

## 正点原子 littleVGL 开发指南

PC 模拟器的使用

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

# PC 模拟器的使用

## 1. 介绍

PC 模拟器的好处就是在于当我们没有实际的硬件开发板时, 我们依然可以测试和调试 GUI 系统, 另外 PC 模拟器对我们的前期界面迭代和后期的产品说明书制作都是可以起到非常大的作用, 作为一款优秀的 GUI 库, littleVGL 也肯定是支持模拟器功能的, 模拟器配合不同的 IDE(Integrated Development Environments) 软件可以运行在不同的操作系统上, 比如 Windows, Linux or OSX, 下面是 littleVGL 所支持的 IDE 软件种类.






Eclipse	CodeBlocks	Visual Studio	PlatformIO	Qt Creator
				
Cross-platform with SDL	Native Windows	Cross-platform with SDL	Cross-platform with SDL	Cross-platform with SDL

图 1.1 littleVGL 所支持的 IDE

其中 littleVGL 官方文档上主要介绍是 Eclipse CDT, 虽然 Eclipse 是跨平台的, 但在笔者亲身体验一番之后, 发现 Eclipse 的环境搭建还是比较麻烦的, 考虑到大部分初学者的感受, 我这里准备打算采用 CodeBlocks 来实现模拟器, 相比 Eclipse 而言, 去掉了很多不必要的步骤, 如果有同学还是执意想用 Eclipse 的话, 请单独联系笔者, 下面让我们正式开始讲述一下 CodeBlocks 模拟器的环境搭建.

## 2.环境搭建

### 2.1 材料准备

俗话说的好,工欲善其事,必先利其器,我们先提前把所需要的材料给全部准备好.所需要的材料如下:

1. CodeBlocks 17.12,下载到的文件名为 codeblocks-17.12mingw-setup.exe
2. SDL2,下载到的文件名为 SDL2-devel-2.0.10-mingw.tar.gz
3. littleVGL 模拟器库,下载到的文件名为 lv\_pc\_simulator.zip

上面所说的材料.笔者已经为大家全部准备好了,可以省去再次下载的过程,但是这个材料的获取过程我还是有必要给大家讲一下的.笔者的搭建环境为 Win10 64 位,建议同学们最好都使用 Windows 系统的电脑来搭建,不管是 Win7 还是 Win8 等,操作步骤是一样的.

#### 2.1.1 CodeBlocks 下载

下载链接为: <http://www.codeblocks.org/downloads/26>

打开链接之后,选择下载 codeblocks-17.12mingw-setup.exe 文件,如下图所示

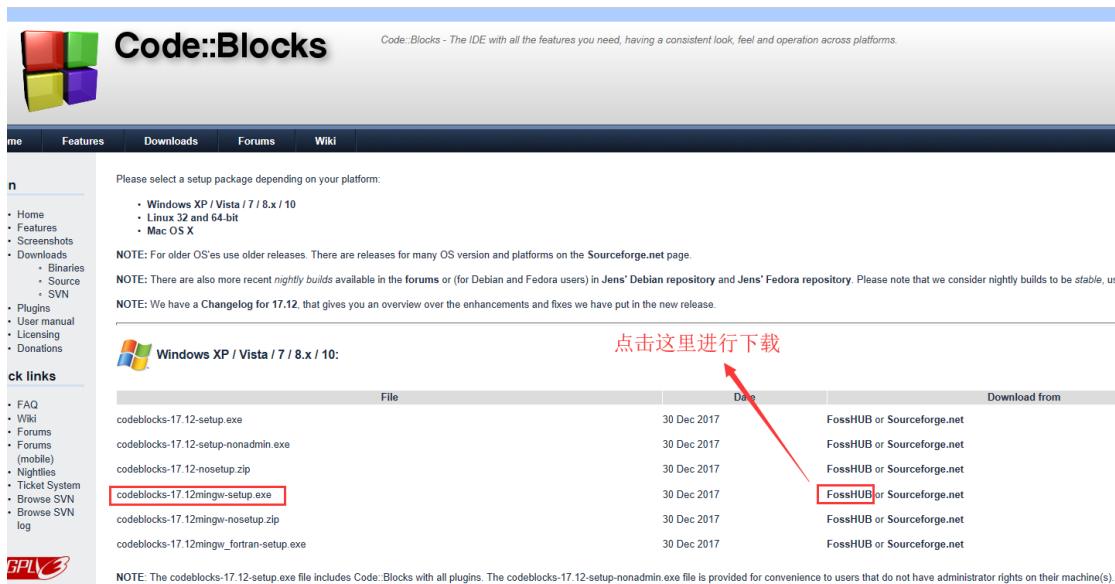


图 2.1.1.1 codeblocks 下载示意

## 2.1.2 SDL2 下载

下载链接为:<http://www.libsdl.org/download-2.0.php>

打开链接之后,选择下载 SDL2-devel-2.0.10-mingw.tar.gz 文件,如下图所示



图 2.1.2.1 SDL2 下载示意

## 2.1.3 littleVGL 模拟器库下载

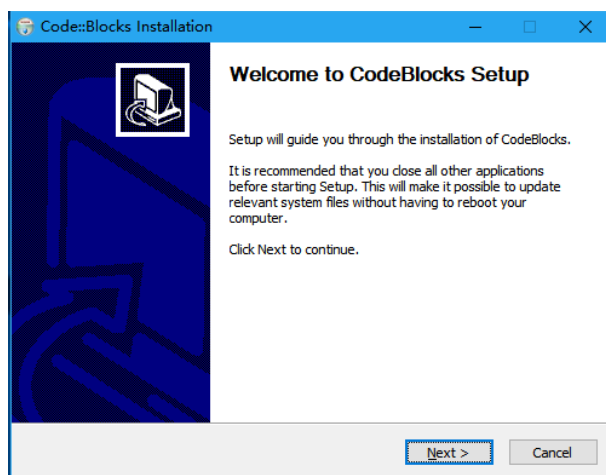
下载链接为: [https://littlevgl.com/download/lv\\_pc\\_simulator.zip](https://littlevgl.com/download/lv_pc_simulator.zip)

这个很简单,复制链接地址到浏览器直接进行下载即可

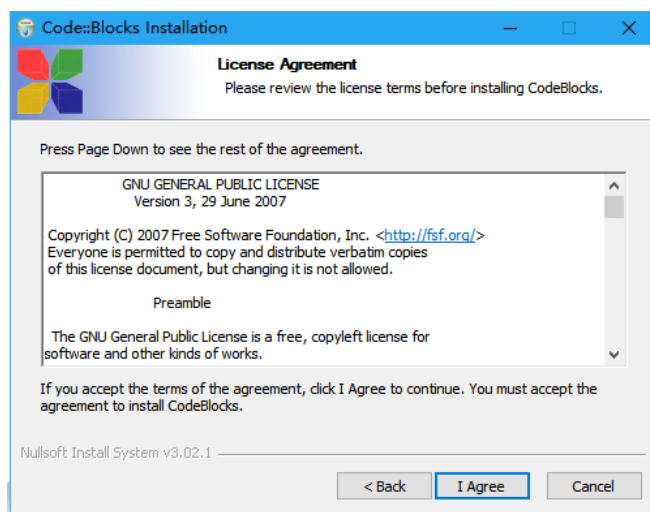
## 2.2 软件安装和配置

其实这里只需要安装 CodeBlocks 这一个软件就可以了,安装过程也很简单,就是一路 Next 的操作,我这里准备把其安装到我的 D:\baseSoftware 目录下。

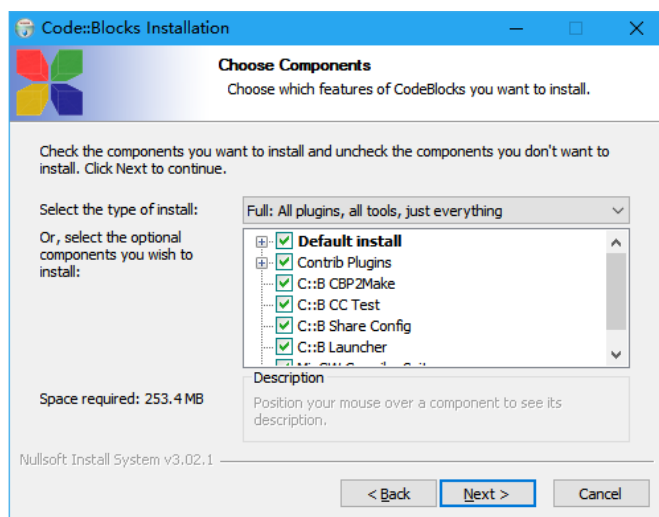
直接双击 codeblocks-17.12mingw-setup.exe 文件,弹出如下对话框:



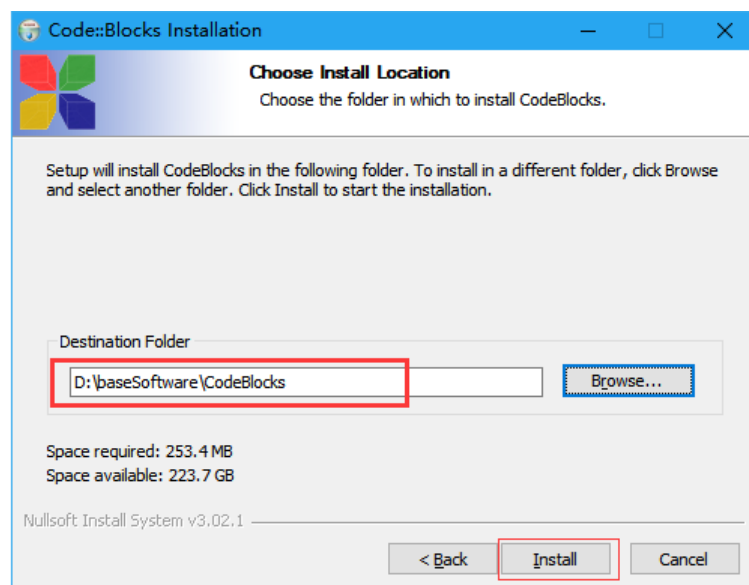
直接点击 Next,接着进入到如下界面:



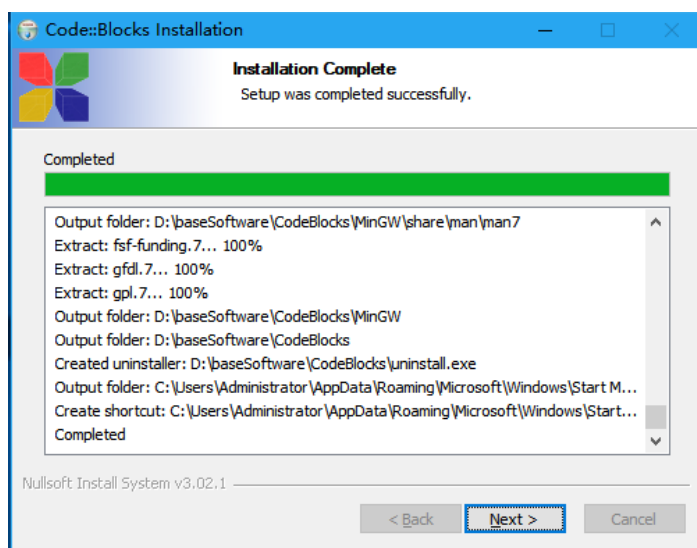
直接点击 I Agree 按钮即可,接着进入到如下界面:



什么都不要去动,直接点击 Next 按钮,接着进入到如下界面



在这里,请根据自己的需求,选择合适的安装路径,选择好之后,直接点击 **Install** 按钮进行安装,接下来就是等此软件安装完成,在安装过程中,会弹出对话框询问我们,是否立即启动 CodeBlocks,我们先选择否即可,最后软件安装完成之后,如下图所示:



接着点击 **Next** 按钮,最后再点击 **Finish** 按钮结束安装.

接着我们需要把 `lv_pc_simulator.zip` 解压缩到一个英文目录下,我这里先在桌面上新建一个 `lv_pc_simulator` 目录,然后把 `lv_pc_simulator.zip` 里的所有文件解压缩到 `lv_pc_simulator` 目录里面去,接着再把 `lv_pc_simulator` 目录下的 `lv_drivers.zip`, `lv_examples.zip`, `lvgl.zip` 三个压缩包依次进行在当前目录解压缩的操作,再接着在 `lv_pc_simulator` 目录下新建一个名为 `codeblocks` 的子目录,这个子目录主要是用来存放 CodeBlock 项目工程文件的,最后的目录结构如下图所示:

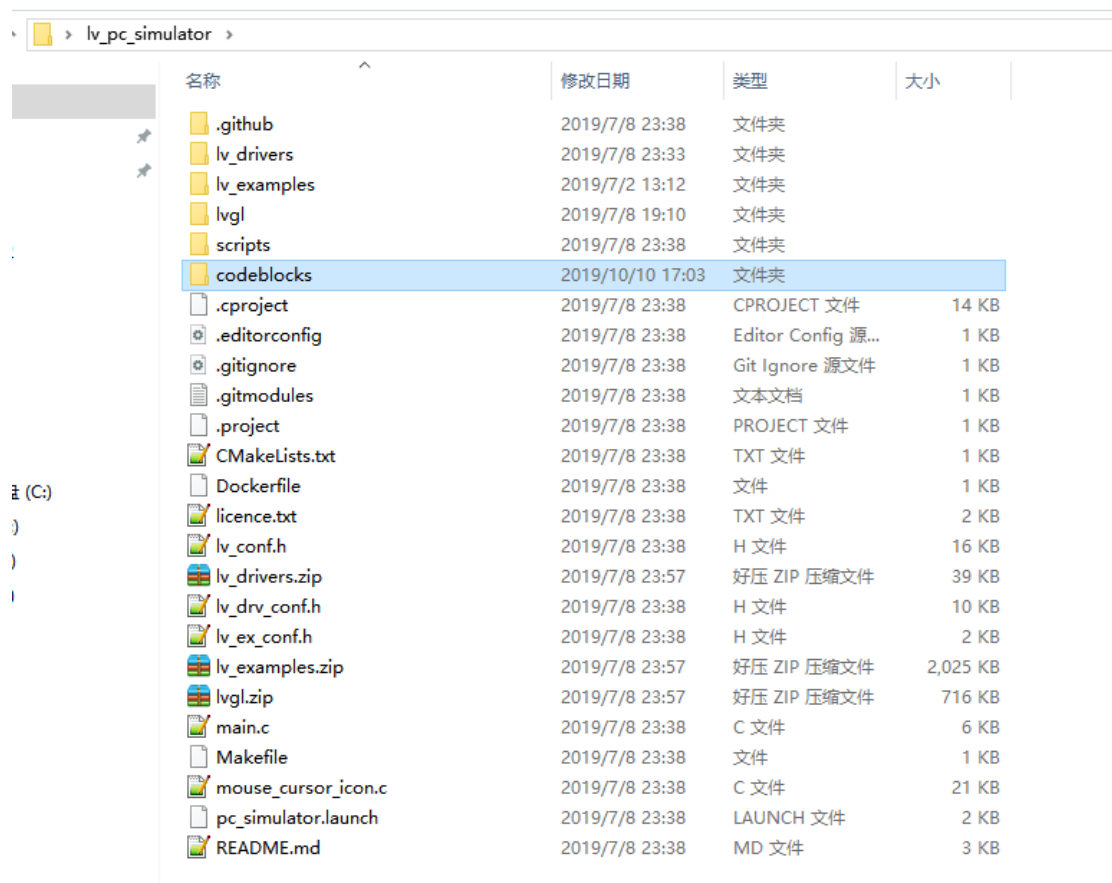


图 2.2.1 lv\_pc\_simulator 的目录结构

接下来,我们需要打开 CodeBlock 软件来新建工程了,初次打开 CodeBlock 软件,会弹出如下界面:

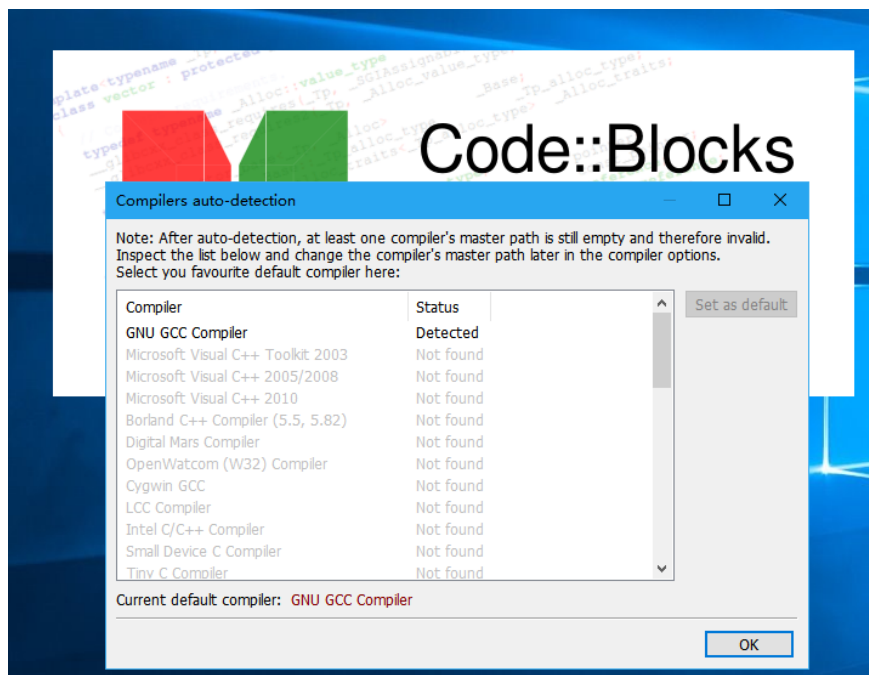


图 2.2.2 CodeBlock 自动检测编译器

当看到这个界面时,不要惊慌,这是在自动检测编译器,我们直接点击 OK 按钮即可.打开软件

之后,点击菜单栏左上角的 File->New->Project...来弹出项目新建向导,我们选择 Console 应用,如下图所示:

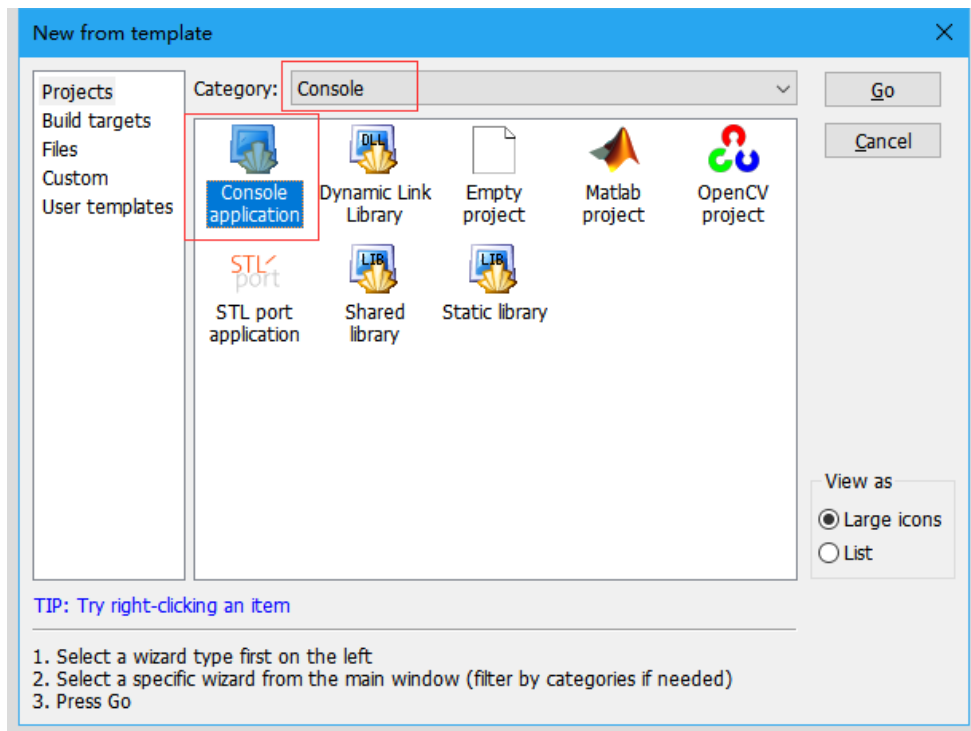


图 2.2.3 选择 Console 应用

接着点击 Go 按钮,又会弹到另外一个对话框,点击 Next 按钮,接着进入到语言选择对话框,如下图所示:

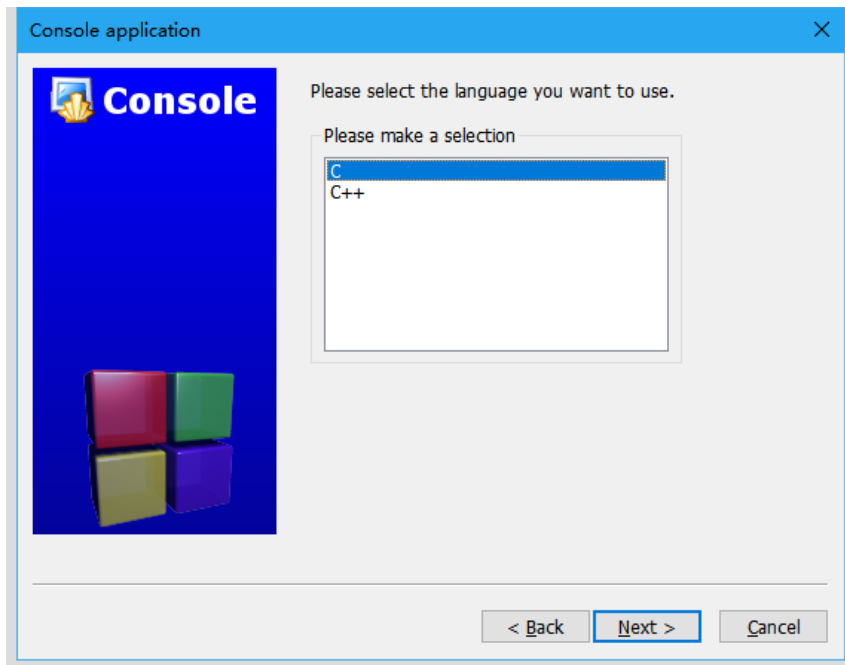


图 2.2.4 语言选择

这里我们选择 C 语言,然后再点击 Next 按钮,又会进入到项目信息输入对话框,如下图所示:



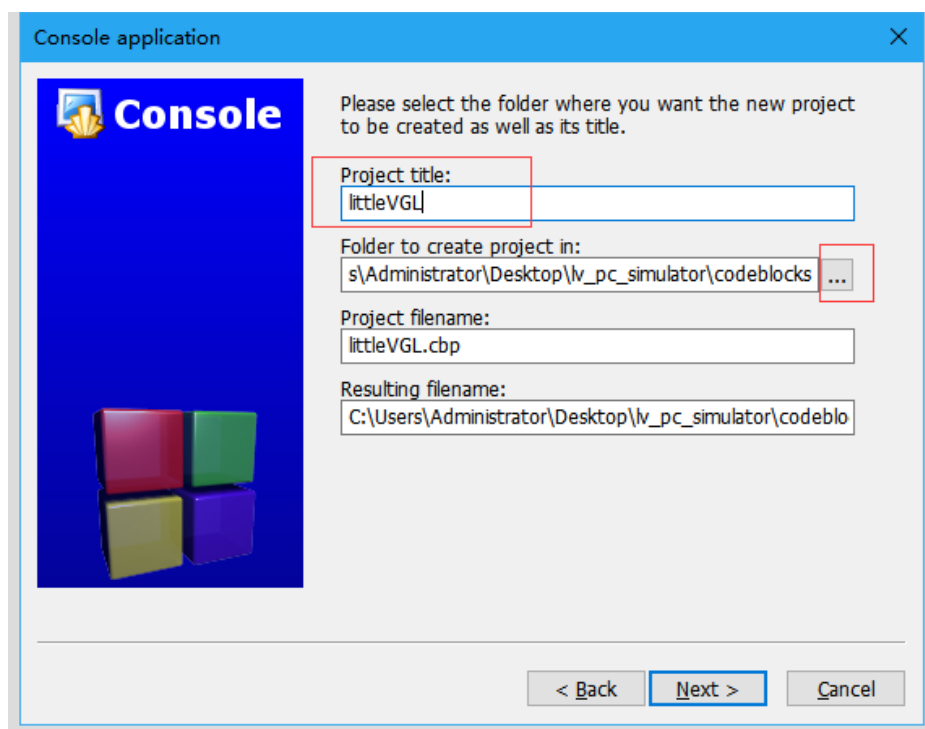


图 2.2.5 项目信息输入

这里我们把项目名输入为 littleVGL,然后选择项目的存放路径到 lv\_pc\_simulator 目录下的 codeblocks 子目录下,最后就是一路的Next操作来完成项目的创建.项目创建完成之后,默认只有一个 main.c 文件,如下图所示:

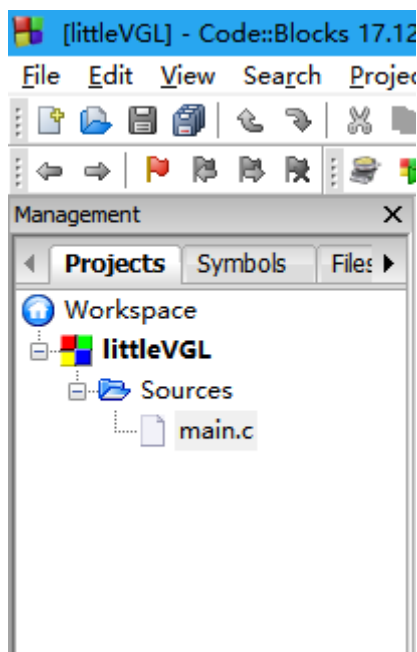


图 2.2.6 项目的默认结构

这里我们不需要默认创建的 main.c 文件,右键 main.c 文件,点击 Remove file from project 来去除此文件,同时最好把 main.c 从磁盘上彻底删除,其位于 lv\_pc\_simulator\codeblocks\littleVGL 目录下,接下来我们需要把 lv\_pc\_simulator 模拟器文件递归添加到 CodeBlock 上去,右键



图标,点击 Add files recursively 之后,弹出如下路径选择对话框:

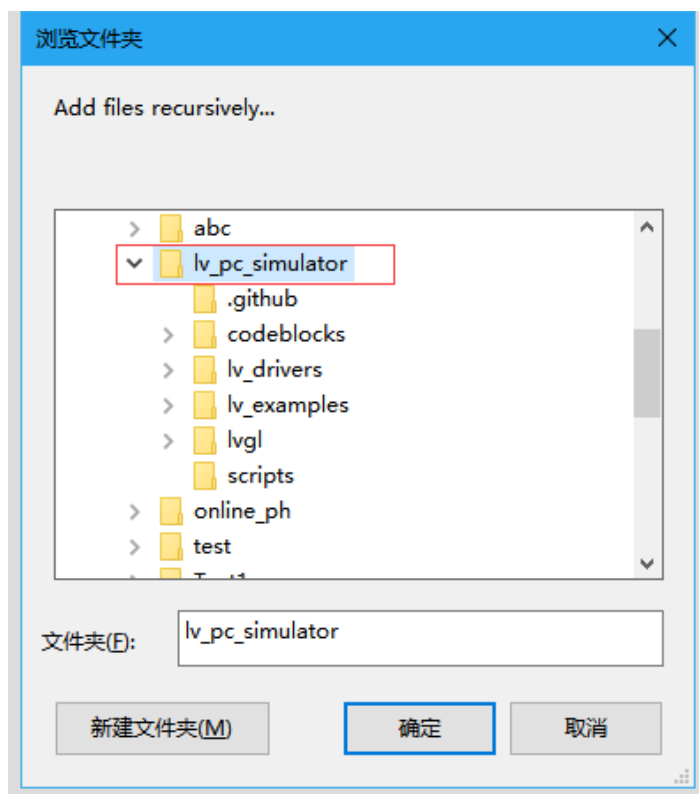


图 2.2.7 路径选择对话框

这里我们一定要选择 lv\_pc\_simulator 顶层目录,然后点击确定按钮,又弹出如下对话框:

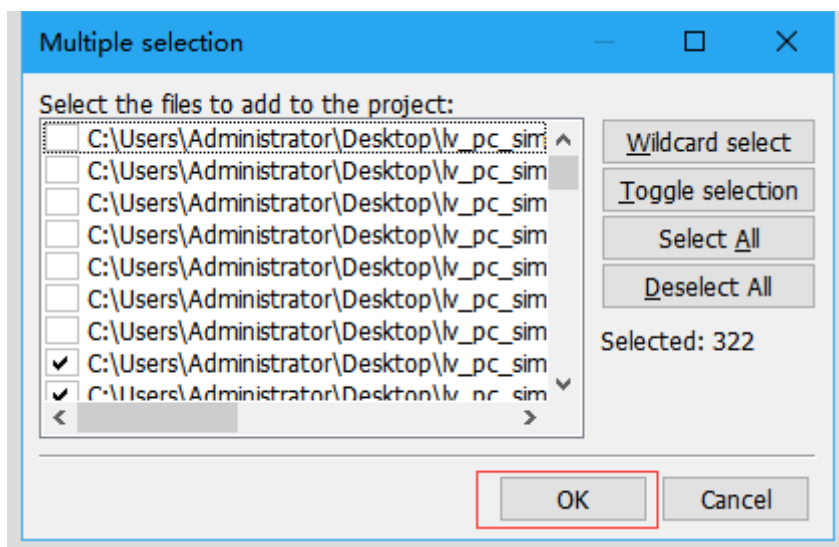


图 2.2.8 具体文件选择

在这里我们什么都不要动,直接点击 OK 按钮即可,接着又弹出

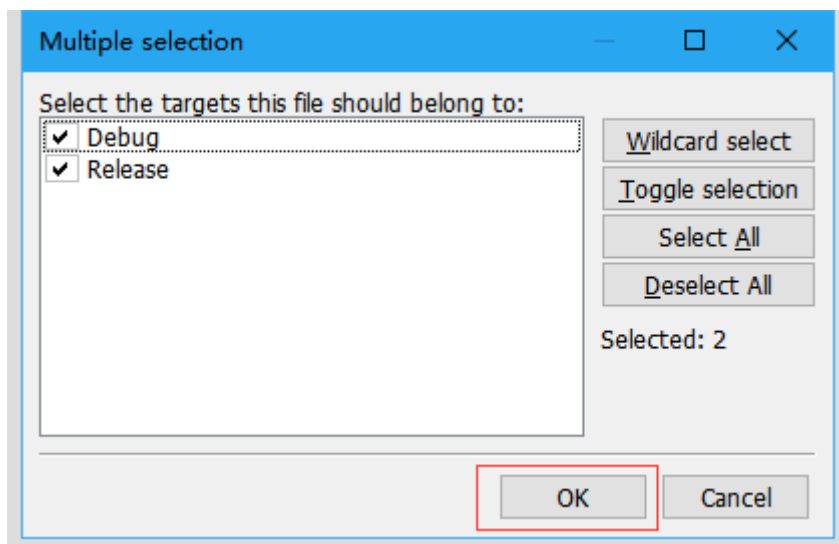



图 2.2.9 targets 选择

同样道理,什么也不动,直接点击 OK 按钮,至此文件的添加操作就完成了,接下来我们需要用到

SDL2-devel-2.0.10-mingw.tar.gz 里面的文件了,我们先可以把其解压缩到桌面,然后做下面 2 点操作:

1.把 SDL2-2.0.10\i686-w64-mingw32\include 目录下的 SDL2 目录拷贝到 CodeBlock 的安装目录 D:\baseSoftware\CodeBlocks\MinGW\include 下面

2.把 SDL2-2.0.10\i686-w64-mingw32\lib 下的 7 个.a 库文件拷贝到 CodeBlock 的安装目录 D:\baseSoftware\CodeBlocks\MinGW\lib 下面

做完上面 2 点操作之后,我们需要配置一下 CodeBlock 软件,右键  图标,点击 Build options,打开对话框之后,在 Other linker options 下输入 -lmingw32 -lSDL2main -lSDL2,如下图所示:

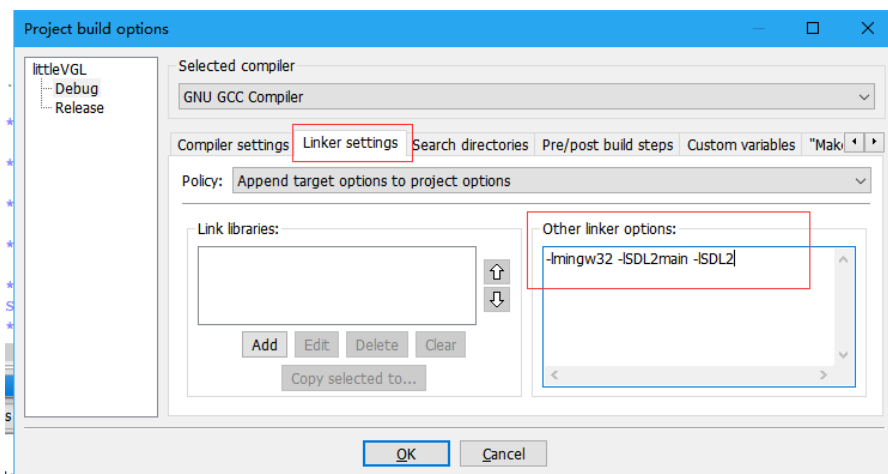


图 2.2.10 添加链接选项

然后我们还需要添加头文件搜索路径,如下图所示:

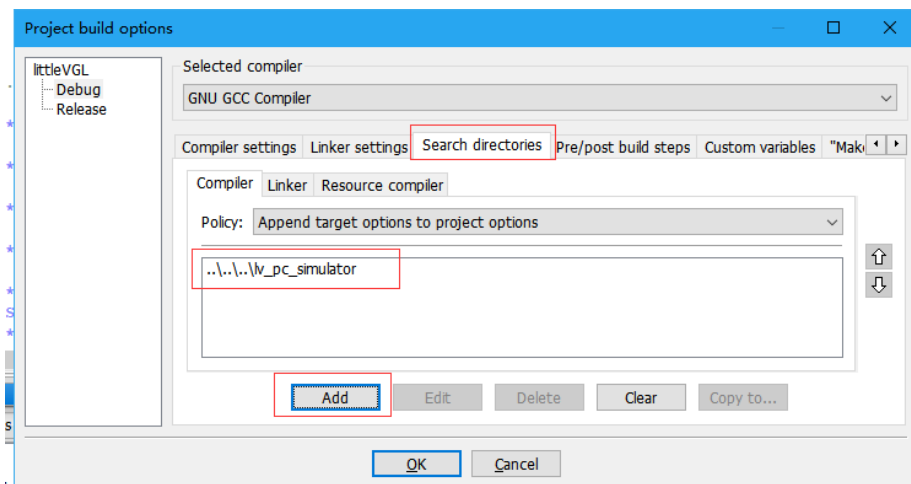


图 2.2.11 添加头文件路径

注意,添加头文件路径时,选择 lv\_pc\_simulator 相对路径即可,最后点击 OK 按钮保存配置.接着我们点击打开 main.c 文件

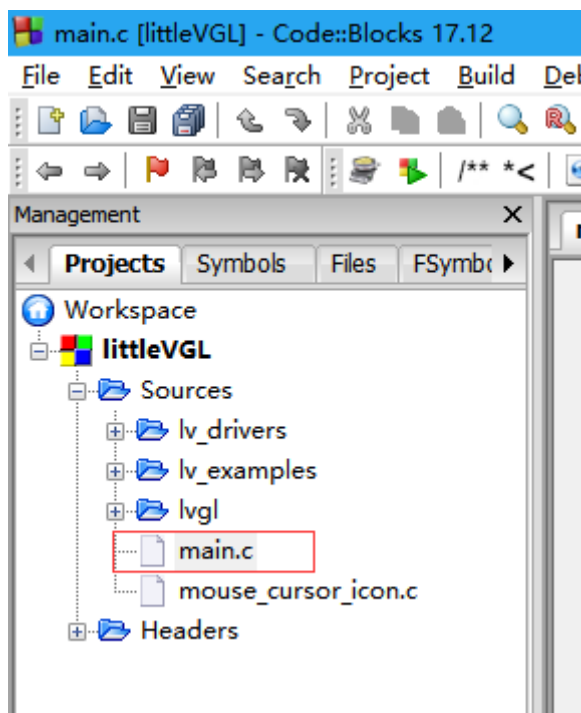



图 2.2.12 main.c 文件

打开之后,我们去掉 71 行处的行注释,即让 demo\_create(); 例程得到运行,当然你也可以选择另外 4 个例程中的一个来运行,如下所示:

```
benchmark_create();      在 74 行
lv_test_theme_1(lv_theme_night_init(15, NULL));  在 77 行
lv_test_theme_2();       在 79 行
lv_test_group_1();       在 81 行
```

选择好例程之后,终于迎来了我们的第一次编译,我们点击  图标来编译,不出意外的话,会

报 1 个错误,如下图所示:

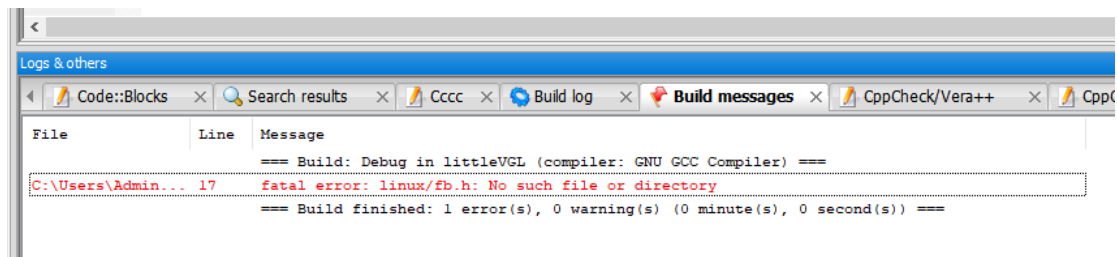


图 2.2.13 初次编译报 1 个错

别慌,这个错误很好解决,这是因为 Windows 系统不支持 Linux 帧缓冲,把宏定义 USE\_FBDEV 的值改为 0 就可以了,USE\_FBDEV 宏在 lv\_pc\_simulator\lv\_drv\_conf.h 头文件中的 186 行处,改成 0 之后,编译不会报错了,但是运行起来后,会报如下图所示的一个错误.

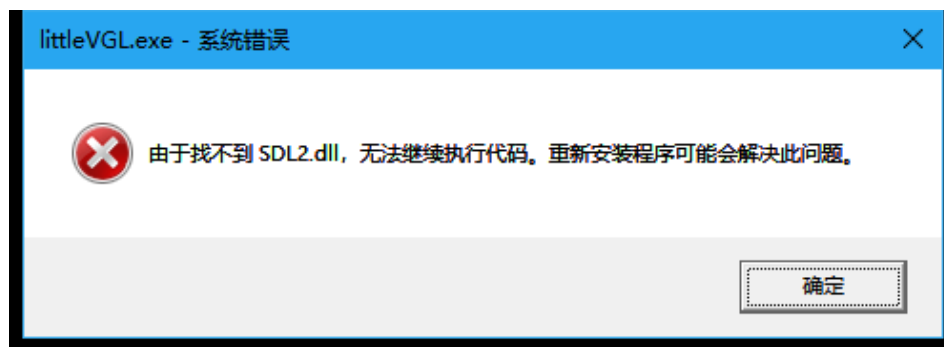


图 2.2.14 运行报错

这个错误也很好解决,是因为找不到 SDL2.dll 库,我们需要把 SDL2-2.0.10\i686-w64-mingw32\bin 目录下的 SDL2.dll 文件拷贝到 lv\_pc\_simulator\codeblocks\littleVGL\bin\Debug 目录下面,如下图所示:

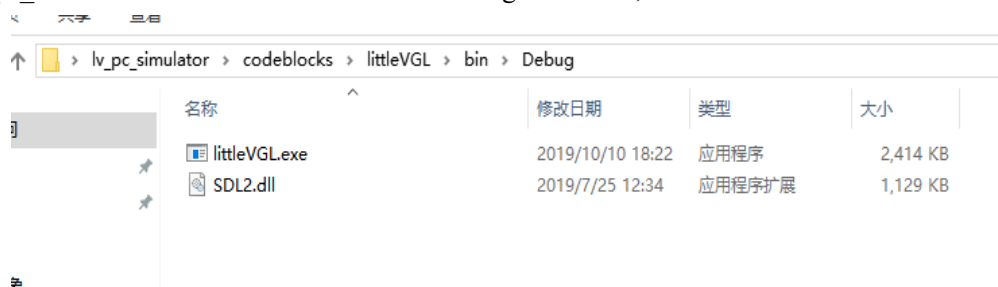


图 2.2.15 拷贝 SDL2.dll 库

拷贝完成之后,再次编译运行就不会有任何错误了,最终的模拟器例程演示效果请看下面

## 2.3 模拟器演示效果

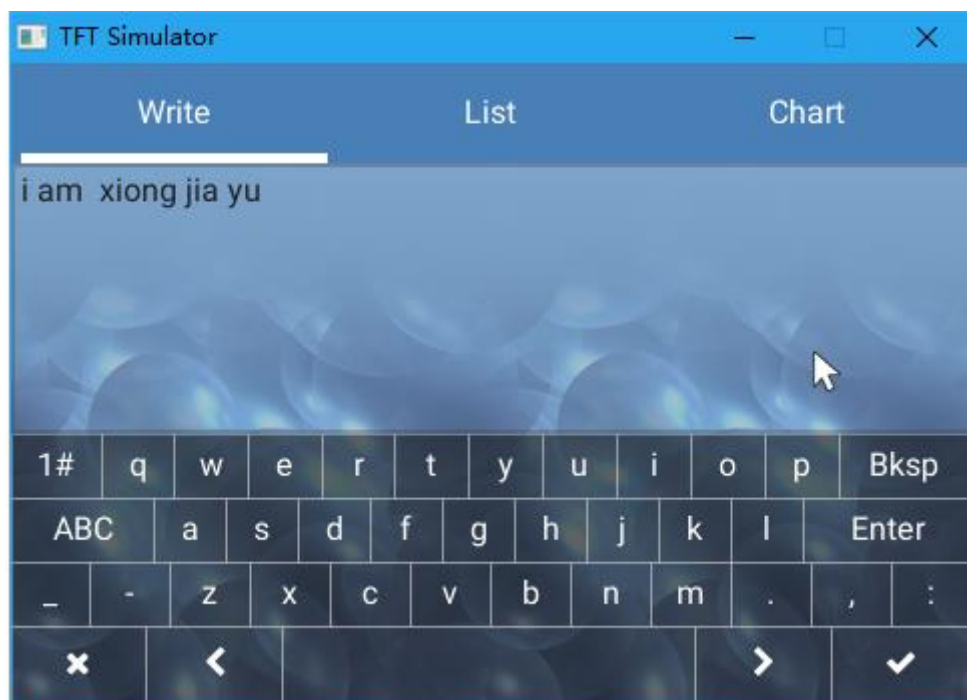


图 2.3.1 Write 界面

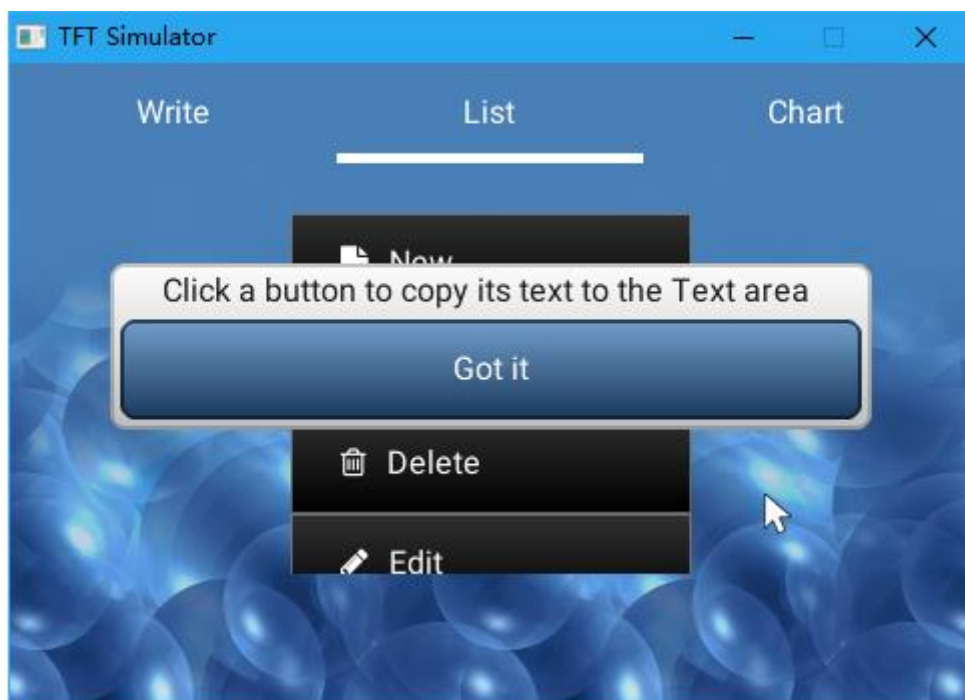


图 2.3.2 List 界面



图 2.3.3 Chart 界面

上面的三张演示效果图是由 `demo_create()` 例程产生的,你们自己也可以通过行注释来运行其他的例程.

### 3.如何添加自己的 GUI 测试代码

怕有的同学不知道怎么添加自己的 GUI 测试代码,特此稍微讲解一下,步骤也是很简单的,

你可以直接在 `main.c` 文件中添加你自己的测试函数,然后放在 `main` 函数中的 `hal_init()`;后面进行调用即可,当然你得先把官方自带的几个例程先给注释掉,否则会影响你的测试效果的,直接在 `main.c` 文件中添加的方式虽然简单,但是我不推荐,因为不符合我们的模块化编程思想,最好的操作步骤应该是这样的,我们在 `lv_pc_simulator\lv_examples\lv_apps` 目录下新建一个 `test` 目录,然后在 `test` 目录下新建 `test.c` 和 `test.h` 文件,然后用前面介绍过的方法,把 `test` 目录递归添加到 `lv_apps` 分组下,如下图所示:

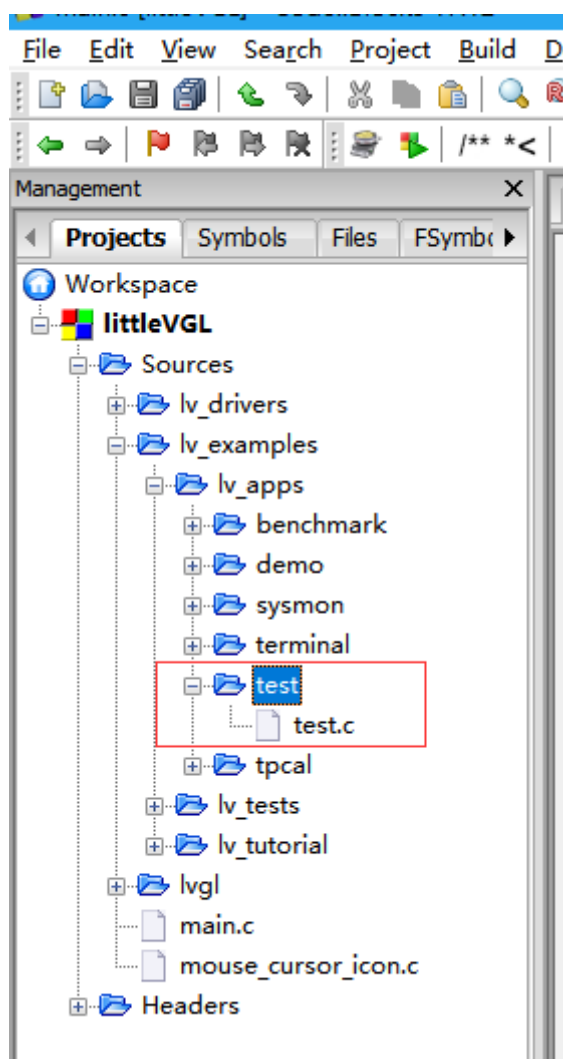


图 3.1 添加 test 目录

`test.h` 文件的内容可以大致如下:

```
#ifndef __TEST_H__
#define __TEST_H__
```



```
#ifndef __cplusplus
extern "C" {
#endif

/*****
 *      INCLUDES
 *****/

#ifdef LV_CONF_INCLUDE_SIMPLE
#include "lvgl.h"
#include "lv_ex_conf.h"
#else
#include "../..../lvgl/lvgl.h"
#include "../..../lv_ex_conf.h"
#endif
```

```
void test_start(void);
```

```
#ifndef __cplusplus
} /* extern "C" */
#endif

#endif // __TEST_H__
```

test.c 文件的内容可以大致如下:

```
#include "test.h"

void test_start()
{
    //获取当前的屏幕对象
    lv_obj_t * scr = lv_disp_get_scr_act(NULL);

    //在屏幕上创建一个 label 控件
    lv_obj_t * label1 = lv_label_create(scr, NULL);

    //设置 label 的文本内容
    lv_label_set_text(label1, "I am xiong jia yu");

    //设置文本和父控件(在这里就是屏幕)居中对齐
    lv_obj_align(label1, NULL, LV_ALIGN_CENTER, 0, 0);
}
```

然后在 main.c 文件中引入 test.h 的头文件,如下所示

```
#include "lv_examples/lv_apps/test/test.h"
```

再接着把 test\_start();放在 main 函数中的 hal\_init();后面进行调用,最后编译运行可以看到如下效果:

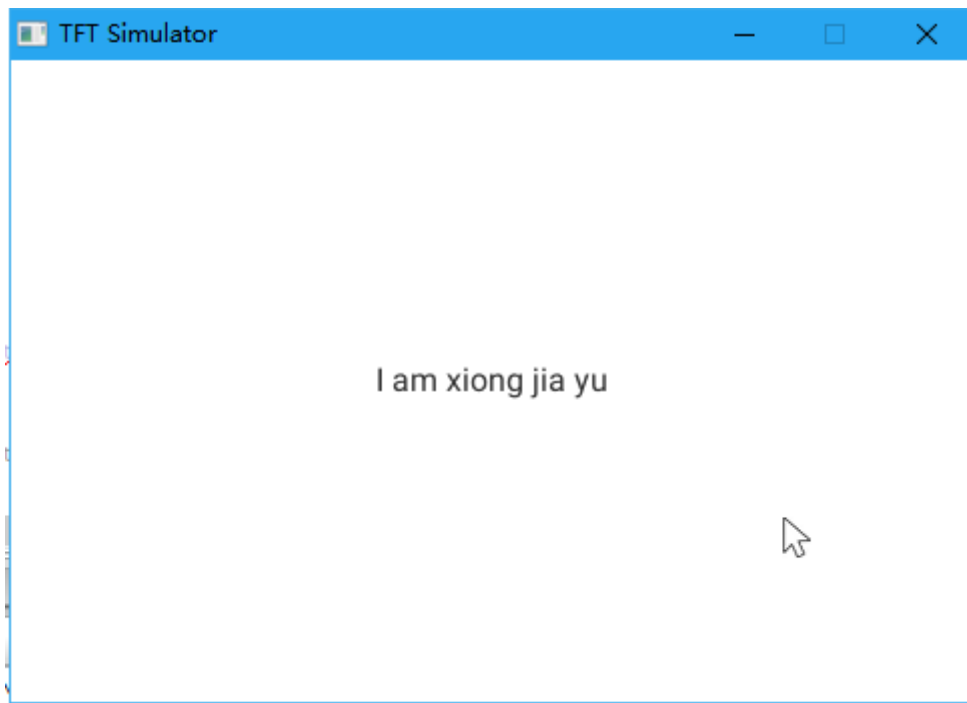


图 3.2 自己的代码测试效果

## 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