

正点原子 littleVGL 开发指南

lv_list 列表

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_list 列表

1. 介绍

lv_list 列表控件是由一个作为背景的 page 页面和一些作为列表项的按钮构成的,其中的每一个列表项按钮是由可选的左右俩部分构成,左侧是用来放图标的,右侧是用来放文本的,具体的构成示意图如下所示:



图 1.1 lv_list 列表控件的构成

当 lv_list 列表中的列表项足够多时,lv_list 列表就可以被滚动了,你可以通过 lv_list_add_btn(list, &icon_img, text)接口来给 lv_list 列表添加列表项按钮,icon_img 就是按钮内部左侧的图标,跟 lv_img 图片章节中的图片源用法是一样的,text 就是按钮内部右侧的文本,其中 icon_img 图标是可选的,当传入 NULL 值时,代表无图标,那么它占据的位置也会被释放出来留给文本使用,列表项按钮的宽度和 lv_list 列表控件的宽度保持一致,列表项按钮的高度是根据其内部的内容大小来自动确定的,具体值为内容的高度+padding.top+padding.bottom,按钮内部右侧的文本标签默认是具有 LV_LABEL_LONG_SCROLL_CIRC 长文本模式的,也就是说当文本内容足够长时,它是会进行循环滚动显示的,然后被添加的列表项按钮其实也是可以被删除的,直接使用 lv_obj_del(btn)接口来删除就可以了,其中的 btn 就是 lv_list_add_btn 接口返回出来的对象,如果你想把列表中的所有列表项都给清空的话,那么你可以使用 lv_list_clean(list)接口.

然后我们来说一下 lv_list 列表控件的手动导航,前提是列表控件中的列表项能够被滚动,否则手动导航 API 接口是无效的,使用 lv_list_up(list)接口可以将列表底部的一个不可见列表项按钮往上移动一步,使此列表项按钮处于可见状态,或者使用 lv_list_down(list)接口将列表顶部的一个不可见列表项按钮往下移动一步,同样使此列表项按钮处于可见状态,再或者使用 lv_list_focus(btn, LV_ANIM_ON/OFF)接口将任意一个不可见的列表项按钮处于可见状态.

2. lv_list 的 API 接口

2.1 主要数据类型

2.1.1 列表样式数据类型

```
enum {  
    LV_LIST_STYLE_BG,//这上面 4 种是用于修饰列表中的背景页面的  
    LV_LIST_STYLE_SCRL,  
    LV_LIST_STYLE_SB,  
    LV_LIST_STYLE_EDGE_FLASH,  
    LV_LIST_STYLE_BTN_REL,//这下面 5 种是用于修饰列表中的列表项按钮的  
    LV_LIST_STYLE_BTN_PR,  
    LV_LIST_STYLE_BTN_TGL_REL,  
    LV_LIST_STYLE_BTN_TGL_PR,  
    LV_LIST_STYLE_BTN_INA,  
};  
typedef uint8_t lv_list_style_t;
```

虽然有 9 种样式,但是最上面 4 种是用于修饰列表中的背景页面的,跟 lv_page 页面章节中的使用方法是一样的,最下面的 5 种是用于修饰列表项按钮的,跟 lv_btn 按钮章节中的使用方法是一样的。

2.2 API 接口

2.2.1 创建对象

```
lv_obj_t * lv_list_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 清空所有的列表项按钮

```
void lv_list_clean(lv_obj_t * list);
```

参数:

list: 列表对象

2.2.3 添加列表项按钮

```
lv_obj_t * lv_list_add_btn(lv_obj_t * list, const void * img_src, const char * txt);
```

参数:

list: 列表对象

img_src: 按钮内部左侧的图标,跟 lv_img 图片章节中的图片源用法是一样的

txt: 按钮内部右侧的文本

返回值:

把被添加的列表项按钮对象给返回出来,拿到此对象之后,我们可以用 lv_obj_set_event_cb 接口给此列表项按钮设置事件回调函数,以实现我们想要的逻辑功能,或者在后面还可以用 lv_obj_del 接口把此列表项按钮给删除掉

2.2.4 删除某位置上的列表项按钮

```
bool lv_list_remove(const lv_obj_t * list, uint16_t index);
```

参数:

list: 列表对象

index: 列表项按钮对应的索引位置,由上到下,从 0 开始

2.2.5 是否使能列表的单选模式

```
lv_list_set_single_mode(lv_obj_t * list, bool mode);
```

参数:

list: 列表对象

mode: 是否使能单选模式

默认情况下,单选模式是没有被使能的,假如使能之后,我们可以实现单选按钮组的功能,或者实现选中行高亮的功能,如下图所示,Open 列表项按钮就处于选中或者说是高亮状态,这完全取决于你样式修饰的效果.

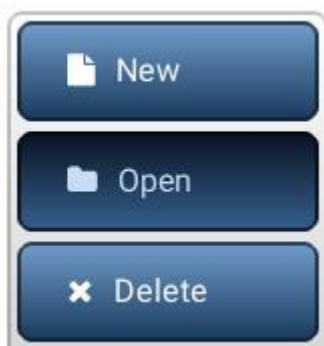


图 2.2.5.1 列表的单选模式效果

2.2.6 让某列表项按钮处于选中状态

```
void lv_list_set_btn_selected(lv_obj_t * list, lv_obj_t * btn);
```

参数:

list: 列表对象

btn: 列表项按钮对象

2.2.7 设置滚动条的模式

```
static inline void lv_list_set_sb_mode(lv_obj_t * list, lv_sb_mode_t mode);
```

参数:

list: 列表对象

mode: 滚动条的模式

此 API 接口的使用方法和 lv_page 页面章节中的 lv_page_set_sb_mode 接口的使用方法是一样的

2.2.8 是否使能边缘半圆弧动画效果

```
static inline void lv_list_set_edge_flash(lv_obj_t * list, bool en);
```

参数:

list: 列表对象

en: 是否使能

此 API 接口的使用方法和 lv_page 页面章节中的 lv_page_set_edge_flash 接口的使用方法是一样的

2.2.9 设置动画时长

```
static inline void lv_list_set_anim_time(lv_obj_t * list, uint16_t anim_time);
```

参数:

list: 列表对象

anim_time: 滚动或者 focus 动画的时长,单位 ms

2.2.10 设置样式

```
void lv_list_set_style(lv_obj_t * list, lv_list_style_t type, const lv_style_t * style);
```

参数:

list: 列表对象

type: 设置哪一部分的样式,详情请看 2.1.1 小节

style: 样式

2.2.11 获取列表项按钮内的文本内容

```
const char * lv_list_get_btn_text(const lv_obj_t * btn);
```

参数:

btn: 列表项按钮对象

返回值:

返回此列表项按钮内的文本内容

2.2.12 获取列表项按钮内的标签对象

```
lv_obj_t * lv_list_get_btn_label(const lv_obj_t * btn);
```

参数:

btn: 列表项按钮对象

返回值:

返回此列表项按钮内的标签对象,拿到此标签对象之后,你就可以用 lv_label 标签控件专有的 API 接口来操作它了

2.2.13 获取列表项按钮内的图片对象

```
lv_obj_t * lv_list_get_btn_img(const lv_obj_t * btn);
```

参数:

btn: 列表项按钮对象

返回值:

返回此列表项按钮内的图片对象,拿到此图片对象之后,你就可以用 lv_img 图片控件专有的 API 接口来操作它了

2.2.14 获取某列表项按钮所在的位置索引

```
int32_t lv_list_get_btn_index(const lv_obj_t * list, const lv_obj_t * btn);
```

参数:

list: 列表对象

btn: 列表项按钮对象

返回值:

返回 btn 列表项按钮对象在列表中的位置索引值

2.2.15 将列表上移一步

```
void lv_list_up(const lv_obj_t * list);
```

参数:

list: 列表对象

此 API 接口的作用是将列表底部的一个不可见列表项按钮往上移动一步,使此列表项按钮处于可见状态

2.2.16 将列表下移一步

```
void lv_list_down(const lv_obj_t * list);
```

参数:

list: 列表对象

此 API 接口的作用是将列表顶部的一个不可见列表项按钮往下移动一步,使此列表项按钮处于可见状态

2.2.17 将某列表项按钮处于可见状态

```
void lv_list_focus(const lv_obj_t * btn, lv_anim_enable_t anim);
```

参数:

list: 列表对象

anim: 是否开启动画效果,有如下两个可选值

LV_ANIM_OFF: 不开启动画效果

LV_ANIM_ON: 开启动画效果

此 API 接口的作用是可以将任意一个不可见的列表项按钮处于可见状态

2.2.18 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列表控件的背景,然后创建一个列表控件,使其能其单选模式,接着设置它的滚动条模式为自动模式,最后往它里面添加 5 个列表项按钮,并为每一个列表项按钮对象设置了事件回调函数,当按下 KEY0 按键时,会将此列表控件往上移动一步,当按下 KEY1 按键时,会将此列表控件往下移动一步,当按下 WKUP 按键时,会将最后一个列表项按钮处于可见状态.

3.2 硬件设计

本例程所用到的硬件有:

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

3.3 软件设计

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

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

lv_style_t bg_style;
lv_obj_t * list1;
lv_obj_t * last_item_btn;

//定义列表项按钮
const char * const LIST_ITEM_BTN[][2] = {
    //含有长文本内容,将会看到循环滚动效果
    {LV_SYMBOL_FILE,"This is a long long text!"};
    {NULL,"Open"},//不给图标
    {LV_SYMBOL_CLOSE,"Close"},
```



```
{LV_SYMBOL_EDIT,"Edit"},
{LV_SYMBOL_SAVE,"Save"}
};
//总共有多少个列表项按钮
#define LIST_ITEM_BTN_NUM (sizeof(LIST_ITEM_BTN)/sizeof(LIST_ITEM_BTN[0]))

//事件回调函数
static void event_handler(lv_obj_t * obj,lv_event_t event)
{
    if(event==LV_EVENT_RELEASED)
    {
        printf("list item btn: %s\n",lv_list_get_btn_text(obj));
    }
}

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

    //1.创建背景样式
    lv_style_copy(&bg_style,&lv_style_transp);
    bg_style.body.border.width = 2;
    bg_style.body.border.color = LV_COLOR_MAKE(0x36,0x5D,0x85);
    bg_style.body.radius = 6;
    bg_style.body.padding.right = 8;//给右侧的滚动条留点空间

    //2.创建列表对象
    list1 = lv_list_create(scr,NULL);
    lv_obj_set_size(list1,150,150);//设置大小
    lv_obj_align(list1,NULL,LV_ALIGN_CENTER,0,0);//与屏幕居中对齐
    lv_list_set_single_mode(list1,true);//使能列表的单选模式
    lv_list_set_sb_mode(list1,LV_SB_MODE_AUTO);//设置滚动条模式
    lv_list_set_style(list1,LV_LIST_STYLE_BG,&bg_style);//设置背景样式

    //设置页面中容器载体的样式
    lv_list_set_style(list1,LV_LIST_STYLE_SCRL,&lv_style_transp);

    //2.1 添加列表项按钮
    uint8_t i;
    lv_obj_t * list_item_btn;
```

```
for(i=0;i<LIST_ITEM_BTN_NUM;i++)
{
    //添加列表项按钮
    list_item_btn = lv_list_add_btn(list1,LIST_ITEM_BTN[i][0],LIST_ITEM_BTN[i][1]);
    lv_obj_set_event_cb(list_item_btn,event_handler);//给列表项按钮设置事件回调函数
}
last_item_btn = list_item_btn;//保存最后一个列表项按钮对象
}

//按键处理
void key_handler()
{
    u8 key = KEY_Scan(0);

    if(key==KEY0_PRES)
    {
        lv_list_up(list1);//让列表向上移动一步
    }else if(key==KEY1_PRES)
    {
        lv_list_down(list1);//让列表向下移动一步
    }else if(key==WKUP_PRES)
    {
        //让最后一个列表项按钮直接处于可见状态
        lv_list_focus(last_item_btn,LV_ANIM_ON);
    }
}
```

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:

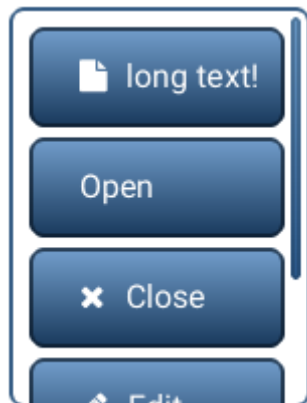


图 3.4.1 初始界面效果

然后我们可以随便滚动列表,并按一下 Close 按钮,使其处于选中状态,如下图所示:

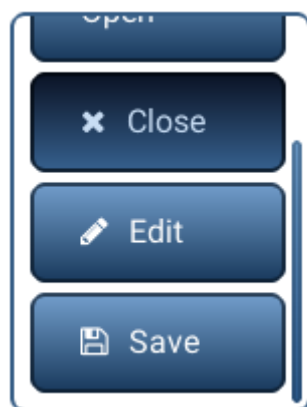


图 3.4.2 列表的单选效果

最后我们可以按下 KEY0,KEY1 或者 WKUP 按键来测试列表的手动导航效果.

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