Date	Version	Notes
2023-08-22	1.1.00	Initial release

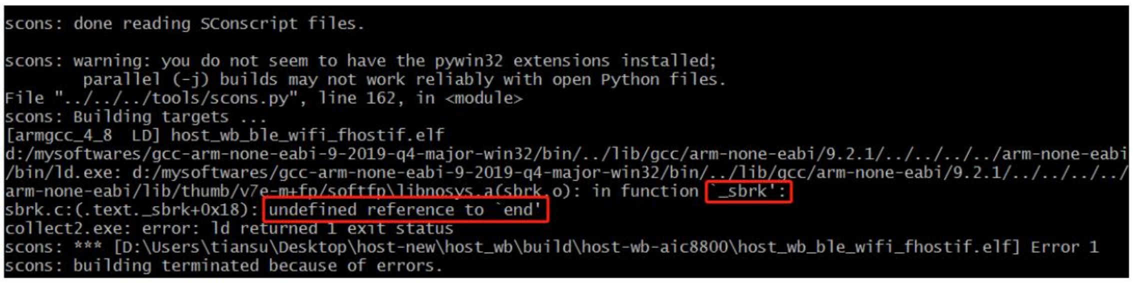
FAQ

This document focuses on describing the problems that customers may encounter during the AIC8800x-SDK development process and their solutions. Customers can refer to this document during the development process.

1. **Unable to start**

```
#The following log shows boot abort
Bootrom [Aug 4 2022, gb08afb5]
Copyright (C) 2018-2022 AICSemi Ltd.

RstCause:0000,c0,Boot:0d,c
Mcu mode
boot abort: -1

# Common is -1, flash is not initialized or the configuration is erased, need to re-
initialize flash f 1 3 1 2 1
f 3
Then re-burn the bin file
```

2. **Use standard library functions**



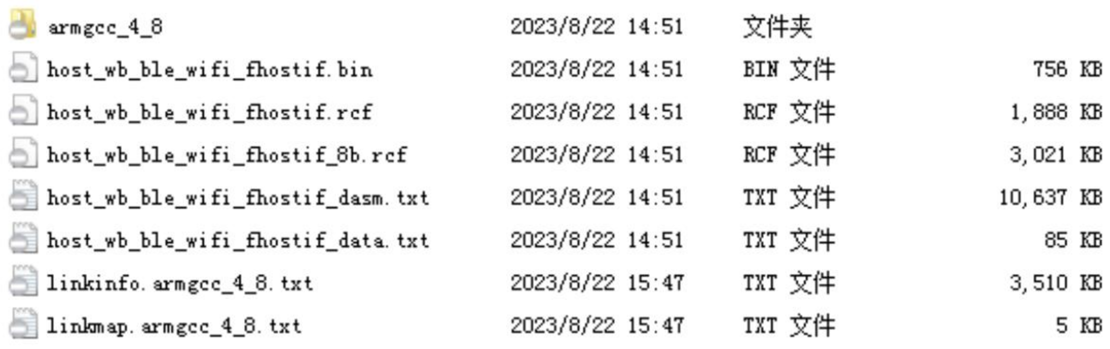
```
scons: done reading SConscript files.

scons: warning: you do not seem to have the pywin32 extensions installed;
        parallel (-j) builds may not work reliably with open Python files.
File "../../../tools/scons.py", line 162, in <module>
scons: Building targets ...
[armgcc_4_8  LD] host_wb_ble_wifi_fhostif.elf
d:/mysoftwares/gcc-arm-none-eabi-9-2019-q4-major-win32/bin/../lib/gcc/arm-none-eabi/9.2.1/../../../../arm-none-eabi
/bin/ld.exe: d:/mysoftwares/gcc-arm-none-eabi-9-2019-q4-major-win32/bin/../lib/gcc/arm-none-eabi/9.2.1/../../../../
arm-none-eabi/lib/thumb/v7e-m+fp/softfp\libnosys.a(sbrk.o): in function `_sbrk':
sbrk.c:(.text._sbrk+0x18): undefined reference to `end'
collect2.exe: error: ld returned 1 exit status
scons: *** [D:\Users\tiansu\Desktop\host-new\host_wb\build\host-wb-aic8800\host_wb_ble_wifi_fhostif.elf] Error 1
scons: building terminated because of errors.
```

In order to save the space for compiling and generating firmware code, this compilation tool does not use the C language standard library for compilation by default. If the user's application code uses the standard library function interface, the above compilation error will appear. You can open the standard library for compilation by adding compilation instructions.

```
./build_wifi_case.sh STDLIB=on -j8
```

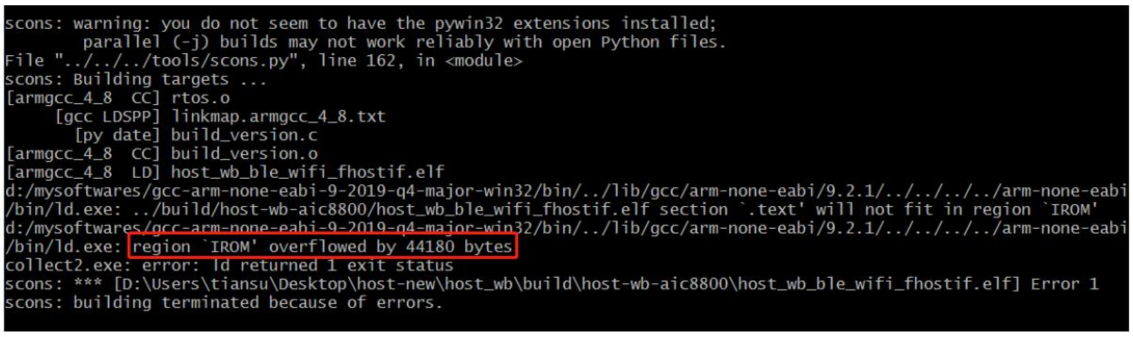
3. Compile and generate directory and disassemble debug

After different series of chips and targets are compiled, target files will be generated in the build path according to certain rules, including bin, link files, map files, disassembly files, etc. For example, when compiling 8800M/target_ble_wifi_fhostif, the build/host-wb-aic8800 directory will be generated. If **CM4 Fault** Handler appears during program operation, users can view the disassembly file in this directory.



```
armgcc_4_8                          2023/8/22 14:51   文件夹
host_wb_ble_wifi_fhostif.bin        2023/8/22 14:51   BIN 文件        756 KB
host_wb_ble_wifi_fhostif.rcf        2023/8/22 14:51   RCF 文件      1,888 KB
host_wb_ble_wifi_fhostif_8b.rcf     2023/8/22 14:51   RCF 文件      3,021 KB
host_wb_ble_wifi_fhostif_dasm.txt   2023/8/22 14:51   TXT 文件     10,637 KB
host_wb_ble_wifi_fhostif_data.txt   2023/8/22 14:51   TXT 文件         85 KB
linkinfo.armgcc_4_8.txt             2023/8/22 15:47   TXT 文件      3,510 KB
linkmap.armgcc_4_8.txt              2023/8/22 15:47   TXT 文件          5 KB
```

4. **Code space overflow**



```
scons: warning: you do not seem to have the pywin32 extensions installed;
        parallel (-j) builds may not work reliably with open Python files.
File "../../../tools/scons.py", line 162, in <module>
scons: Building targets ...
[armgcc_4_8  CC] rtos.o
     [gcc LDSPP] linkmap.armgcc_4_8.txt
       [py date] build_version.c
[armgcc_4_8  CC] build_version.o
[armgcc_4_8  LD] host_wb_ble_wifi_fhostif.elf
d:/mysoftwares/gcc-arm-none-eabi-9-2019-q4-major-win32/bin/../lib/gcc/arm-none-eabi/9.2.1/../../../../arm-none-eabi
/bin/ld.exe: ../build/host-wb-aic8800/host_wb_ble_wifi_fhostif.elf section `.text' will not fit in region `IROM'
d:/mysoftwares/gcc-arm-none-eabi-9-2019-q4-major-win32/bin/../lib/gcc/arm-none-eabi/9.2.1/../../../../arm-none-eabi
/bin/ld.exe: region `IROM' overflowed by 44180 bytes
collect2.exe: error: ld returned 1 exit status
scons: *** [D:\Users\tiansu\Desktop\host-new\host_wb\build\host-wb-aic8800\host_wb_ble_wifi_fhostif.elf] Error 1
scons: building terminated because of errors.
```

The compilation tool configures the memory, data, and code storage addresses through map files. In the SDK, different targets will link different map files when compiled. For the above problem, you need to modify the IROM space limit in the corresponding map file.

```
MEMORY {
        IROM (rwx)              : ORIGIN = CODE_START_ADDR, LENGTH = 960K #Based on the maximum
small modification
        DRAM (rwx)              : ORIGIN = 0x00100000, LENGTH = 384K
        USB_MEM (rwx) : ORIGIN = 0x0019F000, LENGTH = (4K - 0x100)
        IRAM (rwx)              : ORIGIN = 0x0019FF00, LENGTH = 0x100
         ...
}
```

Note that IROM stores code instructions, and IROM data is directly stored in flash, which is only limited by the size of the flash partition and can be modified by the user. Other areas such as DRAM and IRAM will directly affect the allocation of RAM memory addresses, and users cannot modify them directly.

5. **How to confirm the corresponding map file**

When compiling, how different targets link maps depends on the set rules. You can check the compilation instructions in the compilation script in plf/aic8800x/config/SAdditions

Or, compare directly in the build directory generated by the compilation

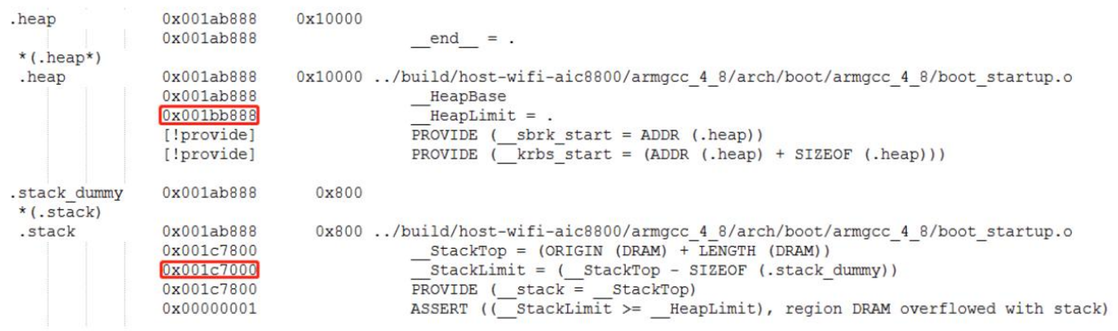
```
For example, 8800M/target_ble_wifi_fhostif
links the map_cm4_ble_fhostif_wifi.txt file, the compiled file is located in build/host-wb-aic8800, and the map
file is copied to linkmap.armgcc_4_8.txt
```

6. **Adjust HEAP**

In the map file configuration, the heap is usually set in the DRAM segment, and the DRAM segment is placed in the system configuration segment, heap, and stack in sequence. Users can adjust the heap size by adding compilation instructions. When the heap length covers the stack, a compilation error will be reported. Users can use this to determine how to set the maximum heap.

```
./build_wifi_case.sh HEAP_SIZE=0x20000 -j8
```

In addition, you can also check the linkinfo.armgcc_4_8.txt in the build directory to make sure that HeapLimit is smaller than StackLimit when adjusting.



```
.heap                0x001ab888     0x10000
                     0x001ab888                 __end__ = .
 *(.heap*)
 .heap               0x001ab888     0x10000 ../build/host-wifi-aic8800/armgcc_4_8/arch/boot/armgcc_4_8/boot_startup.o
                     0x001ab888                 __HeapBase
                     0x001bb888                 __HeapLimit = .
                     [!provide]                 PROVIDE (__sbrk_start = ADDR (.heap))
                     [!provide]                 PROVIDE (__krbs_start = (ADDR (.heap) + SIZEOF (.heap)))

.stack_dummy         0x001ab888       0x800
 *(.stack)
 .stack              0x001ab888       0x800 ../build/host-wifi-aic8800/armgcc_4_8/arch/boot/armgcc_4_8/boot_startup.o
                     0x001c7800                 __StackTop = (ORIGIN (DRAM) + LENGTH (DRAM))
                     0x001c7000                 __StackLimit = (__StackTop - SIZEOF (.stack_dummy))
                     0x001c7800                 PROVIDE (__stack = __StackTop)
                     0x00000001                 ASSERT ((__StackLimit >= __HeapLimit), region DRAM overflowed with stack)
```

7. **Modify the system clock**

Enter the target directory that needs to be compiled, such as open target_ble_wifi_fhostif/tgt_cfg_hw/hw_cfg.h, and modify the macro

Configuration

```
/**
 *   System initial clock configure
 *     use values under CLK_CFG_MAX defined in sysctrl_api.h
 */
#define CONFIG_INITIAL_SYSTEM_CLOCK                    CLK_CFG_D240S240P120F60
```

8. **Modify the wifi radio frequency power**

Enter the target directory that needs to be compiled, such as open target_ble_wifi_fhostif/tgt_cfg/tgt_cfg_wifi.h, and modify the power parameters.

The set power parameters need to be confirmed by the RF instrument.

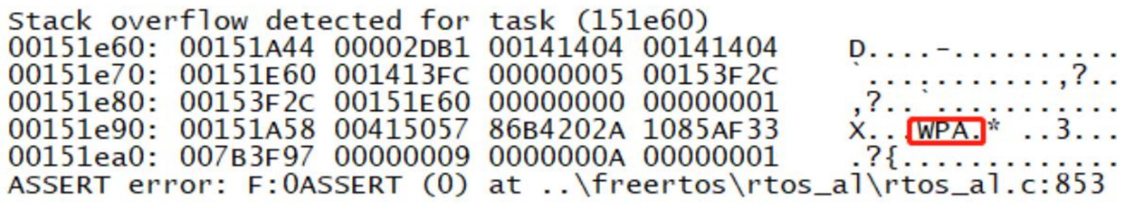
9. **No host buffer appears during the streaming process**

```
# wifi protocol stack log
_no host buffer:0
_host buffer recovery
```

When the chip is running, the serial port is prone to the above logs. When the two logs appear in pairs, it means that the protocol stack buff will be automatically restored.

No problem will occur, users can ignore

10. **Task overflow**



```
Stack overflow detected for task (151e60)
00151e60: 00151A44 00002DB1 00141404 00141404    D....-..........
00151e70: 00151E60 001413FC 00000005 00153F2C    `...........,?..
00151e80: 00153F2C 00151E60 00000000 00000001    ,?..`...........
00151e90: 00151A58 00415057 86B4202A 1085AF33    X...WPA.* ..3...
00151ea0: 007B3F97 00000009 0000000A 00000001    .?{.............
ASSERT error: F:0ASSERT (0) at ..\freertos\rtos_al\rtos_al.c:853
```

As shown in the log above, insufficient memory is allocated when creating the task, causing the task to fail to run normally. You need to increase the memory allocated when creating the task.

live

```
# rtos.h is the configuration of system tasks, generally do not change
/// Definitions of the different FHOST task stack size requirements
enum
{
        TASK_STACK_SIZE_CONSOLE                    = 512,
        TASK_STACK_SIZE_TEST                       = 2048,
        TASK_STACK_SIZE_BT_TASK                    = 1024,
        TASK_STACK_SIZE_ASIO                       = 2048,
        TASK_STACK_SIZE_AUDIO                      = 2048,
        TASK_STACK_SIZE_BLE_TASK_ONLY              = 512,
         ......
};
```

When users create their own tasks, it is recommended to add their own task id in rtos_al.h so that they can trace back when problems occur.

```
/**
  * RTOS task identifier
  */
enum rtos_task_id {
        IDLE_TASK                    = 0,
        TMR_DAEMON_TASK = 1,
        CONSOLE_TASK                 = 2,
         ......
        TCP_WAKEUP_TASK = 35,
        WLAN_CONN_TASK               = 36,
        MAX_TASK,
        UNDEF_TASK                   = 255,
};
```