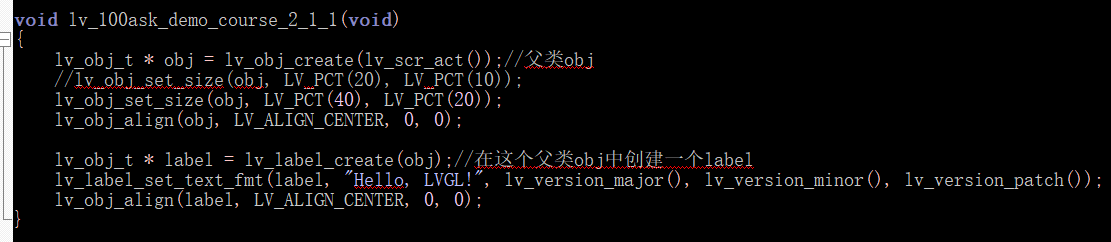
LVGL库函数

**lv\_obj\_t 基类**



屏幕是没有父类的基础对象（NULL）



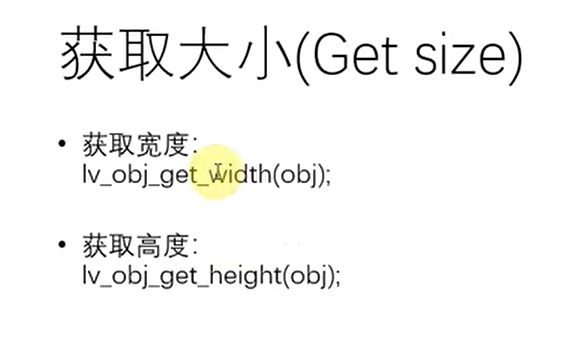
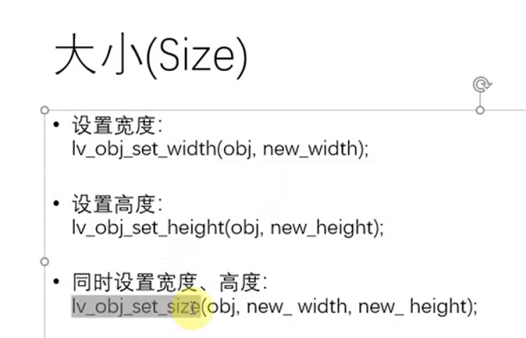
其余对象都在屏幕这个父类里定义，并可以嵌套。

屏幕层级：



用户层显示基本内容，顶层显示弹窗，系统层处理系统事务，如（鼠标）

**对象size**

返回int

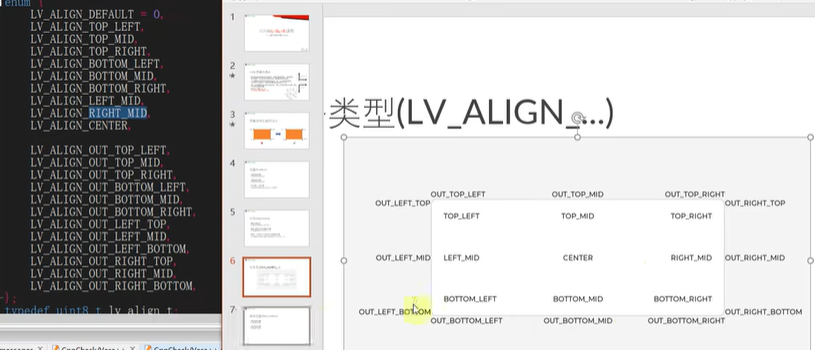


如果size大于屏幕，屏幕会变为可滚动的

**对象位置**

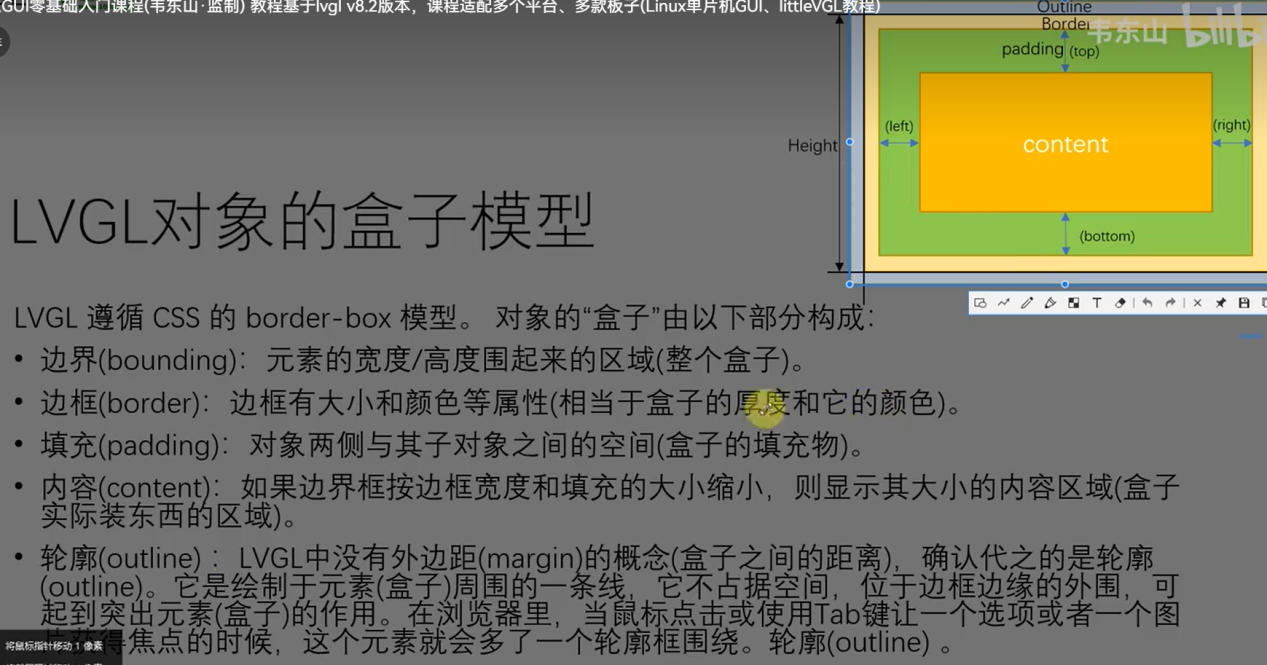


注意，参照对齐后，设置的xy是xy偏离参考对象左上的距离



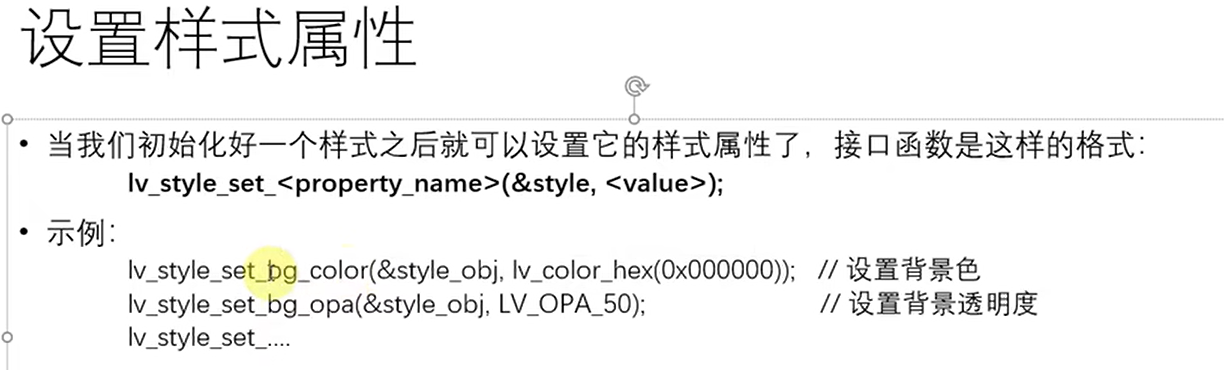


**盒子模型**

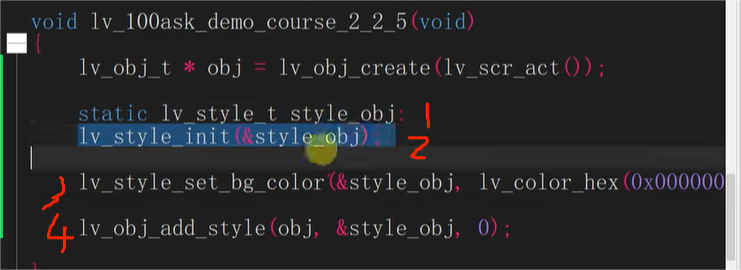


Outline 可以在框出**选中**的东西

**对象样式style（很类似html）**

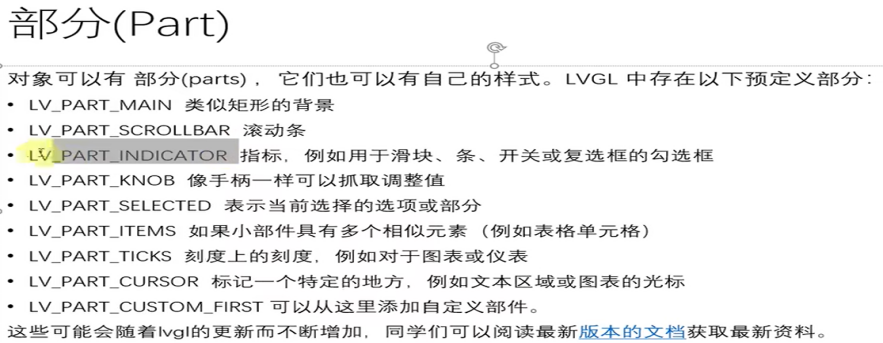


步骤

lv\_style\_t是一个临时变量，用于传参（可以全局声明共用）类似stm32的gpio结构体作用

Add\_style的最后一个参数，0，是一个状态（部分）参量，比如规定0是为按下的样式，1是按下的样式





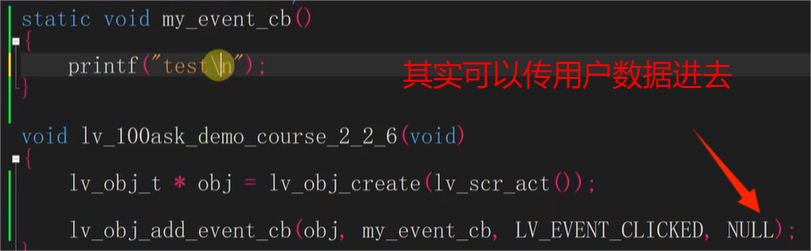
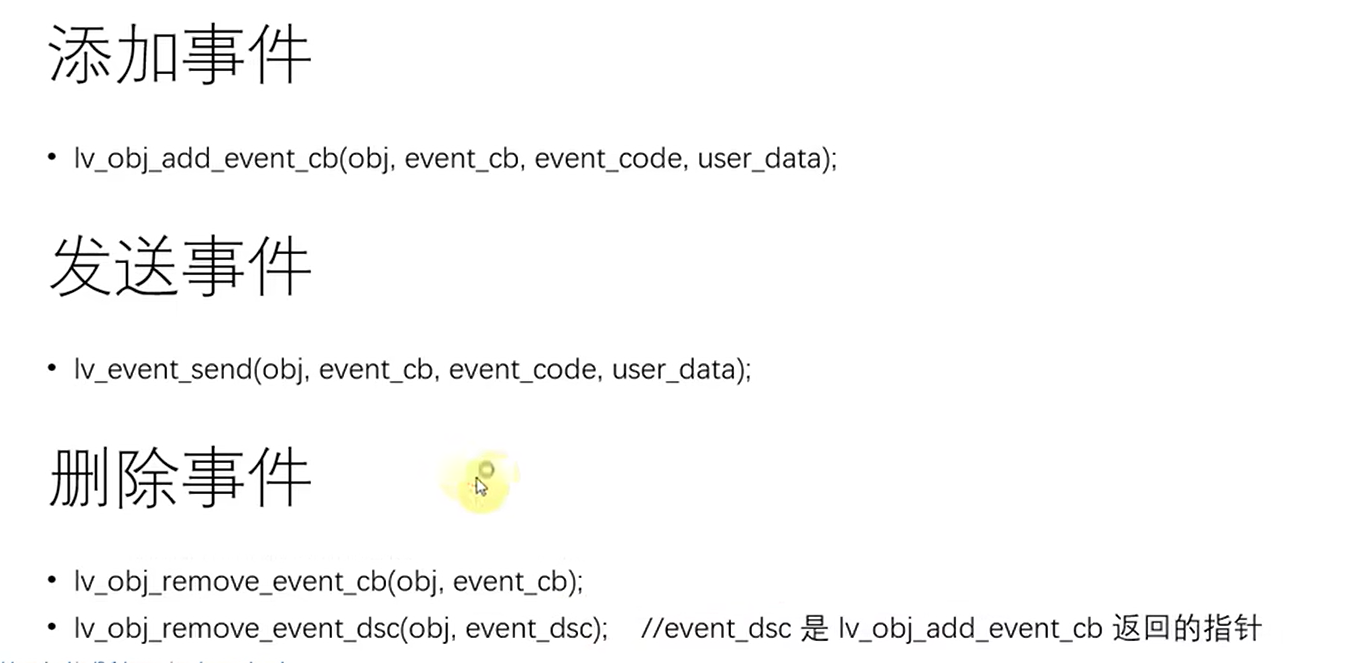
可以组合使用用|分隔

不想用结构体可以用本地样式来设置

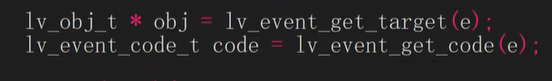


本地样式优先级高，不会被结构体覆盖改变

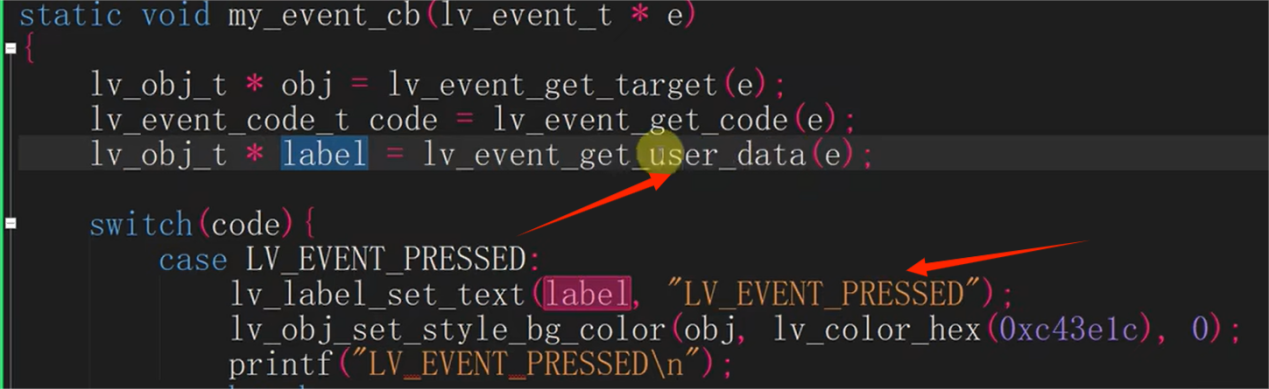
**对象事件**



回调函数中可以获取发生事件的对象、发生的事件的代码、、、、作判断

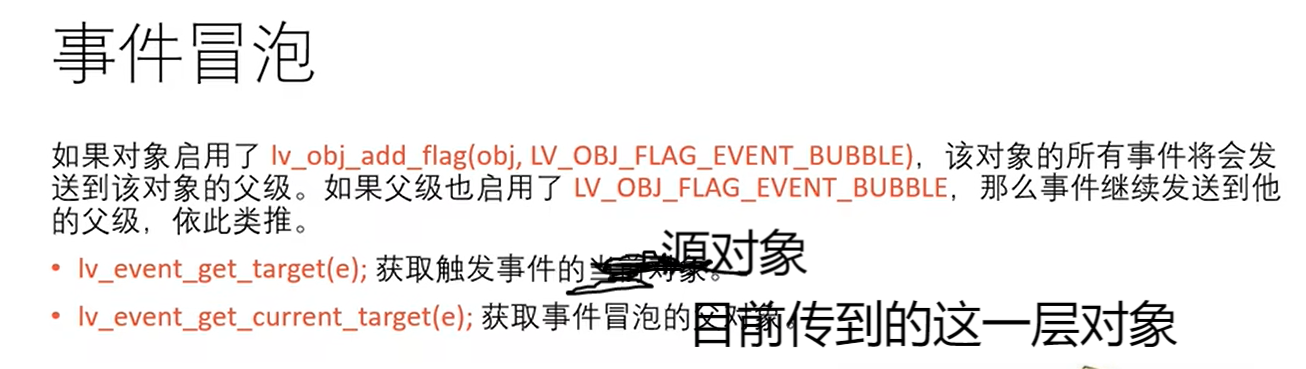


假设userdata传入了一个label，则cb中获取后可以直接修改label的值



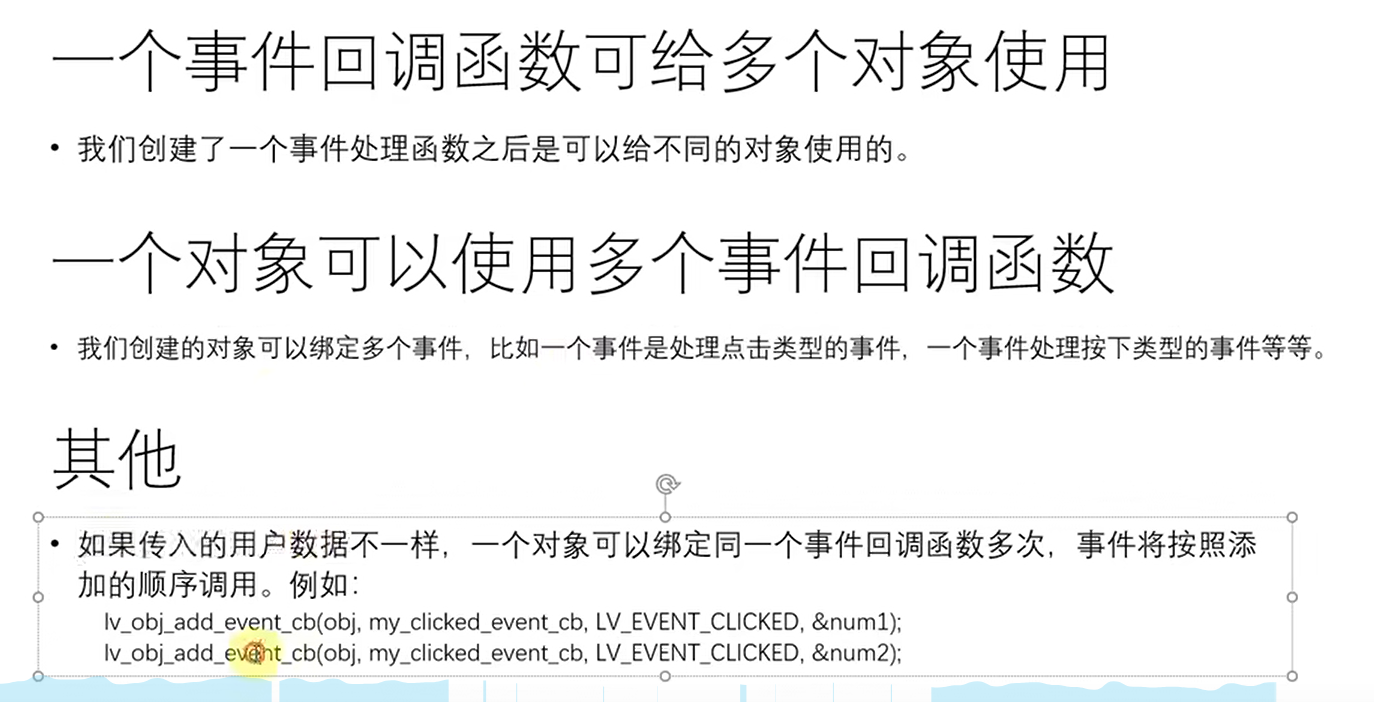
发送事件用于作组件间传递事件

**事件冒泡**



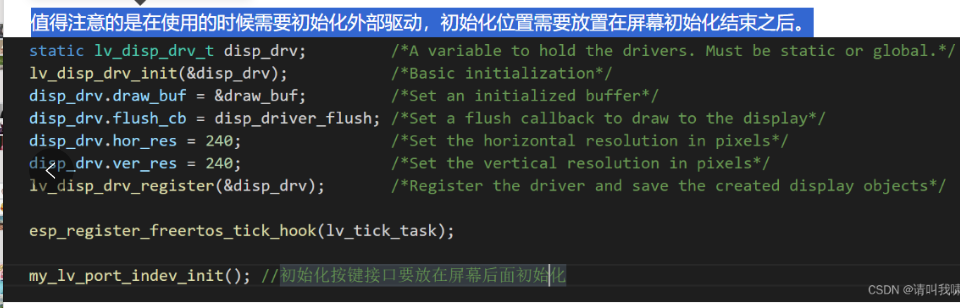
在回调函数中，获取到父对象或触发源对象后，就可以对它们作样式改变之类

同时由于事件e是直接上传的（保持不变），所以父类的回调函数如果有处理这个事件的内容，也会因为子对象的触发而开始运行



添加外部输入设备：

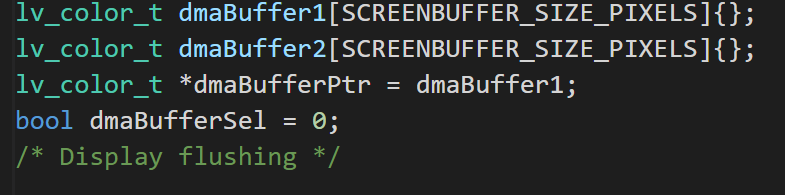
<https://blog.csdn.net/HuangBJ200/article/details/136807387>



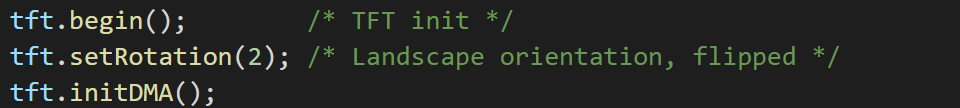
**由于屏幕转了180度，而触屏坐标不变**

**所以，如果用button模拟点击，x=240-x0;y=300-y0**

Dma尝试：（记得开启color swap）





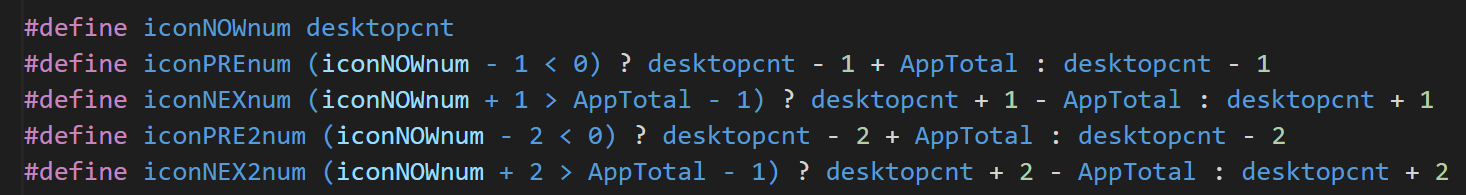


数组存储lv\_img方便遍历调用不同图片：



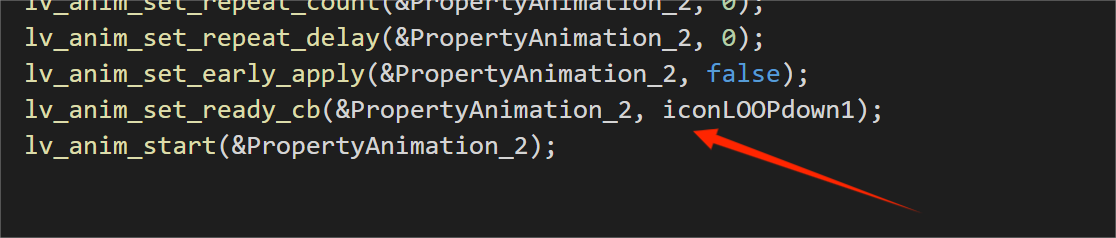
自适应循环桌面：

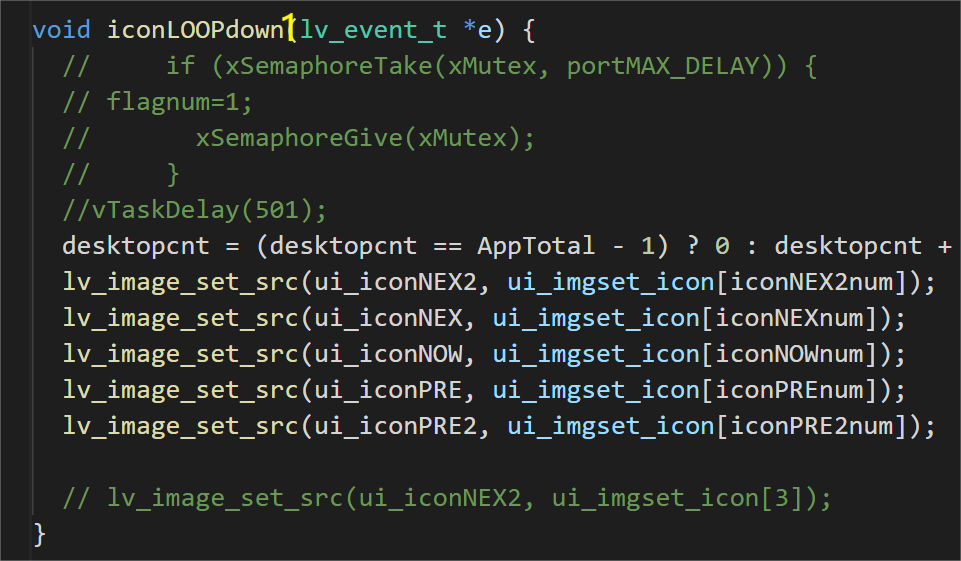
数组存储lv\_img



开启动画

加入动画结束回调函数来刷新新一轮图标





APP的规范

1. **前台GUI，**由Squareline Studio制作，导出UI\_APP.c/.h文件，并将ui逻辑、动画放置到ui.c里。然后将icon加入到ui\_imgset\_icon[]中，调整Apptotal的数值。即可实现自适应的转轮桌面

1. **后台逻辑，**采用freeRTOS，

void Rtctaskcreate() {

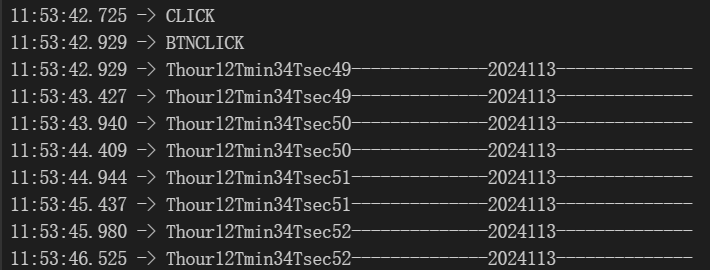
xTaskCreatePinnedToCore(Rtctask, "Rtc\_task", 4096, NULL, 2, &Rtc\_task, 0);

}

void Rtctaskdelete() {

vTaskDelete(Rtc\_task);

}



并在ui逻辑中调用即可。注意进入时创建，退出时删除。

*互斥量要不要？暂时不需要，直接改数据值即可正常显示*

1. c++库如何被ui.c调用——extern “C”修饰

