User Guide

正点原子 littleVGL 开发指南

Events 事件

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

Events 事件

1. 介绍

在 littleVGL 中任何对象都可以注册事件,这是在新版本中才加入的特性,分为通用事件和专用事件,总共支持 20 种事件类型,这是一个总和哈,并不是指每一个对象都具有 20 种事件类型,事件可以是由 littleVGL 库自身触发的,也可以是由外部物理操作触发的,比如触摸,点击等等,当然了,我们也可以通过调用 lv_event_send 接口来手动发送事件进行触发,同时可以携带用户自定义的数据.

2. Events 的 API 接口

2.1 主要数据类型

2.1.1 事件数据类型

```
enum {
LV_EVENT_PRESSED,//对象被按下时触发,每次按下时只触发一次
LV_EVENT_PRESSING,//对象正在被按下中,只要按下不放,就会一直被触发
LV_EVENT_PRESS_LOST,//在按下的过程中,手指从对象的可视区域划出时被触发

//在 LV_INDEV_LONG_PRESS_TIME 时间之前松手触发,如果是在被拖拽的话,则不会
//被触发
LV_EVENT_SHORT_CLICKED,

//按下时长超过 LV_INDEV_LONG_PRESS_TIME 值时触发, 只会触发一次,如果是在被
//拖拽的话,则不会被触发
LV_EVENT_LONG_PRESSED,

//在触发 LV_EVENT_LONG_PRESSED 事件之后,接下来按下的时长每超过
//LV_INDEV_LONG_PRESS_REP_TIME 值一次,就会被触发一次,如果是在被拖拽的话,
//则不会被触发
LV_EVENT_LONG_PRESSED_REPEAT,

//只要松手了就会被触发,但是如果触发了 LV_EVENT_PRESS_LOST 事件的话,那么此
//事件会被触发
LV_EVENT_CLICKED,
```

```
        //和 LV_EVENT_CLICKED 事件一样,没啥区别,位于 LV_EVENT_CLICKED 事件之
        //后触发
        LV_EVENT_RELEASED,
        LV_EVENT_DRAG_BEGIN,//拖拽开始时触发
        LV_EVENT_DRAG_END,//拖拽结束时触发
        LV_EVENT_DRAG_THROW_BEGIN,//拖拽漂移开始时触发

        //当实体按键被按下时触发,我们一般都是用触摸屏作为输入,所以此事件
        //一般用不到
        LV_EVENT_KEY,
        LV_EVENT_FOCUSED,//当对象在其所在的 group 组获得焦点时触发
        LV_EVENT_DEFOCUSED,//当对象在其所在的 group 组失去焦点时触发
        LV_EVENT_VALUE_CHANGED,//对象的数值改变时被触发,如 lv_slider 滑动控件

        LV_EVENT_INSERT,//有东西插入到对象上时触发,如 lv_ta 文本框控件

        //可以说是留给用户使用的一种事件,用户只能通过 lv_event_send 接口来手动发送
        //触发此事件
        LV_EVENT_REFRESH,

        //点击 lv_kb 键盘控件上的”OK”,”Apply”等相似词义的按钮时触发
        LV_EVENT_APPLY,

        //点击 lv_kb 键盘控件上的”Close”,”Cancel”等相似词义的按钮时触发
        LV_EVENT_CANCEL,
        LV_EVENT_DELETE //对象被删除时触发
    };
    typedef uint8_t lv_event_t;
```

2.1.2 事件回调函数数据类型

```
    typedef void (*lv_event_cb_t)(struct _lv_obj_t * obj, lv_event_t event);
```

2.2 API 接口

2.2.1 设置事件回调函数

```
    void lv_obj_set_event_cb(lv_obj_t * obj, lv_event_cb_t event_cb);
```

参数:
obj: 对象句柄
event_cb: 事件回调函数

2.2.2 手动发送事件

```
lv_res_t lv_event_send(lv_obj_t * obj, lv_event_t event, const void * data);
```

参数:
obj: 给哪一个对象发送事件
event: 需要发送的事件名
data: 携带的用户自定义数据

返回值:
LV_RES_OK:对象没有被删除
LV_RES_INV:对象在事件中被删除了

这里需要注意 event 参数,系统是自带了 20 种事件类型,其对应的值是从 0 到 19,除了给 event 参数传系统自带的事件外,其实我们还可以传用户自定义的事件的,范围为:[20,255],举个简单的例子如下(只给出示意代码):

```
#define USER_EVENT_START        20
#define USER_EVENT_1            (USER_EVENT_START+1)
static void btn_event_cb(lv_obj_t * obj,lv_event_t event)
{
    if(event==USER_EVENT_1)
    {
        printf("用户自定义事件 1 被触发了\r\n");
    }
}
lv_obj_t* btn1 = lv_btn_create(src, NULL);
lv_obj_set_event_cb(btn1,btn_event_cb);//设置回调函数

//手动发送 USER_EVENT_1 事件,不携带参数
lv_event_send(btn1,USER_EVENT_1,NULL);
```

2.2.3 给任意事件回调函数手动发送事件

```
lv_res_t lv_event_send_func(lv_event_cb_t event_xcb, lv_obj_t * obj, lv_event_t event,
const void * data);
```

参数:
event_xcb: 给哪一个事件回调函数发送事件
obj: 给哪一个对象发送事件
event: 需要发送的事件名
data: 携带的用户自定义数据

返回值:

LV_RES_OK:对象没有被删除
LV_RES_INV:对象在事件中被删除了

其实这个接口也是很简单的,它只是把设置回调函数和发送事件的二步操作变成了一步,所以这个接口其实是等价 lv_obj_set_event_cb 和 lv_event_send 二个接口的调用

2.2.4 获取当前事件的用户自定义参数

```
const void * lv_event_get_data(void);
```

返回值:
返回用户自定义的数据

请注意,这个是返回当前事件的用户数据,是当前已经被触发了的事件,此 API 接口一般的用法就是放在事件回调函数中进行调用,如下所示(只给出示意代码):

```
#define USER_EVENT_START        20
#define USER_EVENT_1            (USER_EVENT_START+1)

//构建一个用户自定义数据结构体,当然了,如果你的用户数据简单,可以不用结构体
typedef struct{
char name[20];
u8 age;
}USER_DATA;
USER_DATA user_data = {{"xiong jia yu"},25};//初始化一下

//事件回调函数
static void btn_event_cb(lv_obj_t * obj,lv_event_t event)
{
USER_DATA *data;
    if(event==USER_EVENT_1)//当然了,这里可以是其他的任何事件,
    {
        data = (USER_DATA*) lv_event_get_data();//获取到用户的自定义数据
        printf(“name:%s,age:%d\r\n”,data ->name,data->age);
    }
}

void user_data_test()
{
lv_event_send(btn1,USER_EVENT_1,&user_data);//发送事件,同时携带用户自定义参数
}
```

3.例程设计

3.1 功能简介

创建一个默认按钮,用来测试各种事件,比如短按下,长按下,松手,故意在按下时划出按钮的可视区域等等操作,当用户按下 KEY0 键时,手动来发送事件,我这里弄了二种方式,第一种是发送用户自定义事件,并且同时携带用户自定义参数,第二种是发送系统自带的事件,不携带用户自定义参数,当用户按下 KEY1 键时,会使能按钮的拖拽和拖拽惯性漂移功能,使能之后,我们可以拖动按钮来测试拖拽事件,当用户按下 KEY2 键时,会把按钮对象给删除掉,这是用来测试 LV_EVENT_DELETE 事件的

3.2 硬件设计

本例程所用到的硬件有:

1) 液晶屏
2) KEY0,KEY1,KEY2 按键
3) 串口

3.3 软件设计

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

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


#define USER_EVENT_START        20
#define USER_EVENT_1            (USER_EVENT_START+1) //用户自定义事件 1

//构建一个用户自定义数据结构体,当然了,如果你的用户数据简单,可以不用结构体
typedef struct{
    char name[20];
    u8 age;
}USER_DATA;

USER_DATA user_data = {{"xiong jia yu"},25};//初始化一下
```

```
lv_obj_t * btn1;

//事件回调函数
static void btn_event_cb(lv_obj_t * obj,lv_event_t event)
{
    if(event==LV_EVENT_PRESSED)
    {
        //对象被按下时触发,每次按下时只触发一次
        printf("LV_EVENT_PRESSED\r\n");
    }else if(event==LV_EVENT_PRESSING)
    {
        //对象正在被按下中,只要按下不放,就会一直被触发
        printf("LV_EVENT_PRESSING\r\n");
    }else if(event==LV_EVENT_PRESS_LOST)
    {
        //在按下的过程中,手指从对象的可视区域划出时被触发
        printf("LV_EVENT_PRESS_LOST\r\n");
    }else if(event==LV_EVENT_SHORT_CLICKED)
    {
        //在 LV_INDEV_LONG_PRESS_TIME 时间之前松手触发,如果是
        //在被拖拽的话,则不会被触发
        printf("LV_EVENT_SHORT_CLICKED\r\n");
    }else if(event==LV_EVENT_LONG_PRESSED)
    {
        //按下时长超过 LV_INDEV_LONG_PRESS_TIME 值时触发,
        //只会触发一次,如果是在被拖拽的话,则不会被触发
        printf("LV_EVENT_LONG_PRESSED\r\n");
    }else if(event==LV_EVENT_LONG_PRESSED_REPEAT)
    {
        //在触发 LV_EVENT_LONG_PRESSED 事件之后,接
        //下来按下的时长每超过 LV_INDEV_LONG_PRESS_REP_TIME 值一次,就会
        //被触发一次,如果是在被拖拽的话,则不会被触发
        printf("LV_EVENT_LONG_PRESSED_REPEAT\r\n");
    }else if(event==LV_EVENT_CLICKED)
    {
        //只要松手了就会被触发,但是如果触发了 LV_EVENT_PRESS_LOST
        //事件的话,那么此事件将不会被触发
        printf("LV_EVENT_CLICKED\r\n");
    }else if(event==LV_EVENT_RELEASED)
    {
        //和 LV_EVENT_CLICKED 事件一样,没啥区别,位于 LV_EVENT_CLICKED 事
        //件之后触发
        printf("LV_EVENT_RELEASED\r\n");
```

```
    }else if(event==LV_EVENT_DRAG_BEGIN)
    {
        //拖拽开始时触发
        printf("LV_EVENT_DRAG_BEGIN\r\n");
    }else if(event==LV_EVENT_DRAG_END)
    {
        //拖拽结束时触发
        printf("LV_EVENT_DRAG_END\r\n");
    }else if(event==LV_EVENT_DRAG_THROW_BEGIN)
    {
        //拖拽漂移开始时触发
        printf("LV_EVENT_DRAG_THROW_BEGIN\r\n");
    }else if(event==LV_EVENT_REFRESH)
    {
        //可以说是留给用户使用的一种事件,用户只能通过 lv_event_send
        //接口来手动发送触发此事件
        printf("LV_EVENT_REFRESH\r\n");
    }else if(event==LV_EVENT_DELETE)
    {
        //对象被删除时触发
        printf("LV_EVENT_DELETE\r\n");
    }else if(event==USER_EVENT_1)
    {
        //用户自定义事件 1
        //获取用户自定义数据
        USER_DATA* data = (USER_DATA*)lv_event_get_data();
        printf("USER_EVENT_1,name:%s,age:%d\r\n",data->name,data->age);
    }
}

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

    //3.创建一个默认按钮,用来测试事件
    btn1 = lv_btn_create(scr, NULL);
    lv_obj_set_pos(btn1,20,20);//设置坐标
    lv_obj_set_size(btn1,150,50);//设置大小
    lv_obj_set_event_cb(btn1,btn_event_cb);//设置回调函数
    lv_obj_t* label = lv_label_create(btn1,NULL);//给 btn1 添加 label 子对象
    lv_label_set_text(label,"Click me");//设置文本
}
```

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

    if(btn1==NULL)
        return;
    if(key==KEY0_PRES)
    {
        //手动发送事件
        //方式 1:发送用户自定义事件,同时携带用户自定义数据
        lv_event_send(btn1,USER_EVENT_1,&user_data);

        //方式 2:发送系统自带的事件,不携带用户自定义数据
        //这里主要是为了演示一下 lv_event_send_func 接口
        //lv_event_send_func(btn_event_cb,btn1,LV_EVENT_REFRESH,NULL);
    }else if(key==KEY1_PRES)
    {
    //使能之后,主要是用来测试// LV_EVENT_DRAG_THROW_BEGIN 三个事件的
        lv_obj_set_drag(btn1,true);//使能拖拽功能
        lv_obj_set_drag_throw(btn1,true);//使能拖拽惯性漂移功能
    }else if(key==KEY2_PRES)
    {
        //删除对象,主要是用来测试 LV_EVENT_DELETE 事件
        if(btn1)
        {
            lv_obj_del(btn1);
            btn1 = NULL;
        }
    }
}
```

3.4 下载验证

把代码下载进去之后,正常的话,会看到如下所示的初始界面效果:

图 3.4.1 初始界面效果

然后大家可以按照功能简介中的操作,利用串口助手来观看实验现象

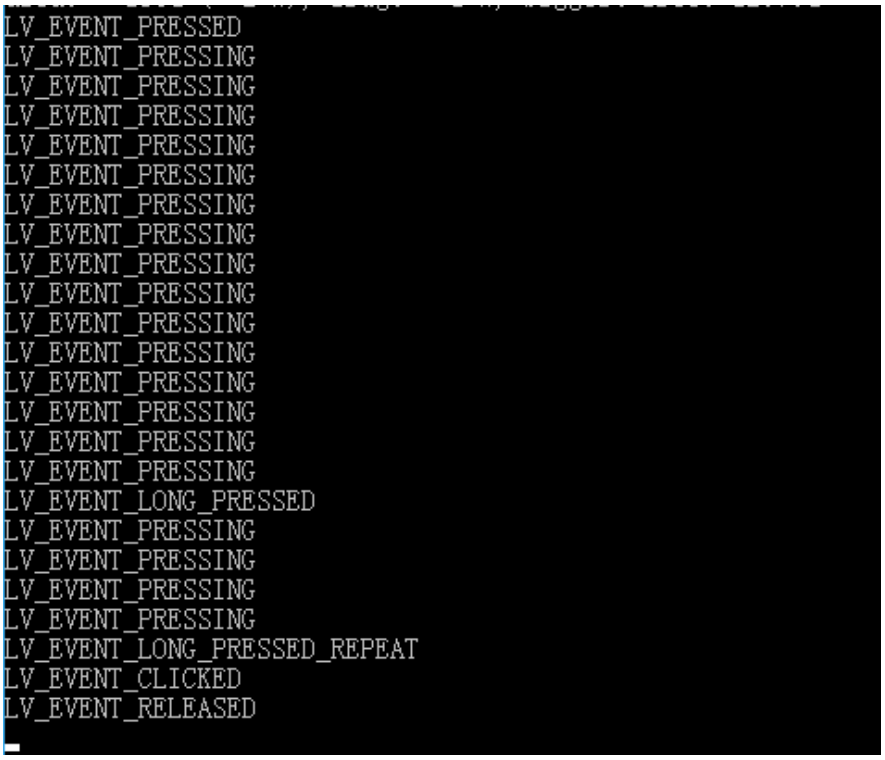

图 3.4.2 串口输出信息

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