

Tina

Bluetooth API v1.0

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2015/12/28		初始版本
V1.1	2016/01/13		增加相关说明和设置蓝牙名称接口
V2.0	2016/09/19		增加相关接口



目 录

1. 概述.....	4
1.1. 编写目的.....	4
1.2. 适用范围.....	4
1.3. 相关人员.....	4
2. Bluetooth 相关说明.....	5
2.1. sdk 代码目录.....	5
2.2. 编译配置.....	5
2.3. APP 编写说明.....	5
2.3.1. 导入接口文件.....	5
2.3.2. 链接动态库.....	5
2.3.3. 示例代码.....	5
2.4. bt 打开和关闭.....	5
3. API 说明.....	6
3.1. 蓝牙打开与关闭.....	6
3.1.1. bt_on.....	6
3.1.2. bt_on_no_avrcp.....	6
3.1.3. bt_off.....	6
3.2. 蓝牙连接与断开.....	6
3.2.1. connect_auto.....	6
3.2.2. connect_dev_by_addr.....	6
3.2.3. int disconnect.....	7
3.3. 蓝牙状态获取.....	7
3.3.1. set_callback.....	7
3.3.2. tBtCallback.....	7
3.4. 设置本地蓝牙设备属性.....	7
3.4.1. set_bt_name.....	7
3.4.2. set_dev_discoverable.....	7
3.4.3. set_dev_connectable.....	7
3.5. 扫描设备.....	8
3.5.1. start_discovery.....	8
3.5.2. get_disc_results.....	8
3.6. 音乐播放与控制.....	8
3.6.1. avk_play.....	8
3.6.2. avk_pause.....	8
3.6.3. avk_previous.....	8
3.6.4. avk_next.....	8
3.6.5. avk_get_music_info.....	9
3.7. 电话接听与挂断.....	9
3.7.1. hs_pick_up.....	9
3.7.2. hs_hung_up.....	9
4. Demo.....	10
5. Declaration.....	12

1. 概述

1.1. 编写目的

介绍 Tina 蓝牙应用开发接口和 Demo 代码。

1.2. 适用范围

硬件平台：AW R16、R18 及 R40 等

系统版本：Tina v1.0 及以上版本

1.3. 相关人员

Tina app 开发人员。



2. Bluetooth 相关说明

2.1. sdk 代码目录

sdk 中 bluetooth 相关代码目录为 `package\allwinner\bluetooth`。包括蓝牙连接管理，事件监听和 demo 程序。

2.2. 编译配置

蓝牙 sdk 相关 menuconfig 配置如下：

tina 根目录下，输入 `make menuconfig`

选择：`Allwinner` --->

↳ <*> bluetooth..... Bluetooth APP(bsa 01062) --->

↳ < > bluetooth-demo..... BT app demo

注意：如果要参考蓝牙 app demo 程序，需要先选择

<*> bluetooth..... Bluetooth APP(bsa 01062) --->

再选择 Bluetooth-demo 包，表示 bt app demo 程序。

2.3. APP 编写说明

2.3.1. 导入接口文件

```
#include <bluetooth_socket.h>
```

2.3.2. 链接动态库

libbtapp.so 和 libbsa.so

2.3.3. 示例代码

蓝牙 APP demo 源码目录为：

`tina\package\allwinner\bluetooth\3rdparty\embedded\bsa_examples\linux\app_bluetooth\bt_test`。

2.4. bt 打开和关闭

1. 由于底层蓝牙服务启动时会生成相关配置文件，所以蓝牙运行目录必须可写。
2. 打开：调用 `bt_on` 函数打开蓝牙。如定制 bt mac 地址，需传入存有蓝牙 mac 地址文件的绝对路径。
3. 关闭：调用 `bt_off` 函数关闭蓝牙。

3. API 说明

Tina 平台蓝牙 API 包括底层蓝牙状态获取，音乐播放与控制及电话接听与挂断。

3.1. 蓝牙打开与关闭

3.1.1. bt_on

函数原型	int bt_on(char *bt_addr)
参数说明	bt_addr 蓝牙 MAC 地址存放文件(绝对路径)，内容为如下格式 00:11:22:33:44:55，使用场景为客户定制 MAC 地址。当 bt_addr 为 NULL 时，采用系统随机生成的 MAC 地址。
返回说明	int 0:成功；非 0:失败。
功能描述	打开本地蓝牙设备。

3.1.2. bt_on_no_avrcp

函数原型	int bt_on_no_avrcp(char *bt_addr)
参数说明	bt_addr 蓝牙 MAC 地址存放文件(绝对路径)，内容为如下格式 00:11:22:33:44:55，使用场景为客户定制 MAC 地址。当 bt_addr 为 NULL 时，采用系统随机生成的 MAC 地址。
返回说明	int 0:成功；非 0:失败。
功能描述	打开本地蓝牙设备(不启动 avrcp 服务),用于连接某些蓝牙音频测试仪器(没有 avrcp 连接)。

3.1.3. bt_off

函数原型	int bt_off()
参数说明	无
返回说明	int 0:成功；非 0:失败。
功能描述	关闭本地蓝牙设备。

3.2. 蓝牙连接与断开

3.2.1. connect_auto

函数原型	int connect_auto()
参数说明	无
返回说明	int 0:成功；非 0:失败。
功能描述	自动重连上次已连接的蓝牙设备，用于音箱断开连接或者关闭蓝牙后，自动重连上次已连接设备。

3.2.2. connect_dev_by_addr

函数原型	int connect_dev_by_addr(BT_ADDR bt_addr)
参数说明	bt_addr 蓝牙设备地址。蓝牙设备地址结构见 BT_ADDR。
返回说明	int 0:成功；非 0:失败。
功能描述	连接指定 mac 地址的蓝牙设备。

3.2.3. int disconnect

函数原型	int disconnect()
参数说明	无
返回说明	int 0:成功; 非 0:失败。
功能描述	断开与已连接设备的连接。

3.3. 蓝牙状态获取

3.3.1. set_callback

函数原型	void set_callback(tBtCallback *pCb)
参数说明	tBtCallback 蓝牙事件回调接口, 具体见 3.3.2
返回说明	无
功能描述	APP 注册回调函数, 底层蓝牙将状态信息通过 event 传递给 APP。此函数在 bt_on()之前调用, 捕获蓝牙从打开之后的所有事件。

3.3.2. tBtCallback

函数原型	typedef void (tBtCallback)(BT_EVENT event, void *reply, int *len);
参数说明	BT_EVENT 底层蓝牙传递给 APP 的 event。 BT_AVK_CONNECTED_EVT, => 媒体音频连接建立 BT_AVK_DISCONNECTED_EVT, => 媒体音频连接断开 BT_AVK_START_EVT, => 音乐开始播放 BT_AVK_STOP_EVT, => 音乐停止播放 BT_HS_CONNECTED_EVT = 0xf0, => 手机音频连接建立 BT_HS_DISCONNECTED_EVT, => 手机音频连接断开 BT_HS_RING_EVT, => 外部电话到来 BT_HS_OK_EVT, => 蓝牙耳机端操作 OK BT_DISCOVER_COMPLETE, => 蓝牙扫描完成事件 BT_HS_ERROR_EVT, => 蓝牙耳机端操作出错
返回说明	无
功能描述	蓝牙事件回调函数

3.4. 设置本地蓝牙设备属性

3.4.1. set_bt_name

函数原型	int set_bt_name(const char *bt_name)
参数说明	bt_name 本机蓝牙名称, 可以动态修改本机蓝牙名称。长度小于 256 字节。
返回说明	int 0:成功; 非 0:失败。
功能描述	动态设置本机蓝牙名称, 如果 AP 不设置, 则采用系统默认蓝牙名称。

3.4.2. set_dev_discoverable

函数原型	int set_dev_discoverable(int enable)
参数说明	enable 1: 蓝牙设备可被发现; 0: 蓝牙设备不可被发现。
返回说明	int 0:成功; 非 0:失败。
功能描述	动态设置本地蓝牙是否可被发现, 即可被其它设备搜索得到。

3.4.3. set_dev_connectable

函数原型	int set_dev_connectable(int enable)
------	-------------------------------------

参数说明	enable 1: 蓝牙设备可被连接; 0: 蓝牙设备不可被连接。
返回说明	int 0:成功; 非 0:失败。
功能描述	动态设置本地蓝牙设备是否可被连接。

3.5. 扫描设备

扫描周围的蓝牙设备，获取扫描到的设备列表。

3.5.1. start_discovery

函数原型	int start_discovery(int time)
参数说明	int time 扫描时间，当前无效。
返回说明	int 0:成功; 非 0:失败。
功能描述	启动扫描周围蓝牙设备。

3.5.2. get_disc_results

函数原型	int get_disc_results(char *disc_results, int *len)
参数说明	char *disc_results: 存储扫描结果 buf int *len: 初始值为 disc_results buf 长度，函数返回后为扫描结果长度。
返回说明	int 0:成功; 非 0:失败。
功能描述	获取蓝牙扫描结果。

3.6. 音乐播放与控制

接收端对源端音乐播放进行控制。接收端一般指蓝牙音箱或者耳机，源端一般指手机或平板。

3.6.1. avk_play

函数原型	int avk_play()
参数说明	无
返回说明	int 0:成功; 非 0:失败。
功能描述	源端音乐播放暂停后，接收端重启播放。

3.6.2. avk_pause

函数原型	int avk_pause()
参数说明	无
返回说明	int 0:成功; 非 0:失败。
功能描述	接收端暂停音乐播放。

3.6.3. avk_previous

函数原型	int avk_previous()
参数说明	无
返回说明	int 0:成功; 非 0:失败。
功能描述	接收端控制播放上一首。

3.6.4. avk_next

函数原型	int avk_next()
参数说明	无
返回说明	int 0:成功; 非 0:失败。
功能描述	接收端控制播放下一首。

3.6.5. avk_get_music_info

函数原型	int avk_get_music_info(tBT_AVK_MUSIC_INFO *p_avk_music_info);
参数说明	p_avk_music_info 指向歌曲信息结构的指针，通过此结构返回歌曲信息。
返回说明	int 0:成功；非 0:失败。
功能描述	获取手机端当前正在播放的歌曲信息。

3.7. 电话接听与挂断

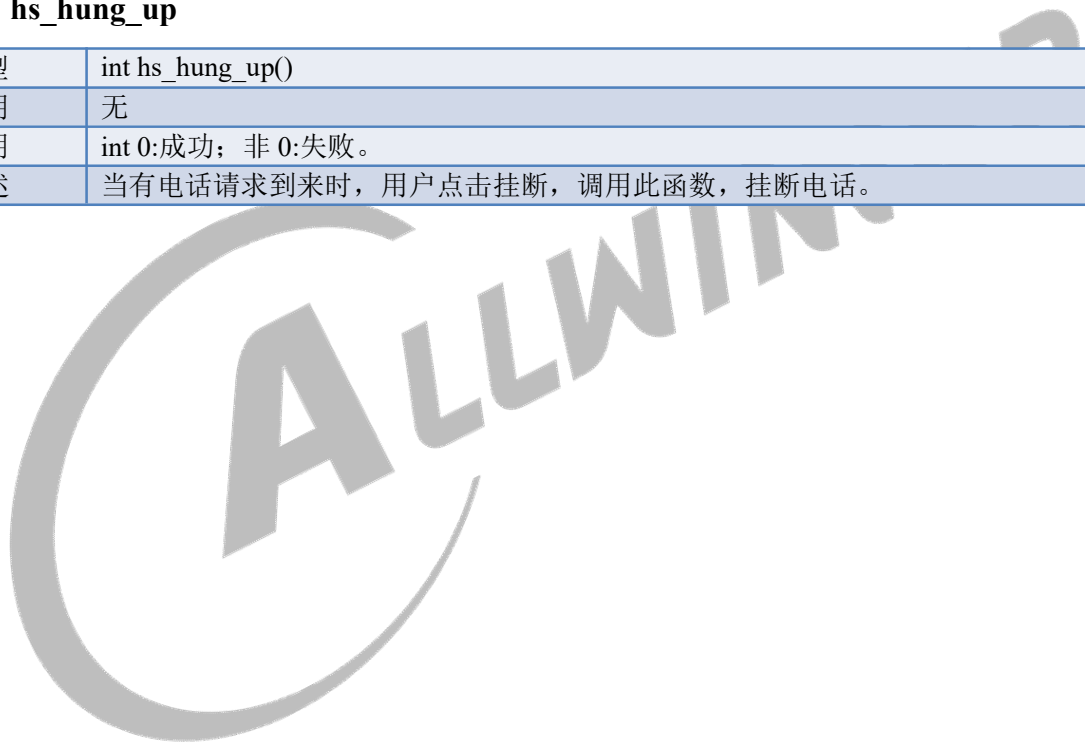
接收端控制电话的接听和挂断。

3.7.1. hs_pick_up

函数原型	int hs_pick_up()
参数说明	无
返回说明	int 0:成功；非 0:失败。
功能描述	当有电话请求到来时，用户点击接听，调用此函数，接听电话。

3.7.2. hs_hung_up

函数原型	int hs_hung_up()
参数说明	无
返回说明	int 0:成功；非 0:失败。
功能描述	当有电话请求到来时，用户点击挂断，调用此函数，挂断电话。



4. Demo

```

#include <bluetooth_socket.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>

#define DEBUG_MUSIC_INFO 0

static int last_status = 0;
static int status = 0;
static int playing = 0;
c_bt c;

void bt_event_f(BT_EVENT event, void *reply, int *len)
{
    switch(event)
    {
        case BT_AVK_CONNECTED_EVT:
        {
            printf("Media audio connected!\n");
            status = 1;
            break;
        }

        case BT_AVK_DISCONNECTED_EVT:
        {
            printf("Media audio disconnected!\n");
            printf("link down reason %d\n", *(int *)reply);
            status = 0;
            break;
        }

        case BT_AVK_START_EVT:
        {
            printf("Media start playing!\n");
            playing = 1;
            break;
        }

        case BT_AVK_STOP_EVT:
        {
            printf("Media stop playing!\n");
            playing = 0;
            break;
        }

        default:
            break;
    }
}

int main(int argc, char *args[]){

```

```

int times = 0;
int save_fd = -1, fd = -1;
int i = 0;
tBT_AVK_MUSIC_INFO music_info;

c.set_callback(bt_event_f);

printf("bt off before on\n");
c.bt_off();

last_status = 0;
status = 0;
if(argc >= 2){
    c.bt_on(args[1]);
} else {
    c.bt_on(NULL);
}

c.set_bt_name("aw bt test001");

while(1){
    usleep(2000*1000);

    /* connected */
    if ((last_status == 0) && (status == 1)){
        c.set_dev_discoverable(0);
        c.set_dev_connectable(0);
        last_status = 1;
    }

    /* disconnected */
    if ((last_status == 1) && (status == 0)){
        c.set_dev_discoverable(1);
        c.set_dev_connectable(1);
        last_status = 0;
    }

#ifdef DEBUG_MUSIC_INFO == 1
    if(playing == 1){
        c.avk_get_music_info(&music_info);
        printf("Title: %s\n", music_info.title);
        printf("Artist: %s\n", music_info.artist);
        printf("Album: %s\n", music_info.album);
        //printf("Time: %s\n", music_info.playing_time);
    }
#endif
}
}
}

```

5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgment to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

