User Guide

正点原子 littleVGL 开发指南

lv_spinbox 递增递减

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_spinbox 递增递减

1. 介绍

lv_spinbox 递增递减控件本质上就是一个 lv_ta 文本域控件,只不过在功能上做了某种延伸,主要是用于递增或者递减地调节某数值,它是可以按照某个 step 步值来精确调节的,具体的 step 步值是通过 lv_spinbox_set_step(spinbox, step)接口来设置的,比如 step 设置为 10,那么当你递增一次或者递减一次时,原来的数值将会增加 10 或者减少 10,而一次递增操作是通过 lv_spinbox_increment(spinbox)接口来完成的,递减操作是通过 lv_spinbox_decrement(spinbox)接口来完成的.当递增或者递减之后的值将要超出所设置的范围时,数值将会保持不变,而这个限定范围是通过 lv_spinbox_set_range(spinbox, min, max)接口来设置的.

先给大家上一张 lv_spinbox 的大致外观图,如下所示:

图 1.1 lv_spinbox 大致外观

有了感性认识之后,我们再来说一下 lv_spinbox 在显示方面的格式控制,我们可以把整个显示内容细分成五个小部分,第一部分就是最前面的符号位,当数值是正数时,显示+号,当数值是负数时,显示-号,然后第二部分就是处于符号位和数值之间的空格间隙了,至于有多少个空格,这是通过 lv_spinbox_set_padding_left(spinbox, cnt)接口来设置的,然后第三部分就是数值了,第四部分就是分割符小数点了,然后这里有一点必须要说清楚,那就是分隔符小数点的作用,它仅仅在显示上起到示意作用,它不属于数值部分,比如图 1.1 中显示 13.2,对于用户来说,看上去就是一个 13.2 的小数,但是对于 lv_spinbox 控件自身而言,代码层面而言,它实际是一个 132 的整数,因为在 lv_spinbox 中,数值的存储数据类型是 int32_t,所以它是不可能存储小数的,大家一定要认识到这一点,但是利用这个小数点的显示,你是可以间接实现小数功能的,因为显示的数和实际的数之间就是一个倍数转换关系而已,然后数值的总位数和分割符小数点的位置是通过 lv_spinbox_set_digit_format(spinbox, digit_count, separator_position)接口来设置的,第五部分就是闪烁显示的光标了,这个光标是用来示意当前的 step 步值精度的,比如当 step 小于 10 时,光标就会在个位上闪烁,当 step 小于 100 时,光标就会在十位上闪烁,以此类推.除了通过 lv_spinbox_set_step(spinbox, step)接口改变 step 值来改变光标的位置外,还可以通过 lv_spinbox_step_prev(spinbox)接口或 lv_spinbox_step_next(spinbox)接口来改变光标位置,这俩个 API 接口从字面意思上来理解就是将光标左移一位或者右移一位,其实它的本质还是通过修改 step 值来实现的,一个是将当前的 step 值扩大 10 倍,一个是将当前的 step 值缩小 10 倍.

最后来说一下 lv_spinbox 控件的事件,当它的数值改变时,LV_EVENT_VALUE_CHANGED 事件将会被触发.

2. lv_spinbox 的 API 接口

2.1 主要数据类型

2.1.1 样式数据类型

```
enum {
    LV_SPINBOX_STYLE_BG,
    LV_SPINBOX_STYLE_SB,
    LV_SPINBOX_STYLE_CURSOR,
};
typedef uint8_t lv_spinbox_style_t;
```

LV_SPINBOX_STYLE_BG: 修饰递增递减控件的背景,默认值为 lv_style_pretty
LV_SPINBOX_STYLE_SB: 修饰递增递减控件的滚动条,基本上用不到,我们说了 lv_spinbox 其实就是一个 lv_ta 控件,而 lv_ta 是具有滚动条的,所以 lv_spinbox 也具有滚动条,默认值为 lv_style_pretty_color
LV_SPINBOX_STYLE_CURSOR: 修饰递增递减控件的光标

2.2 API 接口

2.2.1 创建对象

```
lv_obj_t * lv_spinbox_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 设置数值范围

```
void lv_spinbox_set_range(lv_obj_t * spinbox, int32_t range_min, int32_t range_max);
```

参数:
spinbox: 递增递减对象
range_min: 最小值
range_max: 最大值

2.2.3 直接设置数值

```
void lv_spinbox_set_value(lv_obj_t * spinbox, int32_t i);
```

参数:
spinbox: 递增递减对象
i: 数值,只能是一个整数,配合分隔符小数点的位置可以实现显示小数,比如你要显示 1.23,那么这里 i 你要传入 123,然后再设置小数点到相应位置就可以了

2.2.4 设置 step 步值

```
void lv_spinbox_set_step(lv_obj_t * spinbox, uint32_t step);
```

参数:
spinbox: 递增递减对象
step: 只能传入一个整数,比如 step 设置为 5,那么当你递增一次或者递减一次时,原来的数值将会增加 5 或者减少 5

如果不设置的话,那么 step 的默认值为 1

2.2.5 设置空格间隙

```
void lv_spinbox_set_padding_left(lv_obj_t * spinbox, uint8_t padding);
```

参数:
spinbox: 递增递减对象
padding: 在符号位和数值之间的空格个数

2.2.6 设置样式

```
static inline void lv_spinbox_set_style(lv_obj_t * spinbox, lv_spinbox_style_t type, lv_style_t * style);
```

参数:
spinbox: 递增递减对象
type: 设置哪一部分的样式,有如下 3 个可选值
LV_SPINBOX_STYLE_BG: 修饰背景的
LV_SPINBOX_STYLE_SB: 修饰滚动条的
LV_SPINBOX_STYLE_CURSOR: 修饰光标的
style: 样式

2.2.7 设置显示格式

```
void lv_spinbox_set_digit_format(lv_obj_t * spinbox, uint8_t digit_count,
uint8_t separator_position);
```

参数:
spinbox: 递增递减对象
digit_count: 设置数值的总位数,当数值的实际位数不够此位数时,会在数值的前面加前导 0 进行补齐
separator_position: 分割符小数点的位置,从数值左边的第一位开始算为 1,后面的依次增 1,当设置为 0 时,代表无小数点

2.2.8 将光标右移一位

```
void lv_spinbox_step_next(lv_obj_t * spinbox);
```

参数:
spinbox: 递增递减对象

此 API 接口的实现本质是将当前值的 step 值缩小 10 倍

2.2.9 将光标左移一位

```
void lv_spinbox_step_prev(lv_obj_t * spinbox);
```

参数:
spinbox: 递增递减对象

此 API 接口的实现本质是将当前值的 step 值扩大 10 倍

2.2.10 将数值进行递增一次

```
lv_spinbox_increment(lv_obj_t * spinbox);
```

参数:
spinbox: 递增递减对象

2.2.11 将数值进行递减一次

```
void lv_spinbox_decrement(lv_obj_t * spinbox);
```

参数:
spinbox: 递增递减对象

2.2.12 获取当前的数值

```
int32_t lv_spinbox_get_value(lv_obj_t * spinbox);
```

参数:
spinbox: 递增递减对象

返回值:
返回当前的数值

2.2.13 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 spinbox 对象,设置其 step 步值为 10,数值范围为[-1000,1000],设置空格间隙为 3 个空格,设置数值显示格式为 4 位长度,小数点位置为第 3 位,最后给其设置事件回调函数,在 LV_EVENT_VALUE_CHANGED 事件中,通过串口打印当前的数值,当按下 KEY0 按键时,进行一次递增操作,当按下 KEY1 按键时,进行一次递减操作.

3.2 硬件设计

本例程所用到的硬件有:

1) 液晶屏
2) KEY0,KEY1 按键
3) 串口

3.3 软件设计

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

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


lv_obj_t * spinbox1;

//事件回调函数
void event_handler(lv_obj_t * obj,lv_event_t event)
{
    if(event==LV_EVENT_VALUE_CHANGED)
    {
        printf("spinbox value:%d\r\n",lv_spinbox_get_value(spinbox1));
    }
}
```

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

        //1.创建 spinbox 对象
    spinbox1 = lv_spinbox_create(scr,NULL);
    lv_obj_set_size(spinbox1,80,30);//设置大小
    lv_obj_align(spinbox1,NULL,LV_ALIGN_CENTER,0,0);//与屏幕居中对齐
    lv_spinbox_set_step(spinbox1,10);//设置 step 步值为 10
    lv_spinbox_set_range(spinbox1,-1000,1000);//设置数值范围[-1000,1000]
    lv_spinbox_set_padding_left(spinbox1,3);//设置空格间隙

    //设置显示格式,数值总共 4 位,小数点在第 3 位
    lv_spinbox_set_digit_format(spinbox1,4,3);
    lv_spinbox_set_value(spinbox1,0);//设置起始值为 0
    lv_obj_set_event_cb(spinbox1,event_handler);//设置事件回调函数
}


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

        if(key==KEY0_PRES)
        {
                lv_spinbox_increment(spinbox1);//进行递增操作
        }else if(key==KEY1_PRES)
        {
                lv_spinbox_decrement(spinbox1);//进行递减操作
        }
}
```

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

当按一下 KEY0 按键时,可以看到当前数值会加 10,当按一下 KEY1 按键时,可以看到当前数值会减 10,同时串口也会打印出当前的数值.

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