

发布时间：2022/09/19

此文档适用于 wireless\_mic\_sdk-v1.4.0\_2t1 SDK 及以上版本

说明：

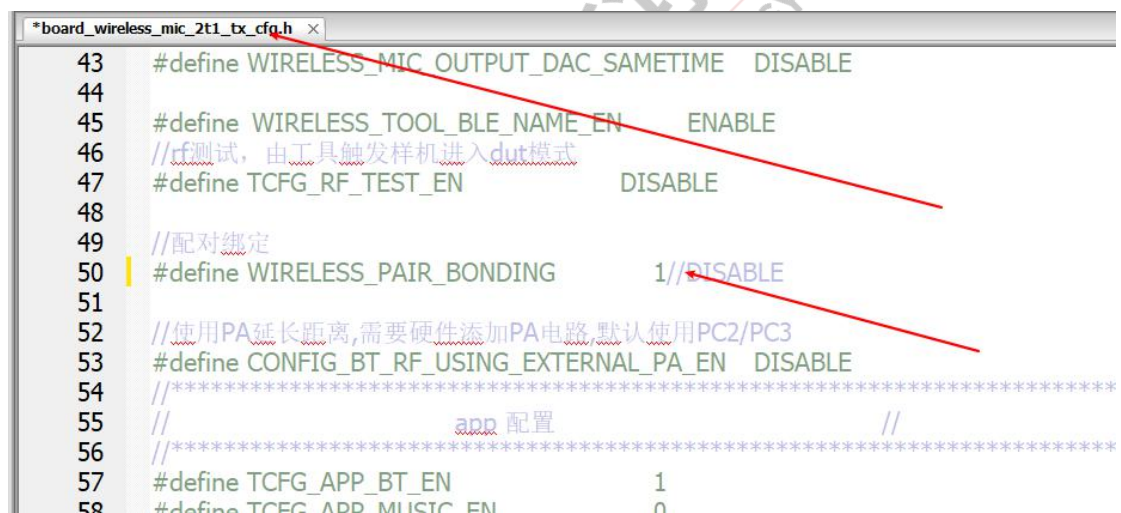
此文档用于说明如何添加按键实现两发一收 SDK 的绑定配对功能

## 1. 功能说明

绑定配对用于实现 TX 和 RX 的绑定连接，两个设备绑定之后，除非清除绑定记录，否则无法与其他设备进行连接。下面以双击按键清除绑定配对流程为例：

## 2. TX 端相关修改

### 2.1. 开启绑定配对的宏



### 2.2. 添加按键清除绑定记录

```

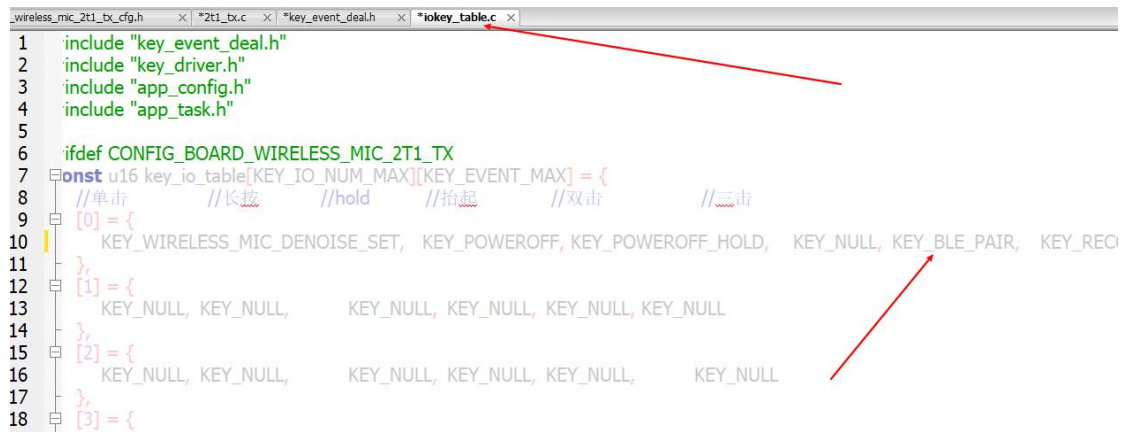
*board_wireless_mic_2t1_tx_cfg.h  x  *2t1_tx.c  x
487     if (flag_poweroff) {
488         if (++key_poweroff_cnt >= POWER_OFF_CNT) {
489             key_poweroff_cnt = 0;
490             ret = 1;
491         }
492     }
493
494     break;
495     case KEY_BLE_PAIR:
496     #if WIRELESS_PAIR_BONDING
497         clear_bonding_info();
498         ble_module_enable(1);
499     #endif // WIRELESS_PAIR_BONDING
500     break;
501     case KEY_NULL:
502     break;
503 }
504 return ret;
505 }
506

```

```

*board_wireless_mic_2t1_tx_cfg.h  x  *2t1_tx.c  x  *key_event_deal.h  x
185     KEY_WIRELESS_MIC_ECHO_SYNC, // 混响同步
186     KEY_ENTER_PAIR, // 1tn 进入配对模式
187     KEY_EXIT_PAIR, // 1tn 退出配对模式
188     KEY_WIRELESS_MIC_CH_SW, // 2t1通道切换
189     KEY_SW_SAMETIME_OUTPUT, // 开关同时输出到dac
190     KEY_MODE_SW, // 模式切换
191     KEY_RECORD_SW, // 开关录音
192     KEY_WIRELESS_2t1_RX_SEND_DATA, // 两发一收rx发送数据
193     KEY_BLE_PAIR,
194     // 不会出现在按键主流程，用于不重要得其他操作
195     KEY_MINOR_OPT,
196
197     KEY_NULL = 0xFFFF,
198
199     KEY_MSG_MAX = 0xFFFF,
200     // 音箱sdk 按键消息已经加太为0xffff
201 };
202
203

```

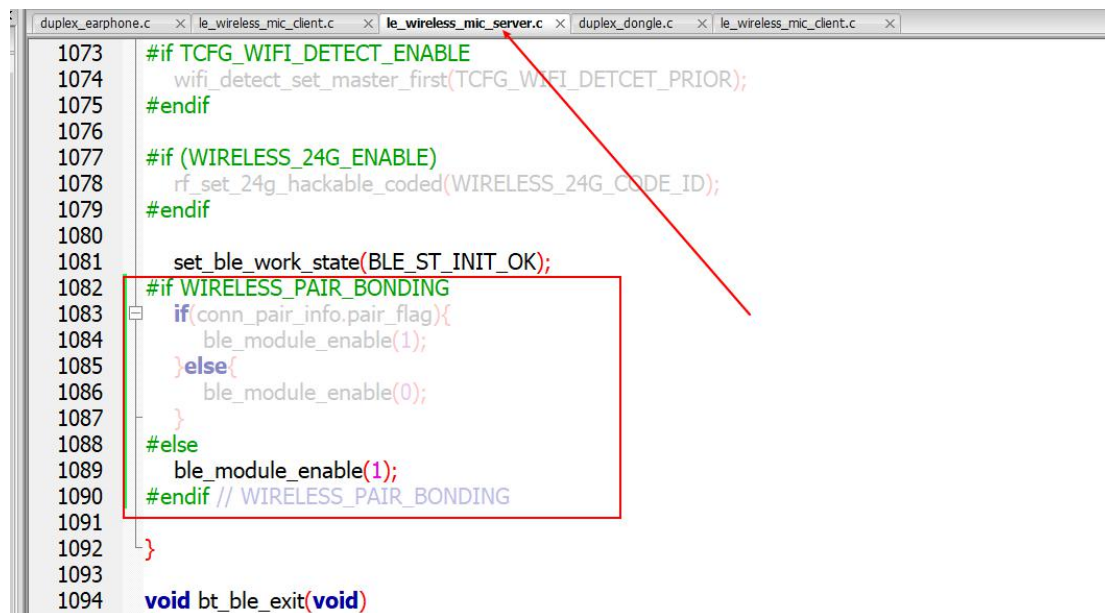


```

1  include "key_event_deal.h"
2  include "key_driver.h"
3  include "app_config.h"
4  include "app_task.h"
5
6  #ifdef CONFIG_BOARD_WIRELESS_MIC_2T1_TX
7  #const u16 key_io_table[KEY_IO_NUM_MAX][KEY_EVENT_MAX] = {
8      //单击      //长按      //hold      //抬起      //双击      //三击
9      [0] = {
10         KEY_WIRELESS_MIC_DENOISE_SET, KEY_POWEROFF, KEY_POWEROFF_HOLD, KEY_NULL, KEY_BLE_PAIR, KEY_REC
11     },
12     [1] = {
13         KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL
14     },
15     [2] = {
16         KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL
17     },
18     [3] = {

```

## 2.3. 控制第一次上电连接状态



```

1073  #if TCFG_WIFI_DETECT_ENABLE
1074      wifi_detect_set_master_first(TCFG_WIFI_DETECT_PRIOR);
1075  #endif
1076
1077  #if (WIRELESS_24G_ENABLE)
1078      rf_set_24g_hackable_coded(WIRELESS_24G_CODE_ID);
1079  #endif
1080
1081      set_ble_work_state(BLE_ST_INIT_OK);
1082  #if WIRELESS_PAIR_BONDING
1083      if(conn_pair_info.pair_flag){
1084          ble_module_enable(1);
1085      }else{
1086          ble_module_enable(0);
1087      }
1088  #else
1089      ble_module_enable(1);
1090  #endif // WIRELESS_PAIR_BONDING
1091
1092  }
1093
1094  void bt_ble_exit(void)

```

## 3. RX 端相关修改

### 3.1. 开启绑定配对的宏

```
*board_wireless_mic_2t1_rx_cfg.h x
43
44 #define WIRELESS_TOOL_BLE_NAME_EN    ENABLE
45 //rf测试, 由工具触发样机进入 dut模式
46 #define TCFG_RF_TEST_EN              DISABLE
47 //产线近距离快速配对测试功能
48 //如果配对名以特殊字符串开头, 程序会判断信号强度来进行连接, 即靠得近的才可以连上
49 #define SPECIFIC_STRING              "@#@#@@"//用户可自定义修改
50 #define TCFG_WIRELESS_RSSI           50//demo板测试50大概对应1米的距离, 越大越远
51
52 //配对绑定
53 #define WIRELESS_PAIR_BONDING        1/DISABLE
54
55 //使用PA延长距离, 需要硬件添加PA电路, 默认使用PC2/PC3
56 #define CONFIG_BT_RF_USING_EXTERNAL_PA_EN  DISABLE
57
58 //*****
59 //*****
60 //
```

### 3.2. 添加按键清除绑定配对

```
*board_wireless_mic_2t1_rx_cfg.h x *2t1_rx.c x 2t1_tx.c x
496
497 if (flag_poweroff) {
498     if (++key_poweroff_cnt >= POWER_OFF_CNT) {
499         key_poweroff_cnt = 0;
500         ret = 1;
501     }
502 }
503 break;
504 case KEY_BLE_PAIR:
505     #if WIRELESS_PAIR_BONDING
506         clear_bonding_info();
507         ble_module_enable(1);
508     #endif // WIRELESS_PAIR_BONDING
509     break;
510
511 case KEY_NULL:
512     break;
513 }
514 return ret;
515 }
```



```

duplex_dongle.c  x board_wireless_duplex_dongle_cfg.h  x le_wireless_mic_client.c  x key_event_deal.h  x key_ever
187 KEY_EXIT_PAIR, //1tn 退出配对模式
188 KEY_WIRELESS_MIC_CH_SW, //2t1通道切换
189 KEY_SW_SAMETIME_OUTPUT, //开关同时输出到dac
190 KEY_MODE_SW, //模式切换
191 KEY_RECORD_SW, //开关录音
192 KEY_WIRELESS_2t1_RX_SEND_DATA, //两发一收rx发送数据
193 KEY_BLE_PAIR,
194 //不会出现在按键主流程, 用于不重要得其他操作
195 KEY_MINOR_OPT,
196
197 KEY_NULL = 0xFFFF,
198
199 KEY_MSG_MAX = 0xFFFF,
200 //音箱sdk 按键消息已经加大为0xffff
201 };
202
203
204 #endif

```

```

board_wireless_mic_2t1_rx_cfg.h  x *2t1_rx.c  x 2t1_tx.c  x iokey_table.c  x *iokey_table.c  x
1 nt_deal.h"
2 ver.h"
3 ifig.h"
4 k.h"
5
6 JARD_WIRELESS_MIC_2T1_RX
7 table[KEY_IO_NUM_MAX][KEY_EVENT_MAX] = {
8 //长按 //hold //抬起 //双击 //三击
9
10 ESS_MIC_DENOISE_SET, KEY_POWEROFF, KEY_POWEROFF_HOLD, KEY_NULL, KEY_BLE_PAIR, KEY_WIRELESS_MIC_
11
12 KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL
13
14 KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL
15
16 KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL, KEY_NULL
17

```

### 3.3. 控制第一次上电连接状态

```

board_wireless_mic_2t1_rx_cfg.h  x 2t1_rx.c  x 2t1_tx.c  x iokey_table.c  x iokey_table.c  x le_multi_client.c  x key_event_deal.h  x
1921 #endif
1922
1923 #if WIRELESS_PAIR_BONDING
1924 for (int i = 0; i < SUPPORT_MAX_CLIENT; i++) {
1925 if (conn_pair_info_table[i].pair_flag) {
1926 ble_client_module_enable(1);
1927 break;
1928 } else {
1929 ble_client_module_enable(0);
1930 }
1931 }
1932 #else
1933 ble_client_module_enable(1);
1934 #endif
1935
1936
1937 #if TCFG_WIFI_DETECT_ENABLE
1938 wifi_detect_set_master_first(TCFG_WIFI_DETECT_PRIOR);
1939 #endif
1940
1941 #if SHOW_RX_DATA_RATE
1942 client_timer_start();
1943 #endif /* SHOW_RX_DATA_RATE */

```