

HF-BSL600 BLE&SLE 模块

Hilink SDK 开发指导

V1.0

本文档适用于如下系列产品，产品硬件说明请参阅产品手册。

	HF-BSL600	SMT: 22.5 x 13.5 x 2.5mm
---	------------------	-------------------------------------

目录

1. 功能描述	4
1.1. EVK 样式	4
2. SDK 编译环境	6
2.1. 编译方式说明	6
2.2. 搭建 UBUNTU 编译环境	6
2.2.1. 安装 Vmware Tools	6
2.2.2. 安装编译环境	6
2.2.3. 编译	8
2.2.4. SDK 目录结构	9
2.3. 设置共 SAMBA 代码编辑	11
2.4. 固件烧录	11
2.5. 用户添加源代码文件	15
2.5.1. 用户函数定义约定	15
2.5.2. 用户添加源代码文件	16
2.5.3. 用户添加第三方库	16
2.5.4. 用户添加编译文件及其头文件目录	17
2.6. 开发注意事项	19

2.7.	常见问题分析.....	19
2.7.1.	线程崩溃问题.....	19
2.8.	资源分配	20
2.8.1.	1MB Flash 分区表.....	20
2.8.2.	RAM 资源.....	20
2.9.	串口打印调试信息.....	21
2.9.1.	串口调试说明.....	21
2.9.2.	串口 log 输出说明.....	21
3.	HILINK 开发指导.....	23
3.1.	产品信息修改.....	23
3.1.1.	产品信息对应.....	23
3.1.2.	产品信息修改.....	24
3.1.3.	BLE 配网添加设备.....	24
3.2.	状态和收发数据 API 接口介绍	27
3.3.	APP_MAIN.C 程序说明	27
3.3.1.	hf_bsl_600_gpio_fid_to_pid_map_table	27
3.3.2.	user_define_at_cmds_table	28
3.3.3.	uart_rcv_callback	28
3.3.4.	app_main	28
3.4.	例子	28
3.4.1.	PWM 接口.....	28
3.4.2.	ADC 功能.....	29
3.4.3.	线程、定时器接口.....	29
4.	AT 指令说明	30
4.1.	模块运行模式配置.....	30
4.1.1.	从透传模式切换到命令模式.....	30
4.1.2.	SecureCRT 软件使用	31
4.1.3.	SSCOM 软件使用.....	33
4.2.	AT+指令集概述	34
4.2.1.	命令格式.....	34
4.2.1.1.	AT+WSMAC: 设置/查询模块的 STA MAC 地址参数, 复位后设置生效.....	35
4.2.1.2.	AT+SLEMAC: 查询模块的星闪 MAC 地址参数.....	35
4.2.1.3.	AT+Z: 重启模块.....	35
4.2.1.4.	AT+UART: 设置或查询串口操作, 复位后设置生效.....	35
4.2.1.5.	AT+VER: 查询 SDK 版本号	36
4.2.1.6.	AT+SN: 设置/查询 Hilink SN 值.....	36
4.2.1.7.	AT+PRODUCT: 设置/查询 产测模式.....	37
4.2.1.8.	AT+RESET: 重置模块, 恢复成配网状态并重启。	37
4.2.1.9.	AT+SLE: 设置/查询 星闪开关.....	37
4.2.1.10.	AT+MODE: 设置/查询 hilink 通讯协议类型.....	38
4.2.1.11.	AT+BLEADP: 配置/查询 BLE 广播参数	38
4.2.1.12.	AT+SLEADP: 配置/查询休眠状态时 SLE 广播参数.....	38

附录 A: 汉枫联系方式 39

历史记录

V 1.0	02-14-2025	初版
V 1.1	2025-11-28	更新 V15.2.0.301_1.3.1 版本对应分区

1. 功能描述

HF-BSL600 系列模块是上海汉枫电子科技有限公司开发的一款低功耗嵌入式蓝牙+BLE 模块。目前系列中包含多款封装不同的模块。

本系列中所有模块可以使用同一份 SDK 和 API 手册

本文档所提到的资料都可在链接中获取, <http://hftp.hi-flying.com:9000/>

Index of /

- [HF-BSL600/](#)
- [HF-LPT262/](#)
- [Tools/](#) ← 烧录工具
- [hiflying.txt](#)
- [Parent Directory](#)
- [SDK/](#) ← SDK开发相关资料
- [general/](#) ← 标准固件相关资料

1.1. EVK 样式

开发版图如下，开发板使用 **USB 转串口** 的芯片，可从如下链接下载对应驱动。
<http://www.hi-flying.com/download-center-1/applications-1/download-item-usb-serial-drivers>

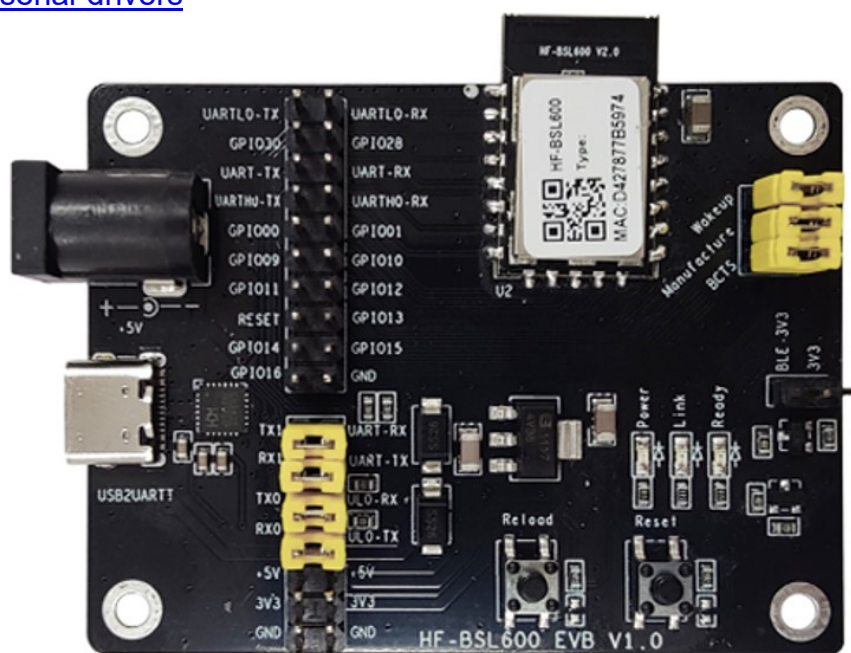


Figure 1. 开发板外观图

Table1. EVK 接口表

功能	名称	描述
外面接口	USB2UART	提供 USB type C 转两路串口功能，同时用于供电输入

	TX/RX	通讯串口，与 MCU 应用数据交互或者 AT 指令 可以取下跳帽，这样断开与底板 USB 串口的连接，把模块的 UART-RX 和 UART-TX 接到客户 MCU 上来联调测试
	L0TX/L0RX	调试串口，用于 log 打印和固件烧录用，log 打印波特率 115200
	电源接口	直流 5V 输入，可悬空，USB 供电就行了。
LED	Power	电源指示灯
	Link	nLink 指示灯，用作连接指示灯，低电平有效 亮：连接手机成功 灭：未连接手机 需连接开发板右上角 wakeup 跳帽
	Ready	默认输出低电平,有数据发送时输出高电平,用于唤醒 MCU,数据发送完毕后输出低电平 需连接开发板右上角 BCTS 跳帽
按钮	Reload/GPIO14	长按 3 秒松开后进入重置功能，解绑并重新配网功能 需连接开发板右上角 Manufacture 跳帽
	Reset	复重启按键
	GPIO30	内有上拉电阻，拉低唤醒蓝牙，不需要休眠的话请接下拉到 GND 无手机连接时为低功耗工作状态，需要发串口指令时，拉低此引脚唤醒模组后再发。 当有手机连接时退出低功耗，可正常收发数据，无须拉低此引脚。

2. SDK 编译环境

2.1. 编译方式说明

BSL600 SDK 采用 Ubuntu 环境下进行编译，SDK 中提供的 example、第三方库等都以此种方式编译提供。

2.2. 搭建 Ubuntu 编译环境

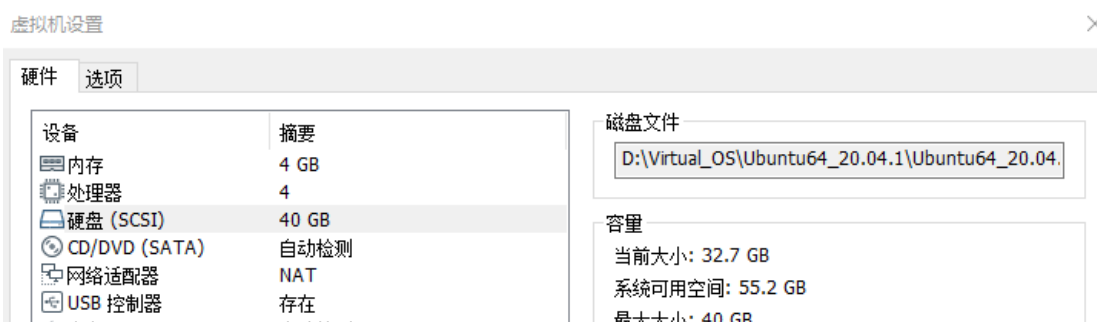
下载 Linux64 位系统，如下是 ubuntu 系统。

<https://ubuntu.com/download/desktop>

安装 VMWARE WORKSTATION，安装 ubuntu 系统。更换系统软件源：

<https://blog.csdn.net/YooLcx/article/details/104527734>

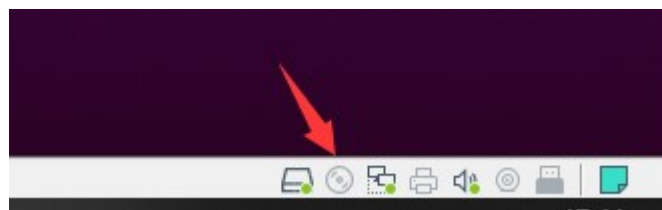
虚拟机的内存调大一点：



如下章节根据开发习惯，可选择安装。

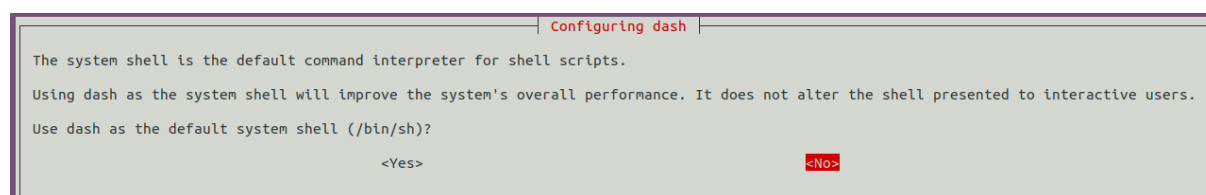
2.2.1. 安装 VMware Tools

安装 VMware tools，点击 VMware 右下角光盘图标，点击设置选择如下文件后点击连接。



2.2.2. 安装编译环境

配置 shell：默认使用 bash，打开命令终端，执行 `sudo dpkg-reconfigure dash`，选择 no 即可；



安装 Cmake：打开命令终端，执行 `sudo apt install cmake`，完成安装；

```
user@ubuntu:/mnt/hgfs/LPB100_Project/6_LPT262/LPT26x-HSF-4MB$ sudo apt install cmake
Reading package lists... Done
Building dependency tree
Reading state information... Done
cmake is already the newest version (3.16.3-1ubuntu1.20.04.1).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
```

安装 python3 环境:

步骤①打开命令终端, 输入命令 `python3 -V`, 查看 Python 版本号, 推荐 python3.8.0 以上版本;

```
user@ubuntu:/mnt/hgfs/LPB100_Project/6_LPT262/LPT26x-HSF-4MB$ python3 -V
Python 3.8.10
```

步骤②如果 Python 版本太低, 请使用命令 `sudo apt-get update` 更新系统到最新, 或通过命令 `sudo apt-get install python3 -y` 安装 Python3 (需 root/sudo 权限安装), 安装后再次确认 Python 版本;

如果仍不能满足版本要求, 请从 <https://www.python.org/downloads/source/> 下载对应版本源码包, 下载与安装方法请阅读

<https://wiki.python.org/moin/BeginnersGuide/Download> 和源码包内 README 内容

步骤③安装 Python 包管理工具, 运行命令 `sudo apt-get install python3-setuptools python3-pip -y` (需 root/sudo 权限安装)

步骤④安装 Kconfiglib 14.1.0+, 使用命令 `sudo pip3 install kconfiglib` (需 root/sudo 权限安装), 或从 <https://pypi.org/project/kconfiglib> 下载.whl 文件 (例如: `kconfiglib-14.1.0-py2.py3-none-any.whl`) 后, 使用命令 `pip3 install kconfiglib-xxx.whl` 进行安装 (需 root/sudo 权限安装), 或者下载源码包到本地并解压, 使用 `python setup.py install` 进行安装 (需 root/sudo 权限安装)。安装完成界面如下图所示

```
# pip3 install kconfiglib-13.2.0-py2.py3-none-any.whl
Processing ./kconfiglib-13.2.0-py2.py3-none-any.whl
Installing collected packages: kconfiglib
Successfully installed kconfiglib-13.2.0
tools/menueconfig
```

步骤⑤安装升级文件签名依赖的 Python 组件包 `pycparser`, 从 <https://pypi.org/project/pycparser/> 下载.whl 文件 (例如: `pycparser-2.21-py2.py3-none-any.whl`),

使用 `pip3 install pycparser-xxx.whl` 进行安装（需 `root/sudo` 权限安装），或者下载源码包到本地并解压，使用 `python setup.py install` 进行安装（需 `root/sudo` 权限安装）。安装完成后界面会提示 Successfully intalled pycparser-2.21

`sudo apt update`

`sudo apt install ninja-build`

```
user@ubuntu:~/mnt/hgfs/LPB100_Project/6_LPT262/LPT26x-HSF-4MB$ sudo apt install ninja-build
Reading package lists... Done
Building dependency tree
Reading state information... Done
ninja-build is already the newest version (1.10.0-1build1).
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
```

2.2.3. 编译

在 Ubuntu 下打开命令终端进入工程根目录，执行 `./build.py -c` 进行全量编译后，提示 Input index numbers for build or q for quit(num/q)，输入 0 选择 standard-bs20-n1200 进行编译。

```
chen1@ubuntu:~/Project/BSL600-HSF-1MB-Hilink$ sudo ./build.py -c
[sudo] password for chen1:
[Warning]:sdk libs import failed.
Target_name invalid
Here are the matching target_names
0 : standard-bs20-n1200
1 : bs20-n1200-rcu
2 : standard-bs21-n1100
3 : standard-bs21e-1100e
4 : bs21e-1100e-rcu
5 : bs21e-1100e-slp
6 : standard-bs22-n1200

Here are the matching aliases target_name

Do you want to build them?
Input index numbers for build or q for quit(num/q):0

[0, 1, 1, 1, 1, 1, 1]
['/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/loaderboot_sign.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/partition/standard-bs20-n1200/part
ition.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/flashboot_si
gn_a.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/tools/Product_0xF8000_V1.0.0.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/flashboot_sign_b.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/output/bs20/acore/standard-bs20-n1200/application_sign.bin',
'/home/c
hen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/nv/standard-bs20-n1200/bs20_all_nv.bin']
[0, 2416967680, 2416971776, 2417983488, 2417012736, 2417053696, 2418008064]
[0, 4096, 40960, 4096, 40960, 532480, 8192]
[25296, 1024, 37952, 512, 37952, 481824, 4096]
[0, 1, 1, 1, 1, 1, 1]
['/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/loaderboot_sign.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/flashboot_sign_a.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/flashboot_sign_b.bin']
[0, 2416971776, 2417012736]
[0, 40960, 40960]
[25296, 37952, 37952]
[0, 1, 1]
['/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/loaderboot_sign.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/output/bs20/acore/standard-bs20-n1200/application_sign.bin']
[0, 2417053696]
[0, 532480]
[25296, 481824]
[0, 1]
['/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/boot_bin/loaderboot_sign.bin',
'/home/chen1/Project/BSL600-HSF-1MB-Hilink/interim_binary/bs20/bin/nv/standard-bs20-n1200/bs20_all_nv
.bin']
[0, 2418008064]
[0, 8192]
[25296, 4096]
[0, 1]
packet success!
```

编译完成，会生成 output 文件夹及以下文件：

网络 > 192.168.1.111 > share > BSL600-HSF-1MB-Hilink > output

名称	修改日期	类型	大小
bs20	2025/1/26 17:56	文件夹	
BSL600_hilink.fwpkg	2025/1/27 15:25	FWPKG 文件	576 KB
BSL600_hilink_UPGRADE.bin	2025/1/27 15:25	BIN 文件	309 KB

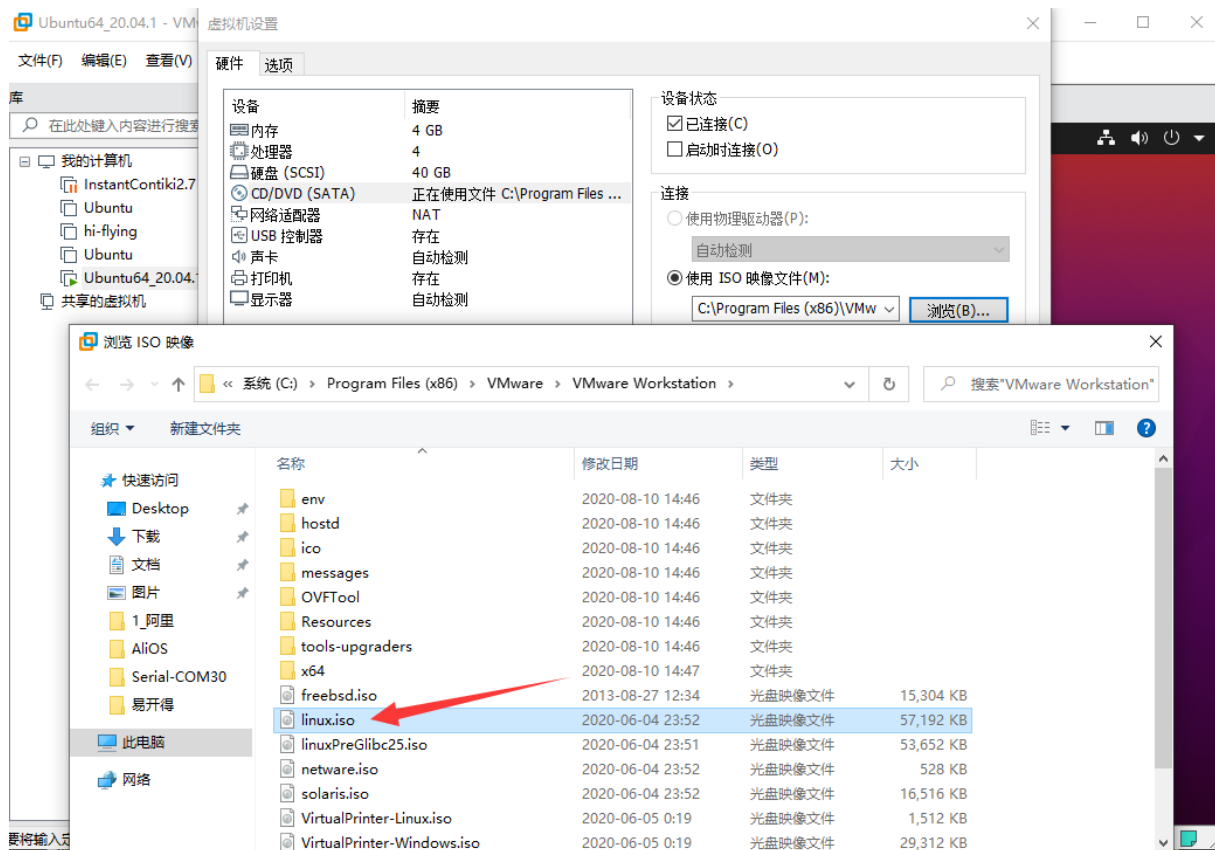
生成的固件说明：

BSL600_hilink.fwpkg：固件包包含 BootLoader、APP 固件、JSON 标志头合一文件，可使用原厂串口工具刷此固件

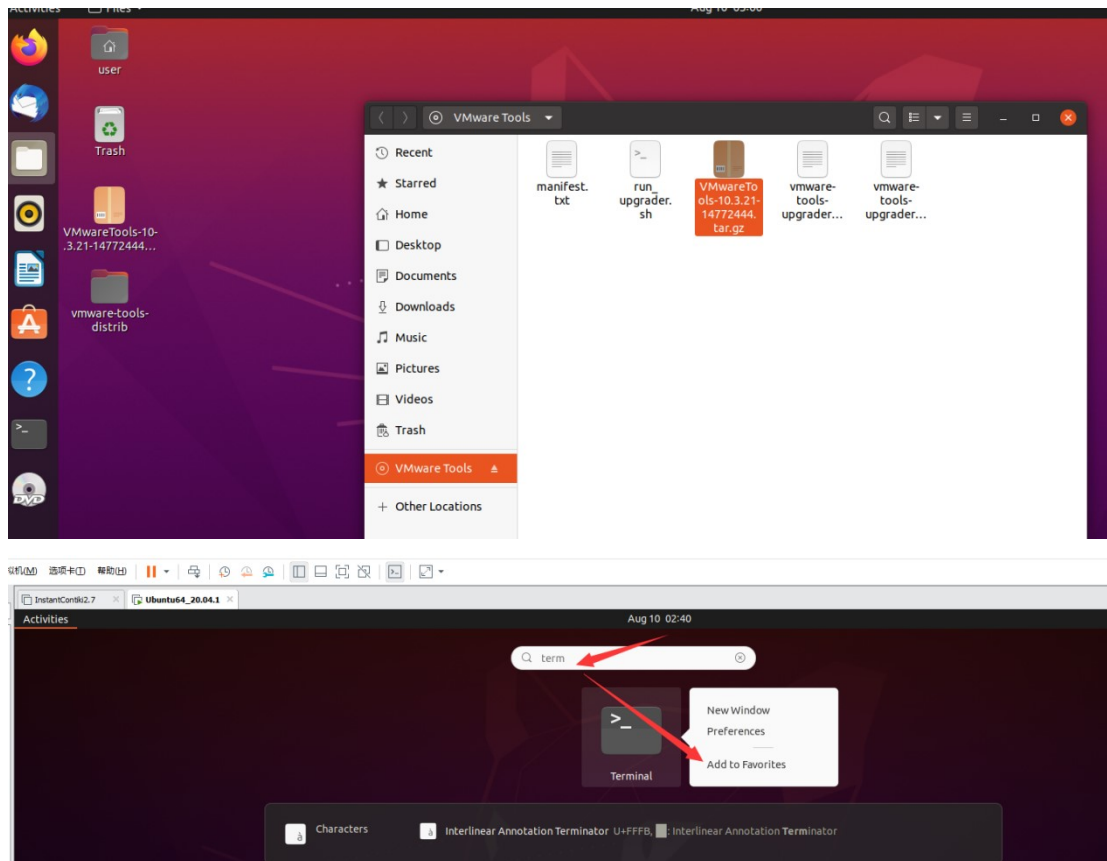
BSL600_hilink_UPGRADE.bin：APP OTA 全量升级固件，在 APP 固件的基础上有压缩算法，适用于 OTA 升级使用。

2.2.4. SDK 目录结构

目录	说明
application	Hilink 应用层代码（其中包含 demo 程序为参考示例）
build	SDK 构建所需的脚本、配置文件。
drivers	驱动代码。
include	API 头文件存放目录。
build.py	编译入口脚本。
interim_binary	库存放目录。
kernel	内核代码和 OS 接口适配层代码。
middleware	中间件代码
open_source	开源代码。
output	编译时生成的目标文件与中间文件（包括库文件、打印 log、生成的二进制文件等）。
tools	包含编译工具链（包括 linux 和 windows）、镜像打包脚本、NV 制作工具和签名脚本等。
CMakeLists.txt	Cmake 工程顶层 “CMakeLists.txt” 文件。
config.in	Kconfig 配置文件。



复制文件到桌面，解压缩后安装



打开终端，运行如下指令 `sudo ./vmware-install.pl`，完成安装。

```
user@ubuntu:~/Desktop/vmware-tools-distrib$ sudo ./vmware-install.pl
[sudo] password for user:
The installer has detected an existing installation of open-vm-tools packages
on this system and will not attempt to remove and replace these user-space
applications. It is recommended to use the open-vm-tools packages provided by
the operating system. If you do not want to use the existing installation of
open-vm-tools packages and use VMware Tools, you must uninstall the
open-vm-tools packages and re-run this installer.
The packages that need to be removed are:
open-vm-tools
Packages must be removed with the --purge option.
The installer will next check if there are any missing kernel drivers. Type yes
if you want to do this, otherwise type no [yes] ^[A
```

2.3. 设置共 Samba 代码编辑

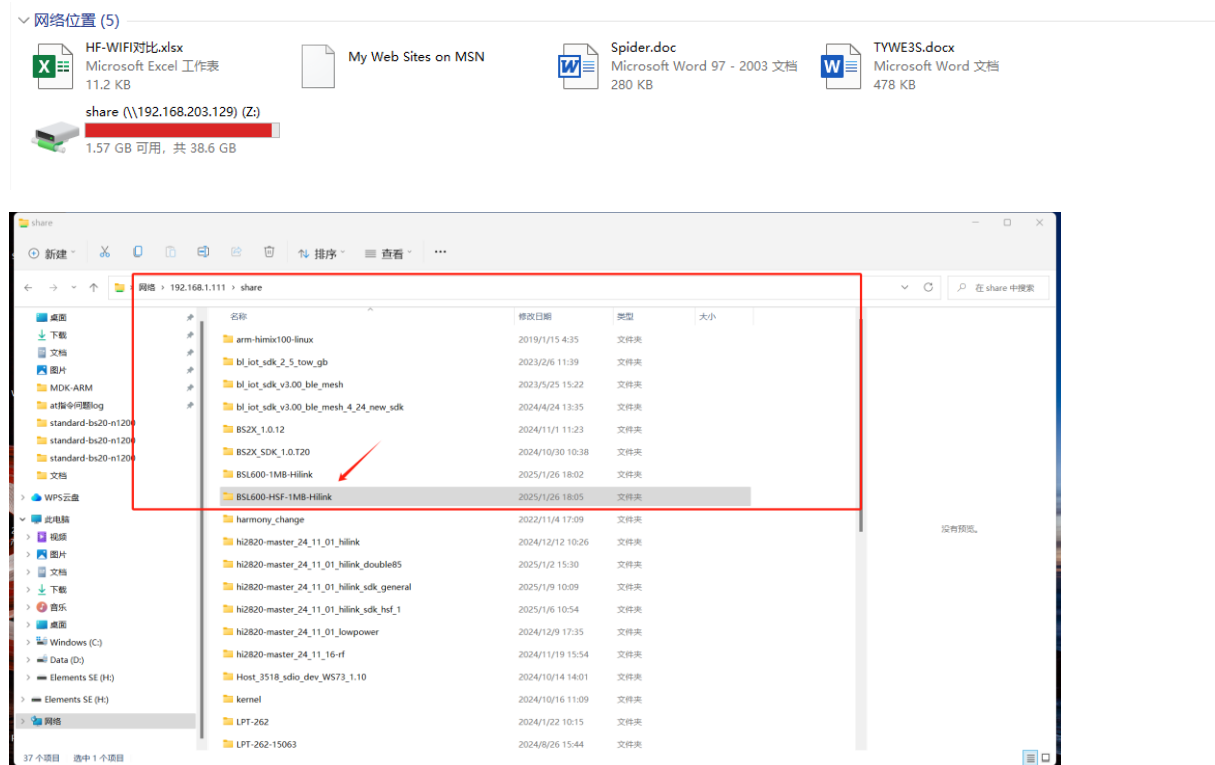
使用 Samba 方式共享 linux 中的代码（SDK 放在 Linux 环境中），在 windows 下用 sourceinsight(版本需要 V4 以上)，vscode 等工具进行编辑，linux 下编译生成固件，然后 windows 下刷写固件，samba 共享方式操作可参考如下链接。

<https://www.linuxidc.com/Linux/2018-11/155466.htm>

对于可以共享编辑的目录，需要对所有文件增加写权限，指令如下。

- `chmod 777 ~/XXXXXXX/ -R`

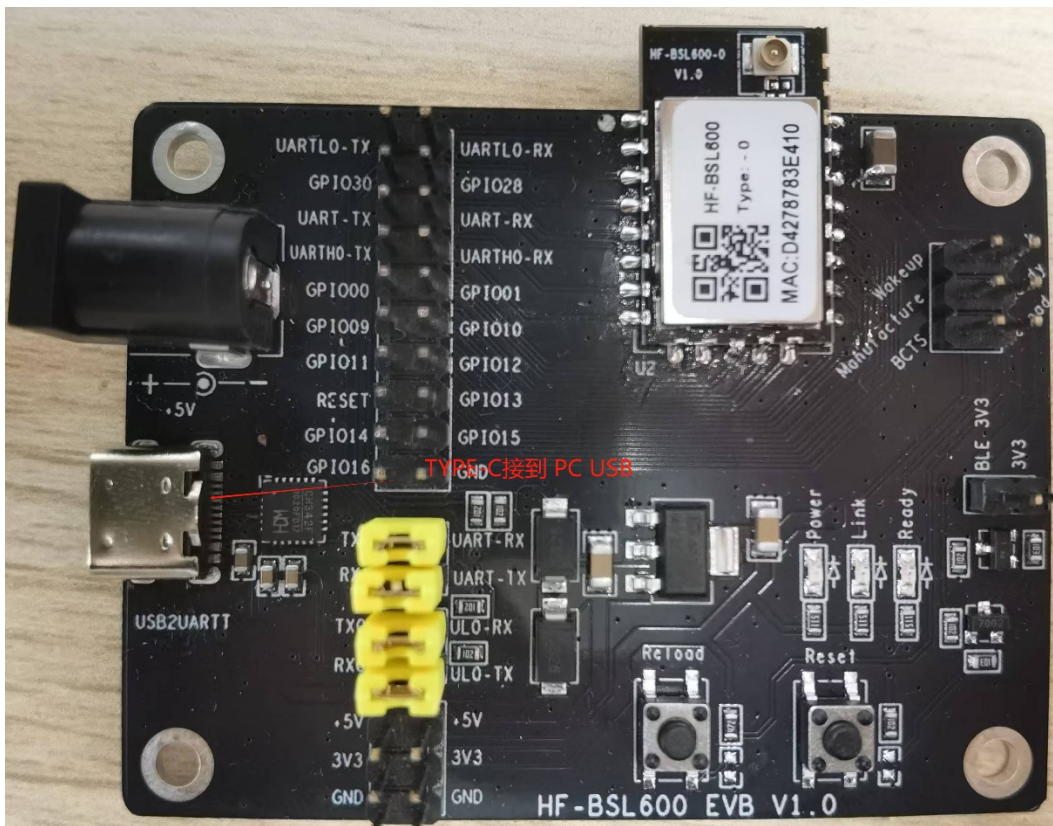
映射到驱动盘之后，就可以在 windows 下使用编辑工具进行代码编辑。



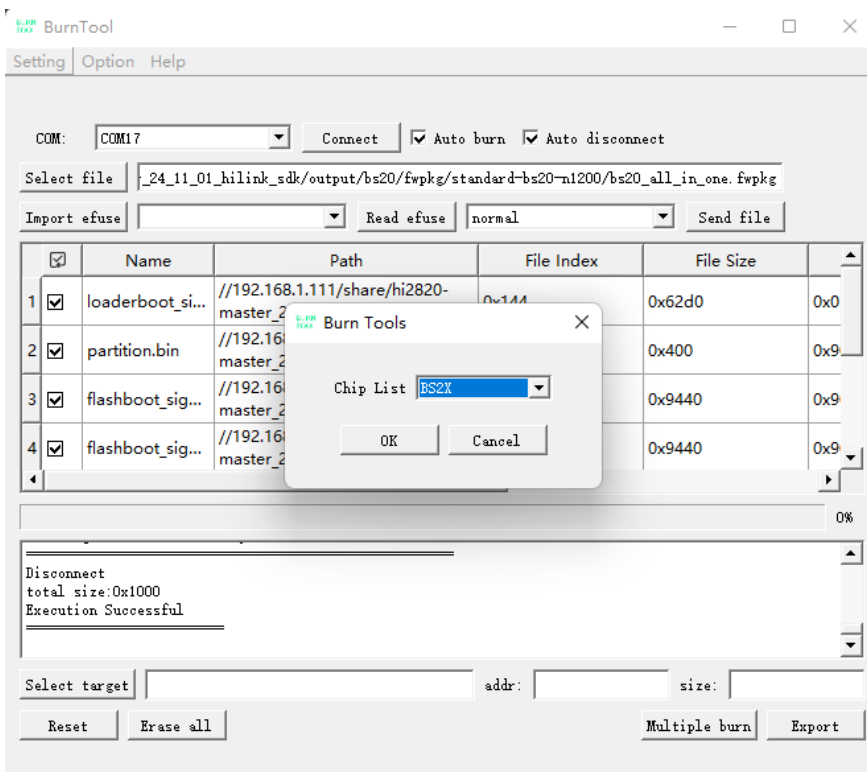
2.4. 固件烧录

默认编译出的固件即为通用固件
下载 `burnttool` 工具。

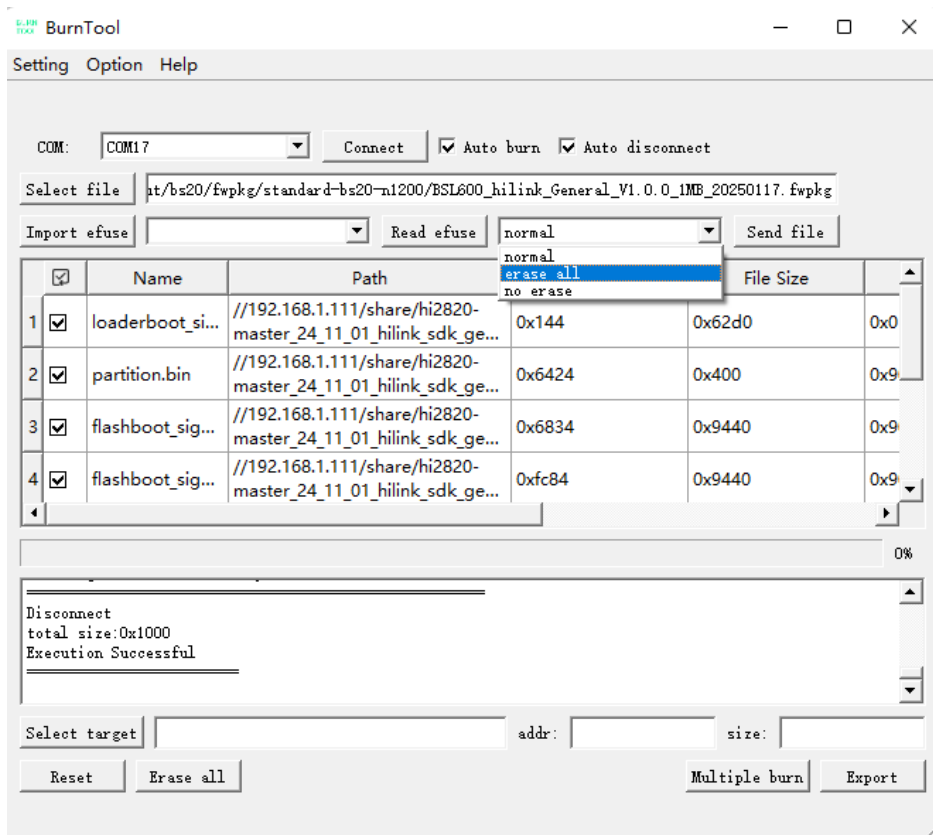
按下图方式接到 PC USB。



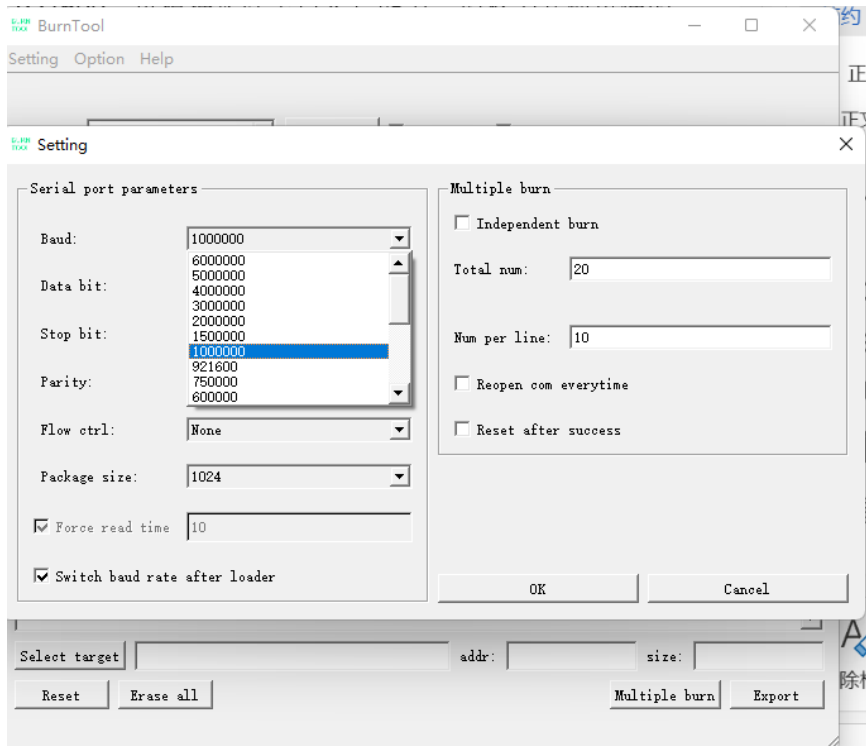
打开工具，点击【option】选择 BS2X 芯片。



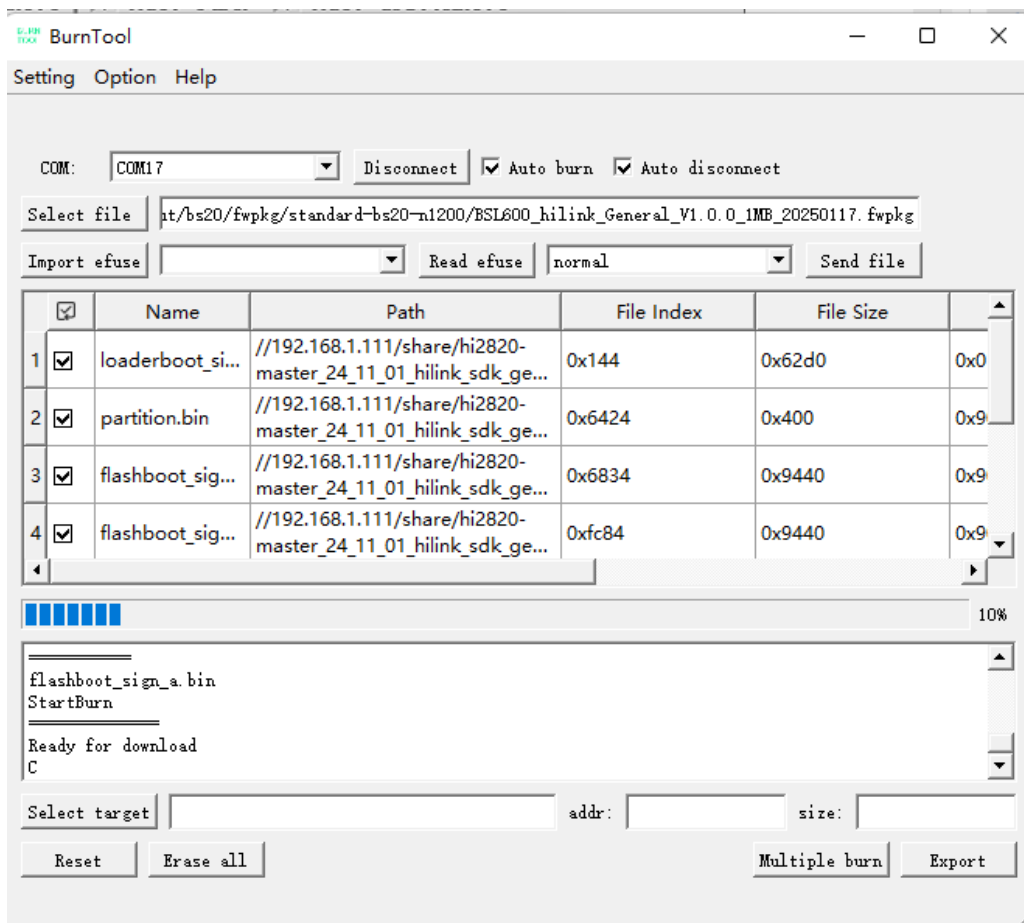
点击【Select file】选择固件，勾选【Auto burn】以及【Auto disconnect】选项，首次刷写选择 **erase all**（表明 flash 内容全擦后烧录，后续再刷的话可选择 **normal**，节约时间）。

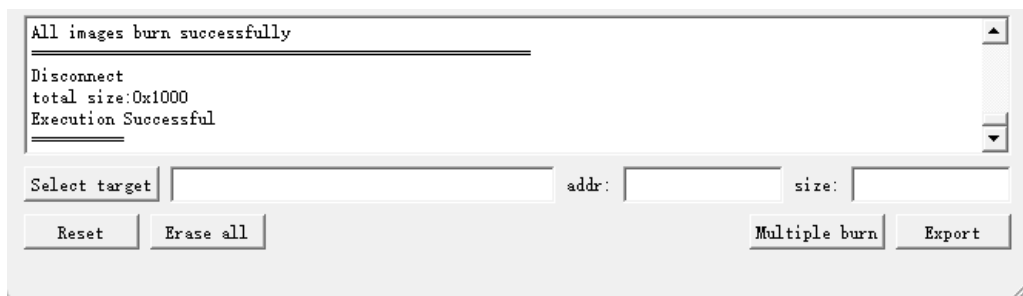


点击【Setting】->【Setting】设置波特率 1000000，可根据实际串口芯片能力，调整为其他更低的波特率。

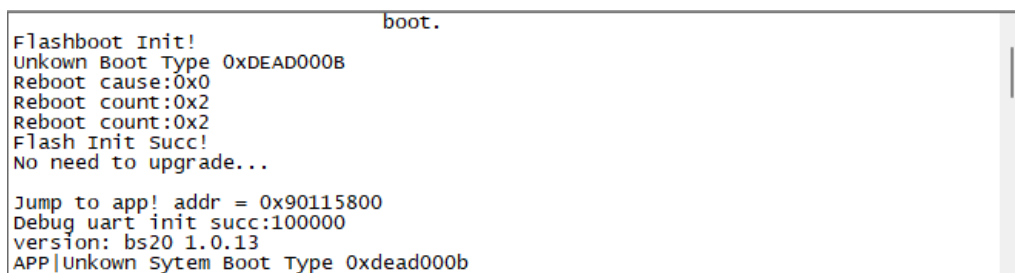


点击【Connect】，等待交互完成烧录，COM口由于有两个，如果一个不行，可切换尝试另外一路。

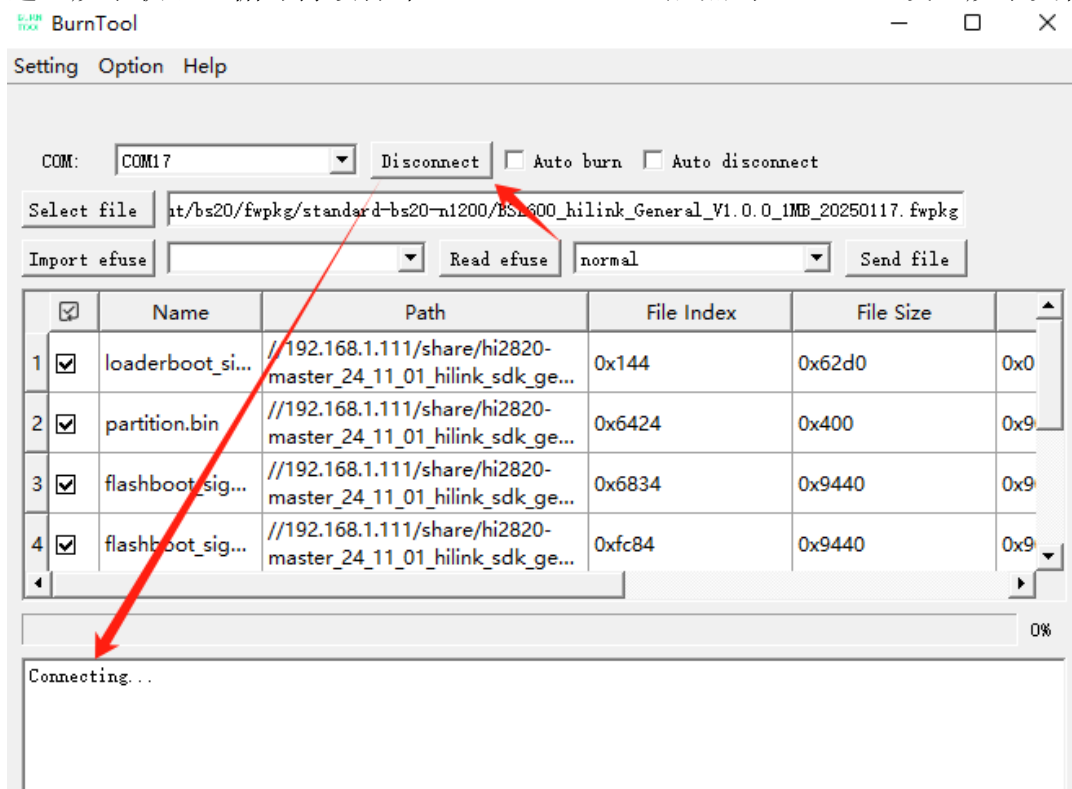




下载完成后，调试串口波特率 115200 下可以看到如下启动的日志



备注：也可以手动进行烧录，如下图，不勾选【auto burn】和【auto disconnect】点击 connect 之后工具显示连接设备中，此时按 reset 按键重启模块，进入烧录状态（循环持续打印 CCCCCC），之后点击 Send file 发送烧录文件即可。



2.5. 用户添加源代码文件

2.5.1. 用户函数定义约定

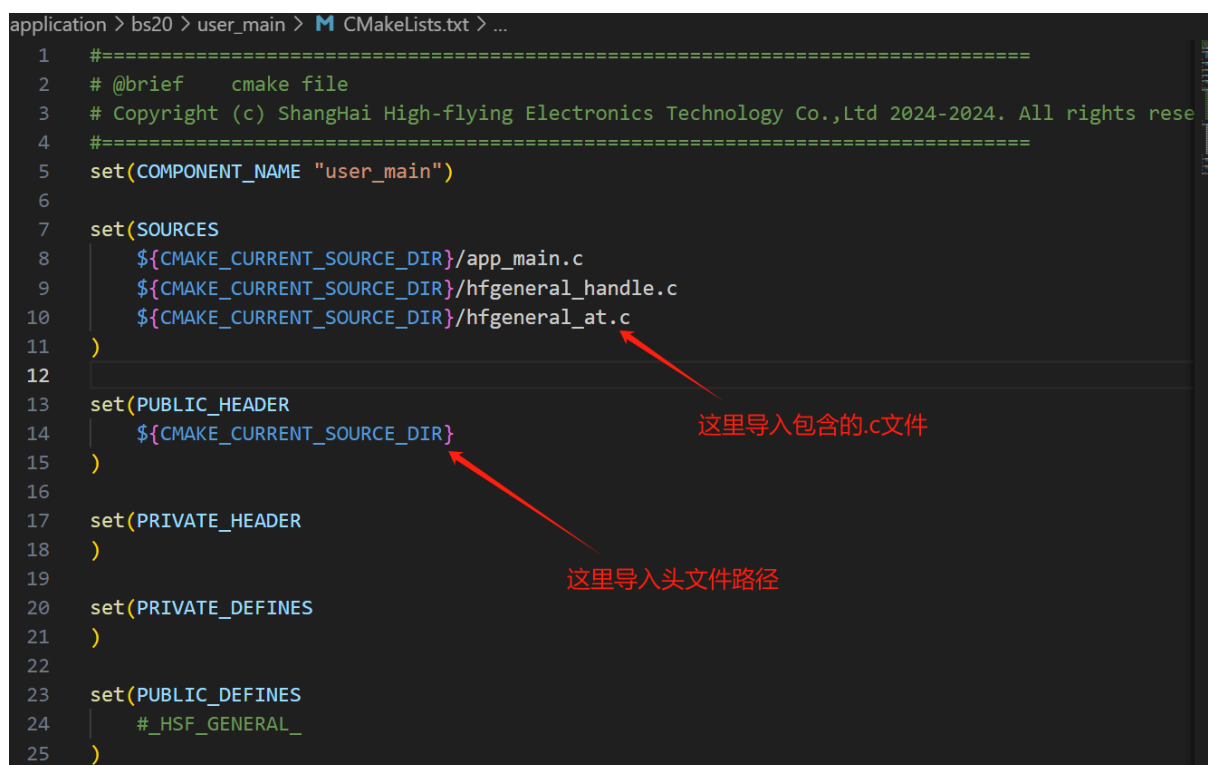
返回类型 + USER_FUNC+函数名称+参数

例如：void USER_FUNC test_func1(char *a); USER_FUNC 为函数修饰符号，为了更好的兼容性请加上 USER_FUNC 这个标识，如果不加在有的平台编译出来的程序将无法运行

2.5.2. 用户添加源代码文件

添加.c 文件，基于 HSF 的源文件都要包含<hsf.h>头文件，包含这个头文件后，源代码里面可以调用基于 HSF 的 API 函数；如果要使用 libc 接口函数，请#include 相关的头文件，例如如果调用字符串操作函数#include <string.h>,调用时间函数#include <time.h>等。

工程管理 Cmake 文件是在 application\bs20\user_main 文件夹下的 CMakeLists.txt 文件：



```
1  #=====
2  # @brief    cmake file
3  # Copyright (c) ShangHai High-flying Electronics Technology Co.,Ltd 2024-2024. All rights reserved
4  #=====
5  set(COMPONENT_NAME "user_main")
6
7  set(SOURCES
8      ${CMAKE_CURRENT_SOURCE_DIR}/app_main.c
9      ${CMAKE_CURRENT_SOURCE_DIR}/hfgeneral_handle.c
10     ${CMAKE_CURRENT_SOURCE_DIR}/hfgeneral_at.c
11 )
12
13 set(PUBLIC_HEADER
14     ${CMAKE_CURRENT_SOURCE_DIR}
15 )
16
17 set(PRIVATE_HEADER
18 )
19
20 set(PRIVATE_DEFINES
21 )
22
23 set(PUBLIC_DEFINES
24     #_HSF_GENERAL_
25 )
```

这里导入包含的.c文件

这里导入头文件路径

2.5.3. 用户添加第三方库

将三方库文件夹放在 open_source 文件夹下，如下图：

网络 > 192.168.1.111 > share > BSL600-HSF-1MB-Hilink > open_source >

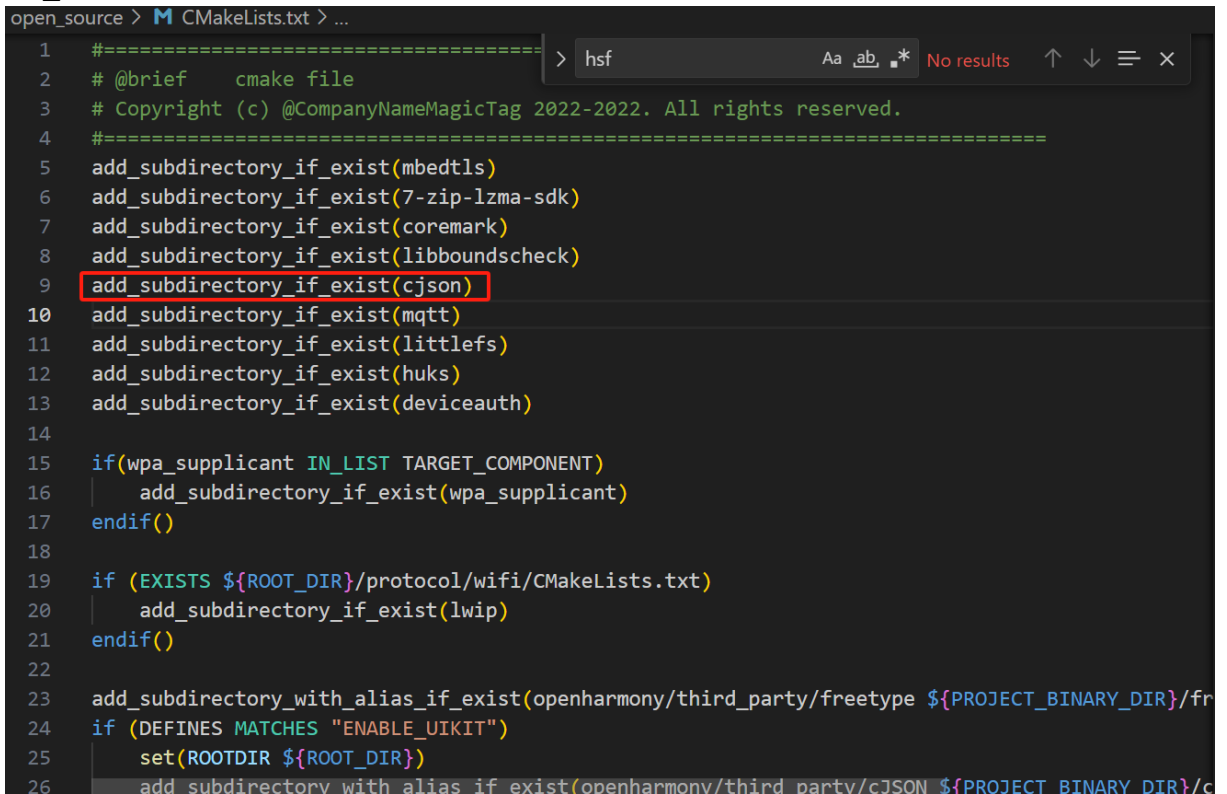
	名称	修改日期	类型	大小
	7-zip-lzma-sdk	2025/1/26 15:00	文件夹	
	cjson	2025/1/26 15:00	文件夹	
	GmSSL3.0	2025/1/26 15:00	文件夹	
	libboundscheck	2025/1/26 15:00	文件夹	
	mbedtls	2025/1/26 15:00	文件夹	
11200	CMakeLists.txt	2025/1/26 15:00	文本文档	3 KB
11200				

2.5.4. 用户添加编译文件及其头文件目录

将 cjson 的 CMakeLists.txt 文件拷贝到自己的第三方库文件夹下，根据下图修改组件名、添加编译文件、添加头文件目录：

```
open_source > cjson > CMakeLists.txt
1
2  set(COMPONENT_NAME "cjson")           修改cjson为第三方库名称或自定义也可以
3
4  set(CMAKE_CJSON_SOURCE_DIR
5      ${CMAKE_CURRENT_SOURCE_DIR})
6
7  set(SOURCES
8      ${CMAKE_CJSON_SOURCE_DIR}/cjson/cJSON.c
9      ${CMAKE_CJSON_SOURCE_DIR}/cjson/cJSON_Utils.c  添加需要编译的.c文件
10 )
11
12 set(PUBLIC_HEADER
13     ${CMAKE_CJSON_SOURCE_DIR}/cjson
14 )           添加头文件所在路径
15
16 set(COMPONENT_PUBLIC_CCFLAGS
17 )
18
19 set(COMPONENT_CCFLAGS
20     -Wno-error=logical-op
21     -Wno-error=sign-compare
22     -Wno-error=jump-misses-init
23     -Wno-sign-compare
24     -Wno-jump-misses-init
25     -Wno-error=unused-parameter
26     -Wno-unused-parameter
27     -Wno-unused-but-set-variable
28     -Wno-error=unused-variable
29 )
30
31 set(WHOLE_LINK
32     true
33 )
34
35 set(MAIN_COMPONENT
36     false
37 )
38
39 build_component()
40
41
```

在 open_source/CMakeLists.txt 文件中添加组件:

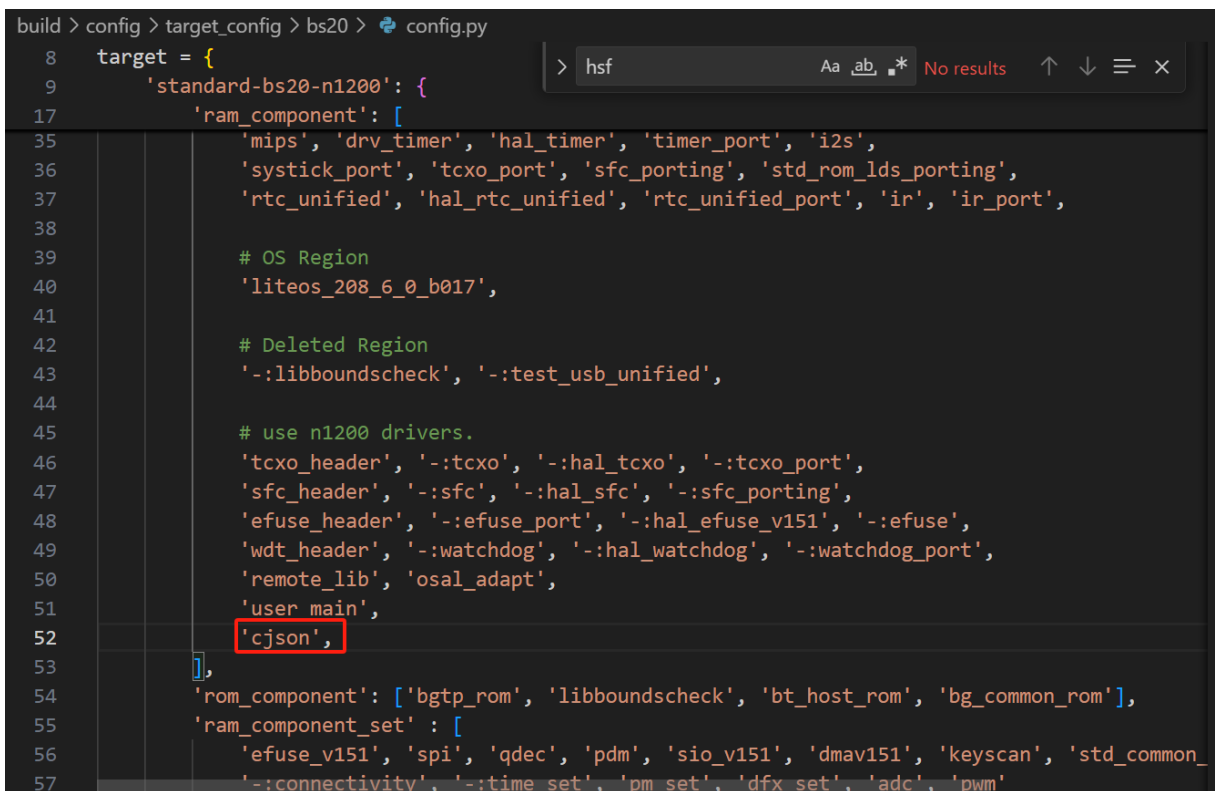


```

1  #=====
2  # @brief      cmake file
3  # Copyright (c) @CompanyNameMagicTag 2022-2022. All rights reserved.
4  #=====
5  add_subdirectory_if_exist(mbedtls)
6  add_subdirectory_if_exist(7-zip-lzma-sdk)
7  add_subdirectory_if_exist(coremark)
8  add_subdirectory_if_exist(libboundscheck)
9  add_subdirectory_if_exist(cjson)
10 add_subdirectory_if_exist(mqtt)
11 add_subdirectory_if_exist(littlefs)
12 add_subdirectory_if_exist(huks)
13 add_subdirectory_if_exist(deviceauth)
14
15 if(wpa_supplicant IN_LIST TARGET_COMPONENT)
16     add_subdirectory_if_exist(wpa_supplicant)
17 endif()
18
19 if (EXISTS ${ROOT_DIR}/protocol/wifi/CMakeLists.txt)
20     add_subdirectory_if_exist(lwip)
21 endif()
22
23 add_subdirectory_with_alias_if_exist(openharmony/third_party/freetype ${PROJECT_BINARY_DIR}/fr
24 if (DEFINES MATCHES "ENABLE_UKIT")
25     set(ROOTDIR ${ROOT_DIR})
26     add_subdirectory_with_alias_if_exist(openharmony/third_party/cJSON ${PROJECT_BINARY_DIR}/c

```

将组件名添加到 build/config/target_config/bs20/config.py 文件中:



```

8  target = {
9      'standard-bs20-n1200': {
10         'ram_component': [
11             'mips', 'drv_timer', 'hal_timer', 'timer_port', 'i2s',
12             'sys_tick_port', 'tcxo_port', 'sfc_porting', 'std_rom_lds_porting',
13             'rtc_unified', 'hal_rtc_unified', 'rtc_unified_port', 'ir', 'ir_port',
14
15             # OS Region
16             'liteos_208_6_0_b017',
17
18             # Deleted Region
19             '-:libboundscheck', '-:test_usb_unified',
20
21             # use n1200 drivers.
22             'tcxo_header', '-:tcxo', '-:hal_tcxo', '-:tcxo_port',
23             'sfc_header', '-:sfc', '-:hal_sfc', '-:sfc_porting',
24             'efuse_header', '-:efuse_port', '-:hal_efuse_v151', '-:efuse',
25             'wdt_header', '-:watchdog', '-:hal_watchdog', '-:watchdog_port',
26             'remote_lib', 'osal_adapt',
27             'user main',
28             'cjson',
29         ],
30         'rom_component': ['bgtp_rom', 'libboundscheck', 'bt_host_rom', 'bg_common_rom'],
31         'ram_component_set': [
32             'efuse_v151', 'spi', 'qdec', 'pdm', 'sio_v151', 'dmav151', 'keyscan', 'std_common',
33             '-:connectivity', '-:time_set', 'pm_set', 'dfx_set', 'adc', 'pwm'
34         ]
35     }
36 }

```

注意:

SDK 中已经集成 cjson、mbedtls 等第三方库，可以直接在工程中添加使用，使用方法参考 example 下的例子。

为了利于 SDK 升级，请不要在 `app_main.c` 文件里面添加太多代码，最好只需要添加一个入口函数。其它的源文件都放在自己的目录下面，可以在 `src` 目录下面建一个目录放置自己的源代码，这样升级的时候，只需要把 `app_main.c` 文件修改一下，其它的 SDK 文件全部用新的替换就可以了。

2.6. 开发注意事项

为避免影响到汉枫出厂测试而导致无法生产，在 SDK 开发过程中请注意以下几点：

- 必须保留汉枫的 AT 命令功能，尽量不要修改汉枫的原始 AT 命令名或与汉枫原始 AT 名称重复，
- 增加自定义查询命令 AT+APPVER 以便区分我司生产区分；

2.7. 常见问题分析

2.7.1. 线程崩溃问题

问题描述：有时候任务栈空间分配的不够大，会出现打印崩溃信息重启如下图

```
HIGHEST task ID: UART1_THREAD:7 SP error! StackPointer: 0x20006140 TopOfStack: 0x200061c0
CURRENT task ID: UART1_THREAD:7 stack overflow!
```

问题分析：上述问题一般是由程序中某个线程任务的栈溢出引起，打印中会指出溢出线程的名称和 id。可以通过 `OhosMemHeapSizePrint` 和 `print_heap_statistics_riscv()` 函数打印系统堆的任务栈的使用情况

```
o
[16:27:57:919]=====mem print=====v==
[16:27:57:920]    used    size [42588]
[16:27:57:982]current free size [18072]
[16:27:57:982] min free size [18072]
[16:27:58:044] reserve size [8192]
[16:27:58:106]task_id taskName      stackTop  stackLen  peakUsage  sp        peakRatio
[16:27:58:185]00    Swt_Task      0x20001220 0x000002C0 0x000001CC 0x20001390 65%
[16:27:58:247]01    IdleCore000 0x200014F0 0x00000600 0x00000284 0x20001890 41%
[16:27:58:309]02    app          0x20001CC0 0x00000C50 0x00000A0C 0x20002810 81%
[16:27:58:454]03    bt           0x20002930 0x00000800 0x000002D4 0x20002FA0 35%
[16:27:58:514]04    bt_service   0x20003150 0x00001000 0x00000684 0x20003FD0 40%
[16:27:58:576]05    schedule_loop 0x20004A00 0x00000A10 0x000006F4 0x200052B0 69%
[16:27:58:639]06    pm_sys       0x200055A0 0x00000800 0x00000174 0x20005C30 18%
[16:27:58:717]07    UART1_THREAD 0x200061C0 0x00000AF0 0x000002D4 0x20006B00 25%
[16:27:58:842]08    BtScheduleTask 0x200082F0 0x00000A00 0x000008B4 0x20008B10 87%
[16:27:58:906]09    UART_THREAD  0x20009150 0x00000AF0 0x00000194 0x20009AB0 14%
[16:27:58:984][SysHeap stat] total:0xecf4, used:0xa65c, current free:0x4698, peak usage:0x0, peak free:0xecf4
```

2.8. 资源分配

2.8.1. 1MB Flash 分区表

地址	描述
0x0000 0000	BOOT(81KB) Bootloader
0x0000B000	App CODE (564KB) 应用层序运行代码区
0x00098000	OTA UPGRADE(372KB) OTA升级区域
0x000F 5000	XOCTRIM区(4KB) 生产测试参数存储区
0x000F 6000	HILINK_PARA(8KB) Hilink参数存储
0x000F 8000	Product(4KB) 生产参数保存区域
0x000F 9000	USERPAGE (4KB) 用户参数保存区域
0x000F A000	USERPAGE_BACKUP (4KB) 用户参数保存备份区域
0x000F B000	F_SETTING (4KB) 出厂参数保存区域
0x000F C000	UFLASH (8KB) 用户flash, SDK开发数据 存储用
0x000F E000	NV (8KB) NV保存区域

其中 App code 区留给可用的空间（hilink release 版本）45KB，OTA 升级的文件区域是指 App ota 固件。

OTA 升级时把固件下载到 OTA 升级备份区，如果校验文件正确，内部设置标记位，下次重启的时候会把备份程序覆盖到 CODE 运行区并清除标记，完成固件更新动作，故升级过程中即使断电，也不会影响程序完成更新动作。

2.8.2. RAM 资源

BSL600 总共有 128KB RAM，留给用户可使用的 RAM 大约在 20K 左右

2.9. 串口打印调试信息

2.9.1. 串口调试说明

BSL600 有两路串口，UART0 对应 PIN38 和 PIN39，UART1 对应 PIN22 和 PIN23，两路串都可用于 AT 指令、透传数据和打印日志，默认程序中两路串口都启用，默认波特率都为 115200。

如果程序想通过串口打印调试信息，HSF 中提供了 `u_printf` 和 `HF_Debug` 两个 API 函数，默认情况下程序中调用这两个函数是不会有打印信息出来的，因为默认调试是关闭的，要通过 `hfdbg_set_level(X)` 打开调试串口输出，X 代表输出的调试信息等级，或者使用 `AT+NDBGL` 命令设置日志打印级别和输出串口，如

“`AT+NDBGL=2,1`”表示设置打印级别为 2，输出串口为 UART1。打开日志打印后重启会输出如下信息：

```
[16:34:30.767]收←◆Flashboot Init!
Power On
Reboot cause:0xF0F0
Reboot count:0x0
Reboot count:0x0
Flash Init Succ!
No need to upgrade...

Jump to app! addr = 0x90115800
Debug uart init succ:100000
version: bs20 1.0.13
APP[System Power On
APP[System Reboot cause:0xF0F0, total reboot count:0, exception reboot count:0
Sys FM: BUCK
kthread_create name:app
kthread_create name:bt
kthread_create name:bt_service
cpu 0 entering scheduler
LOS_QueueCreate queueId 0
LOS_QueueCreate queueId 1

[16:34:30.871]收←◆kthread_create name:schedule_loop
kthread_create name:pm_sys
[pm_sys]pm_sys_entry.
cumulative:1048319008,magic:1437226410
ua\0 謹業}スノ}U□IQ} UM]?
[UART] uart 1:115200, 8, 1, fc:0, parity:1
kthread_create name:UART1_THREAD
create Timer1 success
start Timer1 success
[read mac] read ret:0 lock efuse addr:79,lock:1
[read mac] read ret:0 mac efuse addr:116,mac:281106856959
[read mac] read ret:0 key efuse addr:96,key:633AE83D
read ble mac:281106856959
read ble mac key:633AE83D
BT SDK version: v14.2.0.305[date:2023-04-01 10:42:14 compiler:risov32-linux-musl-gcc]
[HILINK] HILINK_BT_GetDevInfo
HILINK_GetDeviceSn:281106856959
[BLE_HILINK_SERVER] InitBtStack enter.
[BLE_HILINK_SERVER] EnableBtStack enter.

[16:34:30.984]收←◆[ACore] ble enable cbk in, event:c
sdk version(1.0),the hfapp main start time is Jan 24 2025 16:40:49
uapi_uart_init_UART_BUS_2
[UART] uart 0:115200, 8, 1, fc:0, parity:1
kthread_create name:UART_THREAD
```

2.9.2. 串口 log 输出说明

可以淘宝购买 USB 转 TTL 串口，把调试串口上飞线到 USB 串口上，需要接 GND 和调试串口（具体调试串口引脚详见模块手册定义），也可以使用我们的开发板

• 含3个状态灯 • TTL电平5V/3V3可
• 支持Windows98/ME/2000/XP/Server 2003/VISTA
Server 2008/Win7/Win8 32位/64位



USB转TTL USB转串口下载线CH340G模块RS232升级板刷电路板线
PL2303

土豪金CH340GUSB转TTL

价格 ¥5.87

促销价 **¥4.97** 优惠促销

本店活动 满88元,包邮

更多优惠

运费 广东深圳 至 上海 快递: 0.00

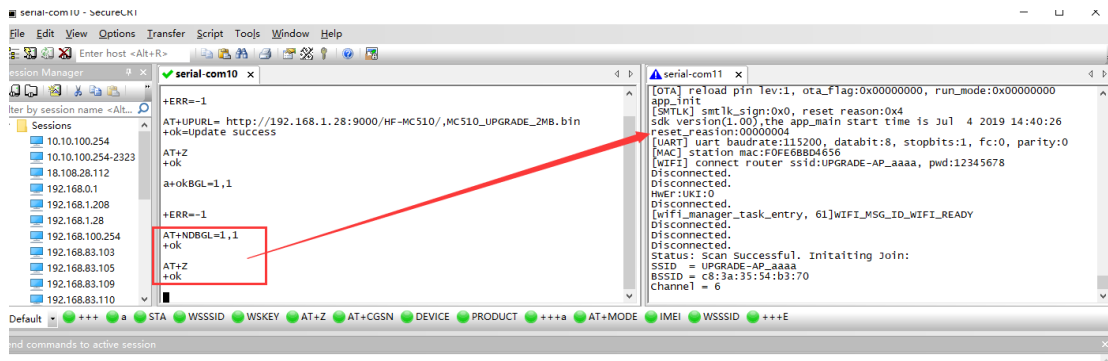
月销量 3179

累计评价 6388

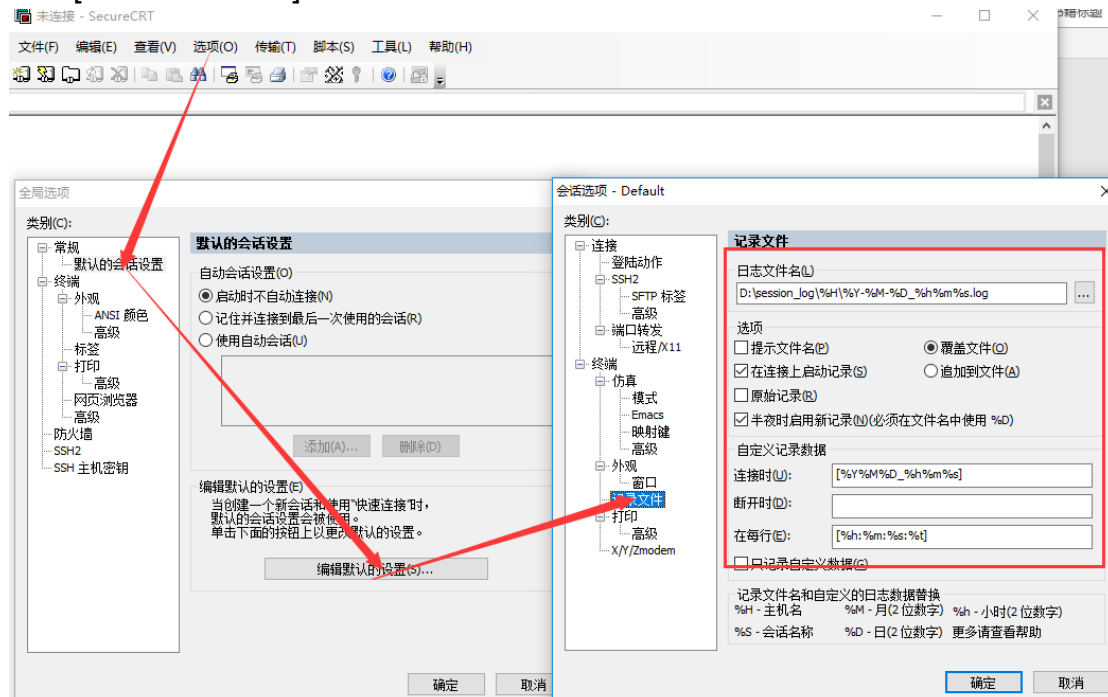
颜色分类



这样双串口可以同时使用，通讯串口用于功能测试，调试串口用于打印日志。



按下图设置 SecureCRT 软件的日志记录功能。
D:\session_log\%H\%Y-%M-%D_%h%m%s.log
[%Y%M%D_%h%m%s]
[%h:%m:%s:t]



这样串口有打印的情况下可自动在对应目录生成文件。

3. Hilink 开发指导

3.1. 产品信息修改

产品开发 > 专用BSL600认证测试

专用BSL600认证测试

开发中

导出产品json文件

ProdID: 2P3W | 品牌: 汉枫 | 系列: | 产品型号: BSL600 | 品类: 风扇 | 软件版本号: | ProdKey: 15145fe44 | 方案: HarmonyOS Connect最小硬件方案 (BLE)

联系人

创建时间: 2024-10-31 12:02:22 | 最近更新: 2024-10-31 12:02:42

产品定义

交互设计

集成开发

自测试

产品定义

物模型定义

蓝牙信道配置

软硬件规格

集成能力

极简交互: 靠近发现(蓝牙)

万能卡片: 智慧生活APP

操作系统

LiteOS

请确保信息准确无误, 华为颁发证书上的字段与您填写信息一致

操作系统版本号

1.0.0

请确保信息准确无误, 华为颁发证书上的字段与您填写信息一致

证书类型

无

通信模组/芯片

汉枫

模组型号: BSL600 | 芯片型号: i2821

操作系统: LiteOS | RAM: (Kibibytes) : 128

申请认证等级

B

代理注册 (华为开放协议)

基本信息

帐号

dao***@****ying.com

企业名称

上海汉枫电子科技有限公司

企业中文名称简称

上海汉枫电子科技有限公司

企业英文名称简称

HighFlying

ManufacturerID

01C

联系邮箱

daoshi@hi-flying.com

已加入的合作计划

HarmonyOS Connect生态解决方案合作伙伴

生态解决方案合作伙伴是指具备智能硬件解决方案开发、模组生产或芯片设计等服务能力, 并通过华为...

已加入

HarmonyOS Connect生态产品合作伙伴

生态产品合作伙伴是基于HarmonyOS Connect技术完成智能硬件产品的开发、认证、量产和推广的合...

已加入

HarmonyOS Connect服务合作伙伴

服务合作伙伴可以通过华为智慧场景联动业务平台, 基于Harmony OS Connect生态产品和丰富的应用...

已加入

3.1.1. 产品信息对应

```

{
  "acKey": "734A684E3B772A4233746D503C6F2E6040CA742725B71A7241CDABD404557DC42B327EB239B1DC2846BB6A77863DD8E3",
  "accessType": "HARD",
  "brand": "汉枫",
  "brandEn": "hiflying",
  "desc": "firmwareVer:固件软件版本号; softwareVer:SDK版本号; hardwareVer:设备硬件版本号",
  "deviceModel": "BSL600",
  "deviceName": "专用BSL600认证测试",
  "deviceTypeId": "029",
  "deviceTypeName": "风扇",
  "deviceTypeNameEn": "Fan",
  "firmwareVer": null,
  "hardwareVer": "V1.0.0",
  "manufacturerId": "01C",
  "manufacturerName": "上海汉枫电子科技有限公司",
  "manufacturerNameEn": "hiflying",
  "os": "LiteOS",
  "osVersion": "1.0.0",
  "prodId": "2P3W",
  "prodKey": "15145efea46d4e188aeccc35f37003e4",
  "productSeries": "",
  "protocolType": "BLUETOOTH_BLE",
  "softwareVer": "未涉及"
}

```

3.1.2. 产品信息修改

修改文件位置:

application\samples\hilink\ohos_connect\hilink_adapt\entry
/hilink_ble_main

ProdID:

■ #define ProductId "2P3W"

产品型号:

■ #define DEVICE_MODEL "BSL600"

DeviceTypeID:

■ #define DEVICE_TYPE "029"

企业英文名简称在配网 ble 名称中显示:

■ #define MANUFACTURER "hiflying"

deviceTypeNameEn

■ #define DEVICE_TYPE_NAME "Fan"

企业英文名简称

■ #define MANUFACTURER_NAME "hiflying"

deviceManuId

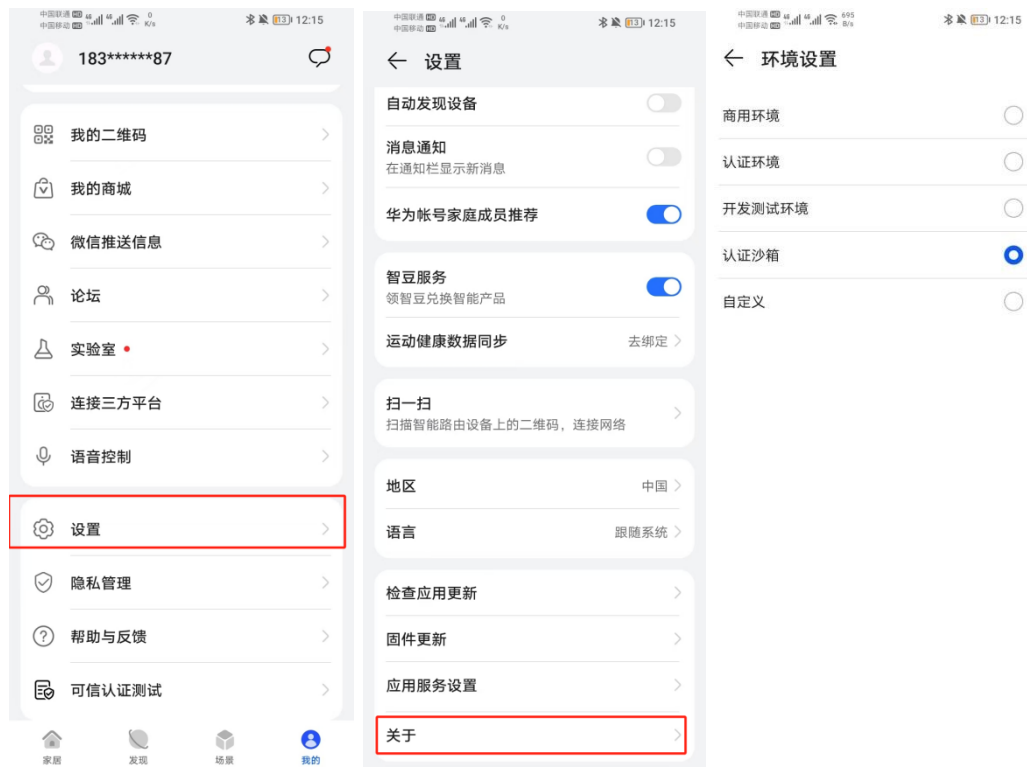
■ #define DEVICE_MANU_ID "01C"

3.1.3. BLE 配网添加设备

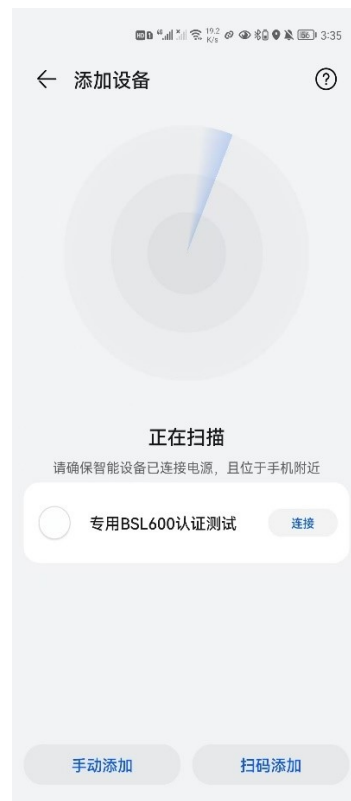
产品后台下载测试用 APK。



切换到认证沙箱环境



默认固件启动后就处于 ble 配网状态
按如下流程添加好设备。



按开关键，即可看到如下的 log 控制信息。



```
[17:00:29.225]收←◆[BLE_HILINK_SERVER] write_request_cb_adapt—request_id:9 att_handle:22
data_len:160
start process bt data, data len:160

[17:00:29.325]收←◆[BLE_HILINK_SERVER] write_request_cb_adapt—request_id:10 att_handle:22
data_len:18
start process bt data, data len:18
custom data, len:32, data:{"sid":"switch","data":{"on":1}}

[17:00:29.380]收←◆send response ok
schedule event:22
[BLE_HILINK_SERVER] BleGattsSendIndication enter, handle:22.
[BLE_HILINK_SERVER] indicationSentCb form Hilink.
schedule event:22
[BLE_HILINK_SERVER] BleGattsSendIndication enter, handle:22.
[BLE_HILINK_SERVER] indicationSentCb form Hilink.
schedule event:20
schedule event:20

[17:00:29.521]收←◆[BLE_HILINK_SERVER] indicate_confirm_cb server_id:1 conn_id:0 status: 0
```

3.2. 状态和收发数据 API 接口介绍

重置配网: [HILINK_BT_HardRevoke](#)

APP 下发数据回调函数: [BleRcvCustomData](#)

设备状态上报 (REPORT): [HILINK_ReportSidState](#)

BLE 辅助配网状态回调函数: [BleCfgNetProcess](#)

BLE 配网状态获取函数: [HILINK_BT_GetRegState](#)

3.3. app_main.c 程序说明

3.3.1. hf_bsl_600_gpio_fid_to_pid_map_table

引脚功能定义, [HFGPIO_F_NLINK](#)、[HFGPIO_F_WAKEUP](#)、[HFGPIO_F_MANU](#)(Reload 配网解绑按键)、[HFGPIO_F_BCTS](#) 是汉枫定义的功能码, 功能码是表示模块用户手册提到的如下功能。

Link↵	nLink 指示灯, 用作联网指示灯, 低电平有效↵ 亮: 连接服务器成功↵ 灭: 未连接服务器↵ ↵
Reload↵	长按 3 秒松开后进入重置功能, 解绑并重新配网功能↵
Reset↵	复重启按键↵
BCTS↵	默认输出低电平, 有数据发送时输出高电平, 用于唤醒 MCU, 数据发送完毕后输出低电平 ↵
WAKEUP↵	拉低唤醒蓝牙, 不需要休眠的话请接下拉到 GND, 内有上拉电阻↵

对功能码做了如下的映射之后，响应的 GPIO 号就具备了对应的功能。

```
BSL600_GPIO15, //HFGPIO_F_NLINK
BSL600_GPIO30, //HFGPIO_F_WAKEUP
BSL600_GPIO14, //HFGPIO_F_MANU
BSL600_GPIO13, //HFGPIO_F_BCTS
```

可以对功能码定义进行修改，使得用户可以自由操作控制 GPIO13、GPIO14、GPIO15，BSL600_GPIO30，详细可参见 `samples` 目录中的外设使用参考代码

```
NOPIN,    // HFGPIO_F_NLINK
NOPIN,    // HFGPIO_F_WAKEUP
NOPIN,    // HFGPIO_F_MANU
NOPIN,    // HFGPIO_F_BCTS
```

3.3.2. user_define_at_cmds_table

用户自定义 AT 指令，如下例子自定义了 AT+APPVER 查询版本号的功能。

```
#define ZJ_LVER      "1_Sam (2024/05/17)"
int HSF_IAP hf_atcmd_zjver(pat_session_t s,int argc,char *argv[],char *rsp,int len);
const hfat_cmd_t user_define_at_cmds_table[]=
{
    {"APPVER", hf_atcmd_zjver, NULL, NULL},
    {NULL,NULL,NULL,NULL} //the last item must be null
};
int HSF_IAP hf_atcmd_zjver(pat_session_t s,int argc,char *argv[],char *rsp,int len)
{
    if(0 != argc)
        return -4;
    sprintf(rsp, "%s", ZJ_LVER);
    return 0;
}
```

3.3.3. uart_recv_callback

串口接收回调

HFNET_UART0_DATA_READY: 通讯串口接收数据的事件。

HFNET_UART1_DATA_READY: Debug 调试串口接收数据的事件，如果两路串口都想使用，可以启用 `hfnet_start_uart_ex(HFUART1...)` API，这样两路串口可同时供数据传输使用，UART1 也支持 AT 指令。

HFSYS_STATE_RUN_CMD: 处于 AT 命令状态，一般汉枫产测使用。

3.3.4. app_main

用户主程序入口，创建线程、定时器等。

3.4. 例子

3.4.1. PWM 接口

参考 `application\samples\peripheral\pwm\pwm_demo.c` 了解 GPIO 的 PWM 输出功能。

3.4.2. ADC 功能

参考 `application\samples\peripheral\adc\adc_demo_inc.c` 中示例

3.4.3. 线程、定时器接口

参考 `application\samples\peripheral\timer` 通过例子可以了解线程创建和定时器等功能。

4. AT 指令说明

4.1. 模块运行模式配置

模块上电后，进入默认的模式即透传模式，用户可以通过串口命令把模块切换到命令行模式。模块的缺省 UART 口参数配置如下：



在命令行模式下，用户可以通过 AT+ 指令利用 UART 口对模块进行设置。

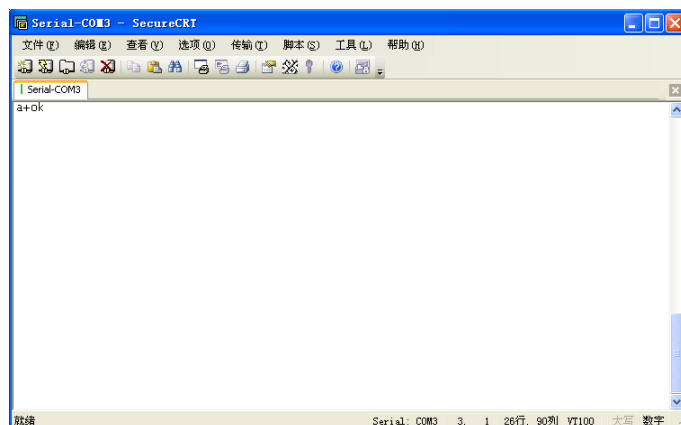
注意：SDK 这个机制与鸿蒙通用固件不同，通用固件默认是命令模式，部分 AT 命令在 SDK 中也有所删减，SDK 支持的 AT 命令以本文档为准

4.1.1. 从透传模式切换到命令模式

从透传模式切换到命令模式分 2 个步骤：

在串口上输入“+++”，模块在收到“+++”后会返回一个确认码“a”；

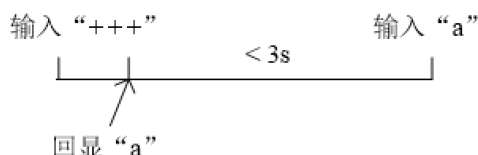
在串口上输入确认码“a”，模块收到确认码后，返回“+ok”确认，进入命令模式；



<说明>:

在输入“+++”和确认码“a”时，串口没有回显，如上图所示。

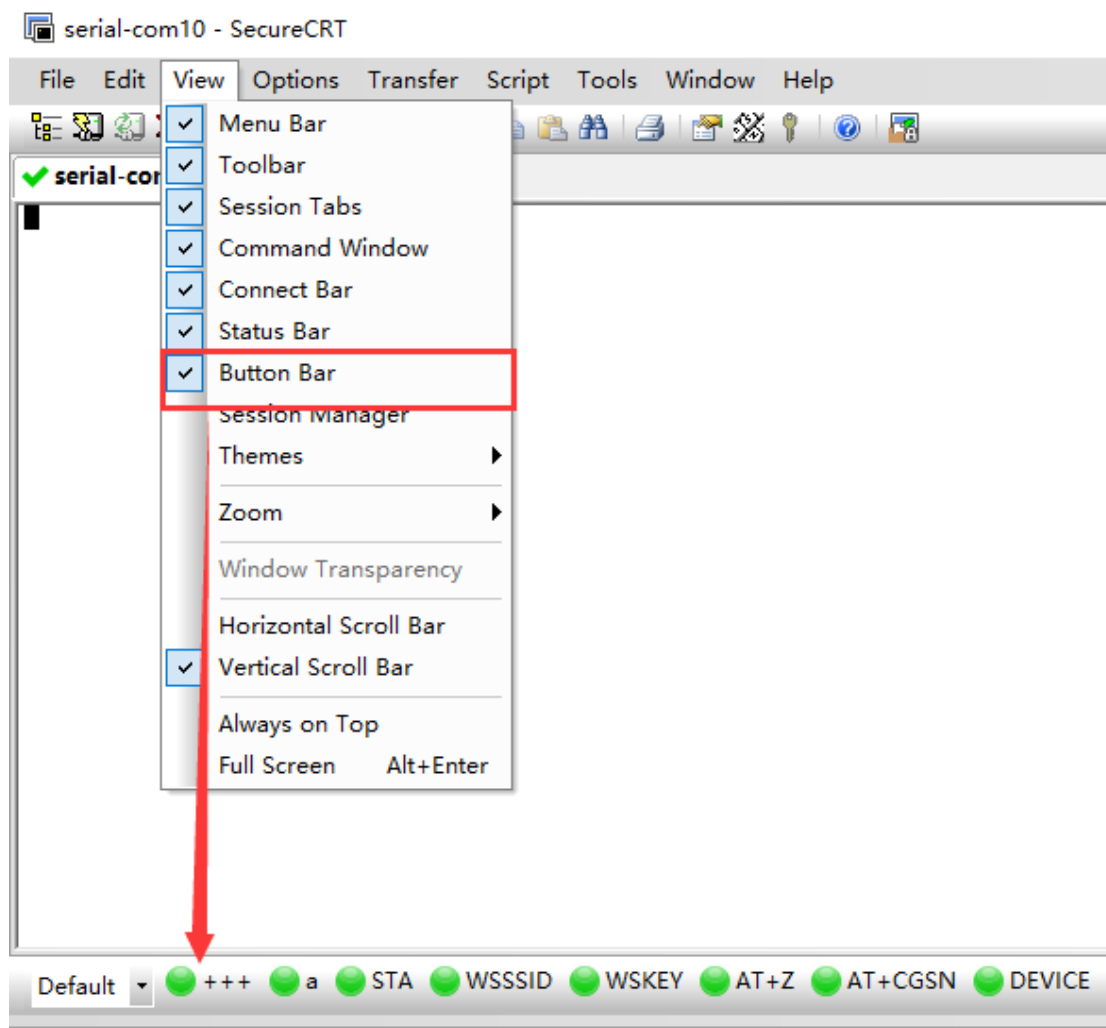
输入“+++”和“a”需要在一定时间内完成，以减少正常工作时误进入命令模式的概率。具体要求如下：



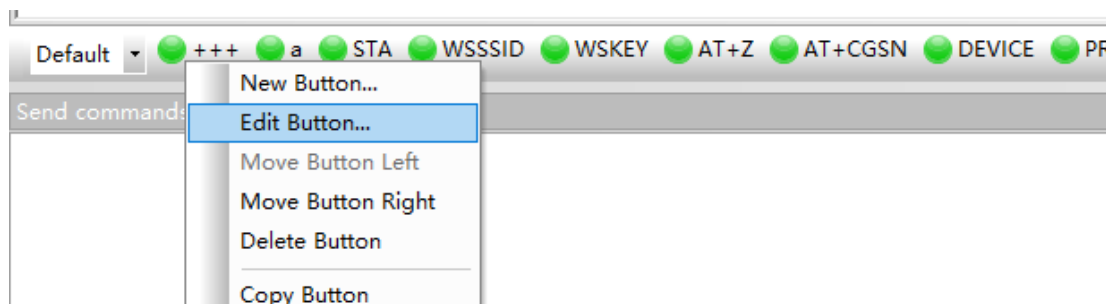
在命令模式下，可以通过 UART 口下 AT+指令对模块进行设置或查询，以及重启等操作，也可以通过 AT+指令回到透传模式。AT+指令具体见下一章节。

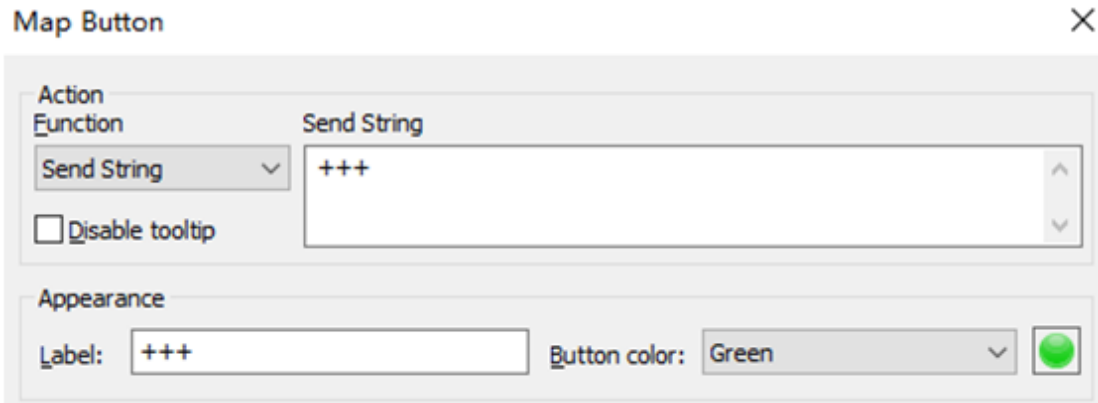
4.1.2. SecureCRT 软件使用

打开工具按钮功能。

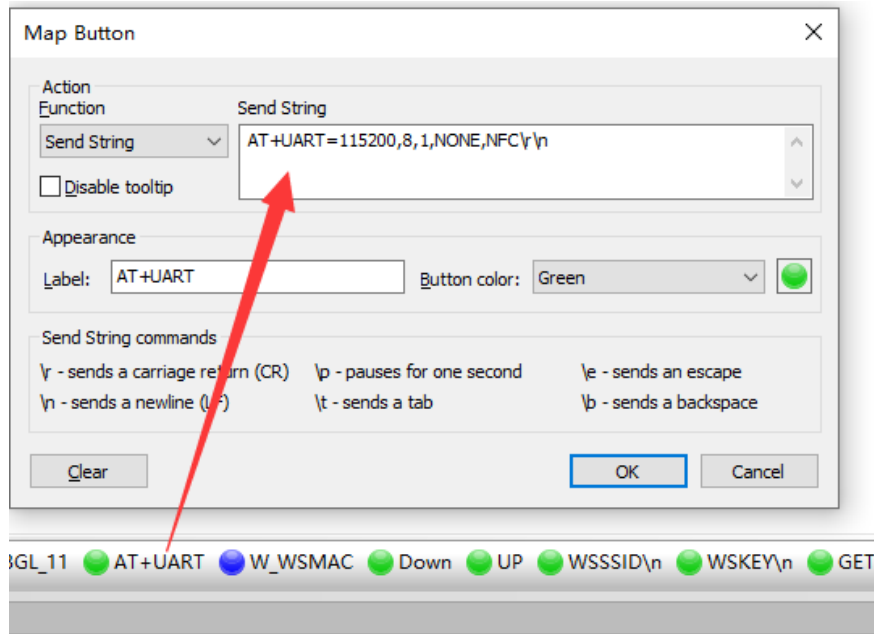


右键编辑按钮功能打开工具按钮功能，可添加常用的发送数据，此处添加进入命令模式所需要的“+++”和“a”。

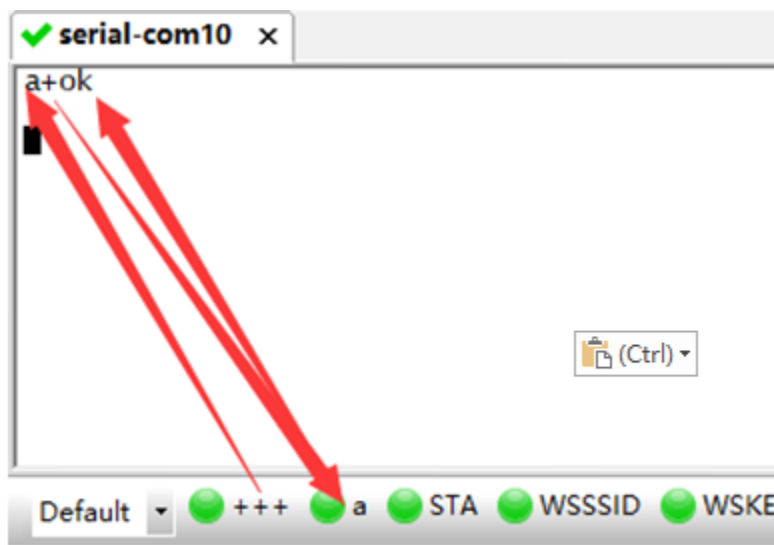




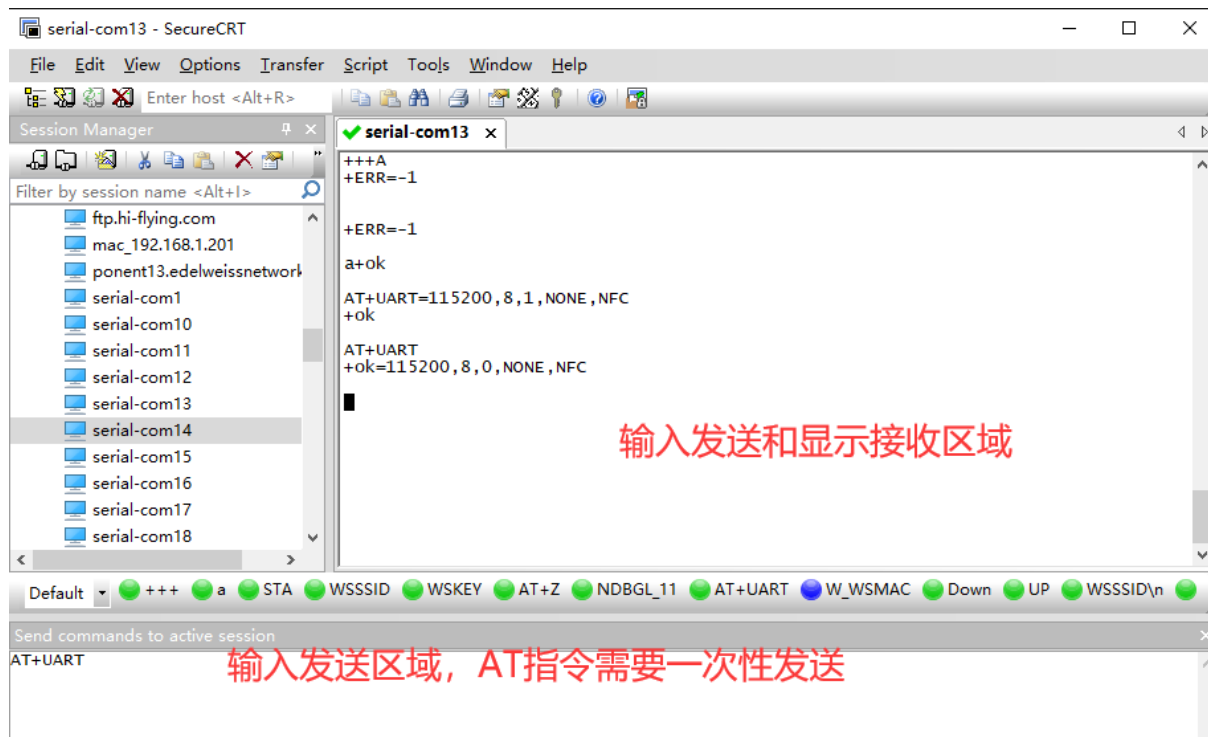
也可以编辑添加常用的 AT 指令，指令以\r 作为结束符。



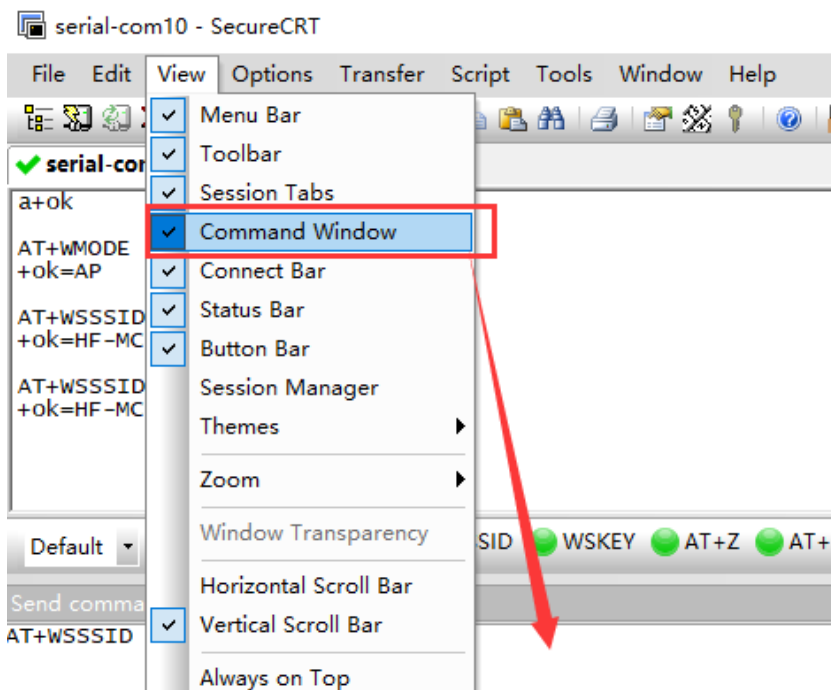
点击对应按钮，发送“+++”，模块回复“a”之后再点击发送“a”，模块回复“+ok”之后即进入命令模式。



进入命令模式之后即可对模块发送指令，主界面输入的数据会立即发送出去，显示的是回显接收到的数据，下方命令栏的数据只有按回车键之后才会发送出去（下方命令栏需要工具使能之后才会显示），命令栏发送的数据带了回车符号，所以无需像按钮栏那样额外添加“\r”。



显示命令栏。



4.1.3. SSCOM 软件使用

进入命令模式后发 AT 命令时，需要带回车换行。

```
[18:23:08.857]发->◇AT+REPORT=switch, {"on":0}
[18:23:08.884]收<-◆+ok=success
```



注意：从透传进入命令模式发+++和 a 的过程中不能有此回车换行

4.2. AT+指令集概述

4.2.1. 命令格式

AT+指令采用基于 ASCII 码的命令行，指令的格式如下：

➤ 格式说明

- <>: 表示必须包含的部分
- []: 表示可选的部分

➤ 命令消息

AT+<CMD>[op][para-1, para-2, para-3, para-4...]<CR>

- AT+: 命令消息前缀；
- CMD: 指令字符串；
- [op]: 指令操作符，指定是参数设置或查询；
 - ◆ “=”: 表示参数设置
 - ◆ “NULL”: 表示查询
- [para-n]: 参数设置时的输入，如查询则不需要；
- <CR>: 结束符，回车，ASCII 码 0x0a 或 0x0d；

<说明>:

输入命令时，“AT+<CMD>”字符自动回显成大写，参数部分保持不变。

➤ 响应消息

+<RSP>[op] [para-1, para-2, para-3, para-4...]<CR><LF><CR><LF>

- +: 响应消息前缀；
- RSP: 响应字符串，包括：
 - ◆ “ok”: 表示成功
 - ◆ “ERR”: 表示失败
- [op]: =
- [para-n]: 查询时返回参数或出错时错误码
- <CR>: ASCII 码 0x0d；
- <LF>: ASCII 码 0x0a；

➤ 错误码

Table2. 错误码列表

错误码	说明
-1	无效的命令格式
-2	无效的命令
-3	无效的操作符
-4	无效的参数
-5	操作不允许

4.2.1.1. AT+WSMAC: 设置/查询模块的 STA MAC 地址参数，复位后设置生效

■ 格式:

◆ 查询

```
AT+WSMAC<CR>
```

```
+ok=<mac_address><CR><LF><CR><LF>
```

◆ 设置

```
AT+WSMAC=<mac_address, key><CR>
```

```
+ok<CR><LF><CR><LF>
```

■ 参数:

◆ mac_address: 模块的 MAC 地址。如: ACCF23FF1234

◆ key: 加密 key, 汉枫特殊加密信息, 不允许随意修改 MAC 地址。

4.2.1.2. AT+SLEMAC: 查询模块的星闪 MAC 地址参数

■ 格式:

◆ 查询

```
AT+SLEMAC<CR>
```

```
+ok=<mac_address><CR><LF><CR><LF>
```

◆ 设置

```
AT+SLEMAC=<mac_address><CR>
```

```
+ok<CR><LF><CR><LF>
```

■ 参数:

◆ mac_address: 模块的星闪 MAC 地址。如: 0000002600DD

星闪 MAC 地址是存在 efuse 的, 如果已经写过, 再写入不同的 MAC 会报错。

4.2.1.3. AT+Z: 重启模块

■ 格式:

```
AT+Z<CR>
```

4.2.1.4. AT+UART: 设置或查询串口操作, 复位后设置生效

■ 格式:

◆ 查询:

AT+UART[=uart_num]<CR>

+ok=<baudrate, data_bits, stop_bit, parity, flowctrl><CR><LF><CR><LF>

◆ 设置:

AT+UART=<baudrate, data_bits, stop_bit, parity, flowctrl>[, uart_num]<CR>

+ok<CR><LF><CR><LF>

■ 参数:

◆ uart_num: 可选串口号, 默认串口 0.

✧ 0: 串口通道 0

◆ baudrate: 波特率

✧ 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 380400, 460800, 921600

◆ data_bits: 数据位

✧ 8

◆ stop_bits: 停止位

✧ 1

◆ parity: 检验位

✧ NONE (无检验位)

✧ EVEN (偶检验)

✧ ODD (奇检验)

◆ flowctrl: 硬件流控 (CTSRTS)

✧ NFC: 无硬件流控

4.2.1.5. AT+VER: 查询 SDK 版本号

■ 格式:

◆ 查询

AT+VER<CR>

+ok=<ver><CR><LF><CR><LF>

■ 参数:

◆ ver: 版本号包括 SDK 版本。

```
AT+VER
+ok=1.0.0 (2024-12-20 16:00 4M)
```

4.2.1.6. AT+SN: 设置/查询 Hilink SN 值

■ 格式:

◆ 查询

AT+SN<CR>

+ok=<sn><CR><LF><CR><LF>

◆ 设置

AT+SN=<sn><CR>

+ok<CR><LF><CR><LF>

■ 参数:

- ◆ sn: Hilink SN 值, 默认值: 使用模块 MAC 地址作为 SN, 有需要的话可以手动设置 SN 值。

```
AT+SN
+ok=ACCF23FF8888

AT+SN=HF123456
+ok

AT+SN
+ok=HF123456
```

4.2.1.7. AT+PRODUCT: 设置/查询 产测模式

■ 格式:

- ◆ 查询

AT+PRODUCT<CR>

+ok=<mode><CR><LF><CR><LF>

- ◆ 设置

AT+PRODUCT=<mode><CR>

+ok<CR><LF><CR><LF>

■ 参数:

- ◆ mode:

- ✧ 1: 进入产测模式设置后立即重启。
- ✧ 0: 关闭产测模式

4.2.1.8. AT+RESET: 重置模块, 恢复成配网状态并重启。

■ 格式:

- ◆ 查询

AT+RESET<CR>

+ok<CR><LF><CR><LF>

■ 参数:

- ◆ 无

4.2.1.9. AT+SLE: 设置/查询 星闪开关

■ 格式:

- ◆ 查询

AT+SLE<CR>

+ok=< status ><CR><LF><CR><LF>

- ◆ 设置

AT+PRODUCT=< status ><CR>

+ok<CR><LF><CR><LF>

■ 参数:

- ◆ status:
 - ✧ on: 打开星闪(默认打开)
 - ✧ off: 关闭星闪

4.2.1.10. AT+MODE: 设置/查询 hilink 通讯协议类型

- 格式:
 - ◆ 查询


```
AT+MODE<CR>
+ok=< mode><CR><LF><CR><LF>
```
 - ◆ 设置


```
AT+MODE=< mode ><CR>
+ok<CR><LF><CR><LF>
```
- 参数:
 - ◆ status:
 - ✧ 4: ble 协议
 - ✧ 16: ble sle 共存协议(默认)

4.2.1.11. AT+BLEADP: 配置/查询 BLE 广播参数

- 格式:
 - ◆ 查询


```
AT+BLEADP<CR>
+ok=<value><CR>< LF><CR>< LF>
```
 - ◆ 设置


```
AT+BLEADP=<value><CR>
+ok<CR>< LF><CR>< LF>
```
- 参数:
 - ◆ value: IntervalMin+IntervalMax+Type+Channel, 格式: AAAABBBBCD, 默认值 0032006407
 - ✧ IntervalMin: 最小间隔, AAAA 位置十进制值, 0032 x 0.625ms=20ms,
 - ✧ IntervalMax: 最大间隔, BBBB 位置十进制值, 1280 x 0.625ms=40ms
 - ✧ Type: 0 或者 1, 0 代表 public, 1 代表 random
 - ✧ Channel: 广播信道, 3bit 的 HEX 值, bit0 代表 channel37, bit1 代表 channel38, bit2 代表 channel39, 值为 7 即表示使用 37、38、39 信道

4.2.1.12. AT+SLEADP: 配置/查询休眠状态时 SLE 广播参数

- 格式:
 - ◆ 查询


```
AT+SLEADP<CR>
+ok=<value><CR>< LF><CR>< LF>
```

◆ 设置

AT+BLEADP=<value><CR>

+ok<CR>< LF><CR>< LF>

■ 参数:

- ◆ value: IntervalMin+IntervalMax+Type+Channel, 格式: AAAAABBBBBBCD, 默认值

001600032007

✧ IntervalMin: 最小间隔, AAAAA 位置十进制值, 00160x 0.125ms=20ms,

✧ IntervalMax: 最大间隔, BBBBB 位置十进制值, 00320x 0.625ms=40ms

✧ Type: 0 或者 1, 0 代表 public, 1 代表 random

✧ Channel: 广播信道, 3bit 的 HEX 值, bit0 代表 channel37, bit1 代表

channel38, bit2 代表 channel39, 值为 7 即表示使用 37、38、39 信

© Copyright High-Flying, April, 2023

The information disclosed herein is proprietary to High-Flying and is not to be used by or disclosed to unauthorized persons without the written consent of High-Flying. The recipient of this document shall respect the security status of the information.

The master of this document is stored on an electronic database and is “write-protected” and may be altered only by authorized persons at High-Flying. Viewing of the master document electronically on electronic database ensures access to the current issue. Any other copies must be regarded as uncontrolled copies.

附录 A: 汉枫联系方式

网址: www.hi-flying.com

在线资询: [400-189-3108/18616078755](tel:400-189-3108/18616078755)

支持邮箱: Support@hi-flying.com

<结束>