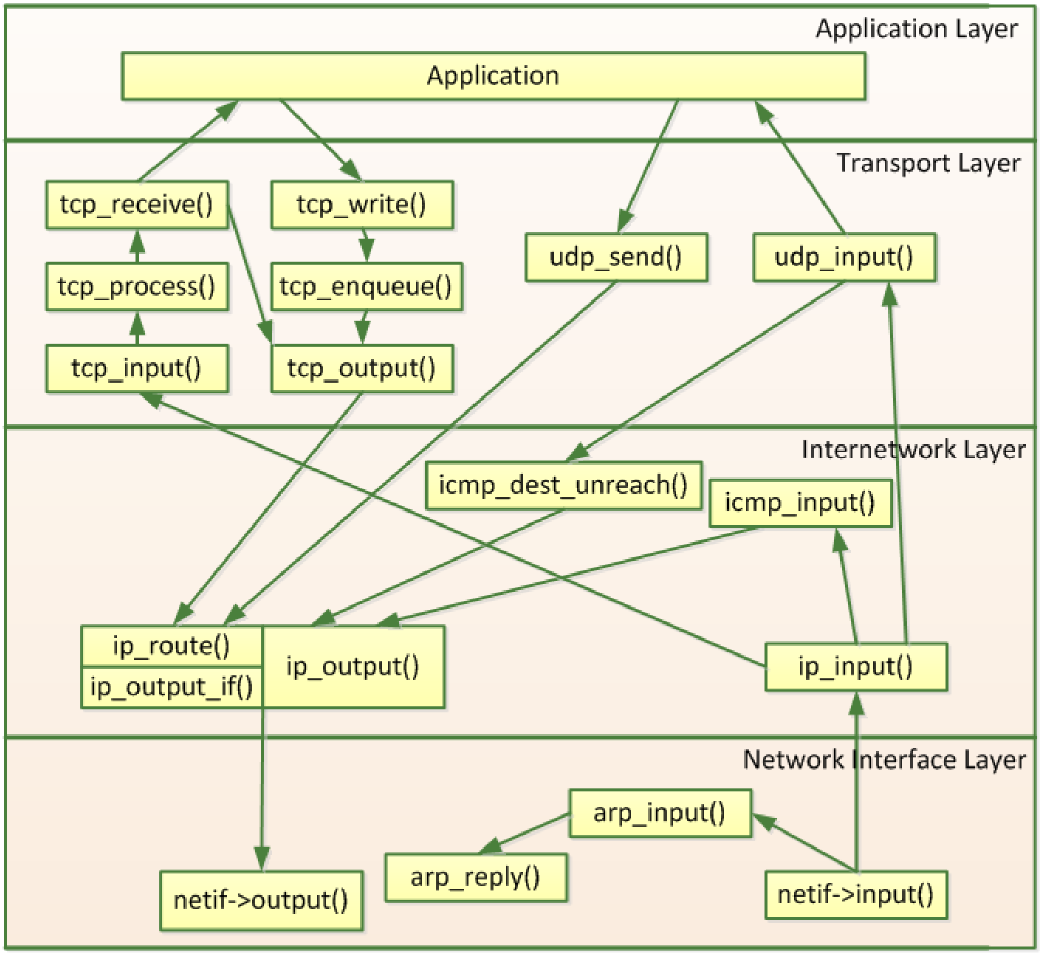
- Introduction to Ameba SDK
- **Network Stack and OS**
- API of Components
- IDE Tool Demo

Introduction to 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

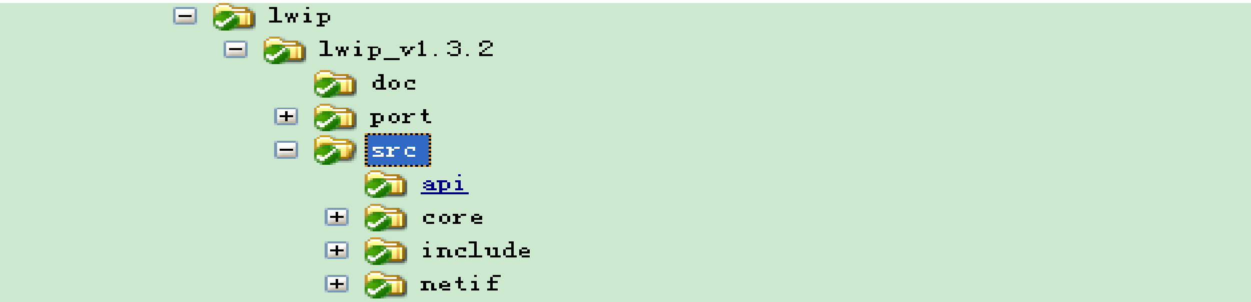
LWIP

- Implemented based on 4 layer TCP/IP Model
- Design with scalability, ARP/IP/ICMP/UDP/TCP /OS API/Memory Management/Socket APIs are supported
- Implement the communications between protocols by memory share



LWIP

- Source Code Directory
  - Port: Adapt different platform
  - Api: BSD and RAW API
  - Core: Implementation of ICMP/IP/UDP/TCP etc
  - Include: header files
  - Netif: Template of ARP and LwIP net device drivers



```
⊟ lwip
  ⊟ lwip_v1.3.2
      doc
    ⊞ port
    ⊟ src
        api
      ⊞ core
      ⊞ include
      ⊞ netif
```

- Reference
  - lwIP Official Website: http://www.nongnu.org/lwip/
  - lwIP Official Documentation: http://www.nongnu.org/lwip/main.html

Introduction to Freertos

- Is known to be reliable.
- Is undergoing continuous active development.
- Has a minimal ROM, 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 3 C files.
- Is very scalable, simple and easy to use.
- Is well established with a large and ever growing user base.
- FreeRTOS offers a smaller and easier real time processing alternative for applications.

Freertos

- C Files
  - Tasks.c
  - Queue.c
  - Heap_4.c / Heap_5.c
  - Timer.c
- Reference
  - http://www.FreeRTOS.org - Documentation, books, training, latest versions, license and Real Time Engineers Ltd.

Content

- Introduction to Ameba SDK
- Network Stack and OS
- **API of Components**
- IDE Tool Demo

AT Command (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

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

- ‘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

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_register_multicast_address

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

- 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

Mbed peripheral API

- Flash
  - flash_init
  - flash_lock
  - flash_unlock
  - flash_write_protect
  - flash_erase_sector
  - flash_read_word
  - flash_write_word
  - flash_stream_read
  - flash_stream_write
- GPIO
  - gpio_init
  - gpio_set
  - gpio_mode
  - gpio_dir
  - gpio_write
  - gpio_read
- I2C
  - i2c_init
  - i2c_frequency
  - i2c_start
  - i2c_stop
  - i2c_read
  - i2c_write
  - i2c_byte_read
  - i2c_byte_write
  - i2c_reset
  - i2c_slave_address
  - i2c_slave_mode
  - i2c_slave_receive
  - i2c_slave_read
  - i2c_slave_write

Mbed peripheral API

- Serial
  - serial_init
  - serial_free
  - serial_baud
  - serial_format
  - serial_irq_handler
  - serial_irq_set
  - serial_getc
  - serial_putc
  - serial_readable
  - serial_writable
  - serial_clear
  - serial_pinout_tx
  - serial_break_set
  - serial_break_clear

- SPI
  - spi_init
  - spi_free
  - spi_format
  - spi_frequency
  - spi_master_write
  - spi_slave_receive
  - spi_slave_read
  - spi_slave_write
  - spi_busy
  - spi_slave_receive_interrupt
  - spi_master_write_interrupt

LWIP API

- Socket
- Shutdown
- Bind
- Listen
- Accept
- Connect
- Recv
- Recvfrom
- Send
- Sendto
- Select
- Ioctlsocket
- Read
- Write
- Close

$sdk\component\common\network\lwip\lwip_v1.3.2\src\api\sockets.c

- 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

Freertos API

- RtlZmalloc
- RtlMalloc
- RtlMfree

- RtlEnterCritical
- RtlExitCritical
- RtlInitSema
- RtlFreeSema
- RtlUpSema
- RtlUpSemaFromISR
- RtlDownSema
- RtlDownSemaWithTimeout

- RtlSystime2Ms
- RtlMs2Systime

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

- RtlMsleepOS
- RtlUsleepOS
- RtlMdelayOS
- RtlUdelayOS

- RTL_ATOMIC_SET
- RTL_ATOMIC_READ
- RTL_ATOMIC_ADD
- RTL_ATOMIC_SUB
- RTL_ATOMIC_INC
- RTL_ATOMIC_DEC

- RtlTimerCreate
- RtlTimerDelete
- RtlTimerStart
- RtlTimerStop
- RtlTimerReset
- RtlTimerChangePeriod

Development Guideline

- Develop cross-platform api in common\api
- Develop driver (ex, sensor driver) in common\driver
- Develop application code in common\application
- Develop general network stack in common\network
- Keep platform dependent project as simple as possible

Content

- Introduction to Ameba SDK
- Network Stack and OS
- API of Components
- **IDE Introduction**

IDE Tool Introduction (Ref: UM0023)

- IDE Tool
  - IAR
- Get Started
  - Build code
  - Load code
  - Debug

Getting Start (Ref: AN0025)

- Check AP setting
- Device run in interactive mode
- Enter command to connect with AP
  - Use AT command
- Enter command to show wifi info
  - Use AT Command
- Ping *.*.*.*

EVB Board

- 8195AM 3V0 Evaluation Board (ref UM0048)
- 8711AM 2V0
  - 8195AM 3V0 is mainstream EVB. It is suggested to use 8195AM 3V0

MP related documentation

- Wi-Fi MP command (ref:AN0004)
  - Wi-Fi RF performance evaluation
  - Command and Operation for wi-fi related mass production

- Calibration data specification (ref:AN0057)
  - Specification for *system* and *wi-fi* board level parameter and calibration data.

- System Mass Production (ref: AN0058)
  - System level mass production flow introduction
  - Command for system level mass production

Trouble shooting

- Project build fail
  - Check IAR version is higher than 7.20.
- Uart log fail
  - Check Pin assignment
  - Check baud rate
- WLAN connect fail
  - Check log for connection status
  - Check security correctness
  - Check sniffer log

Thank you!