WM_W800_SDK User Manual

V1.1

Beijing Lianshengde Microelectronics Co., Ltd. (winner micro)

Address: Floor 18, Yindu Building, No. 67, Fucheng Road, Haidian District, Beijing

Tel: +86-10-62161900

Company website: www.winnermicro.com

Document modification record

Version	revision time	revision history	author	review
V0.1	2019/9/25	[C] Create document	Cuiyc	
V0.2	2020/7/8	unified font	Cuiyc	
V1.0	2020/8/10	Upgrade version number	Cuiyc	
V1.1	2020/11/5	update digital interface description	Ray	

Table of contents

1 **Introduction**

1.1 Overview

This article mainly describes the function and usage of W800 Software Development Kit (SDK), which integrates W800 hardware driver (BSP),

Real-time operating system, TCP/IP protocol stack, WiFi protocol stack, BT protocol stack and other public modules can satisfy most application software

demand.

1.2 Chip Introduction

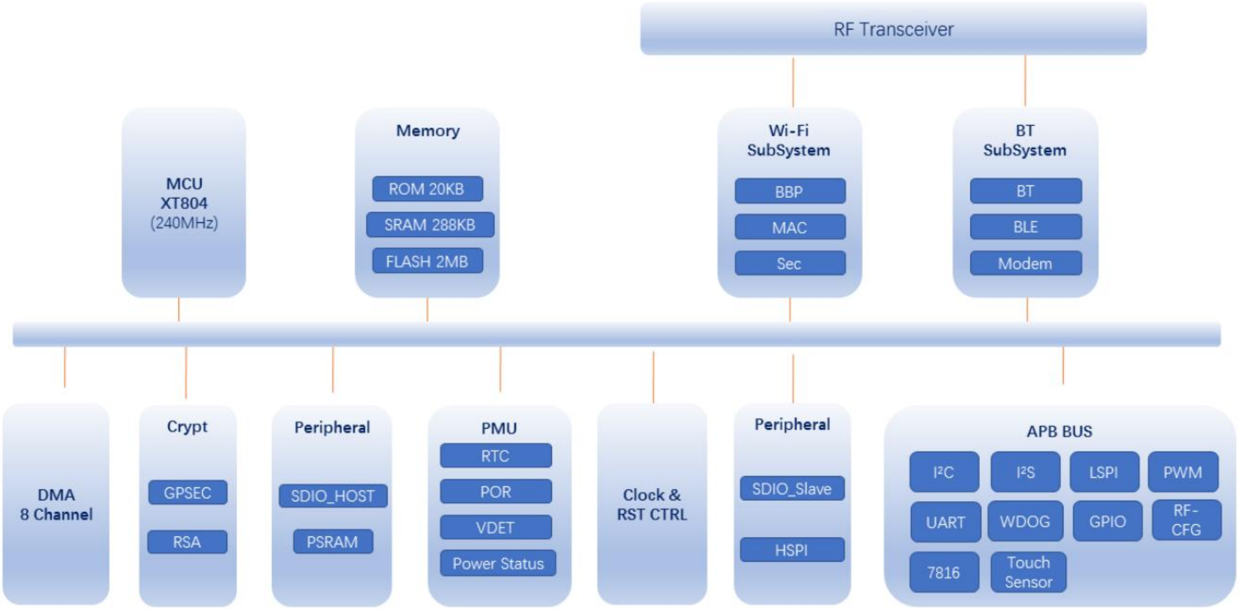


Figure 1 W800 Chip Architecture

ÿ Chip Appearance

ÿ QFN32 package, 4mm x 4mm

ÿ MCU Features

ÿ Integrated 32-bit XT804 processor, operating frequency 240MHz, built-in DSP, floating point unit and security engine

ÿ Built-in 2MB Flash, 288KB RAM

ÿ Integrated PSRAM interface, supports up to 64MB external PSRAM memory

ÿ Integrated 5-way UART high-speed interface

ÿ Integrated 2 channels of 16-bit ADC, the highest sampling rate is 1KHz

ÿ Integrates a high-speed SPI interface, supporting up to 50MHz

ÿ Integrate 1 SDIO_HOST interface, support SDIO2.0, SDHC, MMC4.2

ÿ Integrate 1 SDIO_DEVICE, support SDIO2.0, the highest throughput rate is 200Mbps ÿ Integrated 1 I 2C controller

ÿ Integrated GPIO controller, supports up to 18 GPIOs

ÿ Integrated 5-way PWM interface

ÿ Integrated 1-way Duplex I 2S controller

ÿ Integrated 11 Touch Sensors

ÿ Security Features

ÿ MCU built-in Tee security engine, the code can distinguish safe world/non-safe world

ÿ Integrated SASC/TIPC, memory and internal modules/interfaces can be configured with security attributes to prevent non-secure code access

ÿ Enable firmware signature mechanism to achieve secure Boot/upgrade

ÿ Equipped with firmware encryption function to enhance code security

ÿ Firmware encryption keys are distributed using asymmetric algorithms to enhance key security

ÿ Hardware encryption module: RC4256, AES128, DES/3DES, SHA1/MD5, CRC32, 2048 RSA, true random number generator

ÿ Wi-Fi characteristics

ÿ Support GB15629.11-2006, IEEE802.11 b/g/n

ÿ ÿÿ Wi-Fi WMM/WMM-PS/WPA/WPA2/WPS

ÿ Support EDCA channel access method

ÿ Support 20/40M bandwidth working mode

ÿ Support STBC, GreenField, Short-GI, support reverse transmission

ÿ Support AMPDU, AMSDU

- ÿ Support IEEE802.11n MCS 0~7, MCS32 physical layer transmission rate stalls, the maximum transmission rate is 150Mbps

- ÿ Support Short Preamble when sending at 2/5.5/11Mbps

- ÿ Support HT-immediate Compressed Block Ack, Normal Ack, No Ack response mode

- ÿ Support CTS to self

- ÿ Support Station, Soft-AP, Soft-AP/Station functions

ÿ Bluetooth Features

- ÿ Integrated Bluetooth baseband processor/protocol processor, supports BT/BLE dual-mode working mode, supports BT/BLE4.2 protocol

ÿ Power Management

- ÿ Single 3.3V power supply

- ÿ Support Wi-Fi power saving mode power management

- ÿ Support work, sleep, standby, shutdown working modes

- ÿ Standby power consumption is less than 15uA

1.3 Basic Features of SDK

W800 SDK is a collection of RTOS kernel, hardware driver, Wi-Fi protocol stack, Bluetooth protocol stack, TCP/IP protocol stack, network application protocol, An embedded Wi-Fi and Bluetooth application development platform integrating AT instruction set, multiple application layer protocols and corresponding sample codes.

The functions it provides are:

wireless

ÿ Support IEEE802.11b/g/n wireless standard

ÿ Support BT/BLE4.2

ÿ Supported frequency range: 2.412~2.484 GHz

ÿ Support basic network (Infra)

ÿ Support multiple encryption and authentication mechanisms: OPEN/WEP64/WEP128/TKIP/CCMP/WPA-PSK/WPA2-PSK

ÿ Support fast networking mode (designated channel and BSSID)

ÿ Support wireless roaming

ÿ Support PS-POLL energy-saving mode

ÿ Support WPS function

ÿ Support soft AP

ÿ Support OPEN, WEP, TKIP, AES encryption authentication

ÿ Support up to 8 station connections

ÿ Support STA's PS-POLL energy saving

ÿ Support soft APSTA

ÿ Support 2-level cascading

ÿ Support up to 8 station connections

drive

ÿ Support UART interface communication

ÿ Support UART data interface, the maximum rate of the interface is 2Mbps

ÿ Support user programmable GPIO control

ÿ Support HSPI interface

ÿ Support SDIO interface

ÿ Support external expansion PSRAM interface

ÿ Support I2S interface

ÿ Support I2C interface, the maximum rate is 400Kbps

ÿ Support 7816 interface

ÿ Support hardware encryption and decryption

ÿ Support plug-in SPI-FLASH

ÿ Support built-in FLASH

ÿ Support ADC interface

ÿ Support PWM interface

networking

ÿ Support different network distribution methods

ONESHOT

ÿ WPS

ÿ WEB page

ÿ AIRKISS

ÿ Bluetooth

ÿ Support API networking (for secondary developers)

ÿ Support AT command mode networking (for AT command developers)

TCP/IP protocol

ÿ Support multiple network protocols: TCP/UDP/ICMP/DHCP/DNS

ÿ Support DHCP Server, DNS Server

ÿ Support HTTP Client, HTTP Server function

ÿ Support IPERF

other

ÿ Support AT command set

ÿ Support standard socket interface

ÿ Support m-DNS

ÿ Support web socket

ÿ Support SSL Server, SSL Client

ÿ Support OS replacement

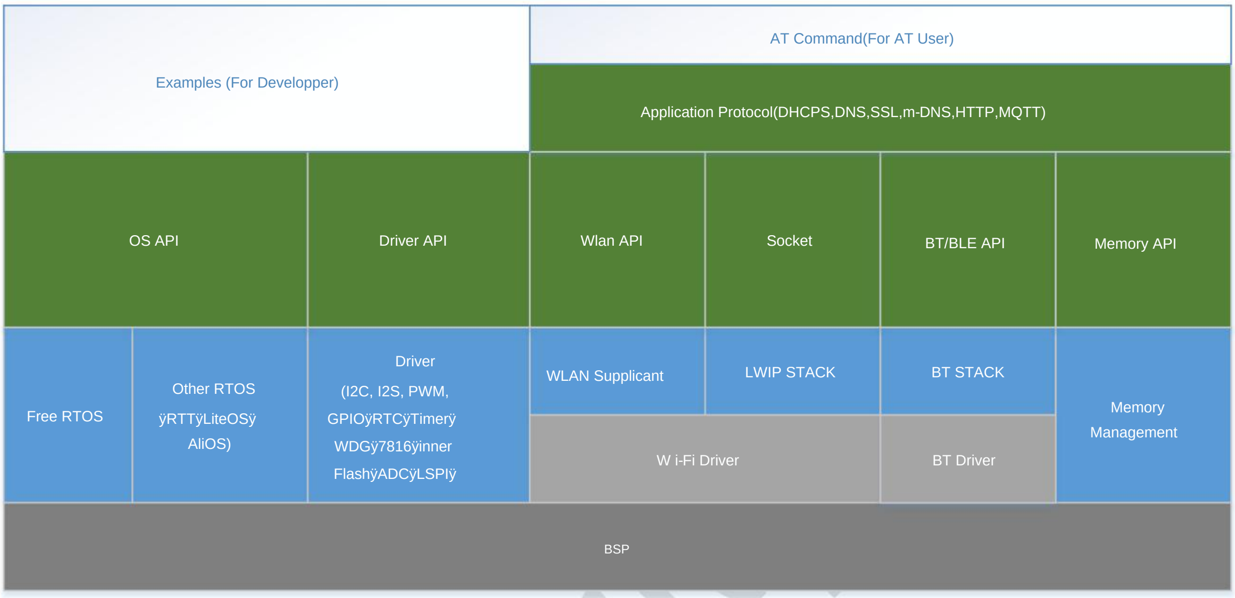
ÿ Support cloud migration

ÿ Support module production test based on serial port

ÿ Provide various usage documents

2 SDK instructions

2.1 Software Architecture



Examples (For Developper)

AT Command(For AT User)

Application Protocol(DHCPS,DNS,SSL,m-DNS,HTTP,MQTT)

OS API | Driver API | Wlan API | Socket | BT/BLE API | Memory API

Free RTOS

Other RTOS ÿRTTÿLiteOSÿ AliOS)

Driver (I2C, I2S, PWM, GPIOÿRTCÿTimerÿ WDGÿ7816ÿinner FlashÿADCÿLSPIÿ

WLAN Supplicant

LWIP STACK

BT STACK

Memory Management

W i-Fi Driver

BT Driver

BSP

Figure 2 Software Architecture Diagram

2.2 Directory structure

WM_SDK

ÿÿapp	User Program Development Entry
ÿÿbin	executable file
ÿÿdemo	Basic DEMO function
ÿÿdoc	Release Notes/API Documentation
ÿÿinclude	API header file
ÿÿld	linker script file
ÿÿlib	Wi-Fi, BT, application library
ÿÿMakefile	make script file

ÿÿplatform chip and platform-related public source code

ÿÿsrc Application program, network protocol stack, OS and third-party open source code set

ÿÿtools Build scripts, CDS IDE projects, CDK projects, and IMAGE generation tools

2.3 Compile and connect

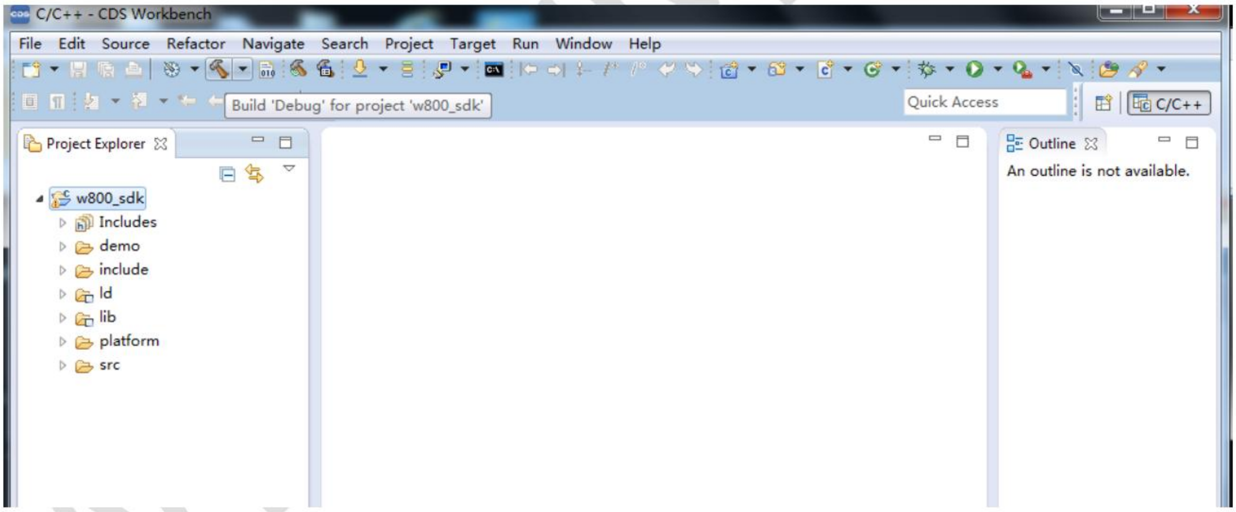
2.3.1 CDS project compilation

Open the CDS integrated compilation environment and import the project file.

When importing project files, set the sdk root path to be imported, and then the project files to be imported will be displayed, just follow the operation.

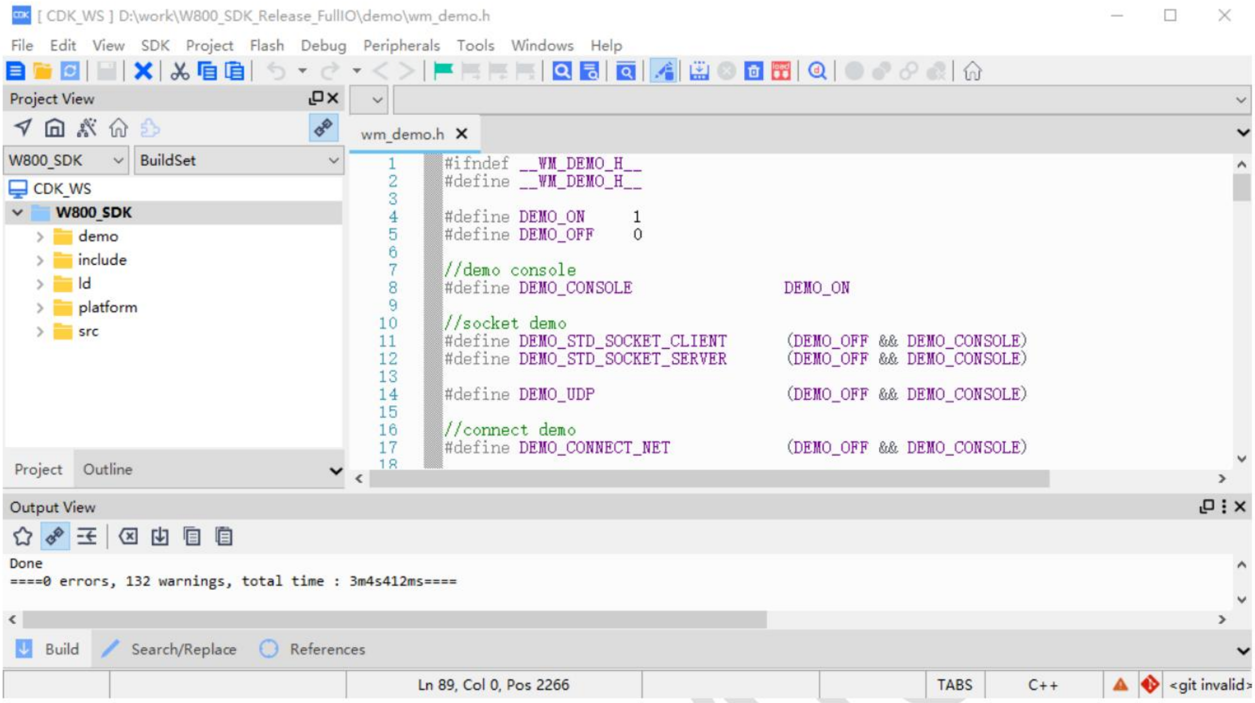
When compiling, after selecting the sdk project, click build or rebuild on the toolbar to start compiling. The compiled files are placed in bin

Under contents.



2.3.2 CDK project compilation

Go to tools\w800\projects\SDK_Project\project\CDK_WS, open the CDK project by opening the file with .cdkws suffix, and then click Rebuild All under Project to start compiling the project.



```
#ifndef __WM_DEMO_H__
#define __WM_DEMO_H__

#define DEMO_ON      1
#define DEMO_OFF     0

//demo console
#define DEMO_CONSOLE                    DEMO_ON

//socket demo
#define DEMO_STD_SOCKET_CLIENT          (DEMO_OFF && DEMO_CONSOLE)
#define DEMO_STD_SOCKET_SERVER          (DEMO_OFF && DEMO_CONSOLE)

#define DEMO_UDP                        (DEMO_OFF && DEMO_CONSOLE)

//connect demo
#define DEMO_CONNECT_NET                (DEMO_OFF && DEMO_CONSOLE)
```

```
Done
====0 errors, 132 warnings, total time : 3m4s412ms====
```

2.3.3 Command line compilation

Install the specified compilation tool and set it according to the requirements of the compilation tool. After decompressing the SDK, open the tool and jump to the root directory of the SDK, execute

just make.

Reference document: "WM_W800_SDK Script Compilation Guide"

2.3.4 Compilation results

w800.bin	original bin file
w800.img	XMODEM/OTA download image file
w800_ota.img	Only compressed firmware for OTA download is supported (save OTA space)
w800.fls	Factory burning image file
w800.map	MAP file

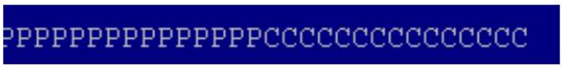
Reference document: "WM_W800_Firmware Generation Instructions"

2.4 Burn firmware

2.4.1 Burning via ROM

If the factory W800 development module has not been programmed with firmware, when it is powered on, the module will enter the ROM (UART0) and print information such as

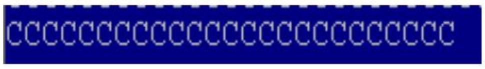
Down:



```
PPPPPPPPPPPPPPPCCCCCCCCCCCCCCC
```

If the development module of W800 leads to the BOOTMODE pin, and when the power is turned on, the BOOTMODE pin is pulled down for more than 30ms, then

The module will also enter the ROM (only UART0) and print the information as follows:



```
CCCCCCCCCCCCCCCCCCCCCCCCC
```

ROM is downloaded according to different stages of XMODEM, the results of command execution and abnormal conditions encountered in different stages of startup will be passed through

UART0 outputs different characters to represent error codes.

See documents: "WM_W800_ROM Function Brief", "WM_W800_Firmware Upgrade Guide"

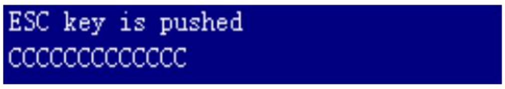
2.4.2 Burn through SECBOOT

W800 SECBOOT is a functional program that realizes user firmware migration, firmware startup verification, startup jump, and jump to ROM

XMODEM upgrade and other functions.

When the W800-based development module is powered on/reset, if UART0 receives the ESC key, the module will detect when SECBOOT starts.

Press the ESC key and jump to ROM, then enter the serial port XMODEM download mode, (UART0) print information as follows:



```
ESC key is pushed
CCCCCCCCCCCCC
```

In different XMODEM upgrade stages, if an exception is encountered, different characters will be output through UART0 to represent the error code.

See documents: "WM_W800_ROM Function Brief", "WM_W800_SECBOOT Function Brief"

2.5 Program debugging

2.5.1 Firmware debugging information

The SDK supports the standard C printf print debugging function, output to the physical serial port UART0, the user can according to their own needs in the debugging stage

The section adds its own debug print information.

The existing LOG information output of the SDK can be set through the following macro definitions in the wm_debug.h file:

```
/** Define the debugging level: info */

#define TLS_DBG_LEVEL_INFO                    TLS_DBG_OFF

/** Define the debugging level: warning */

#define TLS_DBG_LEVEL_WARNING                 TLS_DBG_OFF

/** Define the debugging level: error */

#define TLS_DBG_LEVEL_ERR                     TLS_DBG_OFF

/** Define the debugging level: dump */

#define TLS_DBG_LEVEL_DUMP                    TLS_DBG_OFF

/** general debug info switch, default: off */

#define TLS_GENERAL_DBG                       TLS_DBG_OFF
```

Wi-Fi debugging information requires a separate Wi-Fi Lib to support, and requires the user to implement two interfaces, wm_printf and wm_vprintf.

2.5.2 CK-LINK debugging

W800 supports CK-LINK debugging method.

Reference document: "WM_W800 Debug Configuration Guide"

2.5.3 AT command debugging

W800 supports AT command operation.

Reference document: "WM_W800_AT Instruction User Manual"

2.6 Development tools

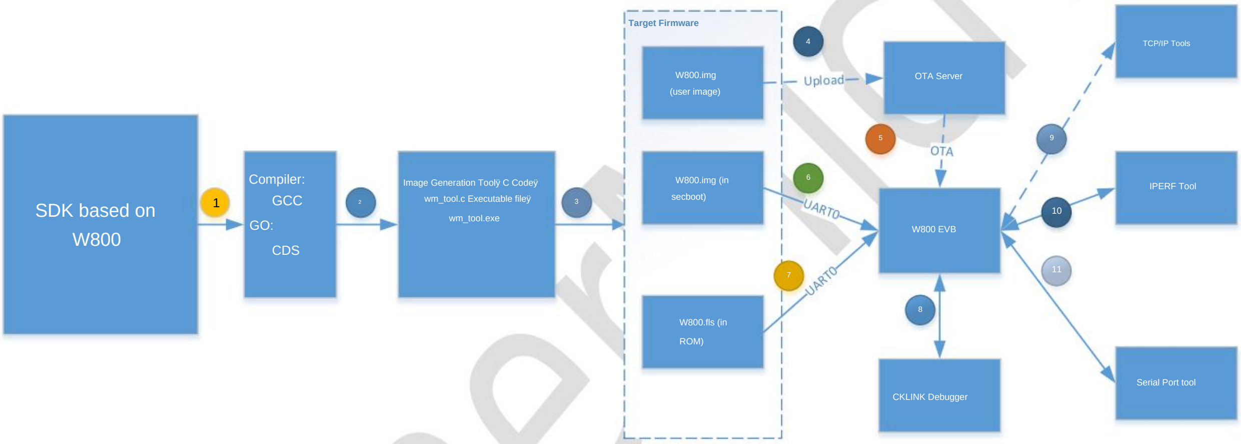


Figure 3 W800 SDK development and use process

2.7 Compilation tools

2.7.1 IDE development environment—CDS/CDK

The SDK of W800 supports CDS/CDK integrated development and compilation environment.

2.7.2 GCC from the Command Line

The SDK of W800 supports the GCC development and compilation environment of the command line, and installs a Linux virtual machine (integrated compilation environment).

Reference document: "WM_W800_SDK Script Compilation Guide"

2.8 Firmware Generation Tool

After the W800 SDK target file is compiled, it needs to be packaged to generate the target upgrade file, and the SDK comes with source code tools.

Reference document: "WM_W800_Firmware Generation Instructions"

2.9 Download Tool

W800 supports the XMODEM protocol upgrade of the serial port, and it is recommended to use SecureCRT.

Reference document: "WM_W800_Firmware Upgrade Guide"

2.10 Debugging tools

TCP/IP tool: TCP debugging assistant, used to test socket communication

Iperf: for testing network performance

3 **Development Guide**

3.1 WM_SDK startup method

The operation mode of WM_SDK is based on the XIP mode implemented by the 32KB Cache integrated in the W800 chip, and runs on QFLASH.

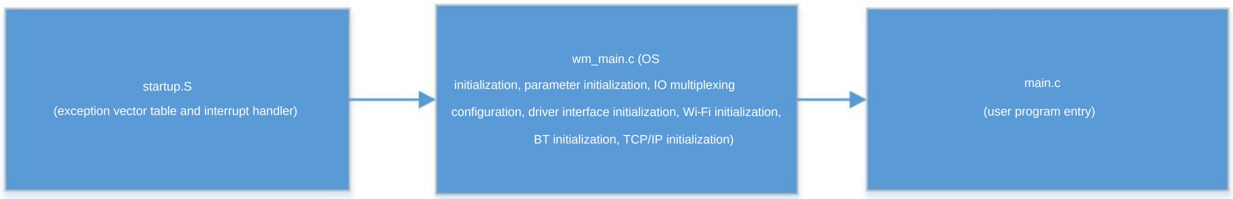
The startup process of W800 jumps to SECBOOT through ROM, and then jumps to user code space through SECBOOT.

ROM is responsible for the initialization of the basic functions of the chip and is solidified in the chip.

SECBOOT is responsible for verifying and upgrading the user space code, and places it in a section of QFLASH.

The cache space size of W800 is 32KB, when the running code size is less than 32K, CPU will not need to read instructions from QFLASH.

3.2 User program entry



startup.S

(exception vector table and interrupt handler)

wm_main.c (OS initialization, parameter initialization, IO multiplexing configuration, driver interface initialization, Wi-Fi initialization, BT initialization, TCP/IP initialization)

main.c

(user program entry)

After the W800 module is started and runs through the ROM and SECBOOT programs, it will finally run to startup.S to start execution, and then jump to

Go to the wm_main.c file to perform some initialization functions, and finally, enter main.c to print the user task.

The user program (main.c) of WM_SDK starts with the entry: UserMain(void) , the user can enter this file based on

Create your own Task in the CreateDemoTask function to complete the customized function. Currently DEMO CODE provides a

CreateDemoTask is used to test the API provided by the SDK, and customers can modify it according to actual needs.

```
void UserMain(void)
{
    printf("\n user task\n");
#if WM_DEMO
    CreateDemoTask();
#endif
}
```

3.3 Memory usage

W800 has a total of 160K Byte SRAM+128K Byte (configurable space), the address space is [0x20000000 – 0x2048000],

Used for system runtime stack and Wi-Fi transceiver BD.

The system heap size setting, located in the **gcc_csky.ld** file, is defined as follows:

```
__min_heap_size = 0x10000;

PROVIDE (__ram_end = 0x2003c000);

PROVIDE (__heap_end = __ram_end);
```

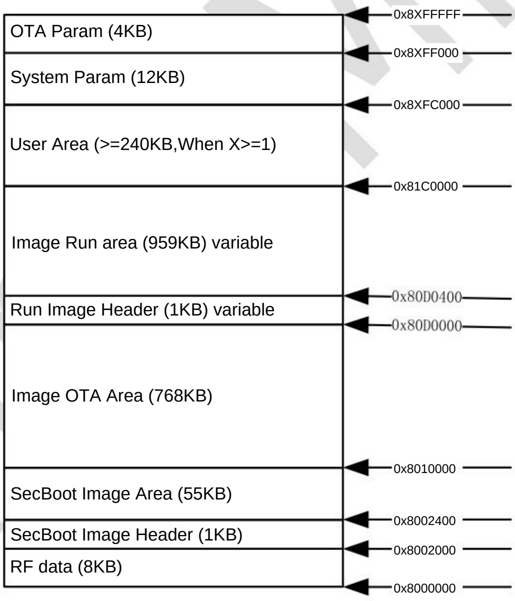
To configure the size of the system heap, where __heap_end determines the actual heap size after linking, you can actually confirm a

The size of the next heap.

Currently, the default minimum heap configuration is 64KBytes.

3.4 FLASH layout

W800 uses the built-in 2M Byte QFlash space by default, and the specific allocation is as follows (X=1 in the figure):



The parameter area available to the user is 240Kbyte, which can be adjusted according to the IMAGE size of the running area and the upgrading area.

Reference documents: "W800 Parameter Area Instructions", "WM_W800_QFLASH Layout Instructions"

3.5 User parameter management

W800 users expect to store custom parameters or operation logs. The available space for the current SDK QFLASH layout is

240KB, address range: 0x81C0000-0x81FBFFF.

If the user's code space is redundant, the user can readjust the space of the QFLASH code area and user parameter area according to his needs.

Increase the user parameter area space.

Reference document: "WM_W800_Parameter Area Instructions"

3.6 System parameter management

System parameters refer to the parameters of networking, interface configuration, mode configuration, etc. required for the operation of the W800 module, as follows:

1) Wi-Fi related (SSID, BSSID, KEY, channel list, power saving flag, speed setting, area code, working mode)

2) IP information (static IP, DHCP enabling information, NTP server, DNS server)

3) Interface configuration (UART mode configuration)

4) Other parameters (WEB)

The system parameter area is located at: 0x81FC000-0x81FEFFF. This parameter area cannot be used by users for other purposes.

Reference document: "WM_W800_Parameter Area Instructions"