

## 正点原子 littleVGL 开发指南

lv\_theme 主题

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\_theme 主题

## 1. 介绍

lv\_theme 主题是 littleVGL 图形库内置的一个特色功能,在我们的项目中直接使用 lv\_theme 主题可以使我们的 UI 界面比较美观而且色调统一,最主要的原因是可以大大缩短我们项目的开发周期,因为我们不需要再去重新定义 UI 控件的样式了,这可以使我们更加专注地开发项目的业务逻辑,而对于那些 UI 控件样式,lv\_theme 主题已经为我们全部做好了,我们只需要从其中挑选我们满意的主题,然后直接应用到项目中即可。

所谓的主题,就是 littleVGL 为内置的所有控件做了一层 UI 样式的封装,相当于把某个控件创建出来之后,此控件就具有了指定主题中的样式外观,当然了你可以在后续的操作中用自己自定义的样式来覆盖掉其原有的主题样式, littleVGL 从清爽,简约的审美观出发,总共为我们制订了 8 种主题,这些主题界面效果分别如下图所示:

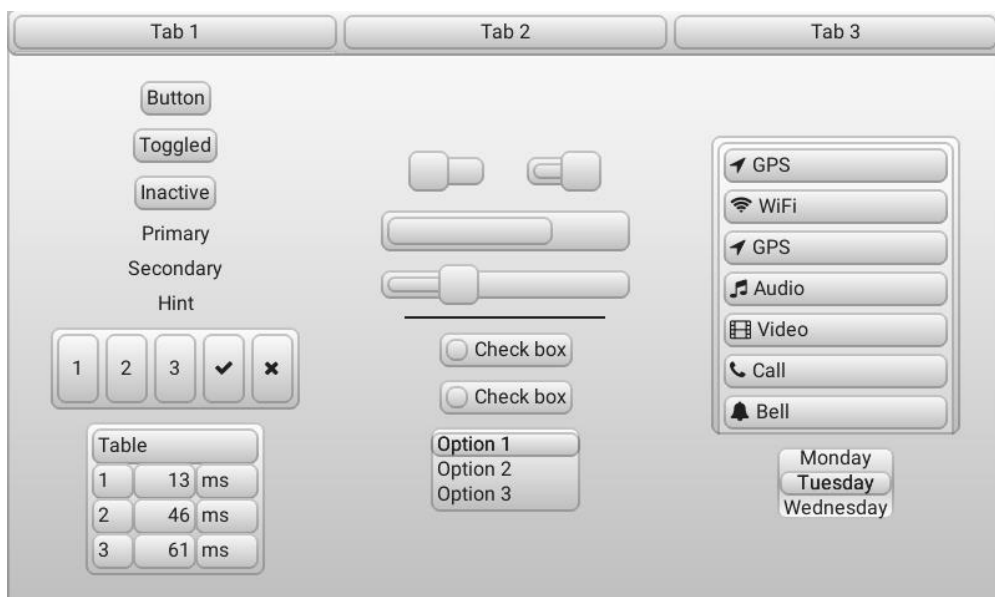


图 1.1 templ 主题

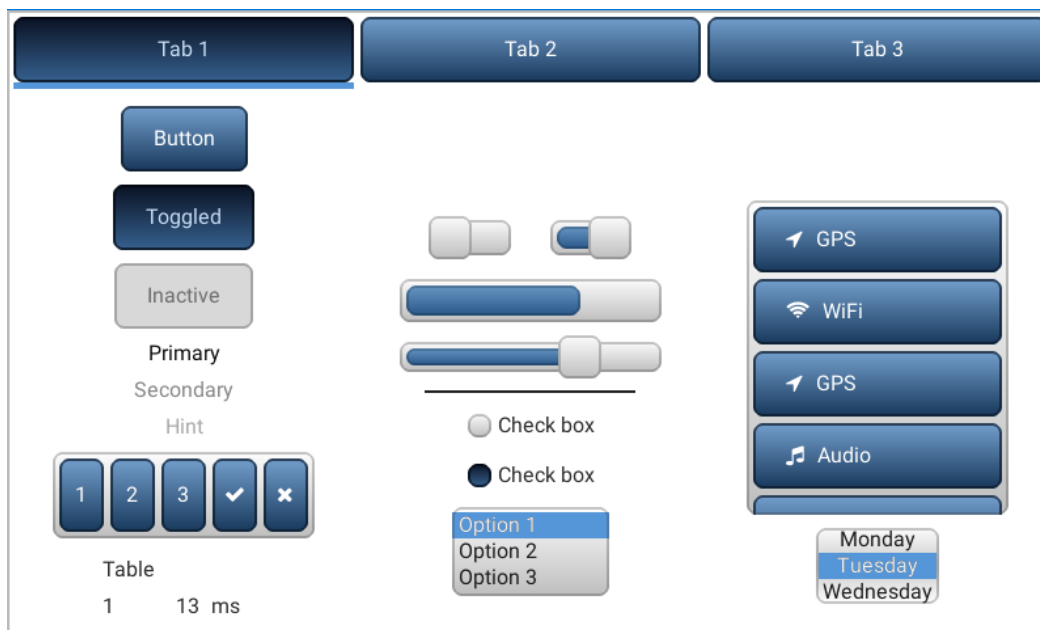


图 1.2 default 主题

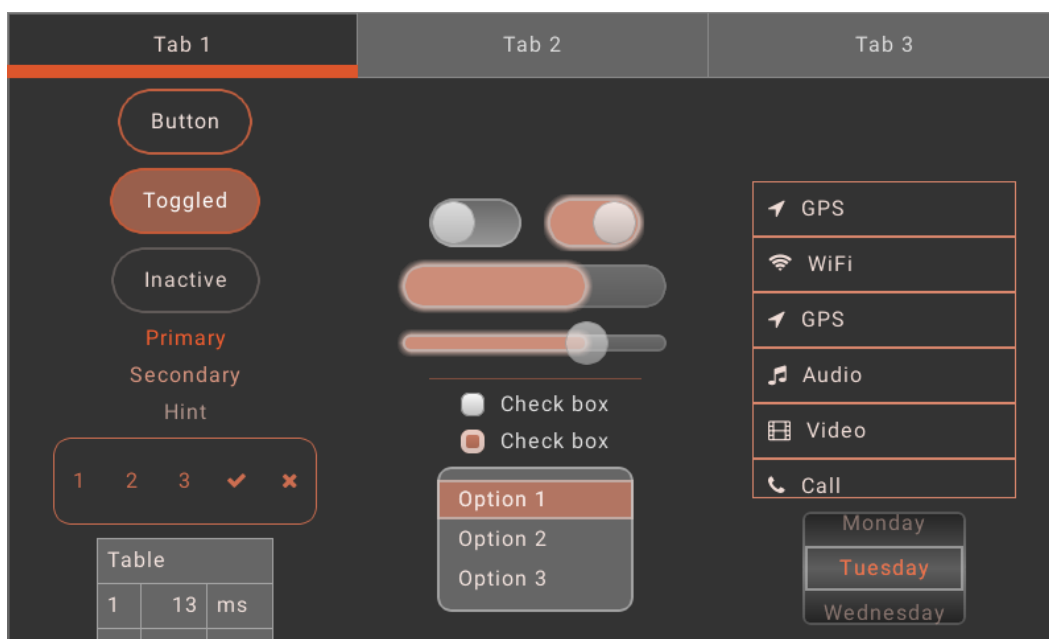


图 1.3 alien 主题



图 1.4 night 主题

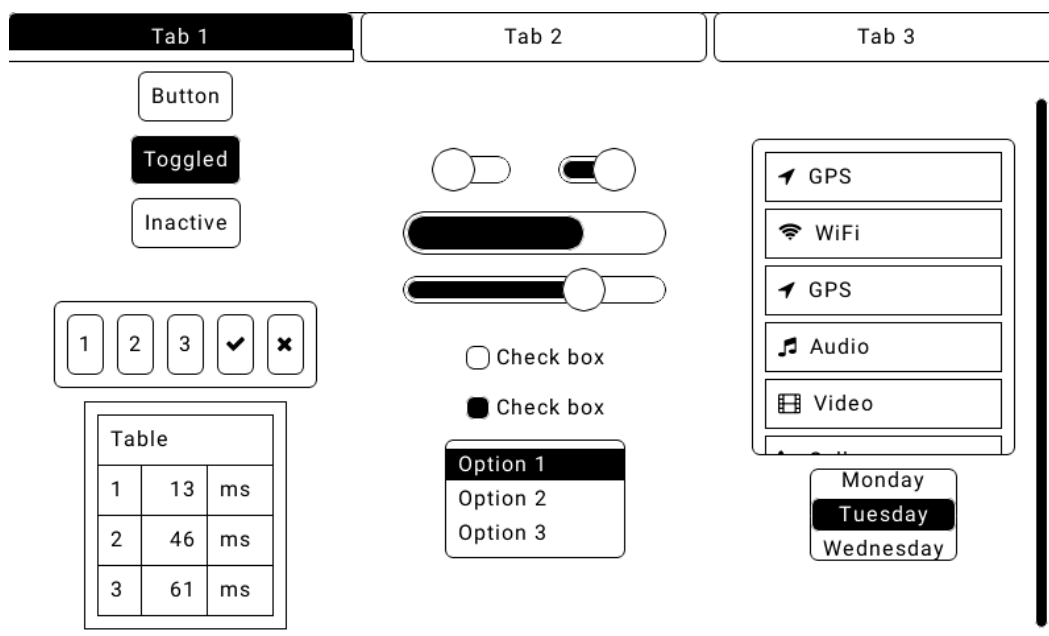


图 1.5 mono 主题

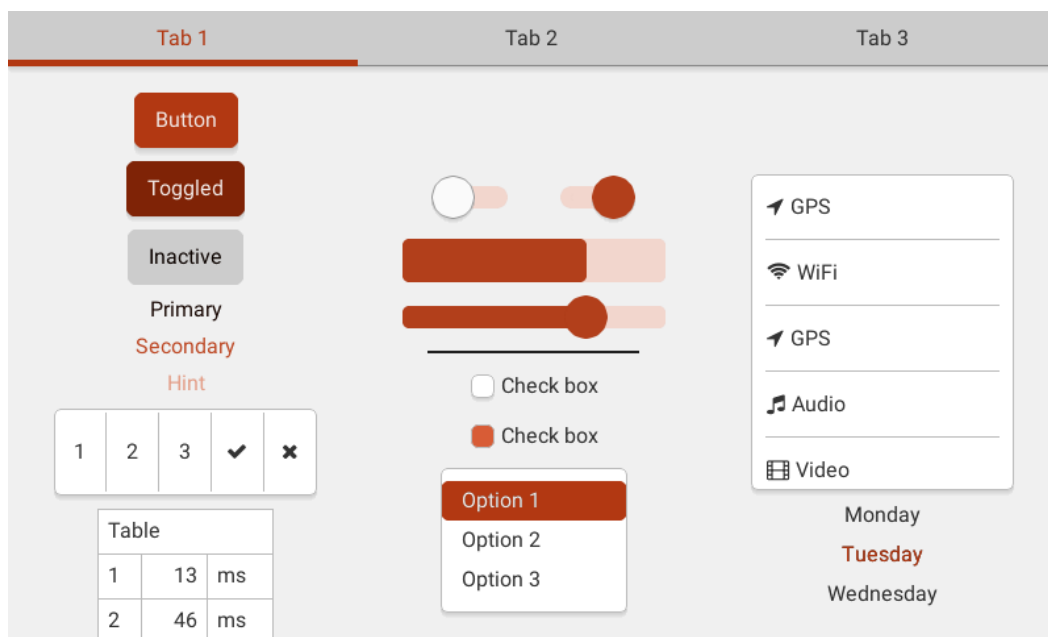


图 1.6 material 主题

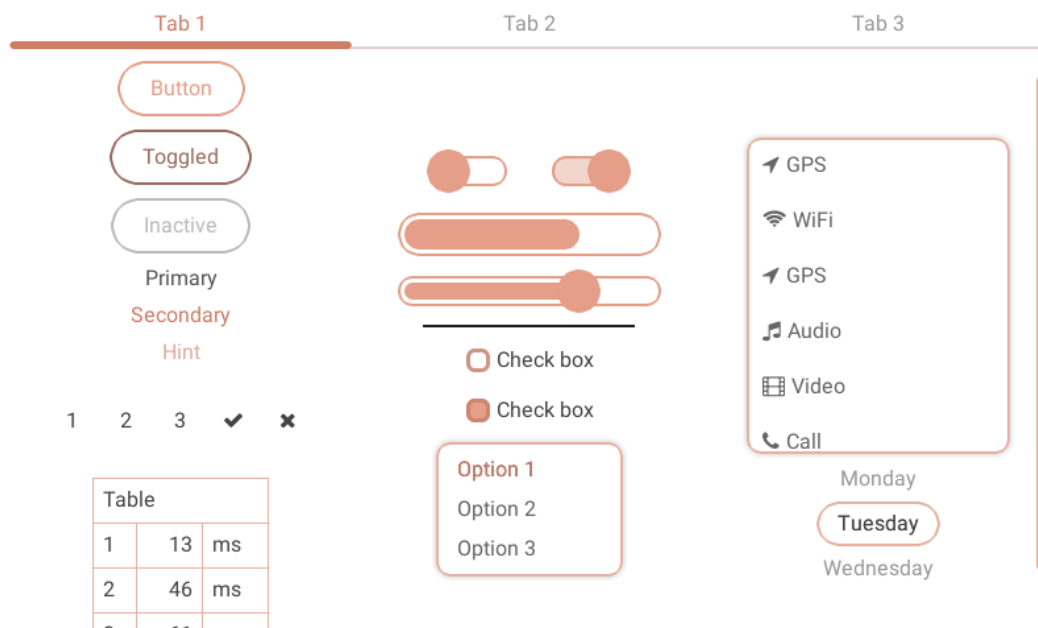


图 1.7 zen 主题



图 1.8 nemo 主题

对于上面的 8 种主题,默认情况下是没有被使能的,你需要在 lv\_conf.h 配置文件中,把你所需要的主题给使能了,而不需要的主题应该继续保持失能状态即可,因为这样可以减少 flash 和 ram 容量的消耗,这 8 种主题在 lv\_conf.h 配置文件中所对应的配置宏分别如下:

```
#define LV_USE_THEME_TEMPL      0    //是否使能 templ 主题
#define LV_USE_THEME_DEFAULT    0    //是否使能 default 主题
#define LV_USE_THEME_ALIEN      0    //是否使能 alien 主题
#define LV_USE_THEME_NIGHT      0    //是否使能 night 主题
#define LV_USE_THEME_MONO       0    //是否使能 mono 主题
#define LV_USE_THEME_MATERIAL   0    //是否使能 material 主题
#define LV_USE_THEME_ZEN        0    //是否使能 zen 主题
#define LV_USE_THEME_NEMO       0    //是否使能 nemo 主题
```

除了上面的 8 个配置宏外,还有一个 LV\_THEME\_LIVE\_UPDATE 配置宏,它是用来定义是否允许在运行时动态切换主题的,如果使能了动态切换主题功能的话,它将会增加 8-10KB 左右的 ram 消耗,如下所示:

```
#define LV_THEME_LIVE_UPDATE    0    //是否允许动态切换主题
```

对于上面的 9 个配置宏,都是 1 代表使能,0 代表失能.

我们已经介绍了 littleVGL 中到底具有多少种主题,那么我们应该如何来使用某个主题呢?这也是很简单的,只需要二个步骤即可,一是先创建主题,二是接着使用此主题,通过如下接口就可以来创建相应的主题:

```
lv_theme_t * theme = lv_theme_templ_init(hue,font);    //创建 templ 主题
lv_theme_t * theme = lv_theme_default_init(hue,font);  //创建 default 主题
lv_theme_t * theme = lv_theme_alien_init(hue,font);    //创建 alien 主题
lv_theme_t * theme = lv_theme_night_init(hue,font);    //创建 night 主题
lv_theme_t * theme = lv_theme_mono_init(hue,font);     //创建 mono 主题
```

```
lv_theme_t * theme = lv_theme_material_init(hue,font); //创建 material 主题
lv_theme_t * theme = lv_theme_zen_init(hue,font);      //创建 zen 主题
lv_theme_t * theme = lv_theme_nemo_init(hue,font);     //创建 nemo 主题
```

通过上面的某个创建主题接口把 theme 主题创建出来之后,就可以接着通过 lv\_theme\_set\_current(theme)接口把此主题给使用起来了.

## 2. lv\_theme 的 API 接口

### 2.1 主要数据类型

#### 3. 主题数据类型

```
typedef struct
{
    struct
    {
        lv_style_t * scr;
        lv_style_t * bg;
        lv_style_t * panel;

#ifdef LV_USE_CONT != 0
        lv_style_t * cont;
#endif

#ifdef LV_USE_BTN != 0
        struct
        {
            lv_style_t * rel;
            lv_style_t * pr;
            lv_style_t * tgl_rel;
            lv_style_t * tgl_pr;
            lv_style_t * ina;
        } btn;
#endif
    ... //因为文本内容过长,此处省略掉其他控件的样式结构体
    } style;

#ifdef LV_USE_GROUP
    struct
    {
        lv_group_style_mod_cb_t style_mod_xcb;
        lv_group_style_mod_cb_t style_mod_edit_xcb;
    } group;
#endif
} lv_theme_t;
```

lv\_theme\_t 是主题所对应的数据类型,它里面就是对所有控件样式的一个封装,了解即可



## 2.2 API 接口

### 2.2.1 创建主题

```
lv_theme_t * lv_theme_tmpl_init(uint16_t hue, lv_font_t * font); //创建 templ 主题
lv_theme_t * lv_theme_default_init(uint16_t hue, lv_font_t * font); //创建 default 主题
lv_theme_t * lv_theme_alien_init(uint16_t hue, lv_font_t * font); //创建 alien 主题
lv_theme_t * lv_theme_night_init(uint16_t hue, lv_font_t * font); //创建 night 主题
lv_theme_t * lv_theme_mono_init(uint16_t hue, lv_font_t * font); //创建 mono 主题
lv_theme_t * lv_theme_material_init(uint16_t hue, lv_font_t * font); //创建 material 主题
lv_theme_t * lv_theme_zen_init(uint16_t hue, lv_font_t * font); //创建 zen 主题
lv_theme_t * lv_theme_nemo_init(uint16_t hue, lv_font_t * font); //创建 nemo 主题
```

#### 参数:

hue: HSV 颜色格式,用来定义此主题的基色,范围为[0,360]

font: 此主题所使用的默认字体,如果传 NULL 的话,则代表使用 lv\_conf.h 文件中的  
LV\_FONT\_DEFAULT 宏所定义的默认字体

#### 返回值:

返回创建出来的主题

### 2.2.2 使用主题

```
void lv_theme_set_current(lv_theme_t * th);
```

#### 参数:

th: 想要被使用的主题

### 2.2.3 获取当前正在被使用的主题

```
lv_theme_t * lv_theme_get_current(void);
```

#### 返回值:

返回当前正在被使用的主题

## 3. 例程设计

### 3.1 功能简介

创建一个 night 主题,并且使用此 night 主题,然后我们在屏幕上创建一个 tabview 选项卡控件来观察主题的界面效果,此 tabview 选项卡控件上有 tab1 和 tab2 俩个子选项卡,我们往 tab1 选项卡上添加了按钮,矩阵按钮,开关,进度条,滑块,下拉列表框等 6 种控件,往 tab2 选项卡上添加了列表,滚轮等 2 种控件.

### 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\_theme\_test.c 和 lv\_theme\_test.h 俩个文件,其中 lv\_theme\_test.c 文件的内容如下:

```
#include "lv_theme_test.h"
#include "lvgl.h"

const char * const btnm_str[] = {"1", "2", "3", LV_SYMBOL_OK, LV_SYMBOL_CLOSE,
""};

//tab1 选项卡初始化
void tab1_init(lv_obj_t * parent)
{
    //1.设置页面的布局方式
    lv_page_set_scrl_layout(parent, LV_LAYOUT_PRETTY);

    //2.创建一个按钮
    lv_obj_t * btn = lv_btn_create(parent, NULL);
    lv_obj_t * btn_label = lv_label_create(btn, NULL);
    lv_label_set_text(btn_label, "Button");

    //3.创建一个矩阵按钮
    lv_obj_t * btnm = lv_btnm_create(parent, NULL);
    lv_obj_set_size(btnm, LV_HOR_RES_MAX/4, 2*LV_DPI/3);
```

```
lv_btmn_set_map(btmn,(const char **)btmn_str);
lv_btmn_set_btn_ctrl_all(btmn,LV_BTNUM_CTRL_TGL_ENABLE);
lv_btmn_set_one_toggle(btmn,true);

//4.创建一个开关
lv_sw_create(parent,NULL);

//5.创建一个进度条
lv_obj_t * bar = lv_bar_create(parent,NULL);
lv_bar_set_value(bar,70,false);

//6.创建一个滑块
lv_obj_t * slider = lv_slider_create(parent,NULL);
lv_slider_set_value(slider,50,false);

//7.创建一个下拉列表框
lv_obj_t * ddlist = lv_ddlist_create(parent,NULL);
//为右侧的箭头腾出点空间
lv_ddlist_set_fix_width(ddlist,lv_obj_get_width(ddlist)+LV_DPI/2);
lv_ddlist_set_draw_arrow(ddlist,true);
}

//tab2 选项卡初始化
void tab2_init(lv_obj_t * parent)
{
    //1.设置页面的布局方式
    lv_page_set_scrl_layout(parent,LV_LAYOUT_PRETTY);

    //2.创建一个列表
    lv_obj_t * list = lv_list_create(parent,NULL);
    lv_obj_set_size(list,LV_HOR_RES_MAX/4,LV_VER_RES_MAX/2);
    lv_list_add_btn(list,LV_SYMBOL_VIDEO,"Video");
    lv_list_add_btn(list,LV_SYMBOL_CALL,"Call");
    lv_list_add_btn(list,LV_SYMBOL_BELL,"Bell");
    lv_list_add_btn(list,LV_SYMBOL_FILE,"File");
    lv_list_add_btn(list,LV_SYMBOL_EDIT,"Edit");
    lv_list_add_btn(list,LV_SYMBOL_CUT,"Cut");
    lv_list_add_btn(list,LV_SYMBOL_COPY,"Copy");

    //3.创建一个滚轮
    lv_obj_t * roller = lv_roller_create(parent,NULL);

    lv_roller_set_options(roller,"Monday\nTuesday\nWednesday\nThursday\nFriday\nSaturday\nSunday",true);
```

```
lv_roller_set_selected(roller,1,false);
lv_roller_set_visible_row_count(roller,3);
}

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

    //1.使用 night 主题,当然了,你也可以换成其他的主题
    lv_theme_t * theme = lv_theme_night_init(210,NULL);//创建主题
    lv_theme_set_current(theme);//使用主题

    //2.在屏幕上创建一些控件来查看主题的界面效果
    lv_obj_t * tabview = lv_tabview_create(scr,NULL);
    //2.1 创建 tab1 选项卡
    lv_obj_t * tab1 = lv_tabview_add_tab(tabview,"Tab 1");
    tab1_init(tab1);
    //2.2 创建 tab2 选项卡
    lv_obj_t * tab2 = lv_tabview_add_tab(tabview,"Tab 2");
    tab2_init(tab2);
}
```

### 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:

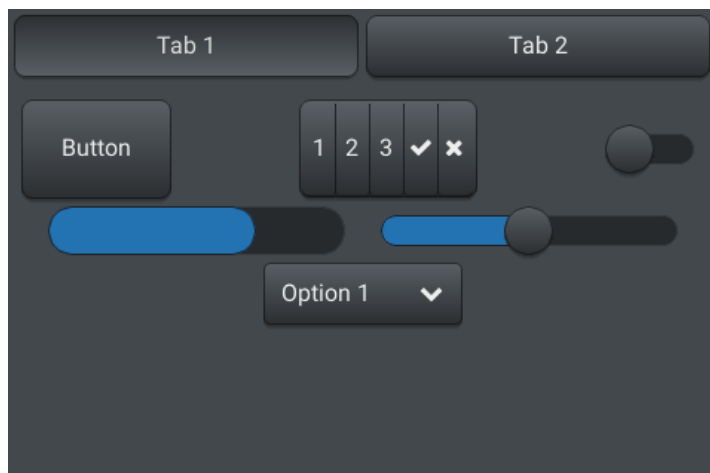


图 3.4.1 tab1 选项卡的初始界面效果(night 主题)

然后我们点击 tab2 选项卡按钮,可以看到 tab2 选项卡的界面如下所示:

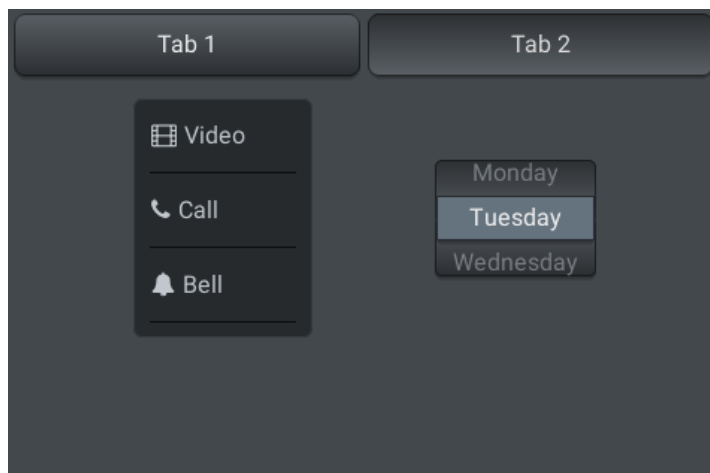


图 3.4.2 tab2 选项卡的初始界面效果(night 主题)

## 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