

正点原子 littleVGL 开发指南

lv_line 线条

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_line 线条

1. 介绍

lv_line 线条简单来说是由多个点连接而成的对象,它可以通过 lv_obj_set_size 接口来设置固定的大小,也可以通过 lv_line_set_auto_size(line,true)接口来设置线条对象的大小自适应,它会根据其内部所有点中最大的 x 和 y 坐标来算出自身的大小,默认情况下,大小自适应功能是被使能了的,对于线条对象来说,其内部的所有点坐标默认情况下都是以其左上角为参考原点的,当然了,你可以通过 lv_line_set_y_invert(line, true)接口来反转 y 轴,从而使左下角变为参考原点.

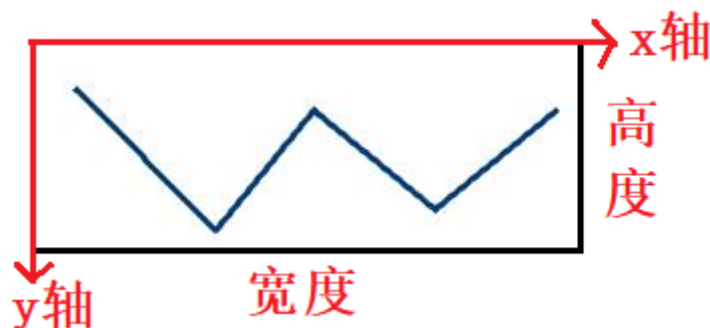


图 1.1 lv_line 对象组成

注:创建出来的 lv_line 对象背景是透明的,而且也是没有边框的,上面图片中具有黑色和红色边框是为了给大家讲解方便,才加上去的

2. lv_line 的 API 接口

2.1 主要数据类型

由于 lv_line 控件过于简单,所以没有啥可讲的数据类型

2.2 API 接口

2.2.1 创建对象

```
lv_obj_t * lv_line_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 设置坐标点集合

```
void lv_line_set_points(lv_obj_t * line, const lv_point_t point_a[], uint16_t point_num);
```

参数:

line: 线条对象

point_a: 坐标点集合,数组的形式,如下所示:

```
const lv_point_t line_points[] = { {20, 30}, {70, 70}, {120, 10}};
```

point_num: 坐标点的个数

线是由点构成的,这里传入 point_num 个坐标点,就可以绘制出 point_num-1 条线

2.2.3 是否使能大小自适应

```
void lv_line_set_auto_size(lv_obj_t * line, bool en);
```

参数:

line: 线条对象

en: 是否使能

默认情况下,线条对象的大小自适应功能是被使能了的,它会根据其内部所有点中最大的 x 和 y 坐标来算出自身的大小,这个大小计算的过程其实是在 lv_line_set_points 接口内部完成的

2.2.4 是否设置 y 轴反转

```
void lv_line_set_y_invert(lv_obj_t * line, bool en);
```

参数:

line: 线条对象

en: 是否使能

默认情况下是以 lv_line 对象的左上角为参考原点的,如果你调用了 lv_line_set_y_invert 接口来反转 y 轴的话,那么参考原点就会变为左下角

2.2.5 设置样式

```
static inline void lv_line_set_style(lv_obj_t * line, lv_line_style_t type, const lv_style_t * style);
```

参数:

line: 线条对象

type: 设置哪一个部件的样式,目前就只有 LV_LINE_STYLE_MAIN 一个可选值

style: 样式

只有 style.line 字段才会被 lv_line 对象使用到

2.2.6 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,用来修饰线条对象,然后接着创建一个线条对象,并为其设置好坐标点集合,当按下 KEY0 键时,我们让此线条对象进行 y 轴反转

3.2 硬件设计

本例程所用到的硬件有:

- 1) 液晶屏
- 2) KEY0 按键

3.3 软件设计

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

```
#include "lv_line_test.h"
#include "lvgl.h"
#include "key.h"

//坐标点集合
const lv_point_t line_points[] = { {10, 20}, {70, 50}, {120, 10}, {140, 60}, {180, 10} };
//坐标点的个数
#define LINE_POINTS_NUM (sizeof(line_points)/sizeof(line_points[0]))

lv_obj_t * line1;

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

    //1.创建自定义样式
    static lv_style_t line_style;
    lv_style_copy(&line_style, &lv_style_plain);
    line_style.line.color = LV_COLOR_RED;//线条的颜色
    line_style.line.width = 4;//线条的厚度
```

```
line_style.line.rounded = 1;//线条的末端是否为圆角

//2.创建线条对象
line1 = lv_line_create(scr, NULL);//创建线条对象
lv_obj_set_pos(line1,20,20);//设置坐标
//使能大小自适应,当然了,你也可以不调用,因为默认就是被使能了的
lv_line_set_auto_size(line1,true);
//设置坐标点集合,同时也会在此内部计算出线条对象的大小
lv_line_set_points(line1, line_points, LINE_POINTS_NUM);
lv_line_set_style(line1, LV_LINE_STYLE_MAIN, &line_style);//设置样式
}

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

    if(key==KEY0_PRES)
    {
        lv_line_set_y_invert(line1,!lv_line_get_y_invert(line1));//来回取反
    }
}
```

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 y 轴未反转时的效果

然后我们按一下 KEY0 键来反转 y 轴,就可以看到如下所示的效果:



图 3.4.2 y 轴反转后的效果

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