

---

文档编号：FlashToolCLI 说明

---

**EigenCOMM Wireless Microcontroller**

产品综述

---

本文档描述移芯通信 FlashToolCLI 命令行接口使用说明。

功能概述

---

【文档描述产品的主要功能、主要特性】

- 【功能描述】
- 【功能描述】
- 【功能描述】
- 【功能描述】

产品图片

---

【添加产品照片，通常包括产品正反面两张，主要针对有关硬件的文档，软件文档可删减】

Figure 1. 产品名称



## 目录

1. 产品参数导入说明 .....	3
1.1 扩展打包机制的引入 .....	3
1.2 产品打包机制说明 .....	3
1.3 扩展打包机制的脚本实现 .....	3
1.3.1 扩展版本包打包的一般脚本说明 .....	3
1.3.2 工具内置产品基础参数打包脚本 .....	4
1.3.3 版本包内置产品基础参数打包脚本 .....	4
1.3.4 外部存储 FLASH 文件形式的打包 .....	5
1.3.5 可选镜像文件的打包脚本 .....	6
1.4 扩展版本包产品基础参数说明 .....	6
1.4.1 产品预置参数文件 .....	6
1.4.2 工具基础配置 baseini 文件 .....	10
1.4.3 存储区域划分的 format.json 文件 .....	13
1.4.4 工具通用产品配置 ini 文件 .....	14
1.4.5 导入工具内置产品参数的过程 .....	16
1.4.6 导入版本包内置产品参数的过程 .....	16
2. 附录 .....	17
2.1 参考文档 .....	17
2.2 术语和解释 .....	18
2.3 安全和警告 .....	19
3. 版本 .....	20
4. 关于我们 .....	21

## 1. 产品参数导入说明

### 1.1 扩展打包机制的引入

基于芯片型号的增加，及芯片封装模式的多样化，主要如 718s/718p/718h/618s 等，不同的芯片或者封装存在存储空间种类、大小差异，软件功能也由此产生了差异化，比如如镜像偏移地址，文件系统空间划分等，FlashTool 工具需要能够解析这些参数，并根据这些参数来决定下载的存储 Flash 设备对象，及擦除时识别对应的擦除区域。由于原版的版本包并不支持附带属性的功能，因此增加了一种新的版本打包机制，SDK 发布版本时，在版本包内增加产品名称，及可选的基础配置文件如 format.json，baseini 等。相对应的，FlashTool V4.1.1 之后，增加了对于新的打包机制发布的版本的支持，通过解析产品名称及基础配置文件，工具、软件镜像、版本包、芯片之间通过这些参数关联后实现联动，进而简化参数配置，提升烧录过程的自动化和统一化管理。

### 1.2 产品打包机制说明

在引入扩展打包机制之前的制作的 binpkg 版本包称为遗留版本包，如原有 618 SDK 发布的 binpkg 版本镜像，其镜像打包时脚本执行 fcelf 过程中不包含 pkgmode，banoldtool，productname 打包参数，定义为 legacy product binpkg。

在引入扩展打包机制之后的制作的 binpkg 版本包称为以产品命名的版本包，718 系列封装及 SDK 发布的 binpkg 版本镜像，其镜像打包时脚本执行 fcelf 过程中包含 pkgmode，banoldtool，productname 打包参数，定义为 named product binpkg。

### 1.3 扩展打包机制的脚本实现

扩展打包也即 named product binpkg 的打包，如前所述，扩展版本包内一般附加有所属产品名称，及可选的基础配置文件如 prmgrcfg.json，format.json，baseini 等，所以根据基础配置文件的选择，打包脚本可以分为工具内置参数的打包脚本和版本包内置参数的打包脚本。

#### 1.3.1 扩展版本包打包的一般脚本说明

扩展版本包打包的一般脚本，其脚本执行 fcelf 的命令行参数一般示例如下：

```
.\fcelf.exe -M ^  
-input .\pkgimg_ec718h\ap_bootloader.bin -addrname BL_IMG_MERGE_LNA -flashsize  
BOOTLOADER_FLASH_LOAD_SIZE ^  
-input .\pkgimg_ec718h\ap_at_command.bin -addrname AP_IMG_MERGE_LNA -flashsize AP_FLASH_LOAD_SIZE ^  
-input .\pkgimg_ec718h\cp-demo-flash.bin -addrname CP_IMG_MERGE_LNA -flashsize CP_FLASH_LOAD_SIZE ^  
-input .\pkgimg_ec718h\lfs-demo-flash.bin -addrname PKGFLXFS_IMG_MERGE_LNA -flashsize  
PKGFLXFS_FLASH_LOAD_SIZE ^  
-banoldtool 1 ^  
-pkgmode 1 ^
```

```
-productname EC718H_PRD ^
```

```
-def.\mem_map.h ^
```

```
-outfile.\merge_multi_test.binpkg
```

参数说明

-addrname NAME\_IMG\_MERGE\_LNA 参数，一般用于指定镜像烧录地址，开始部分的 NAME，同时也用来定义镜像的类型，如 NAME 为 BL 对应 bootloader 镜像，NAME 为 AP 对应 AP 镜像，NAME 为 CP 对应为 CP 镜像。特别的 NAME 为 PKGFLX 开始时，用来定义一些可选文件的烧录，如本例中 NAME 为 PKGFLXFS 定义为 LITTLE FS 文件系统镜像的烧录。

-Banoldtool: 一般用于扩展版本包防止用旧版本 FlashTool 进行下载，配置为 1；

-Pkgmode: 一般是指扩展打包方式，配置为 1

-Productname: 版本包对应的产品名称，用此名称可以索引基础配置文件中的产品配置数据，包括存储空间，format.json,baseini 等。

### 1.3.2 工具内置产品基础参数打包脚本

工具内置产品基础参数的扩展版本包内除烧录镜像外，一般不包含额外的基础配置文件，一般适用于测试的版本或者原厂的 SDK 软件版本，其参数不会因为版本迭代而发生变化，所以这些基础配置文件 prmgrcfg.json, format.json, baseini 在工具发布时已经静态配置并随工具一起发布。

工具内置产品基础参数打包脚本，其形式和前述“扩展版本包打包的一般脚本说明”类似，其脚本执行 fcelf 的命令行参数一般基本示例如下：

```
.\fcelf.exe -M ^
```

```
-input.\pkgimg_ec718s\ap_bootloader.bin -addrname BL_IMG_MERGE_LNA -flashsize  
BOOTLOADER_FLASH_LOAD_SIZE ^
```

```
-input.\pkgimg_ec718s\ap_at_command.bin -addrname AP_IMG_MERGE_LNA -flashsize AP_FLASH_LOAD_SIZE ^
```

```
-input.\pkgimg_ec718s\cp-demo-flash.bin -addrname CP_IMG_MERGE_LNA -flashsize CP_FLASH_LOAD_SIZE ^
```

```
-banoldtool 1 ^
```

```
-pkgmode 1 ^
```

```
-productname EC718S_PRD ^
```

```
-def.\mem_map_ec718s.h ^
```

```
-outfile.\merge_multi_test.binpkg
```

### 1.3.3 版本包内置产品基础参数打包脚本

扩展版本包内置产品基础参数的形式，主要体现在基础配置文件一般如 PrMgrCfg.json，Format.json, base\_uart\_config\_file、base\_usb\_config\_file，这些文件可选的打包到 binpkg 版本包中，其主要作用是，当烧录相关的产品配置参数发生变化、产品的镜像偏移地址变更或者产品的文件系统空间调整时，

FlashTool 工具发布时携带的产品数据已经偏离当前要烧录的版本包中携带的产品数据，此时通过版本包中打包的基础配置数据，可同步获取最新的产品数据集，保证产品数据和版本包的一体化，进而正确的实现烧录功能。

版本包内置产品基础参数打包脚本，其脚本执行 fcelf 的命令行参数一般基本示例如下：

```
.\fcelf.exe -M ^  
  
-input .\pkgimg_ec718s_ff\ap_bootloader.bin -addrname BL_IMG_MERGE_LNA -flashsize  
BOOTLOADER_FLASH_LOAD_SIZE ^  
  
-input .\pkgimg_ec718s_ff\ap_at_command.bin -addrname AP_IMG_MERGE_LNA -flashsize AP_FLASH_LOAD_SIZE ^  
  
-input .\pkgimg_ec718s_ff\cp-demo-flash.bin -addrname CP_IMG_MERGE_LNA -flashsize CP_FLASH_LOAD_SIZE ^  
  
-input .\pkgimg_ec718s_ff\XPK_FF_PrMgrCfg.json.bin -addrname XPKGDPDM_VIRTIMG_MERGE_LNA -flashsize  
XPKG_VIRTIMG_LOAD_SIZE ^  
  
-input .\pkgimg_ec718s_ff\xpk_ff_format_ec718s.json.bin -addrname XPKGDCMN_VIRTIMG_MERGE_LNA -flashsize  
XPKG_VIRTIMG_LOAD_SIZE ^  
  
-input .\pkgimg_ec718s_ff\config_xpk_ff_ec718s_prd_uart.baseini.bin -addrname XPKGDCMN_VIRTIMG_MERGE_LNA -  
flashsize XPKG_VIRTIMG_LOAD_SIZE ^  
  
-input .\pkgimg_ec718s_ff\config_xpk_ff_ec718s_prd_usb.baseini.bin -addrname XPKGDCMN_VIRTIMG_MERGE_LNA -  
flashsize XPKG_VIRTIMG_LOAD_SIZE ^  
  
-banoldtool 1 ^  
  
-pkgmode 1 ^  
  
-productname XPK_FF_EC718S_PRD ^  
  
-def .\mem_map_xpk_ec718s.h ^  
  
-outfile .\merge_multi_test_xpk_ff.binpkg
```

#### 参数说明

-addrname NAME\_VIRTIMG\_MERGE\_LNA, 当 NAME 为 XPKGDPDM 时，也即 XPKGDPDM\_XXX\_XXX\_LNA 时，定义为产品集预定义基础配置文件，如本例为 XPK\_FF\_PrMgrCfg.json.bin；XPKGDPDM\_XXX\_XXX\_LNA 一般在头文件 mem\_map\_XXX.h 中定义，一般定义为 0 即可，XPKG\_VIRTIMG\_LOAD\_SIZE 一般定义为 0x100000 即可。

-addrname NAME\_XXX\_XXX\_LNA, 当 NAME 为 XPKGDCMN 时，也即 XPKGDCMN\_XXX\_XXX\_LNA，定义通用基础配置文件，如本例为 xpk\_ff\_format\_ec718s.json.bin，config\_xpk\_ff\_ec718s\_prd\_uart.baseini.bin，config\_xpk\_ff\_ec718s\_prd\_usb.baseini.bin 等；XPKGDCMN\_XXX\_XXX\_LNA 一般在头文件 mem\_map\_XXX.h 中定义，一般定义为 0 即可，XPKG\_VIRTIMG\_LOAD\_SIZE 一般定义为 0x100000 即可。

### 1.3.4 外部存储 FLASH 文件形式的打包

扩展版本包还可以打包外部存储 FLASH 文件，以此支持工具可选的烧录外部 SPI FLASH 的预置烧录功能，其脚本执行 fcelf 的命令行参数一般示例如下：

```
.\fcelf.exe -M ^  
  
-input .\pkgimg_ec718h\ap_bootloader.bin -addrname BL_IMG_MERGE_LNA -flashsize
```

BOOTLOADER\_FLASH\_LOAD\_SIZE ^

-input .\pkgimg\_ec718h\ap\_at\_command.bin -addrname AP\_IMG\_MERGE\_LNA -flashsize AP\_FLASH\_LOAD\_SIZE ^

-input .\pkgimg\_ec718h\cp-demo-flash.bin -addrname CP\_IMG\_MERGE\_LNA -flashsize CP\_FLASH\_LOAD\_SIZE ^

-input .\pkgimg\_ec718h\lfs-demo-flash.bin -addrname PKGFLXFS\_IMG\_MERGE\_LNA -flashsize

PKGFLXFS\_FLASH\_LOAD\_SIZE ^

-input .\pkgimg\_ec718h\ext\_lfs.bin -addrname EF\_IMG\_MERGE\_LNA -flashsize EF\_FLASH\_LOAD\_SIZE ^

-banoldtool 1 ^

-pkgmode 1 ^

-productname EC718H\_PRD ^

-def .\mem\_map.h ^

-outfile .\merge\_multi\_test.binkg

参数说明

-addrname NAME\_IMG\_MERGE\_LNA 参数，本例中特别的 NAME 定义为 EF 时，用来定义外部 SPI FLASH 镜像的烧录，如本例中 NAME 为 EF 定义为外部 LITTLE FS 文件系统镜像的烧录。

### 1.3.5 可选镜像文件的打包脚本

固定的镜像部分如 bootloader 镜像/AP 镜像/CP 镜像等，其对应的打包脚本参数-addrname NAME\_IMG\_MERGE\_LNA 参数，开始部分的 NAME 形式如 BL/AP/CP。

当 NAME 定义为 PKGFLX 开始时，用来定义一些可选文件的烧录，可以根据需求将多个可选文件打包的扩展版本包中，其打包脚本形式可参考“扩展版本包打包的一般脚本说明”。

## 1.4 扩展版本包产品基础参数说明

产品基础参数一般包含一下几个部分，产品预置参数文件 prmgrcfg.json，产品烧录基础配置 baseini 文件，包含存储区域划分的 format.json 文件。

### 1.4.1 产品预置参数文件

产品预置参数文件定义了所支持的产品列表，指定对应产品的产品名称、产品用于工具加载的 uart/usb 基础配置文件、存储区域划分的 format.json 文件，所支持的存储设备定义。

#### 1.4.1.1 工具内置产品预置参数文件

工具内置产品预置参数文件其形式一般如下：

```
{
    "product_sets"://产品集
    [
```

```
{  
  
    "prod_name": "EC618_OLDPKG_DEFAULT", //618 产品名称  
  
    "base_uart_config_file": "./product_sets/ec618_products/618_legacy/base_config_files/config_ec618_legacy_pkg_uart.baseini",  
    //618 使用 uart 端口下载时的工具基础配置文件  
  
    "base_usb_config_file": "./product_sets/ec618_products/618_legacy/base_config_files/config_ec618_legacy_pkg_usb.baseini",  
    //618 使用 usb 端口下载时的工具基础配置文件  
  
    "specific_info": {  
        "format_path": "./product_sets/ec618_products/618_legacy/format/format_ec618_legacy.json"  
        //618 存储区域划分的 format.json 文件  
  
    },  
    "storage_info": [  
        {  
            "NAME": "AP", //618 支持 AP FLASH 设备  
            "BASE_LNA": "0x800000", //618 AP FLASH 设备的基地址  
            "BASE_SIZE": "0x400000" //618 AP FLASH 设备的大小  
        },  
        {  
            "NAME": "CP", //618 支持 CP FLASH 设备  
            "BASE_LNA": "0x8800000", //618 CP FLASH 设备的基地址  
            "BASE_SIZE": "0x100000" //618 CP FLASH 设备的大小  
        }  
    ]  
},  
  
{  
  
    "prod_name": "EC718H_PRD", //718h 产品名称  
  
    "base_uart_config_file": "./product_sets/ec718_products/ec718h/base_config_files/config_ec718h_prd_uart.baseini",  
    //718h 使用 uart 端口下载时的工具基础配置文件  
  
    "base_usb_config_file": "./product_sets/ec718_products/ec718h/base_config_files/config_ec718h_prd_usb.baseini",  
    //718h 使用 usb 端口下载时的工具基础配置文件
```

```

"specific_info":{
  "format_path": "./product_sets/ec718_products/ec718h/format/format_ec718h.json",
  //718h 存储区域划分的 format.json 文件

  "cp_flash_spec_images": "cp_system"
  //718h 指定 cp_system 镜像存放于 cp flash

},
"storage_info": [
{
  "NAME": "AP", //718h 支持 AP FLASH 设备
  "BASE_LNA": "0x800000", //718h AP FLASH 设备的基地址
  "BASE_SIZE": "0x400000" //718h AP FLASH 设备的大小
},
{
  "NAME": "CP", //718h 支持 CP FLASH 设备
  "BASE_LNA": "0x800000", //718h CP FLASH 设备的基地址
  "BASE_SIZE": "0x100000" //718h CP FLASH 设备的大小
}

]
},
{
  "prod_name": "EC718P_PRD",

  "base_uart_config_file": "./product_sets/ec718_products/ec718p/base_config_files/config_ec718p_prd_uart.baseini",

  "base_usb_config_file": "./product_sets/ec718_products/ec718p/base_config_files/config_ec718p_prd_usb.baseini",
  "specific_info": {
    "format_path": "./product_sets/ec718_products/ec718p/format/format_ec718p.json",
    "ap_flash_spec_images": "cp_system"
  },
  "storage_info": [
    {
      "NAME": "AP",
      "BASE_LNA": "0x800000",

```



```

        "BASE_SIZE":"0x400000"
    }
]
},
{
    "prod_name":"EC718S_PRD",
    "file_basedir_keys_back":"base_uart_config_file;base_usb_config_file;format_path",

    "base_uart_config_file":"./product_sets/ec718_products/ec718s/base_config_files/config_ec718s_prd_uart.baseini",

    "base_usb_config_file":"./product_sets/ec718_products/ec718s/base_config_files/config_ec718s_prd_usb.baseini",
    "specific_info":{
        "format_path":"./product_sets/ec718_products/ec718s/format/format_ec718s.json",
        "ap_flash_spec_images":"cp_system"
    },
    "storage_info":[
        {
            "NAME":"AP",
            "BASE_LNA":"0x800000",
            "BASE_SIZE":"0x200000"
        }
    ]
}
]
}

```

#### 1.4.1.2 版本包内置产品预置参数文件

以“版本包内置产品基础参数打包脚本”中使用的参数为例，版本包内置产品预置参数文件其形式一般如下：

```

{
    "product_sets":
    [
        {
            "prod_name":"XPK_FF_EC718S_PRD", //产品名称，为了区分其他非版本包内置产品参数的产品

```

及版本包，一般以 XPK 定义开始命名产品名称。

```
"file_basedir_keys":"base_uart_config_file;base_usb_config_file;format_path",
//所列关键字部分，其配置的相对路径是以为当前文件所在的目录为基础路径的相对路径，在导入这些文件时，会按照基础路径转换形成一个绝对路径。

"base_uart_config_file":"./config_xpk_ff_ec718s_prd_uart.baseini.bin",
//使用 uart 端口下载使用的 baseini 文件

"base_usb_config_file":"./config_xpk_ff_ec718s_prd_usb.baseini.bin",
//使用 usb 端口下载使用的 baseini 文件

"specific_info":{
"format_path":"./xpk_ff_format_ec718s.json.bin",
//包含存储区域划分的 format 文件

"ap_flash_spec_images":"cp_system"
//718s 不包含 cp flash 设备，所以 cp_system 镜像指定存放在 ap flash 设备
},
"storage_info":[
{
"NAME":"AP", //718s 支持 AP FLASH 设备
"BASE_LNA":"0x800000", //718s AP FLASH 设备的基地址
"BASE_SIZE":"0x200000"//718s AP FLASH 设备的大小
}
]
}
]
```

#### 1.4.2 工具基础配置 baseini 文件

工具基础配置文件一般以.baseini 为后缀，其中内置工具基础配置文件 618 或 718 一般在如  
product\_sets\ec618\_products\618\_legacy\base\_config\_files  
product\_sets\ec718\_products\ec718h\base\_config\_files  
product\_sets\ec718\_products\ec718s\base\_config\_files  
product\_sets\ec718\_products\ec718p\base\_config\_files 等目录下。

比如 config\_xpk\_ff\_ec718s\_prd\_uart.baseini 文件内容配置如下：

[config] //配置

line\_0\_com = COM9 //下载默认使用的端口号

agbaud = 921600 //下载使用的波特率

filter\_embedusb = 1 //ui 工具使用，界面端口选择列表是否过滤内置 boot usb 端口

filter\_externcom = 0 //ui 工具使用，界面端口选择列表是否过滤除 boot usb 端口之外的端口

filter\_explicitcom0 = 19D1 //显示配置端口选择列表需要过滤 usb vid 对应的设备 0

filter\_explicitcom1 = 1366 //显示配置端口选择列表需要过滤 usb vid 对应的设备 1

path\_convert\_cfg\_avail = 1

//baseini 文件中的路径是否支持配置相对路径的基础路径，该参数一般当工具使用的通用产品配置文件

config\_pkg\_product\_uart.ini/ config\_pkg\_product\_usb.ini 文件不在工具根目录下时，由于以 baseini 中配置的以 baseini 文件为基础路径的相对路径和以工具根目录为基础路径的相对路径不一致，存在歧义，所以此时就需要使能该参数项，确认其相对路径的基础路径。

默认配置 0 时，在导入参数时不进行任何修改，即如果配置的为相对路径，最终导入到最终工具配置 ini 文件后，仍旧为相对路径；

当配置为 1 时，可以支持配置相对路径的基础路径为以当前 baseini 文件所在的路径为基础路径或者以工具所在的路径为基础路径。baseini 文件所在的路径为基础路径的参数项在 pathconv\_to\_file\_basedir\_pairs 配置，以工具所在的路径为基础路径参数项在 pathconv\_to\_tool\_basedir\_pairs 配置。该配置项参数

pathconv\_to\_file\_basedir\_pairs=flexfile2:filepath;flexfile3:filepath

//以 seciton:key 按 pairs 进行配置，pairs 于 pairs 之间以“;”为分割符

pathconv\_to\_tool\_basedir\_pairs=flexfile4:filepath; flexfile5:filepath

//以 seciton:key 按 pairs 进行配置，pairs 于 pairs 之间以“;”为分割符

[agentboot] //下载中间件相关的配置

tool\_basedir = 1 //该参数指明当 agpath 配置为相对路径时，其相对路径所对应的基础路径为工具所在的路径。

agpath = .\product\_sets\ec718\_products\common\_data\agentboot\_uart\agentboot.bin

[storage\_cfg] //存储器相关的配置

#will be update when load

format\_path = format\_ec718\_none.json //配置一个无效值即可，导入时，会从 prmgrcfg.json 中导入到通用产品配置文件

config\_pkg\_product\_uart.ini/ config\_pkg\_product\_usb.ini 文件

[bootloader] //bootloader 镜像的相关配置

blpath = .\image\_ec619\ap\_bootloader.bin //bootloader 镜像的路径，默认即可，导入时会从 binpkg 更新

blloadskip = 0 //默认是否取消 bootloader 镜像下载，一般为 0 即可

burnaddr=0xf //bootloader 镜像的烧录地址，默认即可，导入时会从 binpkg 更新

;system.bin file information

[system] //system 镜像的相关配置

syspath = .\image\_ec619\ap\_demo-flash.bin //system 镜像的路径，默认即可，导入时会从 binpkg 更新

sysloadskip = 0 //默认是否取消 system 镜像下载，一般为 0 即可

burnaddr = 0xf //system 镜像的烧录地址，默认即可，导入时会从 binpkg 更新

[cp\_system] //cp\_system 镜像的相关配置

cp\_syspath = .\image\_ec619\cp-demo-flash.bin //cp\_system 镜像的路径，默认即可，导入时会从 binpkg 更新

cp\_sysloadskip = 0 //默认是否取消 cp\_system 镜像下载，一般为 0 即可

burnaddr = 0xf //cp\_system 镜像的烧录地址，默认即可，导入时会从 binpkg 更新

[control] //下载控制参数

detect = 1

//detect 1 使用探测下载，一般使用 uart 端口下载时使用,并且配置 atecrst 复位配置项进行联动控制，多次周期等待进行探测连接，

detect 0 一般使用 usb 下载时使用

detect 2 一般使用 uart 端口下载时使用，且没有用 atecrst 进行联动控制，通过工具持续发送握手信号，进行探测链接。

reset = 2

//reset0 根据 skipconnect 参数进行复位逻辑控制，包括 dtr,rts 信号控制，通过端口打开、然后关闭使 dtr rts 信号控制生效等

Reset1 忽略 skipconnect 参数，在连接之前时钟进行复位逻辑控制，包括 dtr,rts 信号控制，通过端口打开、然后关闭使 dtr rts 信号控制生效等

Reset2 不进行复位逻辑控制

atreset = at+ecrst=delay,600 //在下载时通过 at 复位命令发送给 UE，进行复位

atbaud = 115200 //在下载时通过 at 复位命令发送给 UE 所使用的波特率

preempt\_detect\_time = 6 //在 detect 为 2 时，持续探测的时间长读，单位为 s

msg\_waittime = 2 //下载交互过程中，消息的等待超时时间，单位为 s

max\_preamble\_cnt = 8 //下载交互过程中，preamble 握手的次数。

lpc\_recover\_en = 0 //下载交互过程中，连接失败时，是否允许尝试恢复连接

pullup\_qspi = 1 //下载过程中，是否上拉 spi flash 的 pad，进行初始化控制，一般适用于 616s 芯片 xmc flash 兼容。

trc\_ser = 1 //下载过程，打印串口打开关闭的 log

rom\_version= 0000000203000000 //下载开始前的芯片 version 检查，当不匹配时，停止下载过程

dribble\_dld\_en = 1 //下载过程中 flash 对齐下载优化，一般在 718 芯片的 uart 下载端口时使用。

[flexfile0] //可选文件的下载配置，ui 工具中一般用于配置校准文件镜像的烧录路径，烧录地址和烧录 flash 设备

filepath = .\product\_sets\ec718\_products\ec718h\rfCaliTb\MergeRfTable.bin //配置文件镜像的烧录路径

burnaddr = 0xe7000 //配置文件镜像的烧录地址偏移量

storage\_type=cp\_flash//配置文件镜像的烧录 flash 设备

### 1.4.3 存储区域划分的 format.json 文件

存储区域划分文件一般命名为 format.json，用于选择擦除 NVM 区域，或者开发者模式除校准区域外的数据擦除空间索引。

```
{
  "fs": {
    "begin": "0x9e2000", //NVM little fs 区域起始地址
    "length": "65536" //长度
  },
  "fraw": {
    "begin": "0x9ff000", //NVM 平台裸数据配置区域的起始地址
    "length": "4096" //长度
  },
  "rr": {
    "begin": "0x9f3000", //校准区域的起始地址
    "length": "49152" //长度
  },
  "drr": {
    "begin": "0x813000", //备份校准区域的起始地址
    "length": "20480" //长度
  },
  "hfd": {
    "begin": "0", //异常 dump 时的数据起始地址
    "length": "0" //长度
  },
  "opt": {
    "begin": "0x0800000", //ui 界面上，可选区域回读数据的起始地址
    "length": "2097152" //长度
  }
}
```

```

    },
    "cal":[ //ui 工具，下载校准文件时，会擦除的区域配置
        {
            "stor_type": "ap_flash", //flash 设备类型
            "offst_flag": "1",      //是否为偏移地址
            "begin": "0x1f3000", //起始地址
            "length": "0xc000"     //长度
        }
    ],
    "erallum":[ //ui 工具 eraseall 擦除时，选择的擦除区域，一般为除校准区域外的所有其他区域，分为几块区域。
        {
            "stor_type": "ap_flash",
            "offst_flag": "1",
            "begin": "0x0",
            "length": "0x13000"
        },
        {
            "stor_type": "ap_flash",
            "offst_flag": "1",
            "begin": "0x18000",
            "length": "0x1da000"
        },
        {
            "stor_type": "ap_flash",
            "offst_flag": "1",
            "begin": "0x1ff000",
            "length": "0x1000"
        }
    ]
}

```

#### 1.4.4 工具通用产品配置 ini 文件

工具通用产品配置 ini 文件一般命名为 config\_pkg\_common\_uart.ini, config\_pkg\_common\_usb.ini, 分别对应于 usb 下载端口和 usb 下载端口。其通用的含义主要是指工具以该文件为基础，只需要用 arg\_pkg\_path\_val 指定版本包的路径，则工具通过对外无差异的接口命令，实现通过解压，识别产品名称，导入参数，同步文件等一系列过程，最终完

成产品包的识别和下载。

工具通用产品配置 ini 文件配置项示例关键字段如下：

```
[package_info]
pkgflag = 1 //适用于 binpkg 版本包
pkg_extract_exe = .\fcelf.exe //解压工具所在路径
arg_pkg_path_val = .\image_ec718\named_product\ec718h\pkgdir\at_command.binpkg //版本包所在路径
comment_inicfg_regened_state = "initial_no_change,backward_restored,forward_regened"
//注释的 ini 文件导入后的 3 种状态
select_product_support = 1 //是否支持产品名称识别、产品参数选择和导入
pkg_bins_regen_targetdir = .\pkgimg_gen //对于扩展包解压后同步的镜像路径
pkg_inicfg_regened_state = forward_regened // ini 文件导入后的当前状态
backward_cfg_restore_en = 1 //允许加载扩展包之后，再回退到非扩展包功能
old_package_defined_product = EC618_OLDPKG_DEFAULT //非扩展包的产品默认名字，一般为 618 发布的非扩展模式的版本包
selected_product = EC718H_PRD //当前选择的产品名字
selected_base_inicfg_type = uart //适用 uart 端口或者 usb 端口进行下载
selected_base_inicfg_rec
= ./product_sets/ec718_products/ec718h/base_config_files/config_ec718h_prd_uart.baseini@hash:107ca73751f69f408bad39440d1b2d2b544d54fd567b97464e2be80122729ec2 //导入的基础 baseini 文件所在的路径和校验值。
```

其中蓝色部分配置内容为通用 product 配置文件中新增的参数。

通用配置项中一般会用于最终下载，但是在下载前，需要对即将要下载的 binpkg 进行对应的参数更新，以匹配对应的 Product 的下载参数，工具会先解析配置文件中的 arg\_pkg\_path\_val 值，如

. \image\_ecxxxx\named\_product\xxx\pkgdir\at\_command.binpkg，解析产品名称并获取到预定义信息后，从对应的 base\_uart\_config\_file，base\_usb\_config\_file 更新配置参数，如[config][agentboot][bootloader][system][cp\_system][control][flexfile0][flexfilex]配置段等到当前配置文件，从 prmgrcfg.json 文件种索引 format\_path 并预定义信息 format\_path 到当前配置文件，从 prmgrcfg.json 获取，Flash 地址、大小更新配置内容，以及 cp\_system 的存放于 AP/CP Flash 的信息。

对于 named product binpkg 类型，在此次产品预定义更新之后，再从 binpkg 中解压镜像文件，并同步更新对应的 section[key]，如 [bootloader][blpath]，[bootloader][burnaddr]，[system][syspath]，[system][burnaddr]，[cp\_system][cp\_syspath]，[cp\_system][burnaddr]，[pkgflx0-n][filepath]，[pkgflx0-n][burnaddr]，[pkgflx0-n][storage\_type]等。并同步解压的文件到这些路径。

对于 legacy product binpkg 类型，在此次产品预定义更新之后，保留原有的参数配置习惯，一般不需要额外更新镜像相关如[bootloader][system][cp\_system]的配置参数。

#### 1.4.5 导入工具内置产品参数的过程

当版本包内 **不包含** - addname XPKGDPRM\_XX\_XX\_XX 的定义时，此时版本包内不包含内置产品数据，解析版本包时就依赖工具内置产品参数。工具内置产品参数的文件一般入口为产品集预定义基础配置文件，如 PrMgrCfg.json，Format.json，base\_uart\_config\_file、base\_usb\_config\_file 等。其参数导入过程简要说明参考“工具通用产品配置 ini 文件”。

#### 1.4.6 导入版本包内置产品参数的过程

当版本包内- addname **XPKGDPRM**\_XX\_XX\_XX 的定义时，此时版本包内包含内置产品数据，解析版本包时就依赖工具内置产品参数。产品包内置产品参数的文件以 718s 为例如 xpk\_ec718s\_PrMgrCfg.json，xpk\_ec718s\_Format.json，xpk\_config\_ec718s\_prd\_uart.baseini、xpk\_config\_ec718s\_prd\_usb.baseini、等，附带 xpk 和产品名称 ec718s 前缀，主要用于区分产品工具内置产品参数文件。

xpk\_ec718s\_PrMgrCfg.json 文件说明。

```
{
  "product_sets":
  [
    {
      "prod_name": "XPK_EC718S_PRD", //产品名称
      "file_basedir_keys_comment": "base_uart_config_file;base_usb_config_file;format_path",
      "file_basedir_keys": "base_uart_config_file;base_usb_config_file;format_path",
      //1.此处指定 format_path xpk_format_ec718s.json 文件相对路径的基础路径为
      当前 xpk_ec718s_format.json 文件所在的路径，即解压并同步后
      xpk_ec718s_format.json 文件所在的路径;
      //2 如 file_basedir_keys_comment 注释所示，base_uart_config_file，base_usb_config_file 两个关键字也可以选择不添加到此，那么就可以直接适用工具内置的 config_ec718s_prd_uart.baseini 和 config_ec718s_prd_usb.baseini 文件，并且修改成工具内对应文件的相对路径，一次可以减少打包的文件数量。

      "base_uart_config_file": "./xpk_config_ec718s_prd_uart.baseini",
      "base_usb_config_file": "./xpk_config_ec718s_prd_usb.baseini",
      "specific_info": {
        "format_path": "./xpk_format_ec718s.json",
      }
    }
  ]
}
```



```
        "ap_flash_spec_images":"cp_system"
    },
    "storage_info":[
        {
            "NAME":"AP",
            "BASE_LNA":"0x800000",
            "BASE_SIZE":"0x200000"
        }
    ]
}
```

版本包内置产品参数其参数导入基本原理和版本包无内置产品参数导入工具内置产品参数一致，只不过导入的配置文件是从版本包里面解压后得到的。

2. 附录

2.1 参考文档

Table 2-1. 参考文档

序号	文档名称	注释
【1】		
【2】		

2.2 术语和解释

Table 2-2. 术语和解释

术语	解释

---

## 2.3 安全和警告

3. 版本

版本	日期	备注
Draft	2019-04-08	

## 4. 关于我们

上海移芯通信科技有限公司（[www.eigencomm.com](http://www.eigencomm.com)）坐落于中国·上海张江硅谷，公司于 2017 年 2 月成立，致力于蜂窝物联网芯片的研发和销售，公司创始人及开发团队大部分来自于知名手机芯片厂商 Marvell，团队完整，阵容强大。其中，20%为知名高校博士，80%为知名高校硕士，平均工作年限 10 年以上。团队所开发的手机芯片已累计出货数亿片。开发团队在蜂窝终端芯片上积累了丰富的实战经验，从算法，协议栈，射频到基带 SOC 以及系统软硬件和方案，从低功耗设计经验到射频模拟开发能力具有完整而强大的研发能力。

### 上海移芯通信科技有限公司

地址：中国上海市浦东新区纳贤路 800 号 1 幢 A 座 6 楼 603

邮编：201210

电话：

电邮：

网址：<http://www.eigencomm.com>

### 技术支持窗口

电邮：[support@eigencomm.com](mailto:support@eigencomm.com)