



XR806 Flash 布局方案 开发指南

版本号：1.0

发布时间：2020-11-12

版本历史

版本	日期	责任人	版本描述
1.0	2020-11-12	AWA 1029	创建文档。

目录

版本历史.....	i
目录.....	ii
1 前言.....	1
1.1 文档简介.....	1
1.2 目标读者.....	1
1.3 适用范围.....	1
1.4 文档约定.....	1
1.4.1 标志说明.....	1
1.4.2 地址与数据描述方法约定.....	1
1.4.3 数值单位约定.....	2
2 概述.....	3
2.1 背景说明.....	3
2.2 规格特性.....	3
2.3 文件位置.....	3
3 技术说明.....	5
3.1 整体布局.....	5
3.1.1 Image 1.....	5
3.1.2 Sysinfo 区域.....	5
3.1.3 OTA 区域.....	6
3.1.4 Image 2/compres image.....	6
3.1.4.1 ping pong 模式.....	6
3.1.4.2 压缩模式.....	6
3.2 Cfg 文件.....	7
3.2.1 Bin 文件布局说明.....	8
4 应用说明.....	9
4.1 应用简述.....	9
4.2 配置说明.....	9
5 示例说明.....	10
5.1 示例简介.....	10
5.1.1 获取方式.....	10
5.1.2 准备工作.....	10

5.1.3 操作步骤.....	10
5.1.4 效果展示.....	10
6 常见问题.....	12
6.1 Bin 文件超出 max_len 的解决.....	12
6.2 文件总长超过 max_size 的解决.....	13

1 前言

1.1 文档简介

此文档从较为宏观的角度来说明 XR806 SDK 的 flash 所使用的布局情况，主要介绍整个 image 和 sysinfo 以及 OTA 区域之间的位置关系。

1.2 目标读者

XR806 SDK 用户。

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

符号	例子	说明
Ox	0x0200, 0x79	地址或数据以 16 进制表示。
Ob	0b010, 0b00 000 111	数据采用二进制表示(寄存器描述除外)。
X	00X, XX1	数据描述中，X 代表 0 或 1。 例如，00X 代表 000 或 001；XX1 代表 001, 011, 101 或 111。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

类型	符号	对应数值
数据容量（如 NAND 容量）	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率，数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

2 概述

2.1 背景说明

XR806 SDK 将编译生成的 bin 文件按照固定的结构进行打包生成 image 文件,然后用户再将其烧写到 flash 上启动运行。Flash 资源有限,为了腾出更多的 flash 空间,合理地分配 image 和 OTA 等区域很重要。

2.2 规格特性

Flash 布局支持以下功能特性。

表 2-1 Flash 布局功能特性

规格类型	支持规格	规格描述	备注
文件布局	手动布局	可以使用 max_len 设置 bin 文件的最大占用空间来确认其存储位置,亦可使用 flash_offs 字段设置 bin 文件的 flash 偏移地址来确认其储位置 推荐使用 max_len 字段进行布局	
	自动布局	不对 bin 文件的 flash 偏移和最大占用空间作限制,软件会自动进行排布	
OTA 布局	OTA address	自定义设置 OTA 区域的 flash 偏移地址	
	OTA size	自定义设置 OTA 区域的占用大小	
OTA 升级方式	Ping-pong 方式	两个 image 区域都保存完整固件,轮流进行升级	
	压缩方式	仅第一个 image 区域保存完整固件,第二个区域保存压缩固件,升级时将其解压至第一个区域进行使用	

2.3 文件位置

以 SDK 包为根目录,flash 布局涉及到的主要文件位置如下。

表 2-2 flash 布局的文件位置

组件名	文件分类	文件位置
Flash layout	配置文件	./project/image_cfg/

关键文件说明如下。

表 2-3 flash 布局的文件说明

文件名	文件说明
image.cfg	Flash 布局的通用配置文件

文件名	文件说明
image_img_xz.cfg	压缩模式的 OTA 升级方式所使用的 flash 布局配置文件



说明

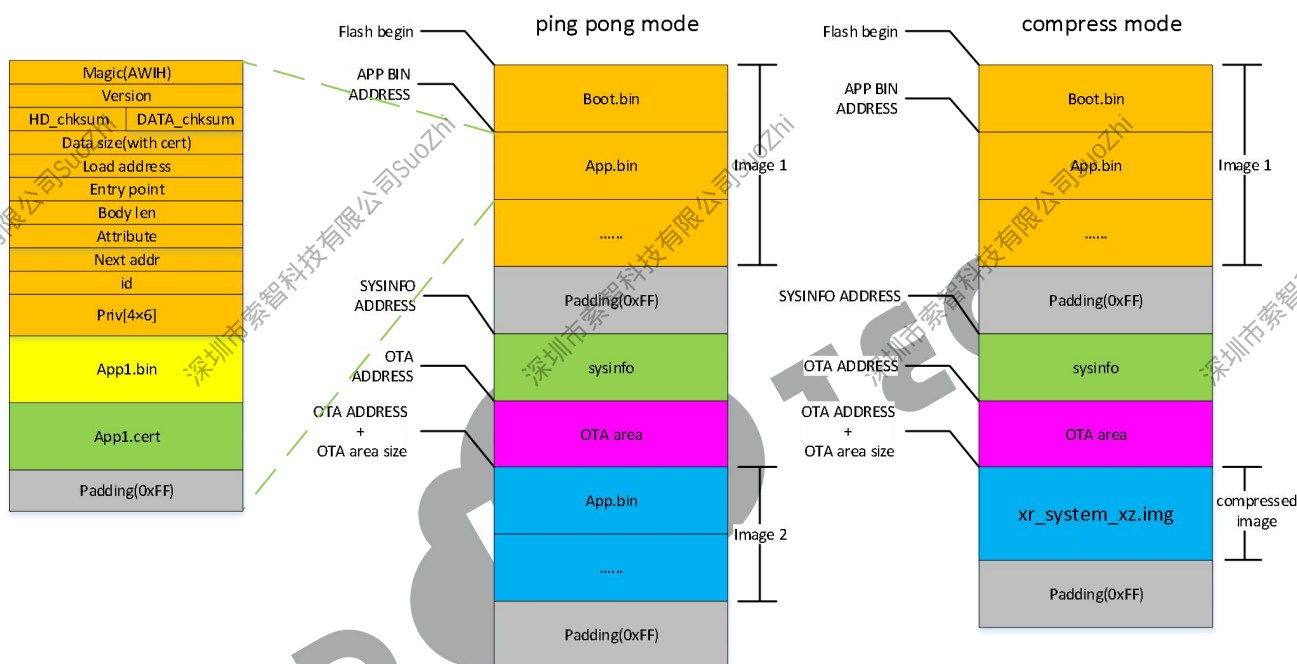
XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

3 技术说明

3.1 整体布局

整个 flash 的布局 and OTA 升级的模式有关，ping pong 模式下，flash 占用会比在压缩模式下大。
两种模式下 flash 的布局如下图所示。

图 3-1 Flash Layout 示例



由上图可知，整个 flash 可以划分为 image1、sysinfo、OTA area、image2 (compressed image) 四个区域，其中 image1、image2 是有多个 bin 组成，每个 bin 又由头部信息 (header)、bin 文件数据 (body) 以及证书文件 (tailer) 组成。

下面详细介绍每个区域的意义和使用。

3.1.1 Image 1

Image 1 是一个完整的镜像，里面包含了 boot.bin、app.bin...xxx.bin，从 flash 的 0 地址开始。

Image 1 区域之后一般都会有一部分 padding 区域，该区域是后期给 OTA 升级预留的空间。后期代码增加，功能扩展，会导致 image 大小增加，所以需要给后期升级扩展预留一定的空间。

padding 的最大值需要在项目开发前确定好，即需要先确定好 image 的最大值，否则容易出现后期开发扩展时，image 超过最大值而覆盖 sysinfo 数据。假设正常编译的 image 是 900K (包含 bootloader)，则可定义 image 的最大值为 1020K，后期开发的 image 就不能超过 1020K。

3.1.2 Sysinfo 区域

Sysinfo 是一段用于存储用户自定义数据的区域，其大小默认被设置为 4K，位置被设置为 1020K (1024K - 4K) 的地址，默认是放在 OTA 区域之前的位置。Sysinfo 的位置和大小是被定义在工程目录下的 prj_config.h 文件中，用户可根据需求进行修改，但是需要注意不能与 image 区域和 OTA 区域重叠。

```
/* sysinfo start address */
#define PRJCONF_SYSINFO_ADDR          ((1024 - 4) * 1024)

/* sysinfo size */
#define PRJCONF_SYSINFO_SIZE          (4 * 1024)
```

3.1.3 OTA 区域

OTA area 用于存储 OTA 使用的参数，该区域的地址 OTA ADDRESS 的配置是在 cfg 文件中指定的，默认是指定为 1024K，即 1M 的偏移地址，大小为 4K。如下标红位置即配置 OTA ADDRESS 和大小；

```
{
    "magic"      : "AWIH",
    "version"    : "0.5",
    "OTA"        : {"addr": "1024k", "size": "4K"},
    #if (__CONFIG_OTA_POLICY == 0x01)
        "image"   : {"max_size": "1020K", "xz_max_size": "600K"},
    #else
        "image"   : {"max_size": "1020K"},
    #endif
    .....
}
```

需要注意 OTA 地址与 flash 大小之间的关系：

1. 如果没有开启 OTA 功能，则用户只需要保证 image 大小小于 max_size 即可，因为此时，OTA ADDRESS 之后的区域都不存在，flash 中只有 image1、padding 和 sysinfo。
2. 如果开启 OTA 功能，则用户需要保证在 OTA ADDRESS 之后仍有一个足够的空间进行 OTA 升级用，该空间需要保证能够存下一个 image 或一个压缩的 image。假设一个 image 大小为 900K（包含 bootloader），image 的最大值定义为 1020K，flash 大小为 4M，那么 OTA 地址设置在 1M 到 3M 之间的区域都是可以的，因为 OTA 地址之后还有足够的空间来存放 image。

3.1.4 Image 2/compressed image

此区域根据 OTA 升级模式不同而保存不同的 image 类型，即 ping pong 模式下保存 image 2，压缩模式下保存 compressed image。

3.1.4.1 ping pong 模式

Image 2 是不包含 boot.bin 的，其起始位置决定于 OTA 的起始位置和 OTA 区域大小，所以 image 2 的 app.bin 是放在 OTA 的起始位置加 OTA 区域大小后。

3.1.4.2 压缩模式

压缩模式下，flash 的分布和 ping pong 模式下大部分相同，唯一不同的就是 Image 2。ping pong 模式下 Image 2 存放的是一个可运行的 image（去除了 bootloader 的 image）；而压缩模式下，Image 2 存放的是一个压缩的 image。所以在压缩模式下，Image 2 需要的空间会比 ping pong 模式下小很多。

3.2 Cfg 文件

Cfg 文件包含整个 image 的组织结构，配置文件使用 Json 标记语言对 bin 文件的头部信息进行配置。

如下是一个常用的 cfg 文件，标为红色的部分配置了每一个 bin 文件的起始 flash 偏移地址。以下是一个经过预处理的 cfg 文件，详细的 cfg 文件可参考 image.cfg。

```
{
  "magic": "AWIH",
  "version": "0.5",
  "OTA": {"addr": "1024K", "size": "4K"},
  "image": {"max_size": "1020K"},
  "section":
  [
    {"id": "0xa5ff5a00", "bin": "boot_40M.bin", "cert": "null", "max_len": "16K", "sram_offs":
    "0x00230000", "ep": "0x00230101", "attr": "0x01"},
    {"id": "0xa5fe5a01", "bin": "app.bin", "cert": "null", "max_len": "150K", "sram_offs":
    "0x00201000", "ep": "0x00201101", "attr": "0x1"},
    {"id": "0xa5fd5a02", "bin": "app_xip.bin", "cert": "null", "max_len": "700K", "sram_offs":
    "0xffffffff", "ep": "0xffffffff", "attr": "0x2"},
    {"id": "0xa5fa5a05", "bin": "wlan_bl.bin", "cert": "null", "max_len": "4K", "sram_offs": "0xffffffff",
    "ep": "0xffffffff", "attr": "0x1"},
    {"id": "0xa5f95a06", "bin": "wlan_fw.bin", "cert": "null", "max_len": "128K", "sram_offs":
    "0xffffffff", "ep": "0xffffffff", "attr": "0x1"},
    {"id": "0xa5f85a07", "bin": "wlan_sdd_40M.bin", "cert": "null", "max_len": "1K", "sram_offs":
    "0xffffffff", "ep": "0xffffffff", "attr": "0x1"},
    {}
  ]
}
```

各字段含义如下：

表 3-1 cfg 配置文件字段说明

主字段	子字段	字段描述
magic	无	image 魔数，表头标识，固定为 AWIH
version	无	版本号
OTA	addr	设置 OTA 区域的 flash 偏移地址
	size	设置 OTA 区域的占用大小
image	max_size	image 的最大值（image1 和 image2 区域的最大值）
	xz_max_size	ota 压缩模式下 compressed image 的最大值，非压缩模式下，禁止使用该字段
section	id	bin 文件的 id，列举如下： boot_id : 0xa5ff5a00 app_id : 0xa5fe5a01 app_xip_id : 0xa5fd5a02 wlan_bl_id : 0xa5fa5a05

主字段	子字段	字段描述
		wlan_fw_id : 0xa5f95a06 wlan_sdd_id : 0xa5f85a07 app_psram_id : 0xa5f65a09
	bin	bin 文件名字
	cert	该 bin 文件对应的证书文件，null 表示没有添加证书
	max_len	该 bin 文件存放在 flash 中可占用的最大空间； 亦可使用 flash_offs 字段，表示该 bin 文件存放在 flash 中的位置偏移；如果上述两个字段都不填写，则软件自动放在上一个文件结束的位置，以 1024 byte 对齐
	sram_offs	该 bin 文件加载到 SRAM 中的物理地址，0xFFFFFFFF 为无效值，使用时忽略
	ep	该 bin 文件的 ENTRY POINT，即入口函数地址，0xFFFFFFFF 为无效值，使用时忽略
	attr	表示该 bin 文件的属性 [1:0]：文件类型，00-普通文件，01-可执行文件，10-XIP 文件，11-保留 [3:2]：加密类型，00-不加密，x1-带证书，1X-flash 加密 [4]：是否压缩 bin 文件，0-不是，1-是 [5]：是否安全文件，0-不是，1-是 [31:6]：保留

3.2.1 Bin 文件布局说明

Boot.bin 固定设置为 0 地址开始，其 max_len 需要设置为 4K 对齐，原因是 OTA 升级时 boot.bin 不会被擦除，之后的区域起始地址需要以最小的可擦除块大小（一般为 4K）对齐。Boot.bin 的 max_len 默认设置为 16K。

其他的 bin 文件的地址并没有强制性的要求，只要没有互相重叠并且不覆盖到 sysinfo 和 OTA 区域，都是可以正常运行的。

如果用户希望将 bin 文件固定在某个 flash 偏移地址进行布局，则使用 flash_offset 字段比较合适；如果用户比较关心每个 bin 文件的最大占用空间，则使用 max_len 字段比较合适；如果用户两者都不关心，可以两个字段都不使用，软件会自动排布。

建议使用 max_len 字段进行排布。

4 应用说明

4.1 应用简述

Flash 布局模块已经内嵌到 XR806 SDK，应用步骤如下：

1. 确认/检查所用 Flash 设备是否能正常运行（Flash 模块设备配置可参考《XR806 Flash 驱动开发指南》文档）。
2. 根据用户需求，适当修改 cfg 配置文件或者直接使用默认配置（参见“2.3 文件位置”章节）。

4.2 配置说明

Flash 布局模块无需修改任何配置项即可直接使用。

如果用户需要使用自己修改的 cfg 配置文件，可以在对应的项目工程目下的/gcc/Makefile 中添加自定义 cfg 路径。修改方式如下：

```
# image config file
# - relative to "../image/${CONFIG_CHIP_TYPE}/", eg. "../image/xr806/"
# - define your own "IMAGE_CFG" to override the default one
IMAGE_CFG := my_image.cfg
```

5 示例说明

本章节默认的 `cfg` 配置文件使用为例，简要说明 `flash` 布局模块的使用。

5.1 示例简介

本例以 `hello_demo` 的编译使用 `cfg` 配置文件的流程，简单介绍 `flash` 布局模块的使用方法。

5.1.1 获取方式

`flash` 布局模块在 `hello_demo` 示例工程中有实现，位于 `XR806 SDK` 的 `/project/demo/hello_demo` 目录，以下此示例工程简称为 `hello_demo` 示例工程。



说明

`XR806 SDK` 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

5.1.2 准备工作

`hello_demo` 示例工程的硬件准备如下：

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 `Uart0` 插针，用于 `console` 控制台的输入输出。
3. PC 机：用于镜像烧录和 `console` 控制的输入输出。

`hello_demo` 示例工程的软件准备，包括烧写工具、代码编译和烧写操作，请参见《`XR806_SDK` 快速入门指南》。

5.1.3 操作步骤

在 `SDK` 顶层目录配置好需要编译的工程为 `hello_demo` 后，执行编译命令 `make build` 即可

5.1.4 效果展示

在完成编译后，可以看到以下打印，即说明 `cfg` 配置文件可正常使用。

```
cfg string:
/*
 *
 * Automatically generated file; DO NOT EDIT.
 * XR806 SDK Configuration
 *
 */
{
    "magic": "AWIH",
    "version": "0.5",
```

```
"OTA" : {"addr": "1024K", "size": "4K"},
"image" : {"max_size": "1020K"},
"section" :
[
    {"id": "0xa5ff5a00", "bin": "boot_40M.bin", "cert": "null", "max_len": "16K", "sram_offs":
"0x00230000", "ep": "0x00230101", "attr": "0x01"},
    {"id": "0xa5fe5a01", "bin": "app.bin", "cert": "null", "max_len": "150K", "sram_offs":
"0x00201000", "ep": "0x00201101", "attr": "0x1"},
    {"id": "0xa5fd5a02", "bin": "app_xip.bin", "cert": "null", "max_len": "700K", "sram_offs":
"0xffffffff", "ep": "0xffffffff", "attr": "0x2"},
    {"id": "0xa5fa5a05", "bin": "wlan_bl.bin", "cert": "null", "max_len": "4K", "sram_offs": "0xffffffff",
"ep": "0xffffffff", "attr": "0x1"},
    {"id": "0xa5f95a06", "bin": "wlan_fw.bin", "cert": "null", "max_len": "128K", "sram_offs":
"0xffffffff", "ep": "0xffffffff", "attr": "0x1"},
    {"id": "0xa5f85a07", "bin": "wlan_sdd_40M.bin", "cert": "null", "max_len": "1K", "sram_offs":
"0xffffffff", "ep": "0xffffffff", "attr": "0x1"},
    {}
]
}
```

Flash Layout:

sec bin 0 boot_40M.bin	:	flash_offs: 0x00000000(0K)	data_size: 0x000025B4(10K)
sec bin 1 app.bin	:	flash_offs: 0x00004000(16K)	data_size: 0x00004C20(20K)
sec bin 2 app_xip.bin	:	flash_offs: 0x00029800(166K)	data_size: 0x00078020(481K)
sec bin 3 wlan_bl.bin	:	flash_offs: 0x000D8800(866K)	data_size: 0x00000950(3K)
sec bin 4 wlan_fw.bin	:	flash_offs: 0x000D9800(870K)	data_size: 0x00004FD4(20K)
sec bin 5 wlan_sdd_40M.bin	:	flash_offs: 0x000F9800(998K)	data_size: 0x00000358(1K)
OTA area	:	flash_offs: 0x00100000(1024K)	data_size: 0x00001000(4K)

generate image: xr_system.img

cp -t .././../out/ ../image/"xr806"/*.bin ../image/"xr806"/xr_system.img *.map

make[1]: Leaving directory `/home/appos/project/demo/hello_demo/gcc'

6 常见问题

6.1 Bin 文件超出 max_len 的解决

如果 bin 文件实际大小超出了 max_len 定义的大小，则软件打包时会报错，并提示出错的 bin 文件以及超出的大小。

```
cfg string:
/*
 *
 * Automatically generated file; DO NOT EDIT.
 * XR806 SDK Configuration
 */
{
    "magic" : "AWIH",
    "version" : "0.5",
    "OTA" : {"addr" : "1024K", "size" : "4K"},
    "image" : {"max_size" : "1020K"},
    "section" :
    [
        {"id" : "0xa5ff5a00", "bin" : "boot_40M.bin", "cert" : "null", "max_len" : "16K", "sram_offs" :
"0x00230000", "ep" : "0x00230101", "attr" : "0x01"},
        {"id" : "0xa5fe5a01", "bin" : "app.bin", "cert" : "null", "max_len" : "150K", "sram_offs" :
"0x00201000", "ep" : "0x00201101", "attr" : "0x1"},
        {"id" : "0xa5fd5a02", "bin" : "app_xip.bin", "cert" : "null", "max_len" : "500K", "sram_offs" :
"0xffffffff", "ep" : "0xffffffff", "attr" : "0x2"},
        {"id" : "0xa5fa5a05", "bin" : "wlan_bl.bin", "cert" : "null", "max_len" : "4K", "sram_offs" : "0xffffffff",
"ep" : "0xffffffff", "attr" : "0x1"},
        {"id" : "0xa5f95a06", "bin" : "wlan_fw.bin", "cert" : "null", "max_len" : "128K", "sram_offs" :
"0xffffffff", "ep" : "0xffffffff", "attr" : "0x1"},
        {"id" : "0xa5f85a07", "bin" : "wlan_sdd_40M.bin", "cert" : "null", "max_len" : "1K", "sram_offs" :
"0xffffffff", "ep" : "0xffffffff", "attr" : "0x1"},
        {}
    ]
}
```

err: bin 2 (app_xip.bin) size is bigger than max_len

bin real size: 563888

bin max_len:512000

Generate image file failed

make[1]: *** [image] Error 255

make[1]: Leaving directory `/appos/project/demo/wlan_demo/gcc'

make: *** [_build] Error 2

上述打印显示，app_xip.bin 实际大小为 550K 左右，超出了 max_len 定义的 500K，此时可以将 cfg 文件中 app_xip.bin 的 max_len 由原来的 500K 调整为 600K 即可，剩下的 50K 做为预留空间。

6.2 文件总长超过 max_size 的解决

如果全部的 bin 文件在经过自动计算后，总长度仍超 max_size 定义的长度，此时用户需要重新考虑 max_size 的大小以及 OTA 区域的位置，然后手动进行 cfg 文件的修改。

cfg string:

```
/*
 *
 * Automatically generated file; DO NOT EDIT.
 * XR806 SDK Configuration
 */
{
    "magic" : "AWIH",
    "version" : "0.5",
    "OTA" : {"addr": "1024K", "size": "4K"},
    "image" : {"max_size": "900K"},
    "section" :
    [
        {"id": "0xa5ff5a00", "bin": "boot_40M.bin", "cert": "null", "max_len": "16K", "sram_offs":
        "0x00230000", "ep": "0x00230101", "attr": "0x01"},
        {"id": "0xa5fe5a01", "bin": "app.bin", "cert": "null", "max_len": "150K", "sram_offs":
        "0x00201000", "ep": "0x00201101", "attr": "0x1"},
        {"id": "0xa5fd5a02", "bin": "app_xip.bin", "cert": "null", "max_len": "700K", "sram_offs":
        "0xffffffff", "ep": "0xffffffff", "attr": "0x2"},
        {"id": "0xa5fa5a05", "bin": "wlan_bl.bin", "cert": "null", "max_len": "4K", "sram_offs": "0xffffffff",
        "ep": "0xffffffff", "attr": "0x1"},
        {"id": "0xa5f95a06", "bin": "wlan_fw.bin", "cert": "null", "max_len": "128K", "sram_offs":
        "0xffffffff", "ep": "0xffffffff", "attr": "0x1"},
        {"id": "0xa5f85a07", "bin": "wlan_sdd_40M.bin", "cert": "null", "max_len": "1K", "sram_offs":
        "0xffffffff", "ep": "0xffffffff", "attr": "0x1"},
    ]
}
```

err: the real image size(1022808) is bigger than img_max_size(921600)!

Overlapped size: 101208 Byte(99kB)

Generate image file failed

make[1]: *** [image] Error 255

make[1]: Leaving directory `/appos/project/demo/wlan_demo/gcc'

make: *** [_build] Error 2

上述打印显示，整个 image 的实际大小为 999K 左右，超出了 max_len 定义的 900K，此时需要将 cfg 文件中 image 的 max_size 由原来的 900K 调整为 1020K 即可，剩下的 20K 做为预留空间。OTA 区域如果也发生重叠，需要一并进行调整。

著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。