

正点原子 littleVGL 开发指南

lv_tabview 选项卡

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_tabview 选项卡

1. 介绍

lv_tabview 选项卡是一个非常实用的控件,在页面稍微复杂或者页面结构层次较深的 GUI 项目中,使用 lv_tabview 选项卡控件来设计界面,会很得心应手,使我们的界面设计得到简化,简单来说,它是由任意多个内容页面构成的,每一个内容页面都会有一个与之对应的页面选择按钮,然后 lv_tabview 选项卡中还有一个页面指示器,用来指示当前是哪一个页面处于选中显示状态,如下图所示:

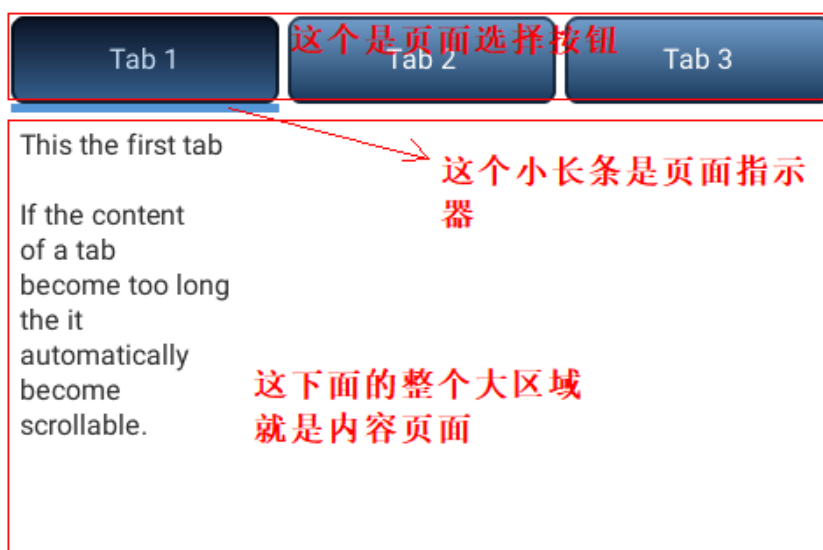


图 1.1 lv_tabview 选项卡组成

我们说了选项卡控件中的内容页面个数是不受限制的,它是通过 lv_tabview_add_tab(tabview, "Tab name")接口来添加一个内容页面的,然后你可以采用如下三种方式来切换内容页面:

- 1) 点击与之对应的页面选择按钮,但有一个前提条件,那就是你没有使用 lv_tabview_set_btns_hidden(tabview, true)这个接口把选择按钮给隐藏了
- 2) 用手指水平滑动页面,但有一个前提条件,那就是你没有使用 lv_tabview_set_sliding(tabview, false)这个接口把滑动功能给禁止了
- 3) 使用 lv_tabview_set_tab_act(tabview, id, LV_ANIM_ON/OFF)接口来选择显示指定 id 页面同时附带指定页面切换时是否具有动画效果

页面选择按钮的位置默认是处于顶部的,其实它的位置是可以通过 lv_tabview_set_btns_pos(tabview, LV_TABVIEW_BTNS_POS_TOP/BOTTOM/LEFT/RIGHT)接口来改变的,但是一般用的最多的就是 TOP 顶部和 BOTTOM 底部位置了。

最后来说一下它的事件,当你通过某种方式切换了页面,一旦页面切换完成之后,它就会给它的事件回调函数发送一个 LV_EVENT_VALUE_CHANGED 事件,注意,如果你是通过调用 lv_tabview_set_tab_act 接口来切换的页面,那么它是不会发送 LV_EVENT_VALUE_CHANGED 事件的。

2. lv_tabview 的 API 接口

2.1 主要数据类型

2.1.1 页面选择按钮栏位置数据类型

```
enum{
    LV_TABVIEW_BTNS_POS_TOP,        //顶部位置
    LV_TABVIEW_BTNS_POS_BOTTOM,    //底部位置
    LV_TABVIEW_BTNS_POS_LEFT,      //左边位置
    LV_TABVIEW_BTNS_POS_RIGHT      //右边位置
};
typedef uint8_t lv_tabview_btns_pos_t;
```

一般最常用的是 TOP 顶部和 BOTTOM 底部位置

2.1.2 选项卡样式数据类型

```
enum {
    LV_TABVIEW_STYLE_BG,           //选项卡的整个背景样式
    LV_TABVIEW_STYLE_INDIC,        //页面指示器的样式
    LV_TABVIEW_STYLE_BTN_BG,       //页面选择按钮栏的背景样式
    LV_TABVIEW_STYLE_BTN_REL,      //下面四种就是页面选择按钮的样式
    LV_TABVIEW_STYLE_BTN_PR,
    LV_TABVIEW_STYLE_BTN_TGL_REL,
    LV_TABVIEW_STYLE_BTN_TGL_PR,
};
typedef uint8_t lv_tabview_style_t;
```

虽然上面有 7 种样式,但是最下面的 4 种是用来修饰页面选择按钮的,至于具体的用法我们在 lv_btn 按钮章节中已经讲过了,是一模一样的,我们这里只介绍下面俩种.

LV_TABVIEW_STYLE_BG: 选项卡的整个背景样式,使用样式中的 body 字段,默认值为 lv_style_plain

LV_TABVIEW_STYLE_INDIC: 页面指示器的样式,使用样式中的 body 字段,它的高度由 body.padding.inner 字段来指定,默认值为 lv_style_plain_color

LV_TABVIEW_STYLE_BTN_BG: 页面选择按钮栏的背景样式,使用样式中的 body 字段,默认值为 lv_style_transp

2.2 API 接口

2.2.1 创建对象

```
lv_obj_t * lv_tabview_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 添加内容页面

```
lv_obj_t * lv_tabview_add_tab(lv_obj_t * tabview, const char * name);
```

参数:

tabview: 选项卡对象

name: 此内容页面的标题,会显示在对应的页面选择按钮上

返回值:

返回被添加的内容页面对象,其实此对象就是 lv_page 页面,所以拿到此对象之后,我们也可以利用 lv_page 页面专有的 API 接口来操作它

2.2.3 清空某内容页面中的所有子对象

```
void lv_tabview_clean(lv_obj_t * obj);
```

参数:

obj: 由 lv_tabview_add_tab 接口返回出来的内容页面对象

其实此 API 接口的实现非常简单,它就是利用到了 lv_page 页面控件专有的 API 接口来实现的,如下图所示:

```
193 void lv_tabview_clean(lv_obj_t * obj)
194 {
195     lv_obj_t * scr1 = lv_page_get_scr1(obj);
196     lv_obj_clean(scr1);
197 }
```

图 2.2.3.1 此接口的实现原理

2.2.4 设置哪一个内容页面处于可见状态

```
void lv_tabview_set_tab_act(lv_obj_t * tabview, uint16_t id, lv_anim_enable_t anim);
```

参数:

tabview: 选项卡对象

id: 内容页面的 id,第一个被添加的内容页面 id 为 0,后面的页面依次递增 1

anim: 在页面切换时,是否具有切换动画,有如下俩个可选值

LV_ANIM_OFF: 不开启动画

LV_ANIM_ON: 开启动画

2.2.5 是否使能手指滑页功能

```
void lv_tabview_set_sliding(lv_obj_t * tabview, bool en);
```

参数:

tabview: 选项卡对象

en: 是否使能手指滑页功能

如果不设置的话,默认是使能的

2.2.6 设置切换动画的时长

```
void lv_tabview_set_anim_time(lv_obj_t * tabview, uint16_t anim_time);
```

参数:

tabview: 选项卡对象

anim_time: 动画时长,单位 ms

2.2.7 设置样式

```
void lv_tabview_set_style(lv_obj_t * tabview, lv_tabview_style_t type, const lv_style_t * style);
```

参数:

tabview: 选项卡对象

type: 设置哪一部分的样式,有 7 个可选值,详情请看 2.1.2 选项卡样式数据类型

style: 样式

2.2.8 设置页面选择按钮栏的位置

```
void lv_tabview_set_btns_pos(lv_obj_t * tabview, lv_tabview_btns_pos_t btns_pos);
```

参数:

tabview: 选项卡对象

btns_pos: 页面选择按钮栏的位置,有如下 4 个可选值:

LV_TABVIEW_BTNS_POS_TOP: 顶部位置

LV_TABVIEW_BTNS_POS_BOTTOM: 底部位置

LV_TABVIEW_BTNS_POS_LEFT: 左边位置,基本用不到

LV_TABVIEW_BTNS_POS_RIGHT: 右边位置,基本用不到

2.2.9 是否隐藏页面选择按钮

```
void lv_tabview_set_btns_hidden(lv_obj_t * tabview, bool en);
```

参数:

tabview: 选项卡对象

en: 是否隐藏页面选择按钮

如果不设置的话,默认是不隐藏的,如果你设置成了隐藏的话,那么页面选择按钮之前占据的空间位置会被释放出来,给内容页面使用

2.2.10 获取内容页面的总个数

```
uint16_t lv_tabview_get_tab_count(const lv_obj_t * tabview);
```

参数:

tabview: 选项卡对象

返回值:

返回 tabview 选项卡中的内容页面总个数

2.2.11 获取当前哪一个内容页面处于可见状态

```
uint16_t lv_tabview_get_tab_act(const lv_obj_t * tabview);
```

参数:

tabview: 选项卡对象

返回值:

返回当前可见内容页面的 id 值

2.2.12 获取某个内容页面对象

```
lv_obj_t * lv_tabview_get_tab(const lv_obj_t * tabview, uint16_t id);
```

参数:

tabview: 选项卡对象

id: 内容页面的 id

返回值:

返回 id 值对应的内容页面对象

拿到此对象之后,我们就可以用 lv_page 页面控件中专有的 API 接口来操作它了

2.2.13 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 7 个自定义样式来修饰选项卡控件,然后创建一个选项卡控件,设置其大小,以及设置其事件回调函数,在事件回调函数中通过串口打印页面 id,接着给此选项卡添加三个内容页面,在内容页面 1 中,添加一个标签子对象和一个按钮子对象,在内容页面 2 和内容页面 3 中都只添加一个标签子对象,当按下 KEY0 按键时,调用 lv_tabview_set_tab_act 接口来切换内容页面,当按下 KEY1 按键时,来设置是否使能手指滑页功能,当按下 WKUP 按键时,来设置是否隐藏页面选择按钮栏。

3.2 硬件设计

本例程所用到的硬件有:

- 1) 液晶屏
- 2) KEY0, KEY1, WKUP 按键
- 3) 串口

3.3 软件设计

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

```
#include "lv_tabview_test.h"
#include "lvgl.h"
#include "key.h"
#include <stdio.h>

lv_style_t bg_style;
lv_style_t indic_style;
lv_style_t btn_bg_style;
lv_style_t btn_rel_style;
lv_style_t btn_pr_style;
lv_style_t btn_tgl_rel_style;
lv_style_t btn_tgl_pr_style;
lv_obj_t * tabview1;
bool is_sliding = true;
```

```
bool is_btns_hidden = false;

//事件回调函数
void event_handler(lv_obj_t * obj,lv_event_t event)
{
    if(obj==tabview1&&event==LV_EVENT_VALUE_CHANGED)
    {
        uint16_t page_id = lv_tabview_get_tab_act(tabview1);//获取当前可见的页面 id
        printf("Current page id:%d\r\n",page_id);//串口打印当前页面的 id
    }
}

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

    //1.创建样式
    //1.1 创建背景样式
    lv_style_copy(&bg_style,&lv_style_plain);
    bg_style.body.main_color = LV_COLOR_MAKE(49,49,49);//纯色背景
    bg_style.body.grad_color = bg_style.body.main_color;
    bg_style.body.border.color = LV_COLOR_MAKE(150,150,150);//边框颜色
    bg_style.body.border.width = 2;//边框宽度
    bg_style.text.color = LV_COLOR_WHITE;

    //1.2 创建页面指示器的样式
    lv_style_copy(&indic_style,&lv_style_plain_color);
    indic_style.body.main_color = LV_COLOR_MAKE(42,212,66);//指示器的颜色,绿色
    indic_style.body.grad_color = indic_style.body.main_color;
    indic_style.body.padding.inner = 3;//设置指示器的高度

    //1.3 创建页面选择按钮栏的背景样式
    //lv_style_transp_tight 样式中的 inner,left,top,right,bottom 等内间距值都为 0,这是为了
    //让页面选择按钮能够紧挨在一起
    lv_style_copy(&btn_bg_style,&lv_style_transp_tight);

    //1.4 创建按钮正常态下的松手样式
    lv_style_copy(&btn_rel_style,&lv_style_plain_color);
    btn_rel_style.body.main_color = LV_COLOR_MAKE(98,98,98);
    btn_rel_style.body.grad_color = btn_rel_style.body.main_color;
    btn_rel_style.body.border.color = LV_COLOR_MAKE(150,150,150);//边框颜色
    btn_rel_style.body.border.width = 1;
```



```
btn_rel_style.text.color = LV_COLOR_WHITE;//字体颜色

//1.5 创建按钮正常态下的按下样式
lv_style_copy(&btn_pr_style,&btn_rel_style);
btn_pr_style.body.main_color = LV_COLOR_GRAY;
btn_pr_style.body.grad_color = btn_pr_style.body.main_color;

//1.6 创建按钮切换态下的松手样式
lv_style_copy(&btn_tgl_rel_style,&btn_rel_style);
btn_tgl_rel_style.body.main_color = bg_style.body.main_color;//和主背景颜色一致
btn_tgl_rel_style.body.grad_color = btn_tgl_rel_style.body.main_color;

//1.7 创建按钮切换态下的按下样式
//保持和 btn_tgl_rel_style 一样就行了
lv_style_copy(&btn_tgl_pr_style,&btn_tgl_rel_style);

//2.创建选项卡
tabview1 = lv_tabview_create(scr,NULL);
//设置选项卡的大小,比屏幕小 16 像素
lv_obj_set_size(tabview1,lv_obj_get_width(scr)-20,lv_obj_get_height(scr)-20);
lv_obj_align(tabview1,NULL,LV_ALIGN_CENTER,0,0);//与屏幕居中对齐
lv_obj_set_event_cb(tabview1,event_handler);//设置事件回调函数
//设置页面选择按钮栏位于顶部
lv_tabview_set_btns_pos(tabview1,LV_TABVIEW_BTNS_POS_TOP);
lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_BG,&bg_style);//设置背景样式
//设置页面指示器的样式
lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_INDIC,&indic_style);
//设置页面选择按钮栏的背景样式
lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_BTN_BG,&btn_bg_style);
//设置按钮正常态下的松手样式
lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_BTN_REL,&btn_rel_style);
//设置按钮正常态下的按下样式
lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_BTN_PR,&btn_pr_style);

lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_BTN_TGL_REL,&btn_tgl_rel_style);
//设置按钮切换态下的松手样式

lv_tabview_set_style(tabview1,LV_TABVIEW_STYLE_BTN_TGL_PR,&btn_tgl_pr_style);
//设置按钮切换态下的按下样式

//2.1 添加 tab1 内容页面
lv_obj_t *tab1_page = lv_tabview_add_tab(tabview1,LV_SYMBOL_WIFI" Tab1");
//往内容页面 1 中添加标签子对象
lv_obj_t * tmp_obj = lv_label_create(tab1_page,NULL);
lv_label_set_text(tmp_obj,"\nThis is the tab1 page\nIf the content\nof a tab\nbecome too
```

```
long\nthe it \nautomatically\nbecome\nscrollable.");
//往内容页面 1 中添加标签子对象
tmp_obj = lv_btn_create(tab1_page,NULL);
lv_obj_set_pos(tmp_obj,50,300);

//2.2 添加 tab2 内容页面
lv_obj_t *tab2_page = lv_tabview_add_tab(tabview1,LV_SYMBOL_AUDIO" Tab2");
//往内容页面 2 中添加标签子对象
tmp_obj = lv_label_create(tab2_page,NULL);
lv_label_set_text(tmp_obj,"\nThis is the tab2 page\nZheng dian yuan zi");

//2.3 添加 tab3 内容页面
lv_obj_t *tab3_page = lv_tabview_add_tab(tabview1,LV_SYMBOL_BELL" Tab3");
//往内容页面 3 中添加标签子对象
tmp_obj = lv_label_create(tab3_page,NULL);
lv_label_set_text(tmp_obj,"\nThis is the tab3 page\nXiong jia yu");
}

//按键处理
void key_handler()
{
    static u16 page_id = 0;

    u8 key = KEY_Scan(0);

    if(key==KEY0_PRES)
    {
        //使用接口来切换内容页面
        page_id++;
        if(page_id==3)
            page_id = 0;
        lv_tabview_set_tab_act(tabview1,page_id,LV_ANIM_ON);//带有切换动画效果
    }else if(key==KEY1_PRES)
    {
        //是否使能手指滑页功能
        is_sliding = !is_sliding;
        lv_tabview_set_sliding(tabview1,is_sliding);
    }else if(key==WKUP_PRES)
    {
        //是否隐藏页面选择按钮栏
        is_btns_hidden = !is_btns_hidden;
        lv_tabview_set_btns_hidden(tabview1,is_btns_hidden);
    }
}
```

3.4 下载验证

把代码下载进去之后,可以看到如下所示的界面效果:

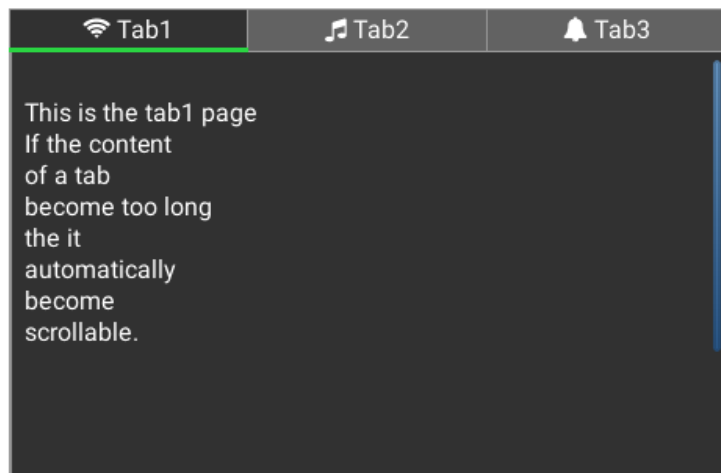


图 3.4.1 初始化界面效果

然后我们可以通过点击顶部的页面选择按钮,或者通过手指滑动页面,或者按 KEY0 按键来切换到 Tab2 页面,如下图所示:

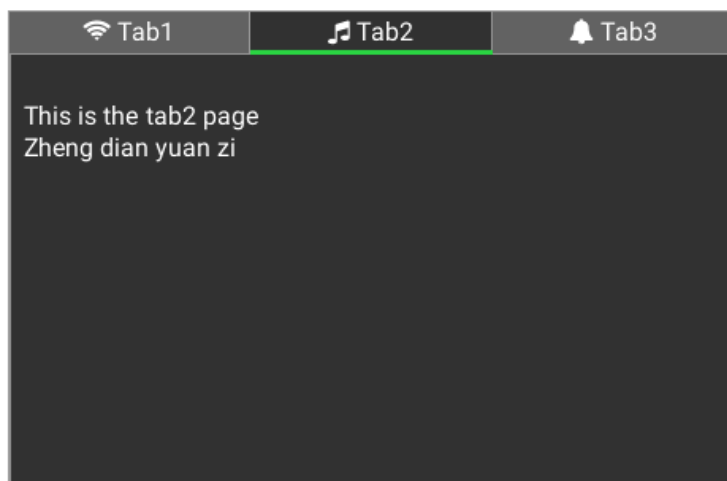


图 3.4.2 页面切换效果

然后我们还可以按下 WKUP 按键来隐藏顶部的页面选择按钮栏,如下图所示:

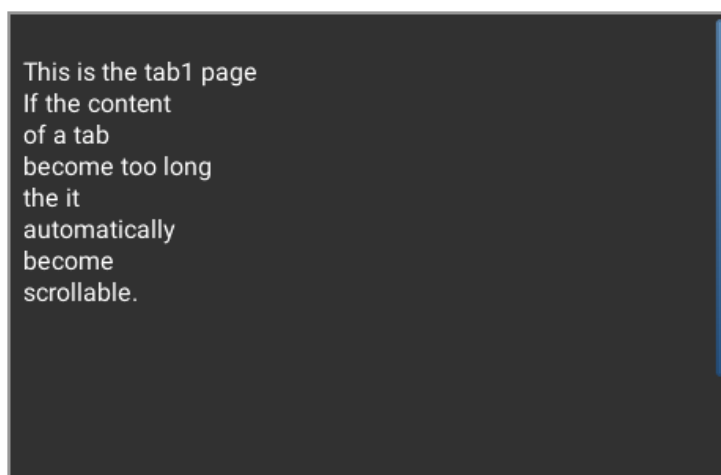


图 3.4.3 隐藏页面选择按钮栏后的效果

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