

2.4G 遥控车_示例_介绍

2.4G 遥控车_示例_介绍 用户手册

Rev 1.0.0—— 2021 年 9 月 29 日

This translated version is for reference only, and the English version shall prevail in case of any discrepancy between the translated and English versions.

版权所有 2021 杰理科技有限公司未经许可，禁止转载

目录

Chapter 1 环境搭建支持.....	5
1.1 说明.....	6
1.2 主机配置.....	7
1.2.1 工程配置.....	7
1.3 从机配置.....	9
1.3.1 工程配置.....	9
Chapter 2 发送数据修改支持.....	12
2.1 发送概述.....	13
2.2 修改 ATT_SEND_CBUF_SIZE.....	14
2.2.1 概述.....	14
2.2.2 修改依据.....	14
2.2.3 具体修改.....	14
2.3 修改 btctrlr 配置.....	15
2.3.1 概述.....	15
2.3.2 具体修改.....	15
Chapter 3 物理通道（PHY）修改支持.....	16
3.1 PHY 概述.....	17
3.2 修改 1M PHY.....	18
3.2.1 概述.....	18
3.2.2 具体修改.....	18
3.3 修改 2M PHY.....	19
3.3.1 概述.....	19
3.3.2 具体修改.....	19
3.4 修改 S2 PHY.....	20
3.4.1 概述.....	20
3.4.2 具体修改.....	20
3.5 修改 S8 PHY.....	21
3.5.1 概述.....	21
3.5.2 具体修改.....	21
Chapter 4 2.4G 配对码修改.....	22
4.1 2.4G 配对码修改.....	23

4.1.1 概述.....	23
4.1.2 主机配对码修改.....	23
4.1.3 从机配对码修改.....	23
Chapter 5 距离测试修改支持.....	25
5.1 距离测试.....	26
5.1.1 概述.....	26
5.1.2 距离测试.....	26

修改日志

版本	日期	描述
1.0.0	2021 / 9 / 29	用户手册
更新:	<ul style="list-style-type: none">● 建立初始版本● 定义文档格● 客户支持	

Chapter 1 环境搭建支持

1.1 说明

本案例用于基于 BLE 的 2.4G 遥控车功能

支持的板级：br30、br34

支持的芯片：AC637N、AC638N

面向环境为：SDK200 版本。主机：遥控 2.4G 主机 case，从机：遥控 2.4G 从机 case

1.2 主机配置

1.2.1 工程配置

代码工程: \apps\spp_and_le\board\brxx\AC63XN_spp_and_le.cbp

(1) 配置板级 board_config.h 为 AC63XN_DEMO, 并且配置 AC63XN_DEMO 对应文件中 BLE 模式使能、低功耗开关

```
1.  /*
2.   *   板级配置选择
3.   */
4.
5.  #define CONFIG_BOARD_AC637N_DEMO
6.  // #define CONFIG_BOARD_AC6373B_DEMO
7.  // #define CONFIG_BOARD_AC6376F_DEMO
8.  // #define CONFIG_BOARD_AC6379B_DEMO
9.
10. #include "board_ac637n_demo_cfg.h"
11. #include "board_ac6373b_demo_cfg.h"
12. #include "board_ac6376f_demo_cfg.h"
13. #include "board_ac6379b_demo_cfg.h"
```

//在 board_ac63xn_demo_cfg.h 中配置是否打开 edr 和 ble 模块

```
1.  #define TCFG_USER_BLE_ENABLE          1    //BLE 功能使能
2.  #define TCFG_USER_EDR_ENABLE          0    //EDR 功能使能
```

//在 board_ac63xn_demo_cfg.h 中配置是否打开 低功耗模块

```
1.  #define TCFG_LOWPOWER_LOWPOWER_SEL    0    // 芯片是否进入 powerdown
```

(2) 配置 app 选择: “app_config.h”, 配置 case 为 REMOTE_24G_C case

```
1.  //apps example 选择, 只能选 1 个, 要配置对应的 board_config.h
2.  #define CONFIG_APP_SPP_LE              0    //SPP + LE or LE's client
```

```

3. #define CONFIG_APP_MULTII          0 //蓝牙LE 多连 + spp
4. #define CONFIG_APP_DONGLE          0 //usb + 蓝牙(ble 主机),PC hid 设备
5. #define CONFIG_APP_CENTRAL          0 //ble client, 中心设备
6. #define CONFIG_APP_LL_SYNC          0 //腾讯连连
7. #define CONFIG_APP_BEACON           0 //蓝牙BLE ibeacon
8. #define CONFIG_APP_NONCONN_24G      0 //2.4G 非连接收发
9. #define CONFIG_APP_TUYA             0 //涂鸦协议
10. #define CONFIG_APP_AT_COM           0 //AT com HEX 格式命令
11. #define CONFIG_APP_AT_CHAR_COM      0 //AT com 字符串格式命令
12. #define CONFIG_APP_IDLE             0 //空闲任务
13. #define CONFIG_APP_REMOTE_24G_C     1 //基于BLE 的2.4g 遥控主机, 板级只需要开BLE

```

(3) 配置 REMOTE_24G_C case 的需求: “app_config.h”, 配置 GATT 公共模块使能、是否走加密流程、GATT 支持从机个数、连接个数

```

1. #elif CONFIG_APP_DONGLE
2. #define EDR_EMITTER_EN              0 //蓝牙(edr 主机)
3.
4. #define CONFIG_BT_GATT_COMMON_ENABLE 1 //GATT 公共模块使能
5. #define CONFIG_BT_SM_SUPPORT_ENABLE 0 //是否支持加密流程
6. #define CONFIG_BT_GATT_CLIENT_NUM   1 //配置 GATT 从机个数
7. #define CONFIG_BT_GATT_SERVER_NUM   0 //(app not support)
8. #define CONFIG_BT_GATT_CONNECTION_NUM (CONFIG_BT_GATT_SERVER_NUM + CONFIG_BT_GATT_CLIENT_NUM) //连接个数

```

(4) 配置搜索从机名字: “ble_remote_24g_client.c”, 设置需要搜索从机名为“REMOTE_24G(BLE)”, 默认开了三个名字配对, 选填一个正确的就可以。若用户只需要一个名字, 可以将别的“”的名字删除

```

1. //配置多个扫描匹配设备
2. static const u8 remote_test_remoter_name1[] = "REMOTE_24G(BLE)";//
3. static const u8 remote_test_remoter_name2[] = "AC630N_mx(BLE)";//
4. static const u8 user_config_tag_string[] = "abc123";

```


1.3 从机配置

1.3.1 工程配置

代码工程：\apps\hid\board\brxx\AC63XN_hid.cbp

(1) 配置板级 board_config.h 为 AC63XN_DEMO，以及配置 AC63XN_DEMO 对应文件中 BLE 模式使能、低功耗开关

```
1.  /*
2.   *   板级配置选择
3.   */
4.
5.  #define CONFIG_BOARD_AC637N_DEMO
6.  // #define CONFIG_BOARD_AC6376F_DEMO
7.  // #define CONFIG_BOARD_AC6373B_MOUSE
8.  // #define CONFIG_BOARD_AC6379B_MOUSE
9.  // #define CONFIG_BOARD_AC6379B_KEYFOB
10. // #define CONFIG_BOARD_AC6376F_MOUSE
11.
12. #include "board_ac637n_demo_cfg.h"
13. #include "board_ac6376f_demo_cfg.h"
14. #include "board_ac6373b_mouse_cfg.h"
15. #include "board_ac6379b_mouse_cfg.h"
16. #include "board_ac6379b_keyfob_cfg.h"
17. #include "board_ac6376f_mouse_cfg.h"
18. #endif
```

//在 board_ac63xn_demo_cfg.h 中配置是否打开 edr 和 ble 模块

```
3.  #define TCFG_USER_BLE_ENABLE          1    //BLE 功能使能
4.  #define TCFG_USER_EDR_ENABLE          0    //EDR 功能使能
```

//在 board_ac63xn_demo_cfg.h 中配置是否打开低功耗模块

```
2. #define TCFG_LOWPOWER_LOWPOWER_SEL 0 // 芯片是否进入 powerdown
```

(2) 配置 app 选择: “app_config.h”, 配置 case 为 REMOTE_24G_S case

```
1. //app case 选择, 只选1, 要配置对应的board_config.h
2. #define CONFIG_APP_KEYBOARD 0//hid 按键 ,default case
3. #define CONFIG_APP_KEYFOB 0// 自拍
   器, board_ac6368a,board_6318,board_6379b
4. #define CONFIG_APP_MOUSE_SINGLE 0// 单模切换
5. #define CONFIG_APP_MOUSE_DUAL 0// 同时开双模
6. #define CONFIG_APP_STANDARD_KEYBOARD 0// 标准HID 键盘,board_ac6351d
7. #define CONFIG_APP_KEYPAGE 0// 翻页器
8. #define CONFIG_APP_GAMEBOX 0// 吃鸡王座
9. #define CONFIG_APP_REMOTE_CONTROL 0// 语音遥控
10. #define CONFIG_APP_IDLE 0//IDLE
11. #define CONFIG_APP_REMOTE_24G_S 1// 基于BLE 的2.4g 遥控从机, 板级配置只需要打
    开BLE
```

(3) 配置 REMOTE_24G_S case 需求: “app_config.h”, 配置 GATT 公共模块使能、是否走加密流程、GATT 支持主机个数、连接个数

```
1. // 蓝牙BLE 配置
2. #define CONFIG_BT_GATT_COMMON_ENABLE 1 // 配置使用gatt 公共模块
3. #define CONFIG_BT_SM_SUPPORT_ENABLE 0 // 配置是否支持加密
4. #define CONFIG_BT_GATT_CLIENT_NUM 0 // 配置主机client 个数(app not support)
5. #define CONFIG_BT_GATT_SERVER_NUM 1 // 配置从机server 个数
6. #define CONFIG_BT_GATT_CONNECTION_NUM (CONFIG_BT_GATT_SERVER_NUM + CONFIG_BT_GA
    TT_CLIENT_NUM) // 配置连接个数
```

(5) 配置从机名字为 REMOTE_24G: 打开\cpu\br30\tools\AC897N_config_tool 目录下的 AC897N_配置工具入口(Config Tools Entry).jlxproj 工具, 点击编译前配置工具并打开, 选择蓝牙配置设置名字为 REMOTE_24G, 如下图所示



完成之后选择保存，再选择全部保存完成配置

Chapter 2 发送数据修改支持

2.1 发送概述

本章主要介绍怎样设置发送缓存大小、基带发送缓存包数的应用方案，让用户能从最接近目标产品的 APP 开始开发。

2.2 修改 ATT_SEND_CBUF_SIZE

2.2.1 概述

修改 ATT_SEND_CBUF_SIZE 参数能够设置发送缓存大小，即设置发送缓存空间，增大参数使发送缓存里面的空间变大，能够存放更多的发送数据，缺点就是实时性较差，减小参数能使空间变小的，提高实时性。因此调节此参数控制了传输精度。

2.2.2 修改依据

$$\text{ATT_SEND_CBUF_SIZE} = \text{L2CAP Len(2bytes)} + \text{L2CAP Ch ID(2bytes)} + \text{Opcode \& Handle(3bytes)} + \text{Send data(0\sim 244bytes)}$$
，分别代表着链路层数据的长度、链路层通道 ID 号、opcode 和 ATT handle、发送数据大小，示意图如下：因为我司标准键盘 Send data 为 8bytes，即一个 ATT_SEND_CBUF_SIZE = 15 bytes

LL Header	L2CAP Len	L2CAP Ch ID	Opcode & HDL	Attribute Value
(2)	(2)	(2)	(3)	(0 ~ 244)

2.2.3 具体修改

在 HID 传输过程中按键是需要抬键操作的，即按下一个按键其实是发送了一个按键按下消息和一个按键抬起消息，因此想要设置发送缓存为一次按键：ATT_SEND_CBUF_SIZE = 30 bytes

具体修改位于 ble_hogp.c 文件中

```
1. //ATT 发送的包长, note: 20 <= need >= MTU
2. #define ATT_LOCAL_MTU_SIZE (64)
3. //ATT 缓存的 buffer 大小, note: need >= 20, 可修改
4. #define ATT_SEND_CBUF_SIZE (30) //发送缓存设置为 30 bytes
5. // 广播周期 (unit:0.625ms)
6. #define ADV_INTERVAL_MIN (160 * 5)
```

2.3 修改 btctrler 配置

2.3.1 概述

修改 config_btctrler_le_acl_total_nums 参数能够设置基带缓存包数，设置它一次能够缓存包数。一般来说这个参数默认为 3，不做修改，目的是为了减少缓存。

2.3.2 具体修改

具体修改位于 lib_btctrler_config.c 文件中,CONFIG_BT_GATT_CONNECTION_NUM 默认为 1

```
1.  const int config_btctrler_le_acl_total_nums = (CONFIG_BT_GATT_CONNECTION_NUM * 3);
```

Chapter 3 物理通道（PHY）修改支持

3.1 PHY 概述

本章主要介绍怎样设置不同 PHY（1M、2M、S2、S8）的应用方案，让用户能从最接近目标产品的 APP 开始开发

3.2 修改 1M PHY

3.2.1 概述

我司发布的 SDK 默认环境就是 1M PHY，直接使用即可，如果需要从别的 PHY 切换回 1M，见 3.2.2。

3.2.2 具体修改

1M 设置：在从机：“app_remote_24g_server.c” 设置为 CONN_SET_1M PHY

1. //选择物理层, 设置之前先配置 feature
2. `#define SELECT_PHY CONN_SET_1M_PHY//1M:CONN_SET_1M_PHY 2M:CONN_SE`
`T_2M_PHY CODED:CONN_SET_CODED_PHY`

3.3 修改 2M PHY

3.3.1 概述

修改 PHY 只需要连接的双方一方发起就可以进行 PHY 的切换，我司采用从机发起的方式进行修改。

3.3.2 具体修改

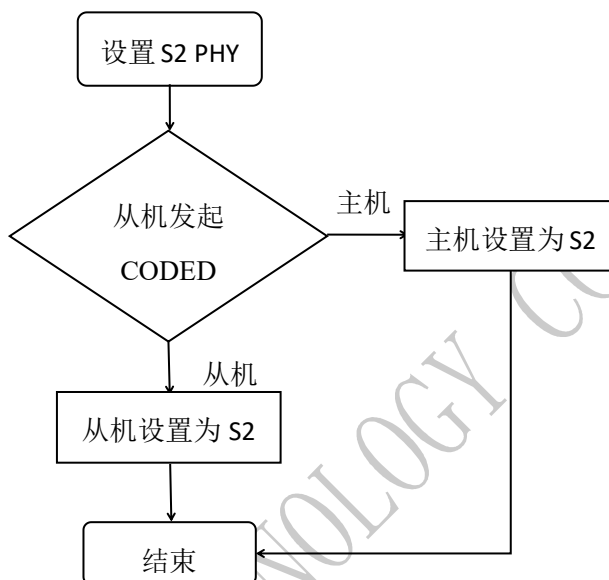
1)、2M 设置：在从机：“app_remote_24g_server.c” 设置为 CONN_SET_2M PHY

1. //选择物理层, 设置之前先配置 feature
2. #define SELECT_PHY CONN_SET_2M_PHY//1M:CONN_SET_1M_PHY
Y 2M:CONN_SET_2M_PHY CODED:CONN_SET_CODED_PHY

3.4 修改 S2 PHY

3.4.1 概述

首先 S2 PHY 和 S8 PHY 是包含于 CODED PHY 里面的，即设置步骤应如下所示。



3.4.2 具体修改

1)、CODED 设置：在从机：“app_remote_24g_server.c”设置为 CONN_SET_CODED PHY

```
1. //选择物理层, 设置之前先配置 feature
2. #define SELECT_PHY CONN_SET_CODED_PHY//1M:CONN_SET_1M_PHY 2M:CONN_
   SET_2M_PHY CODED:CONN_SET_CODED_PHY
```

2)、从机 CODED S2 设置：在：“app_remote_24g_server.c”设置为 CONN_SET_PHY_OPTIONS_S2

```
1. //选择 CODED 设置为 S2 or S8
2. #define SELECT_CODED_S2_OR_S8 CONN_SET_PHY_OPTIONS_S2//S2:CONN_SET_PHY_OPTIO
   NS_S2 S8:CONN_SET_PHY_OPTIONS_S8
```

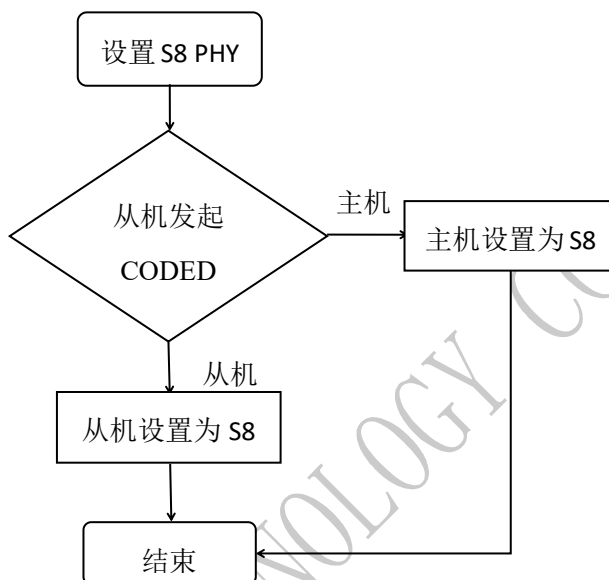
3)、主机 CODED S2 设置：在：“app_remote_24g_client.c”设置为 CONN_SET_PHY_OPTIONS_S2

```
1. //主机端选择 CODED 类型: S2 or S8
2. #define SELECT_CODED_S2_OR_S8 CONN_SET_PHY_OPTIONS_S2//S2:CONN_SET_PHY_OPTI
   ONS_S2 S8:CONN_SET_PHY_OPTIONS_S8
```

3.5 修改 S8 PHY

3.5.1 概述

设置步骤应如下所示，和设置 S2 步骤一样。



3.5.2 具体修改

1)、CODED 设置：在从机：“app_remote_24g_server.c” 设置为 CONN_SET_CODED_PHY

```
1. //选择物理层, 设置之前先配置 feature
2. #define SELECT_PHY CONN_SET_CODED_PHY//1M:CONN_SET_1M_PHY 2M:CONN_SET_2M_PHY CODED:CONN_SET_CODED_PHY
```

2)、从机 CODED S8 设置：在：“app_remote_24g_server.c” 设置为 CONN_SET_PHY_OPTIONS_S8

```
1. //选择 CODED 设置为 S2 or S8
2. #define SELECT_CODED_S2_OR_S8 CONN_SET_PHY_OPTIONS_S8//S2:CONN_SET_PHY_OPTIONS_S2 S8:CONN_SET_PHY_OPTIONS_S8
```

3)、主机 CODED S8 设置：在：“app_remote_24g_client.c” 设置为 CONN_SET_PHY_OPTIONS_S8

```
1. //主机端选择 CODED 类型: S2 or S8
2. #define SELECT_CODED_S2_OR_S8 CONN_SET_PHY_OPTIONS_S8//S2:CONN_SET_PHY_OPTIONS_S2 S8:CONN_SET_PHY_OPTIONS_S8
```

Chapter 4 2.4G 配对码修改

4.1 2.4G 配对码修改

4.1.1 概述

只需要主从机在打开 BLE 模式试下设置主从的**配对码**即可，设置主从的配对码一致，主从才会连接

4.1.2 主机配对码修改

打开板级配置文件 board_ac63xn_demo_cfg.h，确认是否打开 BLE 模式：

```
1.  //*****  
/   
2.  //                                蓝牙配置                                //  
3.  //*****  
/   
4.  #define TCFG_USER_TWS_ENABLE          0    //tws 功能使能  
5.  #define TCFG_USER_BLE_ENABLE         1    //BLE 功能使能  
6.  #define TCFG_USER_EDR_ENABLE         0    //EDR 功能使能
```

修改主机配对码：在“app_remote_24g_client.c”设置为 CFG_RF_24G_CODE_ID 为（0x23）

```
1.  //2.4G 模式: 0---ble, 非0---2.4G 配对码  
2.  /* #define CFG_RF_24G_CODE_ID          (0) //<=24bits */  
3.  #define CFG_RF_24G_CODE_ID          (0x23) //<=24bits
```

4.1.3 从机配对码修改

打开从机板级配置文件 board_ac63xn_demo_cfg.h，确认是否打开 BLE 模式：

```
1.  //*****  
/   
2.  //                                蓝牙配置                                //  
3.  //*****  
/   
4.  #define TCFG_USER_TWS_ENABLE          0    //tws 功能使能
```

- | | | |
|---------------------------------|---|------------|
| 5. #define TCFG_USER_BLE_ENABLE | 1 | //BLE 功能使能 |
| 6. #define TCFG_USER_EDR_ENABLE | 0 | //EDR 功能使能 |

修改主机配对码：在“app_remote_24g_server.c”设置为 CFG_RF_24G_CODE_ID 为（0x23）

- | |
|--|
| 1. //2.4G 模式: 0---ble, 非0---2.4G 配对码 |
| 2. /* #define CFG_RF_24G_CODE_ID (0) //<=24bits */ |
| 3. #define CFG_RF_24G_CODE_ID (0x23) //<=24bits |

Chapter 5 距离测试修改支持

5.1 距离测试

5.1.1 概述

用户可以使用我司使用的定时器模拟发数，通过打印进行距离测试。

5.1.2 距离测试

距离测试使能：“app_remote_24g_server.c” 设置为 1 打开

1. //是否打开 2.4G 持续发送数据
2. `#define REMOTE_24G_KEEP_SEND_EN 1` //just for 2.4gtest keep data

说明：测试程序循环发送一组数据，发送的数据为循环发送 26 个字母（每次发送一个）。用户可以把主机端通过 USB 直接连接 PC 端，然后输入法调为英文之后再打开一个文档，就可以直观看到发送效果