High-Flying Software Framework (HSF-LPBx30) User Manual V1.1x

Vserion 1.1x
2018.01

Update Record：

Date	Editor	Modify	Delete
2017.08.10	**HF**	First Version	
2017.10.20	**HF**	Supplement	
2018.01.02	**HF**		

CONTENT

1. SDK Introduction

1.1 Item Number Introduction

HF-LPX30 series is a low-power embedded Wi-Fi module ,developed by Shanghai High-Flying. Contain HF-LPT230、HF-LPT130 and HF-LPB130 etc..

The series modules can use same SDK and API manual. The following types will all named as‘HF-LPTX30’. In some cases, it will be HF-LPB130 to represent these series.

1.2 Version Modify History

2017-0810：

First Version

2017-1020：

Complete and improve format.

Add HF-LPT130

Add Keil+GCC compile method

2018-0102：

Add platform difference description

2. Compile Environment Install

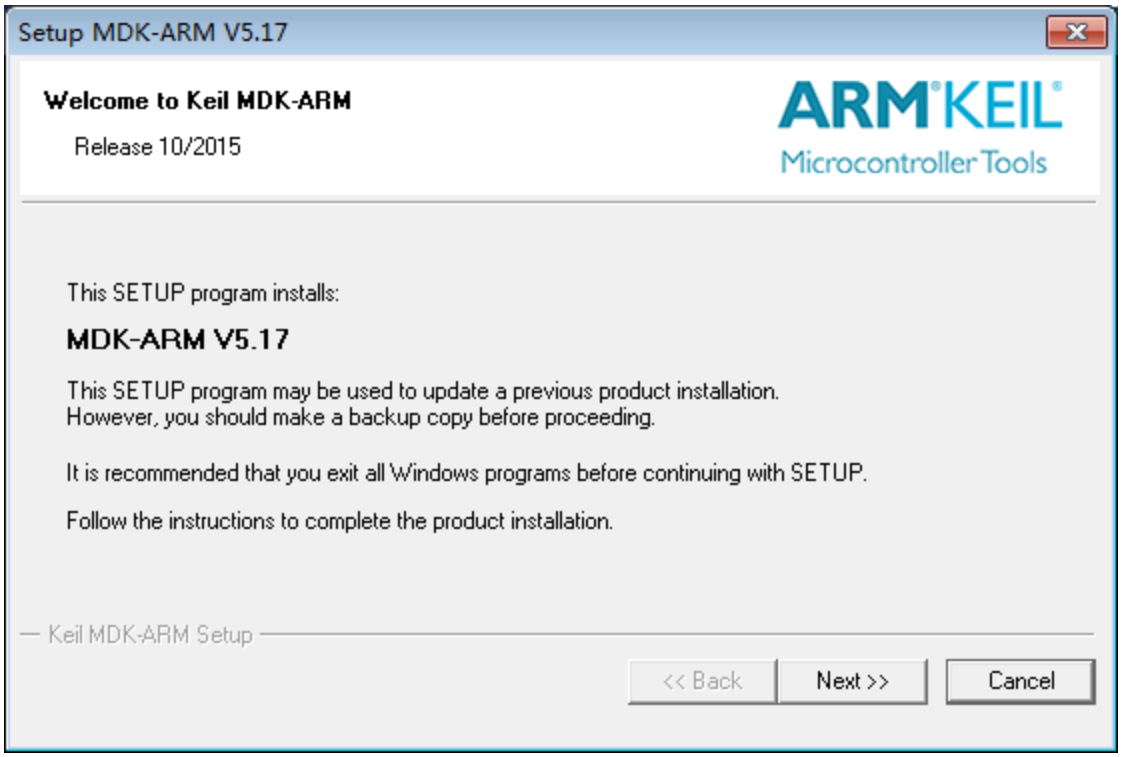
2.1 Compile Method Introduction

HF-LPX30 SDK use Keil+GCC complile method, The Example、third-party library etc...of SDK all use this kind of complile.

2.2 Creat Keil+GCC Compile Environment

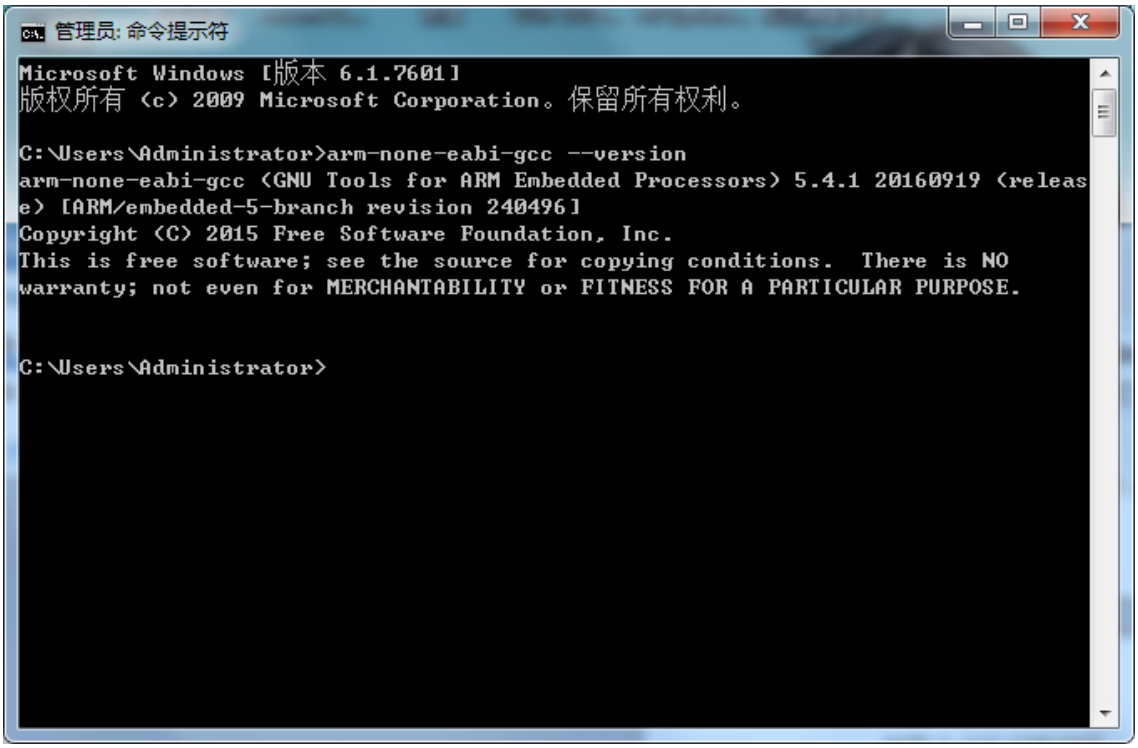
HF-LPX30 SDK use MDK-ARM 5.17 compile. To avoid compatibility problem, pls install following referred version Keil_v5.Download weblink:http://pan.baidu.com/s/1nvHUQ1b.

Choose default install Keil After downloaded.



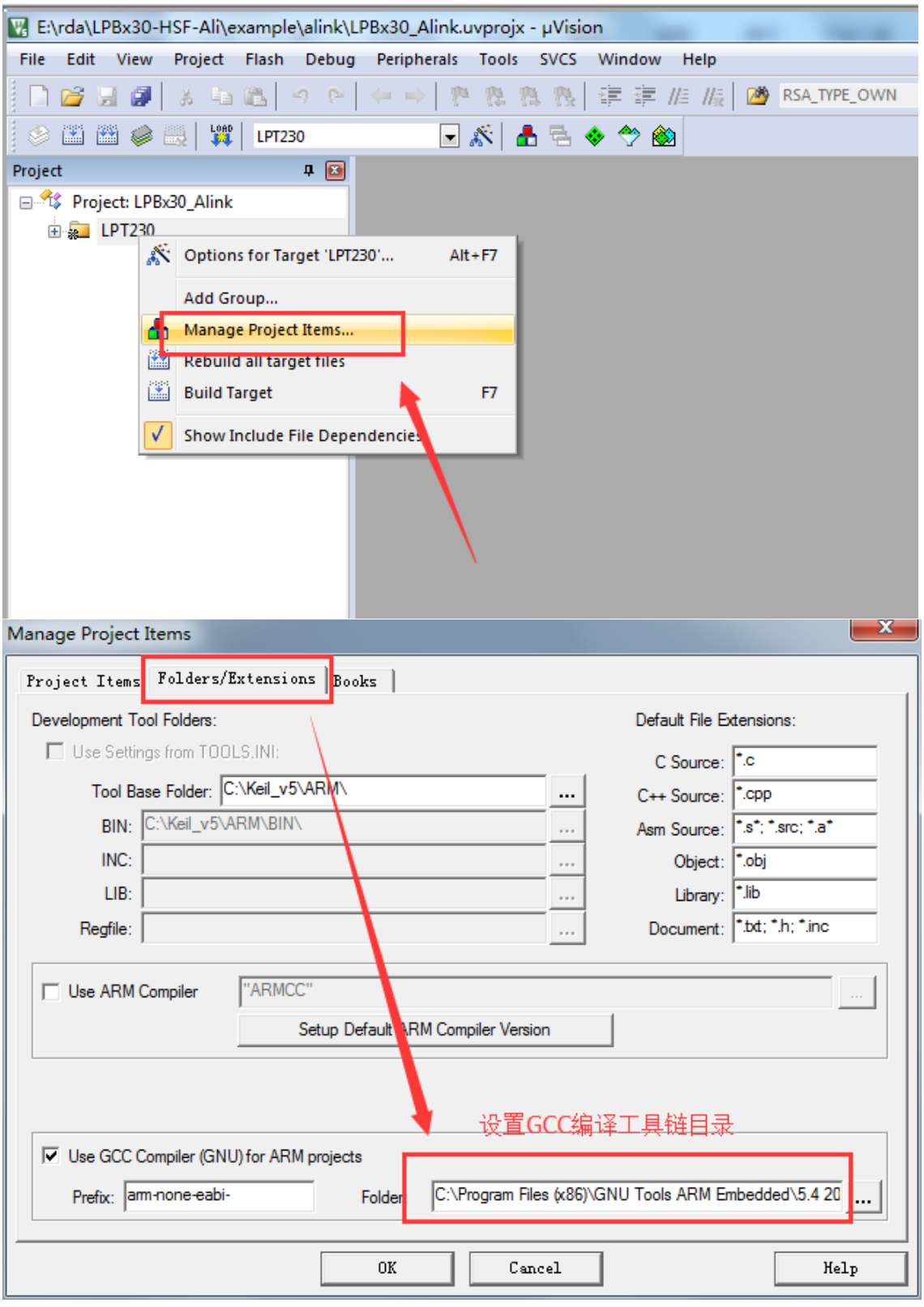
GCC pls **use at least** 5_4-2016q3 or above version,Pls download from GCC official website or following address to get the GCC compiler：https://pan.baidu.com/s/1bpMwJ4F Password：97gi，Follow the video to install after downloaded.

After installed, Pls open windows command line and Input arm-none-eabi-gcc --version to check if already successfully installed, Pls make sure the version is **at least** 5_4-2016q3 or above.

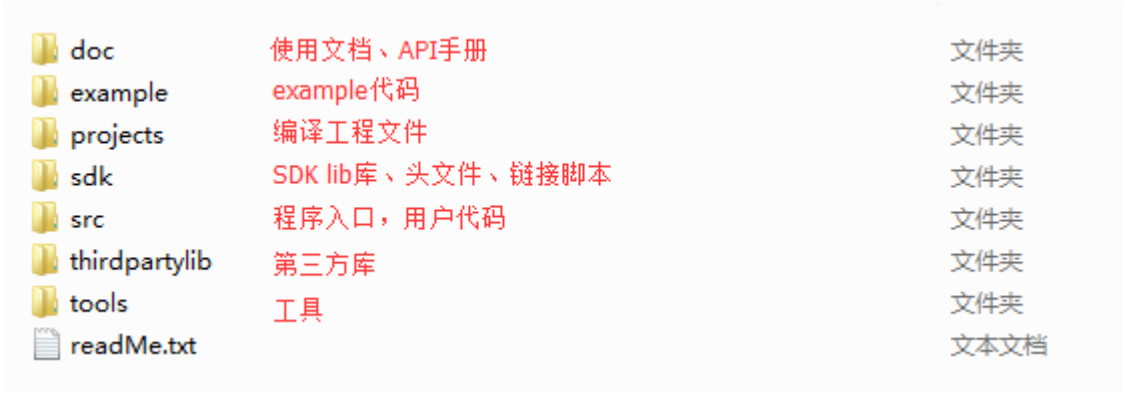


After installed GCC compiler tool, pls confirm the installation directory is ：C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 2016q3.

If the installation directory is wrong，Pls modify project configuration，Open project file and modify it according to following diagram：


E:\rda\LPBx30-HSF-Ali\example\alink\LPBx30_Alink.uvprojx - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
RSA_TYPE_OWN
LPT230
Project
Project: LPBx30_Alink
LPT230
Options for Target 'LPT230'... Alt+F7
Add Group...
Manage Project Items...
Rebuild all target files
Build Target F7
Show Include File Dependencies
Manage Project Items
Project Items
Folders/Extensions
Books
Development Tool Folders:
Use Settings from TOOLS.INI:
Tool Base Folder: C:\Keil_v5\ARM\
BIN: C:\Keil_v5\ARM\BIN\
INC:
LIB:
Regfile:
Default File Extensions:
C Source: *.c
C++ Source: *.cpp
Asm Source: *.s*; *.src; *.a*
Object: *.obj
Library: *.lib
Document: *.txt; *.h; *.inc
Use ARM Compiler
"ARMCC"
Setup Default ARM Compiler Version
设置GCC编译工具链目录
Use GCC Compiler (GNU) for ARM projects
Prefix: arm-none-eabi-
Folder: C:\Program Files (x86)\GNU Tools ARM Embedded\5.4 20
OK
Cancel
Help

2.3 SDK Catalogue Structure



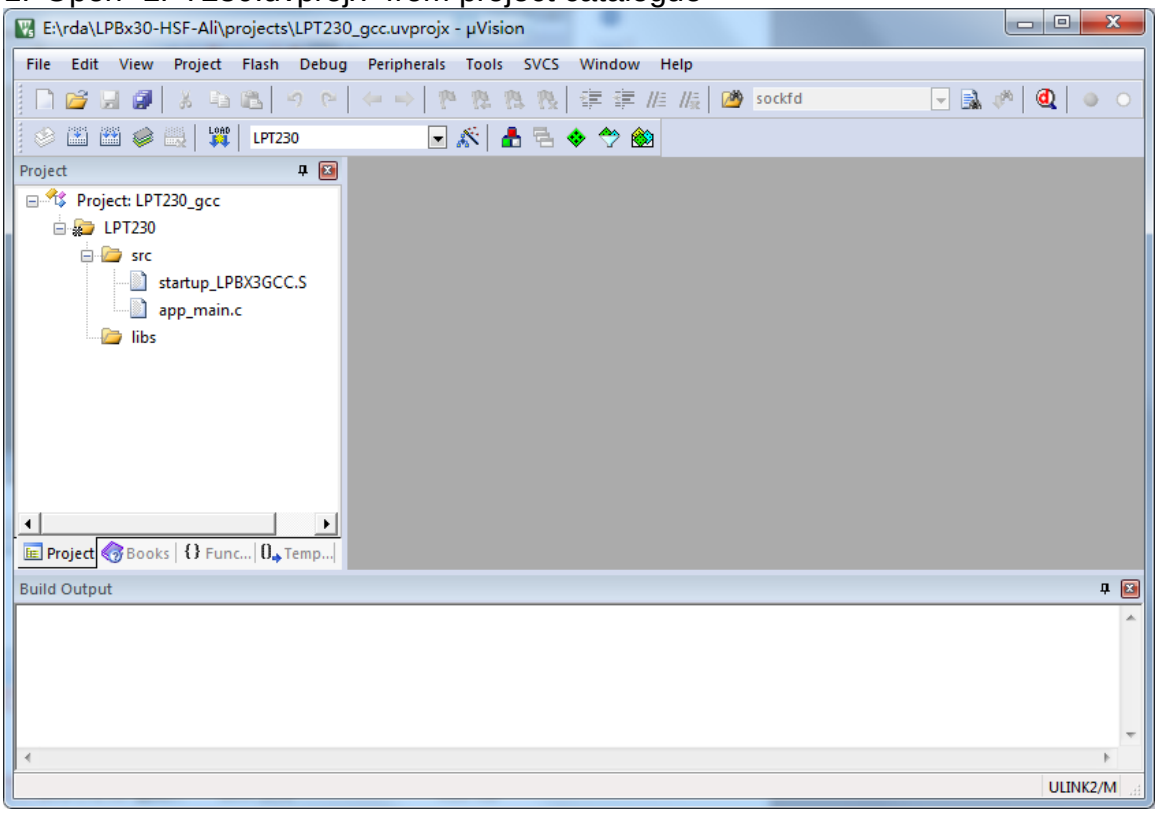
3. Start Compile

3.1 Project File Introduction

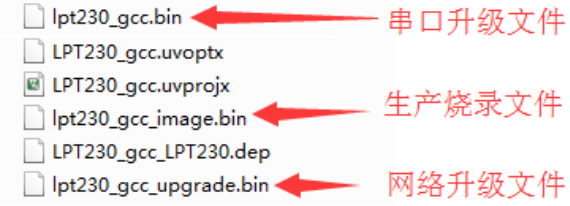
Pls choose the project that corresponding with module number to development，If choose wrong module number, It may cause poor RF performance and GPIO pin abnormal.

3.2 Start Compile

1. Open “LPT230.uvprojx” from project catalogue



2. Press “build” ,execute bin files



3.3 Source Code File for Users

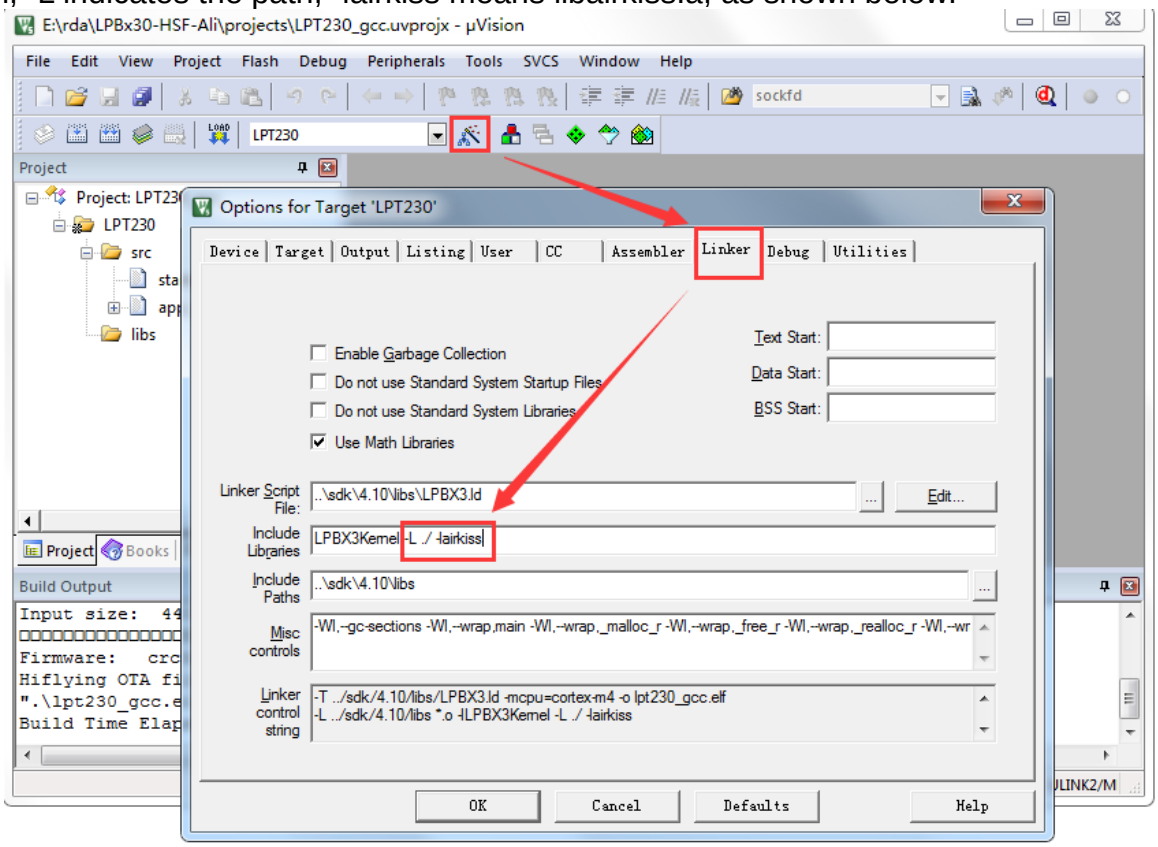
User function definition Convention
[Return Type] + USER_FUNC + [Function Name] + [Parameter]
Example:
void USER_FUNC test_func1(char *a);
“SER_FUNC” is function modifier symbol, Pls add USER_FUNC symbol for better compatibility. The program compiled in some platform may not work if don’t add “USER_FUNC”.

User add source code file

Add. C files, HSF-based source files must contain the <hsf.h> header file, including the header file, the source code can be called inside the HSF-based API functions; If you want to use the libc interface function, Pls # include related header file,For example, if you call the string manipulation function #inlcude <string.h>, call the time function #include <time.h>, etc.

User add third-party libraries

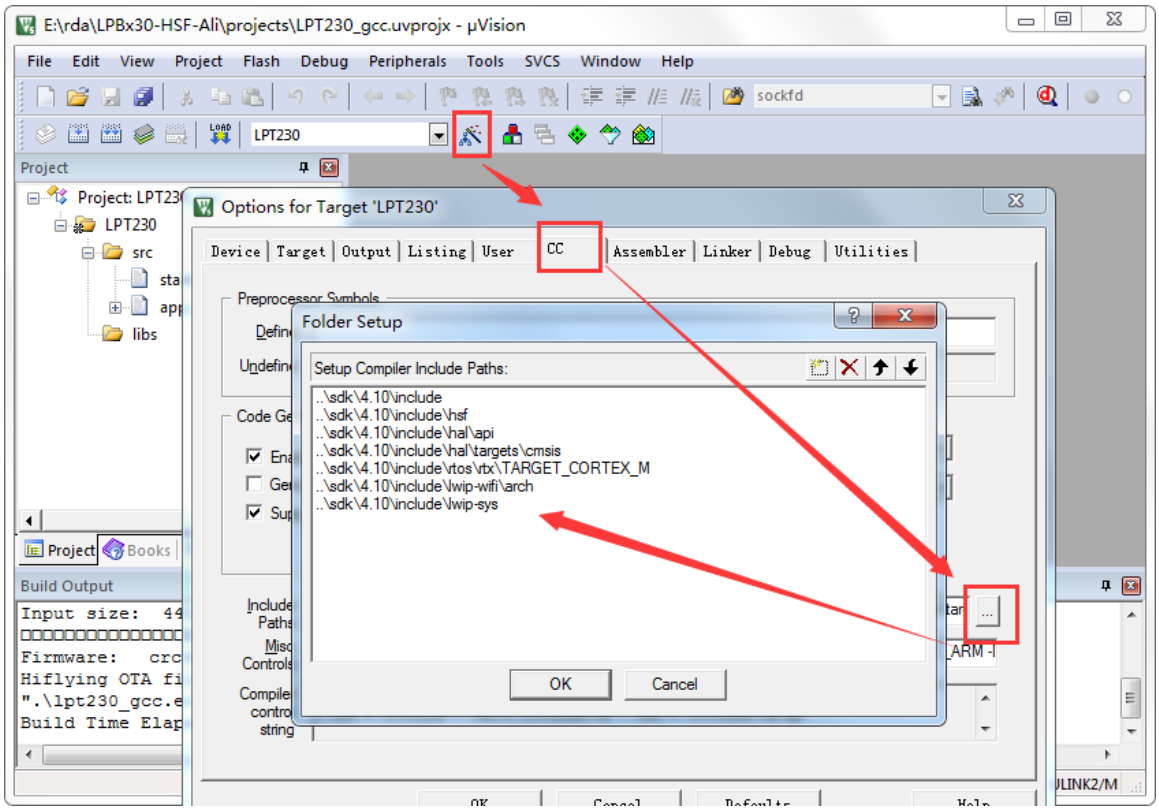
Add a third-party library, for example: Add libairkiss.a library file under the current path, -L indicates the path, -lairkiss means libairkiss.a, as shown below:



Due to the GCC compiler link feature，User may need to repeatedly link library files，For example: The a library calls b library function ， The b library calls a library function，The link needs to write：-la -lb -la, Otherwise it may be hints that the a library function called in the B library is not defined.

User add the header file directory

Add the header file directory based on the following diagram：



Attention：

SDK has been integrated Cyassl, MQTT, websocket, and other third-party libraries，It can be use directly in the project. For the use of method ,please refer to the following examples .

To better SDK upgrade, Pls don't add too much code in "app_main.c" file, Only need to add an entry function. Put all other source code files in your catalogue, User can also create a catalogue to put your own souce code in SRC catalogue, Then when upgrade, User only need modify the app_main.c file, Other SDK files all can be replace by the new ones.

3.4 Development Note

To avoid affect hi-flying factory test and cause unable to produce,Pls pay

attention to the following points during SDK development :

1.Do not affact the boot up automatic connection with router function. For example, Power on scan router or enter into smartlink will both affect hiflying factory test.

Solution: Add flag bit in flash or adopt reading GPIO level to distinguish bootup procedure. If adopt GPIO method, pls make sure that GPIO level is stable.

2.Must keep hi-flying AT command function and Don' change original AT command name.

3.Try to define user version query command as AT+APPVER. Do not use the original AT+VER and AT+LVER

4.Make sure the buad rate will not change after excute AT+RELD

3.5 Special Note

1.Due to the particularity of compiling environment，Please do not mixture declare of pointers and arrays in the code,If defined as an array char tmp[200]，Please do not use a pointer form declaration ,such as ：char *tmp，Should use char tmp[];；

4. RESOURCE ALLOCATION

4.1 1MB Flash Resource Assignment

0x0000 0000	SYSTEM_SECTOR (4KB) System
0x0000 1000	BOOT(12KB) Bootloader Area
0x0000 4000	BOOT_CONFIG(12KB) Systerm
0x0000 7000	USERPAGE (4KB) User save Area
0x0000 8000	USERPAGE_BACKUP (4KB) Backup Code Area
0x0000 9000	F_SETTING (4KB) Factory parameter area
0x0000 A000	**USER_BIN_FILE (4KB)** **hffile_userbin_write API actual physical address**
0x0000 B000	**USER_BIN_BACK_FILE (4KB)** **hffile_userbin_write API actual physical address backup area**
0x0000 C000	**CODE (588KB)** **Run code area**
0x0009 F000 0x000F C000	**OTA UPGRADE(372KB)** **OTA upgrade backup area** WEB (8KB) External web page
0x000F E000	UFLASH (8KB) User flash

4.2 4MB Flash Resource Assignment

0x0000 0000	SYSTEM_SECTOR (4KB) Systerm
0x0000 1000	BOOT(12KB) Bootloader
0x0000 4000	Reserved(32KB)
0x0000 C000	**CODE (1040KB)**
0x0011 0000	**OTA UPGRADE(1008KB)** **OTA upgrade backup area**
0x0020 C000	Reserved(464KB)
0x0028 0000	**WEB(512KB)** **Webpage area**

0x0030 0000	WEB_SCAN(16KB)
0x0030 4000	BOOT_CONFIG(12KB) Systerm
0x0030 A000	**USER_BIN_FILE (4KB) hffile_userbin_write API actual physical address**
0x0030 B000	**USER_BIN_BACK_FILE (4KB) hffile_userbin_write API actual physical address backup area**
0x0030 C000	Reserved(976KB)

BOOT：bootloader size, maximum support 12KB bootloader.

USER_BIN_FILE：File storage area, size 4KB. Data can be stored and written by SDK.

CODE：Code running area, maximum 558KB,Compilied Bin file can not exceed the size.

OTA UPGRADE：Running code upgrade backup area, maximum 372KB use for OTA update. OTA update file is compressed, so it's smaller than code size.

Unused area: Currently not used, Can use flash_write and flash_read address of API to operate and access(API address need +0x18000000. Such as operate 0xFC000 address of flash space, the API entrance address need adopt 0x180FC000)

4.3 Ram Resource

HF-LPX30 have around 80KB ram rest are available to use.

Global variable use static RAM, there is about 20KB left.

malloc and thread stack will both ocqupy the dynamic RAM resource, there is about 60KB left

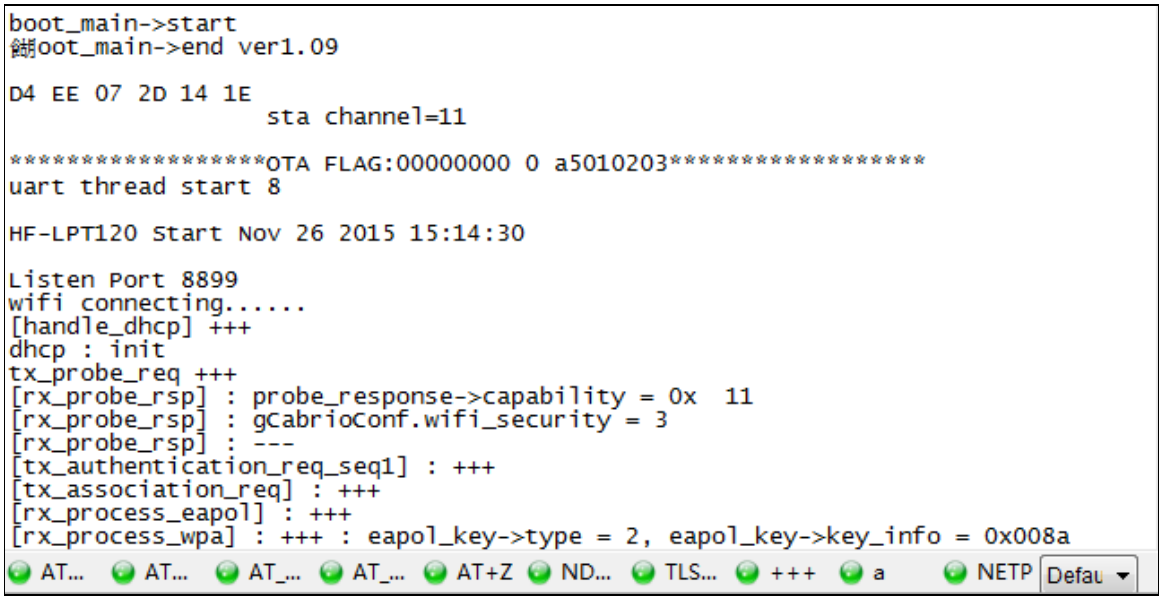
4.4 About 1M Flash Webpage

Due to the limited space of 1M Flash，Only provide 8K flash space to store web pages，A simple Chinese and English language webpage are available for upgrade in the sdk/tools directory. Upgrade method: Open the iweb.html page of the path of the module, Select "Upgrade customized webpage" to upgrade，If module IP is 192.168.11.100, Pls use explorer open the http://192.168.11.100/iweb.html to upgrade.

5. Serial Port Print Debug Information

If program want print debug information through serial port, The HSF provided two API function:u_printf and HF_Debug, In default situation, program cannot print any debug information by calling these two functions. Because default configuration is closed. It needs hfdbg_set_level(X) to open debug serial output, X represents the debug info level. Or use AT command to open debug info and output"AT+NDBGL=2,0" to Open, output "AT+NDBGL=0,0" to close. The debug information will output from serial port one(ModuleUART1_TX pin),see diagram below :

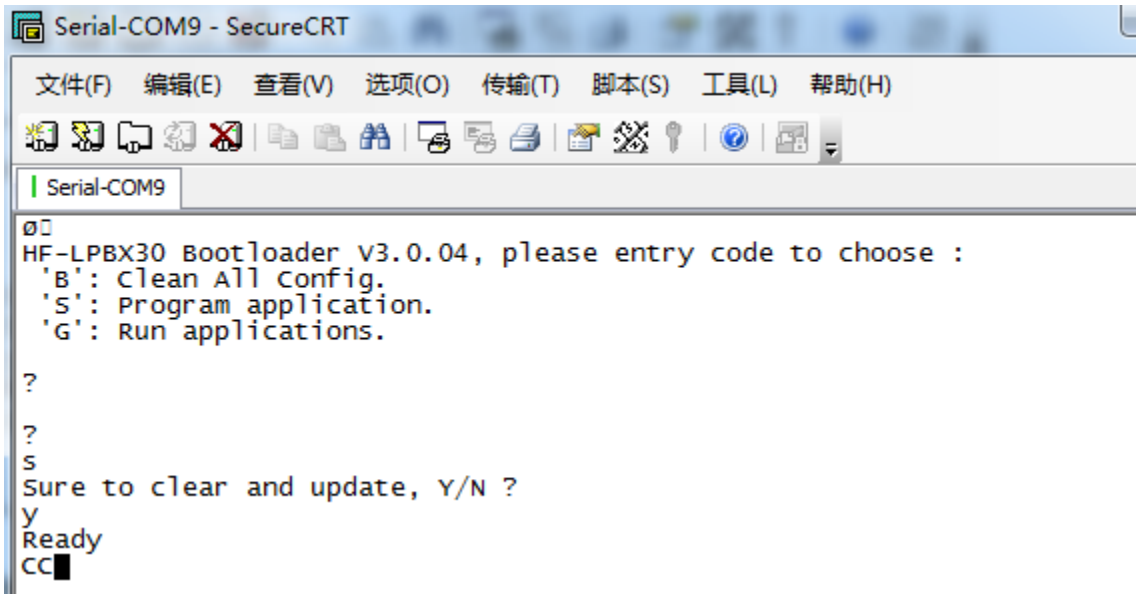
(Remark: Debug mode MUST be closed when the program is released)



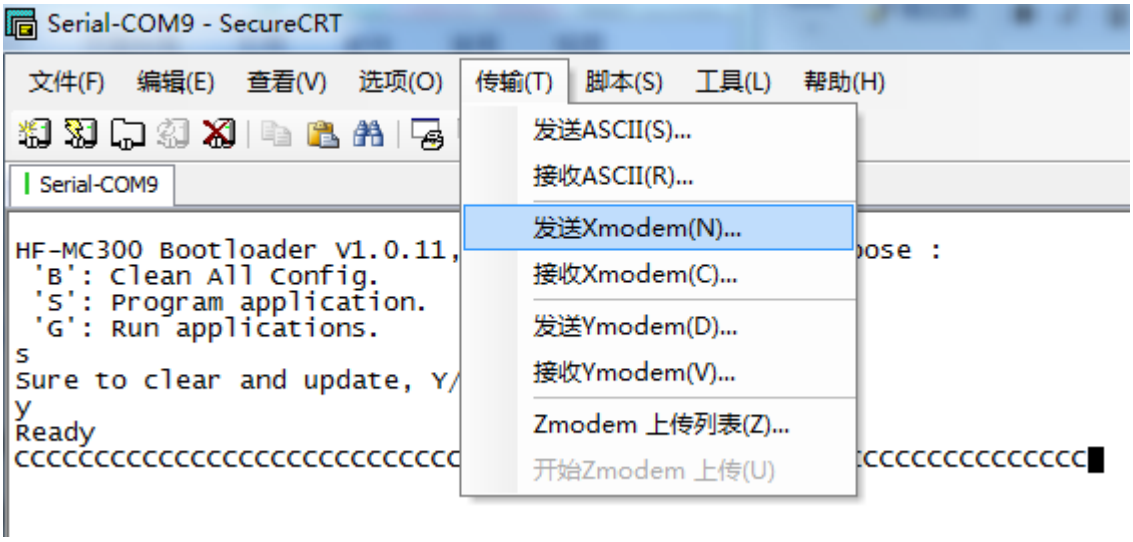
6. How to Update Firmware

6.1 By serial port

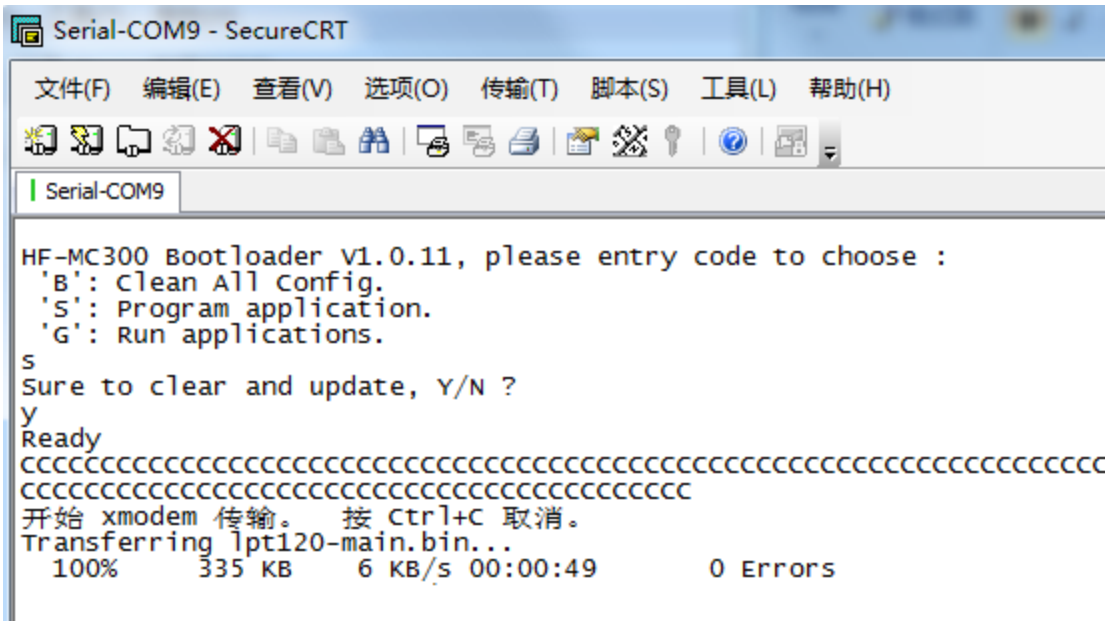
Step 1: Configure the baud rate to **230400** in **SecureCRT.** Press on module nReload button(driving down) to reset the module. And enter a ***space*** character after one second. Then can access moulde **Bootloader** to update. The successful screen shot as below:



Step 2: Enter command **'S'** and enter **'Y'** to upgrade file. Open **Xmode** of **SecureCRT** to transmit file. See diagram below:

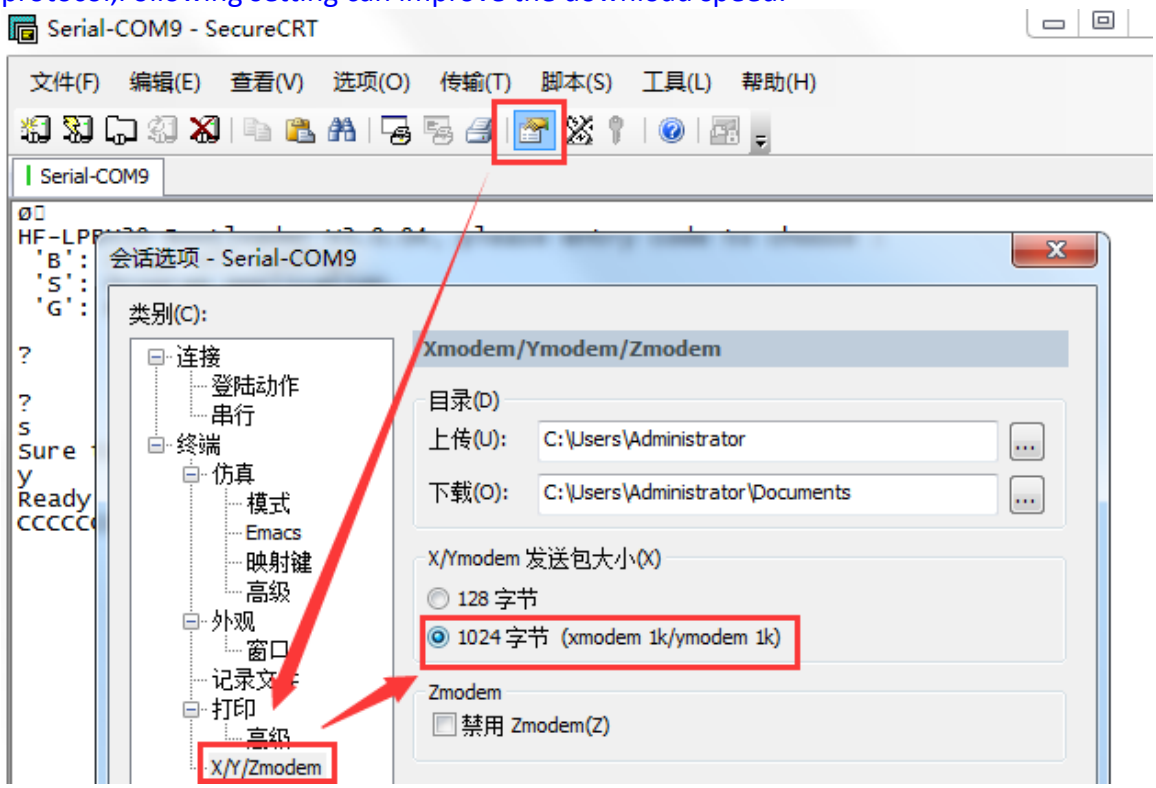


Step 3: Choose upgrade file ,The virtual machine complied without **upgrade** suffix **bin file.**



Step 4: Reset the module after finished transmission. If the software stuck during upgrade, please reset the module and software reenter into Bootloader to start ugrade.

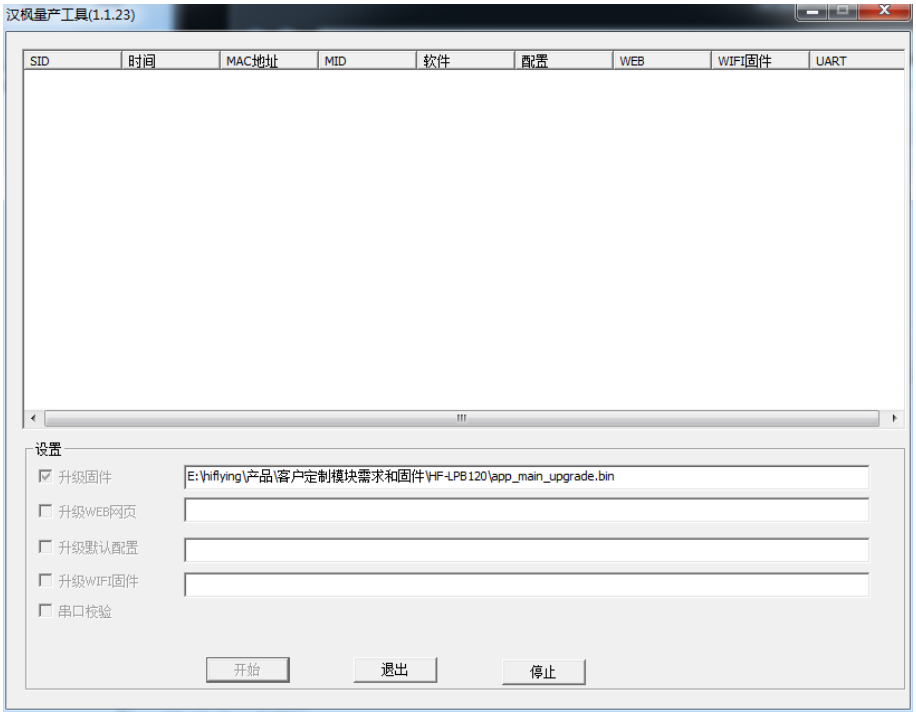
Note: LPBX30 series Bootloader support 1024 bytes tramsmission X modem protocol,Following setting can improve the download speed.



6.2 Through HF Production tool batch Upgrade

Step 1: Download the ***HF Production Upgrade Tool*** from Hi-flying official website.
http://gb.hi-flying.com/download_detail_dc/&downloadsId=1822d146-343d-4332-af8b-137c0fb4d967.html

Step 2 : **PC** connect with router, open the ***HFUpdate.exe*** *tool and* upload upgrade file(app_main_upgrade.bin), If can't opened, please install ***gtk2-runtime*** executing environment . Pls make sure the PC firewall has been closed during the use of this tool, And It's best to put the tool in the disk root directory (NO Chinese pathname)



Step 3 : Configre module, connect to the same router which connected with PC.

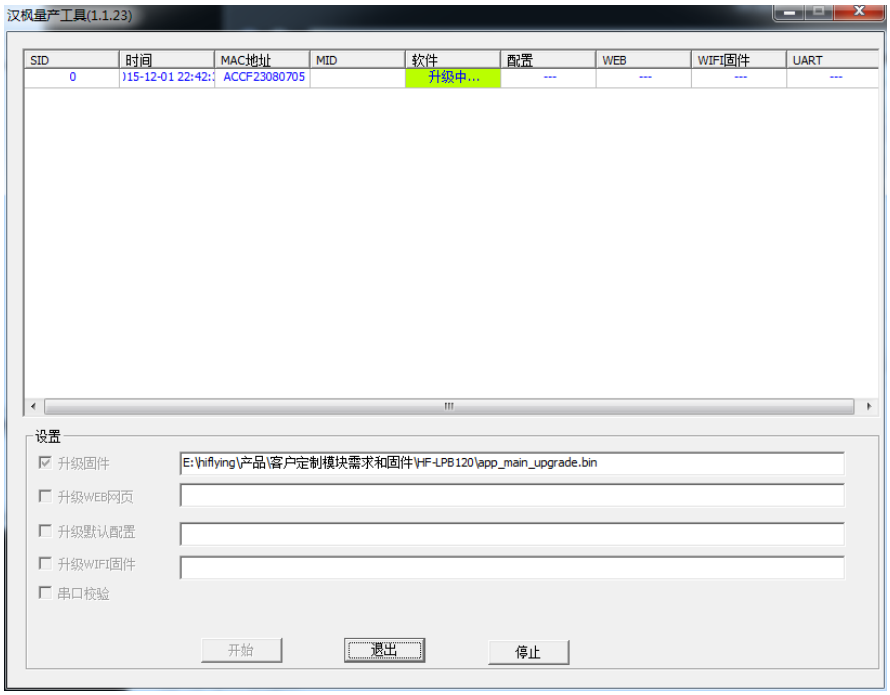
```
AT+WMODE
+ok=STA

AT+WSSSID
+ok=Sam401

AT+WSKEY
+ok=WPA2PSK,AES,gongyuhui

AT+WANN
+ok=DHCP,192.168.199.147,255.255.255.0,192.168.199.1
```

Step 4: Input command ***AT+OTA*** to start upgrade. PC will show the module information, wait until finished upgrade, Open debug information and can check the procedure of upgrade.



Step 5: After finished upgrade,Debug ouput show following info means finished upgrade, Then user can start test new program, The PC may keep reminder upgrading, please ignore it.

```
image file size=000534b4(341172) checksum=02120e54
*write boot image to flash!
**********************************D4 EE 07 2D 14 1E
                                                  sta channel=11

******************OTA FLAG:0505AA50 0 a5010202******************
*
 HF-LPT120 Start Nov 28 2015 01:52:06
```

Attention:

The file used for serial update and HF Production Upgrade Tool is different. The default firmware which with *upgrade*suffix can be upgrade by HF Production Upgrade Tool. app_main_upgrade.bin added CRC algorithm based on app_main. It's used to avoid upgrade wrong file.

7. Example

1. HSF-LPBx30 SDK provide relative function example, User can find relative project files and code via example file. EXAMPLE_USE_DEMO can be modified in example.h to choose compile which example and then example can be run after modification,The default compile module is HF-LPT230. The running example need to set message level to 2 through “AT+NDBGL=2 “, Then u_printf can print debug information through serial.

```
#define _H_EXAMPLE_H_H_H_

#define USER_AT_CMD_DEMO 1

#define USER_GPIO_DEMO  2

#define USER_TIMER_DEMO 3

#define USER_THREAD_DEMO 4

#define USER_CALLBACK_DEMO 5

#define USER_FILE_DEMO     6

#define USER_FLASH_DEMO     7

#define USER_SOCKET_DEMO    8

#define USER_IR_DEMO    9

#define USER_URL_DEMO   10

#define WIFI_TEST_DEMO  11

#define UPDATE_TEST_DEMO    12

#define SSL_TEST_DEMO   13

#define UART_OP_DEMO        14

#define USER_UPNP_DEMO  15

#define WEBSOCKET_TEST_DEMO 16

//通过下面可以选择不同的例子进行编译
#define EXAMPLE_USE_DEMO    USER_AT_CMD_DEMO
```

7.1 Create AT Command

The code in example/at/attest.c can help user understand how to add customized AT command.

7.2 Timer Control nReady Blinking

The Code in example/timer/timertest.c can help user understand the creation of thread and timer and other relative API function.

Running result nReady will blink at 1HZ frequency.

7.3 Serial Callback Control the Status of nLink

The code in example/netcallback/callbacktest.c. can help user be familiar with serial port send to API and serial callback handling mechanism.

Result: Through serial send “GPIO NLINK LOW” to module,nLink light will be low level, Send “GPIO NLINK HIGH” to module,nLink light will be high level, Send”GPIO NLINK FLASH” to module,nLink light will blink at 1HZ frequency

7.4 Wireless OTA

The Code in example/update/updatetest.c can help user be familier with API related to wireless OTA.
Result: serial port use command ‘AT+UPGRADESW=http://192.168.1.1/update.bin’ to start wireless upgradation.

<The end>