

## 正点原子 littleVGL 开发指南

lv\_win 窗体

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\_win 窗体

## 1. 介绍

lv\_win 窗体是一个最复杂的类似于容器的控件,它主要是由上下两部分组成,上面主要是一个 header 容器,下面主要是一个存放内容的 page 页面,在 header 容器里的左侧是一个窗体标题,在右侧是一个控制按钮栏,所以换句话说,它其实就是我们之前学过的 lv\_cont 容器控件,lv\_label 标签控件,lv\_page 页面控件的一个组合使用,具体的构成图如下所示:

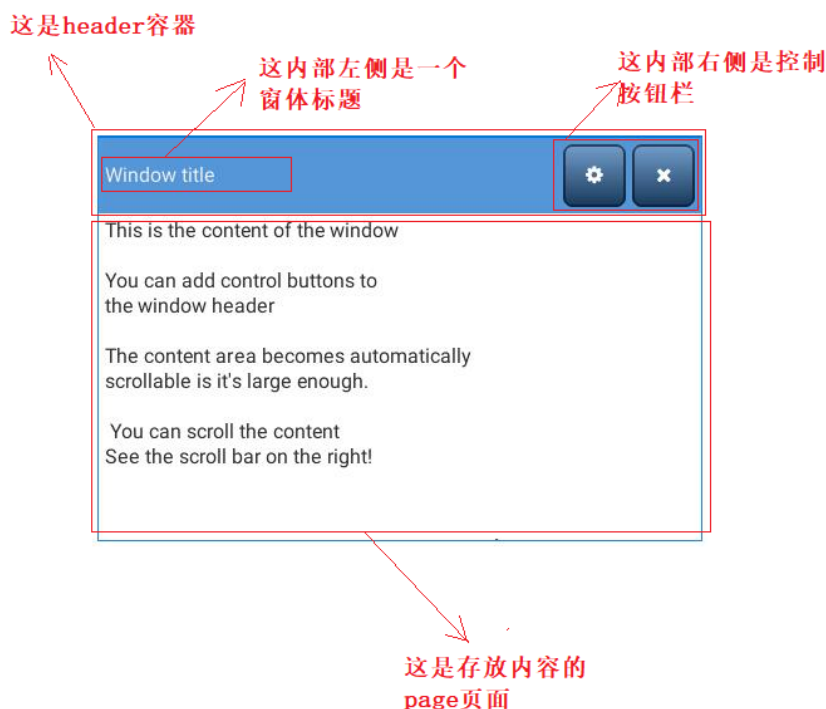


图 1.1 lv\_win 窗体的构成

对于上面的窗体标题你是可以通过 lv\_win\_set\_title(win, title)接口来设置的,当然了,你还可以往右侧的控制按钮栏添加控制按钮,这是通过 lv\_win\_add\_btn(win, &img\_src)接口来完成的,其中的 img\_src 就是 lv\_img 图片章节中学过图片源,被添加的控制按钮是依次从右往左排的,而且按钮的高度和宽度是相等的,不过你可以用 lv\_win\_set\_btn\_size(win, size)接口来给所有的控制按钮设置统一的大小,然后这里有一点你需要注意的是,header 容器的高度会根据控制按钮的高度来自动确定的.

我们知道 lv\_win 窗体的下面是一个用来存放内容的 lv\_page 页面控件,所以 lv\_win 窗体是具有 lv\_page 的某些特性的,比如用 lv\_win\_set\_sb\_mode(win, LV\_SB\_MODE\_...)接口来设置窗体中的滚动条模式,用 lv\_win\_scroll\_hor(win, dist\_px)接口和 lv\_win\_scroll\_ver(win, dist\_px)接口来滚动窗体页面,或者用 lv\_win\_focus(win, child, LV\_ANIM\_ON/OFF)接口来将窗体页面中的某个子对象直接处于可见状态,对于滚动和 focus 操作动画的时长是可以通过 lv\_win\_set\_anim\_time(win, anim\_time\_ms)接口来设置的,如果你想调整窗体页面的布局方式,你可以调用 lv\_win\_set\_layout(win, LV\_LAYOUT\_...)接口来进行设置,上面说到的这些 API 接口都是能够在 lv\_page 页面控件章节中找到对应体的.

## 2. lv\_win 的 API 接口

### 2.1 主要数据类型

#### 2.1.1 窗体样式数据类型

```
enum {  
    LV_WIN_STYLE_BG,  
    LV_WIN_STYLE_CONTENT,  
    LV_WIN_STYLE_SB,  
    LV_WIN_STYLE_HEADER,  
    LV_WIN_STYLE_BTN_REL,  
    LV_WIN_STYLE_BTN_PR,  
};  
typedef uint8_t lv_win_style_t;
```

LV\_WIN\_STYLE\_BG: 用来修饰窗体背景的,使用样式中的 body 字段,默认值为 lv\_style\_plain

LV\_WIN\_STYLE\_CONTENT: 用来修饰存放内容的 page 页面,使用样式中的 body 字段,默认值为 lv\_style\_transp

LV\_WIN\_STYLE\_SB: 用来修饰 page 页面中的滚动条,默认值为 lv\_style\_pretty\_color,具体用法请参考 lv\_page 页面章节

LV\_WIN\_STYLE\_HEADER: 用来修饰 header 容器的,默认值为 lv\_style\_plain\_color

LV\_WIN\_STYLE\_BTN\_REL: 用来修饰控制按钮的松手状态,使用样式中的 body 字段,默认值为 lv\_style\_btn\_rel

LV\_WIN\_STYLE\_BTN\_PR: 用来修饰控制按钮的按下状态,使用样式中的 body 字段,默认值为 lv\_style\_btn\_pr

## 2.2 API 接口

#### 2.2.1 创建对象

```
lv_obj_t * lv_win_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 添加控制按钮

```
lv_obj_t * lv_win_add_btn(lv_obj_t * win, const void * img_src);
```

#### 参数:

win: 窗体对象

img\_src: 按钮上显示的图片,此参数就是 lv\_img 图片章节中的图片来源

#### 返回值:

返回被添加的控制按钮对象

拿到被添加的控制按钮对象之后,你可以用 lv\_obj\_set\_event\_cb 接口来给此对象设置你自定义的事件回调函数,在事件回调函数中实现你自己的控制功能,如果你是想点击某控制按钮去关闭整个窗体的话,那么这里有一种快速的办法,如下所示:

```
lv_obj_set_event_cb(close_btn, lv_win_close_event_cb);
```

其中 close\_btn 是一个被添加的控制按钮,而 lv\_win\_close\_event\_cb 是 littleVGL 系统内部定义的一个事件回调函数,它的作用就是监听 LV\_EVENT\_RELEASED 事件,监听到了的话,把整个窗体给关闭掉.

### 2.2.3 设置窗体的标题

```
void lv_win_set_title(lv_obj_t * win, const char * title);
```

#### 参数:

win: 窗体对象

title: 窗体的标题

### 2.2.4 设置控制按钮的大小

```
void lv_win_set_btn_size(lv_obj_t * win, lv_coord_t size);
```

#### 参数:

win: 窗体对象

size: 按钮的大小,宽=高=size

这是给所有的控制按钮设置统一的大小,当控制按钮的大小发生变化时,header 容器的高度也会自动发生变化,以实现对子对象的包裹

### 2.2.5 设置窗体页面的布局方式

```
void lv_win_set_layout(lv_obj_t * win, lv_layout_t layout);
```

#### 参数:

win: 窗体对象

layout: 布局方式

这个 API 接口的使用方法和 lv\_page 页面章节中的 lv\_page\_set\_scr1\_layout 接口的使用方法是一样的

### 2.2.6 设置窗体页面中的滚动条模式

```
void lv_win_set_sb_mode(lv_obj_t * win, lv_sb_mode_t sb_mode);
```

参数:

win: 窗体对象

sb\_mode: 滚动条模式

这个 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

### 2.2.7 设置动画时长

```
void lv_win_set_anim_time(lv_obj_t * win, uint16_t anim_time);
```

参数:

win: 窗体对象

anim\_time: 滚动和 focus 动画的时长,单位为 ms

### 2.2.8 设置样式

```
void lv_win_set_style(lv_obj_t * win, lv_win_style_t type, const lv_style_t * style);
```

参数:

win: 窗体对象

type: 设置哪一部分的样式,有如下 6 个可选值

LV\_WIN\_STYLE\_BG: 用来修饰窗体背景的

LV\_WIN\_STYLE\_CONTENT: 用来修饰存放内容的 page 页面

LV\_WIN\_STYLE\_SB: 用来修饰 page 页面中的滚动条

LV\_WIN\_STYLE\_HEADER: 用来修饰 header 容器的

LV\_WIN\_STYLE\_BTN\_REL: 用来修饰控制按钮的松手状态

LV\_WIN\_STYLE\_BTN\_PR: 用来修饰控制按钮的按下状态

style: 样式

### 2.2.9 设置窗体是否能够被拖拽

```
void lv_win_set_drag(lv_obj_t * win, bool en);
```

#### 参数:

win: 窗体对象

en: 是否使能窗体的拖拽功能

当使能拖拽之后,要从窗体上面的 header 容器区域拖拽才有效,从下面的 page 页面区域拖拽是无效的

### 2.2.10 将页面中的某个子对象处于可见状态

```
void lv_win_focus(lv_obj_t * win, lv_obj_t * obj, lv_anim_enable_t anim_en);
```

#### 参数:

win: 窗体对象

anim\_en: 是否使能动画效果,有如下 2 个可选值

LV\_ANIM\_OFF:不使能

LV\_ANIM\_ON:使能

### 2.2.11 将窗体页面进行水平滚动

```
static inline void lv_win_scroll_hor(lv_obj_t * win, lv_coord_t dist);
```

#### 参数:

win: 窗体对象

dist: 水平滚动距离

这个 API 接口的使用方法和 lv\_page 页面章节中的 lv\_page\_scroll\_hor 接口的使用方法是相同的

### 2.2.12 将窗体页面进行垂直滚动

```
static inline void lv_win_scroll_ver(lv_obj_t * win, lv_coord_t dist);
```

#### 参数:

win: 窗体对象

dist: 垂直滚动距离

这个 API 接口的使用方法和 lv\_page 页面章节中的 lv\_page\_scroll\_ver 接口的使用方法是相同的

### 2.2.13 获取窗体的 page 页面对象

```
lv_obj_t * lv_win_get_content(const lv_obj_t * win);
```

参数:

win: 窗体对象

返回值:

返回窗体中用于存放内容的页面对象,拿到此页面对象之后,你就可以使用 lv\_page 页面控件专有的 API 接口来操作它了

### 2.2.14 获取某控制按钮所在的窗体对象

```
lv_obj_t * lv_win_get_from_btn(const lv_obj_t * ctrl_btn);
```

参数:

ctrl\_btn: 控制按钮对象

返回值:

返回此控制按钮所在的窗体对象

### 2.2.15 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窗体的背景,然后创建一个窗体对象,设置其窗体标题,设置其控制按钮的大小,然后使其拖拽功能,接着给其添加 2 个控制按钮,一个是用来关闭窗体的,一个是用来修改窗体标题的,最后往窗体页面中添加一个长文本的标签子对象.

### 3.2 硬件设计

本例程所用到的硬件有:

- 1) 液晶屏

### 3.3 软件设计

在 GUI\_APP 目录下创建 lv\_win\_test.c 和 lv\_win\_test.h 两个文件,其中 lv\_win\_test.c 文件的内容如下:

```
#include "lv_win_test.h"
#include "lvgl.h"

lv_style_t bg_style;
lv_obj_t * title_btn;

//事件回调函数
void event_handler(lv_obj_t * obj, lv_event_t event)
{
    if(obj==title_btn)
    {
        if(event==LV_EVENT_RELEASED)
        {
            //获取此控制按钮所在的窗体对象,其实就是 win1 对象
            lv_obj_t * win = lv_win_get_from_btn(title_btn);
            lv_win_set_title(win,"a new title");//修改窗体标题
        }
    }
}
```



```
}

//例程入口
void lv_win_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_plain);
    bg_style.body.border.width = 2;
    bg_style.body.border.color = LV_COLOR_MAKE(0x55,0x96,0xD8);

    //2.创建窗体对象
    lv_obj_t * win1 = lv_win_create(scr,NULL);
    lv_obj_set_size(win1,220,220);//设置窗体的大小
    lv_obj_align(win1,NULL,LV_ALIGN_CENTER,0,0);//与屏幕居中对齐
    lv_win_set_title(win1,"Win title");//设置窗体的标题
    lv_win_set_btn_size(win1,30);//设置控制按钮的大小
    lv_win_set_drag(win1,true);//使能拖拽功能
    lv_win_set_style(win1,LV_WIN_STYLE_BG,&bg_style);//设置背景样式

    //2.1 往窗体中添加一个控制按钮,用于关闭窗体
    lv_obj_t * close_btn = lv_win_add_btn(win1,LV_SYMBOL_CLOSE);

    //给控制按钮设置事件回调函数,lv_win_close_event_cb 是 littleVGL 系统内部自定义的
    //一个事件回调函数
    //专门用于关闭窗体的
    lv_obj_set_event_cb(close_btn,lv_win_close_event_cb);

    //2.2 再往窗体中添加一个控制按钮,用于修改窗体的标题
    title_btn = lv_win_add_btn(win1,LV_SYMBOL_SETTINGS);
    lv_obj_set_event_cb(title_btn,event_handler);

    //2.3 往窗体页面中添加一个长文本的标签子对象
    lv_obj_t * label1 = lv_label_create(win1,NULL);
    lv_label_set_text(label1, "This is the content of the window\n\n"
        "You can add control buttons to\n"
        "the window header\n\n"
        "The content area becomes automatically\n"
        "scrollable is it's large enough.\n\n"
        " You can scroll the content\n"
        "See the scroll bar on the right!");
}
```

### 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:

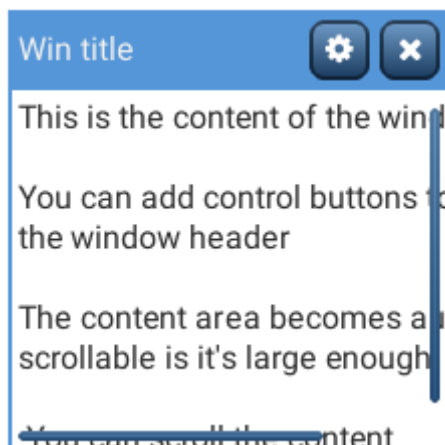



图 3.4.1 初始界面效果

然后我们可以点击一下  控制按钮,可以看到窗体的标题发生了变化,如下图所示:

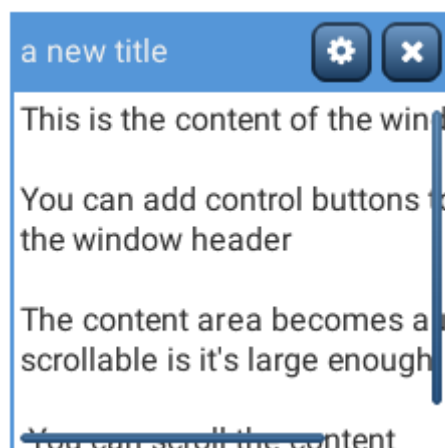



图 3.4.2 窗体标题发生变化

接着你可以从窗体的 header 容器区域测试拖拽效果,最后你可以点击一下  控制按钮来把整个窗体给关闭了.

## 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](http://www.openedv.com/thread-309664-1-1.html)

原子哥在线教学平台：[www.yuanzige.com](http://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](http://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