

**FIBOCOM\_MINIDOWNLOADTOOL**

**下载协议说明**

文档版本：V1.0

更新日期：2017.09.12

适用型号

|  |  |  |
| --- | --- | --- |
| 序号 | 产品型号 | 说明 |
| 1 | NA | NA |

版权声明

版权所有©2017 深圳市广和通无线股份有限公司。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

版本记录

|  |  |  |
| --- | --- | --- |
| 文档版本 | 更新日期 | 说明 |
| V1.0 | 2017-09-12 | 初始版本 |

**目录**

[1 前言 5](#_Toc25058)

[2 升级流程概述 5](#_Toc19479)

[2.1 查找VIDPID流程概述 6](#_Toc19832)

[2.2 获取LIBUSB句柄流程概述 7](#_Toc5061)

[2.3 升级文件解析流程概述 8](#_Toc573)

[2.4 同步流程概述 12](#_Toc20649)

[2.5 BOOT下载流程概述 13](#_Toc15898)

[2.6 DOWNLOAD下载流程概述 14](#_Toc14618)

[3 错误码 15](#_Toc21970)

[3.1 LIBUSB错误码 15](#_Toc3952)

[3.2 m\_error\_txt 15](#_Toc18555)

# 前言

MiniDownloadTool为Intel平台升级工具，此工具基本C代码、仅依赖libusb库文件、使用FLS3格式进行升级。此工具仅满足升级功能，占用空间小，基本C语言实现，易于移植。

# 升级流程概述

开始

查找VIDPID

获取LIBUSB库句柄

解析升级文件

下发同步指令(ATAT)

Boot PSI EBL文件

Boot 结束

Download PSI EBL CODE 等文件

下载结束，重启模块

升级成功

升级失败

程序结束

如上图所述，升级流程，简述如下： 查找升级VIDPID，打开升级设备，下发同步指令，同步完成后，工具会首先进行BOOT文件下载之后再进行DOWNLOAD文件下载，之后升级完成，下发重启模块指令，重启模块。本章节主要介绍，同步流程及BOOT、DOWNLOAD下载的数据格式。

## **同步流程概述**

在升级前需要先进行同步，升级工具下发同步数据给模块，收到模块回复后同步成功，进行升级。

同步数据格式(8087/07F5): 下发2字节长度的 ATAT 数据

同步数据格式(非8087/07F5): 下发2次1字节长度的AT数据

模块收到同步数据后会先回复2字节长度的数据，再回复N字节数据，其中

同步返回数据格式1：

|  |  |
| --- | --- |
| 1字节 | 1字节 |
| LER\_VERSION | 剩余数据长度 |

同步返回数据格式2：

|  |
| --- |
| N字节 |
| 剩余数据 |

当模块正常将数据返回后，同步完成。

## **BOOT下载概述**

### BOOT PSI下载

BOOT PSI下载数据格式：

|  |  |  |  |
| --- | --- | --- | --- |
| 1字节 | 3节字 | N字节 | 1字节 |
| 包头(head) | 包长(length) | 数据(data) | 校验和 |

包头：0x30(固定值)

包长：BOOT PSI文件长度

数据：BOOT PSI文件数据

校验和：BOOT PSI文件数据逐字节异或相加

BOOT PSI下载数据下发顺序：

先下发包头(1字节)+包长(3字节)，再下发N字节数据，最后下发1字节校验和。

BOOT PSI应答数据格式：

|  |  |  |
| --- | --- | --- |
| 1字节 | 1节字 | 2字节 |
| ack | stat | alive\_rsp |

ack：0x01

stat：0x01

alive\_rsp：0xDD01或者0x0001

BOOT PSI应答顺序：

先读1字节ack,再读1字节stat,再读2字节alive\_rsp

### BOOT SLB下载

在BOOT SLB下载数据前需要先下发启动SLB下载的指令，

SLB下载启动指令格式：

|  |  |  |
| --- | --- | --- |
| 4字节 | 2字节 | 2字节 |
| Parameter参数 | CmdValue(值) | 校验和 |

Parameter: 0x00000000(固定值)

CmdValue: 0x0002(固定值)

Xor : Parameter ^ CmdValue

SLB下载启动指令应答格式：

|  |
| --- |
| 2字节 |
| Ack |

Ack:0xAA00 或0xCCCC

SLB下载启动指令结束后，就进行SLB数据下载,下载时序：

先下发SLB数据长度：

|  |
| --- |
| 1字节 |
| 包长度 |

SLB数据长度应答格式：

|  |
| --- |
| 1字节 |
| ACK |

现在阶段ACK值为：0xCCCC

再下发SLB数据及校验：

|  |  |
| --- | --- |
| N字节 | 1字节 |
| 数据 | 校验和 |

数据：对应数据长度的SLB数据

校验和：BOOT EBL文件数据逐字节异或相加

应答格式：

|  |  |
| --- | --- |
| 1字节 | 76字节 |
| ACK | **m\_ebl\_response** |

### BOOT 下载结束

BOOT 下载结束数据格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bytes 0 | Bytes 1 | Bytes 2 | Bytes 3 | Bytes 4 | Bytes 5 | Bytes 6 | Bytes 7 | Bytes N |
| 校验和(checksum) | | 包类型(pack\_id) | | 包长度(payload\_length) | | | | 包数据(data) |

校验和：校验初始值+文件数据逐步字节相加累加和。

其中检验初始值为高16位与低16位的纯数字相加(checksum += ((payload\_length >> 16) + (payload\_length & 0x0000FFFF));)

文件数据逐步字节相加累加和:checksum += pack[EBL2\_16\_HEAD\_SIZE+i];

包类型:

见附表：pack\_id速查表

其中boot下载结束数据为：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bytes 0 | Bytes 1 | Bytes 2 | Bytes 3 | Bytes 4 | Bytes 5 | Bytes 6 | Bytes 7 | Bytes N |
| 校验和(checksum) | | ReqSetProtConf | | 0x4C(76字节) | | | | **m\_ebl\_response** |

BOOT 下载结束应答数据格式：

|  |
| --- |
| 2056字节 |
| ACK |

## **DOWNLOAD下载概述**

每个Download文件下载时序都类似，首先需要下载SecureBlock.bin文件，再下载LoadMap0.bin(LoadMap1.bin). 其中所有.fls后缀文件的SecureBloack.bin文件大小均为2048大小。无论下载何种文件都遵循以下格式。

Download下载数据包格式：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bytes 0 | Bytes 1 | Bytes 2 | Bytes 3 | Bytes 4 | Bytes 5 | Bytes 6 | Bytes 7 | Bytes N |
| 校验和(checksum) | | 包类型(pack\_id) | | 包长度(payload\_length) | | | | 包数据(data) |

校验和：校验初始值+文件数据逐步字节相加累加和。

其中检验初始值为高16位与低16位的纯数字相加(checksum += ((payload\_length >> 16) + (payload\_length & 0x0000FFFF));)

文件数据逐步字节相加累加和:

checksum += pack[EBL2\_16\_HEAD\_SIZE+i];

包类型:

见附表：pack\_id速查表

包长度：文件长度

包数据：对应文件长度的数据

Download下载数据包应答格式：

|  |
| --- |
| 16392字节 |
| ACK |

其ACK的第三个字节为应答的状态。若第三个字节为0x0041， 则表示应答失败。

失败明码为： ERR\_CLASS\_STD ERR\_CLASS\_SEC ERR\_CLASS\_TXT其余值为unkown。

Download下载结束包为：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bytes 0 | Bytes 1 | Bytes 2 | Bytes 3 | Bytes 4 | Bytes 5 | Bytes 6 | Bytes 7 | 2字节 |
| 0x0207 | | ReqSecEnd(0x0205) | | 0x02 | | | | 0x0000 |

Download下载结束应答包为：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bytes0 | Bytes1 | Bytes2 | Bytes3 | Bytes4 | Bytes5 | Bytes6 | Bytes7 | Bytes8 | Bytes9~N |
| 0x0A | 0x02 | 0x05 | 0x02 | 0x04 | 0 | 0 | 0 | 0x01 | 0 |

pack\_id速度查表：

|  |  |
| --- | --- |
| Pack\_id | 注释 |
| ReqRetransmission |  |
| CnfBaudChange |  |
| ReqCfiInfo\_1 |  |
| ReqCfiInfo\_2 |  |
| ReqSetProtConf |  |
| ReqProtocolChange |  |
| ReqSetLed |  |
| ReqSetRamWorkAddress |  |
| ReqWriteToRam |  |
| ReqExecuteFromRam |  |
| ReqSecStart |  |
| ReqSecEnd |  |
| ReqCloseHandle |  |
| ReqBootloaderVer |  |
| ReqForceHwReset |  |
| ReqSecChNonceAndVersion |  |
| ReqSecChValidateVerdict |  |
| ReqFlashId |  |
| ReqFlashSetAddress |  |
| ReqFlashReadBlock |  |
| ReqFlashWriteBlock |  |
| ReqFlashEraseStart |  |
| ReqFlashEraseCheck |  |
| ReqFlashReadChecksum |  |
| ReqCompressedPackage |  |
| ReqSetNandCtrlBitField |  |
| ReqSetNvmOptions |  |
| ReqFlashSetReadAddress |  |
| ReqOutOfSessionControl |  |
| ReqOutOfSessionDataWrite |  |
| ReqOutOfSessionDataRead |  |
| ReqFlashWriteImageRaw |  |