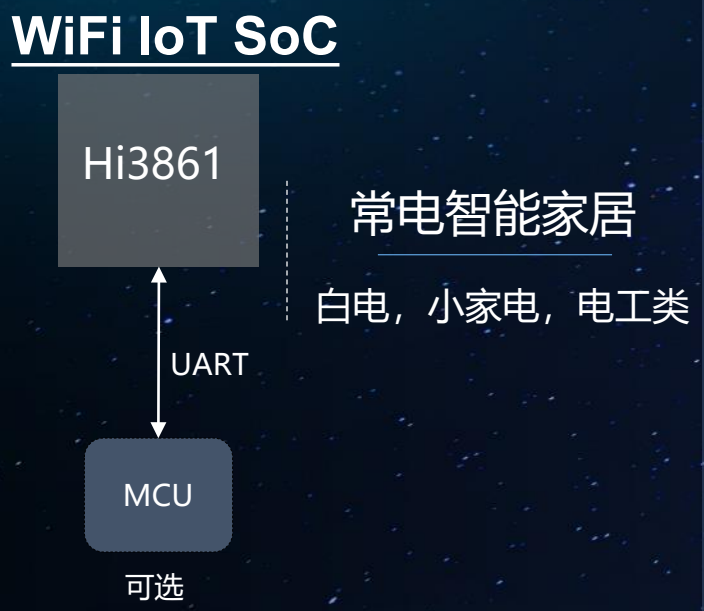
HiSpark WiFi-IoT SDK介绍

目录

Hi3861产品详细规格



主要规格

- IEEE 802.11b/g/n，1×1 2.4GHz频段（ch1～ch14）
- 支持IEEE802.11 d/e/h/i/k/v/w
- 内置PA和LNA，集成TX/RX Switch
- 支持与BT/BLE芯片共存、Mesh组网
- 支持RF自校准方案
- 功耗：Deep Sleep模式：5μA@3.3V
- DTIM1：1.27mA@3.3V
- DTIM3：0.495mA@3.3V

PHY特性

- 支持IEEE802.11b/g/n单流所有的数据速率
- 最大速率：72.2Mbps@HT20 MCS7
- 支持标准20MHz带宽和5M/10M窄带宽
- 支持STBC

MAC特性

- 支持AMPDU、Block-ACK、AMSDU
- 支持Short-GI
- 支持QoS，满足不同业务服务质量需求

CPU子系统

- 高性能32bit CPU，最大工作频率160MHz
- 内嵌SRAM 352KB，ROM 288KB，用户可用RAM 160KB，支持对接双云

外围接口

- 1个SDIO Slave接口、2个SPI Master/Slave接口、2个I2C接口、3个UART接口、15个GPIO接口、7路ADC输入、5路PWM、一路I2S

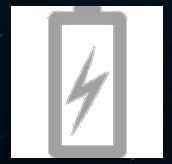
其他信息

- 封装：QFN32 5×5mm
- 工作温度：-40℃～+85℃
- 内置2M flash
- 电源电压输入范围：2.3V～3.6V
- IO电源电压支持1.8V和3.3V

海思SoC WiFi 芯片，6大优势协同切入市场

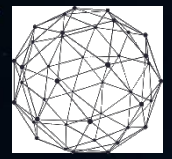
射频性能及抗干扰能力

- ✓ 发射功率及接收灵敏度普遍优于友商2~3dB，多穿一堵木质隔断
- ✓ 中强信号(RSSI>-75dBm)邻频干扰下，吞吐率不受影响



低功耗性能

- ✓ 同等条件下测试，常电功耗优于友商
- ✓ 外加32K晶体，DTIM模式再节电30%



网状Mesh组网能力

- ✓ 针对IoT场景设计
- ✓ 连接节点数250+
- ✓ 路径备份及实时优化

集成多种安全能力

- ✓ 硬件加密，不消耗RAM资源，支持多种加密协议
- ✓ 支持安全启动，防止数据篡改

生态优势

- ✓ 预集成xHiLink，苏宁智能，小京鱼等多种生态，缩短二次开发周期及认证周期
- ✓ 提供160k客户可用RAM，支持双云同时在线

一碰/靠近配网（鸿蒙/Hilink特性）

- ✓ 与SoftAP配网模式相比，进一步简化操作步骤，提升用户配网体验

注：此特性具体以鸿蒙/hilink发布为准

6大优势之 鸿蒙/Hilink配网集成，配网更快

1. 通过NFC/WiFi 靠近，设备自动发现
2. 点击“确认”，自动完成配网及连接

说明：具体配网条件请依据鸿蒙/hilink发布特性为准

常见配网方式对比一览表

（注：海思WiFi-IoT芯片同时支持SmartConfig及SoftAP等常规配网模式）

配网方式	方法	优势	局限性
鸿蒙/Hilink配网	采用NAN协议或者私有协议，近距离接触传输SSID和密码	缩减配网步骤，成功率高，配网快	• 具体配网条件请依据鸿蒙/hilink发布特性为准
SoftAP	设备启动为SoftAP模式，手机做STA连接SoftAP，通过socket将SSID和密码发送给设备	当前最常见配网方式	• 需要手动连接智能设备WiFi网络 • 手动输入要连入连接的WiFi网络的SSID和密码；
SmartConfig	设备进入混杂模式接收空中的包，手机通过一组广播/组播包将SSID和密码进行编码，设备接收到后解码获得SSID和密码	操作简单，直接用手机配网	• 配网速度慢，且成功率低

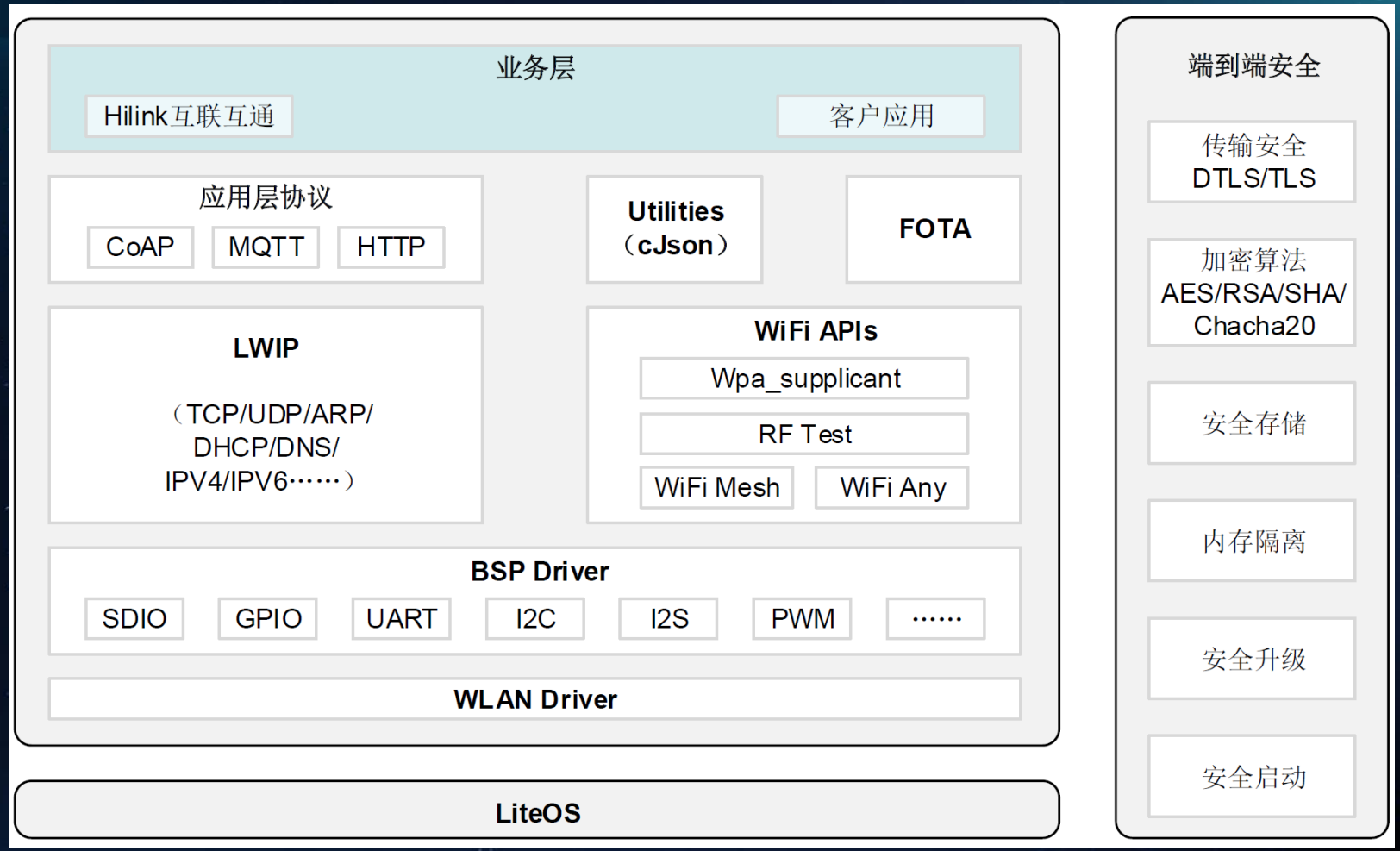
目录

Hi3861 软件架构

从下往上分为五层：

- 系统层
- 驱动层
- 网络服务层
- 组件层
- 业务层

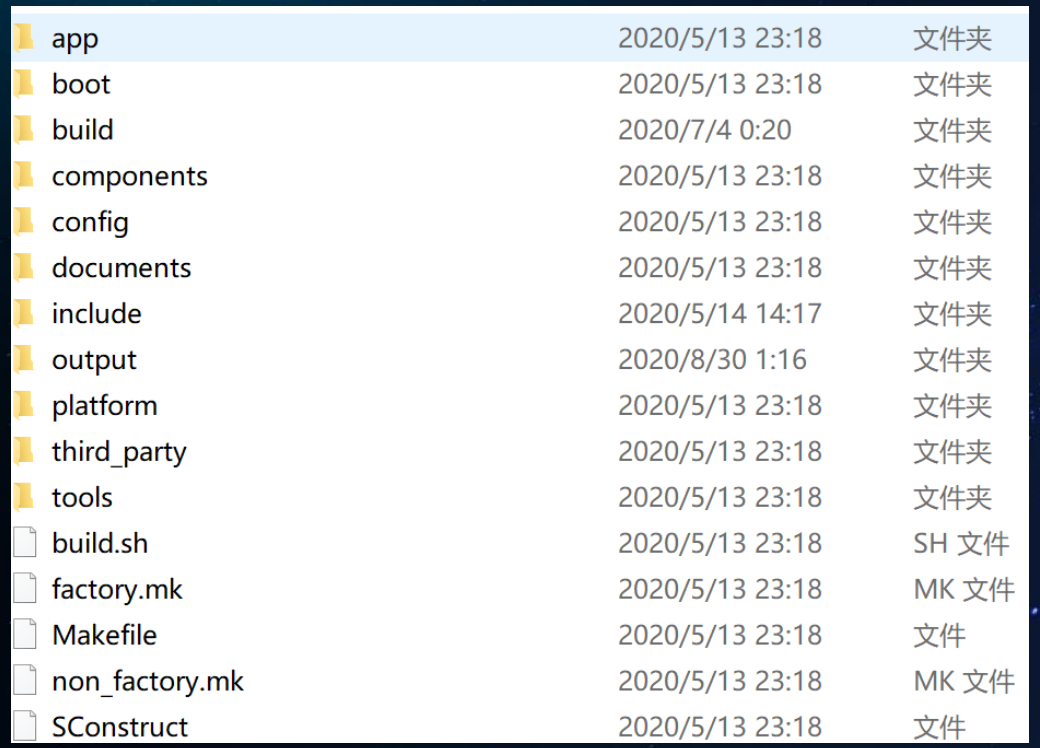
安全作为公共组件分布在相关层中。



目录

Hi3861 SDK目录

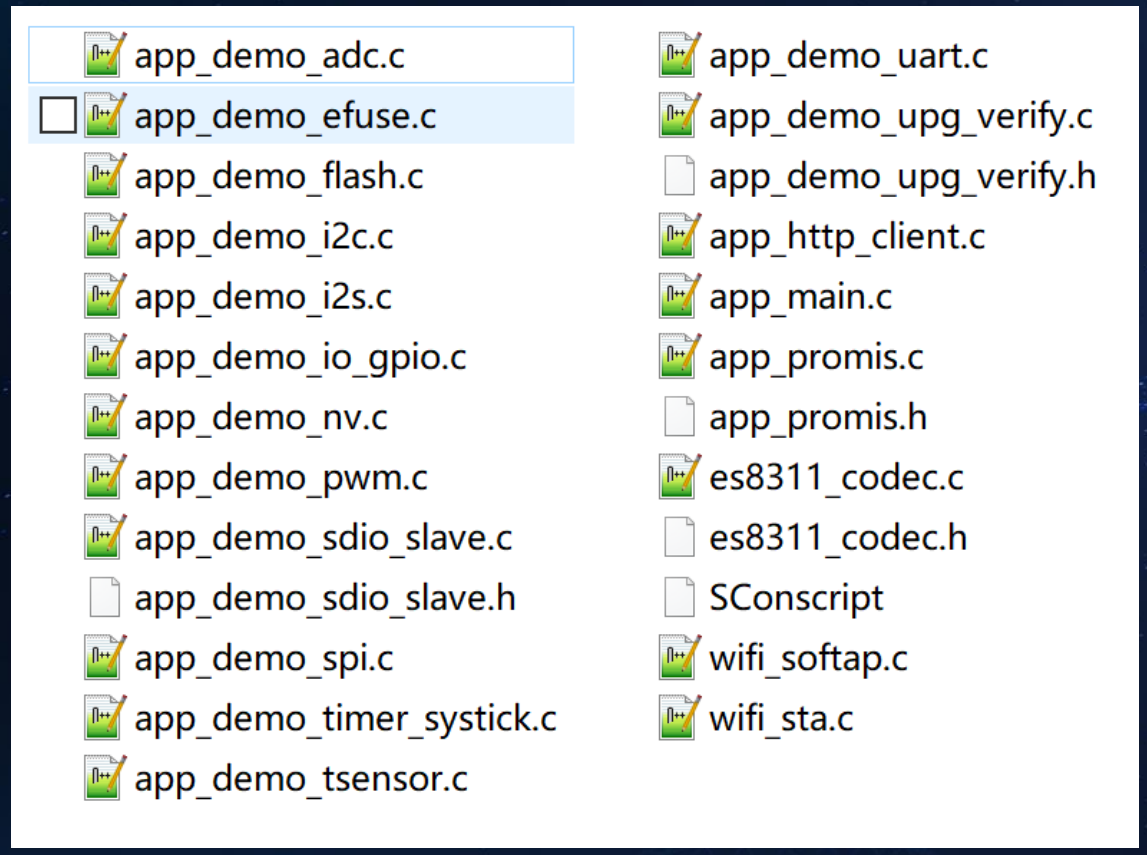
解压sdk后，目录如下：



目录	说明
app	应用层代码（其中demo程序为参考示例）
boot	Flash bootloader代码
build	SDK构建所需的库文件、链接文件、配置文件
components	SDK组件目录
config	SDK系统配置文件
documents	文档目录
include	API头文件存放目录
output	编译时生成的目标文件和中间文件，包括库文件、打印log、生成的二进制文件等
platform	SDK平台相关的文件，包括镜像、内核驱动模块等
third_party	开源第三方软件目录
tools	SDK提供的linux和Windows系统上的使用工具，包括effuse 工具、NV工具、签名工具、Menuconfig等
build.sh	启动编译脚本，同时支持“sh build.sh menuconfig”进行客制化配置
factory.mk	厂测版本编译脚本
Makefile	支持makefile编译，使用“make”或“make all”编译
non_factory.mk	非厂测编译脚本
Sconstruct	Scons编译脚本

Hi3861 参考示例

参考示例在app\demo\src目录下：



Hi3861 API

API接口头文件，均位于/include目录下，详细API接口参考文档《API开发参考.chm》

常用API:

printf()：输出格式化字符串打印到串口，头文件：hi_early_debug.h

hi_task_xx：任务创建，锁任务，sleep等，头文件：hi_task.h

hi_sem_xx：信号量创建与等待，头文件：hi_sem.h

hi_mux_xx：互斥锁创建于PEND/POST，头文件：hi_mux.h

memcpy_s：memcpy_s等标准库，头文件：hi_stdlib.h

hi_timer_start：系统定时器，10ms精度，头文件：hi_timer.h

hi_hrtimer_start：高精度定时器，us精度，头文件：hi_hrtimer.h

hi_get_tick：系统tick获取等，头文件：hi_time.h

hi_uart_write: 串口读写操作，头文件：hi_uart.h

hi_event_wait：事件创建与等待，头文件：hi_event.h

hi_flash_read：flash读写操作，头文件：hi_flash.h

hi_nv_read：NV读写操作，头文件：hi_nv.h

hi_malloc：内存申请与释放，头文件：hi_mem.h

其它重要头文件:

hi_wifi_api.h：WIFI驱动接口函数

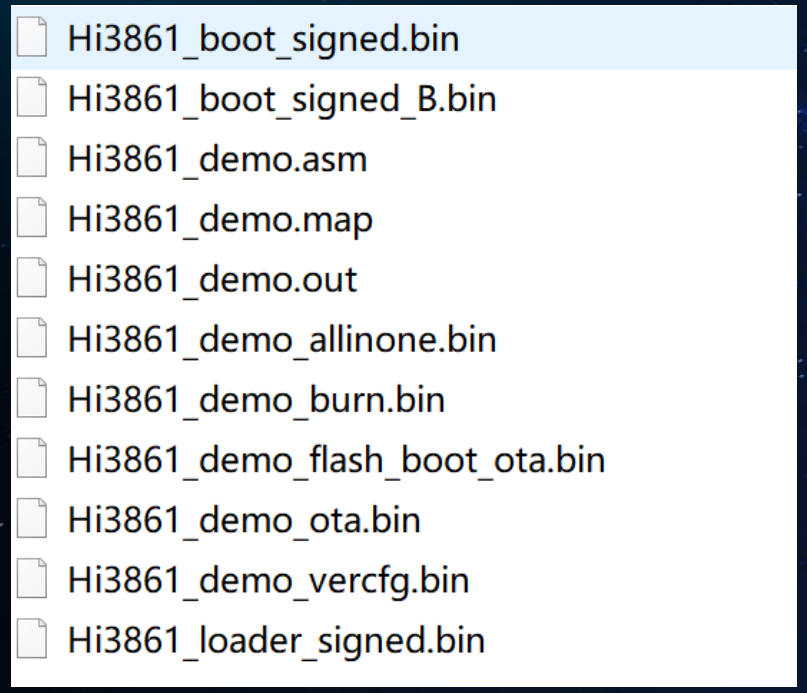
hi_types.h：基础数据类型定义

hi_cipher.h：AES、RSA、TRNG等硬件安全算法

hi_errno.h：错误码定义

Hi3861 编译生成文件说明

编译生成的文件在output/bin目录下（默认压缩升级方案，以压缩方案编译生成文件为例）：



其中Hi3861_demo_burn.bin就是烧写到底板中的程序

文件名	说明	备注
Hi3861_boot_signed.bin	带签名的bootloader文件	Flash boot
Hi3861_boot_signed_B.bin	带签名的bootloader备份文件	Flash boot备份文件
Hi3861_demo.asm	Kernel asm文件	汇编程序源文件
Hi3861_demo.map	Kernel map文件	程序的全局符号，函数的地址、占用的空间等，用于调试
Hi3861_demo.out	Kernel 输出文件	
Hi3861_demo_allinone.bin	产线工装烧写文件（已经包含独立烧写程序和loader程序）	包括3个bin： Hi3861_boot_signed_B.bin Hi3861_demo_burn.bin Hi3861_loader_signed.bin
Hi3861_demo_burn.bin	烧写程序	
Hi3861_demo_flash_boot_ota.bin	Flash Boot升级文件	Flash boot的OTA升级文件 升级方案：Flash有2个boot分区，Flash boot区和Flash boot备份区；升级时，将升级文件传输至备份区，校验通过后，将备份区Flash搬迁至Flash boot区，重启单板，使用新boot引导
Hi3861_demo_ota.bin	Kernel 升级文件	Kernel 的OTA升级文件（压缩升级方案）
Hi3861_demo_vercfg.bin	Kernel和Boot的版本号文件	
Hi3861_loader_signed.bin	烧写工具使用的加载文件	烧写使用的加载文件，只用在烧写，位于内存中

Hi3861 文档

硬件相关文档：

- Hi3861LV100 产品简介.pdf
- Hi3861V100 产品简介.pdf
- Hi3861V100 / Hi3861LV100 产线工装 用户指南.pdf
- Hi3861V100 / Hi3861LV100 单板硬件关键器件 兼容性列表.pdf
- Hi3861V100 / Hi3861LV100 开发板 使用指南.pdf
- Hi3861V100 / Hi3861LV100 开发板功耗 测试指南.pdf
- Hi3861V100 / Hi3861LV100 射频 测试指南.pdf
- Hi3861V100 / Hi3861LV100 硬件设计 Checklist.pdf
- Hi3861V100 / Hi3861LV100 / Hi3881V100 WiFi芯片 硬件用户指南.pdf
- Hi3861V100 / Hi3861LV100 / Hi3881V100 WiFi芯片 用户指南.pdf

软件相关文档：

- Hi3861V100 / Hi3861LV100 ANY软件 开发指南.pdf
- Hi3861V100 / Hi3861LV100 API 开发参考.chm
- Hi3861V100 / Hi3861LV100 AT命令 使用指南.pdf
- Hi3861V100 / Hi3861LV100 Boot API 开发参考.chm
- Hi3861V100 / Hi3861LV100 Boot移植应用 开发指南.pdf
- Hi3861V100 / Hi3861LV100 cJson 开发指南.pdf
- Hi3861V100 / Hi3861LV100 CoAP 开发指南.pdf
- Hi3861V100 / Hi3861LV100 EFUSE 使用指南.pdf
- Hi3861V100 / Hi3861LV100 HiBurn工具 使用指南.pdf
- Hi3861V100 / Hi3861LV100 HTTP 开发指南.pdf
- Hi3861V100 / Hi3861LV100 lwIP 开发指南.pdf
- Hi3861V100 / Hi3861LV100 Mesh AT命令 使用指南.pdf
- Hi3861V100 / Hi3861LV100 Mesh软件 开发指南.pdf
- Hi3861V100 / Hi3861LV100 MQTT 开发指南.pdf
- Hi3861V100 / Hi3861LV100 NV 使用指南.pdf
- Hi3861V100 / Hi3861LV100 OSA&FreeRTOS API适配指南.pdf
- Hi3861V100 / Hi3861LV100 RPL 开发指南.pdf
- Hi3861V100 / Hi3861LV100 SDK 开发指南.pdf
- Hi3861V100 / Hi3861LV100 SDK开发环境搭建 用户指南.pdf
- Hi3861V100 / Hi3861LV100 TLS&DTLS 开发指南.pdf
- Hi3861V100 / Hi3861LV100 Wi-Fi软件 开发指南.pdf
- Hi3861V100 / Hi3861LV100 安全模块 使用指南.pdf
- Hi3861V100 / Hi3861LV100 常见问题 FAQ.pdf
- Hi3861V100 / Hi3861LV100 单板冒烟 测试指南.pdf
- Hi3861V100 / Hi3861LV100 低功耗 开发指南.pdf
- Hi3861V100 / Hi3861LV100 第三方软件 移植指南.pdf
- Hi3861V100 / Hi3861LV100 二次开发网络安全 注意事项.pdf
- Hi3861V100 / Hi3861LV100 设备驱动 开发指南.pdf
- Hi3861V100 / Hi3861LV100 升级 开发指南.pdf
- Hi3861V100 / Hi3861LV100 文件系统 使用指南.pdf

目录

Hi3861 程序入口函数

app/demo/src/app_main.c：app_main()

可在该函数末尾添加应用程序处理，如果需要添加新的app，参考SDK开发指南。

```
hi_void app_main(hi_void)
{
    const hi_char* sdk_ver = hi_get_sdk_version();
    printf("sdk ver:%s\r\n", sdk_ver);

    hi_u32 ret;
    hi_flash_partition_table *ptable;

    peripheral_init();
    hi_lowpower_dsp_register_peripheral_init_entry(peripheral_init);

    ret = hi_factory_nv_init(HI_FNV_DEFAULT_ADDR, HI_NV_DEFAULT_TOTAL_SIZE, HI_NV_DEFAULT
    if (ret != HI_ERR_SUCCESS) {
        printf("factory nv init fail\r\n");
    }

    /* partion table should init after factory nv init. */
    ret = hi_flash_partition_init();
    if (ret != HI_ERR_SUCCESS) {
        printf("flash partition table init fail:0x%x \r\n", ret);
    }
    ptable = hi_get_partition_table();

    ret = hi_nv_init(ptable->table[HI_FLASH_PARTITON_NORMAL_NV].addr, ptable->table[HI_FL
```

```
    ........
    tcpip_init(NULL, NULL);

    ret = hi_wifi_init(APP_INIT_VAP_NUM, APP_INIT_USR_NUM);
    if (ret != HISI_OK) {
        printf("wifi init failed!\n");
    } else {
        printf("wifi init success!\n");
    }

    printf("start usr application here\r\n");
}
```

程序运行效果示例：

```
#
# system init status:0x0
ready to OS start
wifi init success
start usr application here
```

Hi3861 IO功能定义

1. IO复用枚举定义在include路径下的hi_io.h接口文件。

2. IO复用接口调用在app/demo/init/app_io_init.c源文件中的app_io_init函数中实现，如图：

```
hi_void app_io_init(hi_void)
{
    /* 用户需根据应用场景，合理选择各外设的IO复用配置，此处仅列出示例 */
    /* uart0 调试串口 */
    hi_io_set_func(HI_IO_NAME_GPIO_3, HI_IO_FUNC_GPIO_3_UART0_TXD); /* uart0 tx */
    hi_io_set_func(HI_IO_NAME_GPIO_4, HI_IO_FUNC_GPIO_4_UART0_RXD); /* uart0 rx */

    /* uart1 AT命令串口 */
    hi_io_set_func(HI_IO_NAME_GPIO_5, HI_IO_FUNC_GPIO_5_UART1_RXD); /* uart1 rx */
    hi_io_set_func(HI_IO_NAME_GPIO_6, HI_IO_FUNC_GPIO_6_UART1_TXD); /* uart1 tx */

    /* uart2 sigma认证使用串口 */
    hi_io_set_func(HI_IO_NAME_GPIO_11, HI_IO_FUNC_GPIO_11_UART2_TXD); /* uart2 tx */
    hi_io_set_func(HI_IO_NAME_GPIO_12, HI_IO_FUNC_GPIO_12_UART2_RXD); /* uart2 rx */

    /* SPI MUX: */
#ifdef CONFIG_SPI_SUPPORT
    /* SPI IO复用也可以选择5/6/7/8;0/1/2/3, 根据产品设计选择 */
    hi_io_set_func(HI_IO_NAME_GPIO_9, HI_IO_FUNC_GPIO_9_SPI0_TXD);
    hi_io_set_func(HI_IO_NAME_GPIO_10, HI_IO_FUNC_GPIO_10_SPI0_CK);
    hi_io_set_func(HI_IO_NAME_GPIO_11, HI_IO_FUNC_GPIO_11_SPI0_RXD);
```

3. 修改完成之后，重新编译生成新的bin文件，然后烧写到板子上验证。

通过串口工具命令窗口输入AT指令查看修改是否生效：

AT+GETIOMODE=pin（GPIO引脚编号）

返回结果第二位代表IO工作模式，如图：

```
# AT+GETIOMODE=9
+GETIOMODE:9,0,0,3
OK
```

IO工作模式表

模式 IO号	0	1	2	3	4	5	6	7
0	GPIO0	Reserved	UART1_TXD	SPI1_CK	JTAG_TDO	PWM3_OUT	I2C1_SDA	Reserved
1	GPIO1	Reserved	UART1_RXD	SPI1_RXD	JTAG_TCK	PWM4_OUT	I2C1_SCL	BT_FREQ
2	GPIO2	Reserved	UART1_RTS_N	SPI1_TXD	JTAG_TRSTN	PWM2_OUT	Reserved	SSI_CLK
3	GPIO3	UART0_TXD	UART1_CTS_N	SPI1_CSN	JTAG_TDI	PWM5_OUT	I2C1_SDA	SSI_DATA
4	GPIO4	Reserved	UART0_RXD	Reserved	JTAG_TMS	PWM1_OUT	I2C1_SCL	Reserved
5	GPIO5	Reserved	UART0_RXD	SPI0_CSN	Reserved	PWM2_OUT	I2S0_MCLK	BT_STATUS
6	GPIO6	Reserved	UART1_TXD	SPI0_CK	Reserved	PWM3_OUT	I2S0_TX	COEX_SWITCH
7	GPIO7	Reserved	UART1_CTS_N	SPI0_RXD	Reserved	PWM0_OUT	I2S0_BCLK	BT_ACTIVE
8	GPIO8	Reserved	UART1_RTS_N	SPI0_TXD	Reserved	PWM1_OUT	I2S0_WS	WLAN_ACTIVE
9	GPIO9	I2C0_SCL	UART2_RTS_N	SDIO_D2	SPI0_TXD	PWM0_OUT	Reserved	I2S0_MCLK
10	GPIO10	I2C0_SDA	UART2_CTS_N	SDIO_D3	SPI0_CK	PWM1_OUT	Reserved	I2S0_TX
11	GPIO11	Reserved	UART2_TXD	SDIO_CMD	SPI0_RXD	PWM2_OUT	RF_TX_EN_EXT	I2S0_RX
12	GPIO12	Reserved	UART2_RXD	SDIO_CLK	SPI0_CSN	PWM3_OUT	RF_RX_EN_EXT	I2S0_BCLK
13	SSI_DATA	UART0_TXD	UART2_RTS_N	SDIO_D0	GPIO13	PWM4_OUT	I2C0_SDA	I2S0_WS
14	SSI_CLK	UART0_RXD	UART2_CTS_N	SDIO_D1	GPIO14	PWM5_OUT	I2C0_SCL	Reserved

Hi3861 串口

Hi3861芯片有3个UART接口，默认情况下：

- UART0 是调试串口（复用3/4引脚）：烧写程序，打印日志信息
- UART1 是业务串口（复用5/6引脚）：AT以及客户业务串口
- UART2 用于sigma认证（复用11/12引脚）

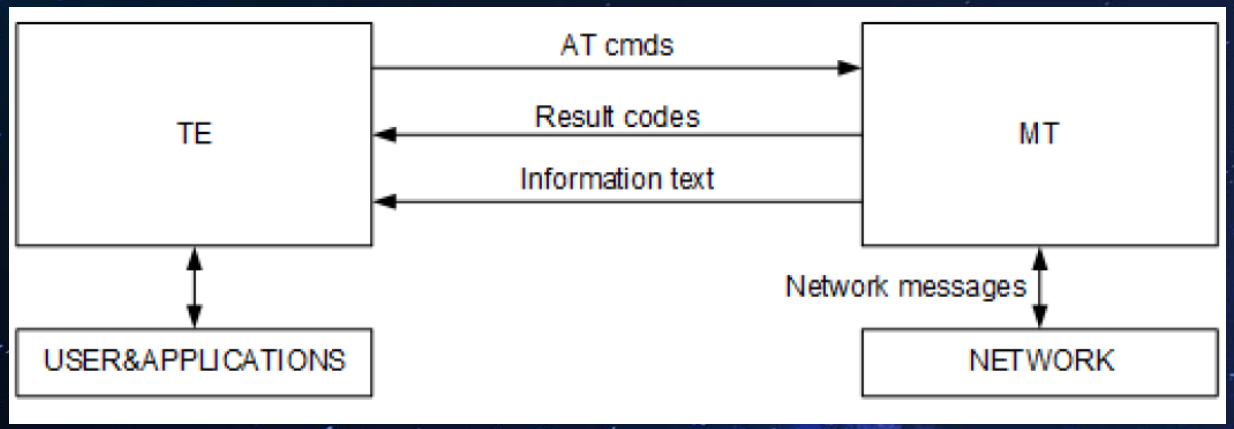
```
    /* 用户需根据应用场景，合理选择各外设的IO复用配置，此处仅列出示例 */
#ifdef CONFIG_UART0_SUPPORT
    /* uart0 调试串口 */
    hi_io_set_func(HI_IO_NAME_GPIO_3, HI_IO_FUNC_GPIO_3_UART0_TXD); /* uart0 tx */
    hi_io_set_func(HI_IO_NAME_GPIO_4, HI_IO_FUNC_GPIO_4_UART0_RXD); /* uart0 rx */
#endif

#ifdef CONFIG_UART1_SUPPORT
    /* uart1 AT命令串口 */
    hi_io_set_func(HI_IO_NAME_GPIO_5, HI_IO_FUNC_GPIO_5_UART1_RXD); /* uart1 rx */
    hi_io_set_func(HI_IO_NAME_GPIO_6, HI_IO_FUNC_GPIO_6_UART1_TXD); /* uart1 tx */
#endif

#ifdef CONFIG_UART2_SUPPORT
    /* uart2 sigma认证使用串口 */
    hi_io_set_func(HI_IO_NAME_GPIO_11, HI_IO_FUNC_GPIO_11_UART2_TXD); /* uart2 tx */
    hi_io_set_func(HI_IO_NAME_GPIO_12, HI_IO_FUNC_GPIO_12_UART2_RXD); /* uart2 rx */
#endif
```

Hi3861 AT命令

AT 命令用于控制TE（例如：PC 等用户终端）和MT（例如：移动台等移动终端）之间交互的规则，如下图所示：



Hi3861的AT命令实现了STA、AP相关wifi功能、TCP/IP相关功能、测试调试相关功能、IO相关功能及基本信息的查询设置功能，AT命令详细使用参考文档《Hi3861V100 / Hi3861LV100 AT命令 使用指南.pdf》

Hi3861 AT命令使用示例

SoftAP：

启动SoftAP示例:

```
AT+MAC=90:2B:D2:E4:CE:28
AT+STARTAP="hisilicon",0,6,2,"123456789"
AT+IFCFG=ap0,192.168.3.1,netmask,255.255.255.0,gateway,192
.168.3.2
AT+DHCPS=ap0,1
```

注意：设置MAC地址命令可选，如果不设置则使用随机MAC；设置的MAC地址为STA的地址，SoftAP的地址为STA的地址+1。

关闭SoftAP示例:

```
AT+DHCPS=ap0,0
AT+STOPAP
```

STA：

启动STA示例:

```
AT+MAC=90:2B:D2:E4:CE:28
AT+STARTSTA
AT+SCAN
AT+SCANRESULT
AT+CONN="hisilicon",,"123456789"
AT+STASTAT
AT+DHCP=wlan0,1
```

关闭STA示例:

```
AT+DHCP=wlan0,0
AT+STOPSTA
```

Hi3861 如何新增AT命令

1. AT指令初始化在app_main.c的入口函数里：
   hi_at_sys_cmd_register();

2. 在components\at\src\目录下创建自定义的AT指令.c和.h文件，
   例如：at_test.c/at_test.h

3. 举例：要增加AT+MYAT指令，at_test.c实现代码：

```
hi_u32 at_hi_test(void)
{
    printf("at cmd test!!\n");
    return HI_ERR_SUCCESS;
}

at_cmd_func g_at_test_func_tbl[] = {
    {"+MYAT", 5, HI_NULL, HI_NULL, HI_NULL, (at_call_back_func)at_hi_test},
};

#define AT_TEST_FUNC_NUM (sizeof(g_at_test_func_tbl) / sizeof(g_at_test_func_tbl[0]))

void hi_at_test_cmd_register(void)
{
    hi_at_register_cmd(g_at_test_func_tbl, AT_TEST_FUNC_NUM);
}
```

4. 在components\at\src\hi_at.c 文件hi_at_sys_cmd_register函数中调用at_test.c中定义的注册方法：

```
hi_void hi_at_sys_cmd_register(hi_void)
{
    hi_at_general_cmd_register();
    hi_at_sta_cmd_register();
    hi_at_softap_cmd_register();
    hi_at_hipriv_cmd_register();
    hi_at_ic_cmd_register();
    hi_at_test_cmd_register();
#ifdef LOSCFG_APP_MESH
    hi_at_mesh_cmd_register();
#endif
}
```

4. 重新编译，烧写新的bin文件。

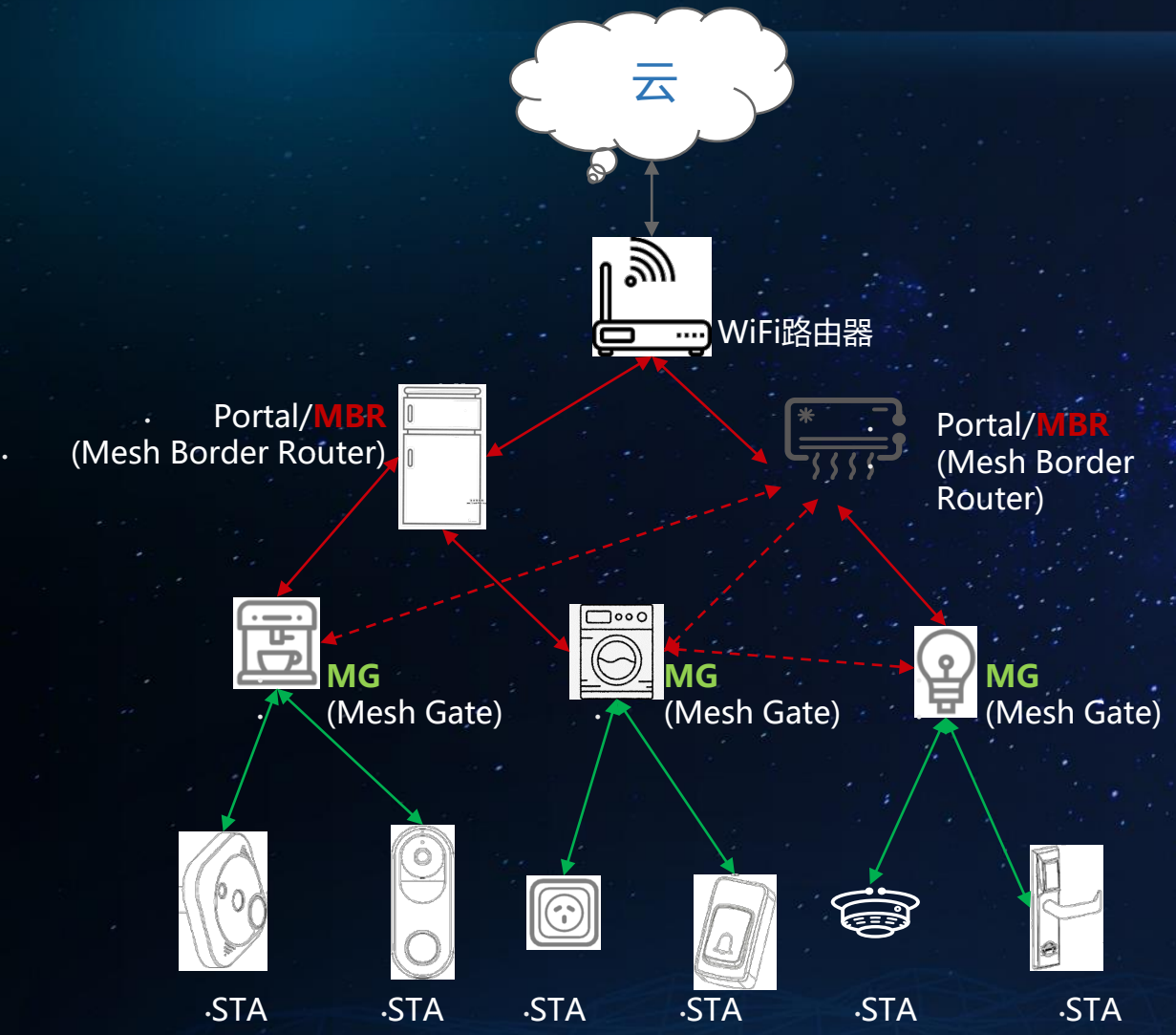
5. 测试AT+MYAT：

```
# AT+HELP
+HELPP
AT              AT+HELP         AT+MYATT        AT+SYSINFO
AT+CSV          AT+RST          AT+WREG         AT+RREG
AT+DHCP         AT+DHCPS        AT+IFCFG        AT+MAC
AT+NETSTAT      AT+IPERF        AT+PING         AT+PING6
AT+DNS          AT+STARTSTA     AT+STOPSTA      AT+SCAN
AT+SCANCHN      AT+SCANSSID     AT+SCANPRSSID   AT+SCANRESULT
AT+CONN         AT+FCONN        AT+DISCONN      AT+STASTAT
AT+RECONN       AT+PBC          AT+PIN         AA+PIISHOW
AT+CALCPSK      AT+STARTAP      AT+SETAPADV     AT+STOPAP
AT+SHOWSTA      AT+DEAUTHSTA    AT+APSCAN       AT+ALTX
AT+ALRX         AT+RXINFO       AT+CC           AT+TPC
AT+SETIOMODE    AT+GETIOMODE    AT+GPIODIR      AT+WTGPIO
AT+RDGPIO       AT+MYAT
OK

# AT+MYAT
at cmd test!!
```

目录

Hi3861 Mesh组网



Mesh组网特点：

- **骨干**(MBR/MG) + **叶节点**(STA)组网模式

Mesh组网角色分类：

- **MBR**：直连路由器WiFi，同时允许其它节点，是Mesh网络的外部出口。Mesh网络中允许存在多个MBR角色（**插座、空调常电设备**）
- **MG**：Mesh网络中间节点，可转发其它节点的数据。（**插座、冰箱常电设备**）
- **STA**：Mesh网络的末端节点，只允许优选一个MBR或MG节点进行连接，不支持其它节点连接自己。（**边缘设备、智能门锁、智能门铃低功耗设备**）

Mesh组网关键特性：

- 资源占用少，最大资源节点MBR:**SRAM < 50K　　Flash < 60K**
- 最大256节点，**自动组网**，支持叶节点节能
- 支持**Multi-Portal**，支持快速切换
- 支持**P2P、P2MP和MP2P**多种通信形式，分组控制
- 支持IPv4/IPv6/**包头压缩**
- **Beacon冲突检测**
- **时钟同步**

Thank You