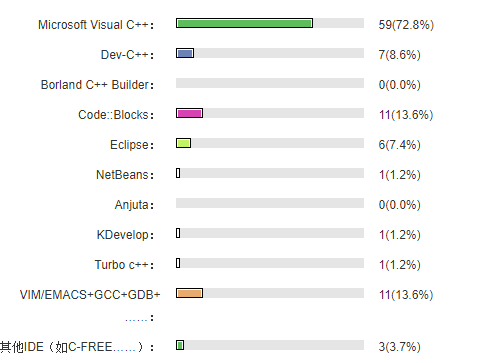
不同于网上各种奇形怪状的只使用Visual Studio编辑器代码补全功能，或者复杂的需要建立交叉编译链的方法，本方法过程简单，无需导入固件库，并且原生支持3.5固件库以及最新的HAL库，FreeRTOS系统等，支持J-link、ST-link等所有常见下载器，是居家旅行,杀人灭口的必备方法，反正我没发现我们学校或者其他学校的有人用过。

哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈嗝~

## Visual Studio

如果你没听说过Visual Studio，请先读完高中再来（手动微笑），如果你已经安装了Visual Studio最新版（Visual Studio **2015** or later），**请跳过本节**。

Visual Studio是目前**最流行**的Windows平台应用程序的集成开发环境。最新版本为 Visual Studio 2017 版本，基于.NET Framework 4.5.2 。以上摘自百度百科，那个红字不是吹，在真实开发中，10个开发者有7个用Visual Studio，为啥？因为好用啊，比IAR、KEIL好了不知道多少条街，以下是贴吧一个统计，意思下：



笔者用的是VS2015社区版，之所以下载这版，是因为这应该是唯一的**正版**免费版，其他的貌似都要钱，当然你非要破解的最新版当我没说。

## VisualGDB

这玩意是Visual Studio的一个插件，有了这个就可以在线调试了，安装包以及破解在压缩包里。

1. 安装VisualGDB-5.2r8-trial.msi，一路Next
2. 使用PatchVisualGDB.exe进行破解

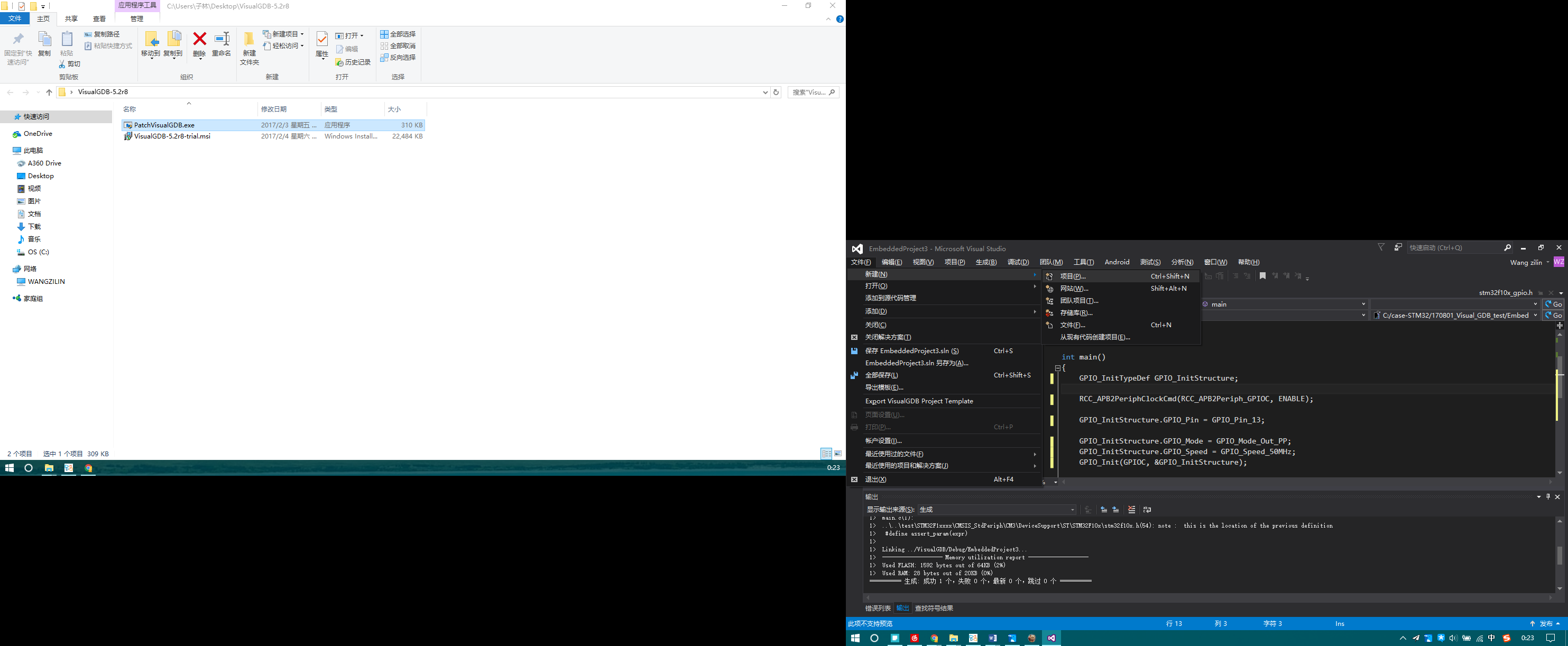
过程非常弱智

1. 打开Visual Studio，别看到英语就恶心，四级就能看懂，如果你破解成功了，他上面会有一个你的剩余天数，我的是10000多天，够用30年了，那就说明破解成功，点下面选项卡**第二个（因为我已经破解成功，而那个界面只有在第一次打开时才会出现，所以以上我全凭印象，没有截图）**
2. 之后会有一个问你什么时候显示代码补全的对话框，我把第一个改成了立即提示，不需要按按键，左下角有个打钩的选项，问你下次要不要再出现这个对话框，我打了勾（也正因为如此，这一条也是我凭印象写的，+1s）

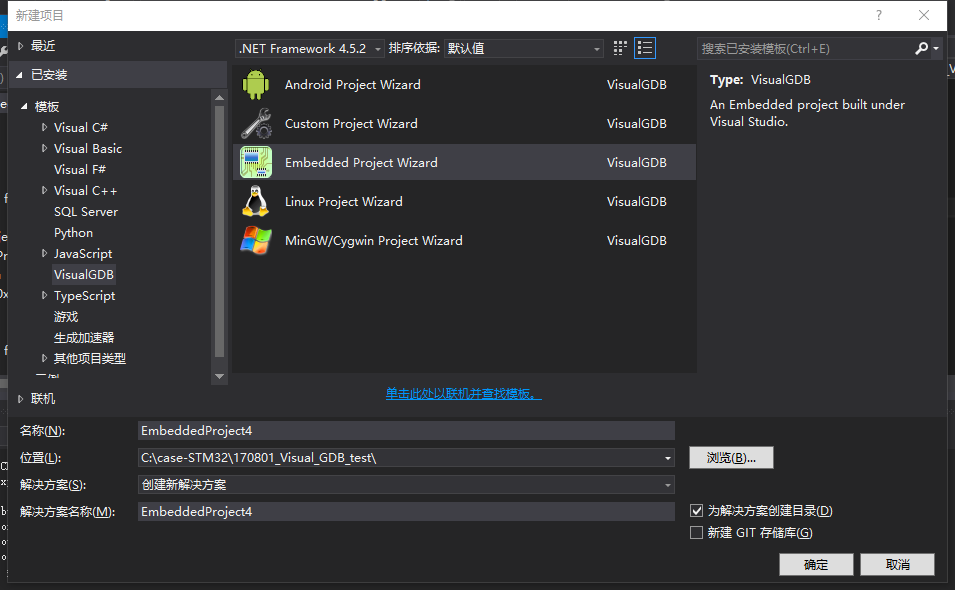
## 新建STM32工程

* 我使用的是STM32F103C8T6，就是那块裸的小板子；
* 要保证你的网络状态很好，很好指的是，既能上国内，也最好能翻墙（并且把VPN设成全局模式，并且不要频繁切换线路），因为刚刚那个插件只是一个框架，各种针对芯片型号的支持文件是需要单独去下载的，下载的网站都在国外，我尝试过翻墙和不翻，速度能差将近5倍。

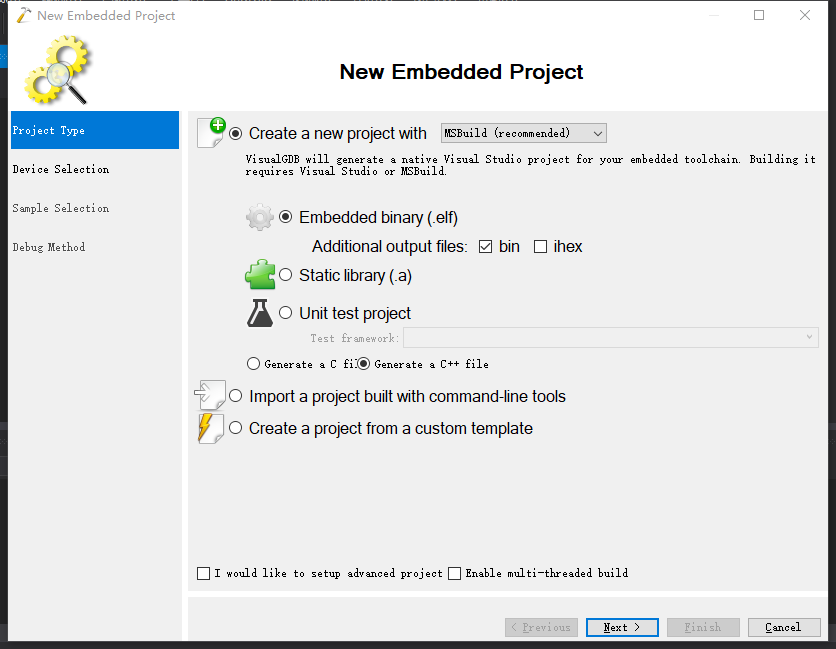
1. 新建项目：



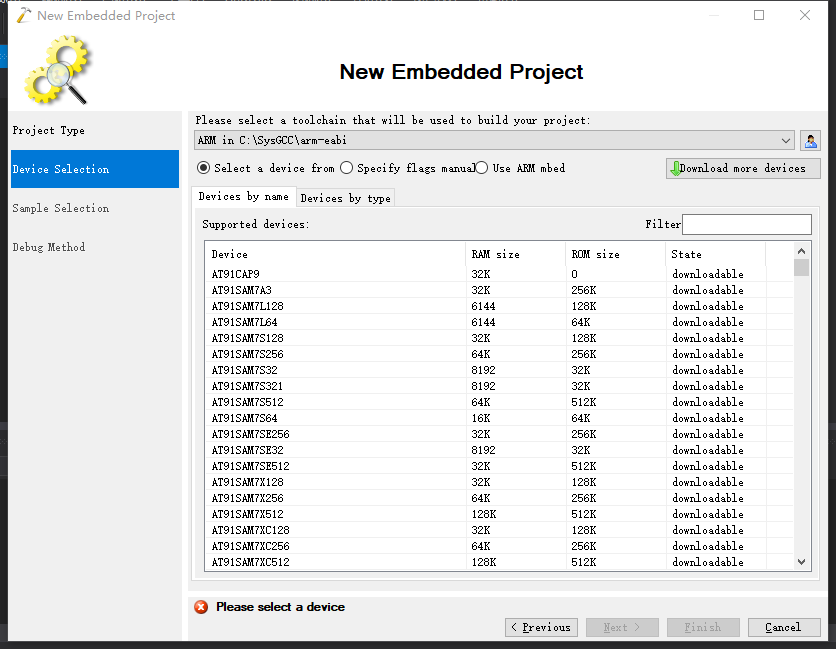
1. 选择工程路径：**（路径不要有中文，这插件会报错！！！！！！）**



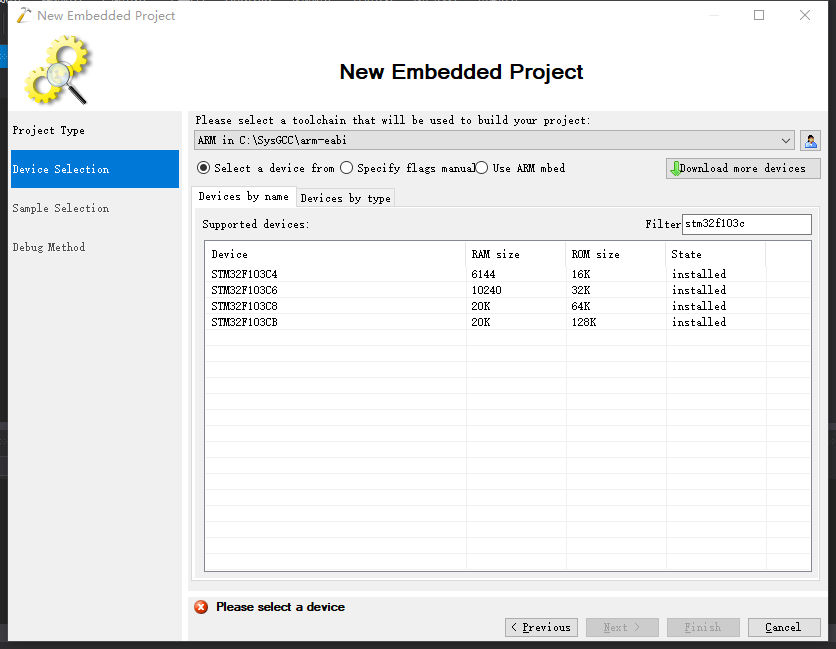
1. 创建C++文件，不要创建C文件，创建C文件会缺少代码高亮，代码会好丑（还是C语言太老了，新的IDE支持C的越来越少了，还好C是C++的真子集）



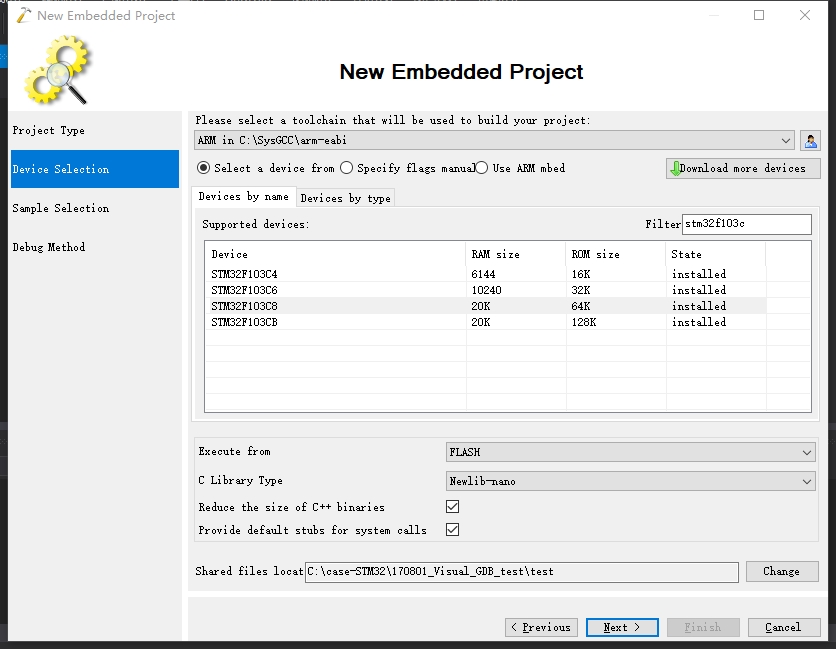
1. Next，选择ARM家的，会弹出下载窗口，打钩，慢慢等下载吧。



1. 在fileter中搜索芯片，下载好了应该就像这样：

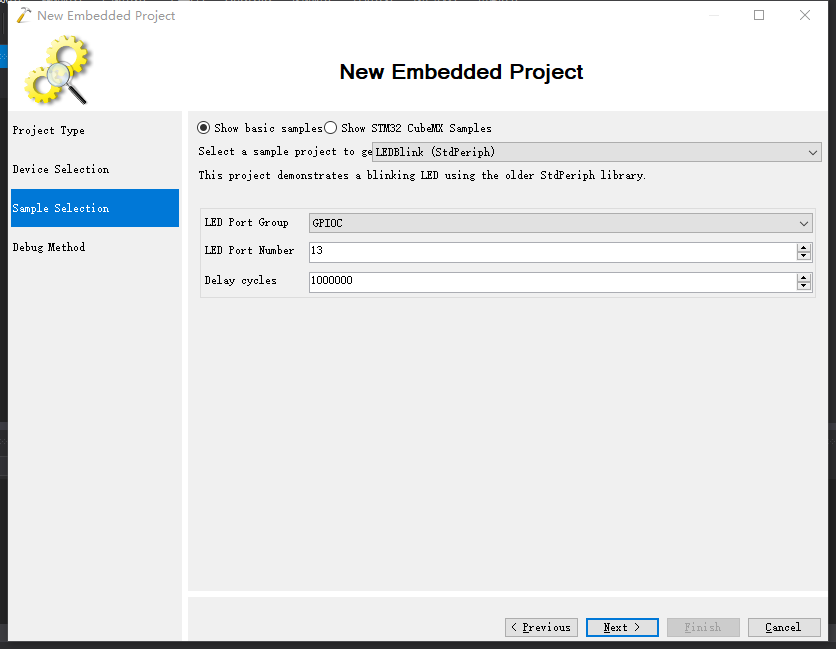


1. Next：下面红色框是分享文件路径，里面是各种支持包**（路径不要有中文，这插件会报错！！！！！！），修改的话一路确定即可**

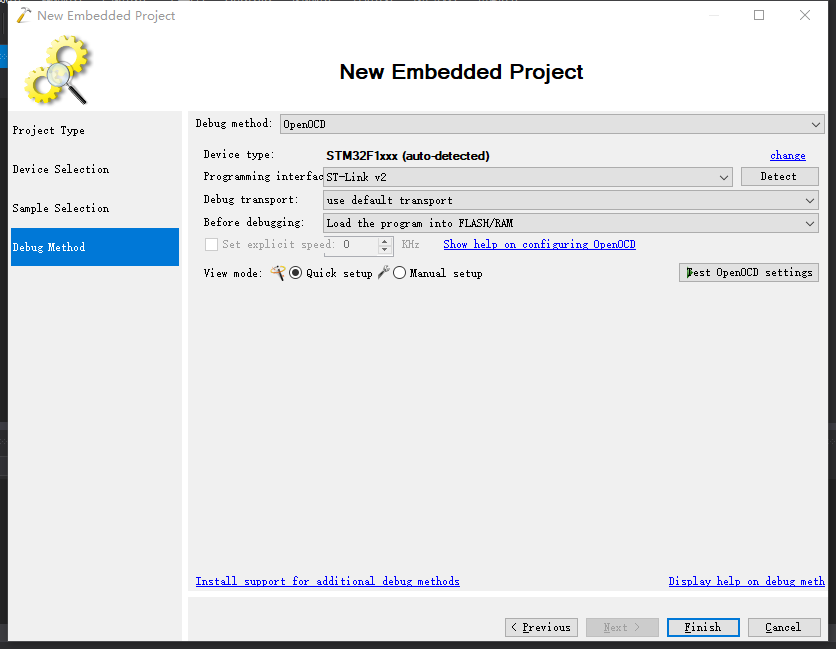


1. Next：

这步是**最最最最最厉害的**：它让你选库，我们最常用的是3.5标准库StdPariph（但HAL库是趋势，之后你们再学），选择指示灯引脚：GPIOC-13，延时那里不用管



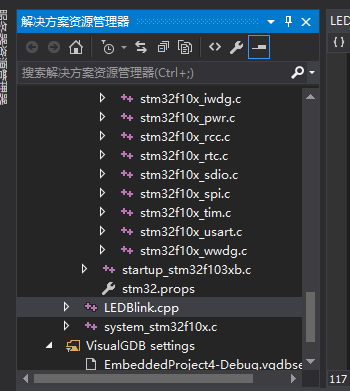
1. Next：选择调试环境：
   1. 我们用的是ST-link，不是ARM的亲儿子，所以默认选项里面没有，需要点击OpenOCD，下载第三方下载器包，继续慢~~~慢~~~~~等~~~~~~~~~~
   2. 下载完，选择ST-link V2，其他不用改，默认就是SWD下载



1. Finash，耐心等待工程建好。

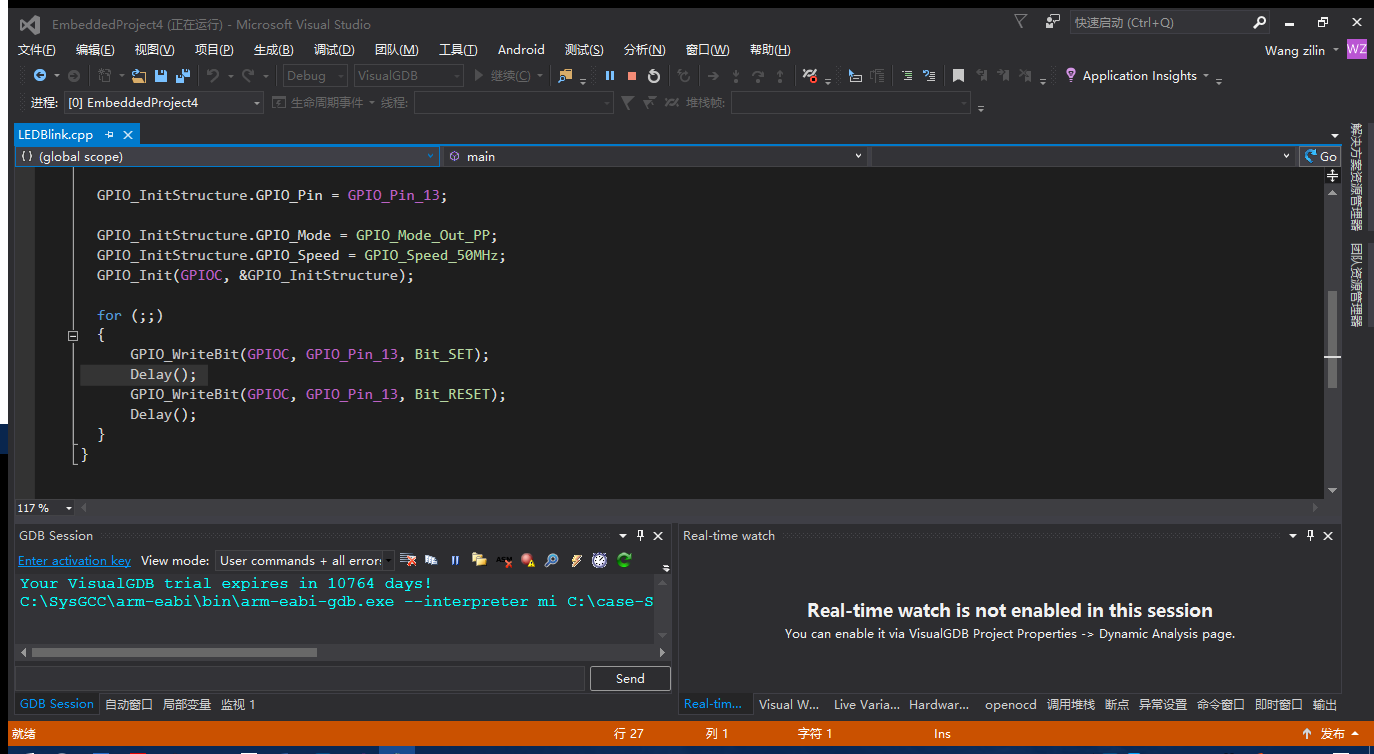
## 在线调试：

1. 在解决方案资源管理器中找到LEDBlink.cpp
   1. 解决方案资源管理器找不到的话Ctrl+Alt+L打开（视图选项卡中找）
   2. 如下图：



* 1. 这时候应该就已经有写好的程序了

1. 生成——生成解决方案（ctrl + shift +B，我的和搜狗冲突了）
   1. 我第一次建立工程时这一步报错，之后再建立工程就没问题了，如果这步报错，但是下一步没问题的话，不用管他，要是下一步也不行，检查路径是否有中文
2. 在线调试工程 中间那个绿色三角（F5）：
   1. 如果只想下载不想调试，Ctrl + F5（比IAR不知道高到哪里去了）



1. 暂停、设置断点等与IAR原理一样，操作习惯不同，不再赘述

## 错误解决

### 复制基于HAL库的FreeRTOS工程到别处再打开时出现错误：

VisualStudio其实对中文支持的挺好的，但是他的插件就不一定了，当移动目录到无中文名后问题解决

### void \_\_attribute\_\_((naked, noreturn)) Default\_Handler()

这个错误与产生的原因是找不到指定的中断处理函数，但是我已经定义了相应的处理函数，那是为什么呢？

由于VS的工程都是c++文件，而c++是支持函数重载的（即一个函数名可以输入不同种类的变量类型），c++编译器的处理机制是给每个重载函数都起一个别名，这也就导致了找不到中断函数入口。

解决方案：

在文件头使用以下方式声明中断函数：（所有用到的中断函数都要）

**extern "C"** void EXTI0\_IRQHandler();

**extern "C"** void SysTick\_Handler();

### -std=C99报错

出现asm（…）未定义错误：

将**该文件**中的asm（…）全部替换为\_\_asm（…）