# 3ch\*\*\*Visual Studio 2017 配置QT5.12.1开发

[Manhattan1](https://blog.csdn.net/yelang10000" \t "https://mail.163.com/js6/read/_blank) 2019-02-19 17:46:25  2004  收藏 6

分类专栏： [windows UI](https://blog.csdn.net/yelang10000/category_8682010.html" \t "https://mail.163.com/js6/read/_blank) 文章标签： [Qt](https://www.csdn.net/tags/MtzaIgwsMzM0NDctYmxvZwO0O0OO0O0O.html" \t "https://mail.163.com/js6/read/_blank) [Vs2017](https://www.csdn.net/tags/MtjaIg1sNTc4NTktYmxvZwO0O0OO0O0O.html" \t "https://mail.163.com/js6/read/_blank) [UI](https://so.csdn.net/so/search/s.do?q=UI&t=blog&o=vip&s=&l=&f=&viparticle=" \t "https://mail.163.com/js6/read/_blank)

版权

本文介绍基于VS2017配置QT

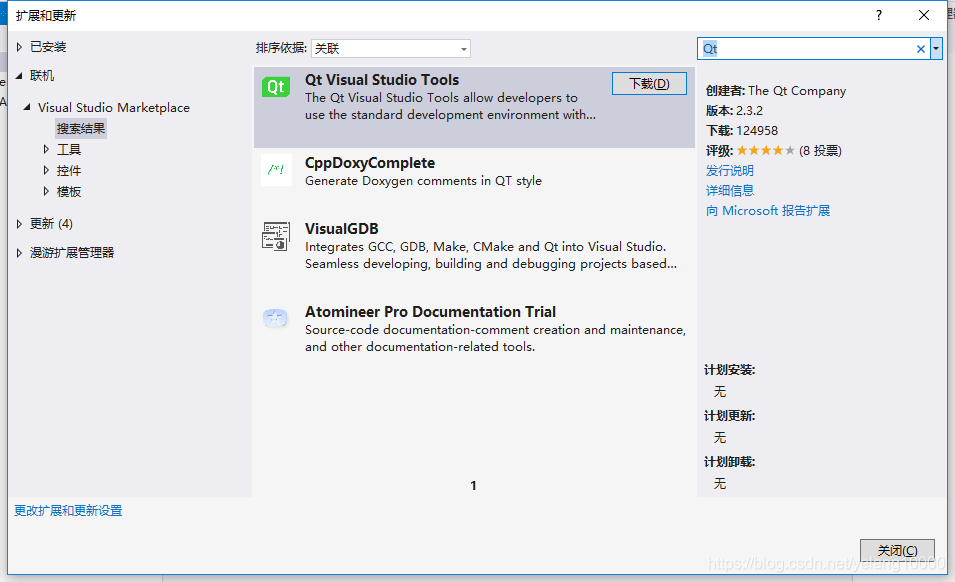
工具及系统：VS2017，Qt5.12.1

## 一、前期准备

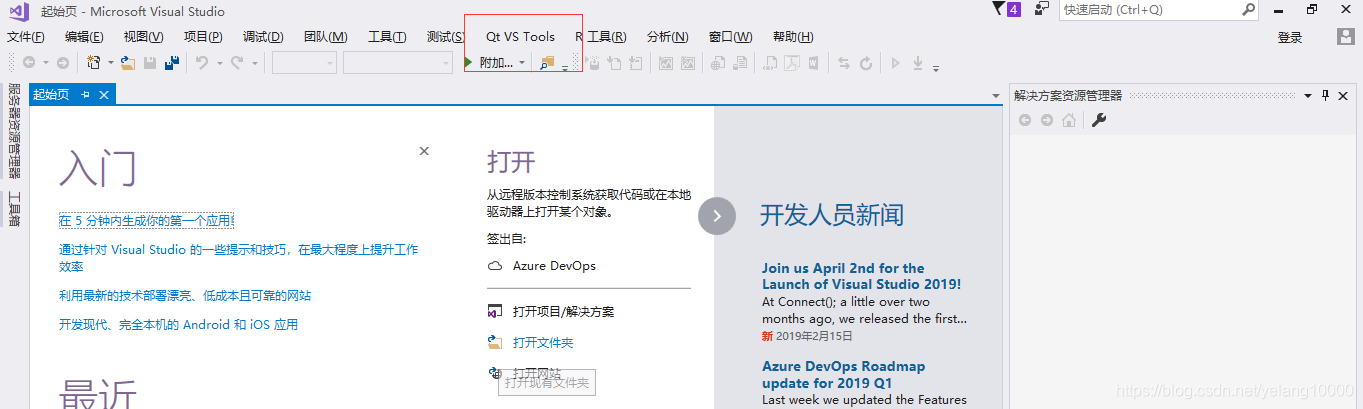
### 1、下载VS2017,并安装。

### 2、QT插件安装

打开vs2017 ，点击工具--扩展和更新---联机

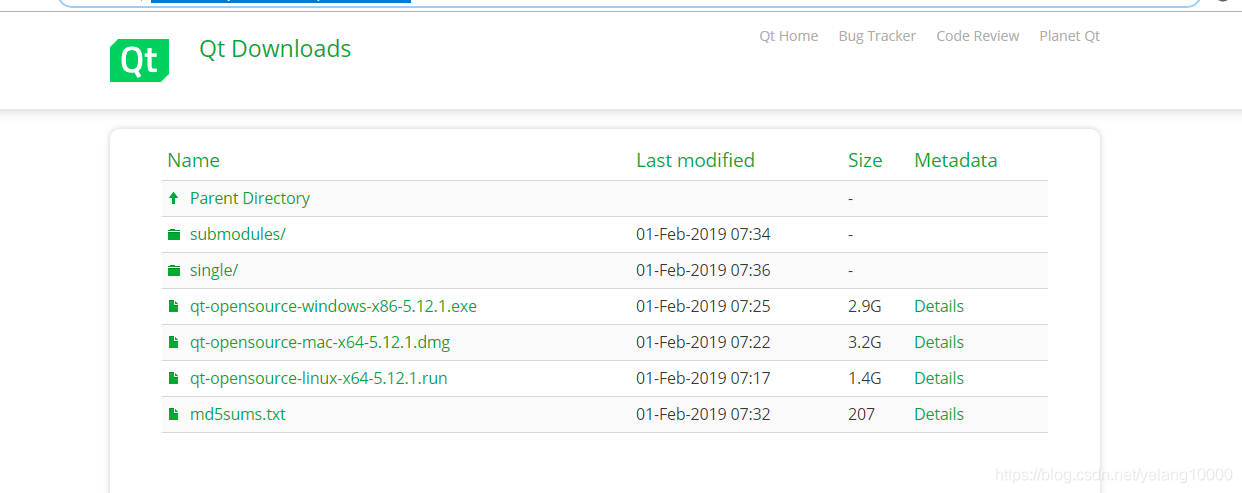


### 3、安装完成以后就可以在工具栏里面会有Qt VS Tools选项



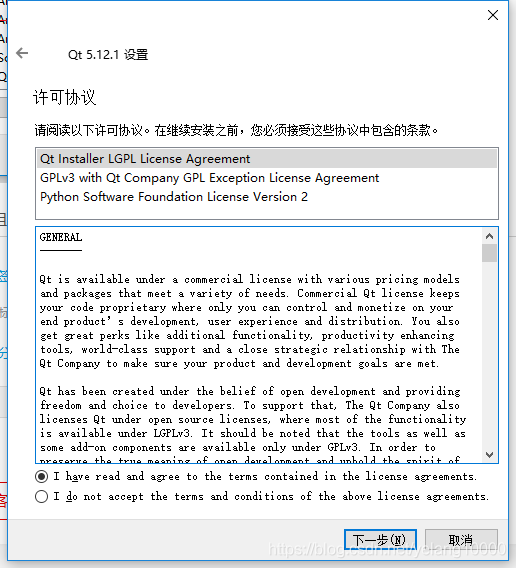
## 二、QT5.12下载安装

方法一：下载地址：<http://download.qt.io/archive/qt/5.12/5.12.1/>

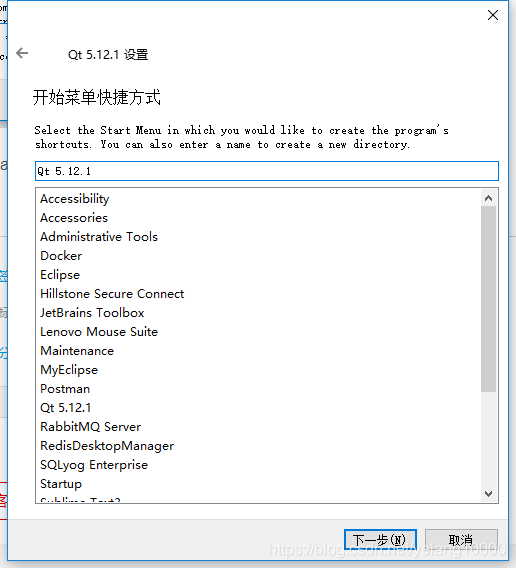




选择自己需要的组件点击下一步



选择 I have read and agree to the terms contained in the license agreements

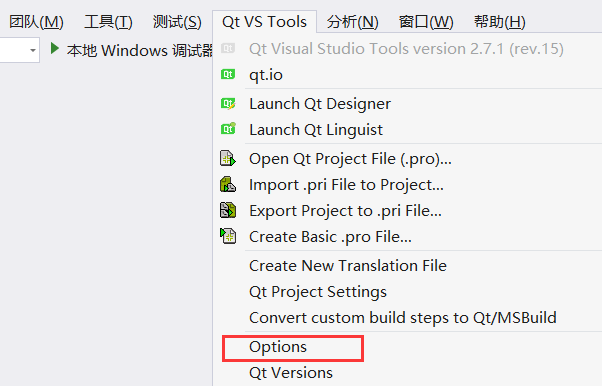


下一步：

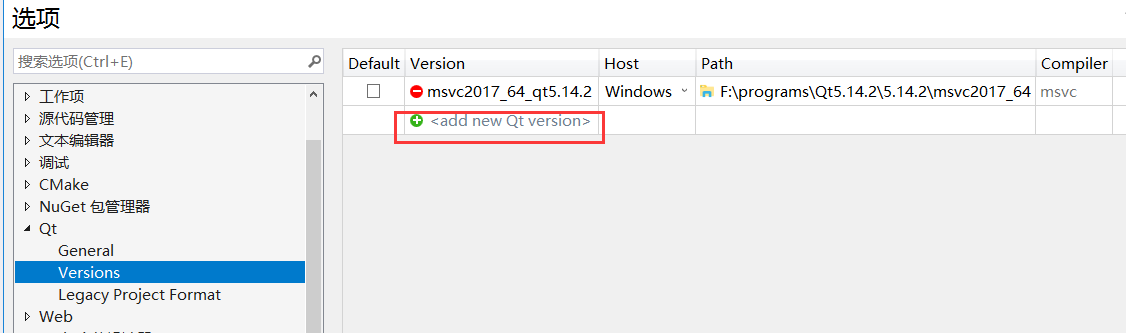
直至出现安装完成。

## 三、VS2017配置QT

1、打开vs2017,点击Q't VS  Tools选择 QT Options



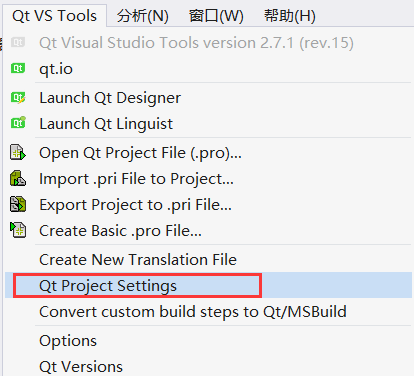
2、点击 add new Qt version，添加自己Qt安装路径



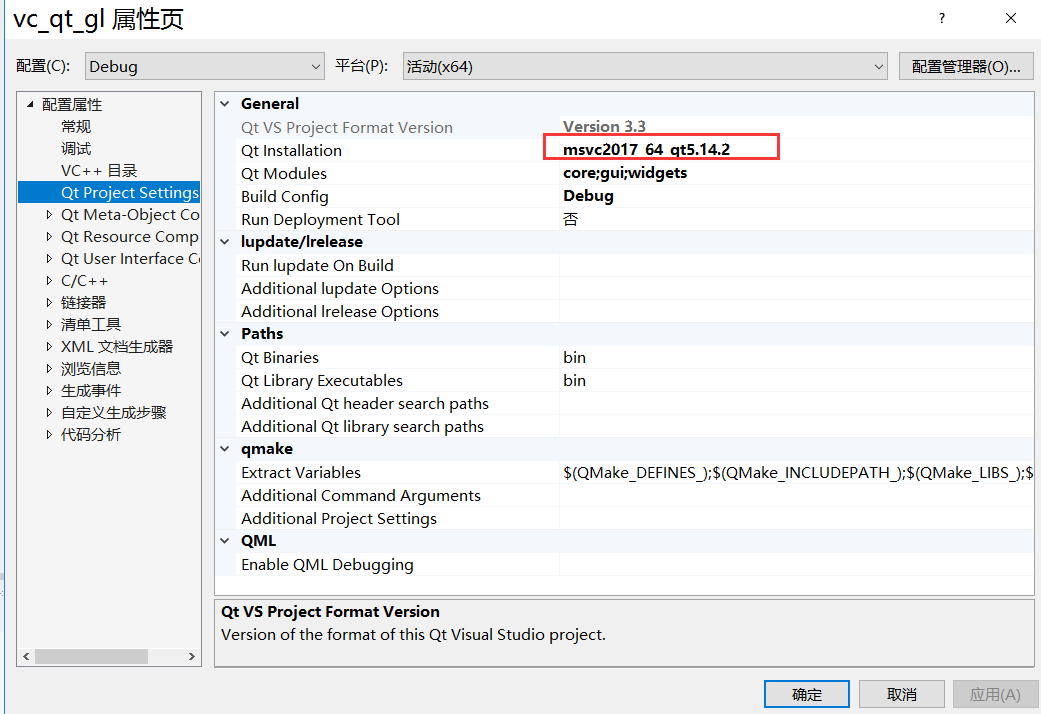
添加版本名字（如 msvc2017\_64\_qt5.14.2）；添加路径，找到你之前安装Qt的路径，比如我的就是F:\programs\Qt5.14.2\5.14.2\msvc2017\_64 。

3.配置现有工程项目

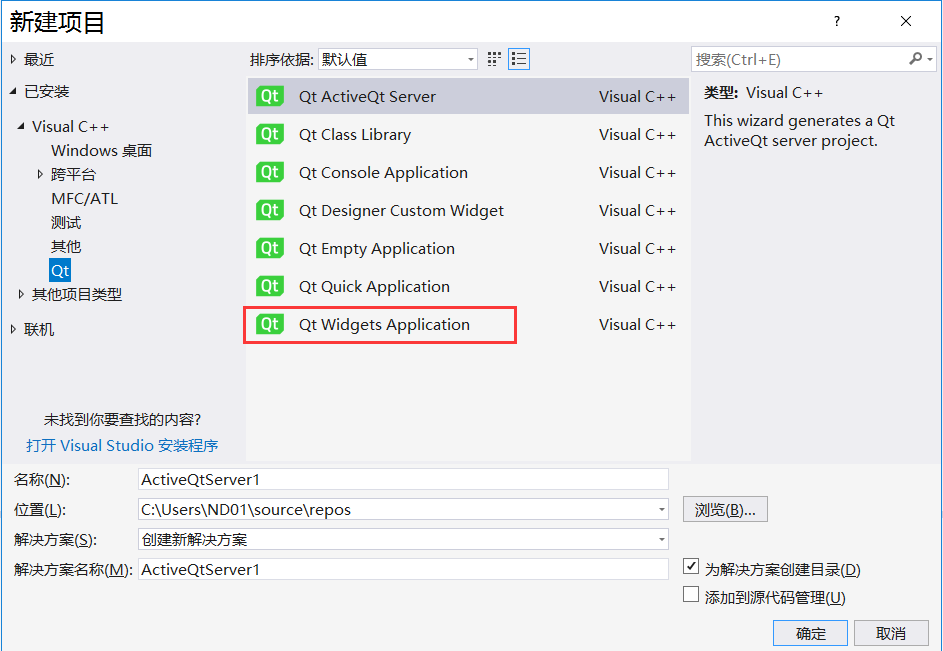
然后类似的打开项目设置，这样才会给你的项目加载Qt，不然会报错There's no Qt version assigned to this project for platform x64. Please use the 'change Qt version.



点击去，一定要确保Version版本是有内容的，且是你刚才创建的那个。



然后在新建项目的界面中就会有Qt的项目了



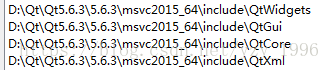
 打开后成功执行就大功告成了！！！

但也别高兴太早，且听下回分解~~~

上面的配置的确能用Qt了，但我在跑其他人代码的时候，还是报一大堆错误呀，什么缺失文件，找不到库，无法解析.......真是让人头大，但配置OpenCV的过程我是记忆犹新的，所以我觉得可以类比一下。

打开项目的属性设置

1.找到C/C++下的常规选项，在附加包含目录中添加这些文件（需要注意的是，这是我所用到的一些包，如果你的程序包其他错误，你就找到是属于哪一个包里面的，然后添加进去就好了）



2.找到链接器下的常规选项，在附加库目录中添加

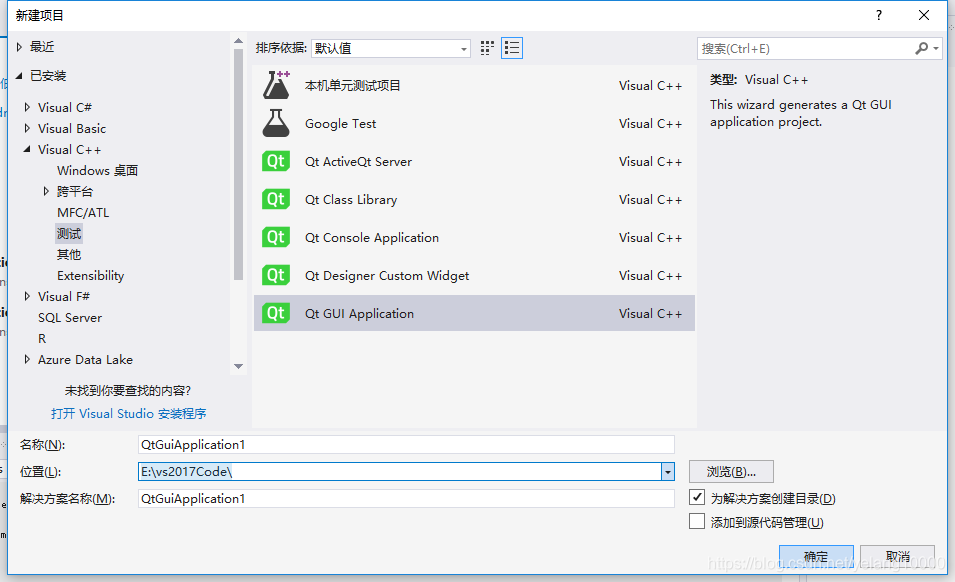
IMG_268

3.找到链接器下的输入选项，在附属依赖项中添加（这里也是添加opencv的opencv\_world342.lib的地方）

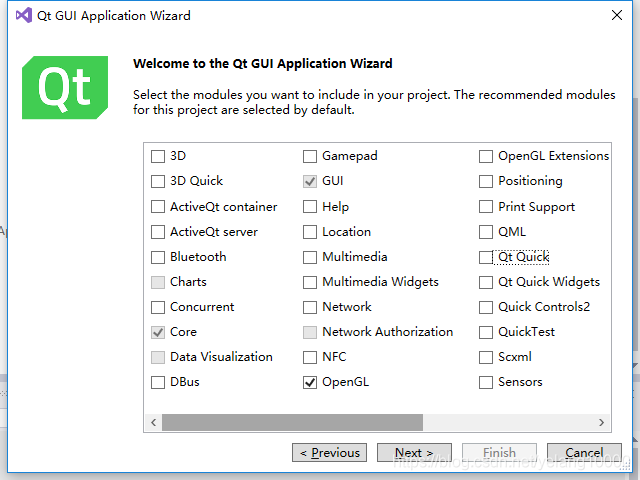
Qt5Core.lib  
Qt5Cored.lib  
Qt5Gui.lib  
Qt5Guid.lib  
Qt5Widgets.lib  
Qt5Xml.lib

## 四、测试程序

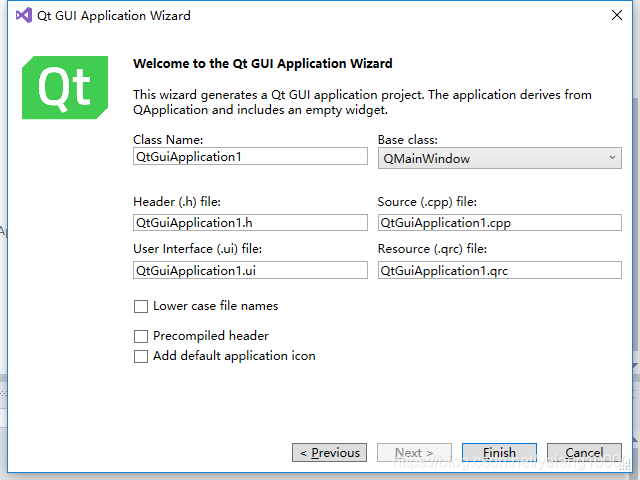
1、新建QT项目



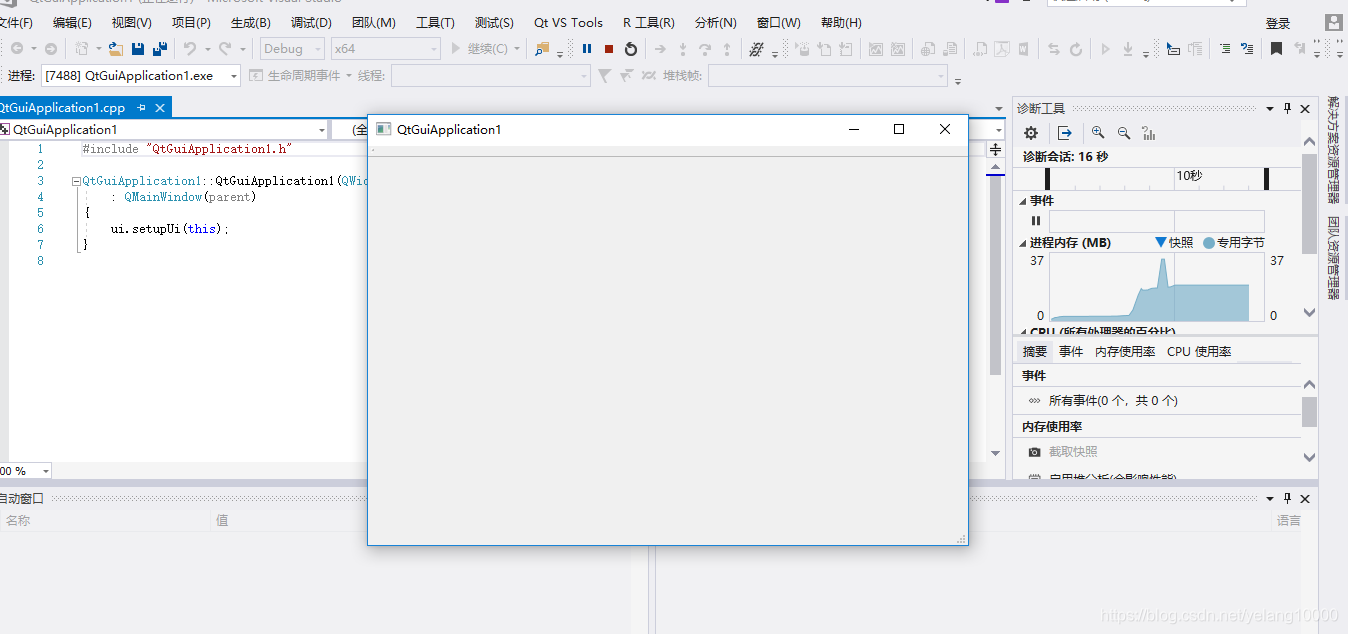
选择自己需要的模块



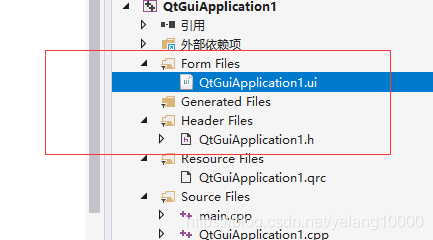
finsh



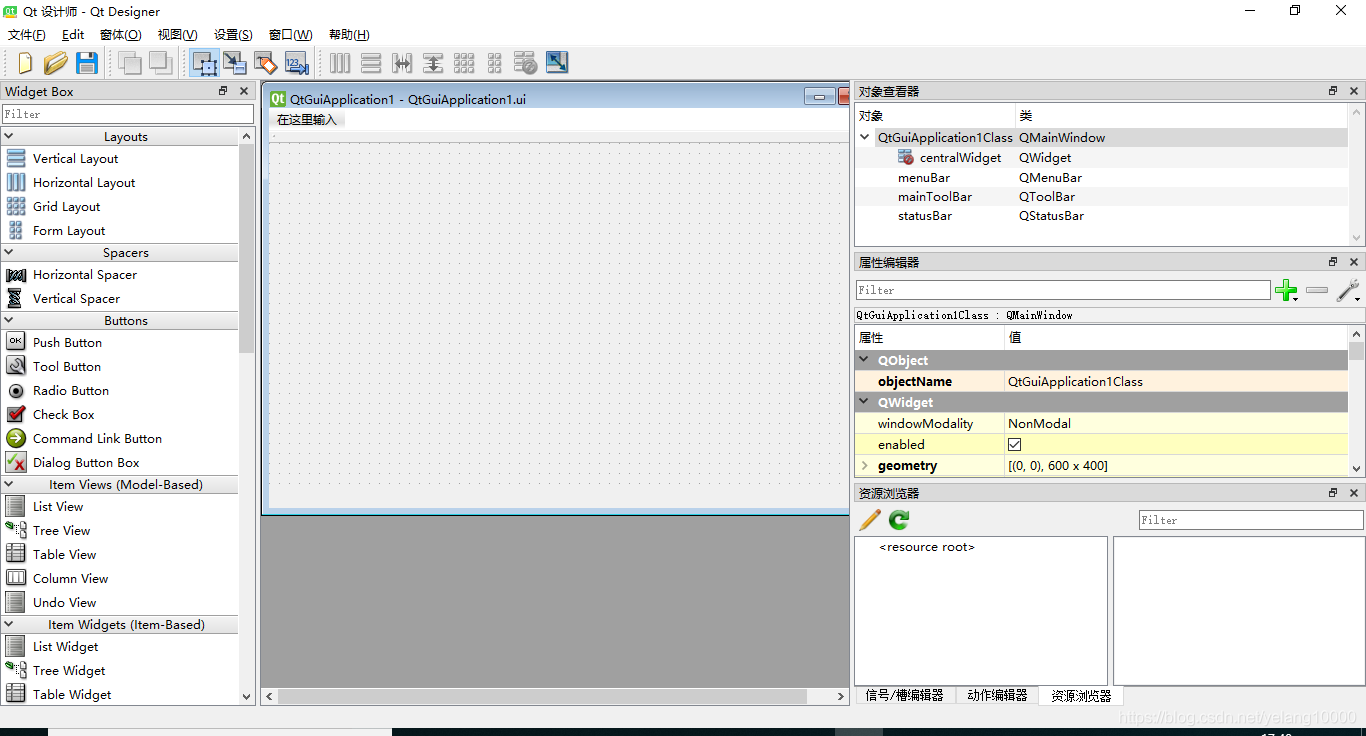
2、运行项目



点击右边的QtGuiApplication1.ui文件



进入Qt界面设计页面：



至此，我们的VS2017与QT5.12.1配置完成，后期陆续推出有关vs结合Qt的软件界面开发入门编程课程以及与OpenGL、OpenCV的图像编程处理课程。

# 4ch\*\*\*Qt 5.14.2超详细安装教程，不会来打我

## 2. 如何下载

这里咱们安装Qt5.14.2(不要问为什么不装更新版本的，从Qt5.15.0起，对于开源用户，Qt官方不再提供独立安装文件，所以源码安装太麻烦，以后再研究^\_^)

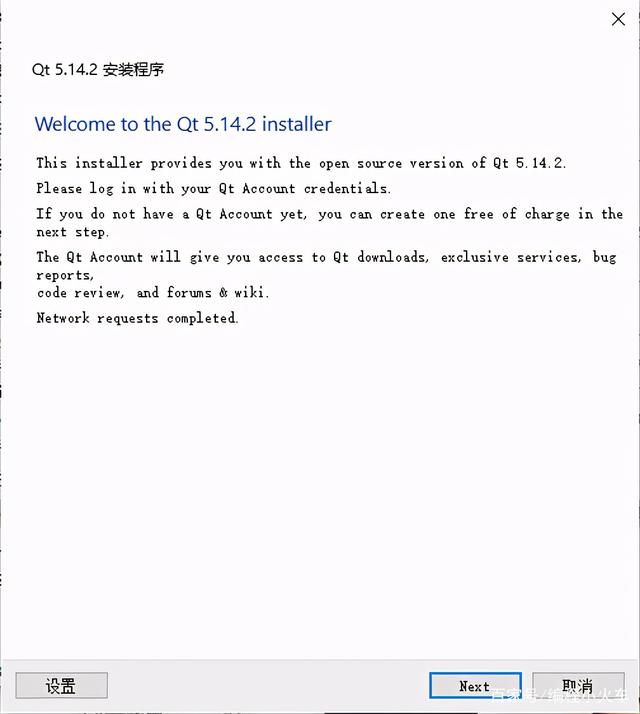
官网下载链接：

http://download.qt.io/archive/qt/5.14/5.14.2/



## 3.双击安装

下载完之后，直接双击打开安装包，点击Next即可



## 4.登录Qt账号

如果有Qt账号，直接登录就可以了，如果没有可以在下面注册一个，然后点Next（如果不想填写，请先断网，然后重新打开安装程序）



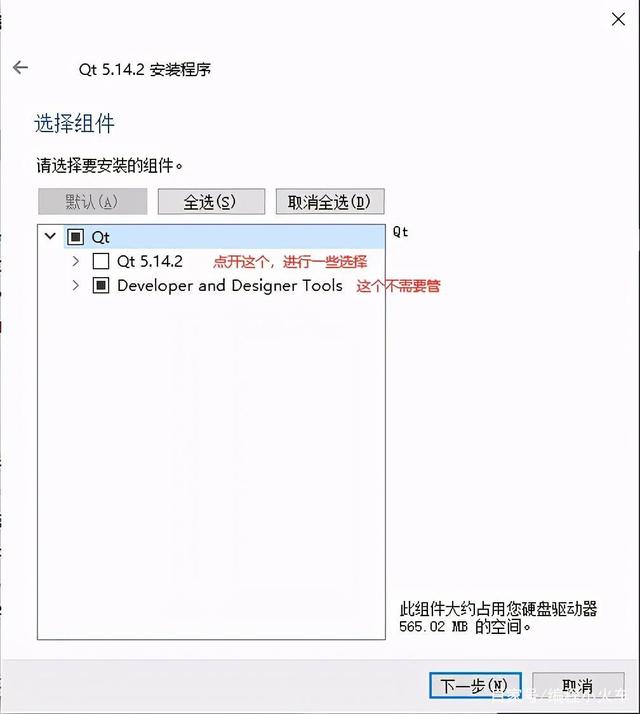
## 5.选择安装路径

现在你可以选择安装路径，建议大家自行修改一下（注意：路径中不能有空格，也不能有中文），确保自己能找到位置，后续可能会有一些配置，需要找到路径

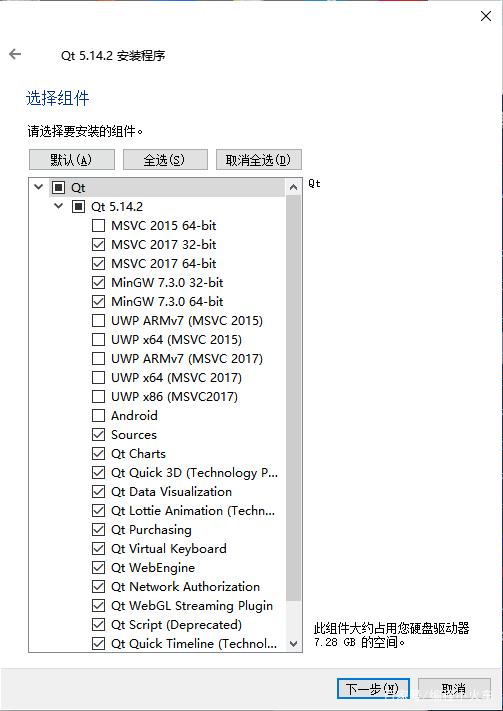


## 6.选择组件

路径设置完之后，到了选择组件的界面，如图：第二个组件不需要管，点开第一个组件Qt 5.14.2。



点开之后，把这些都选上，OK，下一步~(如果没有用VS开发Qt项目的需求可以不点 MSVC 2017)

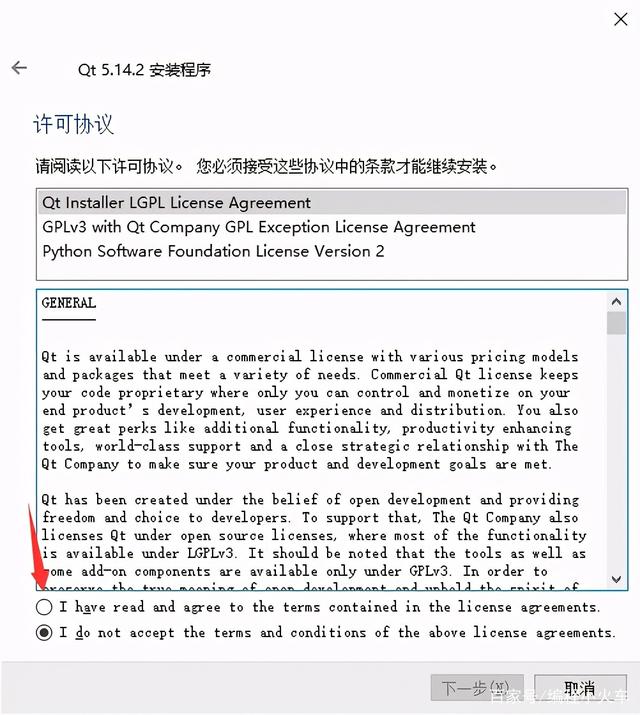


或者仅选择自己需要的组件：



## 7.许可协议

把同意协议选上，才能点击下一步



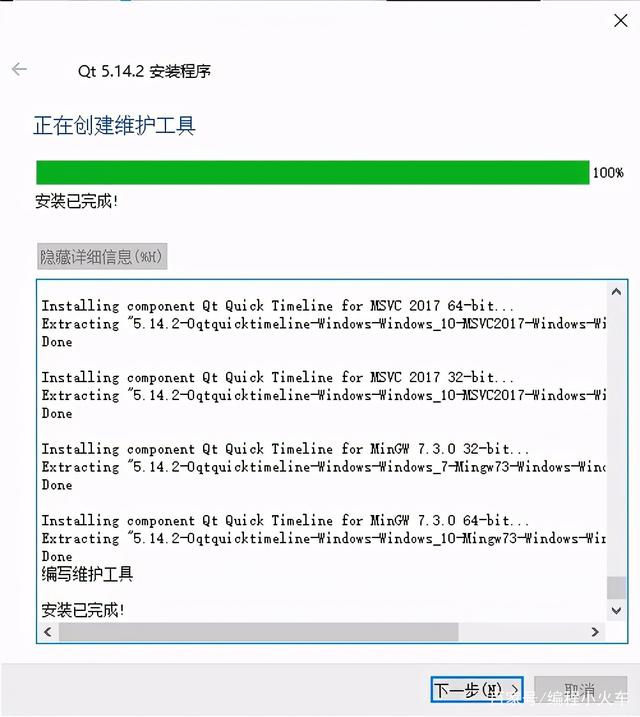
## 8.安装

然后一直点下一步 -> 最后点安装，等待安装完成即可

(**将是一个漫长的安装过程**)



电脑好的会非常快



**好了，至此Qt安装完成~接下来会为大家带来更多Qt使用技巧，喜欢的不如点个关注吧**

# 5ch\*\*\*在Qt里如何使用原生的OpenGL API？

需不需要添加glew库？还有没有其它要注意的事项？在Windows上，除了用MFC外，还有没有其它的合适的库和OpenGL搭配？（因为需要一个带GUI的控制界面）

https://www.zhihu.com/question/34848569

=======================

=======================

**[邱励瑞](https://www.zhihu.com/people/qiu-li-rui-75" \t "https://mail.163.com/js6/read/_blank)** 图形程序员

我刚刚把公司的旧opengl代码搬上qt5.6。

领导规定不能使用QOpenGLFunctions，因为那样的话就需要在opengl函数调用的前面添加上“QOpenGLFunctions对象->”。代码改动比较大。因此只能用glew。

glew的头文件与QT opengl的头文件不能共存，否则会产生冲突。也就是说，同一个cpp文件不能同时include glew的头文件 & QT opengl的头文件。所以，如果某个cpp调用了opengl API（由glew提供的），就不要include QT的头文件了（QOpenGLWidget、QSurfaceFormat、QOpenGLContext，等等），应该把glew和QT隔离开来。QT要求用户重写initializeGL()、resizeGL()、paintGL()这3个函数，这3个函数是一定会调用opengl API的。解决办法是：将这3个函数的具体实现搬到另一个cpp文件里。例如，在另一个cpp文件中定义一个\_resizeGL()函数，它是resizeGL()的具体实现。

myglwidget.cpp的代码片段

**#include** "myglwidget.h"**#include** "opengloperation.h"**void** MyGLWidget**::**resizeGL(**int** width, **int** height){

\_resizeGL(width, height);}

opengloperation.cpp的代码片段

**#include** "GL/glew.h"**#include** <QMatrix4x4> //可以使用QT的数学类头文件**void** **\_resizeGL**(**int** width, **int** height){

glViewport(0, 0, width, height);}

opengloperation.cpp不要include QT opengl的头文件（QOpenGLWidget、QSurfaceFormat、QOpenGLContext，等等）。使用QT的数学类是没有问题的。

其实glew和QOpenGLFunctions做的都是同一件事，就是获取opengl函数指针。glew完全可以替代QOpenGLFunctions。QT负责初始化opengl context并且绑定窗口，glew负责获取opengl函数指针。

[编辑于 2018-04-18](https://www.zhihu.com/question/34848569/answer/126169510" \t "https://mail.163.com/js6/read/_blank)

=======================

=======================

**[chris](https://www.zhihu.com/people/chris-67-13" \t "https://mail.163.com/js6/read/_blank)**

北京航空航天大学 机械电子工程博士

最近刚把纯opengl原生程序移植到qml框架下 只用到glew qt自带的qopenglfunction 之类乱七八糟的类全部都不需要 只需要附加include一个qopenglcontext的头文件即可 用到glew的地方全用pimpl手法进行封装，百分百做到使用了glew的cpp文件没有利用到任何qt自带的类，程序中绘图不用固定管线全靠着色器代码渲染。

整个项目用cmake管理 不用qt的pro文件也不用qtcreator。

[编辑于 2018-08-21](https://www.zhihu.com/question/34848569/answer/474123425" \t "https://mail.163.com/js6/read/_blank)