

# *AC630N\_bt\_data\_transfer\_ sdk\_release\_v0.9.0 补丁*

版权所有 2022 杰理科技有限公司未经许可，禁止转载

## 目录

目录.....	2
AC630N_bt_data_transfer_sdk_release_v0.9.0.....	3
1. 替代工具.....	3
2. 替代 AC635N 的 loader、ota、uboot 文件.....	3
3. 替换 AC635N 的 update.a 和 cpu.a 文件.....	4
4. 修改 AC635N 下载目录下的 isd_config.ini 文件.....	4
5. 修改 AC635N 下载目录下的 testbox_update.c 文件.....	5
6. 修改 AC635N 下载目录下的 update_loader_download.h 文件.....	6
7. 修改 AC635N 下载目录下的 power_interface.h 文件.....	6
8. 替代 AC636N 的 loader、ota、uboot 文件.....	7
9. 替换 AC636N 的 update.a 和 cpu.a 文件.....	7
10. 修改 AC636N 下载目录下的 isd_config.ini 文件.....	8
11. 修改 AC636N 下载目录下的 testbox_update.c 文件.....	8
12. 修改 AC636N 下载目录下的 update_loader_download.h 文件.....	9
13. 修改 AC636N 下载目录下的 power_interface.h 文件.....	9

# AC630N\_bt\_data\_transfer\_sdk\_release\_v0.9.0

## 补丁描述:

日期: 2022-05-24

补丁说明: 该补丁分别更新了 AC635N 和 AC636N 的升级库、CPU 库、工具、OTA 文件, 解决了升级问题。

## 1. 替代工具

名称	修改日期	类型	大小
AC635N	2022-5-24 10:27	文件夹	
AC636N	2022-5-24 10:27	文件夹	
fw_add.exe	2022-5-21 14:17	应用程序	1,785 KB
isd_download.exe	2022-5-21 14:17	应用程序	2,098 KB
ufw_maker.exe	2022-5-21 14:17	应用程序	1,210 KB

将目录下 fw\_add.exe、isd\_download.exe 和 ufw\_maker.exe 工具放到对于的下载目录下。  
(对应的目录路径: sdk/cpu/br23/tools/)

## 2. 替代 AC635N 的 loader、ota、uboot 文件

名称	修改日期	类型	大小
文件修改	2022-5-24 9:35	文件夹	
cpu.a	2022-5-23 20:51	360压缩	1,035 KB
update.a	2022-5-21 14:18	360压缩	237 KB
<input checked="" type="checkbox"/> br23loader.bin	2022-5-21 14:17	BIN 文件	24 KB
<input checked="" type="checkbox"/> ota_all.bin	2022-5-21 14:17	BIN 文件	248 KB
<input checked="" type="checkbox"/> ota_all_debug.bin	2022-5-21 14:17	BIN 文件	356 KB
<input checked="" type="checkbox"/> ota_nor.bin	2022-5-21 14:17	BIN 文件	17 KB
<input checked="" type="checkbox"/> ota_nor_debug.bin	2022-5-21 14:17	BIN 文件	27 KB
<input type="checkbox"/> uboot.boot	2022-5-21 14:17	BOOT 文件	8 KB
<input type="checkbox"/> uboot_no_ota.boot	2022-5-21 14:17	BOOT 文件	5 KB
<input type="checkbox"/> uboot.boot_debug	2022-5-21 14:17	BOOT_DEBUG ...	13 KB
<input type="checkbox"/> uboot_no_ota.boot_debug	2022-5-21 14:17	BOOT_DEBUG ...	9 KB
<input type="checkbox"/> br23loader.uart	2022-5-21 14:17	UART 文件	11 KB

将目录下 loader、ota 和 uboot 文件放到对于的下载目录下。  
(对应的目录路径: sdk/cpu/br23/tools/)

### 3. 替换 AC635N 的 update.a 和 cpu.a 文件

名称	修改日期	类型	大小
文件修改	2022-5-24 9:35	文件夹	
cpu.a	2022-5-23 20:51	360压缩	1,035 KB
update.a	2022-5-21 14:18	360压缩	237 KB
<input checked="" type="checkbox"/> br23loader.bin	2022-5-21 14:17	BIN 文件	24 KB
<input checked="" type="checkbox"/> ota_all.bin	2022-5-21 14:17	BIN 文件	248 KB
<input checked="" type="checkbox"/> ota_all_debug.bin	2022-5-21 14:17	BIN 文件	356 KB
<input checked="" type="checkbox"/> ota_nor.bin	2022-5-21 14:17	BIN 文件	17 KB
<input checked="" type="checkbox"/> ota_nor_debug.bin	2022-5-21 14:17	BIN 文件	27 KB
<input type="checkbox"/> uboot.boot	2022-5-21 14:17	BOOT 文件	8 KB
<input type="checkbox"/> uboot_no_ota.boot	2022-5-21 14:17	BOOT 文件	5 KB
<input type="checkbox"/> uboot.boot_debug	2022-5-21 14:17	BOOT_DEBUG ...	13 KB
<input type="checkbox"/> uboot_no_ota.boot_debug	2022-5-21 14:17	BOOT_DEBUG ...	9 KB
<input type="checkbox"/> br23loader.uart	2022-5-21 14:17	UART 文件	11 KB

将目录下 cpu.a 和 update.a 文件替换工程对应的库。  
(对应的目录路径: sdk/include\_lib/liba/br23/)。

### 4. 修改 AC635N 下载目录下的 isd\_config.ini 文件

```

isd_config.ini
6
[EXTRA_CFG_PARAM]
NEW_FLASH_FS=YES; // 单备份
FORCE_VM_ALIGN=YES; // 强制VM对齐
CHIP_NAME=AC635N; // 芯片型号
ENTRY=0x1E00120; // 程序入口地址
PID=AC635N_TRANS; // 长度16byte, 示例: 芯片封装_应用方向_方案名称
VID=0.01;

RESERVED_OPT=0;
SPECIAL_OPT=0;
FORCE_4K_ALIGN=YES;

#DOWNLOAD_MODEL=SERIAL; //usb
DOWNLOAD_MODEL=usb; //
SERIAL_DEVICE_NAME=JLVirtualJtagSerial;
SERIAL_BARD_RATE=1000000;
SERIAL_CMD_OPT=0;
SERIAL_CMD_RATE=100; [n*10000]
SERIAL_CMD_RES=0;
SERIAL_INIT_BAUD_RATE=9600;
LOADER_BAUD_RATE=1000000;
LOADER_ASK_BAUD_RATE=1000000;
#SERIAL_SEND_KEY=YES;
BEFORE_LOADER_WAIT_TIME=100;
FLASH_BIN_CNT=0;
NEED_RESERVED_4K=YES; // 关闭4K保留

[CHIP_VERSION]
SUPPORTED_LIST=A,B,D;

#UBOOT配置项, 请勿随意调整顺序 #
[SYS_CFG_PARAM]
#data_width[0 1 2 3 4] 3的时候uboot自动识别2或者4线
#clk [0-255]
#mode:
37
#UBOOT配置项, 请勿随意调整顺序 #
[SYS_CFG_PARAM]
#data_width[0 1 2 3 4] 3的时候uboot自动识别2或者4线
#clk [0-255]
#mode:
42 # 0 RD_OUTPUT, 1 cmd 1 addr
43 # 1 RD_I/O, 1 cmd x addr
44 # 2 RD_I/O_CONTINUE] no_send_cmd x add
45 #port:
46 # 0 优先选A端口 CS:PD3 CLK:PD0 D0:PD1 D1:PD2 D2:PB7 D3:PB
47 # 1 优先选B端口 CS:PA13 CLK:PD0 D0:PD1 D1:PA14 D2:PA15 D3:PB
48 #SPI=2 3 0 0; #width_clk_mode_port;
49 #DSC=BIOS;
50 #DSC_FREQ=12MHz; #[24MHz 12MHz]
51 #SYS_CLK=24MHz; #[48MHz 24MHz]
52 #UTTX=PA05; //uboot串口tx
53 #UTBD=1000000; //uboot串口波特率
54 #UTRX=PB01; 串口升级[PB00 PB05 PA05]
55 RESET=PB01_00_0; //port口_长按时间_有效电平 (长按时间有00、01、0
位为秒, 当长按时间为00时, 则关闭长按复位功能。)
56 UPDATE_JUMP=0; //是否支持升级过程中维持I/O电平, 用于一些使用I/O
57
58 #0:disable
59 #1:PA9 PA10
60 #2:USB
61 #3:PB1 PB2
62 #4:PB6 PB7
63
64 #sdap=2;
65 #psram=0;
66
67 VLVD=4; //VDDIO_LVD档位, 0: 1.9V 1: 2.0V 2: 2.1V 3: 2.2V 4:
V 7: 2.6V
68
69 #EADDR=0; //ENTRY_ADDR;
70 EOFFSET=1; // n * 4k

```

```
1 isd_config.ini
04 VM_LEN=8K;
05 VM_OPT=1;
06
07 [BURNER_CONFIG]
08 SIZE=32;
09
10 [TOOL_CONFIG]
11 1T02_MIN_VER=2.26.5;
112 1T08_MIN_VER=3.1.10;
```

可参考文件修改目录下有对应的 isd\_config.ini 文件。

## 5. 修改 AC635N 下载目录下的 testbox\_update.c 文件

```
testbox_update.c
static void testbox_bt_classic_update_state_cbk(int type, u32 state, void *priv)
{
    update_ret_code_t *ret_code = (update_ret_code_t *)priv;
    switch (state) {
        case UPDATE_CH_EXIT:
            log_info("state:%x err:%x\n", ret_code->stu, ret_code->err_code);
            if (UPDATE_DUAL_BANK_IS_SUPPORT()) {
                if ((0 == ret_code->stu) && (0 == ret_code->err_code)) {
                    log_info("bt update succ\n");
                    update_result_set(UPDATA_SUCC);
                } else {
                    log_info("bt update fail\n");
                    update_result_set(UPDATA_DEV_ERR);
                }
            } else {
                if ((0 == ret_code->stu) && (0 == ret_code->err_code)) {
                    //update_mode_api(BT_UPDATA);
                    update_mode_api_v2(BT_UPDATA,
                        testbox_bt_classic_update_private_param_fill,
                        testbox_bt_classic_update_before_jump_handle);
                } else if ((0 == ret_code->stu)) { //文件相同的情况
                    cpu_reset();
                }
            }
            break;
    }
}

static u8 ble_update_ready_jump_flag = 0;
u8 ble_update_get_ready_jump_flag(void)
{
    return ble_update_ready_jump_flag;
}

static void testbox_ble_update_private_param_fill(UPDATA_PARAM *p)
{
    #endif
    } else if ((0 == ret_code->stu)) { //文件相同的情况,默认是复位
    #if TCFG_USER_BLE_ENABLE && (TCFG_BLE_DEMO_SELECT != DEF_BLE_DEMO_NULL) \
    && (TCFG_BLE_DEMO_SELECT != DEF_BLE_DEMO_ADV) || defined(CONFIG_MESH_CASE_ENABLE)) \
    && (TCFG_BLE_DEMO_SELECT != DEF_BLE_DEMO_CLIENT)
        extern void ble_module_enable(u8 en);
        ble_module_enable(0);
        u8 cnt = 0;
        while (!check_le_conn_disconnnet_flag()) {
            log_info("wait disconn\n");
            os_time_dly(2);
            if (cnt++ > 5) {
                break;
            }
        }
    #endif
    cpu_reset();
    break;
}

void testbox_update_msg_handle(int msg)
{
    log_info("msg:%x\n", msg);
    switch (msg) {
        case MSG_BT_UPDATE_LOADER_DOWNLOAD_START:
            if (UPDATE_MODULE_IS_SUPPORT(UPDATE_BT_LMP_EN)) {
                #if (SMART_BOX_EN) && (CONFIG_WATCH_CASE_ENABLE)
                    app_smartbox_task_prepare(0, SMARTBOX_TASK_ACTION_WATCH_TRANSFER, 0);
                #endif
                update_mode_info_t info = {
                    .type = BT_UPDATA,
                    .state_cbk = testbox_bt_classic_update_state_cbk,
                    .p_op_api = &lmp_ch_update_op,
                };
            }
    }
}
```

可参考文件修改目录下有对应的 testbox\_update.c 文件。

## 6. 修改 AC635N 下载目录下的 update\_loader\_download.h 文件

```
update_loader_download.h
68
//update result code;
enum {
    UPDATE_RESULT_ERR_NONE = 0,
    UPDATE_RESULT_FILE_SIZE_ERR = 0x1, //文件大小错误
    UPDATE_RESULT_LOADER_SIZE_ERR = 0x2, //loader大小错误
    UPDATE_RESULT_LOADER_VERIFY_ERR, //update loader校验失败
    UPDATE_RESULT_REMOTE_FILE_HEAD_ERR, //读升级文件头错误

    UPDATE_RESULT_LOCAL_FILE_HEAD_ERR = 0x5, //读flash文件头错误
    UPDATE_RESULT_NOT_FIND_TARGET_FILE_ERR, //找不到目标文件
    UPDATE_RESULT_FILE_OPERATION_ERR, //文件操作失败
    UPDATE_RESULT_FLASH_DATA_VERIFY_ERR, //flash数据校验失败

    UPDATE_RESULT_UBOOT_NOT_MATCH = 0x09, //UBOOT不匹配
    UPDATE_RESULT_PRODUCT_INFO_NOT_MATCH = 0x0a, //芯片型号不匹配
    UPDATE_RESULT_EX_DSP_UPDATE_ERR, //外部IC升级出错;
    UPDATE_RESULT_CFG_UPDATE_ERR, //配置升级出错

    UPDATE_RESULT_FLASH_ERASE_ERR = 0x0d, //flash 擦失败(可能是写保护)
    UPDATE_RESULT_REMOTE_FILE_NOT_MATCH, //升级文件不匹配
    UPDATE_RESULT_ANC_CFG_UPDATE_ERR, //ANC配置升级出错
    UPDATE_RESULT_ANC_CRC_UPDATE_ERR, //ANC配置升级出错
    UPDATE_RESULT_OTA_TWS_NO_RSP, //对耳启动升级传输数据没有回复
    UPDATE_RESULT_RESOURCE_LIMIT, //资源不足
    UPDATE_RESULT_OTA_TWS_START_ERR, //对耳启动升级失败
    UPDATE_RESULT_OTA_TWS_CRC_ERROR, //对耳校验失败
    UPDATE_RESULT_OTA_APP_EXIT, //升级过程APP强制退出
    UPDATE_RESULT_TWS_NO_CONNECT, //对耳未连接
    UPDATE_RESULT_FILE_SAME, //相同文件升级
};

#include "system/task.h"
typedef struct _update_type_info_t {
    int type;
    u8 task_en;
}

NORMAL  release/AC630N_bt_data_transfer_sdk_release_v0.9.0  update_loader_download.h  cpp  utf-8[unix]  33%  60/179 : 1
```

可参考文件修改目录下有对应的 update\_loader\_download.h 文件。

## 7. 修改 AC635N 下载目录下的 power\_interface.h 文件

```
power_interface.h
//Max hardware wakeup port
#define MAX_WAKEUP_PORT 8
144 #define MAX_RTC_PORT 2

struct wakeup_param {
    const PORT_FLT filter;
    const struct port_wakeup *port[MAX_WAKEUP_PORT];
    const struct port_wakeup *rtc_port[MAX_RTC_PORT];
    const struct charge_wakeup *charge;
    const struct alarm_wakeup *alarm;
    const struct lvd_wakeup *lvd;
    const struct sub_wakeup *sub;
};

struct reset_param {
    u8 en;
    u8 mode;
    u8 level;
    u8 lomap; //Port Group, Port Index
};

struct low_power_operation {
    const char *name;
    u32(*get_timeout)(void *priv);
    void (*suspend_probe)(void *priv);
    void (*suspend_post)(void *priv, u32 usec);
    void (*resume)(void *priv, u32 usec);
    void (*resume_post)(void *priv, u32 usec);
    void (*off_probe)(void *priv);
};

NORMAL  release/AC630N_bt_data_transfer_sdk_release_v0.9.0  ~/repo_data_trans_0520_del/SDK/include/lib/driver/cpu/br23/asm/power_interface.h  cpp  utf-8[unix]  47%  144/306 : 1
/\<MAX_RTC_PORT>
```

可参考文件修改目录下有对应的 power\_interface.h 文件。



## 8. 替代 AC636N 的 loader、ota、uboot 文件

名称	修改日期	类型	大小
文件修改	2022-5-23 21:03	文件夹	
cpu.a	2022-5-23 14:23	360压缩	1,003 KB
update.a	2022-5-21 15:43	360压缩	237 KB
<input checked="" type="checkbox"/> br25loader.bin	2022-5-21 14:32	BIN 文件	29 KB
<input checked="" type="checkbox"/> ota.bin	2022-5-21 14:32	BIN 文件	195 KB
<input checked="" type="checkbox"/> ota_debug.bin	2022-5-21 14:33	BIN 文件	268 KB
<input type="checkbox"/> uboot.boot	2022-5-21 14:32	BOOT 文件	4 KB
<input type="checkbox"/> uboot_no_ota.boot	2022-5-21 14:32	BOOT 文件	4 KB
<input type="checkbox"/> uboot.boot_debug	2022-5-21 14:32	BOOT_DEBUG ...	6 KB
<input type="checkbox"/> uboot_no_ota.boot_debug	2022-5-21 14:32	BOOT_DEBUG ...	5 KB
<input type="checkbox"/> br25loader.uart	2022-5-21 14:32	UART 文件	12 KB

将目录下 loader、ota 和 uboot 文件放到对应的下载目录下。

(对应的目录路径: sdk/cpu/br25/tools/)。

## 9. 替换 AC636N 的 update.a 和 cpu.a 文件

名称	修改日期	类型	大小
文件修改	2022-5-23 21:03	文件夹	
cpu.a	2022-5-23 14:23	360压缩	1,003 KB
update.a	2022-5-21 15:43	360压缩	237 KB
<input checked="" type="checkbox"/> br25loader.bin	2022-5-21 14:32	BIN 文件	29 KB
<input checked="" type="checkbox"/> ota.bin	2022-5-21 14:32	BIN 文件	195 KB
<input checked="" type="checkbox"/> ota_debug.bin	2022-5-21 14:33	BIN 文件	268 KB
<input type="checkbox"/> uboot.boot	2022-5-21 14:32	BOOT 文件	4 KB
<input type="checkbox"/> uboot_no_ota.boot	2022-5-21 14:32	BOOT 文件	4 KB
<input type="checkbox"/> uboot.boot_debug	2022-5-21 14:32	BOOT_DEBUG ...	6 KB
<input type="checkbox"/> uboot_no_ota.boot_debug	2022-5-21 14:32	BOOT_DEBUG ...	5 KB
<input type="checkbox"/> br25loader.uart	2022-5-21 14:32	UART 文件	12 KB

将目录下 cpu.a 和 update.a 文件替换工程对应的库。

(对应的目录路径: sdk/include\_lib/liba/br25/)。

## 10. 修改 AC636N 下载目录下的 isd\_config.ini 文件

```

1  #lsd.config.ini
2  #####UBOOT配置项, 请勿随意调整顺序#####
3  [SYS_CFG_PARAM]
4  # # # # # 0 1 2 3 4 3的时候uboot自动识别2或者4线
5  # #clk [0-255]
6  # #mode:
7  # # 0 RD_OUTPUT, 1 cmd 1 addr
8  # # 1 RD_I/O, 1 cmd x addr
9  # # 2 RD_I/O CONTINUE] no_send_cmd x add
10 # #port:
11 # # 0 优先选A端口 CS:PD3 CLK:PD0 D0:PD1 D1:PD2 D2:PB7 D3:PD5
12 # # 优先选B端口 CS:PA13 CLK:PD0 D0:PD1 D1:PA14 D2:PA15 D3:PD5
13 # #SPI-2_3_0_0; #width clk_mode_port;
14 # #OSC-btosc;
15 # #OSC_FREQ=12Mhz; #[24Mhz 12Mhz]
16 # #SYS_CLK=24Mhz; #[48Mhz 24Mhz]
17 # #BTIO-PB05; #串口通信
18 # #BTIO-1800000; #uboot串口波特率
19 # #UTRX-PB05; #串口升级 [PB00 PB05 PA05]
20 # #RESET-PB01_00_0; //port口_长按时间_有效电平 (长按时间有00、01、02、04、08三个值可选, 单
21 # #位为秒, 当长按时间为00时, 则关闭长按复位功能。)
22 # #UPDATE_JUMP=0;
23 # #0:disable
24 # #1:PA9 PA10
25 # #2:USB
26 # #3:PB1 PB2
27 # #4:PB6 PB7
28 # #
29 # #sdap=2;
30 # #psram=1;
31 # #
32 # #VLD=4; //VDDIO_LVD抽位, 0: 1.9V 1: 2.0V 2: 2.1V 3: 2.2V 4: 2.3V 5: 2.4V 6: 2.5
33 # #V: 2.6V
34 # #Erase=1; // n * 4k
35 #####
36 #####
37 #####
38 #####
39 #####
40 #####
41 #####
42 #####
43 #####
44 #####
45 #####
46 #####
47 #####
48 #####
49 #####
50 #####
51 #####
52 #####
53 #####
54 #####
55 #####
56 #####
57 #####
58 #####
59 #####
60 #####
61 #####
62 #####
63 #####
64 #####
65 #####
66 #####
67 #####
68 #####
69 #####
70 #####
71 #####
72 #####
73 #####
74 #####
75 #####
76 #####
77 #####
78 #####
79 #####
80 #####
81 #####
82 #####
83 #####
84 #####
85 #####
86 #####
87 #####
88 #####
89 #####
90 #####
91 #####
92 #####
93 #####
94 #####
95 #####
96 #####
97 #####
98 #####
99 #####
100 #####
101 #####
102 #####
103 #####
104 #####
105 #####
106 #####
107 #####
108 #####
109 #####
110 #####
111 #####
112 #####
113 #####
114 #####
115 #####
116 #####
117 #####
118 #####
119 #####
120 #####
121 #####
122 #####
123 #####
124 #####
125 #####
126 #####
127 #####
128 #####
129 #####
130 #####
131 #####
132 #####
133 #####
134 #####
135 #####
136 #####
137 #####
138 #####
139 #####
140 #####
141 #####
142 #####
143 #####
144 #####
145 #####
146 #####
147 #####
148 #####
149 #####
150 #####
151 #####
152 #####
153 #####
154 #####
155 #####
156 #####
157 #####
158 #####
159 #####
160 #####
161 #####
162 #####
163 #####
164 #####
165 #####
166 #####
167 #####
168 #####
169 #####
170 #####
171 #####
172 #####
173 #####
174 #####
175 #####
176 #####
177 #####
178 #####
179 #####
180 #####
181 #####
182 #####
183 #####
184 #####
185 #####
186 #####
187 #####
188 #####
189 #####
190 #####
191 #####
192 #####
193 #####
194 #####
195 #####
196 #####
197 #####
198 #####
199 #####
200 #####
201 #####
202 #####
203 #####
204 #####
205 #####
206 #####
207 #####
208 #####
209 #####
210 #####
211 #####
212 #####
213 #####
214 #####
215 #####
216 #####
217 #####
218 #####
219 #####
220 #####
221 #####
222 #####
223 #####
224 #####
225 #####
226 #####
227 #####
228 #####
229 #####
230 #####
231 #####
232 #####
233 #####
234 #####
235 #####
236 #####
237 #####
238 #####
239 #####
240 #####
241 #####
242 #####
243 #####
244 #####
245 #####
246 #####
247 #####
248 #####
249 #####
250 #####
251 #####
252 #####
253 #####
254 #####
255 #####
256 #####
257 #####
258 #####
259 #####
260 #####
261 #####
262 #####
263 #####
264 #####
265 #####
266 #####
267 #####
268 #####
269 #####
270 #####
271 #####
272 #####
273 #####
274 #####
275 #####
276 #####
277 #####
278 #####
279 #####
280 #####
281 #####
282 #####
283 #####
284 #####
285 #####
286 #####
287 #####
288 #####
289 #####
290 #####
291 #####
292 #####
293 #####
294 #####
295 #####
296 #####
297 #####
298 #####
299 #####
300 #####
301 #####
302 #####
303 #####
304 #####
305 #####
306 #####
307 #####
308 #####
309 #####
310 #####
311 #####
312 #####
313 #####
314 #####
315 #####
316 #####
317 #####
318 #####
319 #####
320 #####
321 #####
322 #####
323 #####
324 #####
325 #####
326 #####
327 #####
328 #####
329 #####
330 #####
331 #####
332 #####
333 #####
334 #####
335 #####
336 #####
337 #####
338 #####
339 #####
340 #####
341 #####
342 #####
343 #####
344 #####
345 #####
346 #####
347 #####
348 #####
349 #####
350 #####
351 #####
352 #####
353 #####
354 #####
355 #####
356 #####
357 #####
358 #####
359 #####
360 #####
361 #####
362 #####
363 #####
364 #####
365 #####
366 #####
367 #####
368 #####
369 #####
370 #####
371 #####
372 #####
373 #####
374 #####
375 #####
376 #####
377 #####
378 #####
379 #####
380 #####
381 #####
382 #####
383 #####
384 #####
385 #####
386 #####
387 #####
388 #####
389 #####
390 #####
391 #####
392 #####
393 #####
394 #####
395 #####
396 #####
397 #####
398 #####
399 #####
400 #####
401 #####
402 #####
403 #####
404 #####
405 #####
406 #####
407 #####
408 #####
409 #####
410 #####
411 #####
412 #####
413 #####
414 #####
415 #####
416 #####
417 #####
418 #####
419 #####
420 #####
421 #####
422 #####
423 #####
424 #####
425 #####
426 #####
427 #####
428 #####
429 #####
430 #####
431 #####
432 #####
433 #####
434 #####
435 #####
436 #####
437 #####
438 #####
439 #####
440 #####
441 #####
442 #####
443 #####
444 #####
445 #####
446 #####
447 #####
448 #####
449 #####
450 #####
451 #####
452 #####
453 #####
454 #####
455 #####
456 #####
457 #####
458 #####
459 #####
460 #####
461 #####
462 #####
463 #####
464 #####
465 #####
466 #####
467 #####
468 #####
469 #####
470 #####
471 #####
472 #####
473 #####
474 #####
475 #####
476 #####
477 #####
478 #####
479 #####
480 #####
481 #####
482 #####
483 #####
484 #####
485 #####
486 #####
487 #####
488 #####
489 #####
490 #####
491 #####
492 #####
493 #####
494 #####
495 #####
496 #####
497 #####
498 #####
499 #####
500 #####
501 #####
502 #####
503 #####
504 #####
505 #####
506 #####
507 #####
508 #####
509 #####
510 #####
511 #####
512 #####
513 #####
514 #####
515 #####
516 #####
517 #####
518 #####
519 #####
520 #####
521 #####
522 #####
523 #####
524 #####
525 #####
526 #####
527 #####
528 #####
529 #####
530 #####
531 #####
532 #####
533 #####
534 #####
535 #####
536 #####
537 #####
538 #####
539 #####
540 #####
541 #####
542 #####
543 #####
544 #####
545 #####
546 #####
547 #####
548 #####
549 #####
550 #####
551 #####
552 #####
553 #####
554 #####
555 #####
556 #####
557 #####
558 #####
559 #####
560 #####
561 #####
562 #####
563 #####
564 #####
565 #####
566 #####
567 #####
568 #####
569 #####
570 #####
571 #####
572 #####
573 #####
574 #####
575 #####
576 #####
577 #####
578 #####
579 #####
580 #####
581 #####
582 #####
583 #####
584 #####
585 #####
586 #####
587 #####
588 #####
589 #####
590 #####
591 #####
592 #####
593 #####
594 #####
595 #####
596 #####
597 #####
598 #####
599 #####
600 #####
601 #####
602 #####
603 #####
604 #####
605 #####
606 #####
607 #####
608 #####
609 #####
610 #####
611 #####
612 #####
613 #####
614 #####
615 #####
616 #####
617 #####
618 #####
619 #####
620 #####
621 #####
622 #####
623 #####
624 #####
625 #####
626 #####
627 #####
628 #####
629 #####
630 #####
631 #####
632 #####
633 #####
634 #####
635 #####
636 #####
637 #####
638 #####
639 #####
640 #####
641 #####
642 #####
643 #####
644 #####
645 #####
646 #####
647 #####
648 #####
649 #####
650 #####
651 #####
652 #####
653 #####
654 #####
655 #####
656 #####
657 #####
658 #####
659 #####
660 #####
661 #####
662 #####
663 #####
664 #####
665 #####
666 #####
667 #####
668 #####
669 #####
670 #####
671 #####
672 #####
673 #####
674 #####
675 #####
676 #####
677 #####
678 #####
679 #####
680 #####
681 #####
682 #####
683 #####
684 #####
685 #####
686 #####
687 #####
688 #####
689 #####
690 #####
691 #####
692 #####
693 #####
694 #####
695 #####
696 #####
697 #####
698 #####
699 #####
700 #####
701 #####
702 #####
703 #####
704 #####
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
74
```

可参考文件修改目录下有对应的 isd\_config.ini 文件。

## 11. 修改 AC636N 下载目录下的 testbox\_update.c 文件

```

1  testbox_update.c
2  static void testbox_bt_classic_update_state_cbk(int type, u32 state, void *priv)
3  {
4      update_ret_code_t *ret_code = (update_ret_code_t *)priv;
5
6      update_ret_code_t *ret_code = (update_ret_code_t *)priv;
7
8      switch (state) {
9      case UPDATE_CLI_EXIT:
10         log_info("state: %x err: %x\n", ret_code->stu, ret_code->err_code);
11         if (UPDATE_DUAL_BANK_IS_SUPPORT()) {
12             if ((0 == ret_code->stu) && (0 == ret_code->err_code)) {
13                 log_info("bt update suc\n");
14                 update_result_set(UPDATA_SUCC);
15             } else {
16                 log_info("bt update fail\n");
17                 update_result_set(UPDATA_DEV_ERR);
18             }
19         } else {
20             if ((0 == ret_code->stu) && (0 == ret_code->err_code)) {
21                 //update mode api(BT_UPDATA);
22                 update_mode_api_v2(BT_UPDATA,
23                                     testbox_bt_classic_update_private_param_fill,
24                                     testbox_bt_classic_update_before_jump_handle);
25             } else if ((0 == ret_code->stu)) { //文件相同的情况
26                 cpu_reset();
27             }
28         }
29         break;
30     }
31 }
32
33 static u8 ble_update_ready_jump_flag = 0;
34 u8 ble_update_get_ready_jump_flag(void)
35 {
36     return ble_update_ready_jump_flag;
37 }
38
39 static void testbox_ble_update_private_param_fill(UPDATA_PARAM *p)
40 {
41 }
42
43 #include <stdio.h>
44 #include <string.h>
45 #include <unistd.h>
46 #include <sys/types.h>
47 #include <sys/stat.h>
48 #include <fcntl.h>
49 #include <pthread.h>
50 #include <semaphore.h>
51 #include <sys/time.h>
52 #include <sys/mman.h>
53 #include <sys/wait.h>
54 #include <sys/resource.h>
55 #include <sys/queue.h>
56 #include <sys/socket.h>
57 #include <sys/un.h>
58 #include <sys/uio.h>
59 #include <sys/eventfd.h>
60 #include <sys/signalfd.h>
61 #include <sys/timerfd.h>
62 #include <sys/epoll.h>
63 #include <sys/ioctl.h>
64 #include <sys/poll.h>
65 #include <sys/select.h>
66 #include <sys/time.h>
67 #include <sys/times.h>
68 #include <sys/types.h>
69 #include <sys/uio.h>
70 #include <sys/wait.h>
71 #include <unistd.h>
72 #include <fcntl.h>
73 #include <pthread.h>
74 #include <semaphore.h>
75 #include <sys/time.h>
76 #include <sys/mman.h>
77 #include <sys/wait.h>
78 #include <sys/resource.h>
79 #include <sys/queue.h>
80 #include <sys/socket.h>
81 #include <sys/un.h>
82 #include <sys/uio.h>
83 #include <sys/eventfd.h>
84 #include <sys/signalfd.h>
85 #include <sys/timerfd.h>
86 #include <sys/epoll.h>
87 #include <sys/ioctl.h>
88 #include <sys/poll.h>
89 #include <sys/select.h>
90 #include <sys/time.h>
91 #include <sys/times.h>
92 #include <sys/types.h>
93 #include <sys/uio.h>
94 #include <sys/wait.h>
95 #include <unistd.h>
96 #include <fcntl.h>
97 #include <pthread.h>
98 #include <semaphore.h>
99 #include <sys/time.h>
100 #include <sys/mman.h>
101 #include <sys/wait.h>
102 #include <sys/resource.h>
103 #include <sys/queue.h>
104 #include <sys/socket.h>
105 #include <sys/un.h>
106 #include <sys/uio.h>
107 #include <sys/eventfd.h>
108 #include <sys/signalfd.h>
109 #include <sys/timerfd.h>
110 #include <sys/epoll.h>
111 #include <sys/ioctl.h>
112 #include <sys/poll.h>
113 #include <sys/select.h>
114 #include <sys/time.h>
115 #include <sys/times.h>
116 #include <sys/types.h>
117 #include <sys/uio.h>
118 #include <sys/wait.h>
119 #include <unistd.h>
120 #include <fcntl.h>
121 #include <pthread.h>
122 #include <semaphore.h>
123 #include <sys/time.h>
124 #include <sys/mman.h>
125 #include <sys/wait.h>
126 #include <sys/resource.h>
127 #include <sys/queue.h>
128 #include <sys/socket.h>
129 #include <sys/un.h>
130 #include <sys/uio.h>
131 #include <sys/eventfd.h>
132 #include <sys/signalfd.h>
133 #include <sys/timerfd.h>
134 #include <sys/epoll.h>
135 #include <sys/ioctl.h>
136 #include <sys/poll.h>
137 #include <sys/select.h>
138 #include <sys/time.h>
139 #include <sys/times.h>
140 #include <sys/types.h>
141 #include <sys/uio.h>
142 #include <sys/wait.h>
143 #include <unistd.h>
144 #include <fcntl.h>
145 #include <pthread.h>
146 #include <semaphore.h>
147 #include <sys/time.h>
148 #include <sys/mman.h>
149 #include <sys/wait.h>
150 #include <sys/resource.h>
151 #include <sys/queue.h>
152 #include <sys/socket.h>
153 #include <sys/un.h>
154 #include <sys/uio.h>
155 #include <sys/eventfd.h>
156 #include <sys/signalfd.h>
157 #include <sys/timerfd.h>
158 #include <sys/epoll.h>
159 #include <sys/ioctl.h>
160 #include <sys/poll.h>
161 #include <sys/select.h>
162 #include <sys/time.h>
163 #include <sys/times.h>
164 #include <sys/types.h>
165 #include <sys/uio.h>
166 #include <sys/wait.h>
167 #include <unistd.h>
168 #include <fcntl.h>
169 #include <pthread.h>
170 #include <semaphore.h>
171 #include <sys/time.h>
172 #include <sys/mman.h>
173 #include <sys/wait.h>
174 #include <sys/resource.h>
175 #include <sys/queue.h>
176 #include <sys/socket.h>
177 #include <sys/un.h>
178 #include <sys/uio.h>
179 #include <sys/eventfd.h>
180 #include <sys/signalfd.h>
181 #include <sys/timerfd.h>
182 #include <sys/epoll.h>
183 #include <sys/ioctl.h>
184 #include <sys/poll.h>
185 #include <sys/select.h>
186 #include <sys/time.h>
187 #include <sys/times.h>
188 #include <sys/types.h>
189 #include <sys/uio.h>
190 #include <sys/wait.h>
191 #include <unistd.h>
192 #include <fcntl.h>
193 #include <pthread.h>
194 #include <semaphore.h>
195 #include <sys/time.h>
196 #include <sys/mman.h>
197 #include <sys/wait.h>
198 #include <sys/resource.h>
199 #include <sys/queue.h>
200 #include <sys/socket.h>
201 #include <sys/un.h>
202 #include <sys/uio.h>
203 #include <sys/eventfd.h>
204 #include <sys/signalfd.h>
205 #include <sys/timerfd.h>
206 #include <sys/epoll.h>
207 #include <sys/ioctl.h>
208 #include <sys/poll.h>
209 #include <sys/select.h>
210 #include <sys/time.h>
211 #include <sys/times.h>
212 #include <sys/types.h>
213 #include <sys/uio.h>
214 #include <sys/wait.h>
215 #include <unistd.h>
216 #include <fcntl.h>
217 #include <pthread.h>
218 #include <semaphore.h>
219 #include <sys/time.h>
220 #include <sys/mman.h>
221 #include <sys/wait.h>
222 #include <sys/resource.h>
223 #include <sys/queue.h>
224 #include <sys/socket.h>
225 #include <sys/un.h>
226 #include <sys/uio.h>
227 #include <sys/eventfd.h>
228 #include <sys/signalfd.h>
229 #include <sys/timerfd.h>
230 #include <sys/epoll.h>
231 #include <sys/ioctl.h>
232 #include <sys/poll.h>
233 #include <sys/select.h>
234 #include <sys/time.h>
235 #include <sys/times.h>
236 #include <sys/types.h>
237 #include <sys/uio.h>
238 #include <sys/wait.h>
239 #include <unistd.h>
240 #include <fcntl.h>
241 #include <pthread.h>
242 #include <semaphore.h>
243 #include <sys/time.h>
244 #include <sys/mman.h>
245 #include <sys/wait.h>
246 #include <sys/resource.h>
247 #include <sys/queue.h>
248 #include <sys/socket.h>
249 #include <sys/un.h>
250 #include <sys/uio.h>
251 #include <sys/eventfd.h>
252 #include <sys/signalfd.h>
253 #include <sys/timerfd.h>
254 #include <sys/epoll.h>
255 #include <sys/ioctl.h>
256 #include <sys/poll.h>
257 #include <sys/select.h>
258 #include <sys/time.h>
259 #include <sys/times.h>
260 #include <sys/types.h>
261 #include <sys/uio.h>
262 #include <sys/wait.h>
263 #include <unistd.h>
264 #include <fcntl.h>
265 #include <pthread.h>
266 #include <semaphore.h>
267 #include <sys/time.h>
268 #include <sys/mman.h>
269 #include <sys/wait.h>
270 #include <sys/resource.h>
271 #include <sys/queue.h>
272 #include <sys/socket.h>
273 #include <sys/un.h>
274 #include <sys/uio.h>
275 #include <sys/eventfd.h>
276 #include <sys/signalfd.h>
277 #include <sys/timerfd.h>
278 #include <sys/epoll.h>
279 #include <sys/ioctl.h>
280 #include <sys/poll.h>
281 #include <sys/select.h>
282 #include <sys/time.h>
283 #include <sys/times.h>
284 #include <sys/types.h>
285 #include <sys/uio.h>
286 #include <sys/wait.h>
287 #include <unistd.h>
288 #include <fcntl.h>
289 #include <pthread.h>
290 #include <semaphore.h>
291 #include <sys/time.h>
292 #include <sys/mman.h>
293 #include <sys/wait.h>
294 #include <sys/resource.h>
295 #include <sys/queue.h>
296 #include <
```

可参考文件修改目录下有对应的 `testbox_update.c` 文件。（已经进行过 AC635N 的 `testbox_update.c` 文件修改可忽略这一步）



## 12. 修改 AC636N 下载目录下的 update\_loader\_download.h 文件

```
update_loader_download.h
68
//update result code;
enum {
    UPDATE_RESULT_ERR_NONE = 0,
    UPDATE_RESULT_FILE_SIZE_ERR = 0x1, //文件大小错误
    UPDATE_RESULT_LOADER_SIZE_ERR = 0x2, //loader大小错误
    UPDATE_RESULT_LOADER_VERIFY_ERR, //update loader校验失败
    UPDATE_RESULT_REMOTE_FILE_HEAD_ERR, //读升级文件头错误

    UPDATE_RESULT_LOCAL_FILE_HEAD_ERR = 0x5, //读flash文件头错误
    UPDATE_RESULT_NOT_FIND_TARGET_FILE_ERR, //找不到目标文件
    UPDATE_RESULT_FILE_OPERATION_ERR, //文件操作失败
    UPDATE_RESULT_FLASH_DATA_VERIFY_ERR, //flash数据校验失败

    UPDATE_RESULT_UBOOT_NOT_MATCH = 0x09, //UBOOT不匹配
    UPDATE_RESULT_PRODUCT_INFO_NOT_MATCH = 0x0a, //芯片型号不匹配
    UPDATE_RESULT_EX_DSP_UPDATE_ERR, //外部IC升级出错;
    UPDATE_RESULT_CFG_UPDATE_ERR, //配置升级出错

    UPDATE_RESULT_FLASH_ERASE_ERR = 0x0d, //flash 擦失败(可能是写保护)
    UPDATE_RESULT_REMOTE_FILE_NOT_MATCH, //升级文件不匹配
    UPDATE_RESULT_ANC_CFG_UPDATE_ERR, //ANC配置升级出错
    UPDATE_RESULT_ANC_COEF_UPDATE_ERR, //ANC配置升级出错
    UPDATE_RESULT_OTA_TWS_NO_RSP, //对耳同步升级传输数据没有回复
    UPDATE_RESULT_RESOURCE_LIMIT, //资源不足
    UPDATE_RESULT_OTA_TWS_START_ERR, //对耳启动升级失败
    UPDATE_RESULT_OTA_TWS_CRC_ERROR, //对耳校验失败
    UPDATE_RESULT_OTA_APP_EXIT, //升级过程APP强制退出
    UPDATE_RESULT_TWS_NO_CONNECT, //对耳未连接
    UPDATE_RESULT_FILE_SAME, //相同文件升级
};

#include "system/task.h"
typedef struct _update_type_info_t {
    int type;
    u8 task_en;
}

NORMAL  release/AC630N_bt_data_transfer_sdk_release_v0.9.0  update_loader_download.h  cpp  utf-8[unix]  33%  60/179 : 1
```

可参考文件修改目录下有对应的 update\_loader\_download.h 文件。（已经进行过 AC635N 的 update\_loader\_download.h 文件修改可忽略这一步）

## 13. 修改 AC636N 下载目录下的 power\_interface.h 文件

```
power_interface.h
//Max hardware wakeup port
#define MAX_WAKEUP_PORT 8
144 #define MAX_RTC_PORT 2

struct wakeup_param {
    const PORT_FLT filter;
    const struct port_wakeup *port[MAX_WAKEUP_PORT];
    const struct port_wakeup *rtc_port[MAX_RTC_PORT];
    const struct charge_wakeup *charge;
    const struct alarm_wakeup *alarm;
    const struct lvd_wakeup *lvd;
    const struct sub_wakeup *sub;
};

struct reset_param {
    u8 en;
    u8 mode;
    u8 level;
    u8 iomap; //Port Group, Port Index
};

struct low_power_operation {
    const char *name;
    u32(*get_timeout)(void *priv);
    void (*suspend_probe)(void *priv);
    void (*suspend_post)(void *priv, u32 usec);
    void (*resume)(void *priv, u32 usec);
    void (*resume_post)(void *priv, u32 usec);
    void (*off_probe)(void *priv);
};

NORMAL  release/AC630N_bt_data_transfer_sdk_release_v0.9.0  ~/repo_data_trans_0520_del/SDK/include_lib/driver/cpu/br23/asm/power_interface.h  cpp  utf-8[unix]  47%  144/306 : 1
/\<MAX_RTC_PORT>
```

可参考文件修改目录下有对应的 power\_interface.h 文件。（已经进行过 AC635N 的 power\_interface.h 文件修改可忽略这一步）