User Guide

正点原子 littleVGL 开发指南

lv_slider 滑块

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_slider 滑块

1. 介绍

lv_slider 滑块是由 lv_bar 进度条对象外加一个类似于旋钮的东西构成的,这个旋钮可以被拖拽来设置 lv_slider 滑块的值,和 lv_bar 进度条一样,lv_slider 也可以被设置成是水平滑块或者是垂直滑块,不仅如此,lv_bar 对象上的大部分特性在 lv_slider 对象上基本都能找到,比如设置进度值,动画时间,设置最小最大范围等等,API 接口的用法基本是一模一样的.最后来说一下它的事件,当滑块被点击或者滑块上面的旋钮被拖拽导致其进度值发生变化时,它就会给它的事件回调函数发送一个 LV_EVENT_VALUE_CHANGED 事件,如果旋钮是在被持续拖拽的话,那么 LV_EVENT_VALUE_CHANGED 事件也将会被持续发送,有时候我们可能不希望持续接受到此事件,那我们只需忽略就行,只监听它的 LV_EVENT_RELEASED 松手事件来获取最后的进度值.

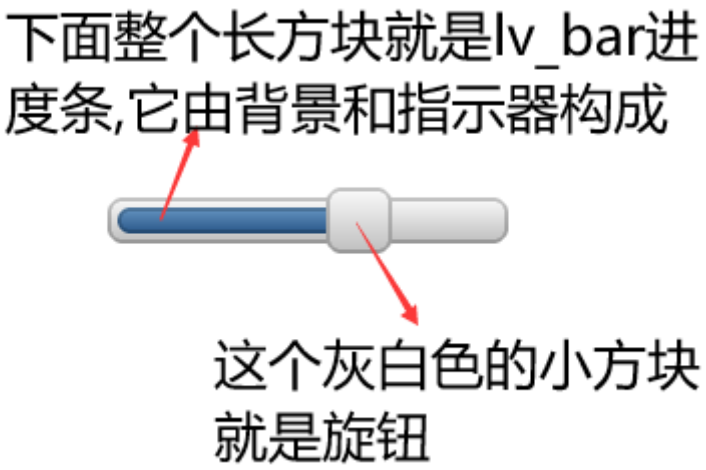

图 1.1 lv_slider 对象组成

2. lv_slider 的 API 接口

2.1 主要数据类型

3. 滑块样式数据类型

```
enum {
    LV_SLIDER_STYLE_BG,
    LV_SLIDER_STYLE_INDIC,
    LV_SLIDER_STYLE_KNOB,
};
typedef uint8_t lv_slider_style_t;
```

LV_SLIDER_STYLE_BG: 滑块的背景样式,其实就是其内部的 lv_bar 进度条的背景样式,使用样式中的 style.body 字段,其中的 padding 字段用来设置背景边框与旋钮边框之间的距离

LV_SLIDER_STYLE_INDIC: 滑块的指示器样式,其实就是其内部的 lv_bar 进度条的指示器样式,使用样式中的 style.body 字段,其中的 padding 字段设置指示器与背景边框之间的距离

LV_SLIDER_STYLE_KNOB: 滑块上旋钮的样式,使用样式中的 style.body 字段,但是其内部的 padding 字段除外

2.2 API 接口

2.2.1 创建对象

```
lv_obj_t * lv_slider_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 设置动画时长

```
static inline void lv_slider_set_anim_time(lv_obj_t * slider, uint16_t anim_time);
```

参数:
slider: 滑块对象
anim_time: 动画时长,单位 ms
注意此 API 接口必须得放在 lv_slider_set_value 接口前面调用,否则无效

2.2.3 设置进度值

```
static inline void lv_slider_set_value(lv_obj_t * slider, int16_t value, lv_anim_enable_t anim);
```

参数:
slider: 滑块对象
value: 新的进度值
anim: 在切换到新的进度值时,是否使能动画效果,有 2 个可选值如下:
LV_ANIM_OFF: 不使能动画效果
LV_ANIM_ON: 使能动画效果

2.2.4 设置进度范围

```
static inline void lv_slider_set_range(lv_obj_t * slider, int16_t min, int16_t max);
```

参数:
slider: 滑块对象
min: 最小值
max: 最大值

2.2.5 设置样式

```
void lv_slider_set_style(lv_obj_t * slider, lv_slider_style_t type, const lv_style_t * style);
```

参数:
slider: 滑块对象
type: 设置哪一个部件的样式,有如下 3 个可选值:
LV_SLIDER_STYLE_BG: 设置背景的样式
LV_SLIDER_STYLE_INDIC: 设置指示器的样式
LV_SLIDER_STYLE_KNOB: 设置旋钮的按钮
style: 样式

2.2.6 获取当前的进度值

```
int16_t lv_slider_get_value(const lv_obj_t * slider);
```

参数:
slider: 滑块对象

返回值:
返回当前的进度值

2.2.7 判断旋钮是否正在被拖拽

```
bool lv_slider_is_dragged(const lv_obj_t * slider);
```

参数:
slider: 滑块对象

返回值:
返回 true 代表正在被拖拽,返回 false 代表没有在被拖拽

2.2.8 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 3 个自定义样式,分别用于修饰滑块的背景,指示器和旋钮,然后创建一个水平滑块和一个标签,并为滑块对象设置事件回调函数,在事件回调函数中,把滑块的当前进度值显示在标签上

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_slider_test.c 和 lv_slider_test.h 俩个文件,其中 lv_slider_test.c 文件的内容如下:

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

lv_style_t slider_bg_style;//背景的样式
lv_style_t slider_indic_style;//指示器的样式
lv_style_t slider_knob_style;//旋钮的样式

lv_obj_t * label1;

//事件回调函数
static void event_handler(lv_obj_t * obj,lv_event_t event)
{
    char buff[16];
    //lv_slider 的进度值发生了改变或者松手了
    if(event==LV_EVENT_VALUE_CHANGED||event==LV_EVENT_RELEASED)
    {
        //将当前的进度显示在 label1 标签中,如果是松手事件的话,则加上"End:"前缀
        sprintf(buff,event==LV_EVENT_VALUE_CHANGED?"%d%%":"End:%d%%",lv
_slider_get_value(obj));
        lv_label_set_text(label1,buff);
```

```
            lv_obj_realign(label1);
        }
}

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

    //1.创建用于滑块的 3 个样式
    //1.1 创建背景样式
    lv_style_copy(&slider_bg_style,&lv_style_pretty);
    slider_bg_style.body.main_color = LV_COLOR_BLACK;
    slider_bg_style.body.grad_color = LV_COLOR_GRAY;
    slider_bg_style.body.radius = LV_RADIUS_CIRCLE;
    slider_bg_style.body.border.color = LV_COLOR_WHITE;

    //1.2 创建指示器的样式
    lv_style_copy(&slider_indic_style,&lv_style_pretty_color);
    slider_indic_style.body.main_color = LV_COLOR_MAKE(0x5F,0xB8,0x78);
    slider_indic_style.body.grad_color = LV_COLOR_MAKE(0x5F,0xB8,0x78);
    slider_indic_style.body.radius = LV_RADIUS_CIRCLE;
    slider_indic_style.body.shadow.width = 8;
    slider_indic_style.body.shadow.color = slider_indic_style.body.main_color;
    slider_indic_style.body.padding.left = 3;//设置指示器与背景边框之间的距离
    slider_indic_style.body.padding.right = 3;
    slider_indic_style.body.padding.top = 3;
    slider_indic_style.body.padding.bottom = 3;

    //1.3 创建旋钮的样式
    lv_style_copy(&slider_knob_style,&lv_style_pretty);
    slider_knob_style.body.radius = LV_RADIUS_CIRCLE;
    slider_knob_style.body.opa = LV_OPA_70;

    //2.创建滑块对象
    lv_obj_t * slider1 = lv_slider_create(scr,NULL);

    //设置大小,当宽度比高度大时,是水平滑块,当宽度比高度小时,是垂直滑块
    lv_obj_set_size(slider1,200,20);
    lv_slider_set_range(slider1,0,100);//设置进度范围

    //设置动画时长,必须得放在 lv_slider_set_value 前面调用,否则无效
    lv_slider_set_anim_time(slider1,1000);
    lv_slider_set_value(slider1,70,LV_ANIM_ON);//设置当前的进度值,使能动画效果
```

```
    //设置背景样式
    lv_slider_set_style(slider1,LV_SLIDER_STYLE_BG,&slider_bg_style);
    //设置指示器的样式
    lv_slider_set_style(slider1,LV_SLIDER_STYLE_INDIC,&slider_indic_style);
    //设置旋钮的样式
    lv_slider_set_style(slider1,LV_SLIDER_STYLE_KNOB,&slider_knob_style);
    lv_obj_align(slider1,NULL,LV_ALIGN_CENTER,0,0);//设置与屏幕居中对齐
    lv_obj_set_event_cb(slider1,event_handler);//设置事件

    //3.创建标签,用来显示滑块的当前进度
    label1 = lv_label_create(scr,NULL);
    lv_obj_align(label1,slider1,LV_ALIGN_OUT_TOP_MID,0,-10);
    lv_label_set_text(label1,"70%");
}
```

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:

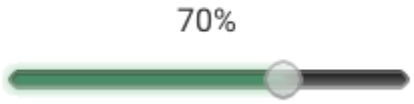

图 3.4.1 滑块演示效果

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