

# HF-LPX70 Series Wi-Fi&BLE Module

## SDK Manual

V 1.3

This doc is applicable for following products, refer to module manual for detailed hardware description.

|   |  |
|---|--|
|    |    |
| <p><b>HF-LPT270</b><br/>SMT18: 22.5 x 13.5 x 3.5mm</p>                              | <p><b>HF-LPT170</b><br/>DIP10: 22 x 15.6 x 8mm</p>                                   |
|   |   |
| <p><b>HF-LPT271</b><br/>SMT16: 24 x 16 x 3mm</p>                                    | <p><b>HF-LPT272</b><br/>Finger8: 16.7 x 15 x 2.2mm</p>                               |
|  |  |
| <p><b>HF-LPB170</b><br/>SMT34: 23.1 x 32.8 x 3.5mm</p>                              | <p><b>HF-LPB175</b><br/>XH2.54-4P: 41.3 x 24.1 x 6mm</p>                             |
|  |  |
| <p><b>HF-LPT570</b><br/>SMT14: 30mm x 17.7mm x 3.6mm</p>                            |  |

## Catalog

|  |           |
|--|-----------|
| <b>1. DESCRIPTION</b>                                      | <b>5</b>  |
| 1.1. HF-LPX70 SERIES EVK                                   | 5         |
| <b>2. COMPILE ENVIRONMENT</b>                              | <b>7</b>  |
| 2.1. LINUX COMPILE ENVIRONMENT                             | 7         |
| 2.1.1. <i>Download All in One Virtual OS</i>               | 7         |
| 2.1.2. <i>Install Compile Environment</i>                  | 9         |
| 2.1.3. <i>Set Shared Directory</i>                         | 10        |
| 2.1.4. <i>Compile Note</i>                                 | 11        |
| 2.2. DOWNLOAD SDK  | 11        |
| 2.3. MAKEFILE  | 13        |
| 2.4. COMPILE   | 13        |
| 2.5. FIRMWARE PROGRAM                                      | 15        |
| 2.6. SAMBA SHARED CODE                                     | 16        |
| 2.7. USER CODE   | 16        |
| 2.8. DEVELOPMENT NOTE                                      | 18        |
| 2.9. CODING TOOLS  | 19        |
| 2.10. Q&A  | 19        |
| 2.10.1. <i>Exception problem</i>                           | 19        |
| 2.10.2. <i>Stack overflow problem</i>                      | 20        |
| 2.10.3. <i>RAM overflowed problem</i>                      | 20        |
| 2.10.4. <i>FLASH overflowed problem</i>                    | 20        |
| 2.10.5. <i>Backtrace</i>                                   | 20        |
| 2.11. CODE SIZE CUT  | 21        |
| <b>3. RESOURCES DESCRIPTION</b>                            | <b>24</b> |
| 3.1. 2MB FLASH PARTION TABLE                               | 24        |
| 3.2. CODE/RAM RESOURCE                                     | 24        |
| <b>4. DEBUG</b>  | <b>25</b> |
| <b>5. SDK GUIDE</b>  | <b>26</b> |
| 5.1. APP_MAIN.C IN SRC DIRECTORY                           | 26        |
| 5.1.1. <i>hf_gpio_fid_to_pid_map_table</i>                 | 26        |
| 5.1.2. <i>user_define_at_cmds_table</i>                    | 28        |
| 5.1.3. <i>sys_event_callback</i>                           | 28        |
| 5.1.4. <i>socketa_recv_callback, socketb_recv_callback</i> | 29        |
| 5.1.5. <i>bt_ntf_recv_callback, bt_ind_recv_callback</i>   | 29        |
| 5.1.6. <i>uart_recv_callback</i>                           | 29        |
| 5.1.7. <i>show_reset_reason</i>                            | 29        |
| 5.1.8. <i>app_init</i>                                     | 29        |
| 5.1.9. <i>app_fill_ble_adv_data</i>                        | 30        |
| 5.1.10. <i>app_main</i>                                    | 30        |
| 5.1.11. <i>hfgpio_fmap_check</i>                           | 30        |

|         |   |    |
|---------|---|----|
| 5.1.12. | <i>hfnet_start_uart</i> .....                         | 30 |
| 5.1.13. | <i>hfsmtlk_is_start</i> .....                         | 30 |
| 5.1.14. | <i>hfnet_start_ble_ntf, hfnet_start_ble_ind</i> ..... | 30 |
| 5.1.15. | <i>hf_ble_start_switch</i> .....                      | 30 |
| 5.1.16. | <i>hf_get_ble_run_mode</i> .....                      | 30 |
| 5.1.17. | <i>hf_start_ble</i> .....                             | 30 |
| 5.1.18. | <i>hfnet_wifi_is_active</i> .....                     | 31 |
| 5.1.19. | <i>hfnet_start_assis</i> .....                        | 31 |
| 5.1.20. | <i>hfnet_start_socketa, hfnet_start_socketb</i> ..... | 31 |
| 5.1.21. | <i>hfnet_start_httpd</i> .....                        | 31 |
| 5.2.    | EXAMPLES.....   | 31 |
| 5.2.1.  | <i>AT command-example/at</i> .....                    | 31 |
| 5.2.2.  | <i>Data storage-example/file</i> .....                | 31 |
| 5.2.3.  | <i>Webpage user definition-example/httpd</i> .....    | 31 |
| 5.2.4.  | <i>JSON-example/json</i> .....                        | 32 |
| 5.2.5.  | <i>mdns-example/mdns</i> .....                        | 33 |
| 5.2.6.  | <i>mqtt-example/mqtttest</i> .....                    | 33 |
| 5.2.7.  | <i>netcallback-example/netcallback</i> .....          | 33 |
| 5.2.8.  | <i>PWM-example/pwmtest</i> .....                      | 33 |
| 5.2.9.  | <i>smtp-example/smtp</i> .....                        | 34 |
| 5.2.10. | <i>sniffer-example/sniffer</i> .....                  | 34 |
| 5.2.11. | <i>socket -example/socket</i> .....                   | 34 |
| 5.2.12. | <i>SSL/TLS-example/ssltest</i> .....                  | 34 |
| 5.2.13. | <i>Thread, timer-example/thread, timer</i> .....      | 35 |
| 5.2.14. | <i>uart-example/uart</i> .....                        | 35 |
| 5.2.15. | <i>data storage-example/uflash</i> .....              | 35 |
| 5.2.16. | <i>OTA-example/update</i> .....                       | 35 |
| 5.2.17. | <i>SSDP-example/upnp</i> .....                        | 36 |
| 5.2.18. | <i>WebSocket-example/websocket</i> .....              | 36 |
| 5.2.19. | <i>wifi-example/wifi</i> .....                        | 36 |

## History

**V 1.0** 11-11-2020 Reformat SDK document

**V 1.1** 01-12-2021 Update SDK V2.XXX 2MB mapping

**V 1.2** 03-28-2022 Update SDK3.XXX Version bootloader and change compile environment to Linux(This new V3 SDK does not support Windows compile environment)

**V 1.3** 08-02-2022 Update code size cut description.

# 1. Description

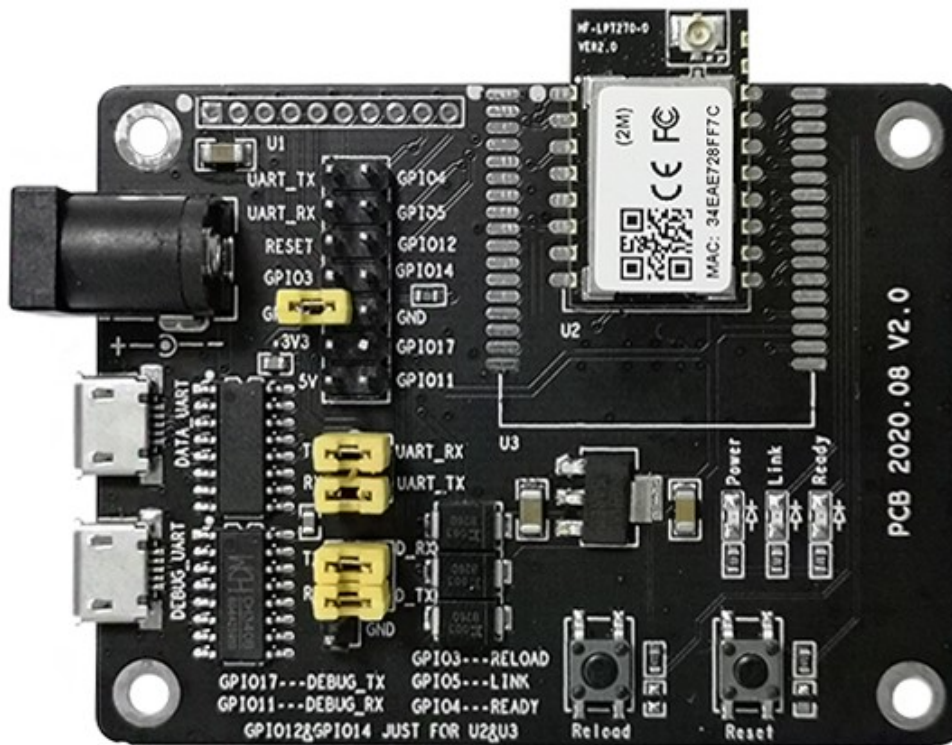
HF-LPX70 (or HF-LPx70) series module is a low-power embedded Wi-Fi + BLE module developed by Shanghai High-Flying Electronics Technology Co., Ltd. The current series includes multiple modules with different footprint.

All modules in this series can use the same SDK and API manual. In the following documents, "HF-LPX70" is collectively referred to as the modules on this platform. In some cases, HF-LPB170 will also refer to this series of modules.

## 1.1. HF-LPX70 Series EVK

Use USB connect DATA\_UART to PC and install CH340B driver from following link. Then do following develop.

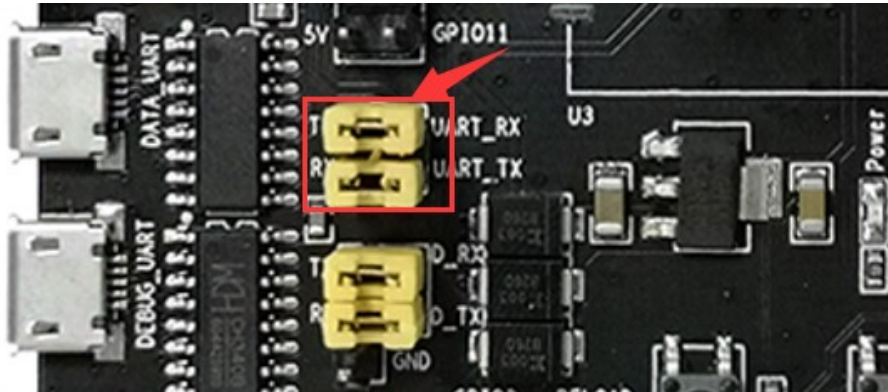
<http://www.hi-flying.com/download-center-1/applications-1/download-item-usb-serial-drivers>



| Function           | Name       | Description  |
|--------------------|------------|--|
| External Interface | DATA_UART  | Power In and communication UART0 port.   |
|                    | DEBUG_UART | Power In and debug UART1 port  |
|                    | DC         | DC jack for power in, 5V input. When USB power supply is not enough, may use external adapter. |
| LED                | Power      | Power LED  |
|                    | Ready      | nReady LED   |
|                    | Link       | nLink LED  |

|        |        |   |
|--------|--------|---|
| Button | Reload | Smart Config and Restore factory default configuration. |
|        | Reset  | Reboot module   |

After test OK on EVK, may remove the following jumper (Break the module and CH340 UART), then connect UART\_RX and UART\_TX to user MCU board for products UART test.



## 2. Compile Environment

HF-LPX70 SDK is compiled with the RISC-V tool chain. The examples and third-party libraries provided in the SDK are compiled and provided in this way.

Latest V3 version SDK only support Linux compile environment, the previous firmware based on V2 version SDK, must transport it to new V3 version SDK and compile in Linux environment.

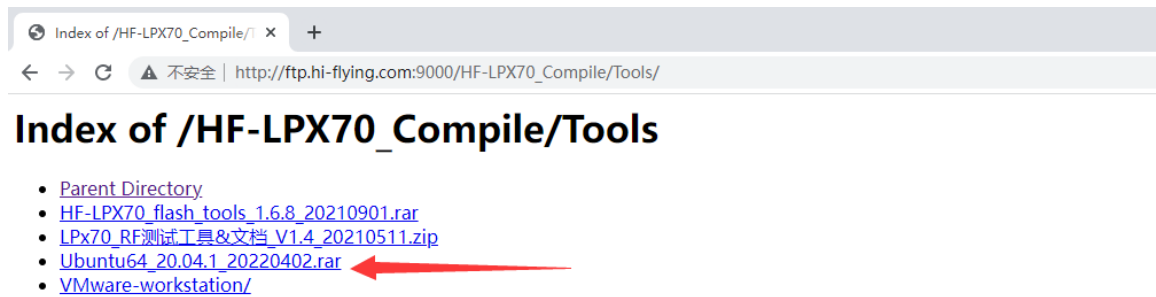
### 2.1. Linux Compile Environment

#### 2.1.1. Download All in One Virtual OS

Install VMWARE WORKSTATION and run our virtual OS, we have the virtual OS ready for compile, login password is 123456. May also follow our later chapter to install the compile environment.

Our Virtual OS:

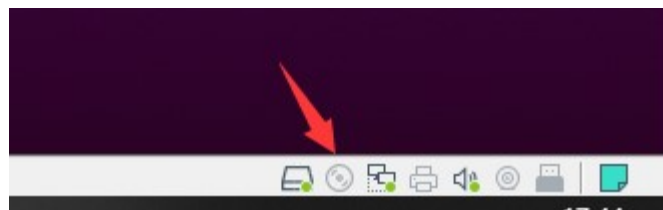
[http://ftp.hi-flying.com:9000/HF-LPX70\\_Compile/Tools/](http://ftp.hi-flying.com:9000/HF-LPX70_Compile/Tools/)

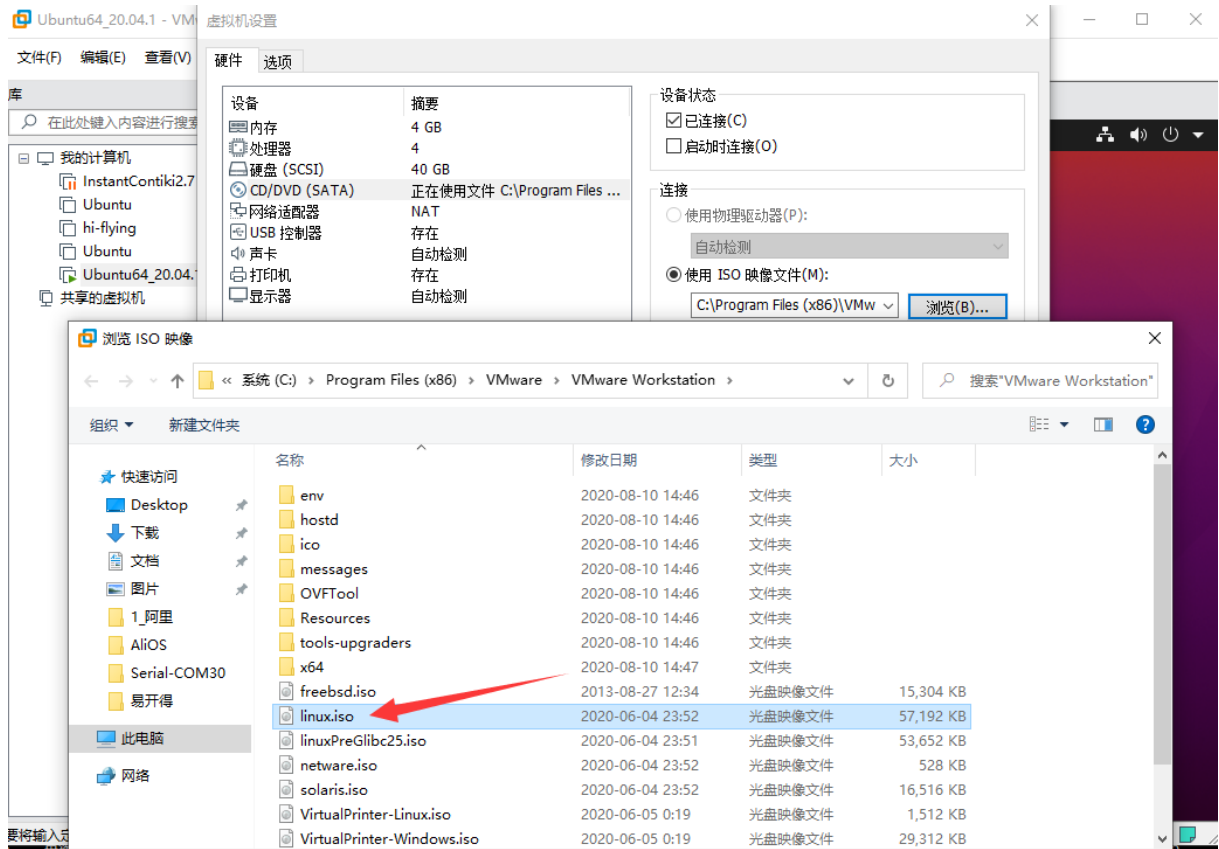


Modify RAM for VMWare.

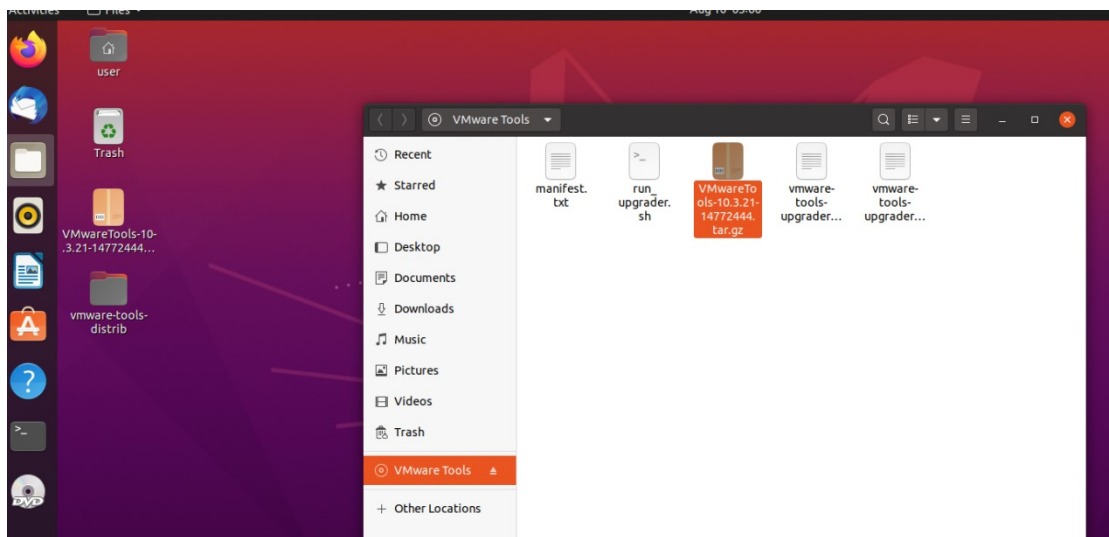


Install Vmware tools, click the following icon.

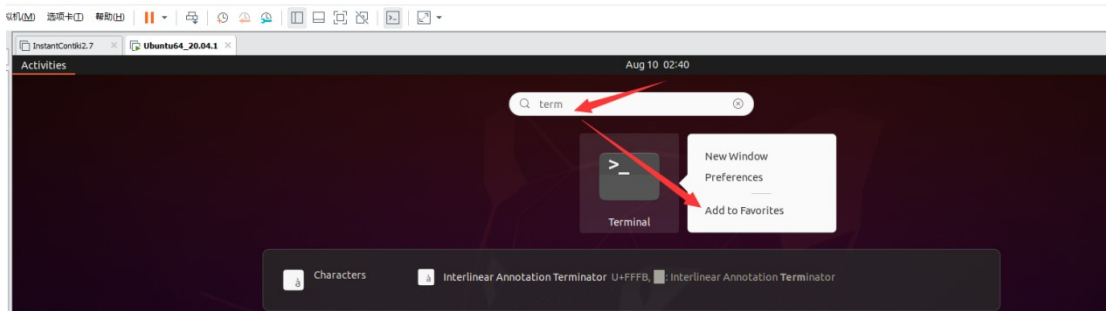




Copy file to desktop and install it.







Open terminal, and run `sudo ./vmware-install.pl` to install.

```
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
```

### 2.1.2. Install Compile Environment

Download any Linux64, the following is Ubuntu OS.

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

Install VMWARE WORKSTATION, and install Ubuntu.

Enlarge RAM



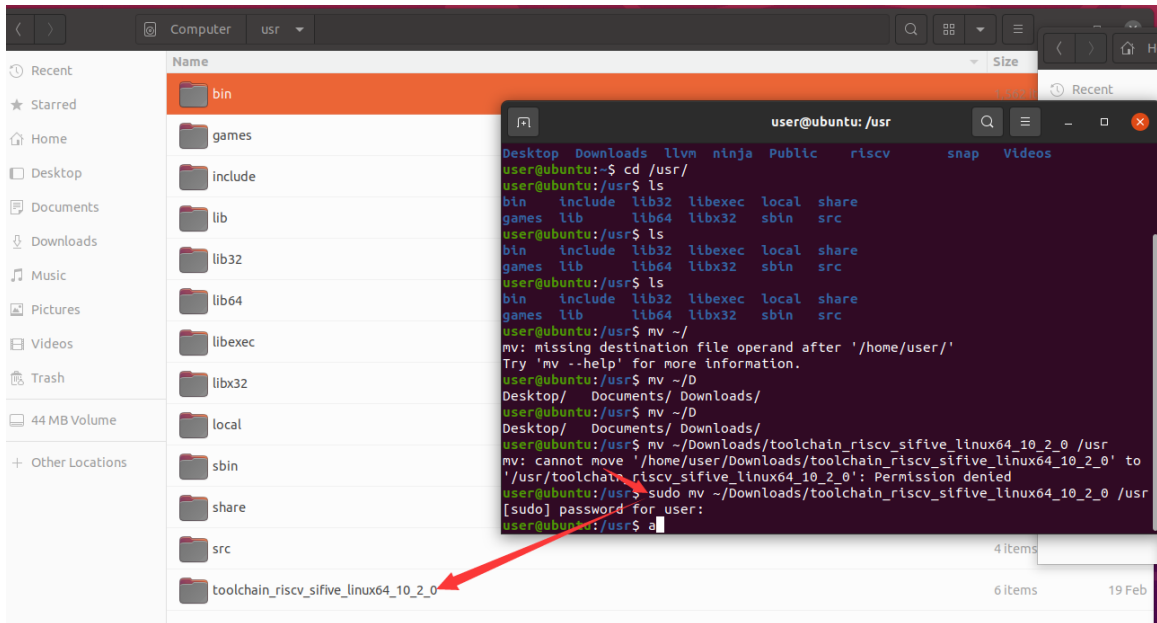
Compile tools is as following

[http://ftp.hi-flying.com:9000/HF-LPX70\\_Compile/Linux/](http://ftp.hi-flying.com:9000/HF-LPX70_Compile/Linux/)

## Index of /HF-LPX70\_Compile/Linux

- [Parent Directory](#)
- [riscv-example-build-lib.zip](#)
- [toolchain\\_riscv\\_sifive\\_linux64-master.zip](#)
- [toolchain\\_riscv\\_sifive\\_linux64\\_10\\_2\\_0.zip](#)

Download and copy the tools to following /usr/ directory.



Install Vmware tool (Same as previous chapter).

### 2.1.3. Set Shared Directory

After the following steps to add shared directory, it makes easy to edit in Windows OS (Source Insight recommended) and compile in Linux environment.

- Power off Ubuntu
- VMware tools select VM->Settings->Options->Shared Folders
- Click Add, Next->Select Windows shared directory, and then click Next->Enable this share->Finish
- VM->Settings->Options->Shared Folders, Folder sharing choose Always enabled
- Click OK to quit
- Install VMWareTools in Ubuntu
  - a) `sudo apt-get install open-vm-dkms`
  - b) `sudo mount -t vmhgfs .host:/ /mnt/hgfs`
  - c) `cd /mnt/hgfs`

### 2.1.4. Compile Note

If occur the following error.

```
AR out/MQTT-C/libMQTT-C.a
CC out/src/app_main.o
AR out/src/libsrc.a
LD out/lpt270.elf
Generating BIN File to /mnt/hgfs/share/LPx70-HSF-2MB_v3.00.13/out/lpt270.bin
Building Finish. To flash build output.
sh: 1: ././tools/lzma: not found
sh: 1: ././tools/lzma: not found
Open file (out/lpt270.bin.lzma) failed
```

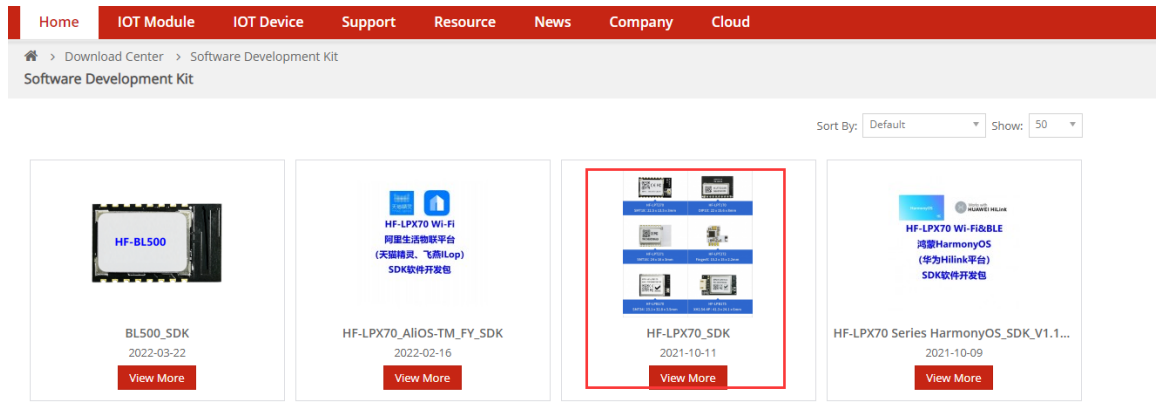
Run the following command to install the 32 bit system lib and then compile again.

- `sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386`

## 2.2. Download SDK

Download SDK V3 version to Linux OS.

<http://www.hi-flying.com/download-center-1/software-development-kit-1/download-item-hf-lpx70-sdk>



- Due to different module RF maybe different, so chose the right type in SDK is recommended, modify the **PROJECT\_NAME** in **build.sh**. Default use HF-LPT270.

此电脑 > Data (D:) > share > LPx70 > LPx70-HSF-2MB\_V3.0

| 名称            | 修改日期            | 类型    | 大小   |
|---------------|-----------------|-------|------|
| doc           | 2022/3/30 14:12 | 文件夹   |      |
| example       | 2022/3/28 14:18 | 文件夹   |      |
| sdk           | 2022/3/28 14:17 | 文件夹   |      |
| src           | 2022/3/28 14:18 | 文件夹   |      |
| thirdpartylib | 2022/3/28 14:17 | 文件夹   |      |
| tools         | 2022/3/28 15:45 | 文件夹   |      |
| build.sh      | 2022/3/30 14:13 | SH 文件 | 1 KB |
| Makefile      | 2022/3/30 14:11 | 文件    | 2 KB |
| Makefile.mk   | 2022/3/28 14:18 | MK 文件 | 1 KB |
| readMe.txt    | 2022/3/28 14:18 | 文本文档  | 1 KB |

D:\share\LPx70\LPx70-HSF-2MB\_V3.0\build.sh - Notepad++

```

1  #!/bin/sh
2  # lpt270
3  # lpt170
4  # lpt271
5  # lpb170
6  # lpb175
7  # lpt570
8
9
10 PROJECT_NAME="lpt270"
11 export PROJECT_NAME
12
13 make
14
15 if [ $? -eq 0 ]
16 then
17 ./tools/IMG_Tools out/$PROJECT_NAME.bin ./out/$PROJECT_NAME-upgrade.bin ./tools/lzma "HF-LPx70x1 Image" --version3
18 fi
19
20 exit $?

```

## ● SDK Directory :

- example: example code.
- sdk: SDK lib, head file, link scripts, usually upgrade this directory for SDK update.
- src: user program entry, user code always should be put in this directory.
- thirdpartylib, third party lib, call this third party API for user application when needed, if the third party lib need commercial license, customer need to get license from corresponding company.
- tools: tools and bootloader file.

|               |                 |              |      |
|---------------|-----------------|--------------|------|
| example       | 2022/3/30 15:43 | 文件夹          |      |
| out           | 2022/3/31 19:11 | 文件夹          |      |
| sdk           | 2022/3/30 15:43 | 文件夹          |      |
| SI            | 2022/3/31 19:00 | 文件夹          |      |
| src           | 2022/3/30 15:43 | 文件夹          |      |
| thirdpartylib | 2022/3/30 15:43 | 文件夹          |      |
| tools         | 2022/3/30 15:43 | 文件夹          |      |
| build.sh      | 2022/3/30 19:01 | SH 文件        | 1 KB |
| Makefile      | 2022/3/31 19:10 | 文件           | 2 KB |
| Makefile.mk   | 2022/3/30 16:04 | Makefile 源文件 | 1 KB |
| readMe.txt    | 2022/3/30 15:43 | 文本文档         | 1 KB |

## 2.3. Makefile

**TARGET:** Indicates the compiled product module, please select the right type according to the module used(**modify build.sh**), the wrong model selection may result in poor RF performance, but can still work.

**SDK\_VERSION:** Indicates the version number of the SDK, which corresponds to the content in the SDK folder, and can be modified to switch between different SDK versions;

**EXAMPLE\_NAME:** Indicates whether to compile src or example directory, the default is null to compile src directory;

**INCLUDE\_COMPONENTS:** Indicates to select the components that need to be compiled.

```
1 #
2 # This is a project Makefile. It is assumed the directory this Makefile resides in is a
3 # project subdirectory.
4 #
5 ifeq ($(PROJECT_NAME),lpt270)
6 TARGET = _LPT270_
7 else ifeq ($(PROJECT_NAME),lpt170)
8 TARGET = _LPT170_
9 else ifeq ($(PROJECT_NAME),lpt271)
10 TARGET = _LPT271_
11 else ifeq ($(PROJECT_NAME),lpt170)
12 TARGET = _LPT170_
13 else ifeq ($(PROJECT_NAME),lpt175)
14 TARGET = _LPT175_
15 else ifeq ($(PROJECT_NAME),lpt570)
16 TARGET = _LPT570_
17 endif
18
19 PROJECT_PATH := $(abspath .)
20 export PROJECT_PATH
21
22 ifeq ($(origin PROJECT_SDK_PATH), undefined)
23 PROJECT_SDK_PATH_GUESS ?= $(shell pwd)
24 PROJECT_SDK_PATH ?= $(PROJECT_SDK_PATH_GUESS)
25 endif
26
27 AFTER_BUILD =
28
29 SDK_VERSION = 3.00
30
31 #EXAMPLE_NAME=ssl
32 EXAMPLE_NAME ?= null
33
34 ifeq ($(EXAMPLE_NAME),null)
35 APPDIR = MQTT-C src
36 else
37 APPDIR = MQTT-C example/$(EXAMPLE_NAME)
38 endif
39
40 ifeq ($(EXAMPLE_NAME),httpsd)
41 APPDIR += Cyassl
42 else ifeq ($(EXAMPLE_NAME), joylink)
43 APPDIR += JD_NEW_SDK Cyassl
44 else ifeq ($(EXAMPLE_NAME), mqtt_ssl)
45 APPDIR += mbedtls
46 else ifeq ($(EXAMPLE_NAME), ssl)
47 APPDIR += mbedtls
48 else ifeq ($(EXAMPLE_NAME), websocket)
49 APPDIR += websocket-c websocketclient
50 endif
51
52 INCLUDE_COMPONENTS += $(SDK_VERSION) $(APPDIR)
53
54 include $(PROJECT_SDK_PATH)/tools/scripts/project.mk
55 CFLAGS += -D_HSF_ -D$(TARGET)
56
```

## 2.4. Compile

Run **./build.sh** to compile.

```

user@ubuntu: /mnt/hgfs/LPB100_Project/HF-LPX70_platfor...
user@ubuntu: /mnt/hgfs/LPB100_Project/HF-LPX70_platform/LPx70-HSF-2MB_V3.0$ ./build.sh
AS out/3.00/libs/start.o
AR out/3.00/lib3.00.a
CC out/MQTT-C/MQTTClient-C/MQTTClient.o
CC out/MQTT-C/MQTTClient-C/MQTTTF.o
CC out/MQTT-C/MQTTPacket/MQTTConnectClient.o
CC out/MQTT-C/MQTTPacket/MQTTConnectServer.o
CC out/MQTT-C/MQTTPacket/MQTTDeserializePublish.o
CC out/MQTT-C/MQTTPacket/MQTTFormat.o
CC out/MQTT-C/MQTTPacket/MQTTPacket.o
CC out/MQTT-C/MQTTPacket/MQTTSerializePublish.o
CC out/MQTT-C/MQTTPacket/MQTTSubscribeClient.o
CC out/MQTT-C/MQTTPacket/MQTTSubscribeServer.o
CC out/MQTT-C/MQTTPacket/MQTTUnsubscribeClient.o
CC out/MQTT-C/MQTTPacket/MQTTUnsubscribeServer.o
AR out/MQTT-C/libMQTT-C.a
CC out/src/app_main.o
AR out/src/libsrc.a
LD out/lpt270.elf
Generating BIN File to /mnt/hgfs/LPB100_Project/HF-LPX70_platform/LPx70-HSF-2MB_V3.0/out/lpt270.bin
Building Finish. To flash build output.
user@ubuntu: /mnt/hgfs/LPB100_Project/HF-LPX70_platform/LPx70-HSF-2MB_V3.0$ a

```

| 工作 (E:) > hiflying > LPB100_Project > HF-LPX70_platform > LPx70-HSF-2MB_V3.0 > out > |                 |           |          |  |
|--|-----------------|-----------|----------|--|
| 名称   | 修改日期            | 类型        | 大小       |  |
| 3.00   | 2022/4/1 10:18  | 文件夹       |          |  |
| gpio   | 2022/3/31 18:03 | 文件夹       |          |  |
| MQTT-C   | 2022/4/1 10:18  | 文件夹       |          |  |
| spi  | 2022/3/31 19:10 | 文件夹       |          |  |
| src  | 2022/4/1 10:18  | 文件夹       |          |  |
| <b>lpt270.bin</b>  | 2022/4/1 10:18  | BIN 文件    | 815 KB   |  |
| lpt270.bin.lzma  | 2022/4/1 10:18  | LZMA 文件   | 479 KB   |  |
| lpt270.bin.unlzma  | 2022/4/1 10:18  | UNLZMA 文件 | 815 KB   |  |
| lpt270.elf   | 2022/4/1 10:18  | ELF 文件    | 8,480 KB |  |
| lpt270.map   | 2022/4/1 10:18  | MAP 文件    | 5,184 KB |  |
| <b>lpt270-upgrade.bin</b>  | 2022/4/1 10:18  | BIN 文件    | 480 KB   |  |

The firmware will be generated at following directory.

工作 (E:) > hiflying > LPB100\_Project > HF-LPX70\_platform > LPx70-HSF-2MB\_V3.0 > out >

| 名称                 | 修改日期            | 类型        | 大小       |
|--------------------|-----------------|-----------|----------|
| 3.00               | 2022/4/1 10:18  | 文件夹       |          |
| gpio               | 2022/3/31 18:03 | 文件夹       |          |
| MQTT-C             | 2022/4/1 10:18  | 文件夹       |          |
| spi                | 2022/3/31 19:10 | 文件夹       |          |
| src                | 2022/4/1 10:18  | 文件夹       |          |
| lpt270.bin         | 2022/4/1 10:18  | BIN 文件    | 815 KB   |
| lpt270.bin.lzma    | 2022/4/1 10:18  | LZMA 文件   | 479 KB   |
| lpt270.bin.unlzma  | 2022/4/1 10:18  | UNLZMA 文件 | 815 KB   |
| lpt270.elf         | 2022/4/1 10:18  | ELF 文件    | 8,480 KB |
| lpt270.map         | 2022/4/1 10:18  | MAP 文件    | 5,184 KB |
| lpt270-upgrade.bin | 2022/4/1 10:18  | BIN 文件    | 480 KB   |

lpt270.bin: firmware used for UART bootloader method

**lpt270-upgrade.bin: firmware used for OTA method. The OTA upgrade file is compressed and has a firmware verification code, so its size is less than the bootloader program raw file.**

If need to recompile, use make clean to clear the \*.o files, and run ./build.sh again.

If compile encounter following failure, delete the whole out directory and compile it again.

```
user@ubuntu:/mnt/hgfs/LPB100_Project/HF-LPX70_platform/LPx70-HSF-2MB_V3.0$ ./build.sh
libs/start.d:1: *** target pattern contains no '%'. Stop.
libs/start.d:1: *** target pattern contains no '%'. Stop.
libs/start.d:1: *** target pattern contains no '%'. Stop.
make: *** [/mnt/hgfs/LPB100_Project/HF-LPX70_platform/LPx70-HSF-2MB_V3.0/tools/scripts/project.mk:517: component-3.00-build] Error 2
```

工作 (E:) > hiflying > LPB100\_Project > HF-LPX70\_platform > LPx70-HSF-2MB\_V3.0 >

| 名称            | 修改日期            | 类型  | 大小 |
|---------------|-----------------|-----|----|
| .svn          | 2022/3/30 15:43 | 文件夹 |    |
| doc           | 2022/3/30 19:08 | 文件夹 |    |
| example       | 2022/3/30 15:43 | 文件夹 |    |
| out           | 2022/3/30 19:03 | 文件夹 |    |
| sdk           | 2022/3/30 15:43 | 文件夹 |    |
| SI            | 2022/3/30 15:45 | 文件夹 |    |
| src           | 2022/3/30 15:43 | 文件夹 |    |
| thirdpartylib | 2022/3/30 15:43 | 文件夹 |    |
| tools         | 2022/3/30 15:43 | 文件夹 |    |

## 2.5. Firmware Program

Default compile src directory is our general firmware, refer to 【High Flying Wi-Fi Module Operation Guide】 for firmware upgrade and function test.

<http://www.hi-flying.com/hf-lpt270?search=LPT270>

| Description       | Specification   | Downloads |
|-------------------|---|-----------|
| User Guide        | HF-LPX70_tuya_20200911 [Download Times: 165, Date Update: 2020-09-16 15:42:03]                                      |           |
|                   | HF-LPX70 Series Wi-Fi_BLE Module User Manual-V1.5(20201027) [Download Times: 157, Date Update: 2020-10-28 14:47:41] |           |
|                   | High Flying Wi-Fi Module Operation Guide_20201110 [Download Times: 101, Date Update: 2020-11-10 17:31:04]           |           |
| Application Notes | EVK Evaluation Board Design Data_20200923 [Download Times: 157, Date Update: 2020-09-24 09:42:15]                   |           |

If module current firmware is V1.XXX version(AT+VER to query this), then the V2.XXX version generated by the SDK can not be upgraded via the above method(V1.XXX and V2.XXX boot mapping table is different), need to use the following **flash\_tools\_1.6.8** to program, and rewrite the MAC address.

[http://ftp.hi-flying.com:9000/HF-LPX70\\_Compile/Tools/](http://ftp.hi-flying.com:9000/HF-LPX70_Compile/Tools/)

← → ↻ ⚠ 不安全 | http://ftp.hi-flying.com:9000/HF-LPX70\_Compile/Tools/

## Index of /HF-LPX70\_Compile/Tools

- [Parent Directory](#)
- [HF-LPX70\\_flash\\_tools\\_1.6.8\\_20210901.rar](#)
- [LPx70\\_RF测试工具&文档\\_V1.4\\_20210511.zip](#)

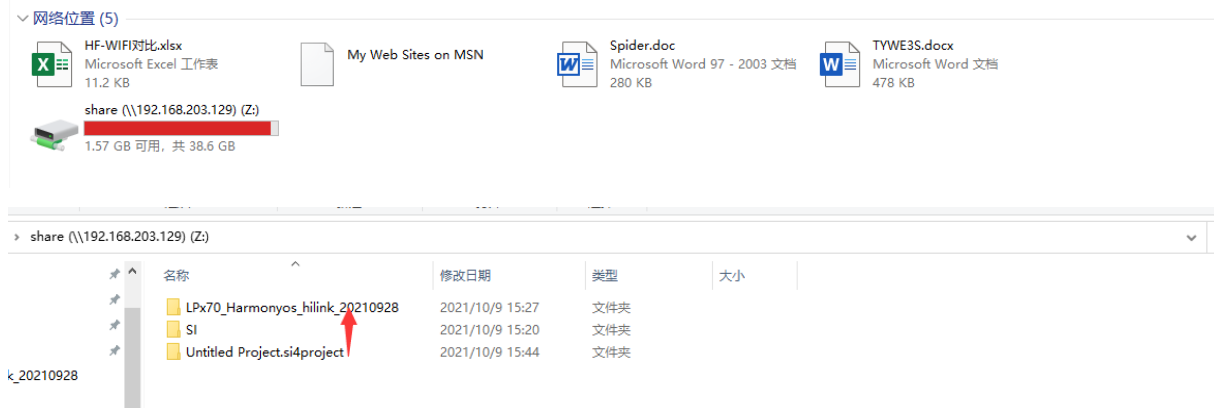
## 2.6. Samba Shared Code

There is also another method to use samba for shared code (code is saved in Linux, share to windows. Use SourceInsight (V4 or above) or vscode in windows for code editor and compile in Linux. Google it for more information.

Add write privilege for these code.

- `chmod 777 ~/share/LPx70_Harmonyos_hilink_20210928/ -R`

Mapping to disk in windows, then these code can be accessed in Windows.



## 2.7. User Code

### ● Function definition

Return type + Function name + Parameters

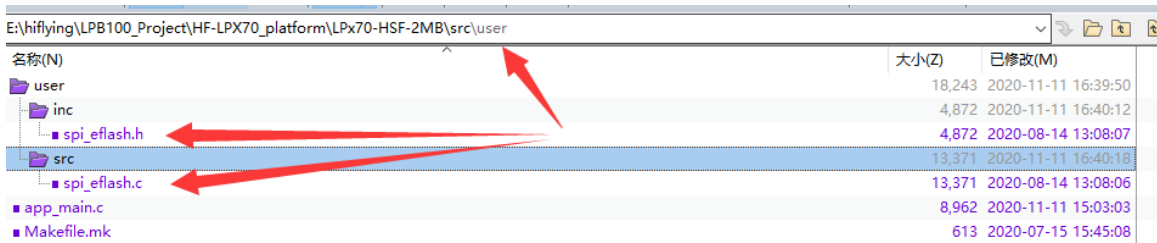
```
void test_func1(char *a);
```



## ● User code add description

The entire SDK is compiled in the form of components. Both sdk and src directory are components. Each component will compile a library file and then link to generate an executable file.

Recommend to create a user directory in src directory, and put user head file and source file all into this like following directory.

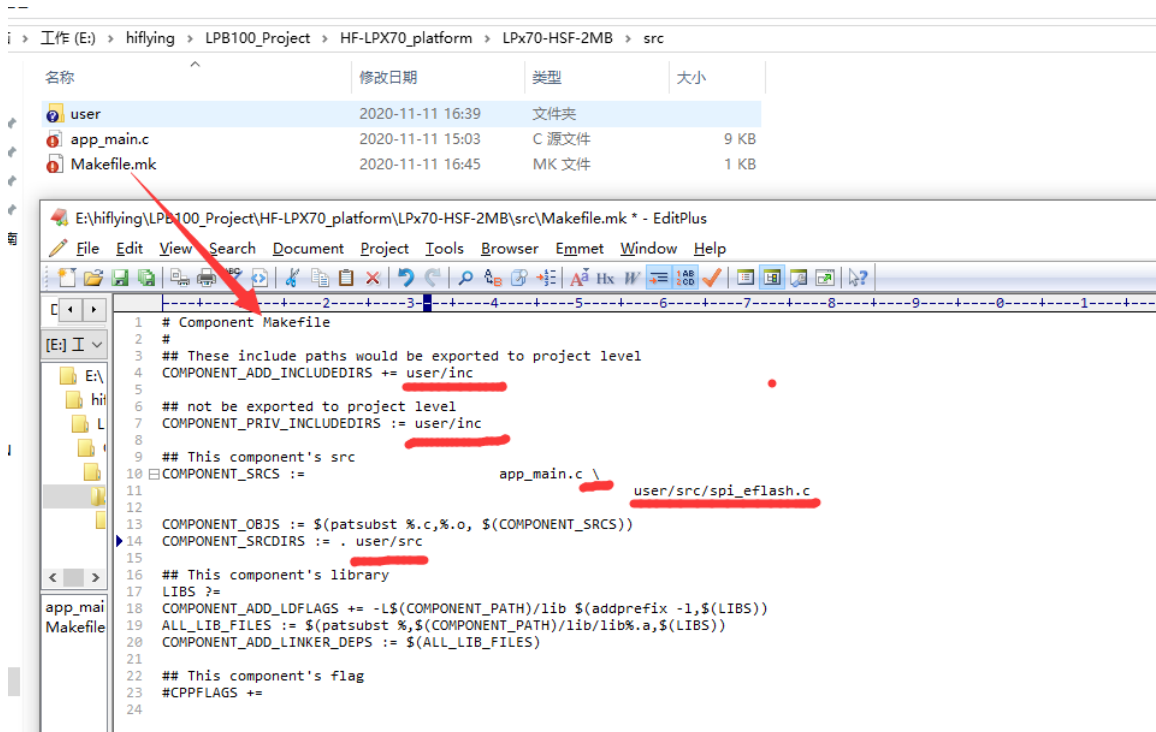


| 名称(N)        | 大小(Z)  | 已修改(M)              |
|--------------|--------|---------------------|
| user         | 18,243 | 2020-11-11 16:39:50 |
| inc          | 4,872  | 2020-11-11 16:40:12 |
| spi_eflash.h | 4,872  | 2020-08-14 13:08:07 |
| src          | 13,371 | 2020-11-11 16:40:18 |
| spi_eflash.c | 13,371 | 2020-08-14 13:08:06 |
| app_main.c   | 8,962  | 2020-11-11 15:03:03 |
| Makefile.mk  | 613    | 2020-07-15 15:45:08 |

## ● User adds source code files

Add a .c file. Source files based on HSF must include the <hsf.h> header file. After including this header file, the source code can call HSF-based API functions; if you want to use libc interface functions, please #include related Header files, such as calling the string manipulation function #include <string.h>, calling the time function #include <time.h>, etc.

COMPONENT\_SRCS in Makefile.mk indicates the source file to be compiled. MCOMPONENT\_ADD\_INCLUDEDIRS  
COMPONENT\_PRIV\_INCLUDEDIRS in Makefile.mk indicates the path of the header file that needs to be added, and the scope of the two is different. Modify this file to compile user file.



### ● Add third-party library

LIBS in Makefile.mk indicates the library files that need to be compiled. The default library is placed in the lib folder and the name starts with lib. If it does not meet the requirements, you need to modify COMPONENT\_ADD\_LDFLAGS and ALL\_LIB\_FILES.

### ● Add compilation options

It can be added to the CPPFLAGS in the Makefile.mk corresponding to each component, or it can be added to the CFLAGS in the Makefile.

### ● SDK Upgrade:

Usually just need to upgrade the sdk directory file.



## 2.8. Development Note

In order to avoid affecting the factory test of High Flying and causing production failure, please pay attention to the following points during the SDK development process:

1. Do not affect the automatic connection to the router function, for example: power on to start scanning the router or power on to enter SmartLink, etc. will affect the production test of High Flying;

Solution: Add a flag bit in Flash or use the method of reading GPIO level to distinguish the startup process. If you use GPIO mode, please ensure that the GPIO level is stable;

2. User defined AT command name should be different from our HF default AT command.

3. Add user version AT command AT+APPVER, for better distinguishment from our default firmware, see 【SDK Guide】 chapter.

## 2.9. Coding Tools

Recommend to use sourceinsight or Windows Visual Studio Code Tools for project development.

The RSIC-V compiler is relatively strict, and may involve some irregular code modifications;

## 2.10. Q&A

### 2.10.1. Exception problem

Problem description: The following print appears during program operation, and then the module restarts or freezes

```
[ERROR] mcause: 38000005, mepc 2306bcc0, mtval 00000000
[ERROR] exception code: 5, Load access fault
```

Problem analysis: The above problem is generally caused by accessing some inaccessible addresses in the program, or reading and writing some uninitialized register addresses. The function address where the problem occurred can be judged by the value of mepc, and mtval is the parameter, as shown in the figure above. As shown, by using the addr2line tool or viewing the map file, it can be judged that the 0x2306bcc0 address is located in the strlen function, and it can be known that the problem is that passing a value of 0x00000000 to the strlen function causes an exception.

addr2line:

```
liubo@Liu-PC MSYS ~/LPx70-HSF
$ c:/toolchain_riscv_sifive_msys/bin/riscv64-unknown-elf-addr2line.exe -e out/lpt270.elf -a -f 0x2306bcc0
strlen
??:?
```

map file:

```
.text      0x000000002306bca4      strlen
          0x000000002306bd30      0x84
```

### 2.10.2. Stack overflow problem

Problem description: The following print appears during program operation, and then the module restarts or freezes

```
[ERROR] task name (assis_scan_thread) id (1107482432) stack overflow
```

Problem analysis: The above problem is generally caused by a stack overflow of a thread in the program. The name and id of the overflow thread will be pointed out in the printing. In many cases, the name may not be printed, and it can be judged by the id. Generally speaking, if the program is not recompiled, the id will not change. You can query the status of all threads created by hftthread\_create through AT+SMEM=thread or hftthread\_show, including task ID, priority, and stack usage. You can also query through hftthread\_get\_current\_taskid The task ID of the current thread.

### 2.10.3. RAM overflowed problem

Problem description: The compilation prompts as follows, and the compilation fails

```
`bss' will not fit in region `ram_tcm'  
region `ram_tcm' overflowed by 1432 bytes
```

Problem analysis: The above problem is caused by the overflow of the static variable area, which can reduce the global variables and static variables in the code and change it to the hfmem\_malloc method.

### 2.10.4. FLASH overflowed problem

Problem description: The compilation prompts as follows, and the compilation fails

```
`rodata' will not fit in region `flash'  
region `flash' overflowed by 9776 bytes
```

Problem analysis: The above problem is caused by the compiled firmware exceeding the limit. You can check the definition of the flash size in the LPBX7.ld file in the SDK.

See following for code size cutting down.

### 2.10.5. Backtrace

Debug level set to 2 (AT+NDBGL=2 or AT+NDBGL=2,1), if system exception, it will print function call relation, use addr2line to locate the exception function name.

```

[ERROR] mcause 38000005, mepc 0x23004534, mtval 0x00000000
Exception Entry-->>>
mcause 38000005, mepc 23004534, mtval 00000000
Exception code: 5
msg: Load access fault
[ERROR] exception code: 5, Load access fault
=== backtrace start ===
backtrace: 0x230034d6
backtrace: 0x23000a5e
backtrace: 0x23004538 <--- TRAP
backtrace: 0x23006fc6
backtrace: 0x23007172
backtrace: 0x2301ed30
backtrace: 0x2301f00a
backtrace: INVALID!!!
=== backtrace end ===

```

## 2.11. Code Size Cut

Default SDK code partition is 916KB, current general firmware is 902KB, general firmware include all the function described in manual. But for SDK usage, these may no needed, so should comment these to cut down code size.

| _platform > LPx70-HSF-2MB_v3.00.16 > out > |                |           |          |  | 在 out 中搜索 |
|--|----------------|-----------|----------|--|-----------|
| 名称   | 修改日期           | 类型        | 大小       |  |           |
| 3.00                                       | 2022/8/2 16:51 | 文件夹       |          |  |           |
| MQTT-C                                     | 2022/8/2 16:51 | 文件夹       |          |  |           |
| src  | 2022/8/2 16:51 | 文件夹       |          |  |           |
| <b>lpt270.bin</b>                          | 2022/8/2 16:51 | BIN 文件    | 902 KB   |  |           |
| lpt270.bin.lzma                            | 2022/8/2 16:51 | LZMA 文件   | 535 KB   |  |           |
| lpt270.bin.unlzma                          | 2022/8/2 16:51 | UNLZMA 文件 | 902 KB   |  |           |
| lpt270.elf                                 | 2022/8/2 16:51 | ELF 文件    | 9,455 KB |  |           |
| lpt270.map                                 | 2022/8/2 16:51 | MAP 文件    | 5,801 KB |  |           |
| lpt270-upgrade.bin                         | 2022/8/2 16:51 | BIN 文件    | 535 KB   |  |           |

The step is as following:

- Modify SDK root directory Makefile, change PROJECT\_MODE to DELETE\_CODE

```

1 #
2 # This is a project Makefile. It is assumed the directory th
3 # project subdirectory.
4 #
5 ifeq ($(PROJECT_NAME),lpt270)
6 TARGET = _LPT270_
7 else ifeq ($(PROJECT_NAME),lpt170)
8 TARGET = _LPT170_
9 else ifeq ($(PROJECT_NAME),lpt271)
10 TARGET = _LPT271_
11 else ifeq ($(PROJECT_NAME),lpt170)
12 TARGET = _LPB170_
13 else ifeq ($(PROJECT_NAME),lpt175)
14 TARGET = _LPB175_
15 else ifeq ($(PROJECT_NAME),lpt570)
16 TARGET = _LPT570_
17 else ifeq ($(PROJECT_NAME),lpt272)
18 TARGET = _LPT272_
19 endif
20
21 #for delete code size
22 # NOMAL_MODE
23 # DELETE_CODE
24 PROJECT_MODE=DELETE_CODE
25

```

- Uncomment the following items to cut corresponding function.

```

76 #delete code size option
77 ifeq ($(PROJECT_MODE),DELETE_CODE)
78 # /*delete socketa, about 142K*/
79 CFLAGS += -D_DELETE_SOCKETA_
80
81 # /*delete MDNS for smartAPLink, about 14K*/
82 #CFLAGS += -D_DELETE_MDNS_
83
84 # /*delete smartAPLink, about 17K*/
85 #CFLAGS += -D_DELETE_SMARTAPLINK_
86
87 # /*delete smartLink, about 10K*/
88 #CFLAGS += -D_DELETE_SMARTLINK_
89
90 # /*delete http server, about 25K*/
91 #CFLAGS += -D_DELETE_HTTPD_
92
93 # /*delete BLE, about 80K*/
94 #CFLAGS += -D_DELETE_BLE_
95 endif

```

- Comment these function if it is called in app\_main.c.

```
//AT+NETP socket
#ifdef _DELETE_SOCKETA_
if(hfnet_start_socketa(HFTHREAD_PRIORITIES_LOW,(hfnet_callback_t)socketa_rcv_callback)!=HF_SUCCESS)
{
HF_Debug(DEBUG_WARN,"start socketa fail\r\n");
}
#endif
//AT+SOCKB socket
if(hfnet_start_socketb(HFTHREAD_PRIORITIES_LOW,(hfnet_callback_t)socketb_rcv_callback)!=HF_SUCCESS)
{
HF_Debug(DEBUG_WARN,"start socketb fail\r\n");
}

//Web Server
if(hfnet_start_httpd(HFTHREAD_PRIORITIES_MID)!=HF_SUCCESS)
{
HF_Debug(DEBUG_WARN,"start httpd fail\r\n");
}
```

| Option                           | Size  | Function  |
|----------------------------------|-------|---|
| CFLAGS += -D_DELETE_SOCKETA_     | 140KB | AT+NETP socket function   |
| CFLAGS += -D_DELETE_MDNS_        | 14KB  | Used for WeChat device find. Suggest to delete                                      |
| CFLAGS += -D_DELETE_SMARTAPLINK_ | 17KB  | SmartAPLink, delete if not used.  |
| CFLAGS += -D_DELETE_SMARTLINK_   | 10K   | SmartLink, delete if not used.  |
| CFLAGS += -D_DELETE_HTTPD_       | 25K   | HTTP Server webpage config, delete if not used.                                     |
| CFLAGS += -D_DELETE_BLE_         | 80KB  | BLE Function(include SmartBLELink and BLE throughput function), delete if not used. |

If still not enough after code cutting down, contact us to get dynamic partition function, make code size much larger then our default(max 1112KB).

## 3. Resources Description

### 3.1. 2MB Flash Partion Table

|             |  |
|-------------|--|
| 0x0000 0000 | BOOT(68KB)<br>Bootloader area  |
| 0x0001 1000 | CODE (916KB)<br>Run code area  |
| 0x000F 6000 | OTA UPGRADE(560KB)<br>OTA upgrade backup area  |
| 0x0018 2000 | WEB (200KB)<br>External web page   |
| 0x001B 4000 | UFLASH (200KB)<br>User flash   |
| 0x001E 6000 | BOOT_CONFIG(8KB)<br>System use   |
| 0x001E 8000 | USERPAGE (4KB)<br>User parameter storage area  |
| 0x001E 9000 | USERPAGE_BACKUP (4KB)<br>User parameter save and backup area   |
| 0x001E A000 | F_SETTING (4KB)<br>Factory parameter storage area  |
| 0x001E C000 | USER_BIN_FILE (4KB)<br>hfile_userbin_write API interface<br>actual physical address                  |
| 0x001E D000 | USER_BIN_BACK_FILE (4KB)<br>hfile_userbin_write API interface<br>actual physical address backup area |
| 0x001E E000 | SYSTEM_SECTOR (8KB)<br>System use  |
| 0x001F 0000 | Reserved area(64KB)  |

OTA UPGRADE area is compressed, so its size(560KB) is smaller than the raw code(916KB).

The module OTA firmware should be download into the OTA upgrade back area via the following area. if check the firmware is correct, it set flag to indicate do upgrade operation in bootloader. If next reboot, bootloader will check this flag, and copy the OTA upgrade backup file to the CODE running area and clear the flag.

### 3.2. Code/RAM Resource

HF-LPX70 remains 100KB Ram for user application. AT+SMEM or hfsys\_get\_memory() API to query this.

```
AT+SMEM
+ok=current_size:98976
```



## 4. Debug

Module use UART to debug, HSF provides two API functions `u_printf` and `HF_Debug` to print debug information. But default debug log output function is disabled. There are two ways to enable it.

- `AT+NDBGL=1,1` enable debug log, `AT+NDBGL=0` to close log, the first parameters indicate the debug level, only level in `HF_debug` API above this setting will it be shown. The second parameter is UART number, 0 for communication UART0, 1 for debug UART1(921600 baud rate)
- `hfdbg_set_level(X)` API to output log, X represents the level of debugging information output, 0 to close. But API can not choose the UART number.

After enable debug, the log will output to corresponding UART. After set `AT+NDBGL=1,1`, the following log for example.

```
Entry HF-LPx70 Main Jun 22 2020 12:52:12>>>>
[VER] 1.00.2 (2020-06-12 16:00 1M)
[OTA] reload pin lev:1, ota_flag:0x00000000, run_mode:0x00000000
app_init
[SYS] reset nvram!
[SMTLK] smtlk_sign:0x0, reset reason:0x0
[MAC] ap mac:F0FE11118889
sdk version(1.00),the app_main start time is Jun 22 2020 12:55:55
reset_reasion:00000000
[UART] uart 0:115200, 8, 1, fc:0, parity:0
[WIFI] start ap ssid:HF-LPT270, ch:9, maxsta:5
[WIFI] start ap ip:10.10.100.254, mask:255.255.255.0
[DHCPD] dhcpd start, range[150-200]
[WIFI] start ap dhcp:10.10.100.150-10.10.100.200
```

When `AT+NDBGL=2,1`(the debugging level is set to 2), more system internal logs will be output, Please use level 2 when encounter SDK internal problems and send the log files to us.

## 5. SDK Guide

Compile SDK will generate our default firmware, the function is right our module manual and operation guide doc mentioned.

<http://www.hi-flying.com/iot-module/hf-lpt270>

| Description       | Specification   | Downloads |
|-------------------|---|-----------|
| User Guide        | HF-LPX70_tuya_20200911 [ Download Times: 168, Date Update: 2020-09-16 15:42:03 ]                                      |           |
|                   | HF-LPX70 Series Wi-Fi_BLE Module User Manual-V1.5(20201027) [ Download Times: 163, Date Update: 2020-10-28 14:47:41 ] |           |
|                   | High Flying Wi-Fi Module Operation Guide_20201110 [ Download Times: 106, Date Update: 2020-11-10 17:31:04 ]           |           |
| Application Notes | EVK Evaluation Board Design Data_20200923 [ Download Times: 159, Date Update: 2020-09-24 09:42:15 ]                   |           |

We provide SDK examples in example directory for developer to get familiar with. Modify the Makefile in root directory to compile our default or example firmware.

工作 (E:) > hiflying > LPB100\_Project > HF-LPX70\_platform > LPx70-HSF-2MB > example

| 名称          | 修改日期             | 类型  |
|-------------|------------------|-----|
| at          | 2020-11-05 18:12 | 文件夹 |
| file        | 2020-07-27 11:51 | 文件夹 |
| gpio        | 2020-08-26 15:50 | 文件夹 |
| httpd       | 2020-07-31 12:55 | 文件夹 |
| ir          | 2020-11-04 18:33 | 文件夹 |
| json        | 2020-07-27 11:51 | 文件夹 |
| mdns        | 2020-07-27 11:51 | 文件夹 |
| mqtt        | 2020-07-27 11:51 | 文件夹 |
| netcallback | 2020-07-27 11:51 | 文件夹 |
| pwm         | 2020-11-05 18:30 | 文件夹 |
| smtp        | 2020-07-27 11:51 | 文件夹 |
| sniffer     | 2020-07-27 11:51 | 文件夹 |
| socket      | 2020-07-27 11:51 | 文件夹 |
| ssl         | 2020-09-24 9:57  | 文件夹 |
| thread      | 2020-07-27 11:51 | 文件夹 |
| timer       | 2020-07-27 11:51 | 文件夹 |
| uart        | 2020-07-27 11:51 | 文件夹 |
| uflash      | 2020-07-27 11:51 | 文件夹 |
| update      | 2020-07-27 11:51 | 文件夹 |
| upnp        | 2020-07-27 11:51 | 文件夹 |
| websocket   | 2020-07-31 12:55 | 文件夹 |
| wifi        | 2020-07-27 11:51 | 文件夹 |

```

1  # This is a project Makefile. It is assumed the directory this Makefile resides in is a
2  # project subdirectory.
3
4  #
5
6  # __LPT270__
7  # __LPT570__
8  # __LPB170__
9
10 TARGET=__LPT270__
11
12 ifeq ($(TARGET), __LPT270__)
13   PROJECT_NAME := lpt270
14 else ifeq ($(TARGET), __LPT570__)
15   PROJECT_NAME := lpt570
16 else ifeq ($(TARGET), __LPB170__)
17   PROJECT_NAME := lpb170
18 endif
19 PROJECT_PATH := $(abspath .)
20 export PROJECT_PATH
21
22 ifeq ($(origin PROJECT_SDK_PATH), undefined)
23   PROJECT_SDK_PATH_GUESS ?= $(shell pwd)
24   PROJECT_SDK_PATH ?= $(PROJECT_SDK_PATH_GUESS)
25 endif
26
27 SDK_VERSION = 1.00
28
29 #EXAMPLE_NAME=at
30 #EXAMPLE_NAME ?= null
31
32 ifeq ($(EXAMPLE_NAME),null)
33   APPDIR = src
34 else

```

Modify as following to compile timer example code. Firmware will be generated in out directory.

```
EXAMPLE_NAME=timer
#EXAMPLE_NAME ?= null
```

The following just take a brief view of the default API used in our SDK, more and detailed API description see API doc and SDK header file.

### 5.1. app\_main.c in src Directory

#### 5.1.1. hf\_gpio\_fid\_to\_pid\_map\_table

Pin definition table, HFGPIO\_F\_NLINK, HFGPIO\_F\_NREADY, HFGPIO\_F\_NRELOAD is our function enum, the detailed function refer to our manual.

nReady Pin (LED) function(Low effective):

1. OS initial finished indicator. Only after this pin output low, can the UART function be used.

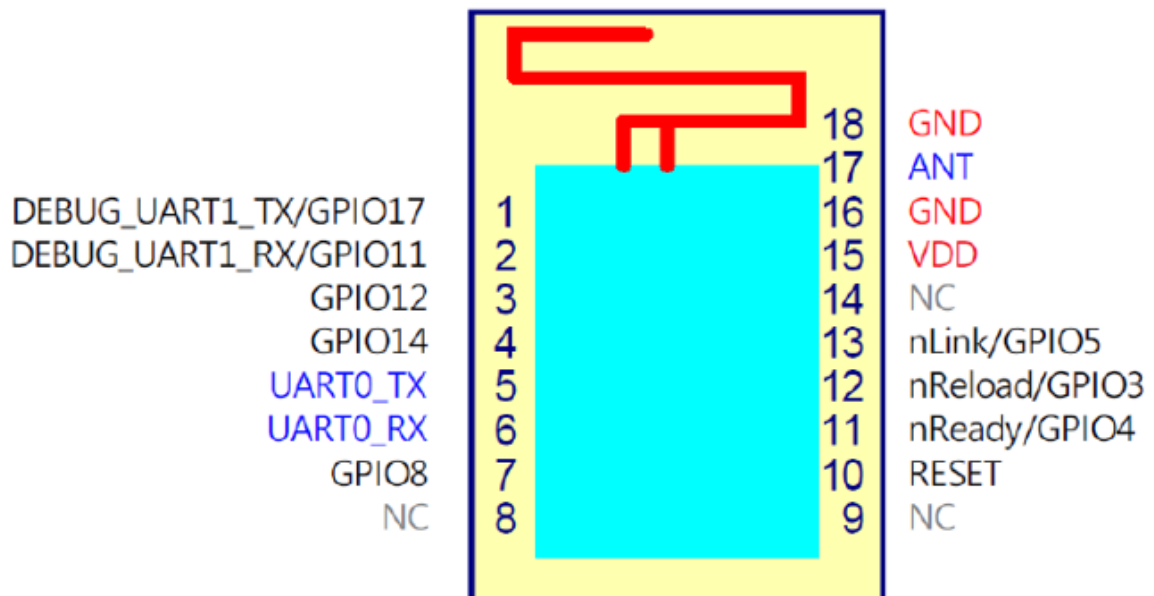
nLink Pin (LED) function(Low effective):

1. At wireless firmware and config upgrade mode , this LED used to indicate configure and upgrade status.
2. At “SmartLink” config mode, this LED is used to indicate APP to finish setting.
3. At normal mode, it's Wi-Fi link status indicator. Output Low when STA mode connect to router AP or other STA connect to it when in AP mode.

High-Flying strongly suggest customer fan out this pin to LED.

After mapping the PIN to the function enum, these PIN then have such feature.

```
LPX70_GPIO5,    //HFGPIO_F_NLINK
LPX70_GPIO4,    //HFGPIO_F_NREADY
LPX70_GPIO3,    //HFGPIO_F_NRELOAD
```



Remapping the GPIO table to redefine the pin function, see example

【example/gpiotest.c】

```
NOPIN,    //HFGPIO_F_NLINK
NOPIN,    //HFGPIO_F_NREADY
NOPIN,    //HFGPIO_F_NRELOAD
```

default IO mapping is as following.

```
const int hf_gpio_fid_to_pid_map_table[HFM_MAX_FUNC_CODE]=
{
    HFM_NOPIN,    //HFGPIO_F_JTAG_TCK
    HFM_NOPIN,    //HFGPIO_F_JTAG_TDO
    HFM_NOPIN,    //HFGPIO_F_JTAG_TDI
    HFM_NOPIN,    //HFGPIO_F_JTAG_TMS
    HFM_NOPIN,    //HFGPIO_F_USBDP
    HFM_NOPIN,    //HFGPIO_F_USBDM
    LPX70_GPIO16, //HFGPIO_F_UART0_TX
```

```

HFM_NOPIN,    //HFGPIO_F_UART0_RTS
LPX70_GPIO7,  //HFGPIO_F_UART0_RX
HFM_NOPIN,    //HFGPIO_F_UART0_CTS

HFM_NOPIN,    //HFGPIO_F_SPI_MISO
HFM_NOPIN,    //HFGPIO_F_SPI_CLK
HFM_NOPIN,    //HFGPIO_F_SPI_CS
HFM_NOPIN,    //HFGPIO_F_SPI_MOSI

LPX70_GPIO17, //HFGPIO_F_UART1_TX,
HFM_NOPIN,    //HFGPIO_F_UART1_RTS,
LPX70_GPIO11, //HFGPIO_F_UART1_RX,
HFM_NOPIN,    //HFGPIO_F_UART1_CTS,

LPX70_GPIO5,  //HFGPIO_F_NLINK
LPX70_GPIO4,  //HFGPIO_F_NREADY
LPX70_GPIO3,  //HFGPIO_F_NRELOAD
HFM_NOPIN,    //HFGPIO_F_SLEEP_RQ
HFM_NOPIN,    //HFGPIO_F_SLEEP_ON

HFM_NOPIN,    //HFGPIO_F_WPS
HFM_NOPIN,    //HFGPIO_F_RESERVE1
HFM_NOPIN,    //HFGPIO_F_RESERVE2
HFM_NOPIN,    //HFGPIO_F_RESERVE3
HFM_NOPIN,    //HFGPIO_F_RESERVE4
HFM_NOPIN,    //HFGPIO_F_RESERVE5

HFM_NOPIN,    //HFGPIO_F_USER_DEFINE
};

```

### 5.1.2. user\_define\_at\_cmds\_table

User defined AT command, the following example define AT+APPVER command.

```

#define ZJ_LVER    "1_Sam (2020/11/11)"
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;
}

```

### 5.1.3. sys\_event\_callback

Define system event, switch case code will be executed when event occur.

HFE\_WIFI\_STA\_CONNECTED: WIFI STA connect to router event

HFE\_WIFI\_STA\_DISCONNECTED: WIFI STA break router connection event

HFE\_CONFIG\_RELOAD: Reload to factory setting event, usually caused by AT+RELD or nReload PIN.

HFE\_DHCP\_OK: STA got DHCP IP from router event, static IP won't trigger this event.

HFE\_SMTLK\_OK: SmartLink success event, for HF-LPX70 modules, SmartBLELink is more recommended, so this event is useless.

HFE\_BLE\_CONNECTED: BLE connected event.

HFE\_BLE\_DISCONNECTED: BLE disconnected event

#### 5.1.4. socketa\_recv\_callback, socketb\_recv\_callback

Packet callback for AT+NETP Socket A and AT+SOCKB Socket B. May modify the received data and sent to UART.

HFNET\_SOCKETA\_DATA\_READY: Received data

HFNET\_SOCKETA\_CONNECTED: Only valid in TCP socket connection created.

HFNET\_SOCKETA\_DISCONNECTED: Only valid in TCP socket break connection.

#### 5.1.5. bt\_ntf\_recv\_callback, bt\_ind\_recv\_callback

BLE notification and indication channel callback, return len will make the received BLE data sent to UART. May return to 0 and use other API to deal with packet such as hfnet\_bt\_ntf\_send, hfnet\_bt\_ind\_send, hfnet\_socketa\_send, hfuart\_send.

HFNET\_BT\_NTF\_DATA\_READY: BLE notification receive data event

HFNET\_BT\_NTF\_DATA\_ENABLE: BLE notification connection event

HFNET\_BT\_NTF\_DATA\_DISENABLE: BLE notification break connection event

HFNET\_BT\_IND\_DATA\_READY: BLE indication receive data event

HFNET\_BT\_IND\_DATA\_ENABLE: BLE indication connection event

HFNET\_BT\_IND\_DATA\_DISENABLE: BLE indication break connection event

#### 5.1.6. uart\_recv\_callback

UART receive callback, return len will make the data[len] sent to socketA, socketB and BLE, may return 0 to cancel this default process, and use other API to deal with packet such as hfnet\_socketa\_send, hfuart\_send.

HFNET\_UART0\_DATA\_READY: UART0 receive data event.

HFNET\_UART1\_DATA\_READY: Debug UART1 receive data event, if need to use both UART, call hfnet\_start\_uart\_ex(HFUART1...) API, Debug UART1 also support AT command.

HFSYS\_STATE\_RUN\_CMD: AT command mode status, usually used for HF factory test.

#### 5.1.7. show\_reset\_reason

Usually used for tracking reboot problem root cause.

#### 5.1.8. app\_init

Main entry for user program. OS has not been initialized, so most OS API can not be used. This entry only used for initialize GPIO for fast control..

### 5.1.9. app\_fill\_ble\_adv\_data

BLE broadcast fill in API, may also use AT+ADV command. Total maximum length of BLE name(AT+BLENAME) and this ADV data is 26 bytes.

[illegible]

### 5.1.10. app\_main

User main entry, create thread, time.

### 5.1.11. hfgpio\_fmap\_check

hf\_gpio\_fid\_to\_pid\_map\_table GPIO mapping check. If multiple or error GPIO definition will return fail.

### 5.1.12. hfnet start uart

Start UART thread, register callback function.

### 5.1.13. hfsmtlk is start

Judge if module is in SmartBLELink, SmartAPLink, SmartLink V8 config mode. When trigger in config mode (nReload Pin or AT+SMARTBTSTART, AT+SMARTAPSTART, AT+SMTLK), module will reboot to enter config mode, show\_reset\_reason can be used to judge this in order to start a timer thread for timeout operation, default SDK will always be in config mode unless rebooted manually.

hfsmtlk\_is\_start waits for config success, user thread should be created after this API.

#### 5.1.14. hfnet start ble ntf, hfnet start ble ind

## Register BLE notification and indication callback.

### 5.1.15. hf ble start switch

Judge if BLE throughput function is enabled(AT+BLE command on or off)

### 5.1.16. hf\_get\_ble\_run\_mode

Judge BLE working mode, if in SmartBLEConfig mode, the SDK lib will create BLE thread internally. If not, user may call `hf_start_ble` to receive and sent BLE data.

### 5.1.17. hf start ble

## BLE start API.

#### 5.1.18. hfnet\_wifi\_is\_active

Module can be allowed to create socket or other network communication when it is working as STA mode and it connected to router. If no connection, lwip has not been initialized, and module don't allow to create socket or relative function. It can also remove this judgment, but start network communication must wait to created until HFE\_DHCP\_OK occur.

There is no effect on AP mode, module will skip this step.

#### 5.1.19. hfnet\_start\_assis

Assistant thread, used for UDP 48899 network AT command, usage see following link.

<http://www.hi-flying.com/download-center-1/applications-1/download-item-wifi-config-tools-v1-0>

#### 5.1.20. hfnet\_start\_socketa, hfnet\_start\_socketb

Default thread for AT+NETP and AT+SOCKB socket communication, deal packet in callback. If use standard socket API to create communication, these two thread can be commented to save RAM resources.

#### 5.1.21. hfnet\_start\_httpd

Web Service thread for config module parameters and OTA upgrade, may comment it for saving RAM.

## 5.2. Examples

#### 5.2.1. AT command-example/at

Learn AT command API to get parameters and define user AT command.

AT+TEST to save data into module flash.

AT+ADCTEST read GPIO14 AD value.

AT+PWMTEST set GPIO12, GPIO3, GPIO4, GPIO5 PWM output

#### 5.2.2. Data storage-example/file

Learn SDK AT command to save user data.

AT+FILE command to save user data module flash.

#### 5.2.3. Webpage user definition-example/httpd

Our webpage source file is open.

<http://www.hi-flying.com/iot-module/hf-lpt270>

| Description       | Specification | Downloads  |
|-------------------|---------------|--|
| User Guide        |               | HF-LPX70_tuya_20200911 [ Download Times: 180, Date Update: 2020-09-16 15:42:03 ]   |
|                   |               | HF-LPX70 Series Wi-Fi_BLE Module User Manual-V1.5(20201027) [ Download Times: 188, Date Update: 2020-10-28 14:47:41 ]    |
|                   |               | High Flying Wi-Fi Module Operation Guide_20201110 [ Download Times: 126, Date Update: 2020-11-10 17:31:04 ]              |
|                   |               | HF-LPX30 Series_HF-LPX70 Series Module Webpage Source_20200710 [ Download Times: 100, Date Update: 2020-11-19 14:07:51 ] |
| Application Notes |               | EVK Evaluation Board Design Data_20200923 [ Download Times: 172, Date Update: 2020-09-24 09:42:15 ]                      |

Modify this webpage source file and deal the webpage request.


The following is our modified webpage. Put these into our webpage source file and generate the upgrade file.

platform > LPx70-HSF-2MB > example > httpd

名称

修改日期

类型

 webpage

2020-07-31 12:55

文件夹

The generated lpb\_web.bin need to upgrade into moduel via following method(login with IP/iweb.html).

← → ↺ ⚠ 不安全 | 192.168.60.129/iweb.html

Upgrade application

选择文件 未选择任何文件

Upload

Upgrade customized webpage

选择文件 未选择任何文件

Upload

The Final webpage and function is as following.

Test page x +

← → ↺ ⚠ 不安全 | 192.168.60.129/index\_en.html

中文 | English

Type: ☒ AT ☐ PING

Cmd: AT+WSSSID Send

Result:

+ok=Sonster1

Firmware Version: 1.00.19 (2020-11-05 16:00 24h)

Webpage Version: 1.0

---

Test page x +

← → ↺ ⚠ 不安全 | 192.168.60.129/index\_en.html

中文 | English

Type: ☐ AT ☒ PING

Cmd: 192.168.60.134 Send

Result:

Success

Firmware Version: 1.00.19 (2020-11-05 16:00 24h)

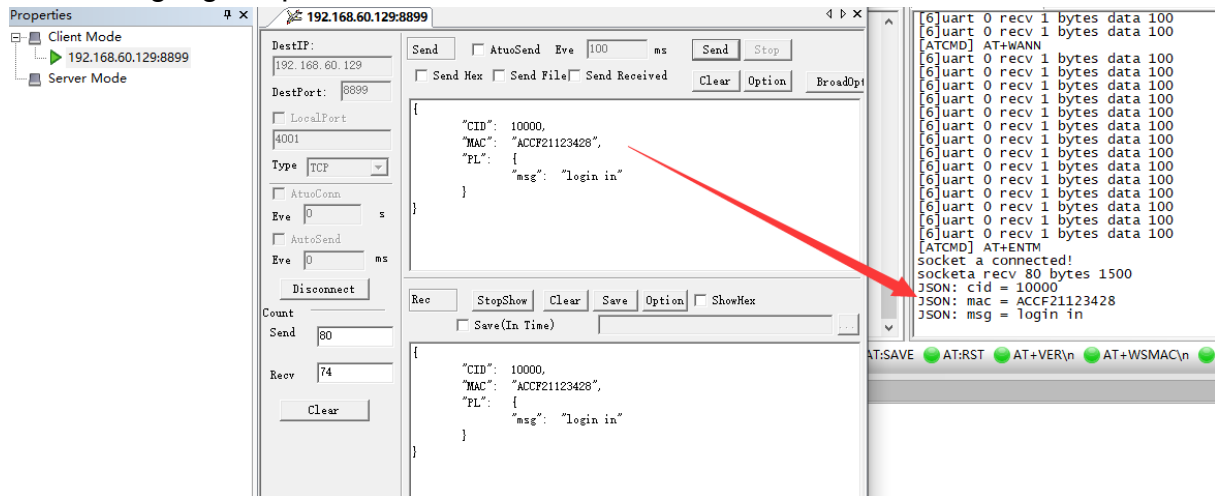
Webpage Version: 1.0

#### 5.2.4. JSON-example/json

##### Learn JSON API



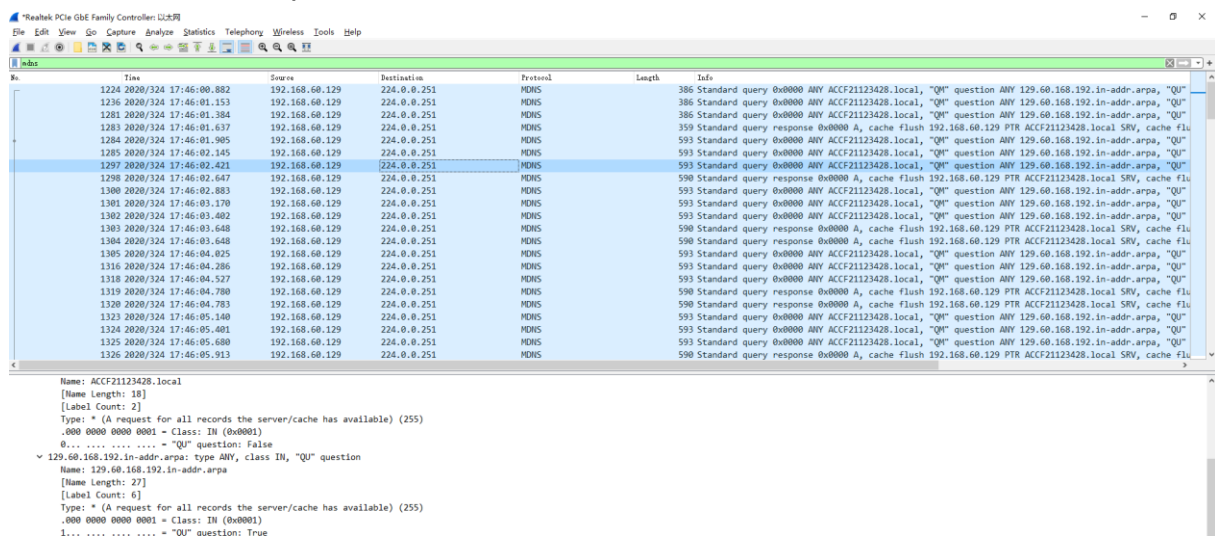
AT+NETP default works at TCP Server, 8899 port, PC create TCP client and will receive JSON packet when connection established, when send data to module, the debug log will print JSON information.



### 5.2.5. mdns-example/mdns

#### Learn mdns API

After module STA connect to router, it will send out mdns packet. Wireshark tools can catch this packet.



### 5.2.6. mqtt-example/mqtttest

#### Learn mqtt API

《HF MQTT AT 配置手册.docx》 list the MQTT command. Our default AT+NETP already support MQTT communication, it is more easier to use.

### 5.2.7. netcallback-example/netcallback

#### Learn UART send and callback scheme.

When send UART data “GPIO NLINK LOW” to module, nLink will output Low. When send UART data “GPIO NLINK HIGH” to module, nLink will output to High. When send UART data “GPIO NLINK FLASH” to module, nLink flash in 1HZ.d

### 5.2.8. PWM-example/pwmtest

#### Learn GPIO PWM usage.

Define following AT command for PWM control, usually used for LED application.

AT+PWM=pin,frequency,duty

pin: 引脚号

1 //GPIO11  
2 //GPIO12  
3 //GPIO3  
4 //GPIO4  
5 //GPIO5

frequency: 频率

20000 //20KHz

duty: 占空比

0~100

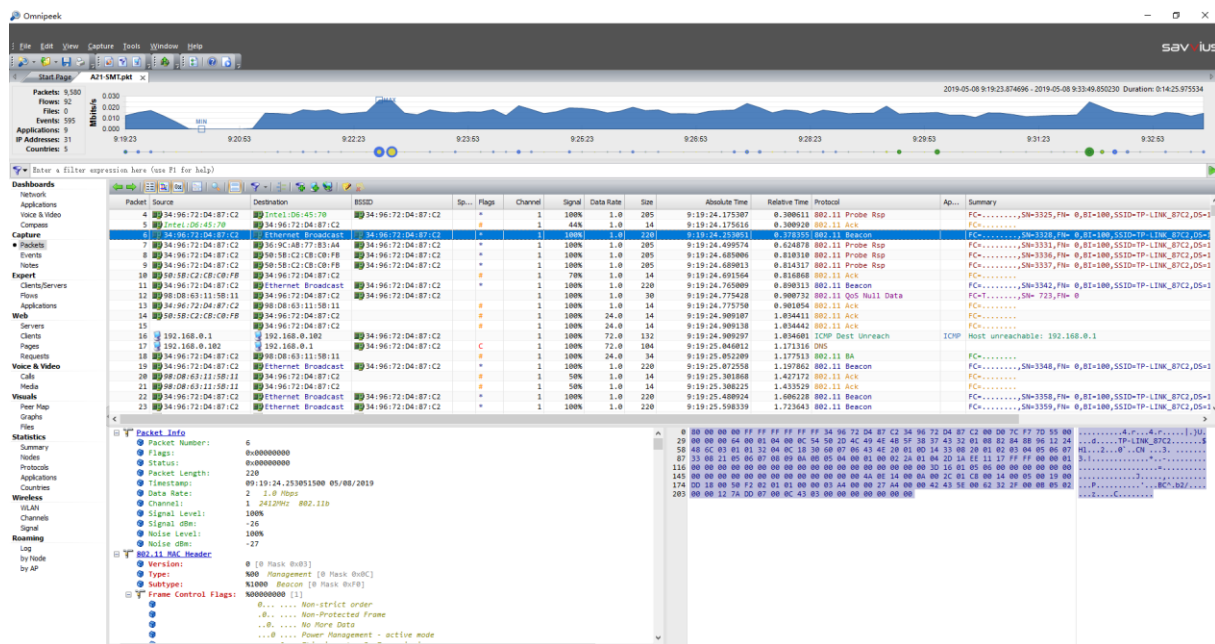
例子: AT+PWM=4,20000,20 //GPIO4 (nReady 引脚) 输出 20KHz 占空比 20%的 PWM 波形。

### 5.2.9. smtp-example/smtp

Learn send mail by user defined socket.

### 5.2.10. sniffer-example/sniffer

Learn Wi-Fi Sniffer function. The detailed packet meaning may use Omnipcap tools to analysis.



### 5.2.11. socket -example/socket

Learn standard socket API communication, create UDP, TCP Client, TCP keepalive.

UDP socket, port 10000, print debug log when receive data.

TCP Client socket, destination IP:10.10.100.150, port 10001, print debug log when receive data.

### 5.2.12. SSL/TLS-example/ ssltest

Learn TLS encryption for socket communication.

AT+SSLADDR set HTTPS server address. Module will send HTTPS request when UART receive HTTP raw data.

GET /api/v1/devices/Infos HTTP/1.1

Host: 192.168.1.1t

Connection: keep-alive

Two TLS lib can be used, mbedTLS or cyassl, recommend to use mbedTLS.

USE\_MBEDTLS\_LIB\_AND\_CYASSL\_API: Define use mbedTLS lib, API header file the same as cyassl.

#### 5.2.13. Thread, timer-example/thread, timer

Learn thread creation and timer function.

nReady flash 1HZ in this example

#### 5.2.14. uart-example/uart

Learn create UART thread to deal with data.

Create debug UART1 thread to deceive and send data. May also use hfnet\_start\_uart\_ex(HFUART1...) API.

#### 5.2.15. data storage-example/uflash

Learn save user data to module flash

SDK provides 200KB size for user application, hfuf flash\_XXXX API to manipulate this area.

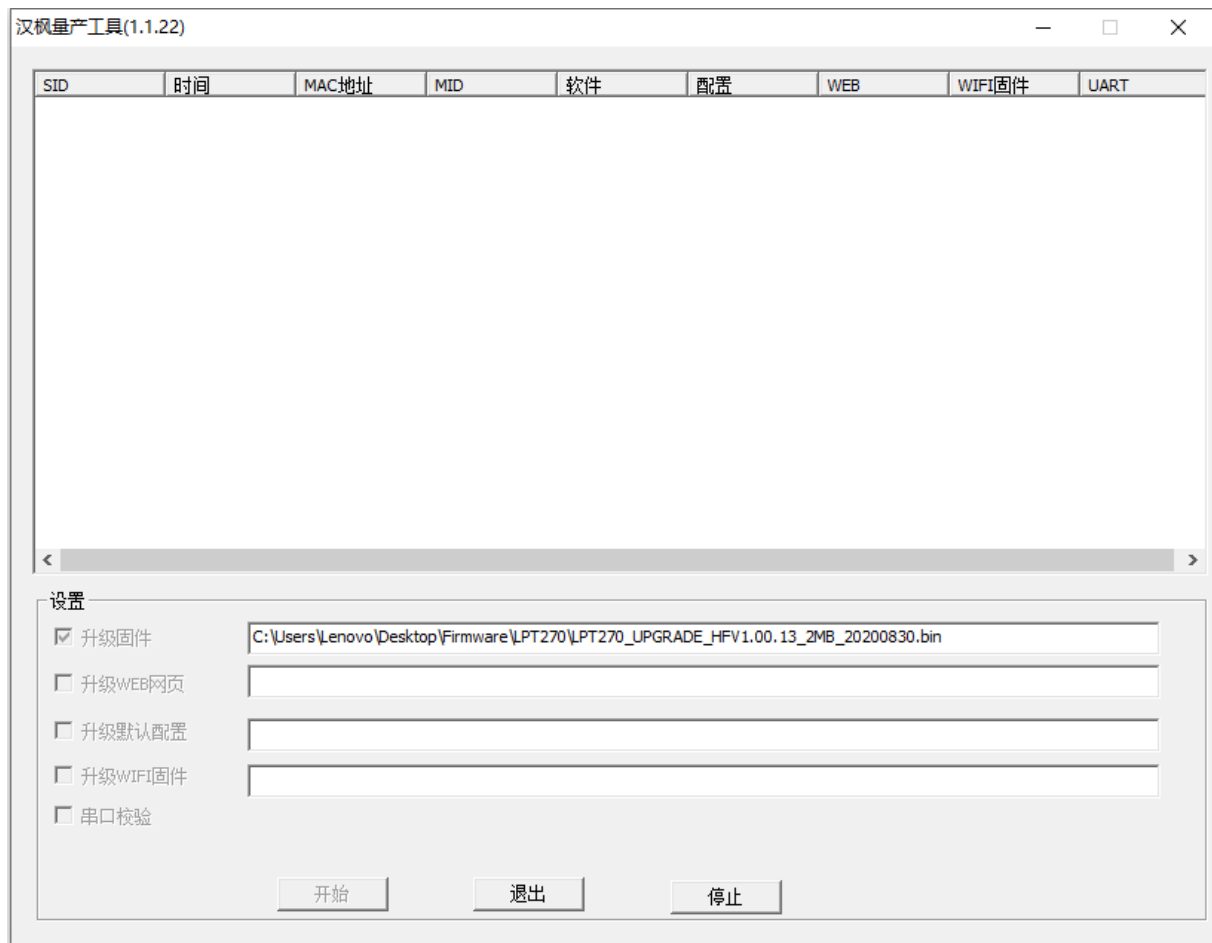
|              |                              |
|--------------|------------------------------|
| 0x001C 3000↵ | UFLASH (200KB)↵<br>用户 flash↵ |
|--------------|------------------------------|

#### 5.2.16. OTA-example/update

Learn OTA upgrade.

Download firmware from HTTP server. For test stage, may use our HFUpdate tools to create local HTTP server. After load firmware, the tools will copy the firmware into its directory and rename to mfw.bin. Shutdown PC fire wall, and open PC local IP/mfw.bin such as <http://192.168.60.134/mfw.bin>, to download the firmware. Then use module to test

【AT+UPGRADESW=<http://192.168.60.134/mfw.bin>】.



#### 5.2.17. SSDP-example/upnp

Learn SSDP multicast device found protocol..

#### 5.2.18. WebSocket-example/websocket

Learn WebSocket communication.

#### 5.2.19. wifi-example/wifi

Learn hfwifi\_scan\_test function.