

lcd 点阵屏-简易架构

珠海市杰理科技股份有限公司

Zhuhai Jieli Technologyco.,LTD

版权所有，未经许可，禁止外传

修改记录

版本	更新日期	描述
V1.0	2021/03/15	撰写调试文档



目录

1. 引 言.....	4
1.1. 编写目的.....	4
1.2. 参考资料.....	4
1.3. 术语和缩写词.....	4
2. 板卡配置.....	5
2.1. 使用该架构需要使能点阵屏板卡.....	5
2.2. 开启 ui 风格和驱动.....	5
3. 默认消息机制.....	6
3.1. 往 ui 发送消息.....	6
3.2. 事件的接收.....	6
3.3. 默认的按键事件.....	7
4. 注意事项.....	7

1. 引言

1.1. 编写目的

该文档为基于 AC695N 平台开发应用的人员提供相应的简易 ui 架构说明文档。也可以为测试应用的测试人员提供参考。

1.2. 参考资料

[1]

1.3. 术语和缩写词

缩写和术语	解 释

2. 板卡配置

2.1. 使用该架构需要使能点阵屏板卡

```
1 #ifndef BOARD_CONFIG_H
2 #define BOARD_CONFIG_H
3
4 /*
5  * 板级配置选择
6  */
7
8 // #define CONFIG_BOARD_AC695X_DEMO
9 // #define CONFIG_BOARD_AC6955F_HEADSET_MONO
10 // #define CONFIG_BOARD_AC6952E_LIGHTER
11 // #define CONFIG_BOARD_AC695X_CHARGING_BIN
12 // #define CONFIG_BOARD_AC695X_BTEMITTER
13 // #define CONFIG_BOARD_AC695X_TWS_BOX
14 // #define CONFIG_BOARD_AC695X_TWS
15 // #define CONFIG_BOARD_AC695X_MULTIMEDIA_CHARGING_BIN
16 // #define CONFIG_BOARD_AC695X_SOUND CARD
17 // #define CONFIG_BOARD_AC6954A_DEMO
18 // #define CONFIG_BOARD_AC695X_SMARTBOX
19 #define CONFIG_BOARD_AC695X_LCD
20
21 // #define CONFIG_BOARD_AC6951G
22 // #define CONFIG_BOARD_AC6083A
23 // #define CONFIG_BOARD_AC6083A_IAP
24
25
26 #define DAC_OUTPUT_MONO_L 0 //左声道
27 #define DAC_OUTPUT_MONO_R 1 //右声道
28 #define DAC_OUTPUT_LR 2 //立体声
29 #define DAC_OUTPUT_MONO_LR_DIFF 3 //单声道差分输出
30
31 #define DAC_OUTPUT_DUAL_LR_DIFF 6 //双声道差分
32 #define DAC_OUTPUT_FRONT_LR_REAR_L 7 //三声道单端输出 前L+前R+后L (不可设置vcmo公共端)
33 #define DAC_OUTPUT_FRONT_LR_REAR_R 8 //三声道单端输出 前L+前R+后R (可设置vcmo公共端)
34 #define DAC_OUTPUT_FRONT_LR_REAR_LR 9 //四声道单端输出(不可设置vcmo公共端)
35
36 #include "board_ac6954a_demo/board_ac6954a_demo_cfg.h" //SPDIF-HDMI-IIS测试板
37 #include "board_ac695x_demo/board_ac695x_demo_cfg.h"
38 #include "board_ac6952e_lighter/board_ac6952e_lighter_cfg.h" //蓝牙点烟器
39 #include "board_ac6955f_headset_mono/board_ac6955f_headset_mono_cfg.h" //蓝牙带卡耳机
```

2.2. 开启 ui 风格和驱动

1、#define CONFIG_UI_STYLE STYLE_UI_SIMPLE //这是 309 sdk ui 风格 需要再开启
TCFG_SIMPLE_LCD_ENABLE
选择 STYLE_UI_SIMPLE 风格

2、#define TCFG_SIMPLE_LCD_ENABLE ENABLE_THIS_MOUDLE
开启去 ui 框架的驱动开关，开启请关闭了 TCFG_SPI_LCD_ENABLE

3、开启屏幕驱动

版权所有，侵权必究

```
#define TCFG_LCD_OLED_ENABLE
```

```
ENABLE_THIS_MOUDLE
```

最后配置如下图:

```
9 //***** UI 配置 *****//
10 //***** UI 配置 *****//
11 //***** UI 配置 *****//
12 #define TCFG_UI_ENABLE ENABLE_THIS_MOUDLE //UI总开关
13 // #define CONFIG_UI_STYLE STYLE_JL_SOUNDBOX
14 #define CONFIG_UI_STYLE STYLE_UI_SIMPLE //这是309 sdk ui 风格 需要再开启TCFG_SIMPLE_LCD_ENABLE
15 // #define TCFG_UI_LED7_ENABLE ENABLE_THIS_MOUDLE //UI使用LED7显示
16 // #define TCFG_UI_LCD_SEG3X9_ENABLE ENABLE_THIS_MOUDLE //UI使用LCD段码屏显示
17 // #define TCFG_LCD_ST7735S_ENABLE ENABLE_THIS_MOUDLE
18 // #define TCFG_LCD_ST7789VW_ENABLE ENABLE_THIS_MOUDLE
19 #define TCFG_LRC_LYRICS_ENABLE ENABLE_THIS_MOUDLE //歌词显示
20 #define TCFG_LCD_OLED_ENABLE ENABLE_THIS_MOUDLE
21 // #define TCFG_SPI_LCD_ENABLE DIENABLE_THIS_MOUDLE //spi lcd开关
22 #define TCFG_SIMPLE_LCD_ENABLE ENABLE_THIS_MOUDLE //去ui 框架的驱动开关 开启请关闭了TCFG_SPI_LCD_ENABLE
23 #define TCFG_TFT_LCD_DEV_SPI_HW_NUM 1// 1: SPI1 2: SPI2 配置lcd选择的spi口
24 #define LCD_SPI_INTERRUPT_ENABLE 0//默认开启 需要主动关闭
25 #define ALL_KEY_EVENT_CLICK_ONLY 1 //是否全部按键只响应单击事件
26
```

3. 默认消息机制

3.1. 往 ui 发送消息

```
int ui_simple_key_msg_post(int key, int value)
```

函数功能: 往 ui 线程发送消息

Key: 按键事件

Value: 按键事件的默认值。可以为 0, 由用户自定义意思

3.2. 事件的接收

Demo 如下图

```
void demo_simple_ui_mode()
{
    int msg[16];
    DEMO_MenuInit();
    while (1) {
        app_task_get_msg(msg, ARRAY_SIZE(msg), 1);
        switch (msg[0]) {
```

```
case APP_MSG_SYS_EVENT:
    printf("ui %s   key = %d value = %x\n", __func__, msg[1], msg[2]);
    switch (msg[1]) {
    case KEY_OK:
        UI_mainmenu(APP_FM_TASK);
        DEMO_UI_MenuList(&FM_Main);
        DEMO_UI_MenuList(&FM_Seek);
        /* UI_mainmenu(APP_RECORD_TASK); */
        ui_simple_key_msg_post(KEY_OK, 0);
        break;
    case KEY_MENU:
        break;
    case KEY_DOWN:
        break;
    case KEY_UP:
        break;
    default:
        break;
    }
    break;
}
}
```

3.3. 默认的按键事件

Demo 默认分配了如下按键事件，用户可以跟进自己的需要进行消息增加

```
case KEY_OK:
case KEY_MENU:
case KEY_DOWN:
case KEY_UP:
case KEY_MODE:
```

4. 注意事项

用户使用 ui 接口必须统一在 ui 线程进行使用，即应用代码通过发送事件到线程里面进行显示操作，不能直接在应用代码线程进行操作（app core），因为旧的 ui 架构是基于裸机的接口实现的，没有进行操作系统多线程处理。

版权所有，侵权必究