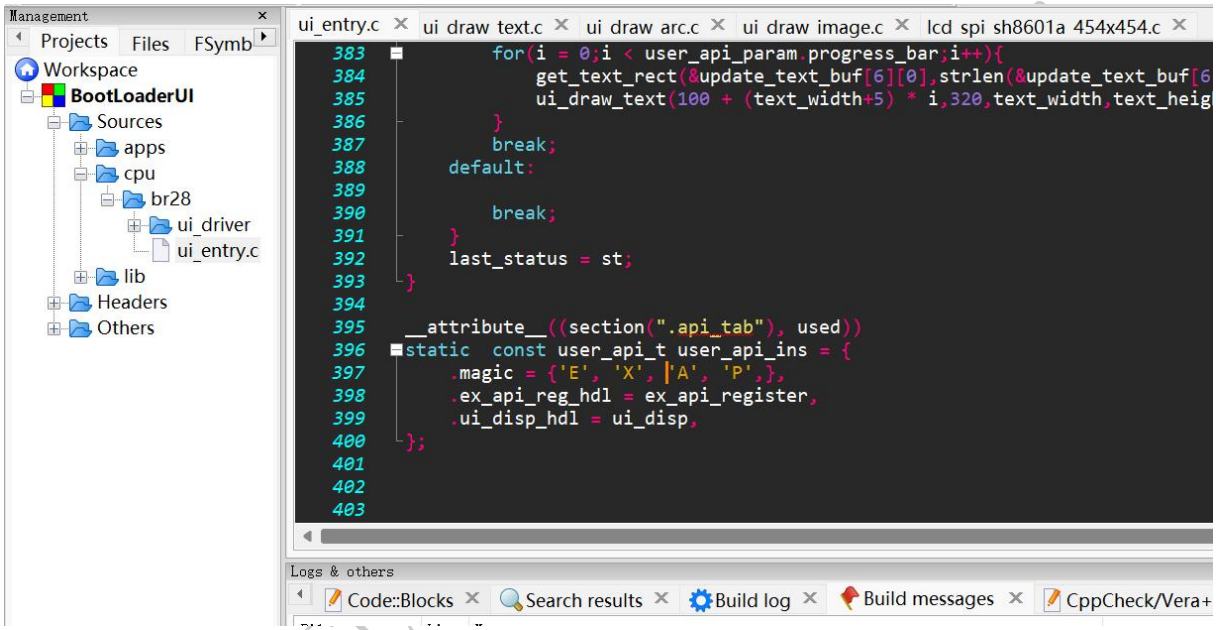
JL701N UI 升级工程添加和使用说明
（2023 年 3 月 2 日）

一、概述

添加了升级过程自定义 UI 功能。用户可以在本工程自行开发 UI。

二、工程使用描述

1、工程结构



2、程序调用逻辑

用户开发工程主要是利用 loader 提供的接口实现 ui 的初始化和扫描，并在 loader 有状态回调时，更新 UI。

2.1、loader 接口注册（在 ui_entry.c）

loader 会调用该函数为 user_ui 提供接口（接口在使用前需判断是否为 NULL），主要的接口为定时器相关的接口，如 sys_timer_add、sys_timer_del 等，用户可以使用该接口初始化对应的显示驱动。

ui_entry.c | ui draw text.c | ui draw arc.c | ui draw image.c | lcd spi sh8601a 454x454.c

```
233  static void ex_api_register(ex_api_t *api)
234  {
235      //extern u32 bss_begin;
236      //extern u32 bss_size;
237      printf("bss_begin :0x%x, bss_size : 0x%x\n", _bss_begin, _bss_size);
238      memset((void *)_bss_begin, 0, (int)_bss_size);
239
240      P3_IOV2_CON = 0x57;
241      printf("[%s] 0x%x, 0x%x, 0x%x, 0x%x, 0x%x, 0x%x, 0x%x, 0x%x, 0x%x\n", __FUNCTION__, api->printf, ap
242      if (api->irq_disable) {
243          my_irq_disable = api->irq_disable;
244      }
245
246      if (api->irq_enable) {
247          my_irq_enable = api->irq_enable;
248      }
249
250      if (api->mdelay) {
251          my_mdelay = api->mdelay;
252      }
253
254      if(api->request_irq_func) {
255          _request_irq = api->request_irq_func;
256      }
257
258      lcd_screen_init(&ui_cfg_data);
259
260  //  draw_ring(227, 227, 227, 0x07e0, 0);//参数 输入：x坐标 、y坐标、半径、颜色、百分百(0~100)
261  };
```

2.2、loader 状态回调

升级过程中 loader 会调用该接口，将升级的状态推给 user_ui 。用户可以在各个状态下刷新自己的 UI 显示。

```
static void ui_disp(UPDATA_STATUS st, u32 param)
{

    printf("[%s] st : %d, param : %d\n", __FUNCTION__, st, param);
    static u8 last_status = 0;
    int text_x,text_y,text_width,text_height;
    u8 str_buf[32] = {0};
    int i = 0;

    switch (st) {
    case UPDATE_START:
        printf("UPDATE_START.param:%d\n",param);
        if(param == 0){
            ui_draw_rect(0,0,454,454);
            draw_ring(227, 227, 227, 0x07e0, 0);
        }else if(param == 1){
            ui_draw_rect(0,0,454,454);
            get_text_rect(&update_text_buf[7][0],strlen(&update_text_buf[7][0]), &text_width, &text_height);
            text_x = 227-text_width/2;
            ui_draw_text(text_x,210,text_width,text_height,0xffff,&update_text_buf[7][0],strlen(&update_text_buf[7][0]));
```

状态说明：

```
ui entry.c × ui draw text.c × ui draw arc.c × ui draw image.c × lcd spi sh8601a 454x454.c × update_status.h ×
33        UPDATE_ERR_TONE_UPDATE_ERR,
34
35        UPDATE_ERR_RESERVED_CONFIG_UPDATE_ERR = 24,
36        UPDATE_ERR_PRODUCT_ID_NOT_MATCH,
37        UPDATE_ERR_CONN_ERR,
38
39
40
41    typedef enum {
42        UPDATE_START = 0,                       //升级开始  param 0:升级开始 1:进入升级 2:进入强制升级
43        UPDATE_STOP,                            //升级结束  param update error code
44        UPDATE_PROCESS,                         //升级中    param 升级进度percent = param/100
45        EX_API_UPDATE_TIPS_WAIT_CONN,           //等待蓝牙连接 param struct _user_api_param_t
46        EX_API_UPDATE_TIPS_WAIT_UPDATE,         //等待升级  param struct _user_api_param_t
47        EX_API_UPDATE_TIPS_WAIT_START_UPDATE,   //等待APP手动触发升级 param struct _user_api_param_t
48    } UPDATA_STATUS;
49
50
```

2.3、lcd 屏幕驱动

lcd_screen_init(&ui_cfg_data);要传入屏幕 io 和 spi 相关的配置，具体的配置的结构体如下图所示，主要是传入 spi 的硬件选择和屏幕相关 io，如 reset、背光等。

```
ui_entry.c × ui draw text.c × ui draw arc.c × ui draw image.c × lcd spi sh8601a 454x454.c × update_status.h ×
138
139
140   void lcd_screen_init(void *arg)
141   {
142       //屏初始化
143       struct lcd_info info = {0};
144       __this->lcd = lcd_get_hdl();
145       ASSERT(__this->lcd);
146
147       if (__this->lcd->power_ctrl) {
148           /* printf("0x%x, 0x%x, 0x%x\n", __this, __this->lcd, __this->lcd->power_ctrl); */
149           __this->lcd->power_ctrl(true);
150       }
151       if (__this->lcd->init) {
152           __this->lcd->init(arg);
153       }
154
155
156       if (__this->lcd->clear_screen) {
157           __this->lcd->clear_screen(0x000000);
158       }
159
160       if (__this->lcd->backlight_ctrl) {
161           __this->lcd->backlight_ctrl(100);
162       }
163
164       if (__this->lcd->get_screen_info) {
165           __this->lcd->get_screen_info(&info);
```

屏幕驱动和 spi 硬件的选择在 app_config.h,可以使能对应的屏幕驱动和 spi 相关。屏幕驱动实现和正常 sdk 实现基本一致，但是要注意不能使用和操作系统有关接口，例如 os_time_dly 这类型的接口，需要使用软件延时的方式实现。

2.4、设备调试 debug 打印

设备的打印接口是依赖 ota.bin 的传递，ex_api_t *api 包括的打印的函数指针，但是我们为了节约代码空间，ota.bin 分为了带打印的版本和不带打印的版本，需要进行驱动调试的时候，请把 ota 打 bebug 的文件替换为 ota.bin，即可进行打印调试。关于打印的 io 与波特率的设定，用户可以在正

常 sdk 工程中 tools 文件夹的 isd_config.ini 文件进行配置，如图。

```
#     0 RD_OUTPUT,       1 cmd       1 addr
#     1 RD_I/O,          1 cmd       x addr
#     2 RD_I/O_CONTINUE] no_send_cmd x add
#port:
#     0  优先选A端口  CS:PD3  CLK:PD0  D0:PD1  D1:PD2  D2:PB7  D3:PD5
#     1  优先选B端口  CS:PA13 CLK:PD0  D0:PD1  D1:PA14 D2:PA15 D3:PD5
SPI=2_3_0_0;    #width_clk_mode_port;
#OSC=btosc;
#OSC_FREQ=12MHz; #[24MHz 12MHz]
#SYS_CLK=24MHz; #[48MHz 24MHz]
UTTX=PA12;//uboot串口tx
UTBD=1000000;//uboot串口波特率
#UTRX=PB01;串口升级[PB00 PB05 PA05]
RESET=PB01_08_0;    //port口_长按时间_有效电平（长按时间有00、01、02、04、08五个值可选，单位为秒，当长按时间为00时，则关闭长按复位功能。）
UPDATE_JUMP=0;      //是否支持升级过程中维持IO电平，用于一些使用IO口维持芯片供电的方案

# 外部FLASH 硬件连接配置
EX_FLASH=PB07_2A_PB11;  //CS_pin / spi (1/2) /port(A/B) / power_io

#0:disable
```

为了方便用户的使用，本工程也封装了一个 debug 的宏在 app_congig.h, LCD_PRINTF。用户可以使用该接口进行打印调试。 例如：LCD_PRINTF("%s %d\n",__func__,__LINE__);

2.5、UI 参考接口

LoaderUI 只能绘制基础图形，或做简单组合

如**组合图形较多、刷新速度较快**可以参考 draw_ring 的方式组合绘制

(1)绘制填充矩形

void ui_draw_rect(int x, int y,int width,int height,int color)

(2) 绘制圆环进度条

void ui_draw_ring(int center_x, int center_y, int radius, u16 color, u16 bg_color,int percent)

注：圆环进度条会清空背景，需要最先绘制作为底图

(3) 绘制文本

void ui_draw_text(int x, int y,int width,int height,int color, u8 *str,u8 str_len)

需用 OTA 文本生产工具转成 L1 格式存放再 text_matrix.h 中

(4) 绘制图片

void ui_draw_image(int x, int y,int width,int height,u8 *image_buf,u16 image_buf_len)

图片需转为 rgb565 格式存放于 buffer 中，尺寸不宜过大（一般不超过 100*100）

2.6、常见问题

1. ram 不够

最常见的问题是更换了屏幕驱动后出现了 ram 不够的情况，这个需要裁剪一下显存

2. 显示不出画面

1）确认 loader 工程是否匹配（如不匹配，应见不到相关打印）

2）确认屏驱、背光等是否按实际配置

三、下载到 FLASH

工程编译之后会生成 user_api.bin 文件，将此文件拷贝到 sdk 工程中的 tools 文件在，然后在 download.bat 做如下修改(isd_download.exe 命令的最后添加 -ex_api_bin user_api.bin)：

```
.bin -res ui_upgrade p11_code.bin config.dat tone.cfg  eq_cfg_hw.bin -uboot_compress -key %CHIPKEY% -ex_flash res.bin -ex_api_bin us
in -res ui_upgrade p11_code.bin config.dat tone.cfg  eq_cfg_hw.bin -uboot_compress -ex_api_bin user_api.bin -format vm
```