

正点原子 littleVGL 开发指南

lv_imgbtn 图片按钮

开发指南

正点原子
广州市星翼电子科技有限公司

修订历史

版本	日期	原因
V1.00	2020/05/01	第一次发布

lv_imgbtn 图片按钮

1. 介绍

lv_imgbtn 图片按钮控件跟 lv_btn 按钮控件是非常相似的,它只不过是在 lv_btn 按钮控件的每一个状态基础上增加了图片显示的功能,之前我们已经学习过了 lv_btn 按钮章节和 lv_img 图片章节,现在再来学习它们的混合体 lv_imgbtn 图片按钮控件就会简单很多.

我们知道按钮是具有五种状态的,而在 lv_imgbtn 图片按钮中也是同样具有五种状态的,我们可以通过 lv_imgbtn_set_src(imgbtn, LV_BTN_STATE_..., &img_src)接口来给每一种状态设置一个对应的图片,不过这里有一点需要注意的是,这里的 img_src 图片源不支持图标字体和文本的方式,当采用这种方式设置图片之后,图片按钮的大小是自动适配图片大小的,用 lv_obj_set_size 接口来修改图片按钮的大小是无效的,另外 littleVGL 中还提供了另一种给图片按钮设置图片的接口,它是通过 lv_conf.h 中的 LV_IMGBTN_TILED 宏来使能的,当设为 1 使能之后,设置图片源的接口就变成了如下所示:

```
lv_imgbtn_set_src(imgbtn, LV_BTN_STATE_..., &src_left,&src_mid,&src_right)
```

从上面可以看出,接口名称是保持不变的,主要是形参发生了变化,之前是一个状态只能设置一个对应的图片,而现在是一个状态能同时设置三个图片,分别位于左边显示,中间显示,右边显示,在这种情况下,lv_imgbtn 图片按钮的宽度是可以设置的,高度是自动适配的,对于 src_left,src_mid,src_right 这三个图片源是可选的,对于不需要的传入 NULL 值就可以了,当 lv_imgbtn 图片按钮中有多余的宽度空间时,它会用 src_mid 图片进行重复填充的,默认情况下, LV_IMGBTN_TILED 功能是没有被使能的.

当 lv_imgbtn 图片按钮使用 lv_imgbtn_set_toggle(imgbtn, true)接口使能 toggle 功能之后,用户的每一次 toggle 操作都会触发一个 LV_EVENT_VALUE_CHANGED 事件,当然了,你可以直接使用 lv_imgbtn_set_state(imgbtn, LV_BTN_STATE_...)接口来改变图片按钮的状态,但是这种方式是不会触发 LV_EVENT_VALUE_CHANGED 事件的.

2. lv_imgbtn 的 API 接口

2.1 主要数据类型

2.1.1 图片按钮样式数据类型

```
enum {  
    LV_IMGBTN_STYLE_REL,  
    LV_IMGBTN_STYLE_PR,  
    LV_IMGBTN_STYLE_TGL_REL,  
    LV_IMGBTN_STYLE_TGL_PR,  
    LV_IMGBTN_STYLE_INA,  
};  
typedef uint8_t lv_imgbtn_style_t;
```

这五种样式就是用来修饰其对应的五种状态,每一种样式中都只用到了 image 和 text 字段,含义如下:

image.color: 图片重绘色时的混合颜色

image.intense: 图片重绘色时的混合强度

image.opa: 图片的透明度

text.color: 修饰图片按钮对象中的文本颜色,text 字段中的其他子字段就不介绍了,很简单的

2.2 API 接口

2.2.1 创建对象

```
lv_obj_t * lv_imgbtn_create(lv_obj_t * par, const lv_obj_t * copy);
```

参数:

par: 父对象

copy: 拷贝的对象,如果无拷贝的话,传 NULL 值

返回值:

返回创建出来的对象,如果返回 NULL 的话,说明堆空间不够了

2.2.2 设置某状态下的图片

```
#if LV_IMGBTN_TILED == 0  
void lv_imgbtn_set_src(lv_obj_t * imgbtn, lv_btn_state_t state, const void * src);  
#else
```

```
void lv_imgbtn_set_src(lv_obj_t * imgbtn, lv_btn_state_t state, const void * src_left, const
void * src_mid, const void * src_right);
#endif
```

参数:

imgbtn: 图片按钮对象

state: 按钮的状态

src: 在 LV_IMGBTN_TILED 没有使能时的图片源

src_left: 在 LV_IMGBTN_TILED 使能时的左边位置图片源, 不需要的话, 传入 NULL 值

src_mid: 在 LV_IMGBTN_TILED 使能时的中间位置图片源, 不需要的话, 传入 NULL 值

src_right: 在 LV_IMGBTN_TILED 使能时的右边位置图片源, 不需要的话, 传入 NULL 值

2.2.3 是否使能 toggle 功能

```
static inline void lv_imgbtn_set_toggle(lv_obj_t * imgbtn, bool tgl);
```

参数:

imgbtn: 图片按钮对象

tgl: 是否使能 toggle 功能

这个 API 接口的使用方法和 lv_btn 章节中的 lv_btn_set_toggle 接口的使用方法是一样的

2.2.4 设置按钮的当前状态

```
static inline void lv_imgbtn_set_state(lv_obj_t * imgbtn, lv_btn_state_t state);
```

参数:

imgbtn: 图片按钮对象

state: 按钮的状态

这个 API 接口的使用方法和 lv_btn 章节中的 lv_btn_set_state 接口的使用方法是一样的

2.2.5 进行一次 toggle 切换操作

```
static inline void lv_imgbtn_toggle(lv_obj_t * imgbtn);
```

参数:

imgbtn: 图片按钮对象

这个 API 接口的使用方法和 lv_btn 章节中的 lv_btn_toggle 接口的使用方法是一样的

2.2.6 设置样式

```
void lv_imgbtn_set_style(lv_obj_t * imgbtn, lv_imgbtn_style_t type, const lv_style_t * style);
```

参数:

imgbtn: 图片按钮对象

type: 设置哪一部分的样式,目前就只有 LV_IMG_STYLE_MAIN 这一个可选值

style: 样式

2.2.7 备注

还有几个 get 获取类型的 API 接口我这里就不列举出来了,比较简单的

3. 例程设计

3.1 功能简介

创建一个自定义样式来修饰图片按钮按下时的效果,然后接着创建一个图片按钮对象,给其设置每个状态下的图片源,并给其添加"img btn"文本标题,然后使能它的 toggle 功能,最后给其设置事件回调函数,本例程中将会用到如下 2 个图片素材:

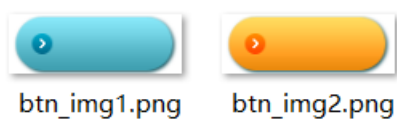


图 3.1.1 图片素材

这两张图片都是背景透明的,我们需要用 littleVGL 官方的图片在线转换工具将其进行转换,Color format 选择"true color with alpha"值,Output format 选择"C array"值即可.

3.2 硬件设计

本例程所用到的硬件有:

- 1) 液晶屏
- 2) 串口

3.3 软件设计

在 GUI_APP 目录下创建 lv_imgbtn_test.c 和 lv_imgbtn_test.h 两个文件,其中 lv_imgbtn_test.c 文件的内容如下:

```
#include "lv_imgbtn_test.h"
#include "lvgl.h"
#include <stdio.h>

//图片申明
LV_IMG_DECLARE(btn_img1);
LV_IMG_DECLARE(btn_img2);

lv_style_t pr_style;
```

```
//事件回调函数
void event_handler(lv_obj_t * obj,lv_event_t event)
{
    if(event==LV_EVENT_VALUE_CHANGED)
    {
        printf("LV_EVENT_VALUE_CHANGED\r\n");
    }
}

//例程入口
void lv_imgbtn_test_start()
{
    lv_obj_t *scr = lv_scr_act();//获取当前活跃的屏幕对象

    //1.创建按下时的样式
    lv_style_copy(&pr_style,&lv_style_plain);

    //图片重绘色时的混合色为黑色,这样看上去有按下的效果
    pr_style.image.color = LV_COLOR_BLACK;
    pr_style.image.intense = LV_OPA_50;//混合强度
    pr_style.text.color = LV_COLOR_MAKE(0xAA,0xAA,0xAA);//按下时的文本色

    //2.创建图片按钮对象
    lv_obj_t * imgbtn1 = lv_imgbtn_create(scr,NULL);

    //设置正常态松手时的图片
    lv_imgbtn_set_src(imgbtn1,LV_BTN_STATE_REL,&btn_img1);

    //设置正常态按下时的图片
    lv_imgbtn_set_src(imgbtn1,LV_BTN_STATE_PR,&btn_img1);

    //设置 toggle 切换态松手时的图片
    lv_imgbtn_set_src(imgbtn1,LV_BTN_STATE_TGL_REL,&btn_img2);

    //设置 toggle 切换态按下时的图片
    lv_imgbtn_set_src(imgbtn1,LV_BTN_STATE_TGL_PR,&btn_img2);

    //设置正常态按下时的样式
    lv_imgbtn_set_style(imgbtn1,LV_BTN_STATE_PR,&pr_style);

    //设置 toggle 切换态按下时的样式
    lv_imgbtn_set_style(imgbtn1,LV_BTN_STATE_TGL_PR,&pr_style);
    lv_imgbtn_set_toggle(imgbtn1,true);//使能 toggle 功能
```

```
lv_obj_align(imgbtn1,NULL,LV_ALIGN_CENTER,0,0);//与屏幕居中对齐
lv_obj_set_event_cb(imgbtn1,event_handler);//设置事件回调函数
lv_obj_t * label1 = lv_label_create(imgbtn1,NULL);//给图片按钮添加标题
lv_label_set_text(label1, "img btn");
}
```


3.4 下载验证

把代码下载进去之后,可以看到如下所示的初始界面效果:



图 3.4.1 初始界面效果

然后我们可以按住此图片按钮不放,就可以看到如下所示的按下效果:

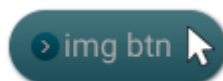


图 3.4.2 按下效果(忽略掉上面的鼠标)

松开手之后,会进入到 toggle 切换态下的松手状态,如下图所示:



图 3.4.3 toggle 切换态下的松手状态

4. 资料下载

正点原子公司名称 : 广州市星翼电子科技有限公司

LittleVGL 资料连接 : www.openedv.com/thread-309664-1-1.html

原子哥在线教学平台 : www.yuanzige.com

正点原子淘宝店铺 : <https://openedv.taobao.com>

正点原子官方网站 : www.alientek.com

正点原子 B 站视频 : <https://space.bilibili.com/394620890>

电话: 020-38271790 传真: 020-36773971

请下载原子哥 APP, 数千讲视频免费学习, 更快更流畅。

请关注正点原子公众号, 资料发布更新我们会通知。



扫码下载“原子哥”APP



扫码关注正点原子公众号