

# USB 升级版本操作说明

**2017-5-31**

Version 0.2

## 目录

一、	初始版本烧写 .....	3
二、	USB 升级版本 .....	3
2.1	dbus 接口说明 .....	3
2.2	其它说明 .....	5
三、	场景示例 .....	5
3.1	升级 bootloader .....	5
3.2	升级版本 .....	6

## 一、 初始版本烧写

使用 mfgtools 工具对单板进行初始版本的烧写，该工具的使用方法详见《Manufacturing Tool Quick Start Guide.docx》。该工具会根据配置文件对 mmc 进行分区格式化，并将初始版本烧写到指定位置，单板上电后会由 uboot 自动加载运行版本。具体操作步骤如下：

1. 将需要烧写的初始版本文件放到“mfgtools\Profiles\Linux\OS Firmware\version\saic”目录下面，包含以下文件。

u-boot.imx	bootloader 镜像文件
uImage	由 zeos_sdk 打包生成的镜像文件(含 ZEOS、XITOS 内核, ZEOS 应用)
rootfs.ext4	虚拟机根文件系统镜像文件
2. 打开 mfgtools 目录下的“MfgTool.exe”程序开始烧写，整个过程大约几分钟。完成后重启单板，会自动加载运行烧写的版本。

## 二、 USB 升级版本

使用 mfgtools 工具烧写版本是比较麻烦的，需要一台专门的 Windows 电脑和 OTG 烧写线。为了更灵活的进行版本升级，提供了 USB 的实现方式，只需要通过一个 U 盘就可以完成版本升级工作。当单板完成初始版本烧写后，后续就可以使用 USB 的方式来升级版本了。

### 版本文件约定：

用于升级的 U 盘需要格式化为 FAT32 文件系统，并将待升级的版本文件存放在其根目录下。通过版本文件的前缀名用于区分其类型，具体如下所示：

bootloader_v1.0	表示 bootloader 镜像文件，如第一节的 u-boot.imx 大小不超过 512K
kernel_v1.0	表示 zeos_sdk 打包生成的镜像文件，如第一节的 uImage 大小不超过 32M
system_v1.0	表示虚拟机根文件系统镜像文件，如第一节的 rootfs.ext4 大小不超过 1G

### 2.1 dbus 接口说明

版本启动时会自动运行“版本升级服务进程”，该进程以 DBUS 接口方式对外提供服务。服务端接收并处理 method call 类型的消息，并返回应答消息。用户可通过 dbus-send 命令或者编写应用程序来与服务进程进行通信。

以如下的 dbus-send 命令为例进行说明。这里 bus name 定义为“saic.ver-up.server”，

interface 定义为“ver.up”。以不同的 method 名称来定义不同的服务类型，比如这里的 method 为“listfile”是表示列出所有可升级文件。

```
dbus-send --session --type=method_call --print-reply --dest=saic.ver-up.server / ver.up.listfile
```

命令的执行结果：

```
int32 2
string "bootloader_v1.0,kernel_v1.0"
```

服务端对各项服务请求都会返回应答消息，用于表示请求是否成功。该消息的内容包含两项（这里简称为 ret 和 info），其类型及意义如下。

ret	类型 DBUS_TYPE_INT32	当 ret 为-1 时表示失败，当 ret>=0 时表示成功
info	类型 DBUS_TYPE_STRING	info 表明失败原因或其它信息

### 1. 查询待升级版本

查询升级 U 盘根目录下的所有文件，并将可用的版本文件列表返回给客户端。

命令格式：

```
dbus-send --session --type=method_call --print-reply --dest=saic.ver-up.server / ver.up.listfile
```

应答消息：

-1	表示查询失败，info 表明失败原因
>=0	表示可用版本文件的个数，info 包含了文件列表（各文件用“,” 隔开）

### 2. 升级指定版本

根据用户指定的版本文件进行升级，若升级中出现错误会立即返回。命令格式如下所示，在参数字符串中指定要升级的版本文件名，多个文件时用“,” 隔开。也可以进行单独升级，即指定 bootloader 或 kernel 或 system 的文件名。

命令格式：

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000
--dest=saic.ver-up.server / ver.up.update string:'kernel_v1.0,system_v1.0'
```

应答消息：

-1	表示升级失败，info 表明失败原因
0	表示升级成功

### 3. 运行新版本

当上述“升级指定版本”成功时，可以由用户指定运行新版本，以便下次重启后 uboot 选择运行新版本。升级后的版本若运行失败，会走到回滚流程；若版本运行正常，则会需要步骤 4。

命令格式：

```
dbus-send --session --type=method_call --print-reply --dest=saic.ver-up.server /
ver.up.runnew
```

应答消息:

- 1        表示操作失败，info 表明失败原因
- 0        表示操作成功

#### 4. 标记当前版本有效

当版本运行正常的情况下，需要标记当前版本有效。因为后续升级和回滚都是以当前可用版本（即最近的一个可用版本）为基础进行操作的。

命令格式:

```
dbus-send --session --type=method_call --print-reply --dest=saic.ver-up.server /  
ver.up.setvalid
```

应答消息:

- 1        表示操作失败，info 表明失败原因
- 0        表示操作成功

#### 5. 查询升级进度

上面步骤 2 升级指定版本耗时比较长，这取决于升级文件的大小。提供查询升级进度的功能，该接口会立即返回。在 info 中包含进度信息，由“已升级字节数”和“总字节数”两部分构成，中间用“:” 隔开。

命令格式:

```
dbus-send --session --type=method_call --print-reply --dest=saic.ver-up.server / ver.up.query
```

应答消息:

- 1        表示操作失败，info 表明失败原因
- 0        表示操作成功

## 2.2 其它说明

bootloader 升级后如果出现不可用的情况，是不能通过版本回滚方式进行回退版本的。只能通过第一节 mfgtools 工具进行重新烧写。

## 三、 场景示例

### 3.1 升级 bootloader

升级成功:

- 1) 插入含有版本文件的 U 盘

- 2) 执行如下命令进行升级  

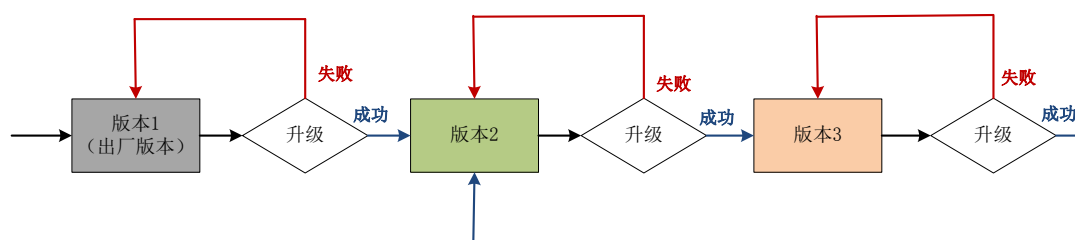
```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000  
--dest=saic.ver-up.server / ver.up.update string:'bootloader_v1.0'
```
- 3) 拔掉 U 盘
- 4) 重启单板，新的 bootloader 运行正常

#### 升级失败:

- 1) 插入含有版本文件的 U 盘
- 2) 执行如下命令进行升级  

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000  
--dest=saic.ver-up.server / ver.up.update string:'bootloader_v1.0'
```
- 3) 拔掉 U 盘
- 4) 重启单板，新的 bootloader 运行不正常
- 5) 使用第一节的 mfgtool 工具重新烧写

## 3.2 升级版本



#### 整体升级成功

- 1) 插入含有版本文件的 U 盘
- 2) 执行如下命令进行升级  

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000  
--dest=saic.ver-up.server / ver.up.update string:'kernel_v1.0,system_v1.0'
```
- 3) 执行如下命令，以便重启后运行新版本  

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000  
--dest=saic.ver-up.server / ver.up.runnew
```
- 4) 拔掉 U 盘
- 5) 重启单板
- 6) 新版本运行正常，执行如下命令设置新版本有效  

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000  
--dest=saic.ver-up.server / ver.up.setvalid
```

#### 部分升级成功

- 1) 插入含有版本文件的 U 盘
- 2) 执行如下命令进行升级  

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000
```

```
--dest=saic.ver-up.server / ver.up.update string:'kernel_v1.0'
```

- 3) 执行如下命令，以便重启后运行新版本

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000
```

```
--dest=saic.ver-up.server / ver.up.runnew
```

- 4) 拔掉 U 盘

- 5) 重启单板

- 6) 新版本运行正常，执行如下命令设置新版本有效

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000
```

```
--dest=saic.ver-up.server / ver.up.setvalid
```

### 升级失败回滚

- 1) 插入含有版本文件的 U 盘

- 2) 执行如下命令进行升级

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000
```

```
--dest=saic.ver-up.server / ver.up.update string:'kernel_v1.0,system_v1.0'
```

- 3) 执行如下命令，以便重启后运行新版本

```
dbus-send --session --type=method_call --print-reply --reply-timeout=180000
```

```
--dest=saic.ver-up.server / ver.up.runnew
```

- 4) 拔掉 U 盘

- 5) 重启单板

- 6) 新版本运行不正常，自动重启

- 7) 回退到升级之前的版本