

实验一. Qt 库的安装及基本使用实验

一. Qt 简介

Qt 是 Digia 公司拥有并开发维护的跨平台 C++ 应用程序开发框架库，是完全面向对象、易于扩展，并允许组件编程的完整 C++ 函数库。除桌面操作系统外，Qt 全面支持 iOS、Android、Windows Phone 等移动操作系统。

Qt 具有优良的跨平台特性，可以面向对象编程，具备丰富的 API，支持 2D/3D 图形渲染，支持 OpenGL，以及大量的开发文档等。

Qt 提供 LGPL、GPL 和商业版三种授权方式。三种授权方式的功能、性能都没有区别，仅在于授权协议的不同。LGPL 和 GPL 是免费发布，主要针对非盈利性质用户；商业版则需收取授权费，主要针对盈利机构或企业用户。

二. Qt 的编译及安装

Qt 的安装有 2 种方法，第一种方法，也是最直接的方法，下载 Qt SDK 的完整安装包至本地驱动器，然后使用向导进行安装。由于编译器的不同，Qt SDK 的版本有很多，在下载安装时需要特别注意版本的选择。目前，Qt 官方最新的 5.7.0 版本中，使用的编译器为 MinGW 5.3.0。

官方网站下载地址为 http://download.qt.io/official_releases/qt/，进入该网址后，浏览器显示内容如图 1-1 所示。图 1-1 中，第一列（Name 列）显示的为 qt 大版本号，第二列（Last modified）为该版本的最初发布日期。以最新版本的“5.7”文件夹为例，点击进入该文件夹后，即可看到该大版本号下的各小版本号文件夹（如“5.5.0”、“5.7.0”）。一般来说，qt 在发布时，大版本号表示包含新特性的重大发布版本，小版本号表示该版本下各个功能修复或补丁（fix、patch）版本。

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
■ 5.7/	16-Jun-2016 09:04	-	
■ 5.6/	12-Oct-2016 07:44	-	
■ 5.5/	15-Oct-2015 08:05	-	
■ 5.4/	02-Jun-2015 07:53	-	
■ 5.3/	16-Sep-2014 08:45	-	
■ 5.2/	24-Feb-2014 13:07	-	
■ 5.1/	06-May-2014 12:44	-	
■ 5.0/	03-Jul-2013 11:57	-	
■ 4.8/	25-May-2015 17:55	-	

图 1-1 Qt 官方下载网站

继续进入“5.7.0”小版本号文件夹，即可看到各个版本的 Qt 安装包，如图 1-2 所示：

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
submodules/	15-Jun-2016 10:12	-	
single/	15-Jun-2016 10:15	-	
qt-opensource-windows-x86-wint-msvc2015-5.7.0.exe	15-Jun-2016 09:49	1.1G	Details
qt-opensource-windows-x86-wint-msvc2013-5.7.0.exe	15-Jun-2016 09:48	1.0G	Details
qt-opensource-windows-x86-msvc2015_64-5.7.0.exe	15-Jun-2016 09:48	918M	Details
qt-opensource-windows-x86-msvc2015-5.7.0.exe	15-Jun-2016 09:47	924M	Details
qt-opensource-windows-x86-msvc2013_64-5.7.0.exe	15-Jun-2016 09:46	904M	Details
qt-opensource-windows-x86-msvc2013-5.7.0.exe	15-Jun-2016 09:46	909M	Details
qt-opensource-windows-x86-mingw530-5.7.0.exe	15-Jun-2016 09:45	1.1G	Details
qt-opensource-windows-x86-android-5.7.0.exe	15-Jun-2016 09:44	1.2G	Details
qt-opensource-mac-x64-clang-5.7.0.dmg	15-Jun-2016 09:57	775M	Details
qt-opensource-mac-x64-android-ios-5.7.0.dmg	15-Jun-2016 09:56	2.3G	Details
qt-opensource-mac-x64-android-5.7.0.dmg	15-Jun-2016 09:55	828M	Details
qt-opensource-linux-x64-android-5.7.0.run	15-Jun-2016 09:53	769M	Details
qt-opensource-linux-x64-5.7.0.run	15-Jun-2016 09:53	715M	Details
md5sums.txt	15-Jun-2016 12:43	1.0K	Details

图 1-2 可供下载的各个 Qt 版本列表

由于该网站在国外，下载速度可能比较缓慢，建议使用中国科技大学的开源镜像进行下载：

IPv4 地址：https://mirrors.ustc.edu.cn/qtproject/official_releases/qt/

IPv6 地址：https://mirrors6.ustc.edu.cn/qtproject/official_releases/qt/

目录结构与官方网站目录结构一样。选择使用 MinGW 5.3.0 编译器(32 位)的 Windows 版本 Qt 库：qt-opensource-windows-x86-mingw530-5.7.0.exe

下载完成后，直接双击运行，按照向导进行安装。

第二种方法，下载相应的编译器、集成开发环境 QtCreator，以及 Qt 库的源码包，自行编译安装。该方法更具灵活性，更能够符合实际情况，因此以下主要介绍 Qt 库的编译安装方法，操作系统为 Windows 10 x64，安装目录为 D:\Dev。

（注：可以在以下网址下载 qt 5.7.0 已编译的完整压缩包及除 Python 外的编译工具：

链接：<http://pan.baidu.com/s/1Sjnci> 密码：2bmi

打开上述网盘，“编译工具”文件夹中为 MinGW 和 Perl 工具，“5.x.x”文件夹中为已编译库文件。其中 i686-5.3.0-release-posix-dwarf-rt_v4-rev0.7z 文件为 MinGW 5.3.0 版本，5.7.0_MinGW-530-posix-dwarf-rev0_with-ssl.7z 文件为使用 MinGW 5.3.0 编译的

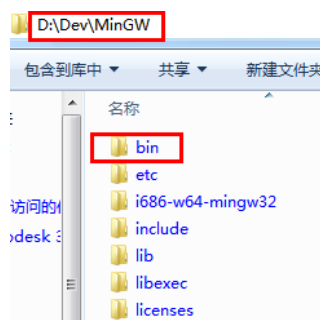
5.7.0 版本 qt 库，后缀 with-ssl 表示该编译包支持 ssl。下载，然后解压到相应的目录即可。)

准备工作：下载编译器 MinGW 5.3.0，QtCreator 以及 Qt 库的源码包（截至 2016 年 9 月，Qt Creator 最新版本为 4.1.0，Qt 库的最新版本为 5.7.0。现有文件可能已更新，请参考修改），文件如下：

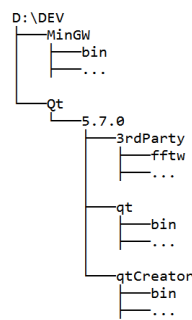
MinGW 5.3.0:	i686-5.3.0-release-posix-dwarf-rt_v4-rev0.7z
Qt 5.7.0:	qt-everywhere-opensource-src-5.7.0.7z
QtCreator 4.1.0:	qt-creator-opensource-src-4.1.0.zip
Perl:	Perl_5.14.2.7z

关于编译器，需要特别注意编译器版本的统一问题。无论选择哪一种版本的编译器，所有的 Qt 库及 Qt Creator 均必须保证使用同一版本编译器进行编译。本书中的编译器选择为 MinGW 5.3.0 版本，在随后相关的内容里，我们均使用该版本编译器进行程序编译。

下载并安装 Python 及 Perl，按照官方文档的说明，这两个软件的版本最低为 2.7 和 5.8。本书编译时使用的版本为 Python 3.5.2 和 Perl 5.14.2。



(a) MinGW 目录结构



(b) 安装目录结构

图 1-3 MinGW 编译器目录结构及 Qt 最终安装目录结构

以上编译器及编译工具准备完成后，我们开始编译 qt 及 qtCreator，基本过程如下：

1) 创建目录 D:\Dev，将 MinGW 5.3.0 文件解压至该目录下，并将解压后的目录重新命名为 MinGW，确保目录结构如图 1-3(a)和(b)所示。

2) 在系统的环境变量中添加环境变量。确保 Path 环境变量中，包含 MinGW 的目录：**D:\Dev\MinGW\bin**，该目录即编译器“g++.exe”文件的位置。（如何设置环境变量，请参考附录 1）

3) 点击 Win7 系统左下角 Windows 徽标（即“开始”菜单），在“搜索程序和文件”中输入：“cmd”，单击 cmd.exe 运行，进入命令提示符终端，在命令提示符终端内输入：

“g++ -v”，按下回车键执行该命令（在命令提示符终端下，输入命令后按下回车键即开始执行所输入的命令）。

4) 如果环境变量配置正确，此时应有 MinGW 的版本信息出现，如图 1-4(a)所示，最后一行方框内显示了 MinGW 的版本信息：“gcc version 5.3.0...”。

5) 如果出现“g++”不是内部或外部命令，也不是可运行的程序或批处理文件。”错误，如图 1-4(b)所示，说明系统无法找到 g++.exe，请检查 Path 环境变量中与 MinGW 相关的路径配置。

```

D:\Dev>g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=D:/Dev/MinGW/bin/./libexec/gcc/i686-w64-mingw32/5.3.0/lto-w
rapper.exe
Target: i686-w64-mingw32
Configured with: ../../src/gcc-5.3.0/configure --host=i686-w64-mingw32 --buil
d=i686-w64-mingw32 --target=i686-w64-mingw32 --prefix=/mingw32 --with-sysroot=/c
/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32 --with-gxx-include-dir=/mingw3
2/i686-w64-mingw32/include/c++ --enable-shared --enable-static --disable-multili
b --enable-languages=c,c++,fortran,lto --enable-libstdcxx-time=yes --enable-thre
ads=posix --enable-libgomp --enable-libatomic --enable-lto --enable-graphite --e
nable-checking=release --enable-fully-dynamic-string --enable-version-specific-r
untime-libs --disable-sjlj-exceptions --with-dwarf2 --disable-lsl-version-check
--disable-libstdcxx-pch --disable-libstdcxx-debug --enable-bootstrap --disable-r
path --disable-win32-registry --disable-nls --disable-werror --disable-symvers -
-with-gnu-as --with-gnu-ld --with-arch=i686 --with-tune=generic --with-libiconv
--with-system-zlib --with-gmp=/c/mingw530/prerequisites/i686-w64-mingw32-static
--with-mpfr=/c/mingw530/prerequisites/i686-w64-mingw32-static --with-mpc=/c/ming
w530/prerequisites/i686-w64-mingw32-static --with-lsl=/c/mingw530/prerequisites/
i686-w64-mingw32-static --with-pkgversion='i686-posix-dwarf-rev0, Built by MinGW
-W64 project' --with-bugurl=http://sourceforge.net/projects/mingw-w64 CFLAGS=-O
2 -pipe -I/c/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32/opt/include -I/c/m
ingw530/prerequisites/i686-zlib-static/include -I/c/mingw530/prerequisites/i686-
w64-mingw32-static/include CXXFLAGS=-O2 -pipe -I/c/mingw530/i686-530-posix-dwa
rf-rt_v4-rev0/mingw32/opt/include -I/c/mingw530/prerequisites/i686-zlib-static/i
nclude -I/c/mingw530/prerequisites/i686-w64-mingw32-static/include CPPFLAGS=LD
FLAGS=-pipe -L/c/mingw530/i686-530-posix-dwarf-rt_v4-rev0/mingw32/opt/lib -L/c/
mingw530/prerequisites/i686-zlib-static/lib -L/c/mingw530/prerequisites/i686-w6
4-mingw32-static/lib -Wl,--large-address-aware
Thread model: posix
gcc version 5.3.0 (i686-posix-dwarf-rev0, Built by MinGW-W64 project)
D:\Dev>

```

(a)正确配置环境变量后的版本信息

```

D:\Dev>g++ -v
'g++' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
D:\Dev>

```

(b)未正确配置环境变量后的错误提示

图 1-4 g++编译器环境变量配置测试

6) 下载并安装 Python（本书使用的版本为 3.4.4，目前最新版本是 3.5.1，应该可以使用，未测试）。安装完成，进入命令提示符，使用“python --version 查看版本号。

7) 下载并安装 Perl，Qt 文档中要求最低版本为 5.8，本书编译时采用 5.14.2。解压 Perl_5.14.2.7z 文件至 D:\Dev 目录，将 perl.exe 文件所在路径（本书编译时该路径为 D:\Dev\Perl\bin）添加至环境变量中。安装完成，使用“perl -v”查看版本号。

8) 解压 qt-everywhere-opensource-src-5.7.0.7z 至 E:\qt 目录下, 注意路径中请勿包含中文或空格。在命令提示符终端下进入 E:\Qt\qt-everywhere-opensource-src-5.7.0 目录, 并执行以下的 configure 命令, 如图 1-5 所示:

```
configure -prefix D:/Dev/Qt/5.7.0/qt -hostprefix D:/Dev/Qt/5.7.0/qt
-opensource -confirm-license -qt-sql-sqlite -plugin-sql-sqlite
-accessibility -opengl desktop -qt-libjpeg -qt-zlib
```

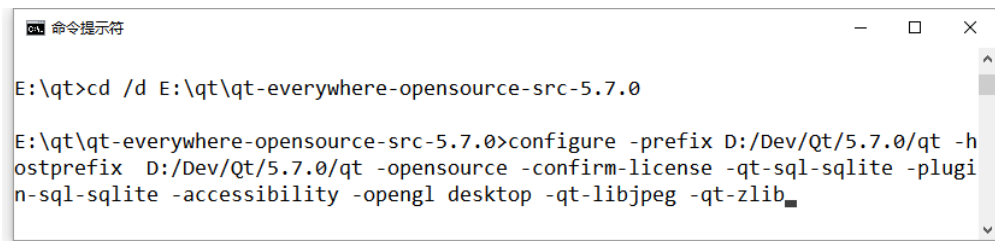


图 1-5 配置 qt 库

9) 执行上述 configure 命令后, 命令提示符终端下会出现大量输出信息, 其主要包括 qt 库编译类型、所选择编译的库模块, 以及 Qt 库的安装配置信息等, 典型的输出如图 1-6 所示。

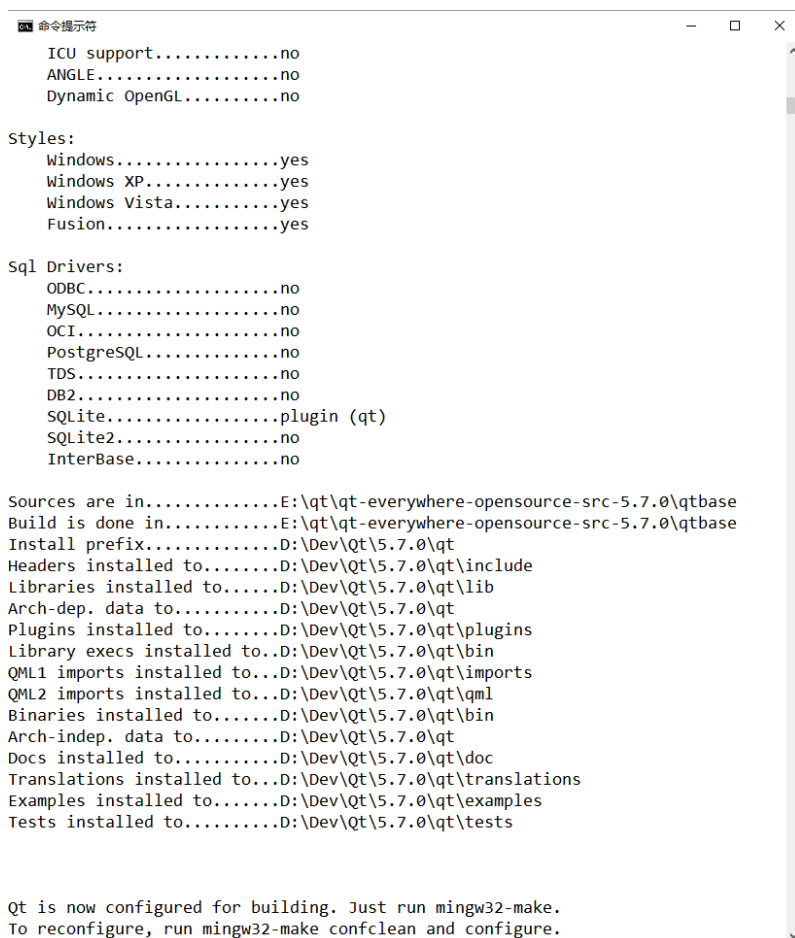


图 1-6 执行 configure 命令后, 命令提示符终端输出信息

10) configure 完成后,即可执行 `mingw32-make & mingw32-make install` 命令来进行编译和安装(这里 `mingw32-make` 命令仅使用了单核模式,编译过程极为漫长,一般需要 4-6 个小时以上,如果 CPU 为多核,可以使用 `mingw32-make -jn` 命令,其中 `n` 为 CPU 内核数,如 `mingw32-make -j4`,可极大加快编译速度)。安装完成后,在 `D:\Dev\Qt\5.7.0\qt` 目录下会出现 `bin` 目录,将这个目录(`D:\Dev\Qt\5.7.0\qt\bin`)同样添加进入环境变量 `Path` 中。重新打开一个新的命令行终端,执行“`qmake -v`”,如果环境变量配置正确,应当出现 `qmake` 的版本号(如图 1-7 所示)。如果没有出现,请检查 `Path` 环境变量中与 `qt` 相关的 `bin` 目录是否与上述路径一致。

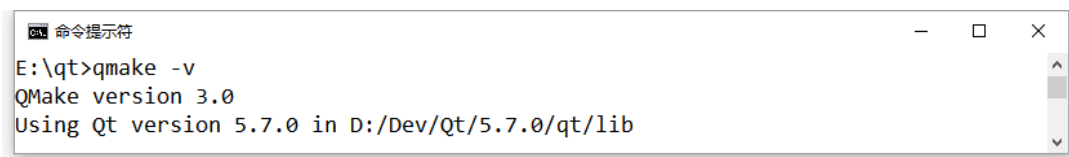


图 1-7 Qt 库环境变量测试

(注: `qmake` 是 Qt 工具包中附带的一个非常方便和易于使用的跨平台工程管理工作。利用该工具,程序员可方便的创建应用于不同系统平台的工程配置和编译指令文件。如对于 GNU C++编译器,该工具可创建 `Makefile` 文件;对于微软的 Visual Studio 编译器,该工具可创建 Visual Studio 的工程文件。

`qmake` 可根据平台无关的工程配置文件(`.pro` 文件),依据一定得逻辑规则调用 Qt 内置代码生成工具(`moc`、`uic`、`rcc` 等),生成相应平台的 `Makefile` 文件。

使用 `qmake` 工具的核心,是编写出符合语法规则的`.pro` 文件。关于`.pro` 文件的具体语法规则,详见 Qt 附带的 `qmake` 文档。)

至此,Qt 库安装完成。接下来,开始编译 Qt 的集成开发环境 Qt Creator。

11) 解压 `qt-creator-opensource-src-4.1.0.zip`, 设源码解压路径为

`D:\tmp\qt-creator-opensource-4.1.0`。在 `D:\tmp` 下新建一个目录 `D:\tmp\qtCreator`, 使用命令提示符终端进入该目录,执行下面两条命令,即可编译。

```
qmake ..\qt-creator-opensource-4.1.0
mingw32-make release
```

Qt Creator 编译完成后,删除 `D:\tmp\qtCreator\src`,然后将 Qt Creator 目录复制到 `D:\Dev\Qt\5.7.0`目录下。

打开“资源管理器”,进入 `D:\Dev\Qt\5.7.0\qtCreator` 目录,双击运行 `qtcreator.exe`。开始配置 `qtcreator` 所使用的 `qt` 库、编译器和调试工具:

依次点击菜单“工具”、“选项”、“构建和运行”。在“构建和运行”菜单项下有六个选

项卡，分别是“概要”、“构建套件(Kit)”、“Qt Version”、“编译器”、“Debugger”和“cmake”。其中“Qt Version”、“编译器”和“Debugger”三个选项卡分别对应所使用的 qt 库、编译器和调试工具，如图 1-9 所示。这里分别设置 qmake，MinGW 和 GDB（没有就添加），配置路径如下：

Qt Version 选项卡，qmake 路径：D:\Dev\Qt\5.7.0\qt\bin\qmake.exe；

编译器选项卡，编译器路径：D:\Dev\MinGW\bin\g++.exe；

Debugger 选项卡，path：D:\Dev\MinGW\bin\gdb.exe；

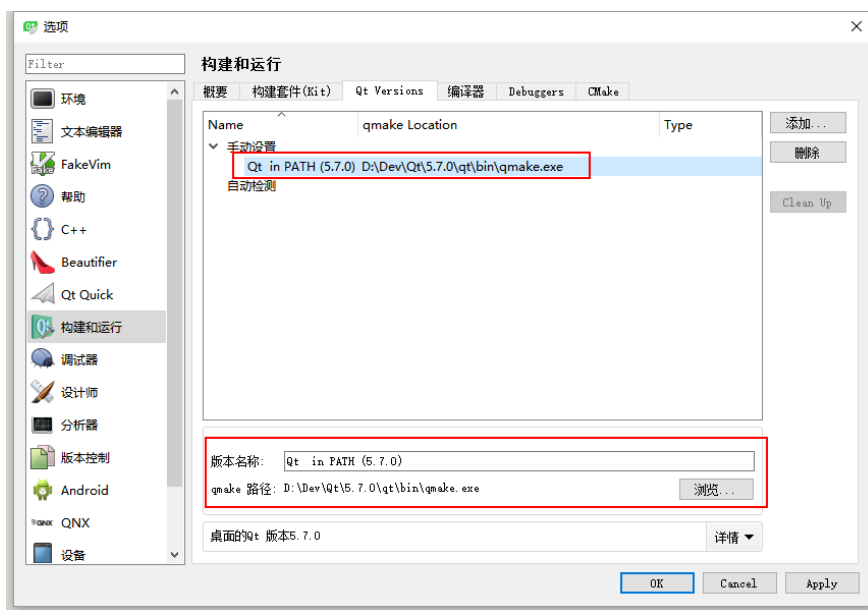


图 1-9 qtCreator 配置图

至此，Qt 及 Qt Creator 的编译、安装和配置全部完成。

三. Qt 库

Qt 库是一个用 C++编写的跨平台开发框架库，可与 Python, Ruby, java 等很多语言绑定。该框架库提供了几乎涵盖所有能够想象的类，结构十分复杂。为了便于管理和使用，Qt 库在设计时被分成不同模块，用户可根据任务需求，选择和使用相应的模块。

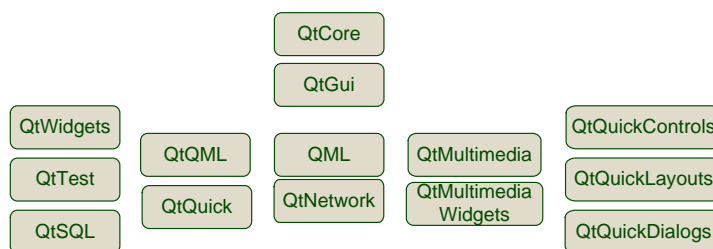


图 1-10 Qt 库 5.7.0 版本包含的基本模块

如图 1-10 所示为 Qt 基本模块，每一个模块均包含了大量相关功能类。对于 Qt 库中的这些类，一般来说可以分成两种类型：直接或间接派生自 Qt 类的类，以及不从任

何类继承的独立类。

直接派生自 Qt 类的类主要可以分成对象类（QObject 类）和事件类（QEvent 类）。QObject 类是所有应用组件类的基类，QEvent 类是所有 Qt 事件响应类的基类。除此之外，类似 QCursor、QPen、QTab 等类可以在窗体的任意地方出现、描述的窗口组件，均从 Qt 类派生。

独立类在 Qt 库中一般用于完成独立的功能，例如操作 XML 文件的 QDomReader 类。

四. Qt Creator 的基本使用方法

本节以一个简单实例，说明在 qtCreator 中利用向导创建一个简单的应用程序，进而编写代码、编译以及使用 GDB 进行程序调试的基本方法和过程。

1. 创建应用程序

1)依次点击“New Project”、“应用程序”、“Qt Widgets Application”、“choose”，在“项目介绍和位置”对话框内，名称和创建路径分别填“exer01”和“D:\Dev\Project”。点击“下一步”、“下一步”。在“类信息”对话框内，首先选择基类为 QWidget，然后在类名内填入 exer01Widget，其他默认，点击“下一步”、“完成”，至此向导即为我们创建了一个基本的 Qt Widget Application。

点击 qtcreator 左下角的绿色三角形按钮，如图 1-11 所示，编译运行刚刚创建的空白应用程序。

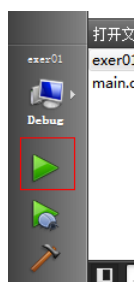


图 1-11 运行键

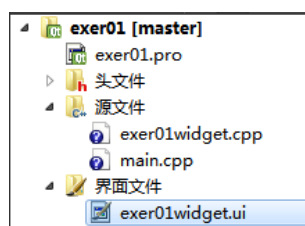


图 1-12 项目浏览器

2)双击 qtcreator 左上角的项目->界面文件->exer01widget.ui 文件，如图 1-12 所示，打开界面编辑器。适当拖拉界面的大小，并在界面上放置一个 QPushButton、Horizontal Spacer、QLabel 和 QTextEditor。并框选前三个控件，点击水平布局，如图 1-13(a)和(b)所示。

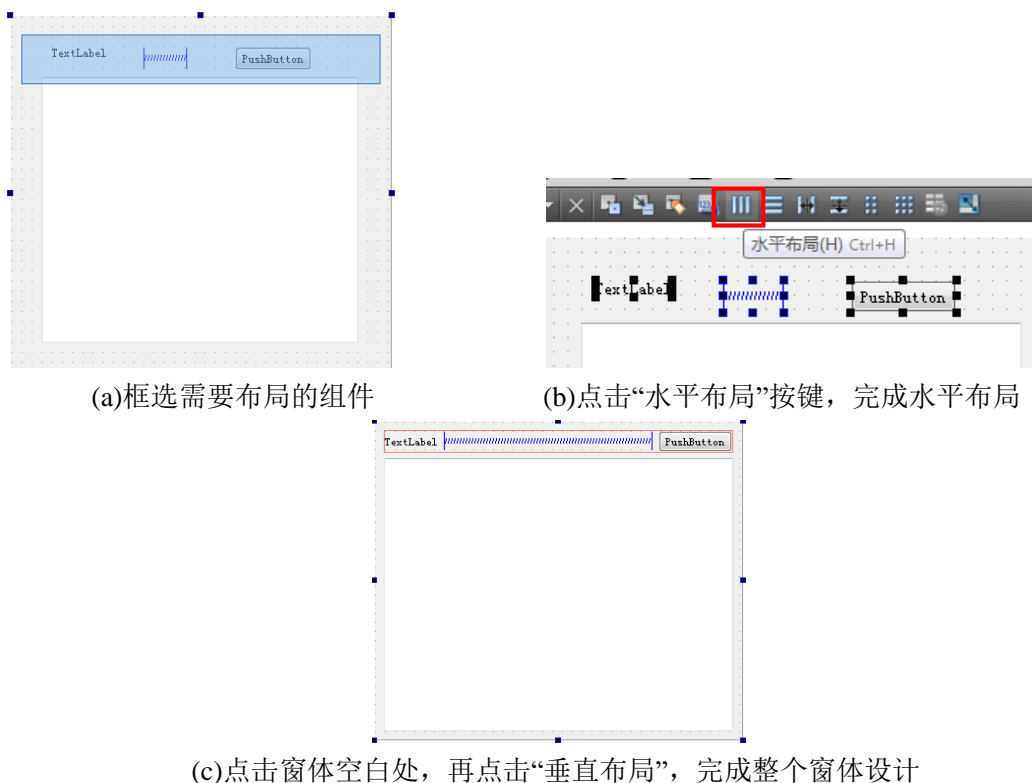


图 1-13 使用界面设计器进行界面布局

3)单击窗体空白处，点击“垂直布局”（在水平布局旁），我们可以得到界面基本布局情况，如图 1-13(c)所示。

完成窗体布局后，可以按下“Alt+Shift+R”快捷键，预览设计后的窗体。如果设计的界面没有达到预期的效果，可以重复上述各个步骤，对界面进行反复修改，直至符合要求为止。

2. 编写功能实现代码

当界面设计完成后，我们开始对程序添加具体的处理功能。

1)先后单击界面上的 TextLabel 和 PushButton，在右侧将 Text 分别改为“随机数:”，“生成并添加”，右键单击“生成并添加”按钮，单击“转到槽”，选择“clicked()”，点击“OK”。这时，qtcreator 会自动跳入源码编辑模式，并自动创建上述点击信号的响应槽函数。在新生成的槽函数内添加如下代码：

```

void exer01Widget::on_pushButton_clicked()
{
    QVector<float> v; //定义QVector对象

    for (int i=0;i<5;i++){ // 生成5个随机数
        v.push_back(float( (rand()%100)/10.0));
    }

    qDebug()<<v; // 调试输出

    QString s="";
    foreach(float f,v){ // 将5个随机数转换为一个字符串
        s += QString("%1 ").arg(f);
    }

    // 将该字符串插入文本编辑框中
    ui->textEdit->insertPlainText(QString("%1%2").arg(s,"\n"));
}

```

找到构造函数，在构造函数内添加：

```

exer01Widget::exer01Widget(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::exer01Widget)
{
    ui->setupUi(this);
    srand((unsigned int)time(NULL));
}

```

并在该文件开始处，添加如下三个头文件：

```

#include <QVector>
#include <ctime>
#include <QDebug>

```

2)点击左下角绿色的运行按钮，再次编译运行程序。注意单击“生成并添加”按钮时，程序的文本框内会出现一系列数字，同时观察 qtcreator 内的应用程序输出框内，也会出现一行 QVector 字样的序列。这是由于在槽函数中有一句 qDebug()字样的语句，该语句的功能就是在 qtcreator 内输出调试信息。qDebug 功能强大，是进行程序调试时的利器，其具体信息请参阅 qt 文档。

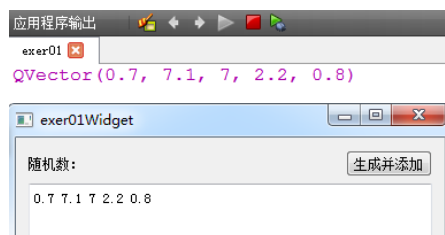
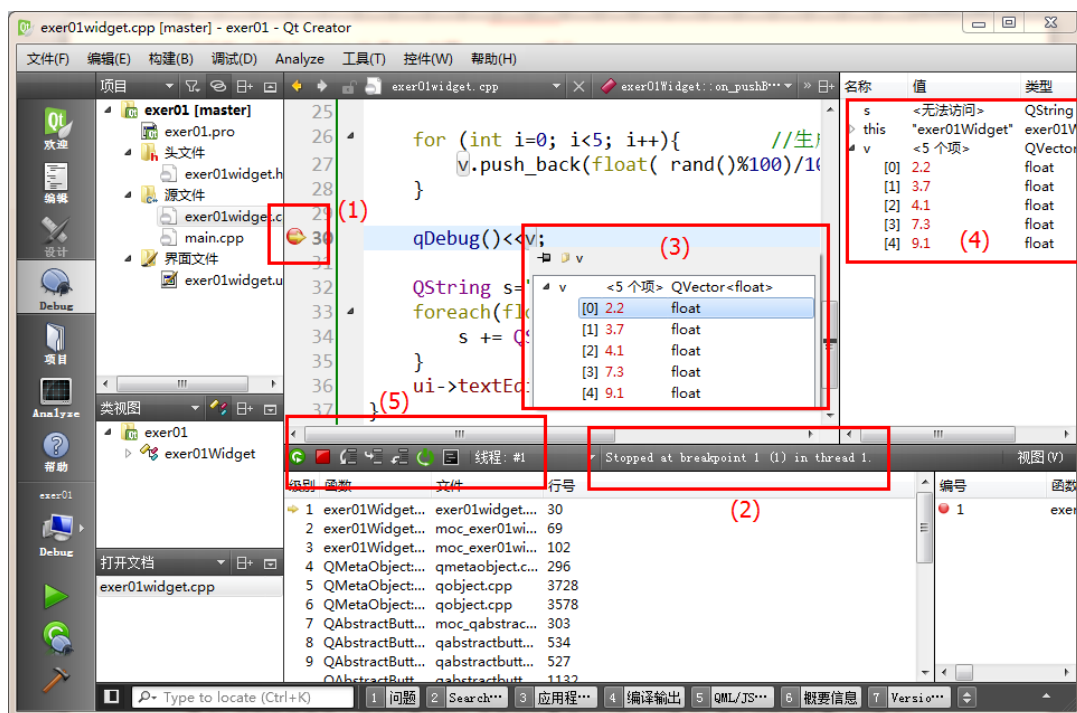


图 1-14 例程运行图及 Qt Creator 输出图

3. 使用断点功能进行应用程序调试。

首先在“qDebug ()<<v;”这一行前方的行号前空白处单击，出现一个红色的断点标志。点击左下角绿色三角形的“运行”按钮下方的“开始调试”按钮（或按下 F5 键），进入调试状态。当程序界面出现后，单击“生成并添加”按钮，Qt Creator 会停止在断点设置的这一行，并且会出现提示：在线程 1 的断点 1（1）处停止。如图 1-15(2)所示。



(1)断点位置 (2)程序停止提示 (3)鼠标跟随方式显示变量值

(4) 当前变量值显示框 (5)命令控制工具条

图 1-15 使用 GDB 对例程进行调试

请注意图 1-15 中各方框中的内容，方框 1 表示当前断点位置，方框 2 提示断点停止，方框 3，将鼠标移至某个有效变量上时，会弹出变量相关内容；方框 4，显示当前的变量，方框 5，从左到右依次为继续运行（到下一个断点）、停止调试、单步运行、单步进入、单步跳出和重新启动调试器。

通过 GDB，我们能够找到并改正程序的错误，创建一个基本正确的应用程序。

五. 预习及实验要求

1. 认真阅读本说明及参考示例；
2. 阅读 Qt 文档：Creating Project Files，并将其按照原文档格式译为中文（代码不需翻译）。
3. 一组学生成绩数据如下所示，参考 Qt 文档，使用 Qt Creator 编写一个控制台应用程序，可以将下列数据分别以姓名排序、以课程 1 排序、以课程 2 排序输出。

学号	姓名	课程 1	课程 2
1403130209	鲁智深	80	72
1403140101	林冲	82	76
1403140102	宋江	76	85
1403140103	武松	88	80
...

例如：以课程 1 成绩为标准排序结果如下：

学号	姓名	课程 1	课程 2
1403140103	武松	88	80
1403140101	林冲	82	76
1403130209	鲁智深	80	72
1403140102	宋江	76	85
...

六. 附录一：设置 Windows 7 的环境变量

右键单击桌面“计算机”图标，点击“属性”，点击左侧最下一个菜单项“高级系统设置”后依次选择“高级”、“环境变量”既可进入设置环境变量。

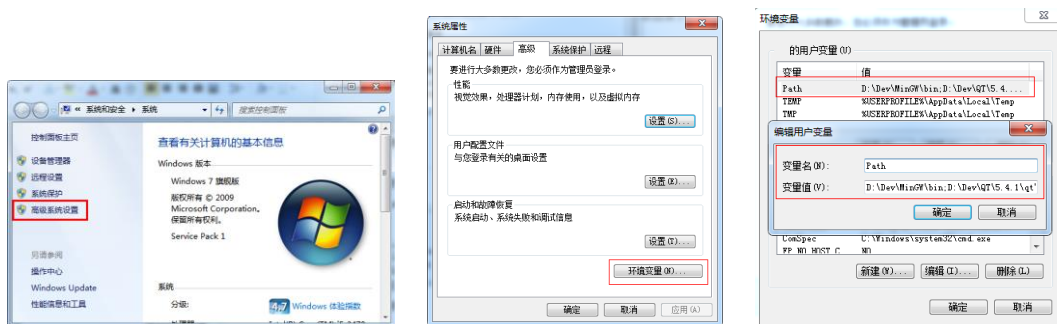


图 1-16 环境变量配置图

或，点击“控制面板”，依次选择“用户帐户和家庭安全”、“用户帐户”，在窗口左边最后一行“改变我的环境变量”，同样可以设置环境变量。

对话框上面的为“用户变量”，对话框下面的为“系统变量”（相当于所有用户的变量）----“新建”，即可添加自己想要的变量。注意路径中最好避免使用空格或中文字符，以避免出现意外问题。