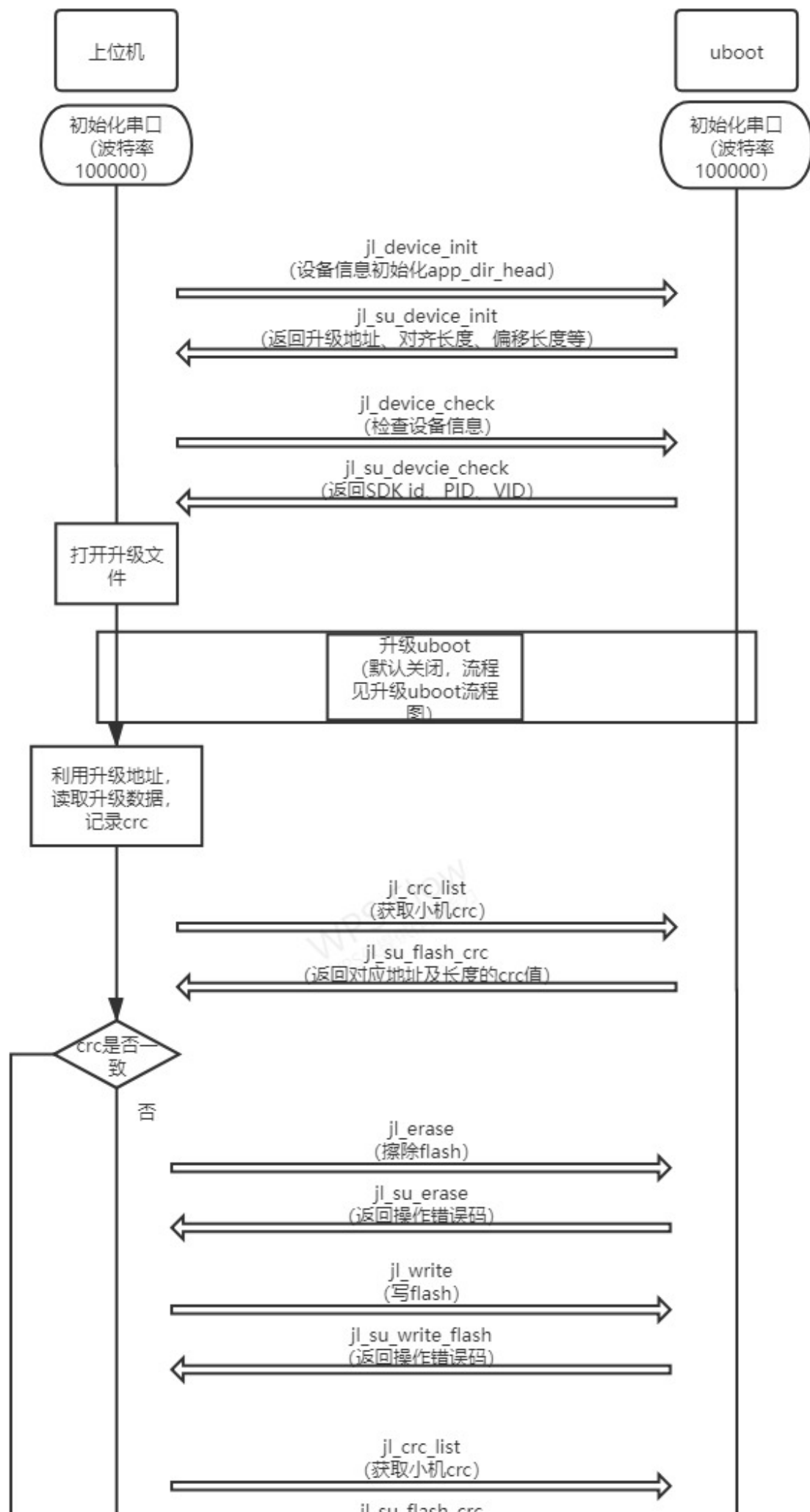


uboot 串口升级协议

版本号	备注
v1.0.0	初始版本

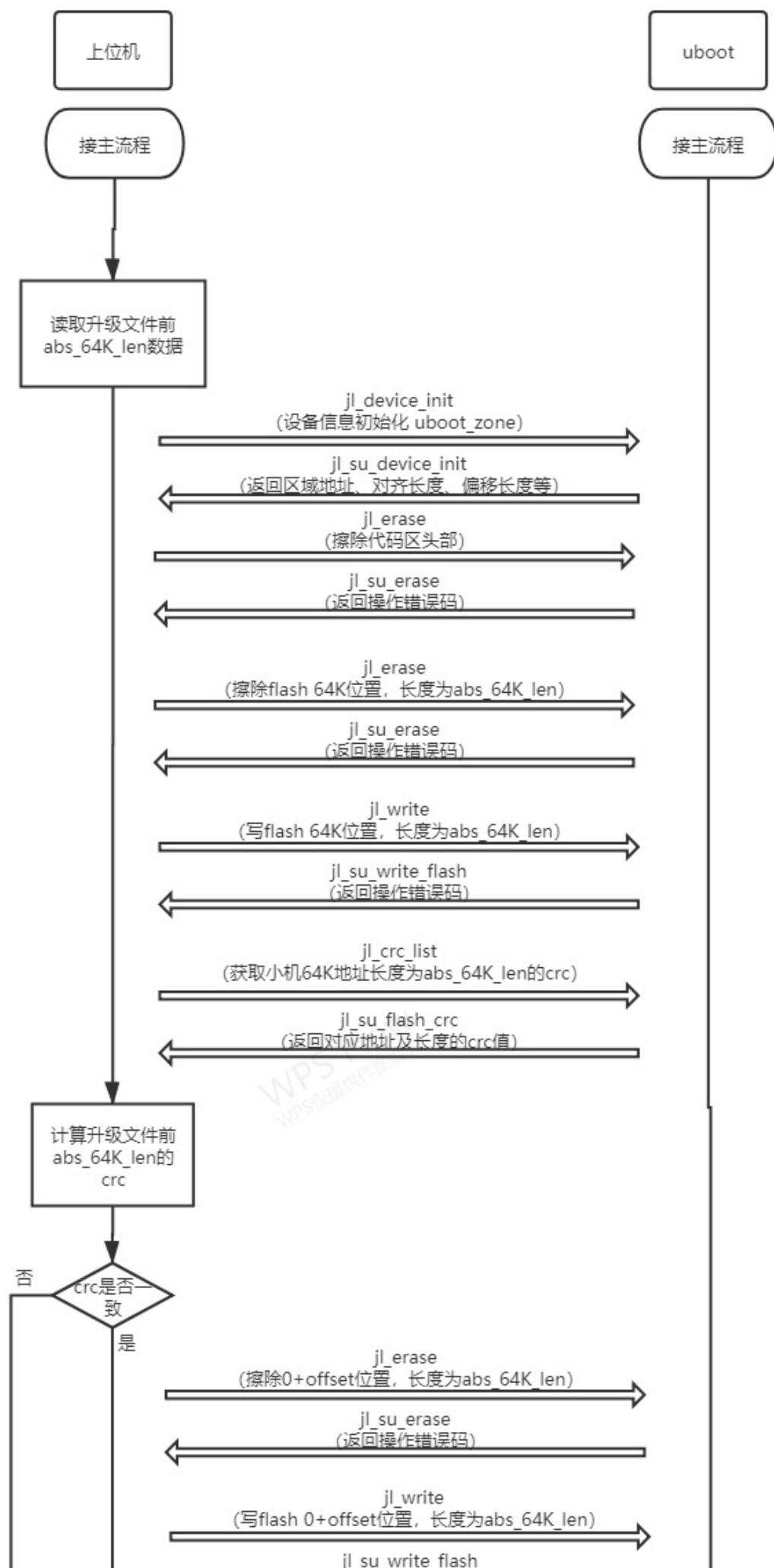
一、实现流程

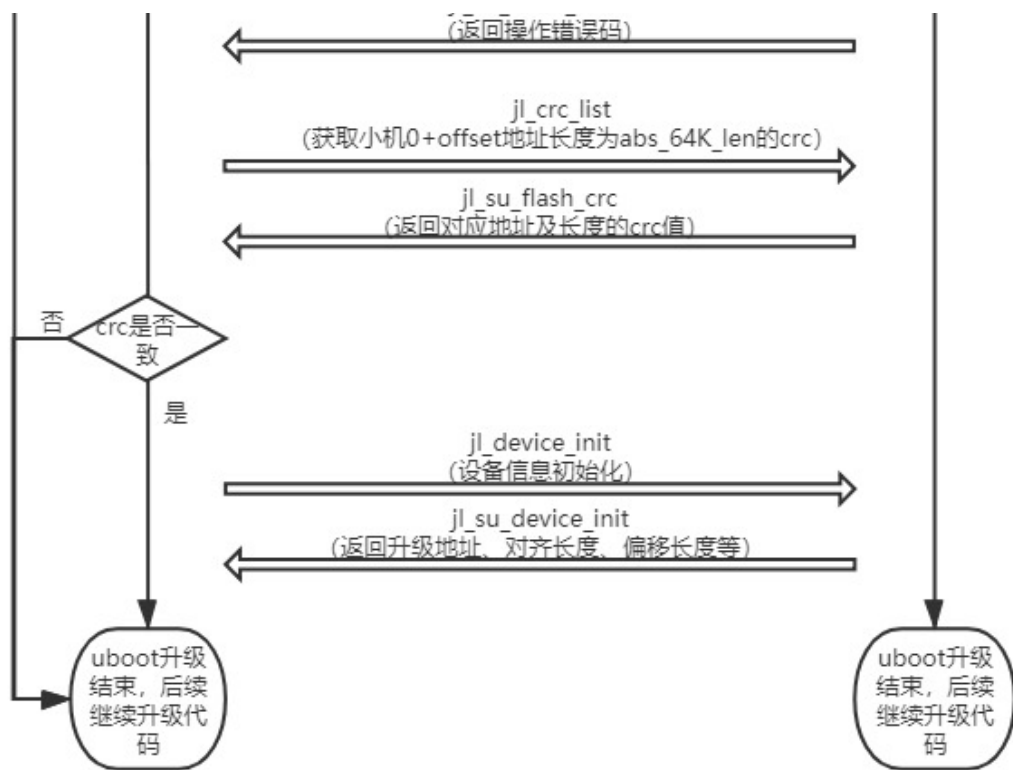
1.主流程图





2.uboot 本身升级流程图





二、协议说明

1.数据包格式

多字节数据：低字节在前，高字节在后，即小端格式(little endian)。

CRC16 标准：CRC-CCITT (XModem)。

回复使用相同指令。

demo只对cmd+rsp_status+param进行了加密，crc为加密后的crc。

Byte[0]	Byte[1]	Byte[3~2]	Byte[4]	Byte[5]	Byte[6~n-2]	Byte[n~n-1]	
syncdata0(0xAA)	syncdata1(0x55)	cmd_len	cmd	rsp_status	param(部分指令无)	crc16	

参数说明：

syncdata0: 固定为 0xAA

syncdata1: 固定为0x55

cmd_len: cmd + rsp_status + union 数据的 长度

cmd: 操作指令

rsp_status: 回复状态

param: 对应操作指令的参数（部分指令无参数）

crc16: 整个命令包（除去crc自身）的crc16结果

2.回复状态

状态	数值	说明
JL_SU_CMD_SUSS	0x0	命令执行成功
JL_SU_CMD_CRC_ERROR	0x1	CRC 错误
JL_SU_CMD_SDK_ID_ERROR	0x2	ID 错误
JL_SU_CMD_OTHER_ERROR	0x3	其他错误

```
enum {
    JL_SU_CMD_SUSS, //命令成功
    JL_SU_CMD_CRC_ERROR, //CRC 出错
    JL_SU_CMD_SDK_ID_ERROR, //ID 错误
    JL_SU_CMD_OTHER_ERROR, //其他错误
};
```

3.操作指令、参数及数据方向

1.JL_SU_CMD_DEVICE_INIT

定义值	说明
0xC0	设备初始化，获取对应 file_name 区域的地址长度等

- 上位机参数
- 方向：上位机->设备

	byte[0~15]	byte[16]
参数	file_name[16]	mode
说明	区域名字	读取模式 app_dir_head: 设置0 uboot_zone: 设置0

```
struct {
    u8 file_name[16];
    u8 mode;
} init;
```

- 设备回复参数
- 方向：设备->上位机

	byte[3~0]	byte[7~4]	byte[11~8]	byte[15~12]
参数	upgrade_addr	upgrade_len	upgrade_eoffset	flash_alignsize
说明	区域地址	区域长度	设备偏移长度	对齐长度

```

struct {
    u32 upgrade_addr;
    u32 upgrade_len;
    u32 upgrade_eoffset;
    u32 flash_alignsize;
} init;

```

2.JL_SU_CMD_DEVICE_CHECK

定义值	说明
0xC1	获取设备的 PID/VID/sdkID

- 上位机参数
- 方向：上位机->设备

	byte[3~0]
参数	sdk_id
说明	上位机sdk id

```

struct {
    u32 sdk_id;
}device_check;

```

- 设备回复参数
- 方向：设备->上位机

	byte[0~3]	byte[4~19]	byte[23~20]
参数	vid	pid	sdk_id
说明	设备 vid	设备 pid	设备sdk id

```

struct {
    u8 vid[4];
    u8 pid[16];
    u32 sdk_id;
}device_check;

```

3.JL_SU_CMD_ERASE

定义值	说明
0xC2	设备擦除指令

- 上位机参数
- 方向：上位机->设备

	byte[3~0]	byte[7~4]
参数	address	type
说明	擦除地址	擦除类型

```

#define JL_ERASE_TYPE_PAGE    1    //page擦除
#define JL_ERASE_TYPE_SECTOR  2    //sector
#define JL_ERASE_TYPE_BLOCK   3    //block
struct {
    u32 address;
    u32 type;
} erase;

```

- 设备回复参数:无 (由回复状态位rsp_status回复)
- 方向: 设备->上位机

4.JL_SU_CMD_WRITE

定义值	说明
0xC3	设备写指令

- 上位机参数
- 方向: 上位机->设备

	byte[3~0]	byte[7~4]	byte[8~n]
参数	address	data_length	data[0]
说明	写地址	数据长度	数据

```

struct {
    u32 address;        // 烧写FLASH地址
    u32 data_length;    // 烧写长度
    u8 data[0];         // 烧写数据
} write;

```

- 设备回复参数:无 (由回复状态位rsp_status回复)
- 方向: 设备->上位机

5.JL_SU_CMD_FLASH_CRC

定义值	说明
0xC4	获取设备 crc

- 上位机参数
- 方向: 上位机->设备

	byte[3~0]	byte[7~4]	byte[11~8]
参数	address	len	block_size
说明	需要校验的地址	校验总长度	单块数据校验长度

```
struct {
    u32 address;
    u32 len;
    u32 block_size;
} crc_list;
```

- 设备回复参数
- 方向：设备->上位机

	byte[0~n]
参数	crc[0]
说明	crc列表 (由len/block_size 个crc16组成)

```
struct {
    u16 crc[0];
} crc_list;
```

6.JL_SU_CMD_EX_KEY

定义值	说明
0xC5	交换密钥

- 上位机参数
- 方向：上位机->设备

	byte[3~0]
	secret_key
说明	密钥

```
struct {
    u32 secret_key;
} ex_key;
```

- 设备回复参数
-
- 方向：设备->上位机

	byte[1~0]	byte[3~2]	byte[n~4]
	rand	data_length	data[0]
说明	随机数	密钥长度	密钥

```
struct {
    u16 rand;
    u16 data_length;
    u8 data[0];
} ex_key;
```

7.JL_SU_CMD_REBOOT

定义值	说明
0xCA	复位（升级成功）指令，无返回

- 无参数

指令定义表：

```
#define JL_SU_CMD_DEVICE_INIT      0xC0
#define JL_SU_CMD_DEVICE_CHECK    0xC1
#define JL_SU_CMD_ERASE           0xC2
#define JL_SU_CMD_WRITE           0xC3
#define JL_SU_CMD_FLASH_CRC       0xC4
#define JL_SU_CMD_EX_KEY          0xC5
#define JL_SU_CMD_REBOOT          0xCA
```

上位机发送结构体参考（注意crc16不包含在结构体里）：

```
typedef struct {
    u8 syncdata0;
    u8 syncdata1;
    u16 cmd_len;
    u8 cmd;
    u8 rsp_status;
    union {
        struct {
            u32 secret_key;
        }ex_key;

        struct {
            u32 sdk_id;
        }device_check;

        struct {
            u32 address;           // 烧写FLASH地址
            u32 data_length;       //烧写长度
            u8 data[0];            // 烧写数据
        };
    };
};
```

```

    } write;

    struct {
        u32 address;    //需要对齐
        u32 type;
    } erase;

    struct {
        u32 address;
        u32 len;
        u32 block_size;
    } crc_list;

    struct {
        u8 file_name[16];
        u8 mode;
    } init;
};
u16 crc16;
} JL_SECTOR_COMMAND_ITEM;

```

设备回复结构体参考(注意crc16不包含在结构体里)

```

typedef struct {
    u8 syncdata0;
    u8 syncdata1;
    u16 cmd_len;
    u8 cmd;
    u8 rsp_status;
    union {
        struct {
            u32 secret_key;
        } ex_key;

        struct {
            u16 crc[0];
        } crc_list;

        struct {
            u32 upgrade_addr;
            u32 upgrade_len;
            u32 upgrade_eoffset;
            u32 flash_alignsize;
        } init;

        struct {
            u8 vid[4];
            u8 pid[16];
            u32 sdk_id;
        } device_check;
    };
    u16 crc16;
} JL_SECTOR_COMMAND_DEV_ITEM;

```

