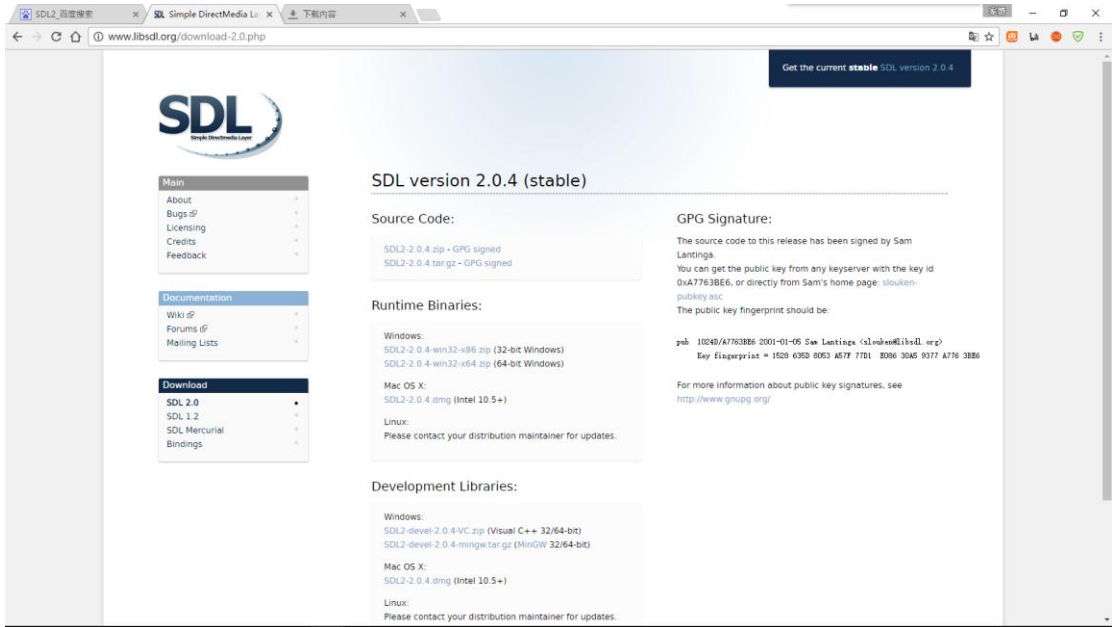





SDL2 安装配置说明书

1427405017\_沈家赟

1、 首先，打开 SDL 官方网站 (<http://www.libsdl.org/>);

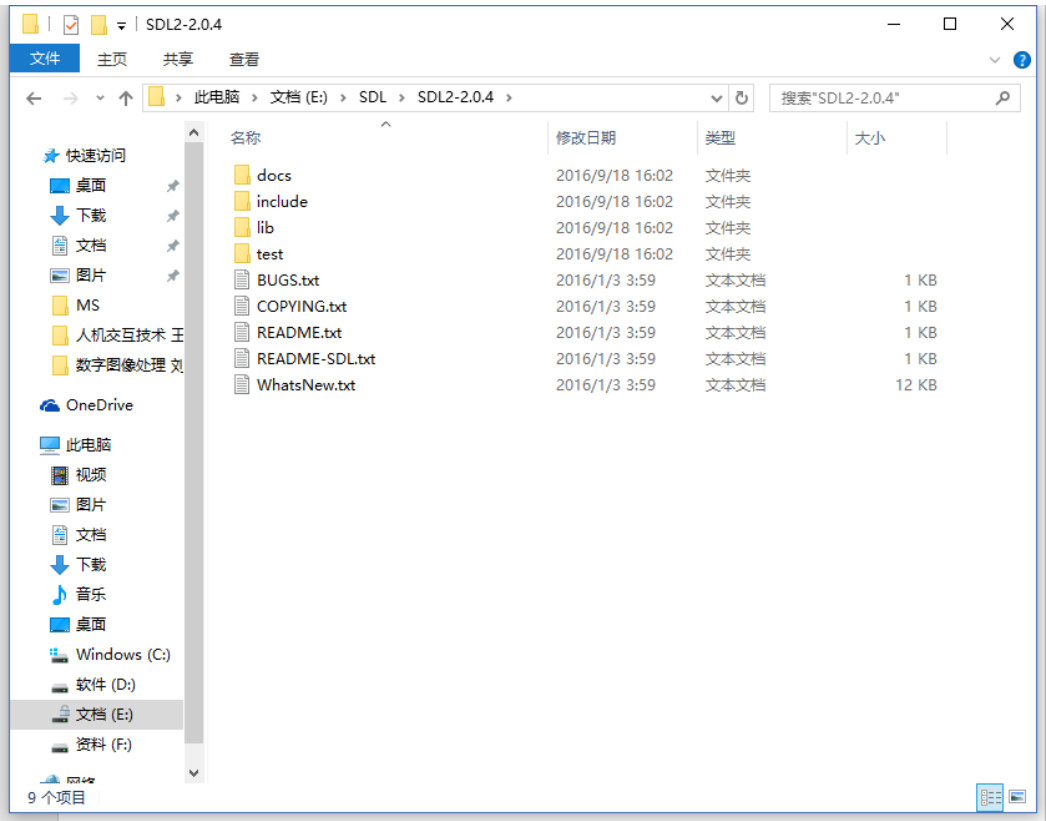


2、 下载相关文件;

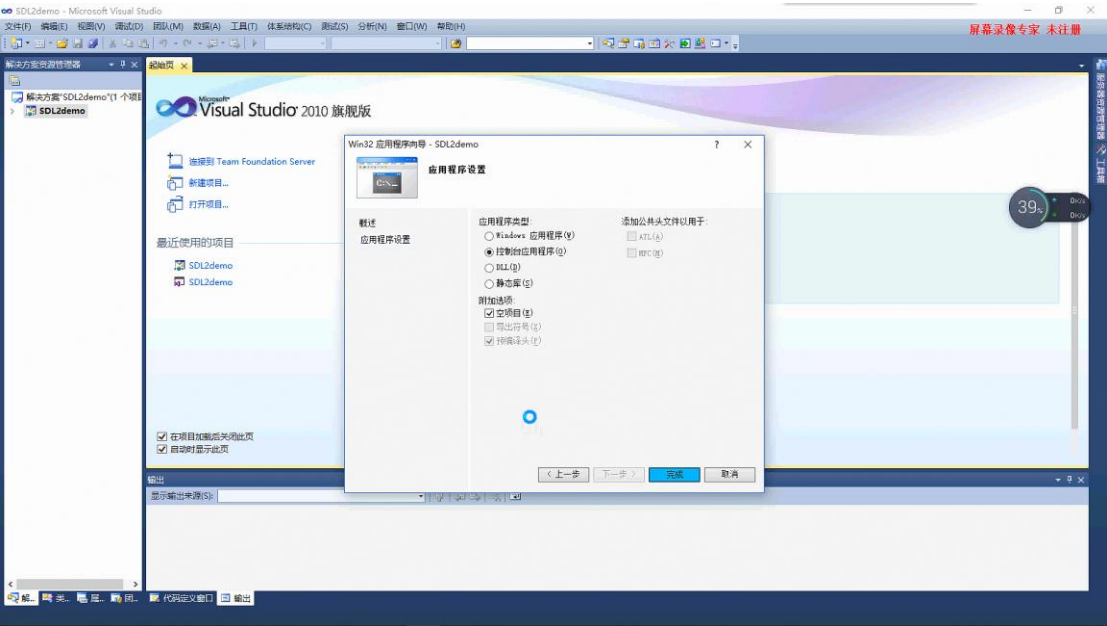
 SDL2-2.0.4.zip	2016/9/17 19:36	360压缩 ZIP 文件	4,957 KB
 SDL2-2.0.4-win32-x64.zip	2016/9/17 19:24	360压缩 ZIP 文件	464 KB
 SDL2-devel-2.0.4-VC.zip	2016/9/17 19:31	360压缩 ZIP 文件	1,692 KB

3、 解压 SDL2-devel-2.0.4-VC.zip 到 SDL-2.0.4 文件夹;

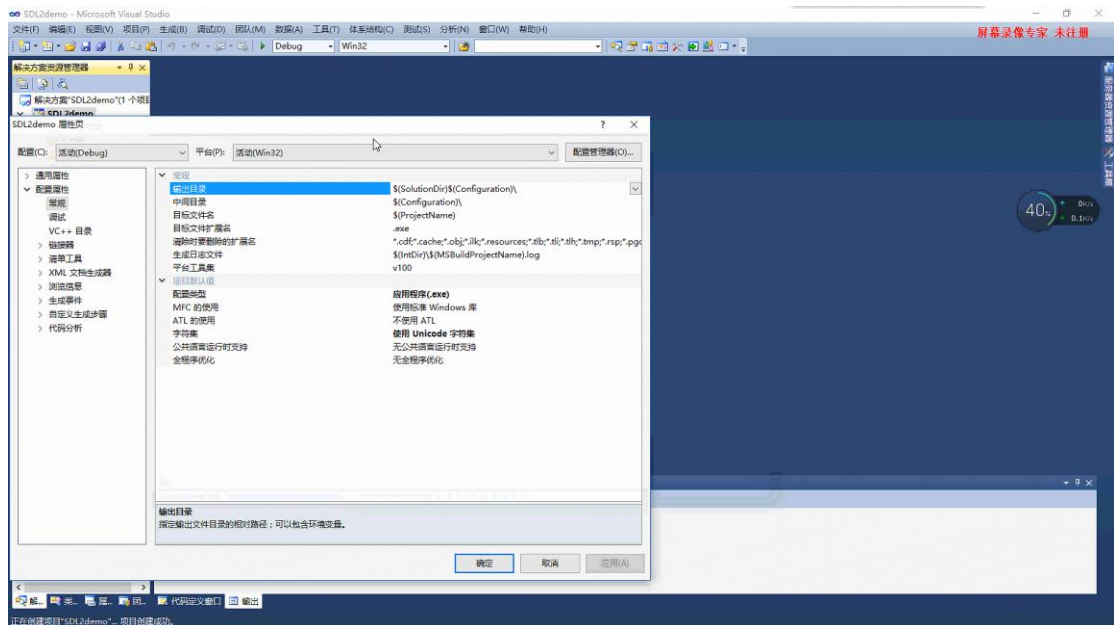
4、 解压 SDL2-2.0.4，将其中的 test 文件夹拷贝进 SDL-2.0.4 文件夹;



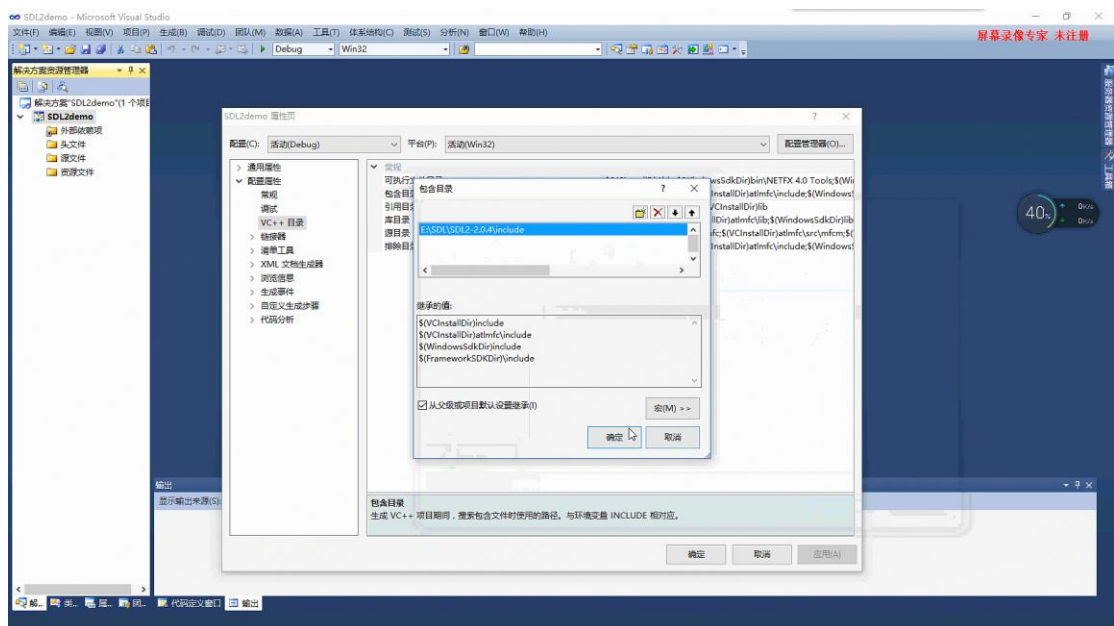
5、启动 VS2010，新建空项目；



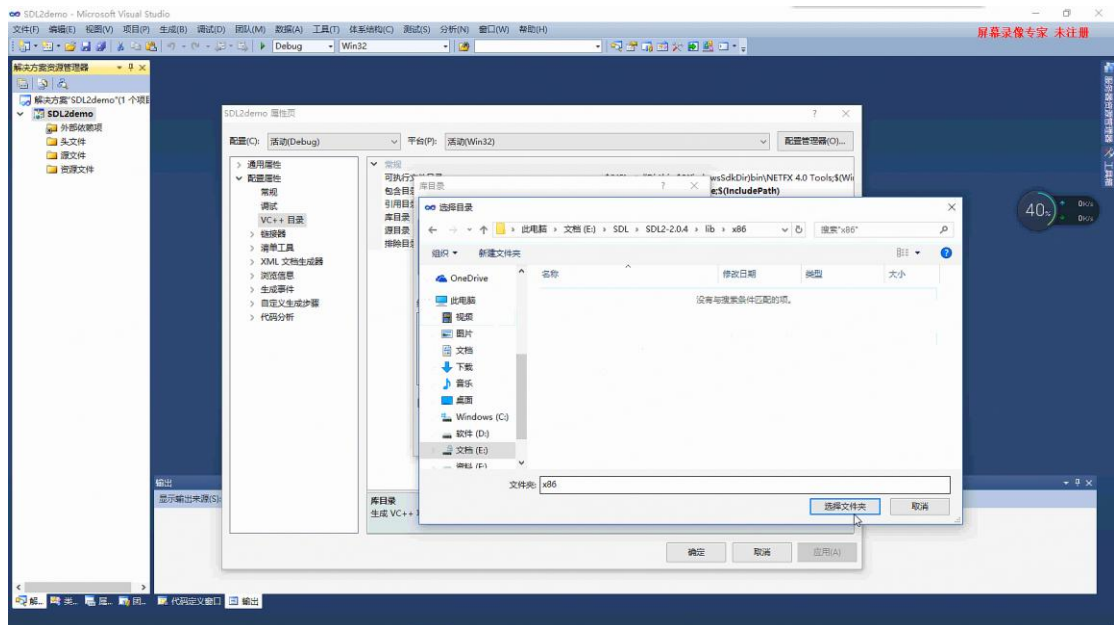
6、打开当前项目的项目属性；



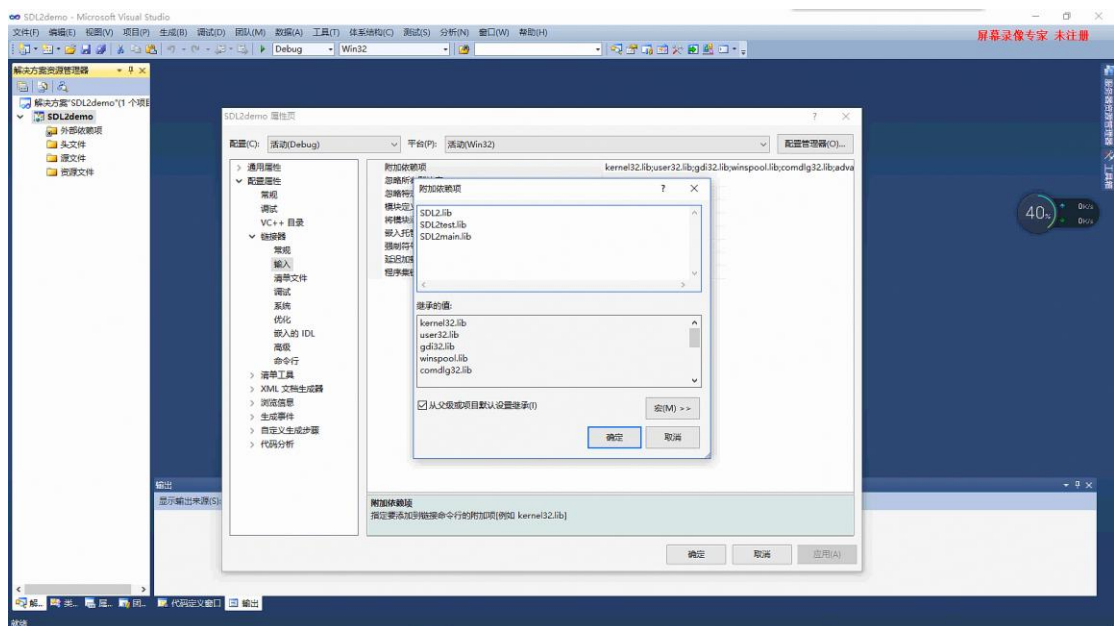
7、选择 VC++ 目录 -> 包含目录，将 SDL2-2.0.4\include\添加进去；



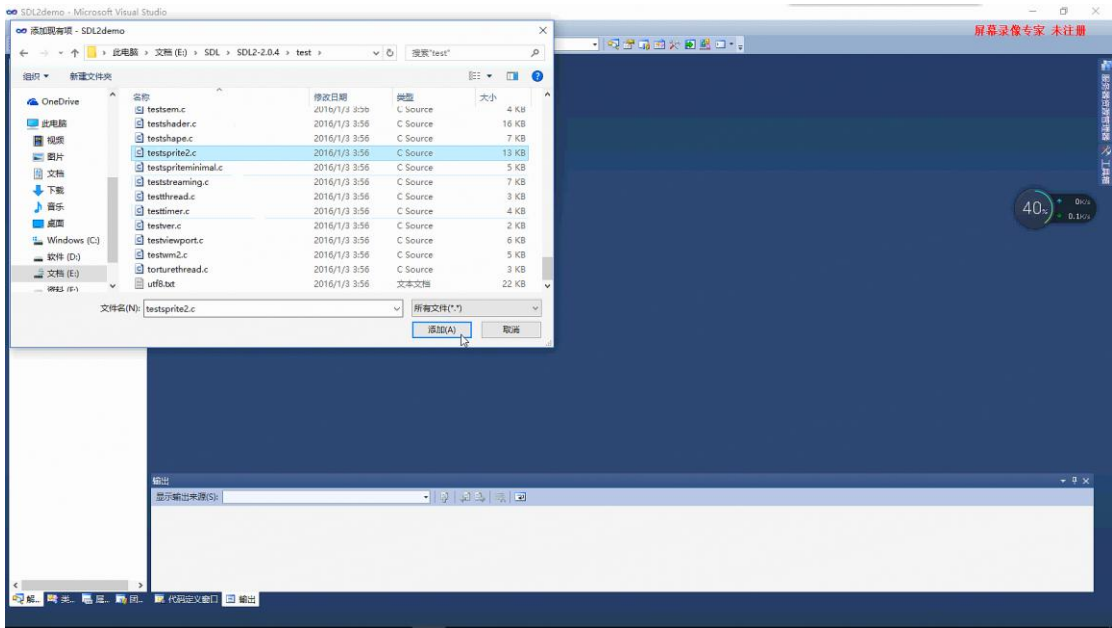
8、选择 VC++ 目录 -> 库目录，将 SDL2-2.0.4\lib\添加进去；



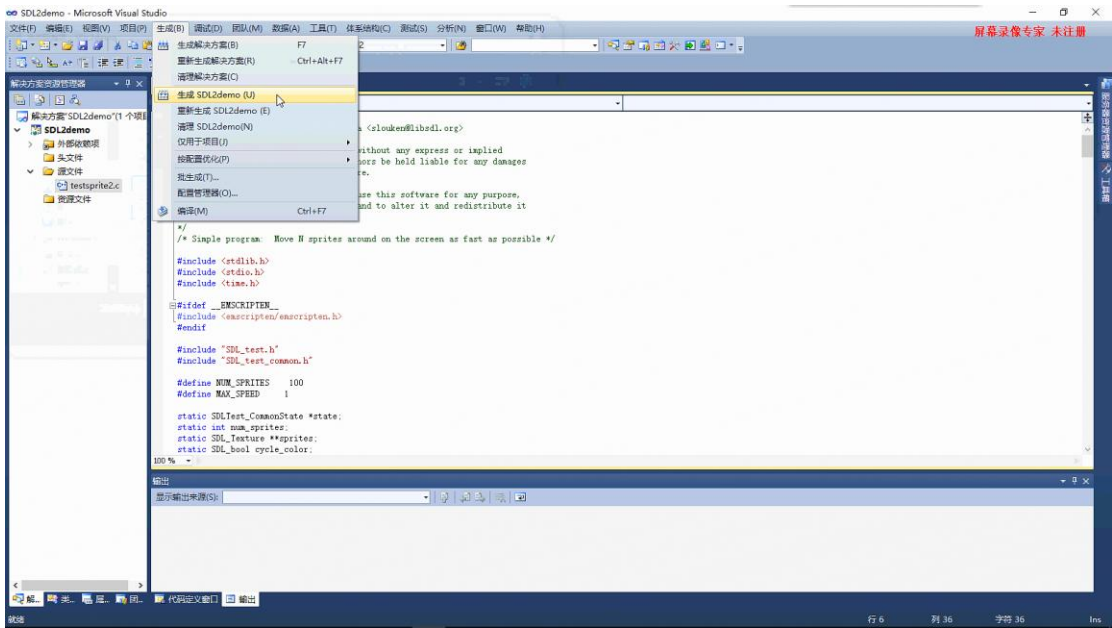
9、在链接器 -> 输入 中，选择 链接依赖项，输入如图三项依赖库：



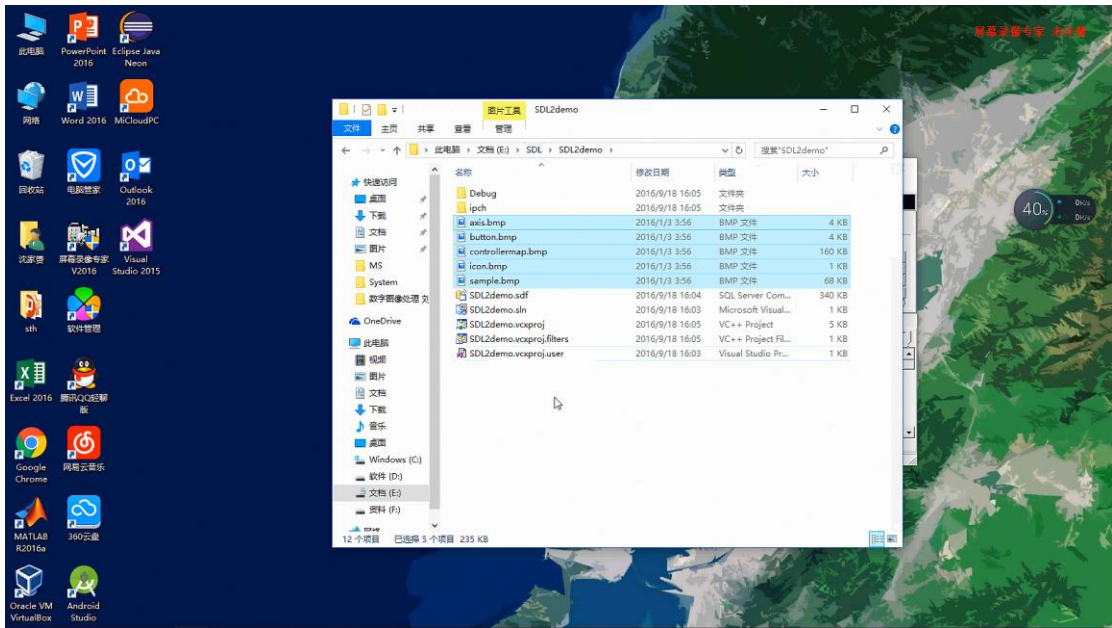
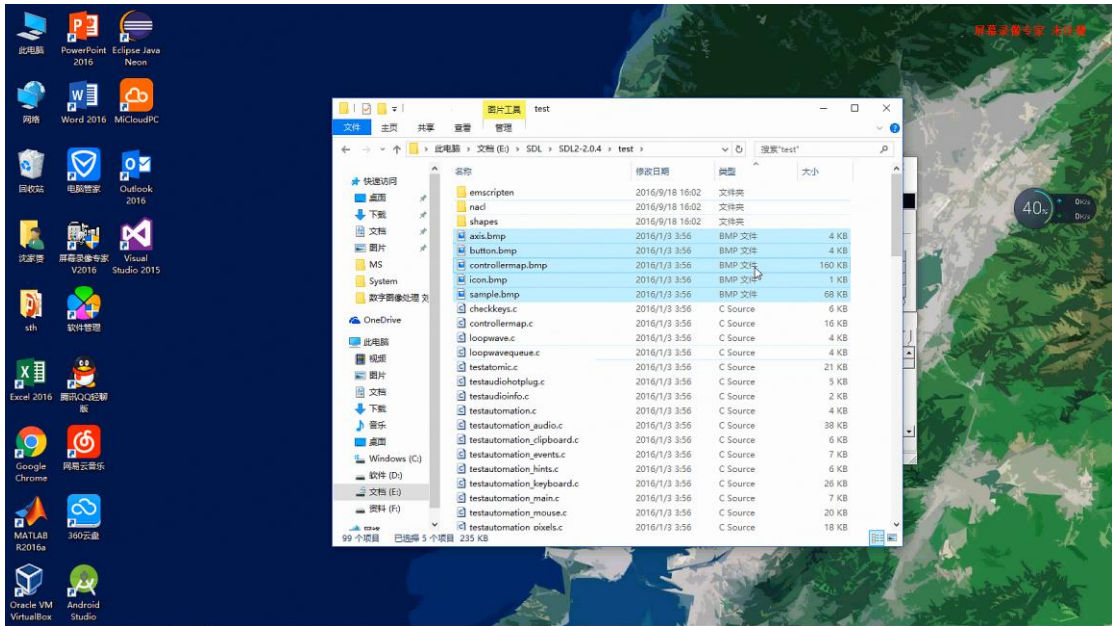
10、源文件，选择添加现有项，添加 SDL2-2.0.4\test 目录下的 demo 样例，如 testsprite2.c；



11、 点击生成项目；

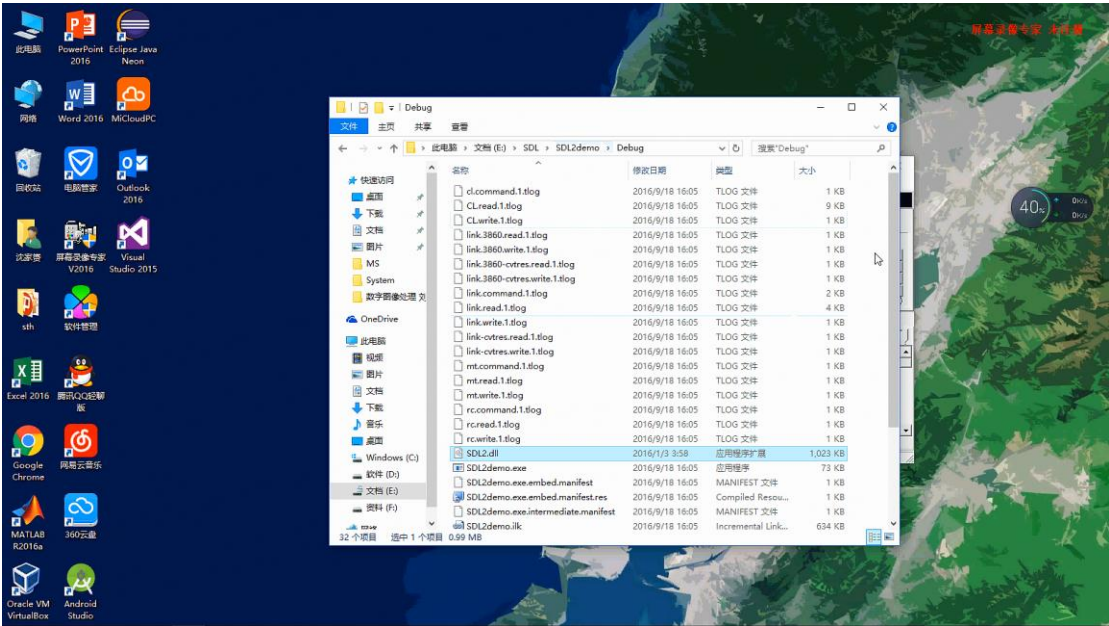
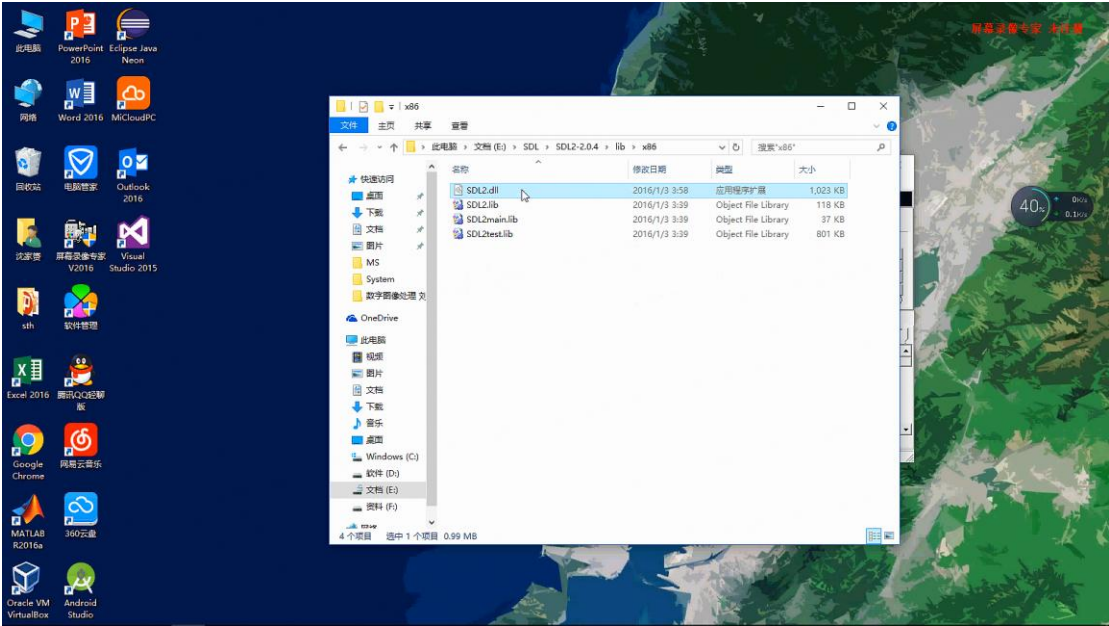


12、 将 test 目录下的图片素材拷贝至 VS 创建的项目目录下；

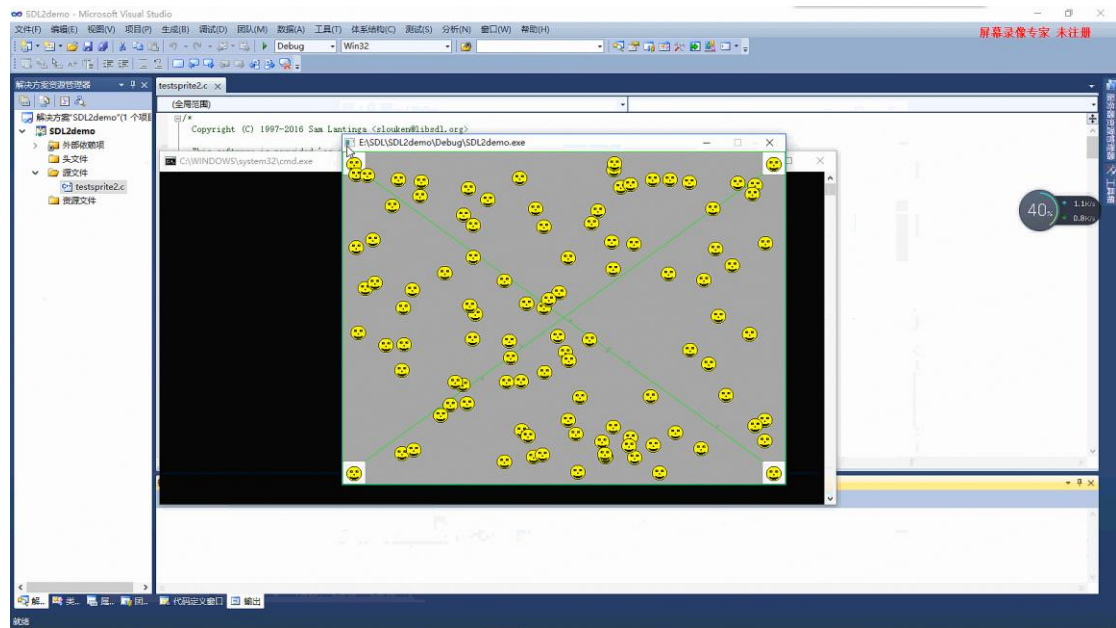


13、将 lib\x86 目录下的 SDL2. dll 拷贝至 VS 创建的项目目录的 Debug 文件夹内；





14、 此时即可成功运行该程序（VS 中使用 Ctrl+F5 启动）；





## QArt Codes 技术报告

1427405017\_沈家赞

## &lt; 概要介绍 &gt;

二维码是根据一定规则设计的一种二维的点阵画面，用于编码任意的文本字符串。通常用于编码 URL 链接，实现扫码登陆的功能。

二维码基于 Reed-Solomon error-correcting codes 原理编码，拥有一定得容错能力，对于特定的编码字符串，并非所以点阵都是严格固定的。因此，二维码的编码原理允许发挥个性，进行改变。

如，左图（“<http://bbc.co.uk/programmes>”）可以变更成右图，而不会影响编/解码的正确性。



QArt Codes 实现个性化定制的原理就基于此。

Demo 如下所示。





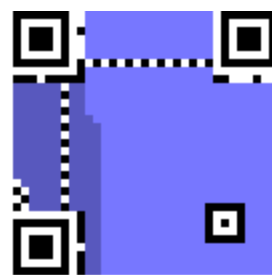
QR v1, L



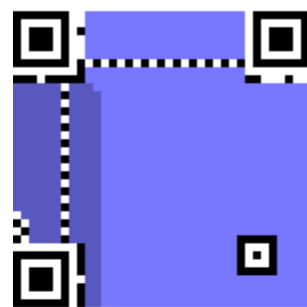
QR v2, L



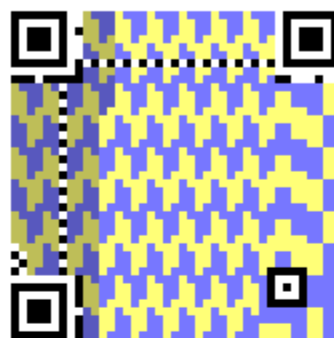
QR v3, L



QR v4, L



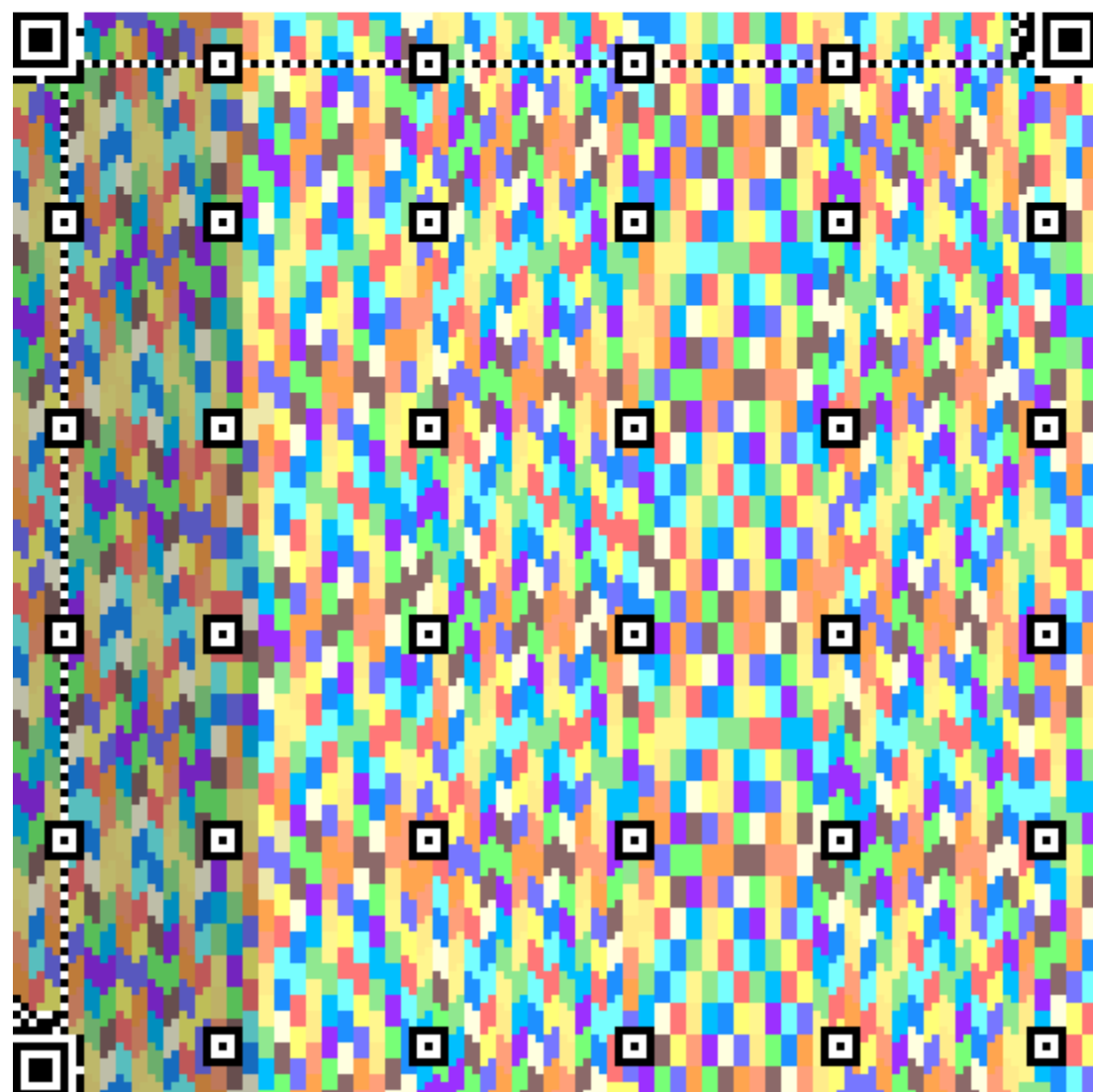
QR v5, L



QR v6, L



QR v7, L

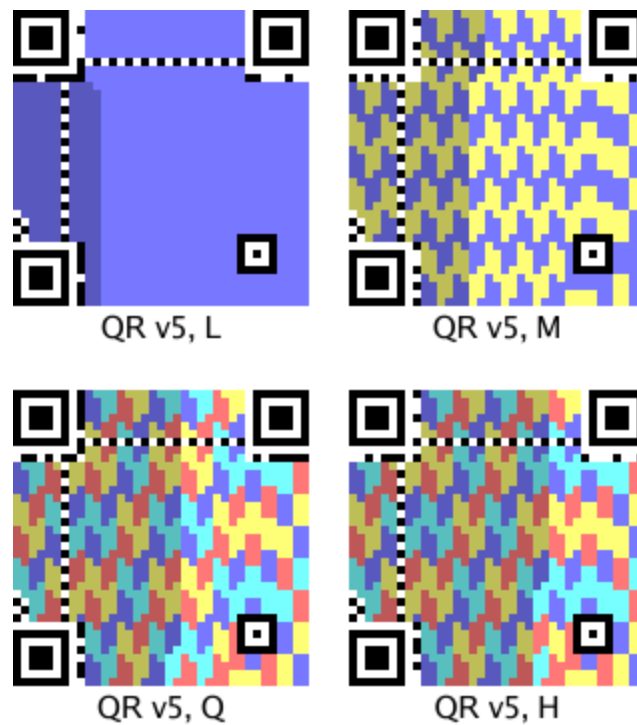


QR v30, L

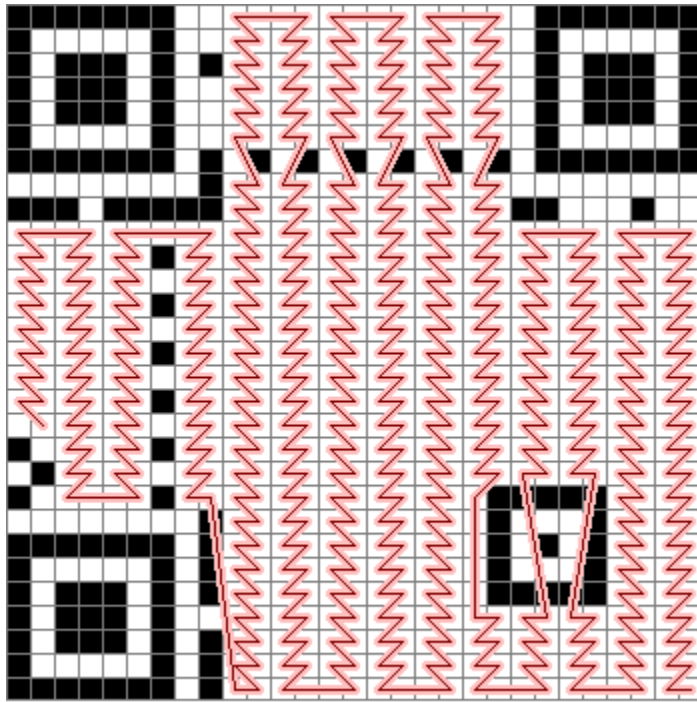
## &lt; 具体原理 &gt;

如图所示，彩色像素就是 Reed-Solomon-encoded 像素位。每个二维码都有若干个如图所示的像素位块，具体取决于错误修正的层级。

例如：L 级别表示最低的纠错冗余度，约 20%的纠错冗余码；M 约 38%，Q 约 55%，H 约 65%。

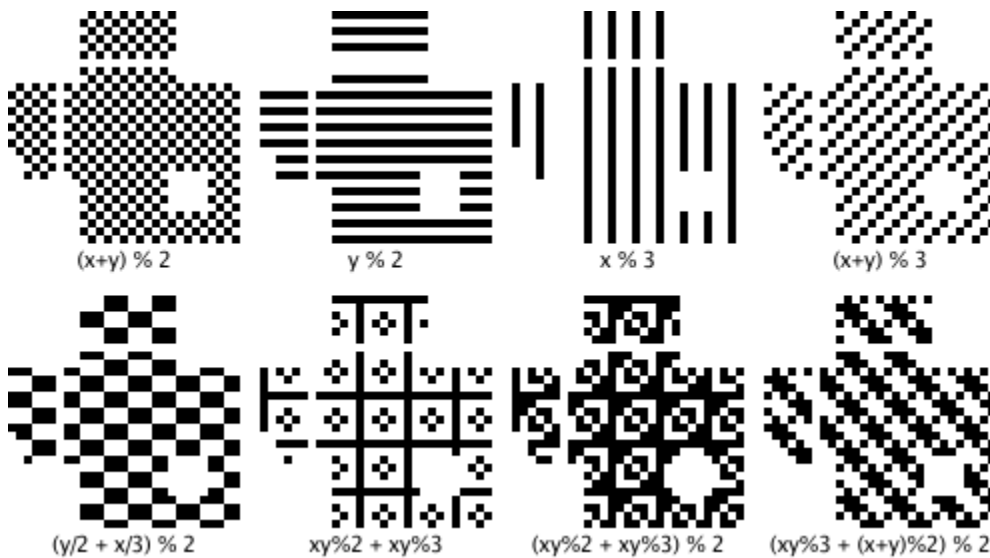


原始信息的数据位被直接用于 Reed-Solomon 编码，因此被编码数据的每一个位，实质上都对应对应着二维码中的一个像素。如上图所示，亮色区域就是编码数据位对应的像素区域。而暗色（阴影）部分，就是错误校验码对应的像素区域。



QR v3 order, from bottom right

具体的编码顺序如图所示。图示为第三代标准的二维码编码顺序，低位数据从右下角开始编码。



QArt Codes 的实质其实是利用二维码具有较多冗余校验码，解码器可以自动实现错误识别和校正修复，最后还原正确的原始信息的原理。QArt Codes 一定程度上破坏原始信息数据对应的编码位，根据自定义函数或映射，设置个性化的图案蒙版，根据需要破坏不符合需要的像素位，以此实现自定义、个性化的二维码图案。

QArt Codes 其实是与二维码结果设计的初衷相违背的，它破坏了二维码原本的原始数据+冗余校验及错误修正的严格设计规范，而是利用其较高程度的错误冗余设计来实现“破坏式”的编码。

## &lt; 小结 &gt;



此案例中，灰色阴影区域表示被占用的像素位区域。具体而言，左边的阴影部分是错误校验码的编码区域，右侧的阴影区域则是原始信息占用的编码像素区域。虽然根据自定义的蒙版有所“破坏”，但仍然可以在解码时通过左侧部分的错误校验码实现恢复。因此，这样的二维码仍然是可以正常扫描并被正确识别出来的。

简而言之，QArt Codes 就是充分利用未被使用的像素区域并可根据需求破坏一定程度的信息编码像素，通过较多的冗余错误校验编码实现信息的准确恢复。