



# Introduction to Ameba SDK

2015.Q1

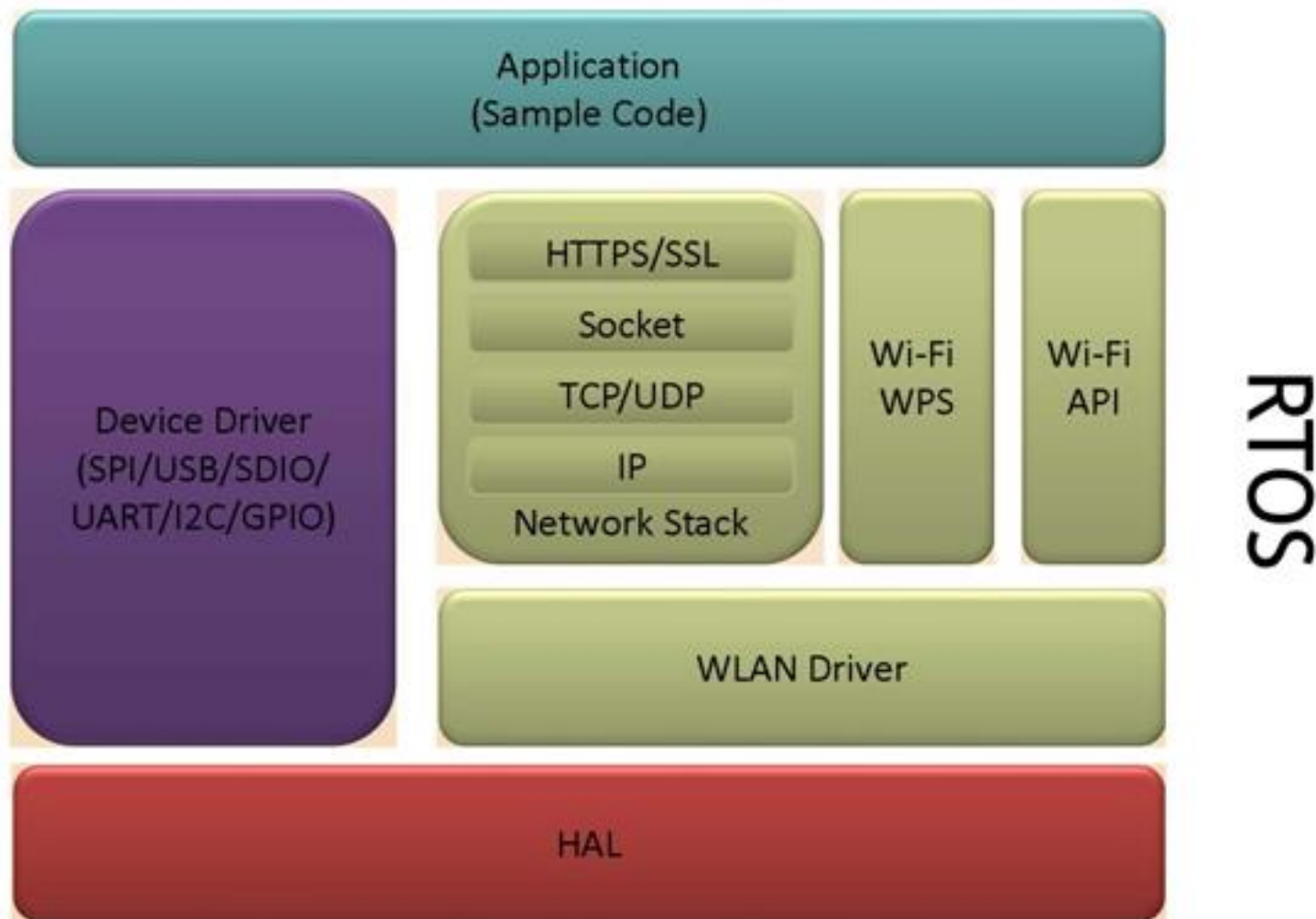


# Content

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

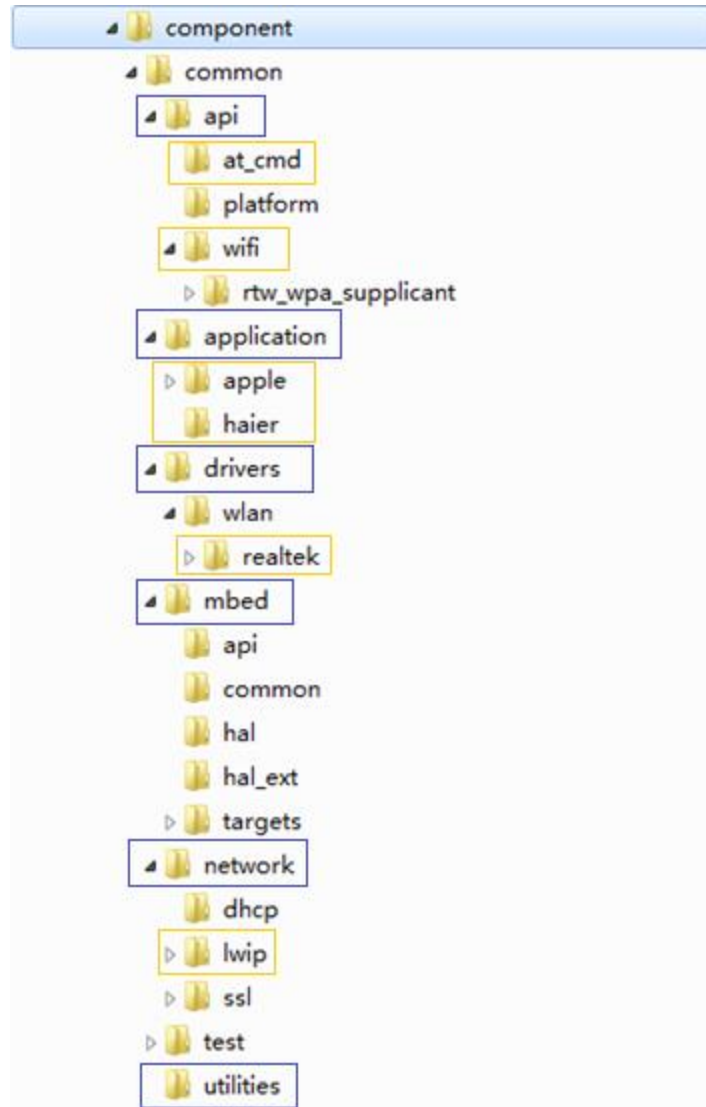
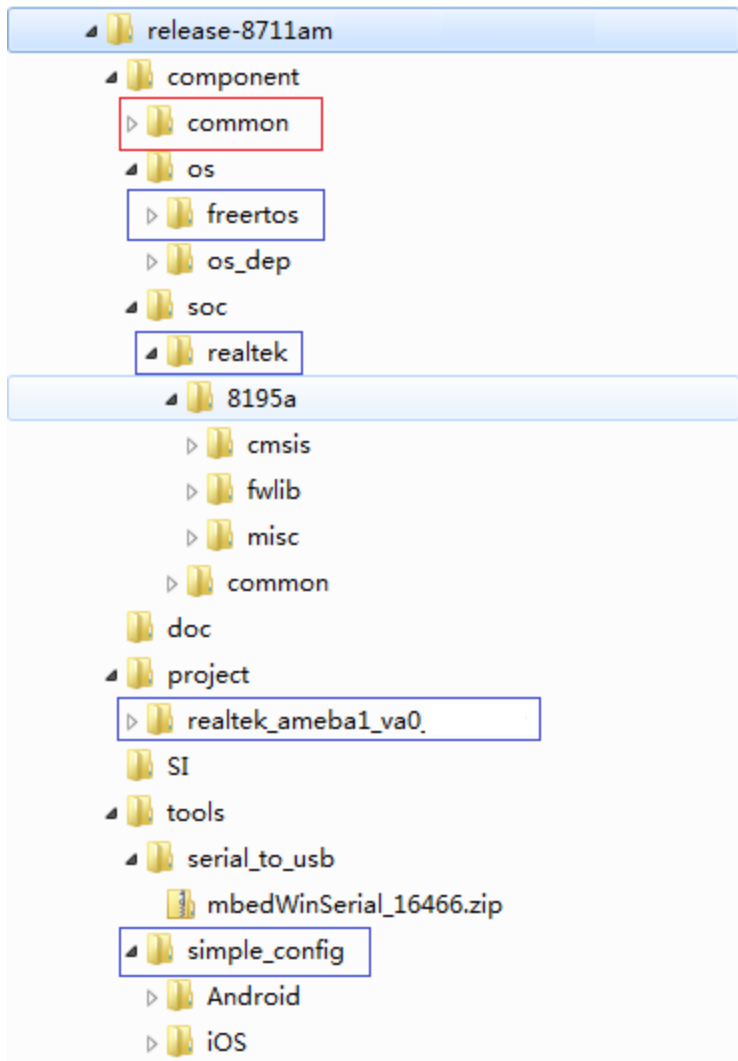


# SDK Software Stack





# Directory Structure





# 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



# Ameba Memory Layout (Ref: UM0034)

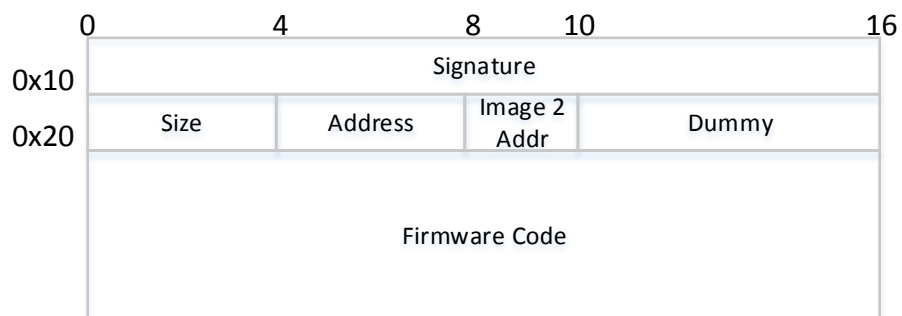
Feature	RTL8195AM	RTL8711AM	RTL8711AF
Package	TFBGA98	QFN56	QFN48
Package Dimension	6x6mm	7x7mm	6x6mm
CPU	ARM Cortex M3 166MHz		
ROM	1MB	1MB	1MB
Flash	selectable	selectable	1MB
RAM	2MB + 512KB	2MB + 512KB	512KB



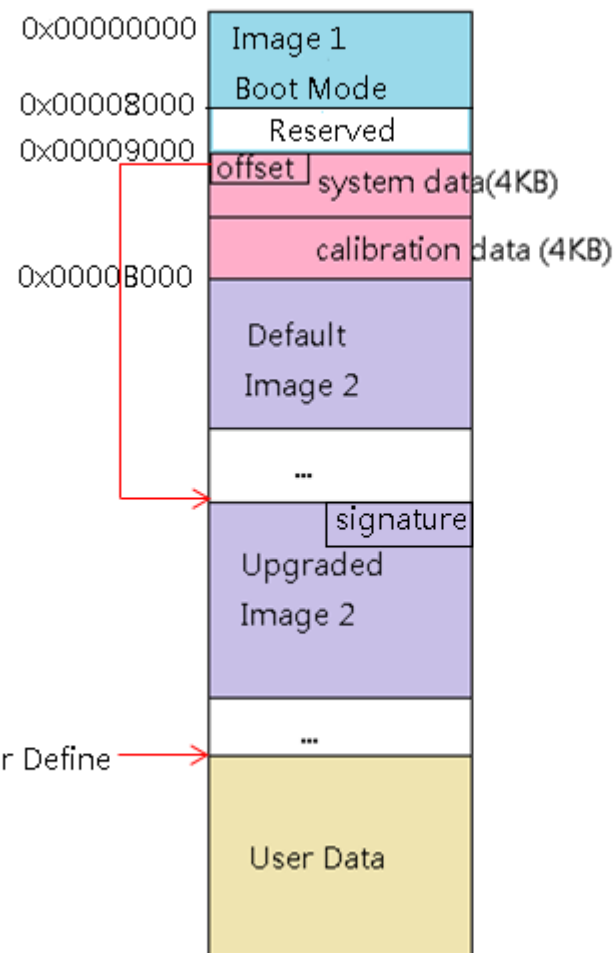
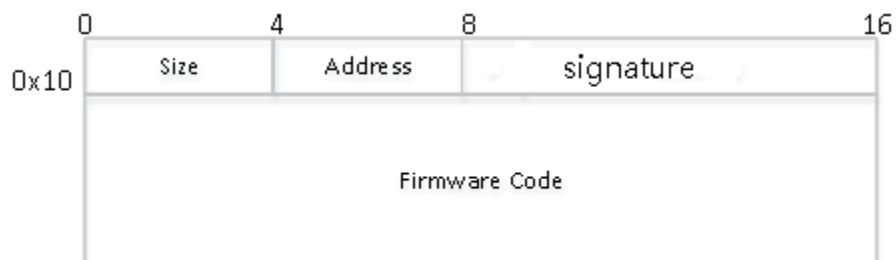
# Ameba Flash Layout (Ref: UM0034)

## ■ Bootloader

- Hardware initialization
- Image 2 loading



## ■ Upgraded Image 2







# 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



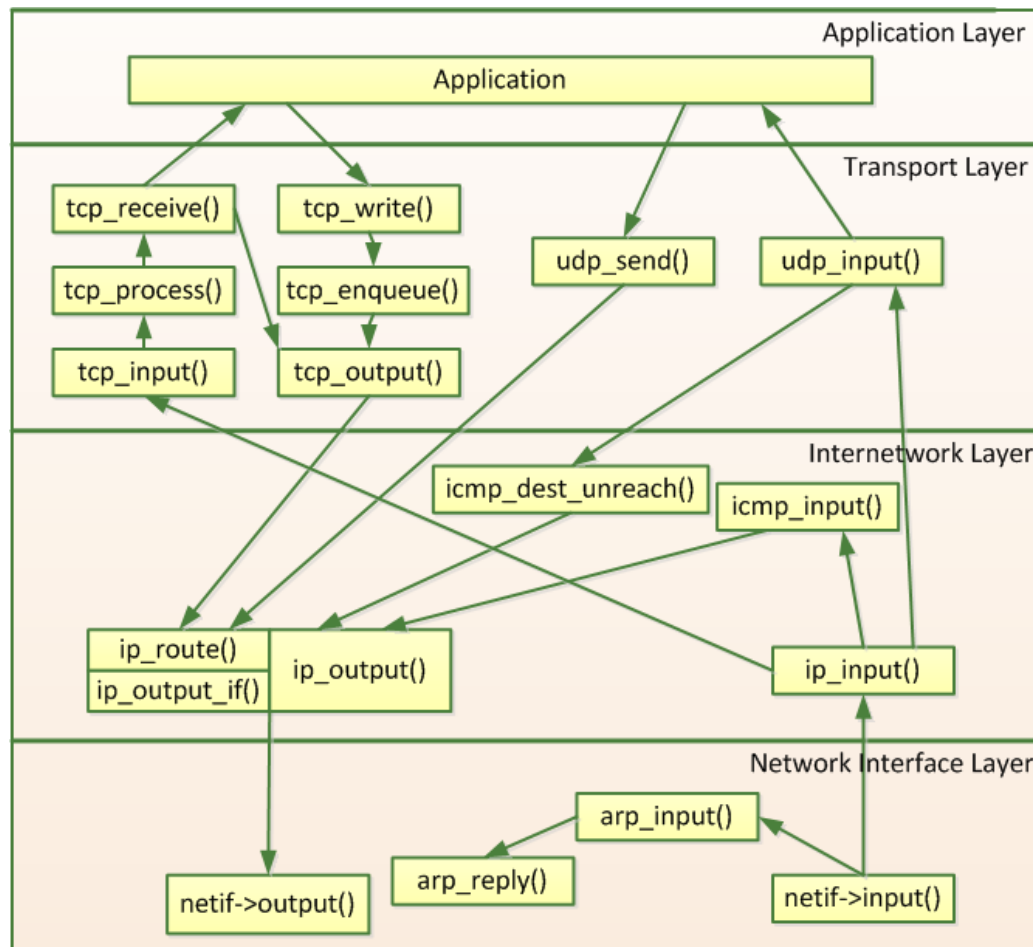
# Introduction to LWIP

- 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

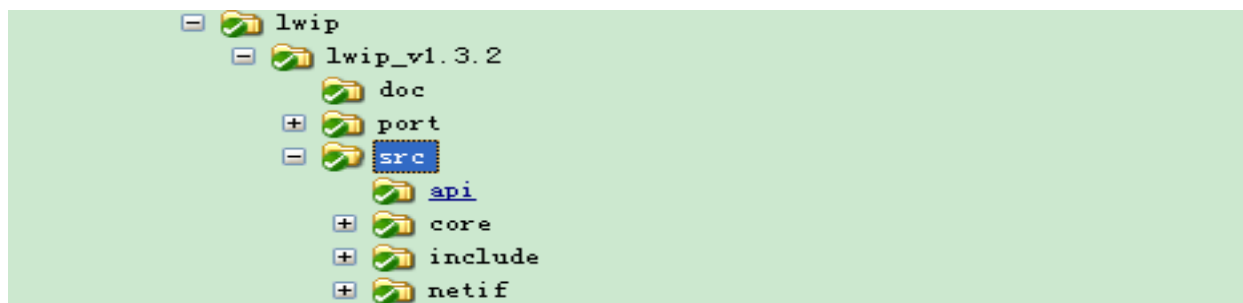




# LWIP

## ■ 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



## ■ 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
- '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`



# 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`



# 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
- 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_  
v1.3.2\src\api\sockets.c
```



# 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.



# Cloud Access

- Google Nest Cloud API (ref:AN0038)



# 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!**