# SimpleGUI

模拟器环境配置

摘要

引导您搭建 SimpleGUI 的模拟器开发环境,帮助您能够脱离目标平台,开发通用的、可移植的人机交互接口。

Polarix

Xuyulin91@163.com

#### 1、 环境介绍

SimpleGUI 为了方便开发者进行开发或预开发作业,特设计了 PC 端模拟环境用以 辅助开发,模拟环境使用 Codeblocks 构建,图形界面基于 wxWidgets 建立。

#### 2、 准备

以下操作仅适用于 WindowsXP 及以上操作系统。

在搭建环境前,请先下载 Codeblocks 集成开发环境、wxWidgets 图形库源码与编译器,三者的主页如下:

Codeblocks:	http://www.codeblocks.org/
wxWidgets:	http://www.wxwidgets.org/
TDM-GCC:	http://tdm-gcc.tdragon.net/

截至此文档最后一次更新, Codeblocks 最新版本为 17.12, wxWidgets 最新版本为 开发版 3.1.1, 稳定版 3.0.4, TDM-GCC 的最新版本为 5.1.0。

关于搭建 SimpleGUI 模拟环境, Codeblocks 应使用 16.01 或以上版本, 其他版本也 可使用但操作上可能有少许出入, 请自行摸索。wxWidgets 请使用 3.0 以上版本。

如果用户从 Codeblocks 主页下载 CodeblocksIDE,最好下载带编译器的版本, Codeblocks 的最新版本 17.12 自带编译器版本携带的就是 TDM-GCC5.1.0,如果您的电脑上已经安装了 GCC 编译器,请确保您的编译器是 TDM-GCC4.9.1 或更高版本,使用 MinGW 编译的 wxWidgets 库无法正常使用。

#### 3、 部署 CodeBlocks

下载并解压 CodeblocksIDE。

share	2017/12/31 1:18	文件夹	
🗗 Addr2LineUI.exe	2017/12/29 15:41	应用程序	199 KB
cb_console_runner.exe	2017/12/29 15:41	应用程序	113 KB
cb_share_config.exe	2017/12/29 15:41	应用程序	834 KB
😼 CbLauncher.exe	2017/12/29 15:41	应用程序	57 KB
📧 cbp2make.exe	2017/12/29 15:41	应用程序	1,314 KB
🗗 cctest.exe	2017/12/29 15:41	应用程序	1,223 KB
🗟 codeblocks.dll	2017/12/29 15:41	应用程序扩展	6,353 KB
😽 Codeblocks.exe	2017/12/29 15:41	应用程序	2,300 KB
🗟 dhahala dli	2017/12/25 2:27	应用程序扩展	1 202 KB

图 1 解压缩后的 Codeblocks (部分)

如果是下载的带编译器版本, 解压后还会有一个叫做 MinGW 的文件夹, 这个 MinGW 文件夹就是 TDM-GCC 编译器, CodeBlocks16.01 带的是 4.9.2 版本。17.12 带的是 5.1.0。 Codeblocks 解压后即可运行,不需要其他的安装或部署。。

## 4、 部署 TDM-GCC 编译器

如果下载的是纯净版不带编译器的 Codeblocks, 那么在部署开发环境之前, 请用户 自行下载好 TDM-GCC5.1.0 并安装, 如果下载的并非安装版或直接使用 Codeblocks 自 带的编译器, 为确保后续操作能够正确执行, 请按照以下操作部署 TDM-GCC。

为避免编译器在部署或运行中出现一些意想不到的问题,请确保编译器被安装或解 压缩在纯英文路径下。本例中,TDM-GCC 被解压缩并放置在 D:\Programs\GCC\TDM-GCC\_5.1.0 中。

脑 > Software (D:) > Programs > GCC	> TDM-GCC_5.1.0		
名称 ^	修改日期	类型	大小
🔒 bin	2017/12/31 1:16	文件夹	
doc	2017/12/31 1:16	文件夹	
gdb32	2017/12/31 1:16	文件夹	
include	2017/12/31 1:16	文件夹	
📊 lib	2017/12/31 1:16	文件夹	
libexec	2017/12/31 1:16	文件夹	
ningw32	2017/12/31 1:16	文件夹	
share	2017/12/31 1:16	文件夹	
COPYING.ISL.txt	2013/9/29 3:43	Notepad++ Doc	0 KB
	004010104 40 55		0.1470

图 2 解压缩后的 TDM-GCC

再次强调,为保证 wxWidgets 编译和使用正常,必须使用 TDM-GCC4.9.1 或更高版本,如果之前有安装过 Mingw、Mingw-w64 或其他 GNU 编译器的,请进行完全卸载和清理,以避免后续操作可能出现的错误。清理如果是下载的带编译器版本,解压后还会有一个叫做 MinGW 的文件夹,这个 MinGW 文件夹就是 TDM-GCC 编译器,CodeBlocks16.01 带的是 4.9.2 版本。17.12 带的是 5.1.0。

以上操作完成后, 进入系统属性设置窗口以进行环境变量的配置。不同版本的 Windows 进入方法稍有差别, 例如 WindwosXP 可以在"我的电脑"上右键单击, 然后在 菜单中选择"属性", Windows7 中需要右键单击"计算机", 然后在显示的"系统"窗口中, 单击左侧边栏中的"高级系统设置"进入系统属性设置窗口, 而在 Win8 及以上版本中, 桌面默认取消了"此电脑"图标, 用户可以在文件管理器右侧的导航窗格中的"此电脑"图 标上单击右键并选择"属性"菜单进入系统属性设置窗口。

此外,还可以使用一个在各个版本中共通的方式来进进入系统属性设置,按 Win+R 组合键,在弹出的运行对话框中输入"sysdm.cpl"并回车,系统直接打开系统属性窗口,然后选择"高级"选项卡,点击"环境变量"按钮,进入环境变量设置窗口。

系统属性	$\times$
计算机名 硬件 高级 革统保护 远程	
要进行大多数更改,你必须作为管理员登录。	
性能	
i 视觉效果,处理器计划,内存使用,以及虚拟内存	
; 设置( <u>S</u> )	
用户配置文件	
与登录帐户相关的桌面设置	
设置(E)	
启动和故障恢复	
系统启动、系统故障和调试信息	
设置(I)	
环境变量( <u>N</u> )	
	-
	۸)
WHALE PLUIA N⊻HI	

图 3 系统属性窗口

安量	值
OneDrive	C:\Users\Polarix\OneDrive
Path	C:\Users\Polarix\AppData\Local\Microsoft\WindowsApps:
TEMP	C:\Users\Polarix\AppData\Local\Temp
ТМР	C:\Users\Polarix\AppData\Local\Temp
	新建(1) 場理(こ) 面除(U)
〔统变量( <u>S</u> )	
统变量( <u>S</u> ) 变量	值
统变量( <u>S)</u> 变量 ADSK_CLM_WPAD_PROXY	值 FALSE
统变量(S) 变量 ADSK_CLM_WPAD_PROXY ComSpec	值 FALSE C:\WINDOWS\system32\cmd.exe
续变量(S) 变量 ADSK_CLM_WPAD_PROXY ComSpec LIBRARY_PATH	值 FALSE C:\WINDOWS\system32\cmd.exe %GCC_ROOT%\lib
续变量(S) 变量 ADSK_CLM_WPAD_PROXY ComSpec LIBRARY_PATH NUMBER_OF_PROCESSORS	fa FALSE C:\WINDOWS\system32\cmd.exe %GCC_ROOT%\lib 4
续变量(S) 变量 ADSK_CLM_WPAD_PROXY ComSpec LIBRARY_PATH NUMBER_OF_PROCESSORS OS	fa FALSE C:\WINDOWS\system32\cmd.exe %GCC_ROOT%\lib 4 Windows_NT
(統变量(S) 变量 ADSK_CLM_WPAD_PROXY ComSpec LIBRARY_PATH NUMBER_OF_PROCESSORS OS Path	fa FALSE C:\WINDOWS\system32\cmd.exe %GCC_ROOT%\lib 4 Windows_NT D:\Programs\GCC\TDM-GCC_5.1.0\bin;C:\WINDOWS\system3
鉄変量(S) 変量 ADSK_CLM_WPAD_PROXY ComSpec LIBRARY_PATH NUMBER_OF_PROCESSORS OS Path PATHEXT	Image: Image
統变量(S) 変量 ADSK_CLM_WPAD_PROXY ComSpec LIBRARY_PATH NUMBER_OF_PROCESSORS OS Path PATHEXT	false     C:\WINDOWS\system32\cmd.exe     %GCC_ROOT%\lib     4     Windows_NT     D:\Programs\GCC\TDM-GCC_5.1.0\bin;C:\WINDOWS\system3     .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JSE;.WSF;.WSF;.WSF\.MSC     ✓

图 4 环境变量设置窗口

然后在环境变量设置窗口中选择下方"系统变量"列表中的"Path"变量,然后点击"编辑"按钮。

<b>辑环境变量</b>	
D:\Programs\GCC\TDM-GCC_5.1.0\bin	新建( <u>N</u> )
%SystemRoot%\system32	
%SystemRoot%	编辑(E)
%SystemRoot%\System32\Wbem	
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\	浏览(B)
C:\Program Files (x86)\Git\cmd	
C:\Program Files (x86)\GitExtensions\	删除(D)
C:\Program Files\TortoiseSVN\bin	
	上移(U)
	下移( <u>O</u> )
	编辑文本(])
确	定取消

图 5 添加 GCC 环境变量

在此窗口中添加GCC运行用的环境变量,变量内容为GCC编译器程序的所在路径,添加完成后,点击"确定"按钮关闭环境变量编辑窗口。

此时,按Win+R键,时输入"cmd"然后回车,打开控制台,然后输入"gcc-v"控制台 上将输出当前配置好的 TDM-GCC 的版本信息。



图 6 部署完成的 TDM-GCC5.1.0.

如果显示的 GCC 版本和预计的不一致,请检查 Path 环境变量中保存的路径是否有 其他注册的 GCC 编译器路径,如果有,请删除。如果使用安装的形式部署 TDM-GCC, 理论上安装完成后,环境变量会被自动注册,如果安装完成后,控制台中无法显示正确 的 GCC 版本信息,请参照上述步骤重新部署 TDM-GCC。

# 5、 部署 wxWidgets 库

接下来的操作将建立在前节 GCC 部署成功的前提下,请确保 TDM-GCC 编译器已 正确部署。

从 wxWidgets 官方网站下载 wxWidgets 源码并在本地解压缩。本例中解压缩至 D:\wxWidgets\Development\Source 文件夹下,实际操作中请以实际解压缩路径为准。 为避免编译过程中出现一些意想不到的问题,请确保解压缩路径为纯英文路径。

wxWidgets 解压缩后,打开控制台,进入解压缩路径。然后在进入解压缩根目录下的 build\msw 目录,准备编译 Windows 版 wxWidgets 库。

然后输入以下命令,开始编译 wxWidgets 库。

mingw32-make -f makefile.gcc MONOLITHIC=1 SHARED=1 UNICODE=1 BUILD=release CXXFLAGS="-fno-keep-inline-dllexport -std=gnu++11"

当然,如果有其他的发布需求,也可以编译其他形式的 wxWidgets 库,具体的编译参数及含义请自行在网络上搜索。

图 7 编译 wxWidgets

编译全程视电脑性能而定,耗时约 30 分钟。编译过程中可能出现因类型转换导致的警告,可以忽略不计。



图 8 编译完成

#### 6、 **配置环境**

编译完 wxWidgets 环境后, 启动 Condeblocks。 选择"Settings"(设置)下的"Global variables"(全局变量)项目。



图 9 Global variables 选项

在打开的全局变量编辑窗口中,点击"New"按钮,新建一个全局变量。

Global Variable Editor		×
Current Set: default		✓ Clone New Delete
Current variable	Built-in fields:	User-defined fields:
	in Please specify a name for the new variables iii wx ol OK	Cancel
	bin	
Help		Close

图 10 创建新的全局变量

然后将全局变量的根路径、包含路径和库路经分别设置为 wxWidgets 的根路径 include 路径和 lib 路径,其他留空即可,参考下图。

Global Variable Editor						×
Current Set: default			~	Clone	New	Delete
Current variable		Built-in fields:		User-	defined field	ds:
New Clone Delete	base	:Widgets\Development\Source				
WX		The base member is mandatory!				
	include	ts\Development\Source\include				
	lib	dgets\Development\Source\lib				
	obj					
	bin					
	cflags					
	Iflags					
Help	,		,			Close

图 11 wx 环境变量的设定

然后点击"Close"关闭环境变量编辑窗口,至此,SimpleGUI 模拟环境开发需要使用的 Codeblocks+wxWidgets 环境搭建完成。

# 7、 编译工程

进入 SimpleGUI 根目录, 各子目录存放内容说明如下:

DemoProc	SimpleGUI 应用演示代码
DemoProject	移植演示代码,目前只配置了 STM32F103 正在 EmBitz下的演示环境。
Documents	说明文档
GUI	绘图引擎接口实现
HMI	交互引擎接口实现
SimulatorEnv	模拟环境工程

进入 SimulatorEnv\Project\CodeBlocks 目录下, SimpleGUI.cbp 文件夹即为模拟环 境的工程文件,使用 Codeblocks 打开。如果用户下载的是 nosetup (绿色版)的 Codeblocks,需要手动建立关联才能直接用双击的方式打开 cbp 文件,否则,用户只能 先打开 Codeblocks,然后从 Codeblocks 中执行打开操作以打开工程文件。



#### 图 12 SimpleGUI 模拟环境工程

	3
📕 [SimpleGUI] - Code::Blocks 17.12 — 🗆 🔿	<
<u>Elle Edit V</u> iew Sea <u>r</u> ch <u>P</u> roject <u>B</u> uild <u>D</u> ebug Fortra <u>n w</u> xSmith <u>T</u> ools T <u>o</u> ols+ Plugins Do <u>x</u> yBlocks <u>S</u> ettings <u>H</u> elp	
📔 🕑 🚘 🗃 🎒   🗞 🐂 🛝 🐘 🌰   🔍 🍭 🔯 🔅 🕨 🖉 Debug 🔷 😼 🙀 🐼 🖾 🔛 🔤 👘	:
Management X	_
Projects Symbols File:	
0 Workspace	
🗄 🍢 SimpleGUI	
👳 🔁 Sources	
🖶 🔁 Headers	
🗄 🔁 Resources	
Logs & others	×
🕢 🚺 Code::Blocks 🛛 兴 Search results 👋 🥂 Cccc 👋 😒 Build log 🗙 🌳 Build messages 🔧	Þ
<pre>\wxExpand\src\wxPixelatedPanel.oBuild\Debug\obj\SimulatorEnv\Resource</pre>	^
Nessource.res -mthreads -lwxmsv3lu -liconv -mwindows Output file is \ NulliQuebuq/bin/SimpleGU.exe with size 4.24 MB	
Process terminated with status 0 (0 minute(s), 9 second(s))	
0 error(s), 0 varning(s) (0 minute(s), 3 second(s))	
	*
default	

点击工具栏上的编译按钮或按快捷键 Ctrl+F9, 开始编译模拟环境演示工程。

图 13 编译完成

编译完成,显示无错误和警告,现在点击工具栏上的运行按钮或按快捷键 Ctrl+F10 就可以看到模拟环境的运行效果了。

SimpleGUI Simulator	×
۱ 🖬 🖬	
欢迎来到SimpleGUI 演示工程,本工程用 演示工程,本工程用 演示SimpleGUI各API的 最示效果、使用方法以 要注意,	- 与 了
Initialzied.	

图 14 SimpleGUI 模拟环境演示程序

### 8、 环境配置

模拟环境中的虚拟显示屏幕的配色方案为黄底黑字的 LCD 点阵显示屏,为最大程度上模拟真实情况下的使用场景,模拟环境的虚拟 LCD 面板可以通过修改配置定义修改颜色和尺寸。

进入工程根目录,打开工程跟目录下的 SimulatorEnv\Common\inc 文件夹(对应工 程路径/SimpleGUI/Headers/SimulatorEnv/Common/inc),打开 Common.h 文件,找到 以下宏定义:

LCD_COLOR_OBJ_PIX	逻辑有效像素点颜色(前景颜色)
LCD_COLOR_OBJ_BKG	逻辑无效像素点颜色(背景颜色)
LCD_COLOR_OBJ_EDGE	边框颜色
LCD_COLOR_OBJ_GRID	网格颜色

PARAM_DEFAULT_PIXEL_NUM_H	横向像素点数量
PARAM_DEFAULT_PIXEL_NUM_V	纵向像素点数量
PARAM_DEFAULT_PIXEL_SIZE	像素尺寸
PARAM_DEFAULT_EDGE_WIDTH	边框宽度

工程默认情况下为 128\*64, 黄黑色屏幕, 不显示像素网格。如果想改变外观, 可以 同通过修改宏定义的值, 颜色为 RGBA 格式小端对其的 32 位数, 可以通过屏幕取色工 具获取 RGB 值。

例如,想修改成192\*64,黑底蓝字的OLED效果,可以按照如下方式修改:

1、使用一张目标屏幕的照片,使用取色工具获取屏幕背景、像素以及边框颜色的 RGB 值,比如淘宝上的照片,就可以直接使用。



图 15 淘宝上的商品照片,可以直接从画红圈的位置获取需要的颜色

2、将颜色的 RGBA 值分别更新入三个宏定义中,例如上图中屏幕背景色、像素色 和边框色分别设定为 0xFF070707、0xFFF1FA4F 和 0xFFC9652F,由于不需要显 示网格,所以不需要修改网格颜色。

#define LCD_COLOR_OBJ_PIX	(0xFFF1FA4F)
#define LCD_COLOR_OBJ_BKG	(0xFF070707)
<pre>#define LCD_COLOR_OBJ_EDGE</pre>	(0xFFC9652F)
<pre>#define LCD_COLOR_OBJ_GRID</pre>	(0xFF383834)

图 16 修改宏定义的颜色信息

- 3、修改 PARAM\_DEFAULT\_PIXEL\_NUM\_H 宏定义的值为 192。
- 4、重新编译工程即可看到效果。

■ SimpleGUI Simulator ×
测试列表
「「 系统时间
2018-06-18
15:48:34
Initialzied.

图 17 修改配色和尺寸后的效果

上述的宏用于 SimulatorEnv\Common\src\Common.c (对应工程路径/SimpleGUI/Sources/SimulatorEnv/Common/src/Common.c) 文件中,对默认参数设定函数 SetDefaultParameterData 功能能定义,此函数将在编译好的模拟器启动时对像素面板进行初始化时调用,届时此处设定的参数将会被用来设置像素面板的显示。



图 18 使能网格线后的效果

100

Initialzied.

注意,为了避免画面显示失调,模拟器只有在像素点尺寸(也就是宏定义 PARAM\_DEFAULT\_PIXEL\_SIZE)的值**大于 4**时,网格设定才会有效,在像素点尺寸小于 等于 2 时,网格线将被强制关闭。上图为像素尺寸为 4 时,开启预览的效果。

# 9、 开发接口

模拟环境的开发用接口在工程目录下的 SimulatorEnv\Interface\src 文件夹下(工程路径),共有两个文件:

UserActionInterface.cpp	用户操作接口,用于环境初始化、用户按键 操作做、定时器动作等。
VirtualDeviceInterface.cpp	虚拟设备接口,用于模拟器显示面板操作, 通常用于更新屏幕显示,刷新屏幕、清空屏 幕等。

以上两个文件中定义的接口函数将完成用户代码和模拟器之间的交互、这在逻辑上, 相当于目标平台上的驱动程序。

UserActionInterface.cpp 文件中定义了以下接口:

UAIF_OnInitialize	模拟器程序启动时调用,通常用于初始化操 作。
UAIF_OnKeyPress	用户键盘事件响应函数。
UAIF_OnTimerEventProcess	模拟定时器函数,每1ms触发一次。
UAIF_OnRTCUpdateEventProcess	实时时钟函数,每秒触发一次用于更新系统 时间。

此文件中定义的函数将被模拟环境中相应的处理函数调用,分别对应启动、用户按 键(交互)、定时器三种事件,其他需求可以在定时器中模拟或按照需求增加接口。

VirtualDeviceInterface.cpp 文件中定义了以下接口:

VDIF_SetPixel	设定像素点值。
VDIF_GetPixel	获取像素点值。
VDIF_RefreshDisplay	更新屏幕显示,设定像素点值后,需要调用 此函数以更新屏幕显示。
VDIF_ClearDisplay	清空屏幕显示。

此文件中定义的函数使用于 GUI 函数的 Basic 模块中,用于控制屏幕。 以上接口的使用和原理可以参照模拟环境工程中相应的示例代码。

# 10、 文件编码

在 SimpleGUI 中, 有关文字的处理都使用一个名为 ENCODE 的宏进行转码, 此宏定 义声明在 SGUI\_Common.h 中。

为了避免在不同语言环境或不同编码下产生乱码, SimpleGUI 所有的源代码文件均 以 UTF-8 格式进行编码, 但为了方便中文显示, 程序运行时对中文字符的解码全部按照 GB2312 编码规则进行, 所以需要将 UTF-8 字符串转码为 GB2312 字符串, 转码函数定 义在 SGUI\_Common.c 中, 依赖 iconv 外部库。

在实际使用中,开发者可以根据自身系统对字符的解码算法或使用的编码集,重写 SGUI\_Text.c 中的 SGUI\_Text\_GetCharacterTableIndex 函数以适应相应的编码规则,进而 实现正常的文字显示。

# 11、 联系开发者

首先, 感谢您对 SimpleGUI 的赏识与支持。

虽然最早仅仅作为一套 GUI 接口库使用,但我最终希望 SimpleGUI 能够为您提供一套完整的单色屏 GUI 及交互设计解决方案,如果您有新的需求、提议亦或想法,可以联系 QQ 326684221 或电子邮件 xuyulin91@163.com,也可以在以下地址留言:

SimpleGUI@开源中国: https://www.oschina.net/p/simplegui

SimpleGUI@码云: https://gitee.com/Polarix/simplegui

本人并不是全职的开源开发者,依然有工作及家庭的琐碎事务要处理,所以对于大家的需求和疑问反馈的可能并不及时,多有怠慢,敬请谅解。

最后,再次感谢您的支持。