**前言**

使用GUI GUIDER来创建界面效率很高，且非常直观，推荐使用！

声明：文章只记录本人的使用及学习经历，在此过程中参考了大量的别人的文章，表示感谢。本人水平有限，难免有不合理之处，分享也是为网络做点贡献，如果对你有帮助那就更好了。

# 一、准备工作

准备一个可用屏幕显示的工程

文中移植基于正点原子精英版的示例工程。

安装好GUI GUIDER

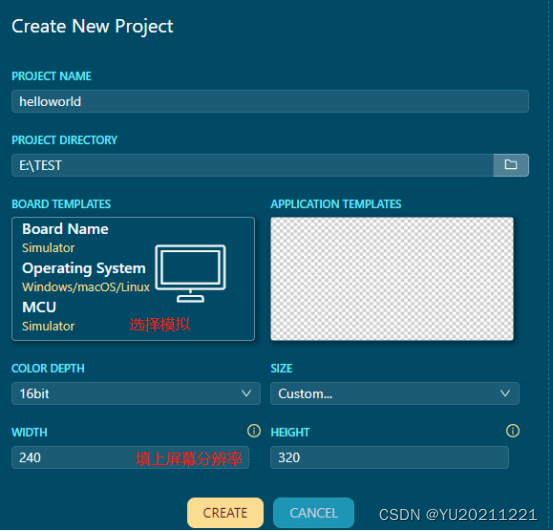
链接：https://pan.baidu.com/s/1iavFlsh\_h5KZjQTbZWCH1w

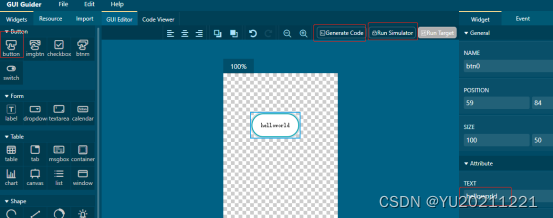
提取码：6666

安装很简单，这里就不详述了。

# 二、创建一个简单的工程

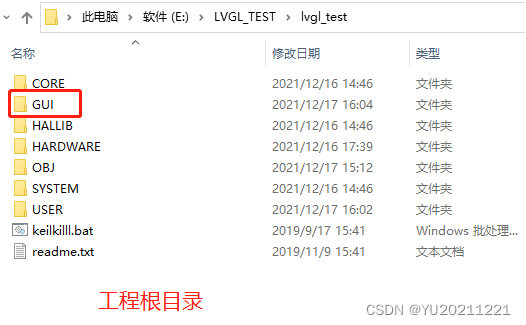
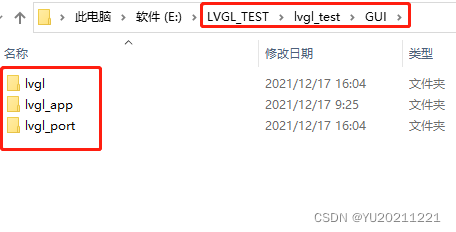
路径最好不要有中文



拖一个按钮  
  
可以启动模拟器看看效果

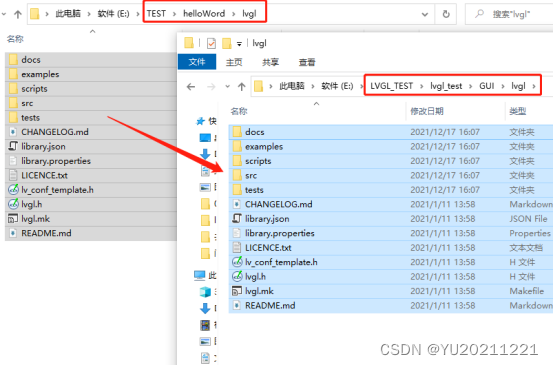
# 三、添加文件

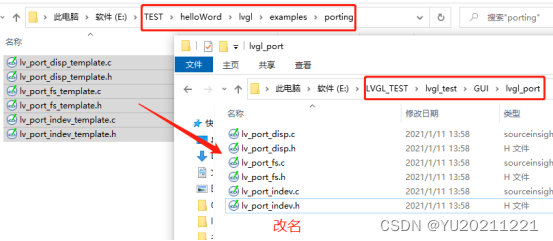
## 1.创建路径

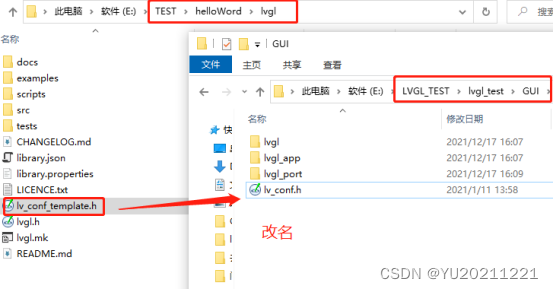
stm32工程新建GUI文件夹（名字任意，看你喜欢）  
  
GUI文件夹内再新建如图文件夹  


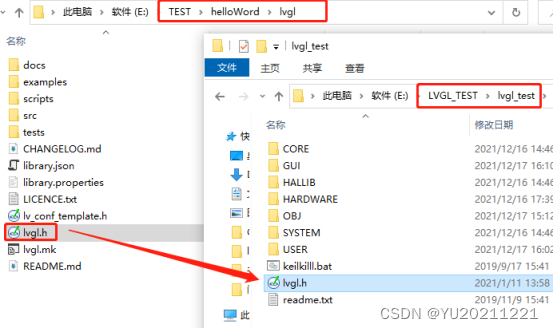
## 2.复制gui生成的文件到stm32工程中

文件较多，要仔细

复制LVGL源文件（图中左边目录是gui工程的路径，右边是stm32工程的路径）  


应用文件  
  
移植文件（改不改名无所谓，看个人喜欢）  
  
配置文件

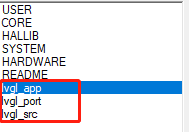


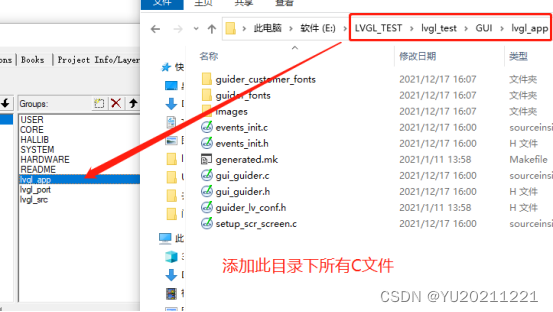
头文件  
  
终于复制完了，开始修改工程啦！

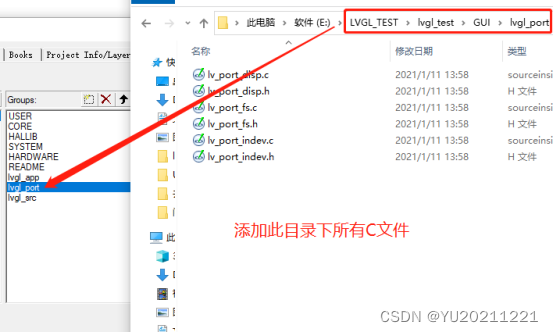
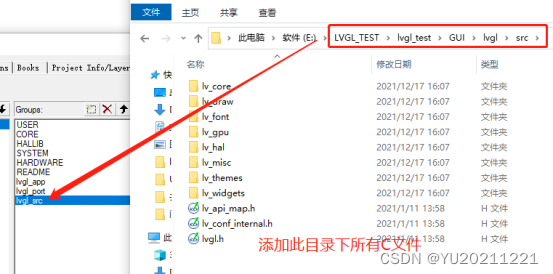
# 四、修改stm32工程实现界面显示

同样修改的地方很多要仔细

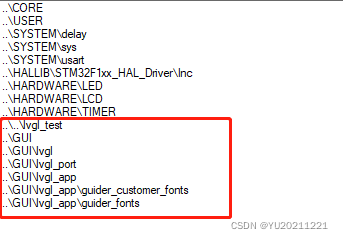
## 1.添加C文件路径

创建3个文件夹  


添加应用文件  


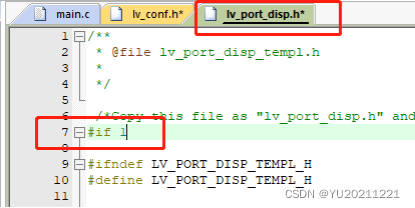
添加移植文件  
  
添加源文件  
（当然也可根据所需添加，但是为了避免出错或懒得整理直接全部添加就行了）  


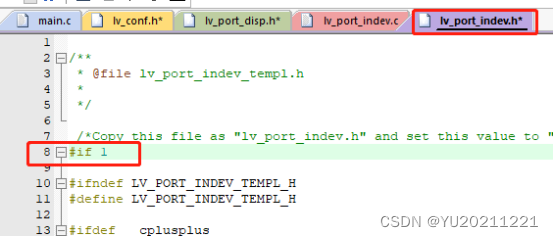
## 2.添加头文件路径

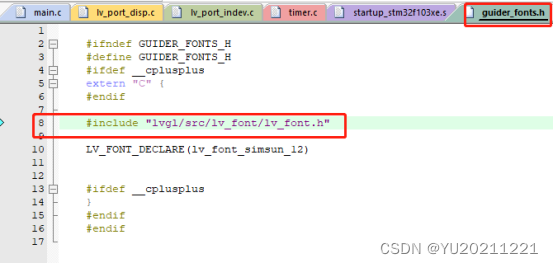


## 3.修改头文件





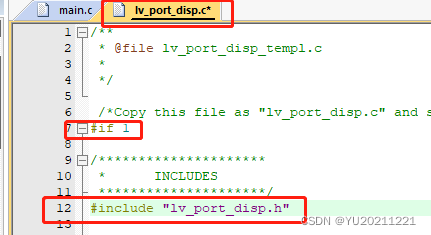


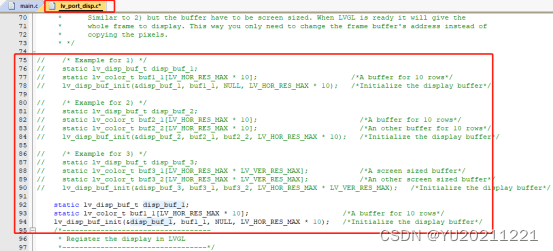


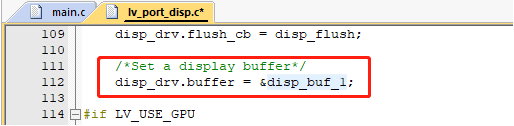
不建议直接这样添加路径，还是直接在c/c++中直接添加其路径就好了

## 4.修改源文件

启用该文件

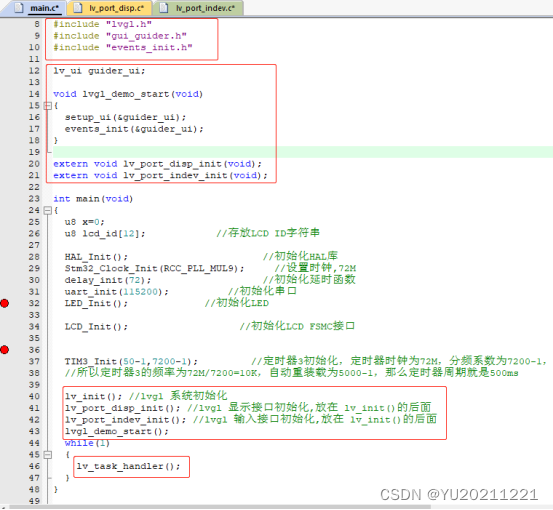


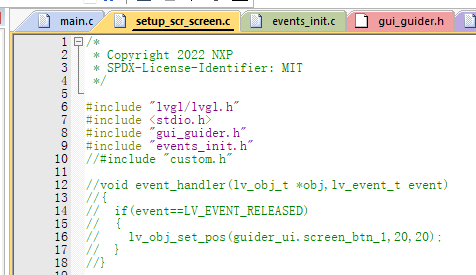
选择一个实例，如果内存够大可以将缓存改大点或用双缓存，提高刷屏速度。  




填充屏幕刷新绘制  
  
启用触摸等输入  

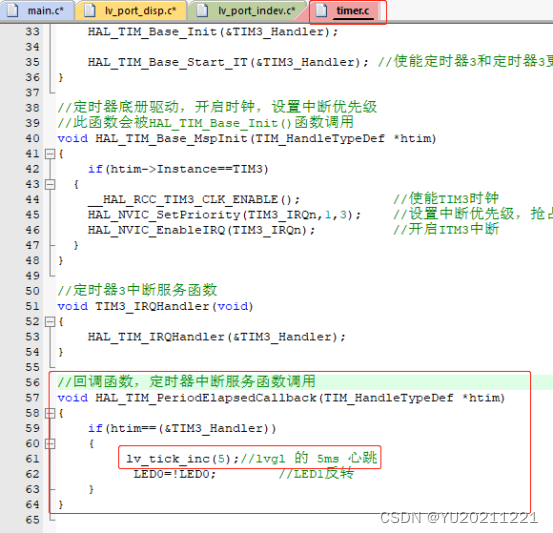

## 5.添加应用到主程序



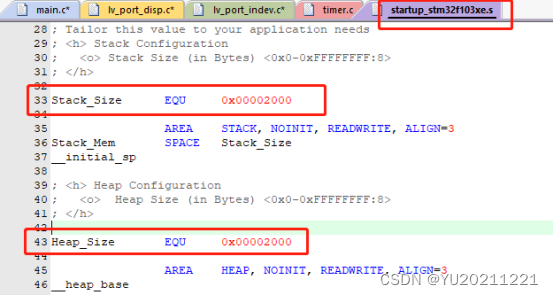


将#include“custom.h”注释

## 5.添加lvgl“心跳”

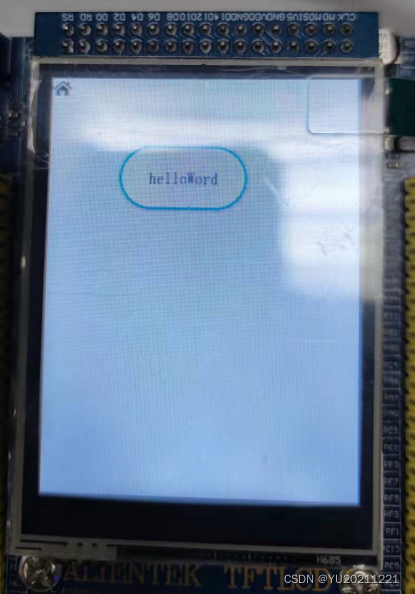
可以有很多种方式，这里简单点用了一个定时器。  


## 6.修改堆栈

  
至此大功告成

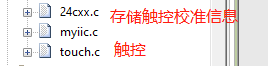
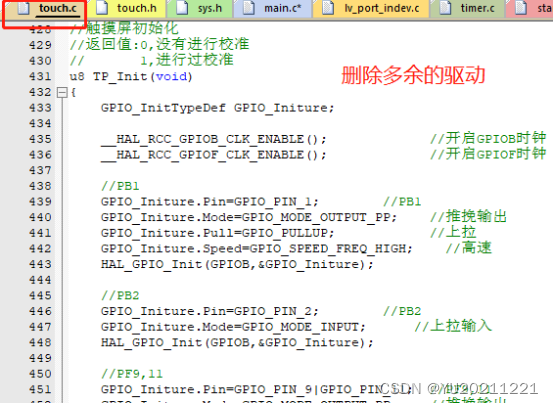
## 7.编译下载

若编译出错，仔细看看是不是缺或错步骤了。

看效果是不是还挺不错  
  
显示部分到此就结束了，后面说说事件的响应。

# 五、添加事件实现对触控的响应

## 1.移植触摸驱动

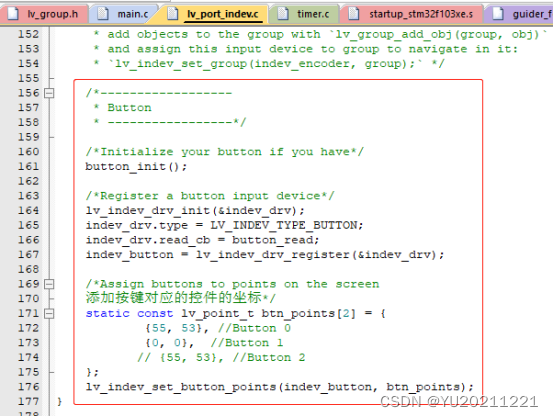
触摸接口一般是iic或spi，以正点原子的触摸为例，只需要这三个文件  
  


## 2.触控移植到lvgl

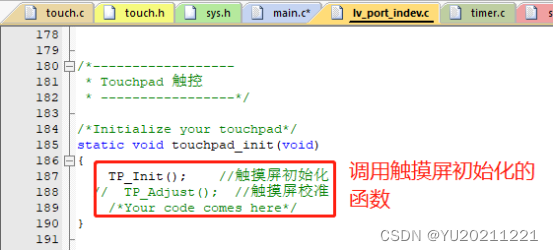
观察lv\_port\_indev.c文件会发现触控的框架已经搭建的很完善了，我们要做的就只是将触摸屏的坐标数据给到lvgl。框架中包含了触控、鼠标、键盘、按钮、编码器，我们只用常见的触控和按钮就好。

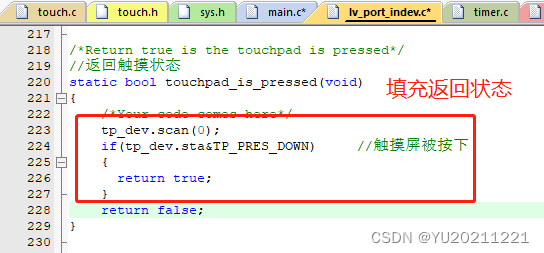
lv\_port\_indev\_init 函数中只保留touchpad和button相关的，其余注释掉，如下图：

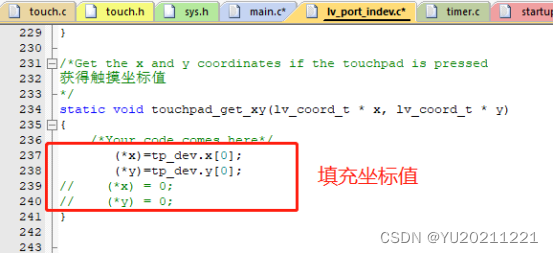
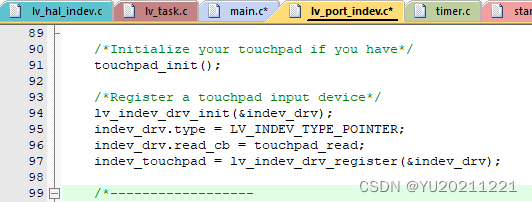




添加驱动

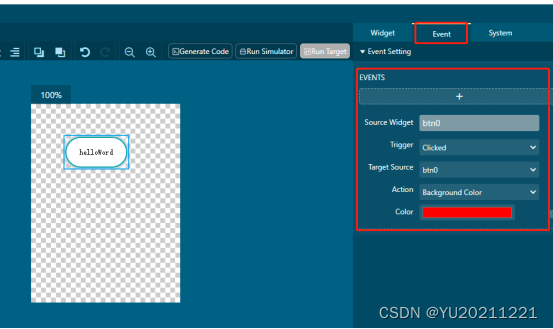




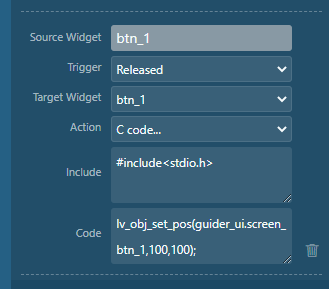
  
到这里就移植好了驱动了，在初始化调用lv\_port\_indev\_init后，触摸驱动会被注册到LVGL  
  
注册到LVGL其实就是创建了一个受lvgl控制的任务，可以看做一个线程，而任务的时基由调用函数lv\_tick\_inc()实现。

## 3.添加触控事件

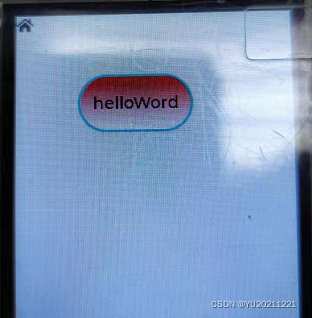
再次打开gui guider 添加如图的事件，重新生成代码，将events\_init.c文件替换到工程



C事件添加



再稍作填充即可

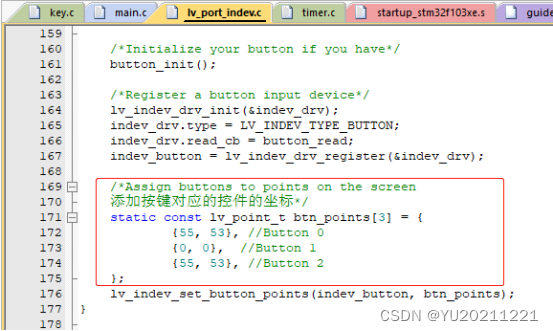


这样就完成了按钮按下的事件绑定，至于如何将坐标映射到具体按钮的过程就不需要我们关心了，lvgl已经帮我们实现了，我们做的就只是绑定到控件，非常方便。

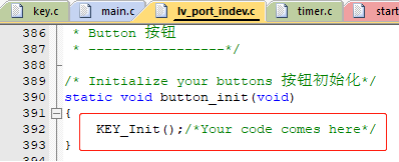
再试试，按下按钮后发现能变红色了。

# 六、添加事件实现对按钮的响应

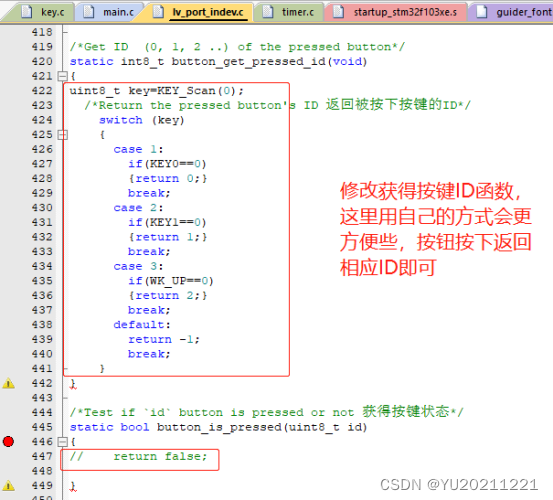
## 1.添加按键对应的控件的坐标



## 2.添加按钮初始化

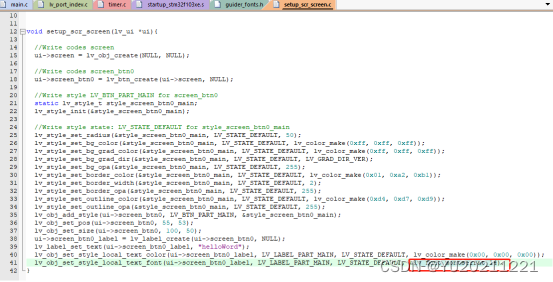


## 3.修改按键返回ID

  
这样就可以啦，按键是很灵活的，其实不一定要按照框架来实现，完全可以自己实现在按键按下的时候去调用某个事件会更快更直接。

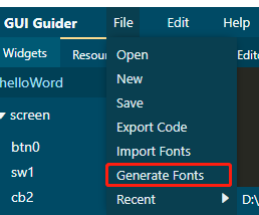
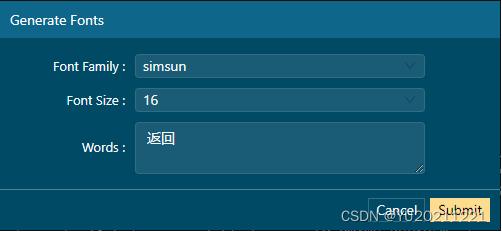
# 七、其他

## 1.修改字体

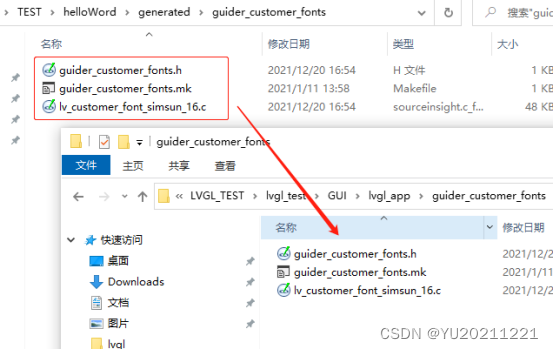
修改调用即可  
  
界面设计更新后直接再将生成代码的generated文件夹内容替换到lvgl\_app文件夹，再稍微修改即可

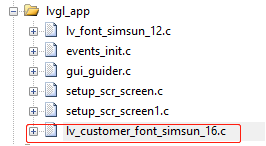
## 2.中文显示

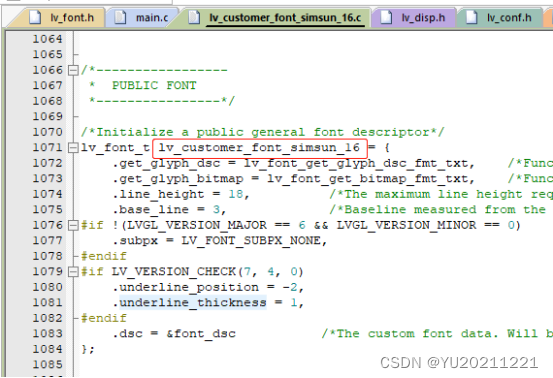
### 1）.gui guider生成字体

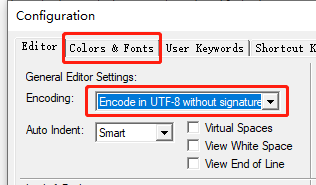
### 2）.字体添加到工程

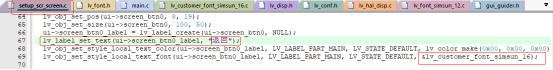
复制文件  


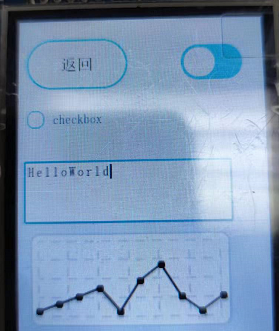
添加到工程目录  
  
添加声明，名称和字体结构体名称一致  

修改字符编码为utf-8（不修改的话编译出来是无效的）

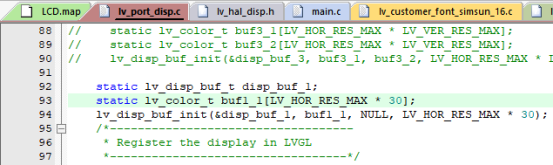
  
在控件生成函数中修改显示文字及字体

  
编译下载则可显示中文了



## 3.加快刷屏速度

提高缓冲区大小或使用双缓存



## 后话

虽然lvgl宣称占用资源极少，但是对于空间紧张的单片机来说还是占用太多了，尤其是内存，稍微用复杂的控件就得分配大块的内存，否则在运行几次之后就会卡住或者直接就没法显示。如果要显示好看的复杂的界面还是得外扩内存及flash，用上更好的CPU才行，否则还是用基本的绘图吧。

另外，gui guider目前也有不足的点，生成代码稍微有点慢，不支持最新版本的库。