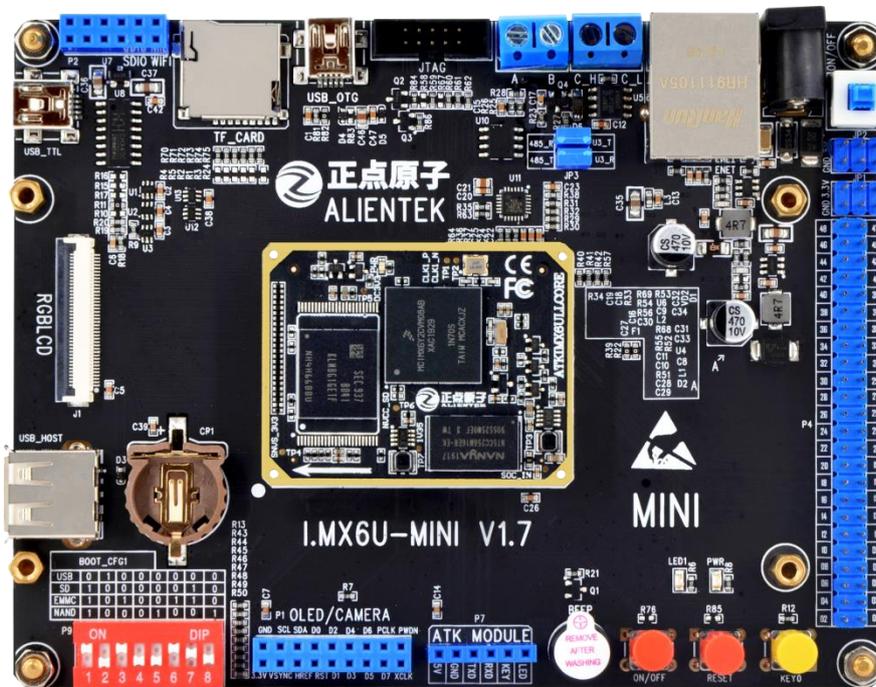
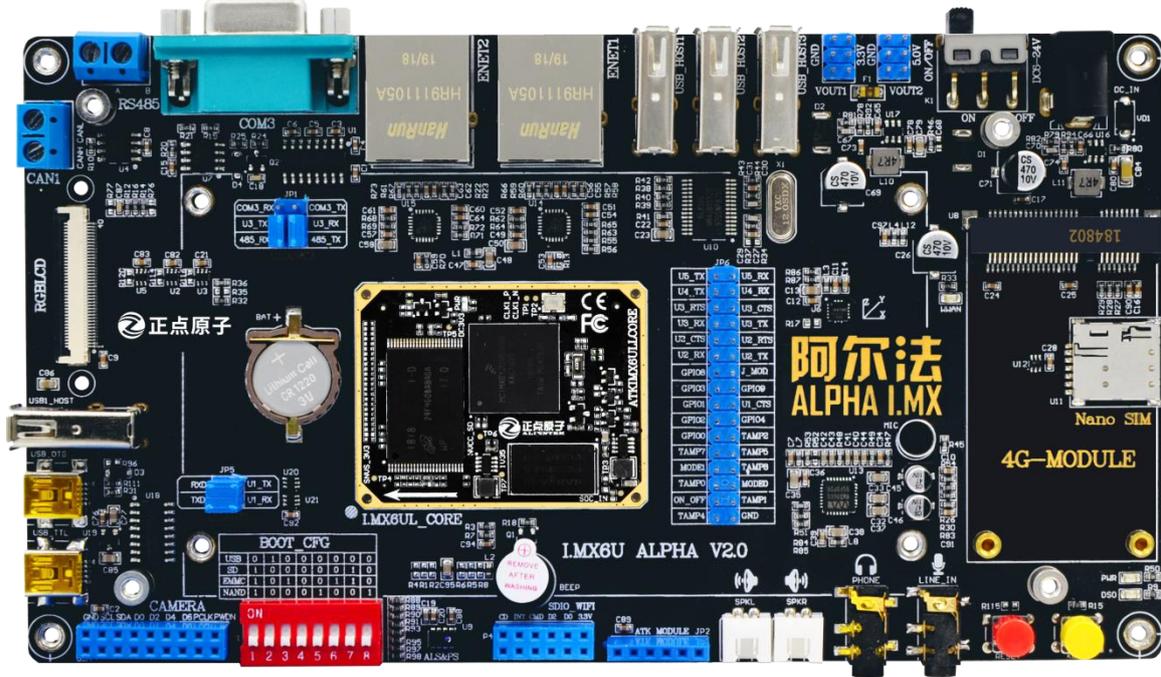


# 【正点原子】I.MX6U

## 出厂系统 Qt 交叉编译环境搭建 V1.6





正点原子公司名称：广州市星翼电子科技有限公司

原子哥在线教学平台：[www.yuanzige.com](http://www.yuanzige.com)

开源电子网 / 论坛：<http://www.openedv.com/forum.php>

正点原子淘宝店铺：<https://openedv.taobao.com>

正点原子官方网站：[www.alientek.com](http://www.alientek.com)

正点原子 B 站视频：<https://space.bilibili.com/394620890>

电话：020-38271790 传真：020-36773971

请关注正点原子公众号，资料发布更新我们会通知。

请下载原子哥 APP，数千讲视频免费学习，更快更流畅。



扫码关注正点原子公众号



扫码下载“原子哥”APP

## 文档更新说明

版本	版本更新说明	负责人	校审	发布日期
V1.0	初稿:	正点原子 linux 团队	正点原子 linux 团队	2019.10.26
V1.1	1.补充了文档需要注意的问题, 已经使用红色标注。 2.增加了附录 常见问题汇总, 以方便用户查找搭建环境所遇到的问题。	正点原子 linux 团队	正点原子 linux 团队	2019.12.30
V1.2	1.更新官方下载 Qt 的链接	正点原子 linux 团队	正点原子 linux 团队	2020.4.28
V1.3	1. 更改文档名称由 Qt 交叉编译环境搭建改为出厂系统 Qt 交叉编译环境搭建。 2. 出厂系统 Qt 版本由 Qt5.6.2 版本升级到 Qt5.12.9 LTS(Qt 官方长期支持)版本 3. 重写 Qt Creator 的交叉编译环境配置	正点原子 linux 团队	正点原子 linux 团队	2020.11.22
V1.4	1、添加 2.5.1 小节 rsync 方式 Qt 远程调试	正点原子 linux 团队	正点原子 linux 团队	2021.2.1
V1.5	1、在 2.3 小节 配置 Qt Creator Kits 的脚本中添加使能环境变量的指令	正点原子 linux 团队	正点原子 linux 团队	2021.3.8
V1.6	1、在 2.4 小节添加 Qt Creator 使用出厂系统交叉编译工具链识别类错误的解决方法	正点原子 linux 团队	正点原子 linux 团队	2021.3.25

## 目录

前言.....	5
<b>第一章 安装交叉编译器.....</b>	<b>6</b>
1.1 安装出厂系统 Qt 交叉编译器 .....	6
<b>第二章 搭建 QT CREATOR 交叉编译环境.....</b>	<b>7</b>
2.1 下载 QT CREATOR .....	7
2.2 安装 QT CREATOR .....	8
2.3 配置 QT CREATOR KITS.....	11
2.4 验证搭建的交叉编译 KITS .....	15
2.5 远程调试 QT 程序 .....	20
2.5.1 rsync 方式 Qt 远程调试 .....	20
2.5.2 无 rsync 方式 Qt 远程调试 .....	27
<b>第三章 UBUNTU 本机 KITS 编译 QT 应用程序.....</b>	<b>33</b>
3.1 UBUNTU 编译 QT 应用程序 .....	33
<b>附录-A .....</b>	<b>33</b>

## 前言

在 2020.11.20 日以前正点原子出厂 I.MX6U 出厂系统的 Qt 版本是 5.6.2 版本的。我们在 2020.11.20 已经将它升级为 Qt 5.12.9 LTS (Qt 官方长期支持版本。至今天 2020.11.22 日 Qt 官方还在更新 Qt5.12 版本)。升级 Qt 版本带来很多好处,更多的功能,更多的模块可用。以后正点原子都是以 Qt5.12.9 和 Qt4.8.4 (Qt4 版本, 客户需要) 作主要使用版本, Qt5.6.2 正点原子已经不再使用。此文档是针对使用 **正点原子出厂文件系统 Qt** 所搭建的环境 (注: 如果不想用出厂系统, 那么您需要移植 Qt 到其他不含 Qt 的文件系统上才可以运行 Qt 程序, 可以参考【正点原子】I.MX6U 移植 Qt4.8.4 V1.x.pdf 和【正点原子】I.MX6U 移植 Qt5.12.9 V1.x.pdf)。如有错漏, 请到正点原子论坛指正, 或者联系本文档编写者 QQ1252699831 指正错误。

本文档所使用的环境:

- ✚ Windows 7 64bits, 也适用于 Windows 8-10。不建议用 Windows 32 位来开发, Windows 32 位支持的内存大小有限, 系统性能有限。
- ✚ Ubuntu16.04, Ubuntu 建议使用 16.04 否则安装及编译环境不一样导致出错, 请自行解决!
- ✚ 要求读者会使用 FileZilla、WinSCP 及 Windows Git 进行 Ubuntu 与 Windows 间互传文件的方法。

## 第一章 安装交叉编译器

### 1.1 安装出厂系统 Qt 交叉编译器

注: 若用户在快速体验文档已经安装过可跳过此 1.1 小节。2020.11.20 日后出厂系统的 Qt 交叉编译工具链里的 Qt 版本已经升级为 Qt5.12.9 版本。2020.11.20 日前安装过 `fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh` 需要重新下载重新安装。直接删除 `sudo rm -rf /opt/fsl-imx-x11/4.1.15-2.1.0` 再按以下重新安装。

把开发板光盘 A-基础资料->5、开发工具->1、交叉编译器->`fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh` 拷贝到 Ubuntu 虚拟机

如下图本文已经把交叉编译工具拷贝到了 Ubuntu 虚拟机。

```
allentek@ubuntu:~$ ls
fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
allentek@ubuntu:~$
```

图 1.1 1 拷贝 sdk 工具包到 ubuntu 系统里

执行下面的指令修改脚本的权限, 修改权限后可以看到此脚本颜色显示改变, 说明修改成功。

```
chmod u+x fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
```

直接执行脚本安装交叉编译工具, 连续敲下两次回车键确认, 再输入用户密码即可。本次安装的目录为脚本所指定的默认安装的目录, 后面的内核编译环境的交叉编译都是按这个安装目录去操作, 所以建议用户也是默认安装到 `/opt/fsl-imx-x11/4.1.15-2.1.0` 这个默认目录。

```
allentek@ubuntu:~$ ./fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
Freescal e.MX Release Distro SDK installer version 4.1.15-2.1.0
=====
Enter target directory for SDK (default: /opt/fsl-imx-x11/4.1.15-2.1.0):
You are about to install the SDK to "/opt/fsl-imx-x11/4.1.15-2.1.0". Proceed[Y/n]?
[sudo] password for allentek:
Extracting SDK.....done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
$ . /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allentek@ubuntu:~$
```

图 1.1 2 安装 sdk 工具到默认的目录

使用方法也十分简单, 根据上面打印出来的提示, 直接使能环境变量就可以了。但是在不同终端或者切换用户时需要重新使能环境变量方可使用。

```
source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
```

```
allentek@ubuntu:~$ source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allentek@ubuntu:~$
```

图 1.1 3 使能环境变量

使能环境变量后可以使用 `env` 指令查看生效的环境变量, 下图为部分截图, 可以看出使能了这个环境变量后 `gcc` 已经配置好编译时所用的参数, 如硬浮点参数 `-mfpu=neon -mfloat-abi=hard`。使用硬浮点交叉编译, 可以使用 CPU 自带 FPU。下图为环境变量部分截图。

```
env
LESSOPEN=| /usr/bin/lesspipe %s
ARCH=arm
RANLIB=arm-poky-linux-gnueabi-ranlib
OE_QMAKE_CFLAGS=
INSTANCE=Unity
UPSTART_JOB=unity-settings-daemon
CROSS_COMPILE=arm-poky-linux-gnueabi-
CC=arm-poky-linux-gnueabi-gcc -march=armv7ve -mfpu=neon -mfloat-abi=hard -mcpu=cortex-a7 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
XDG_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDG_CURRENT_DESKTOP=Unity
GTK_IM_MODULE=ibus
OE_QMAKE_LIBDIR_QT=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi/usr/lib
OBJDUMP=arm-poky-linux-gnueabi-objdump
```

图 1.1 4 查看使能后的环境变量

使用 `arm-poky-linux-gnueabi-gcc -v` 指令可以查看 `gcc` 版本, 表明环境变量已经生效。

```
arm-poky-linux-gnueabi-gcc --version
```

```
allentek@ubuntu:~$ arm-poky-linux-gnueabi-gcc --version
arm-poky-linux-gnueabi-gcc (GCC) 5.3.0
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
allentek@ubuntu:~$
```

图 1.1 5 查看 gcc 版本信息

## 第二章 搭建 Qt Creator 交叉编译环境

我们想要编译 Qt 应用程序在开发板上运行起来, 那么需要搭建交叉编译环境, 使用 ARM 平台的交叉编译器编译的程序才能在 I.MX6U 开发板上运行起来, 正点原子出厂系统所使用的交叉编译器为 fsl-imx-x11-glibc-x86\_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh。出厂系统 Qt 版本已经升级为 Qt5.12.9, 那么我们需要下载 Qt5.12.9 的 Qt 版本。

### 2.1 下载 Qt Creator

我们想要在 Ubuntu 平台开发 Qt, 那么必须要有相应的 IDE 环境, 那么 Qt Creator 就是 Qt 的 IDE(集成环境), 我们就可以使用 Qt Creator 快速编程了。进入网址 <http://download.qt.io/archive/qt/5.12/5.12.9/> 下载 Qt 5.12.9。Qt 官方已经把 Qt Creator 封装在下面这个 qt-opensource-linux-x64-5.12.9.run。我们需要下载这个 qt-opensource-linux-x64-5.12.9.run 直接在 Ubuntu 安装即可。

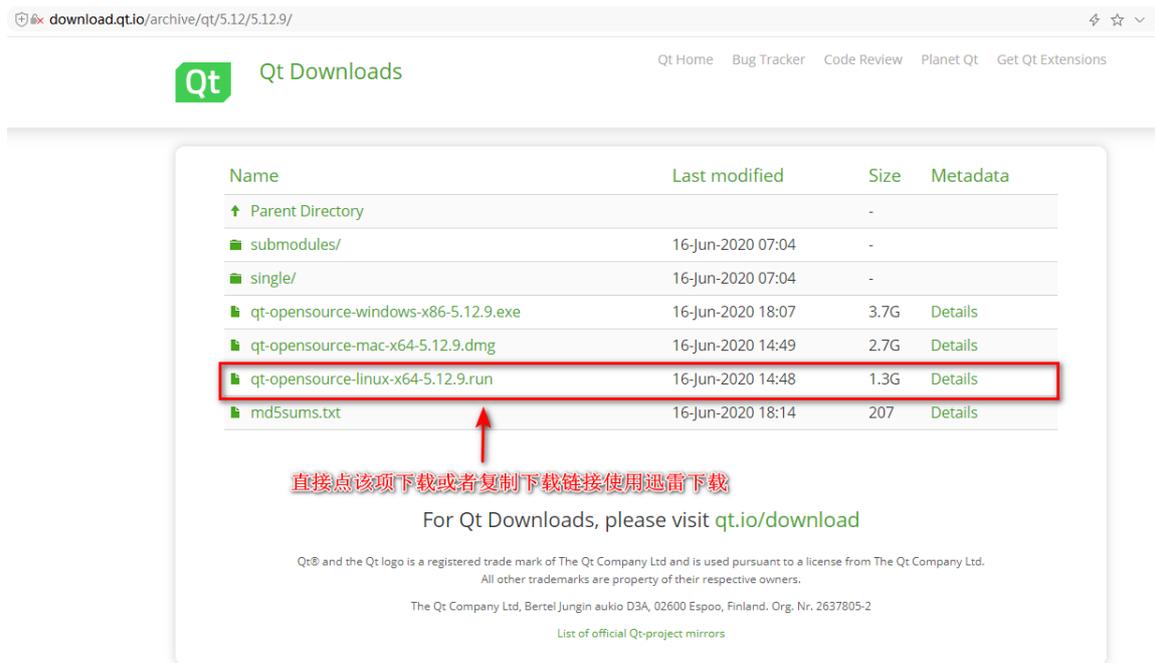


图 5.1 1 点击下载 Qt5.12.9

编者喜欢用指令下载, 编者复制上面的下载地址, 直接在 Ubuntu 终端使用 wget 指令下载。如下。



图 5.1 2 使用 wget 指令在 Ubuntu 终端直接下载

## 2.2 安装 Qt Creator

下载完成得到 qt-opensource-linux-x64-5.12.9.run 这个文件。

赋予该文件可执行权限，再执行安装。

```
chmod u+x qt-opensource-linux-x64-5.12.9.run
```

```
sudo ./qt-opensource-linux-x64-5.12.9.run // 建议加 sudo，否则安装在当前目录下。
```

如无意外，则弹出 Qt 5.12.9 欢迎安装界面。

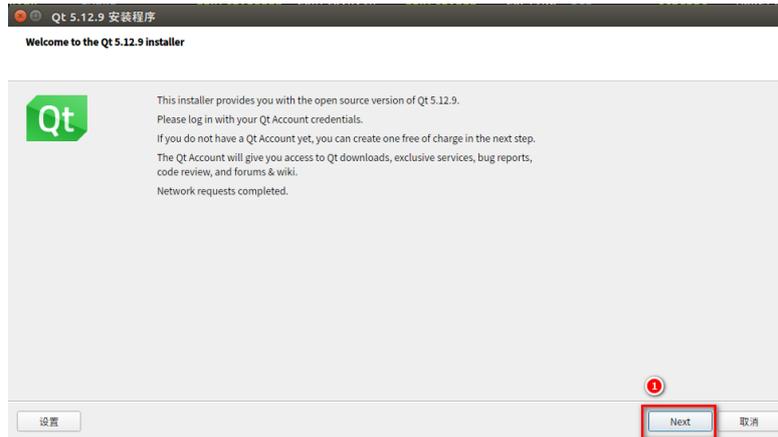


图 5.1 3 欢迎安装界面

这里需要填写 Qt 帐号，去 Qt 官网 <https://www.qt.io/>注册一个帐号。填写后再点击 Next。

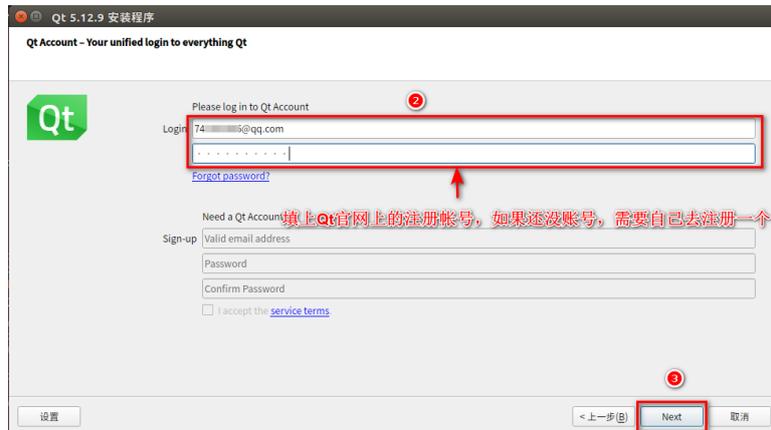


图 5.1 4 填写 Qt 帐号

同意使用条款。按如下步骤操作。

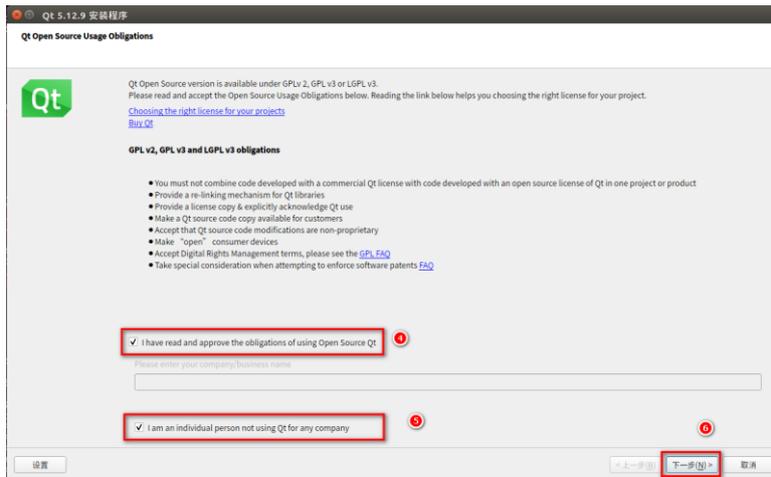


图 5.1 5 同意使用条款

欢迎安装界面，直接点击下一步。

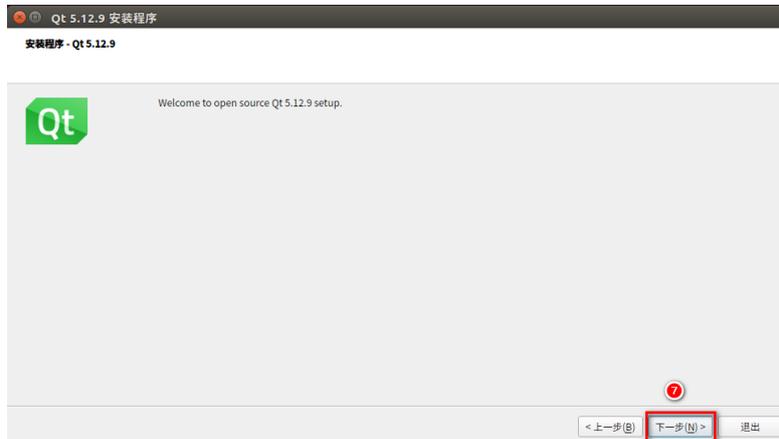


图 5.1 6 欢迎安装界面

选择安装目录，建议默认即可。会安装在/opt 目录下。点击下一步。

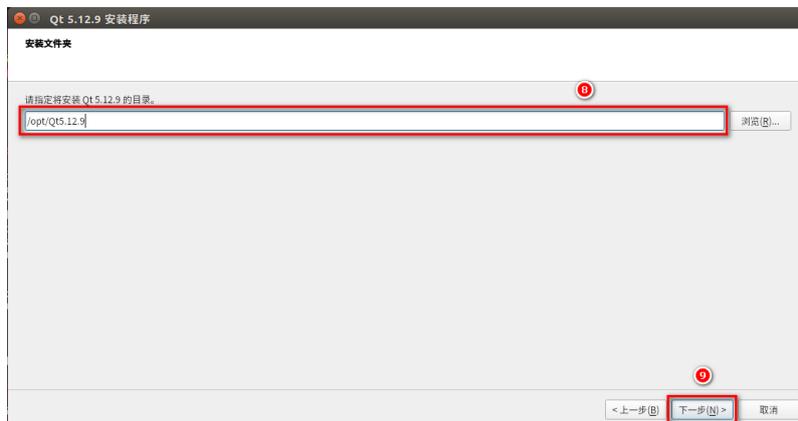


图 5.1 7 默认安装在/opt 目录下

按需要安装，除了安卓选项我们都选择安装。点击下一步。



图 5.1 8 勾选安装选项

同意许可协议，点击下一步。

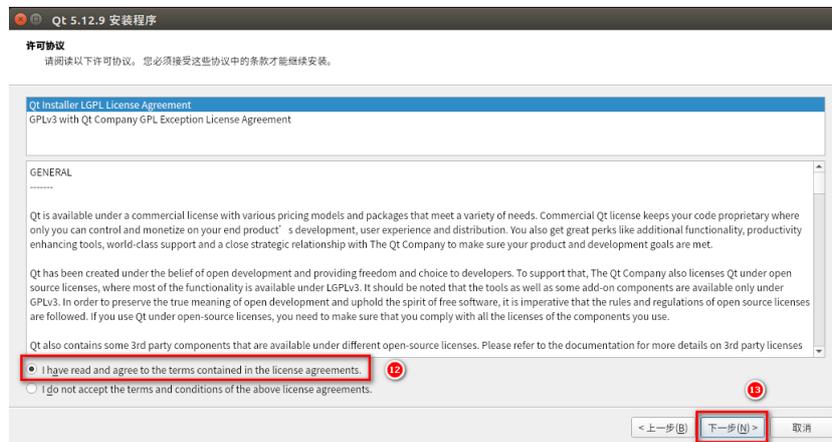


图 5.1 9 同意许可协议

准备安装，点击安装

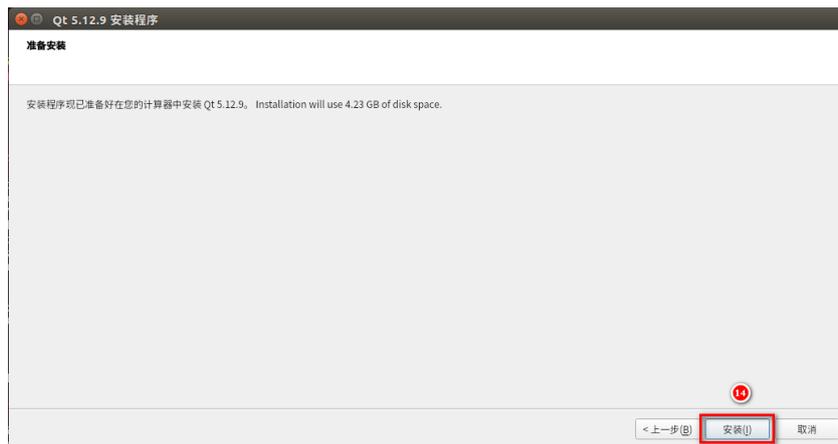


图 5.1 10 开始安装

安装完成, 点击完成

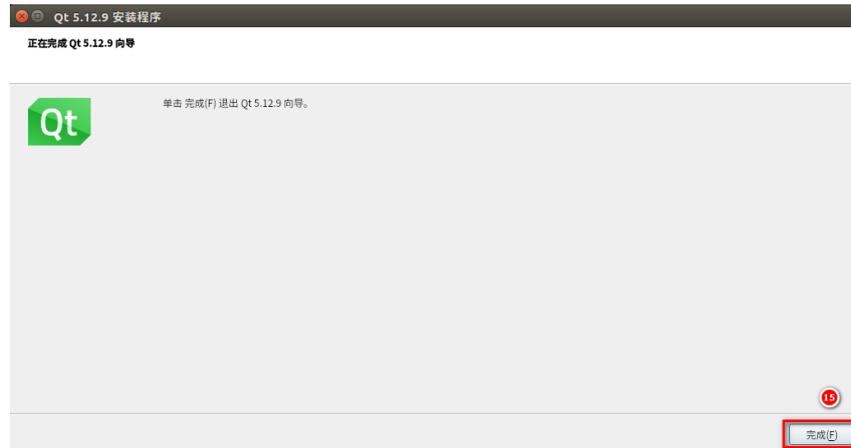


图 5.1 11 安装完成

## 2.3 配置 Qt Creator Kits

Kit 译作套件, 也就是开发编译环境套件, 我们可以搭建不同平台的套件, 以不同的套件编译出不同平台的应用程序, 也验证了 Qt 跨平台的特性!

这里我们需要在脚本里写入设置使能环境变量的指令, 首先打开此脚本。(若安装时和编者安装的目录一样则可以使用如下指令打开)

```
sudo vi /opt/Qt5.12.9/Tools/QtCreator/bin/qtcreator.sh
```

在 qtcreator.sh 里的第一行插入如下指令, 如下图所示。

```
source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
```

```
1 source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
2 #! /bin/sh
3
4 # Use this script if you add paths to LD_LIBRARY_PATH
5 # that contain libraries that conflict with the
6 # libraries that Qt Creator depends on.
7
8 makeAbsolute() {
9     case $1 in
```

图 2.3 1 后台运行 Qt Creator

保存脚本文件。使用如下指令打开 Qt Creator。“&”的作用是后台运行。**(推荐使用这种方式打开 Qt Creator)**

```
/opt/Qt5.12.9/Tools/QtCreator/bin/qtcreator.sh &
```

```
alientek@ubuntu:~$ /opt/Qt5.12.9/Tools/QtCreator/bin/qtcreator.sh &
[1] 10672
alientek@ubuntu:~$
```

图 2.3 2 后台运行 Qt Creator

也可以在左上角的软件中心栏输入搜索出“Qt Creator”图标后再单击打开。(这里**并不推荐**使用图标打开 Qt Creator, 否则编译过程中引用到其他环境变量可能会出现编译报错)

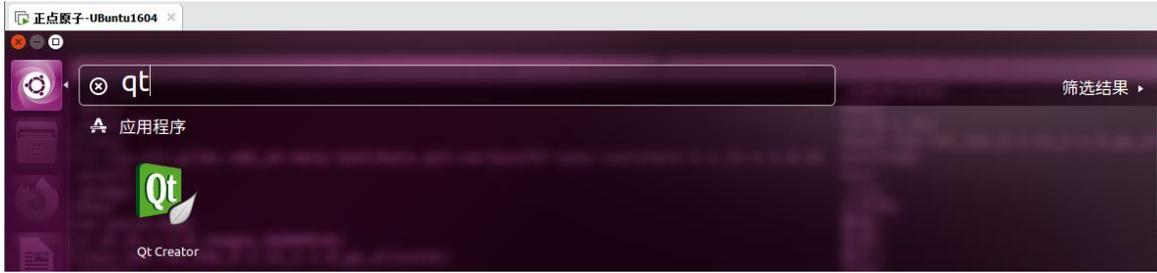


图 2.3 3 从软件搜索栏打开 Qt Creator

打开 Tools(工具), 打开 Options(选项)。

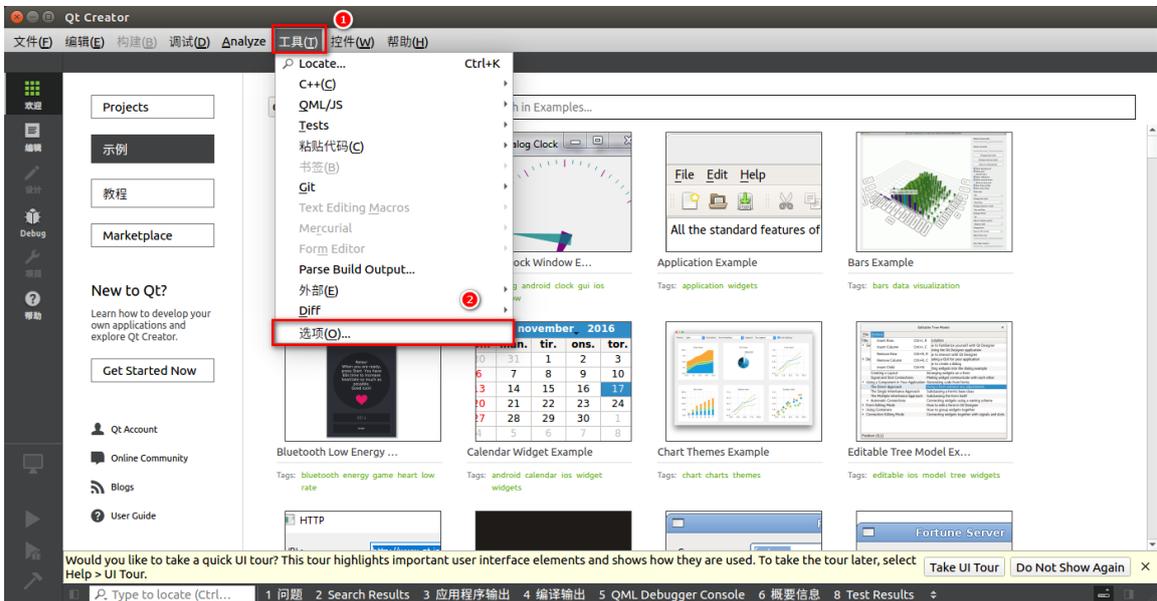


图 2.3 4 打开“选项”

配置 Qt Versions, 点击添加, 选择我们 [1.1](#) 小节安装的交叉编译器路径下的 qmake, 路径为 `/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/qmake`, 用于生成 Makefile, 以编译程序。同时改版本名称为 `ATK-I.MX6U Qt %Qt:Version}`。

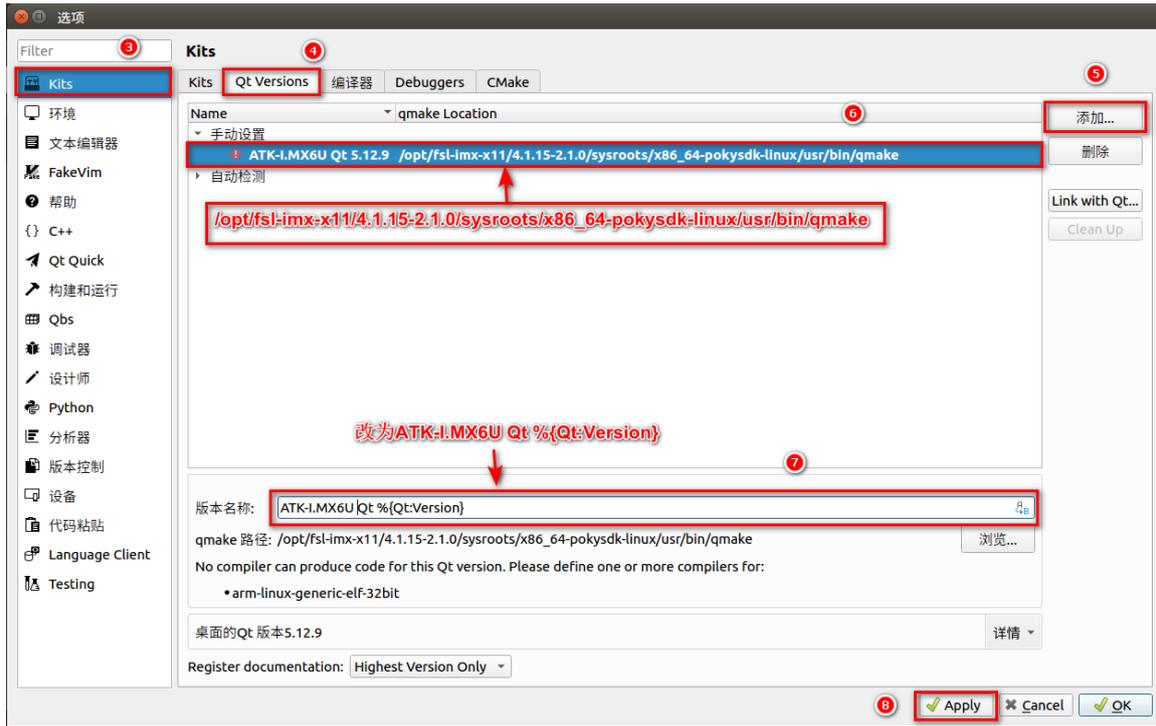


图 2.3 5 配置 Qt Versions

点击添加，选择 C++。

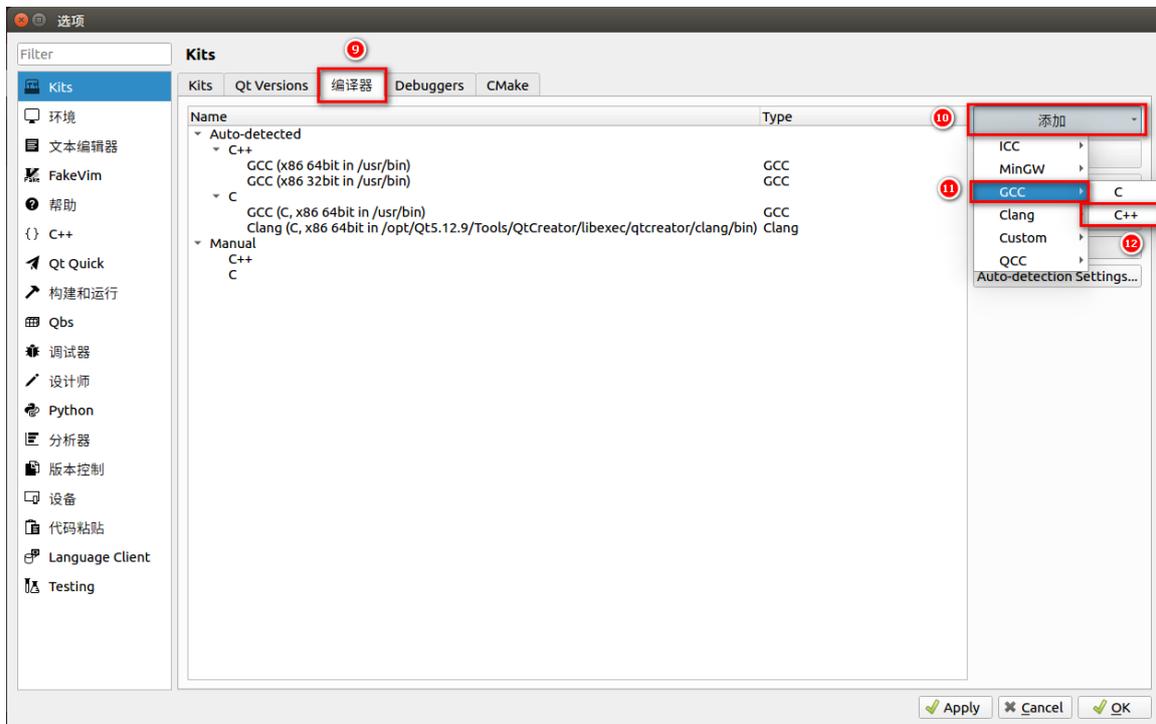


图 2.3 6 添加 C++

配置编译器, 编译器的路径为我们 1.1 小节安装的交叉编译器。路径为 `/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-g++`。将“名称”改为 `ATK-I.MX6U-GCC`。

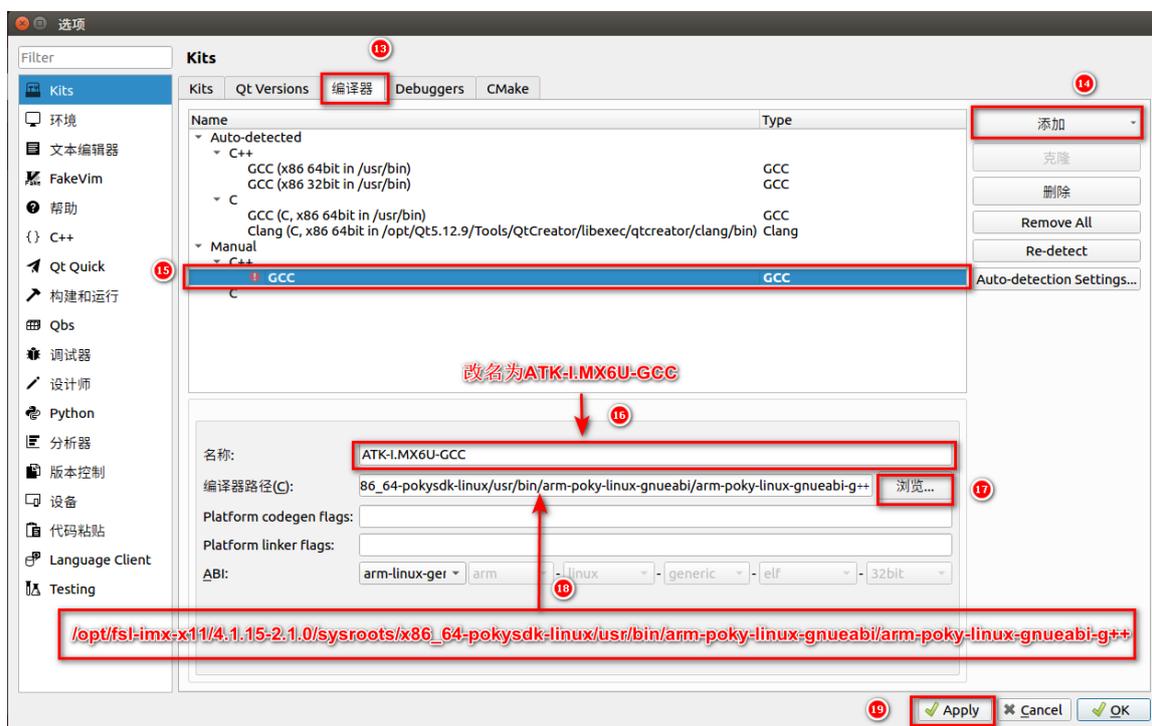


图 2.3 7 配置编译器

配置 Kits, 将“名称”改为 `ATK-I.MX6U`。在 Qt mkspec 处写上“`linux-oe-g++`”。其他按下图操作设置。不要漏了其中一项, 否则可能编译错误。

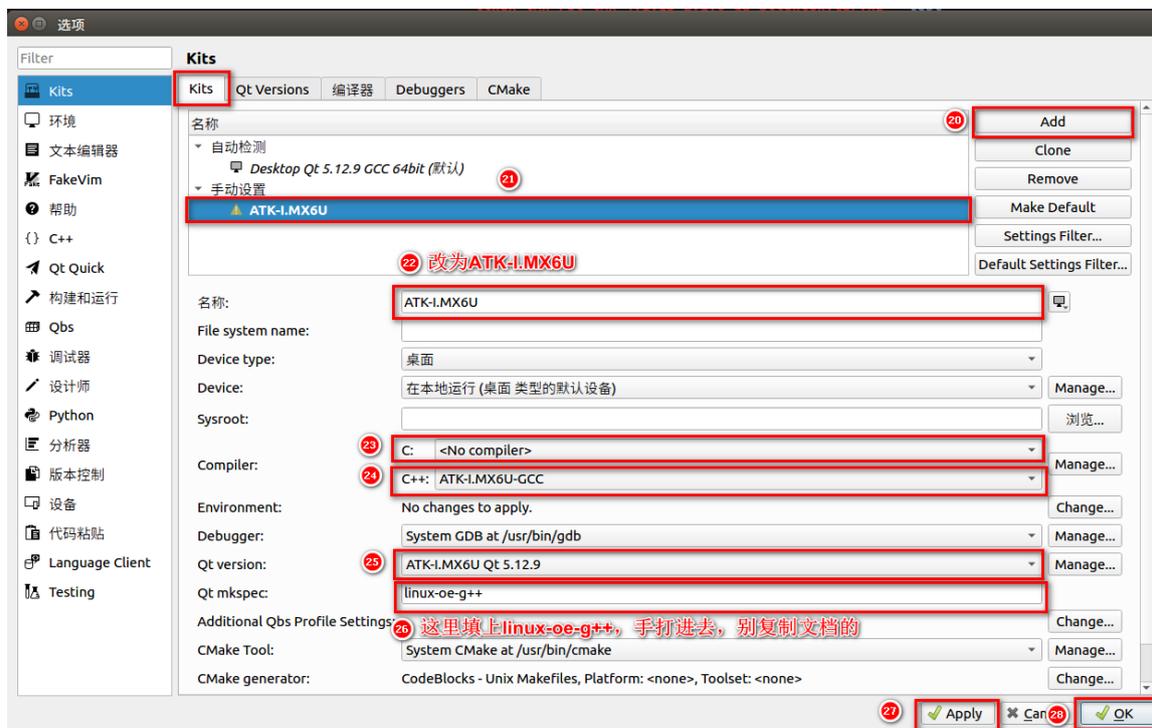


图 2.3 8 配置 Kits

## 2.4 验证搭建的交叉编译 Kits

新建工程

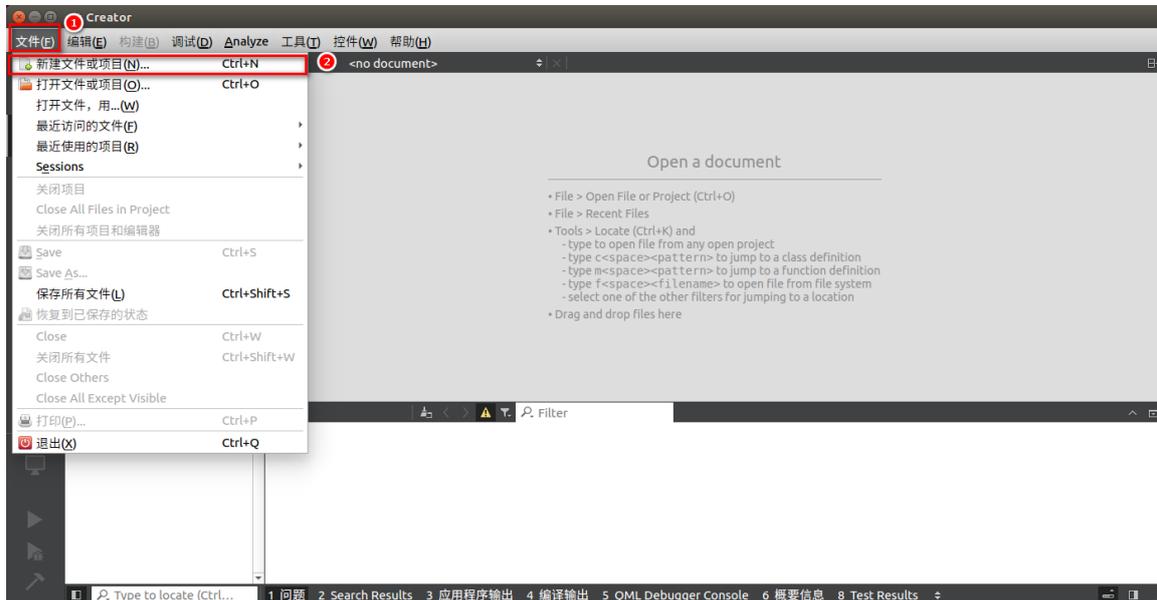


图 2.4 1 新建项目

选择 Application 项目和 Qt Widgets Application 模板。

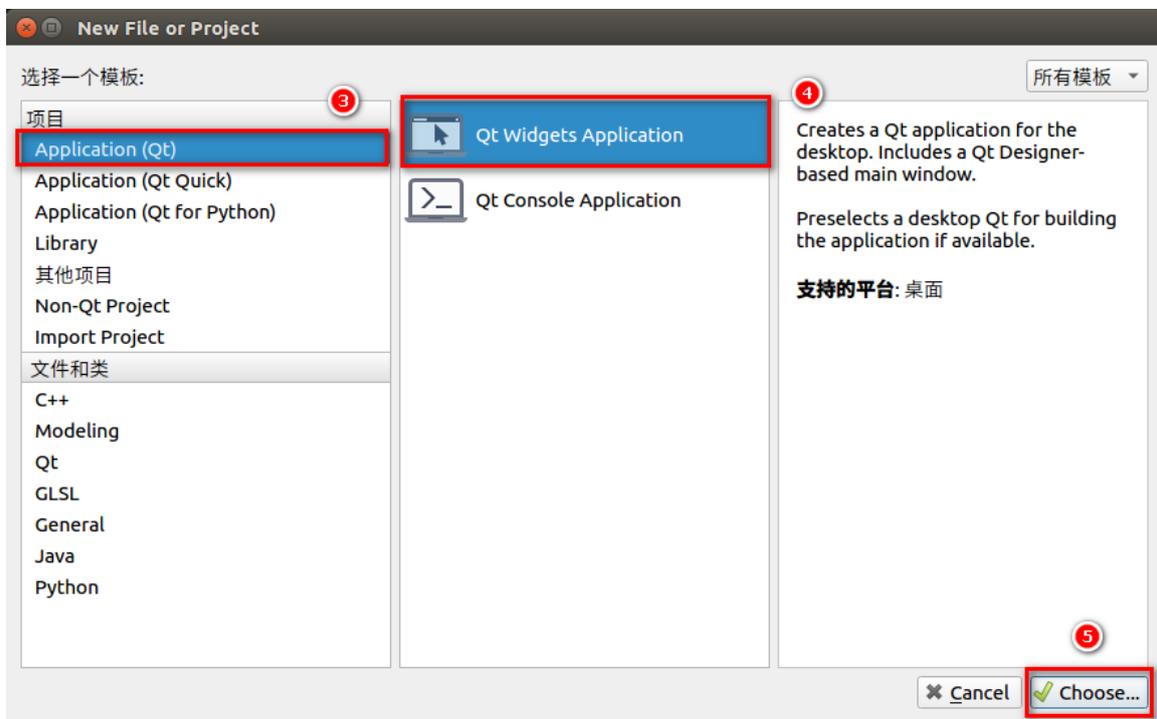


图 2.4 2 选择项目与模板

项目命名为 test, 选择工程的位置, 这里位置不要随便选择, 建议放在家目录下(/home/用户名)。

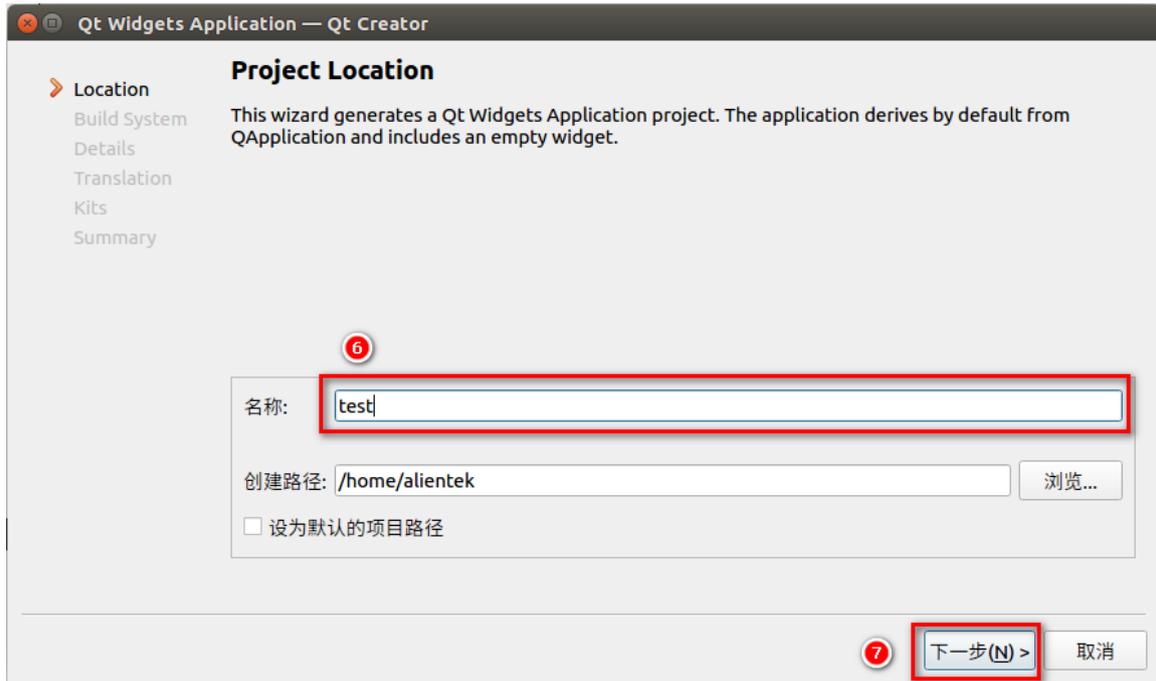


图 2.4 3 项目命名为 test

默认使用 qmake, 点击下一步。

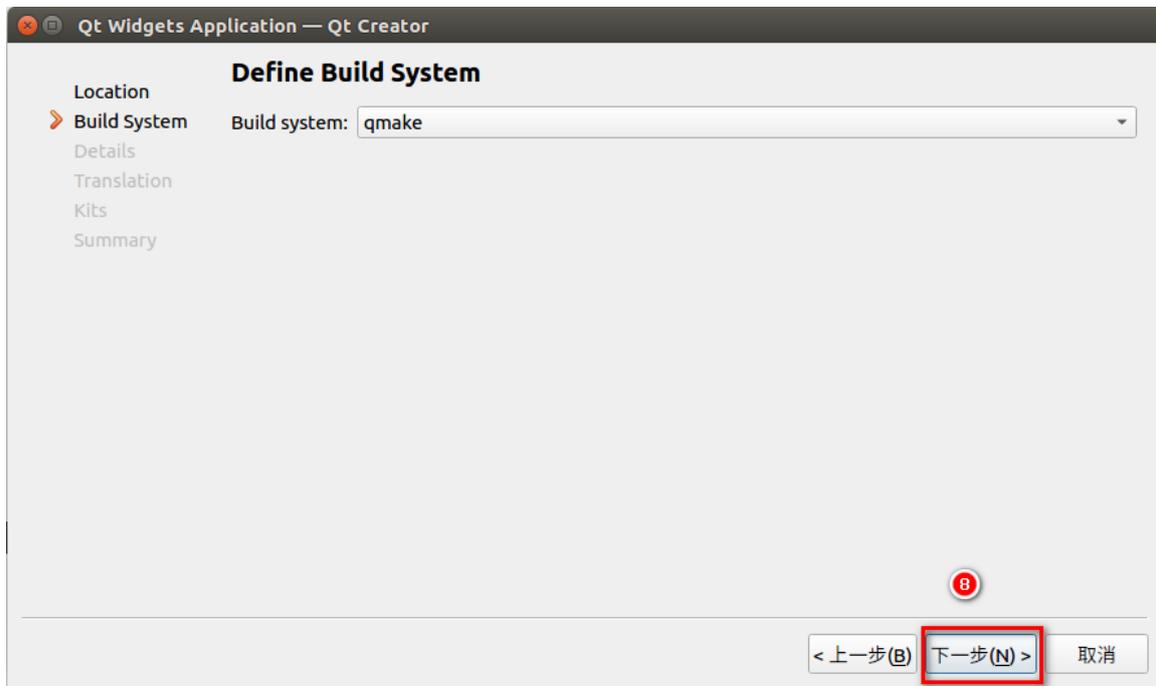


图 2.4 4 默认构建系统

选择类模板, 默认 QMainWindow 类即可。

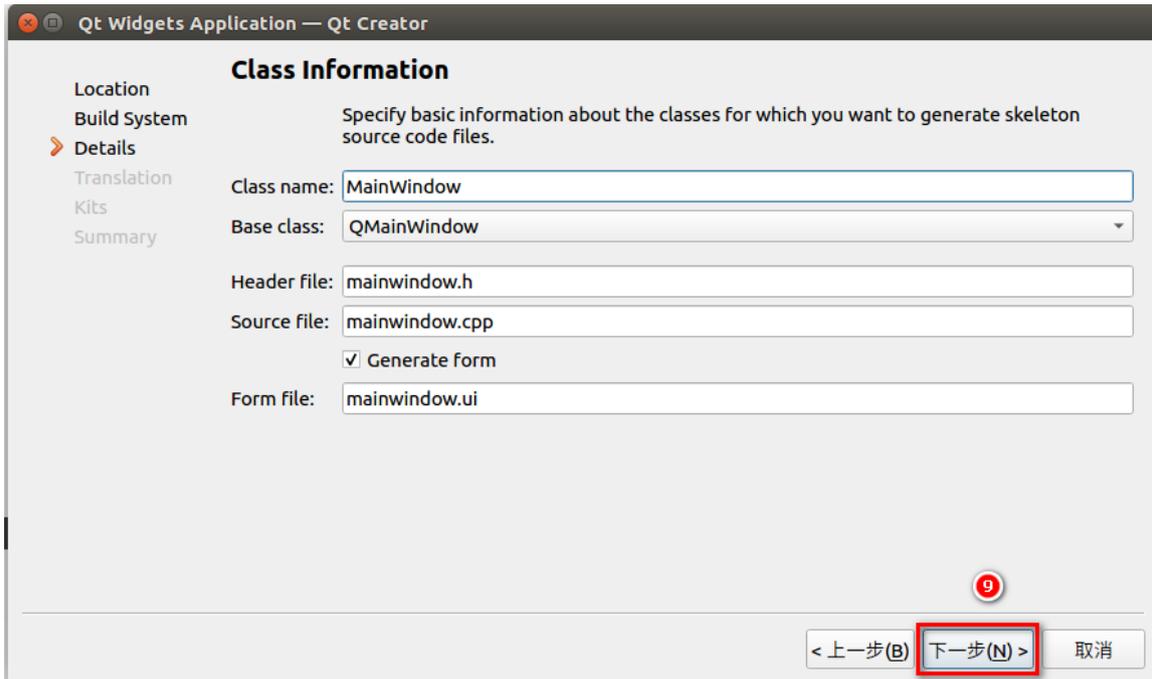


图 2.4 5 选择类模板

是否选择文件翻译, 默认无, 点击下一步。

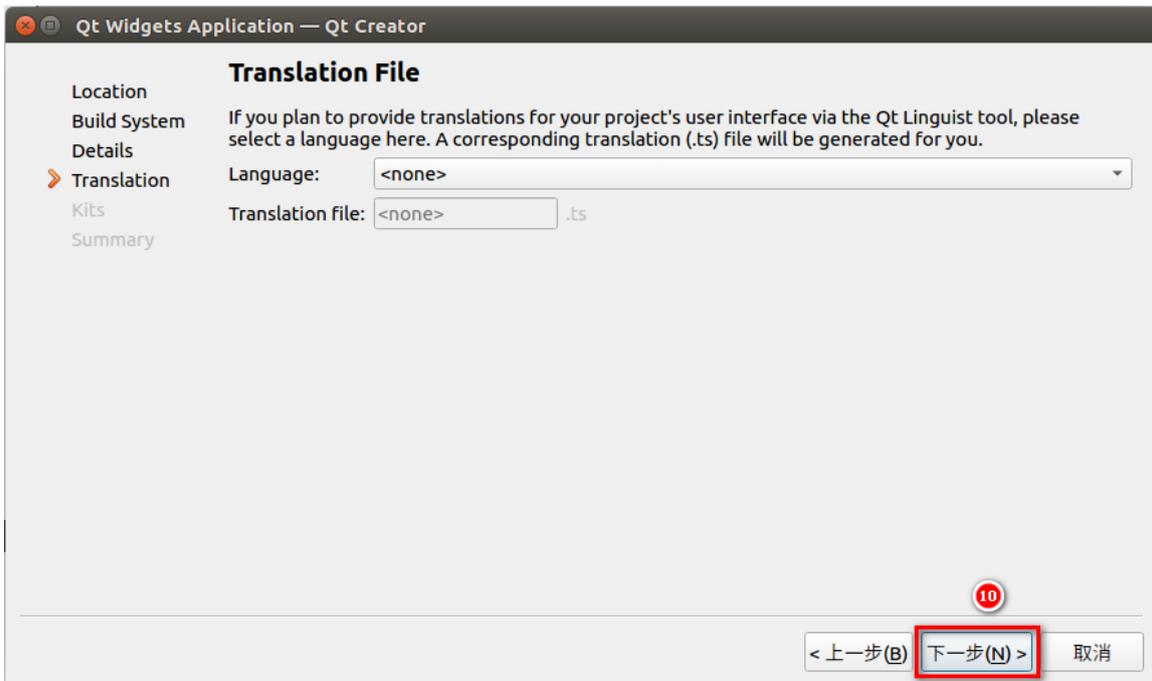


图 2.4 6 跳过文件翻译

选择套件, 这里可以两个套件一起选, 编译时切换选择 I.MX6U ARM 平台的套件, 如下。

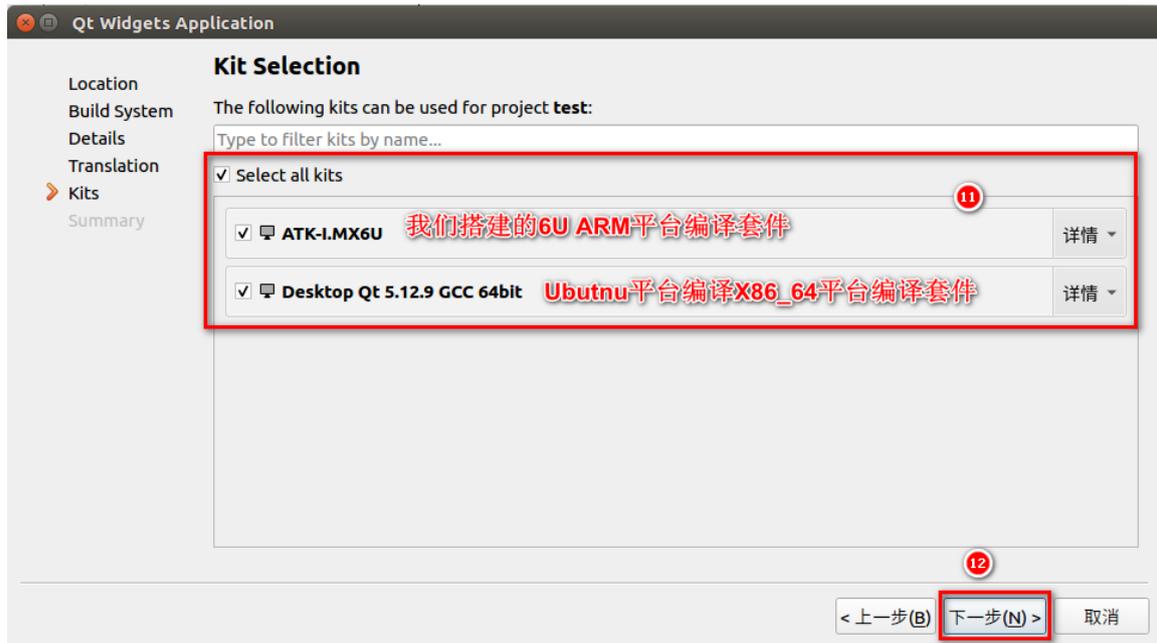


图 2.4 7 选择 Kits

默认, 点击 Finish, 完成。

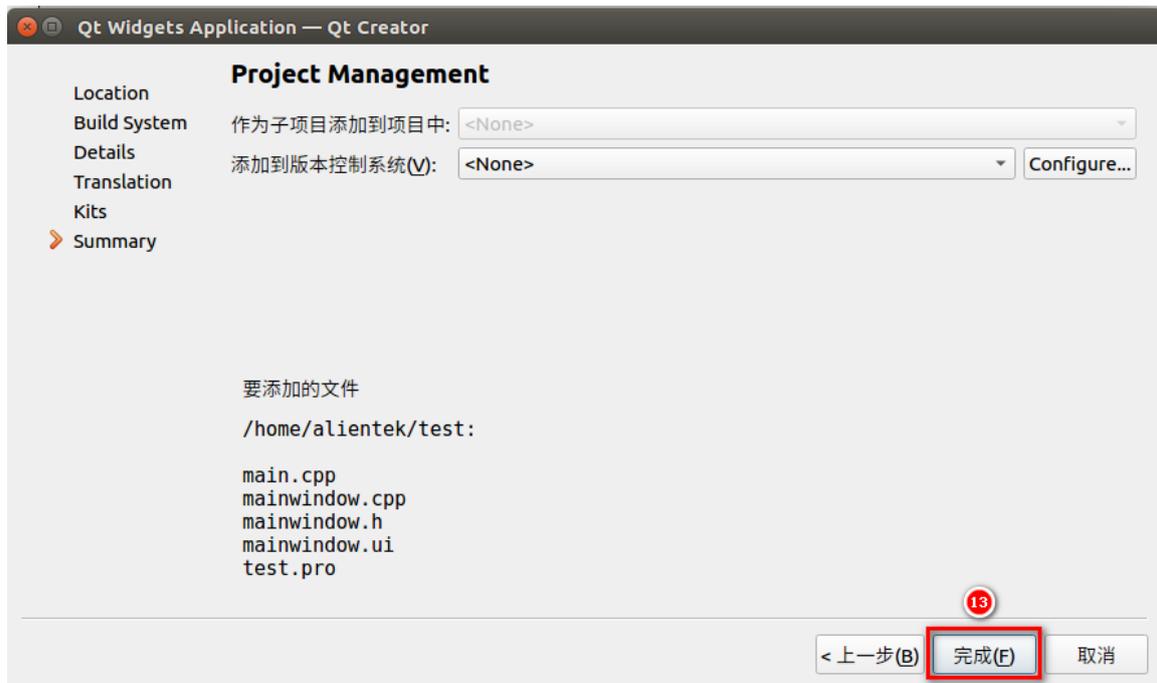


图 2.4 8 跳过版本控制

如下图选择 ATK-I.MX6U Kit 编译, 其中可以看到 Qt Creator 把程序中的类当作未识别的类。这个可以不用理会。

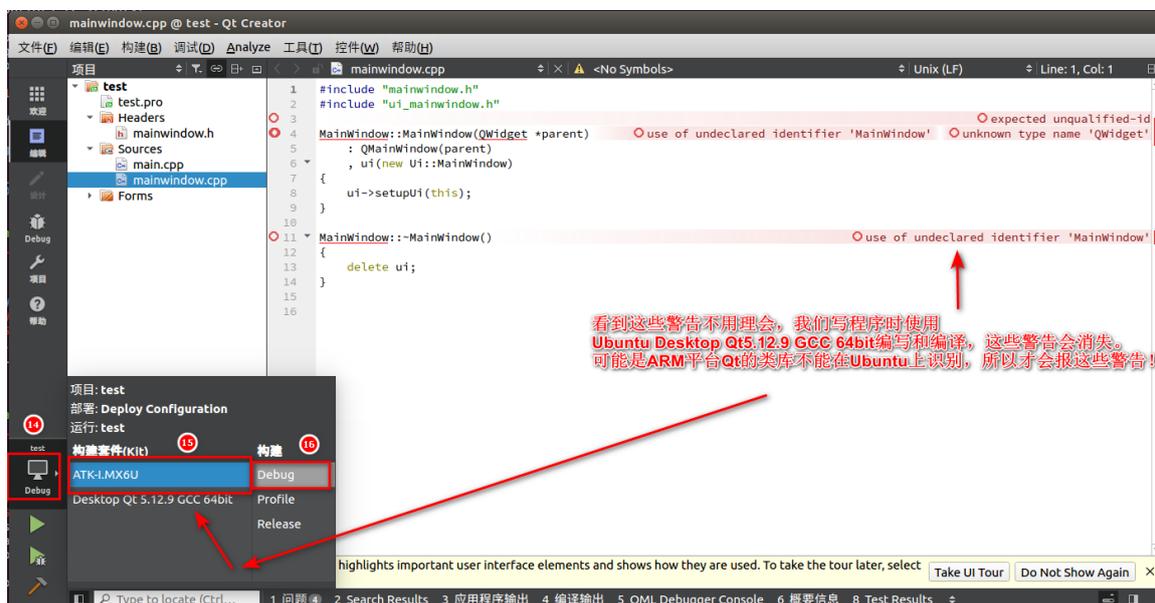


图 2.4 9 切换 ATK-I.MX6U

想解决这个问题也有方法。因为高版本的 Qt creator 多了一个 ClangCodeModel 插件, 把这个插件去掉, 重启 Qt creator 就不会有这个报错信息了。点击“帮助”->“关于插件”, 在 C++ 栏找到“ClangCodeModel”, 取消勾选即可。点击关闭, 会提示是否要现在重启, 点击 Now 即可重启生效。

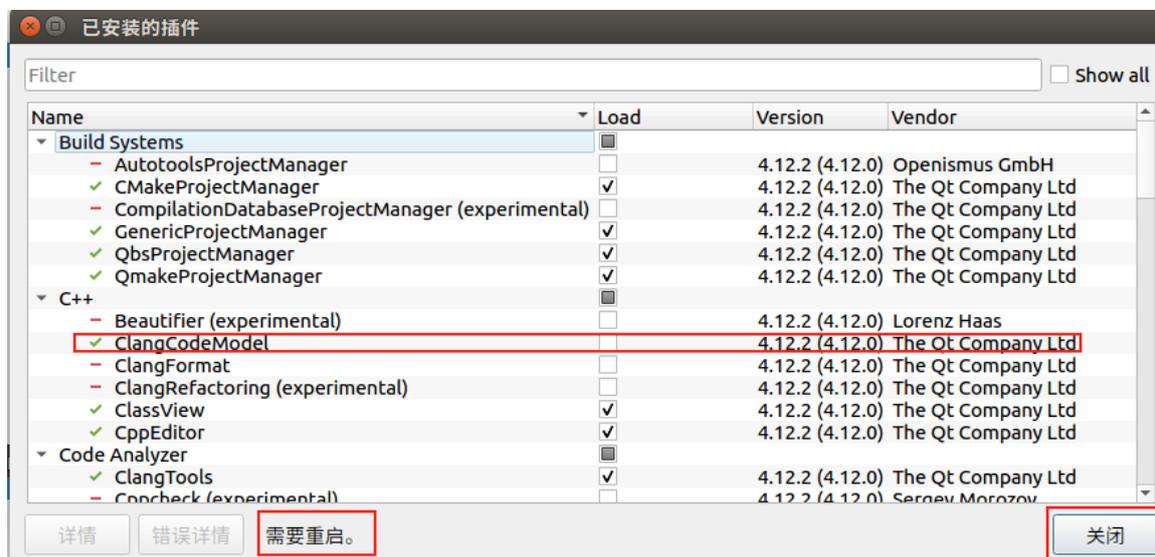


图 2.4 10 取消勾选

开始编译, 点击构建, 不要点击运行! 在 Ubuntu 平台上是运行不了用 ARM 编译器编译出来的程序的!

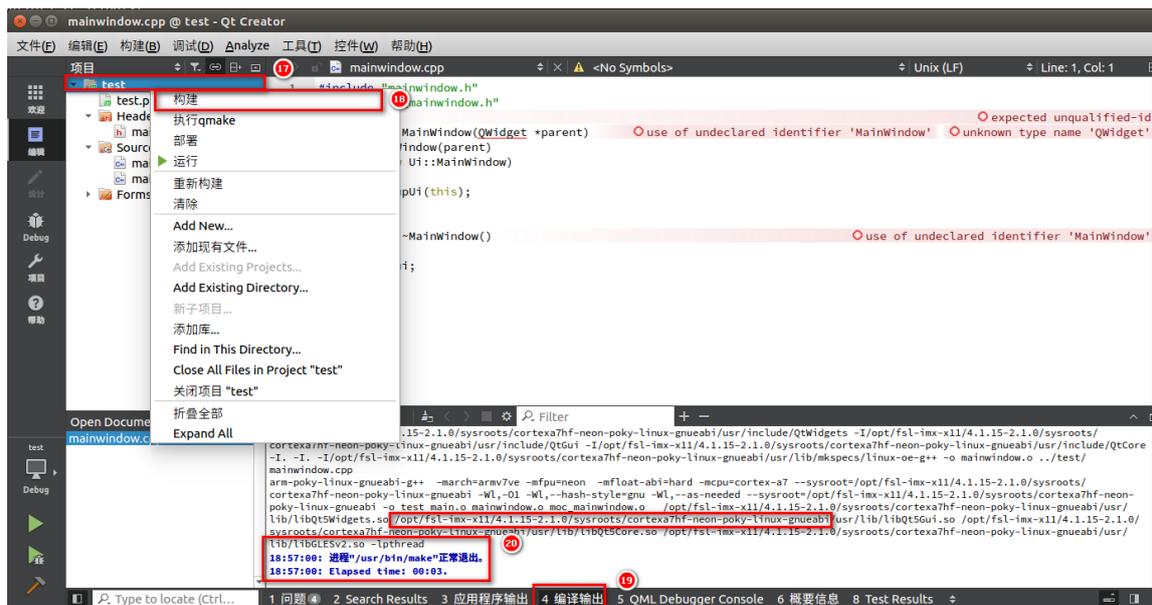


图 2.4 11 编译 ARM 平台的应用程序成功

在 test 工程的同级目录下找到 build-test-ATK\_I\_MX6U-Debug 文件夹，查看编译出来的可执行程序 test。我们拷贝到出厂文件系统目录下，在串口终端直接执行 ./test 就可以了。

```
cd build-test-ATK_I_MX6U-Debug/
ls
```

```
allentek@ubuntu:~$ cd build-test-ATK_I_MX6U-Debug/
allentek@ubuntu:~/build-test-ATK_I_MX6U-Debug$ ls
main.o mainwindow.o Makefile moc_mainwindow.cpp moc_mainwindow.o moc_predefs.h test ui_mainwindow.h
allentek@ubuntu:~/build-test-ATK_I_MX6U-Debug$
```

图 2.4 12 编译出来的 ARM 平台可执行程序

## 2.5 远程调试 Qt 程序

### 2.5.1 rsync 方式 Qt 远程调试

在 Qt Creator 中默认情况下，会使用 sftp 或 rsync 发送程序到板卡。由于正点原子 I.MX6 U 出厂 Qt 文件系统（文件系统 V1.9 及之后的）里有 rsync 指令，因此可以很方便地使用此方式进行 Qt 远程调试。首先我们要确认下当前出厂文件系统版本号，在出厂系统执行以下指令。

```
cat /etc/version
```

```

root@ATK-IMX6U:~# cat /etc/version
Author: 正点原子@qq1252699831

Version v1.0 - Version v1.5
Note:
无记录

Version: v1.6
Date:2020.11.18
Note:
1.升级为Qt 5.12.9文件系统版本

version: v1.7
Date:2021.01.07
Note:
1.修复耳机开机默认没有声音;
2.修复Qt音量70%以下就没有音量的问题等

version: v1.8
Date:2021.01.12
Note:
1.修复Qt插入gif动画无法显示的问题

version: v1.9
Date:2021.1.30
Note:
1.添加rsync指令
2.添加qt mysql支持
3.添加正点原子camera_settings ov5640_camera ov2640_camera ov772x_camera应用程序
4.添加sqlite,mysql,cups,hplip的支持
root@ATK-IMX6U:~#

```

图 2.5.1 1 查看当前文件系统版本

或者直接执行 rsync 指令,也可以查看是否支持 rsync,如下图所示就是支持 rsync。

```

rsync

root@ATK-IMX6U:~# rsync
rsync version 3.1.2 protocol version 31
Copyright (C) 1996-2015 by Andrew Tridgell, Wayne Davison, and others.
Web site: http://rsync.samba.org/
Capabilities:
 64-bit files, 64-bit inums, 32-bit timestamps, 64-bit long ints,
 socketpairs, hardlinks, symlinks, IPv6, batchfiles, inplace,
 append, ACLs, xattrs, iconv, symtimes, prealloc

rsync comes with ABSOLUTELY NO WARRANTY. This is free software, and you
are welcome to redistribute it under certain conditions. See the GNU
General Public Licence for details.

rsync is a file transfer program capable of efficient remote update
via a fast differencing algorithm.

Usage: rsync [OPTION]... SRC [SRC]... DEST
or     rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST

```

以下是帮助信息,截图省略

图 2.5.1 2 rsync 指令

如果文件系统不是 V1.9 的,或者不支持 rsync 指令,请去网盘下载最新的出厂系统,使用最新的系统来进行。如果不想更新文件系统,可以看 2.5.2 小节无 rsync 方式 Qt 远程调试。这里推荐使用最新系统,会方便搭建远程调试环境。

打开 Qt creator 里的工具 -> 选项。

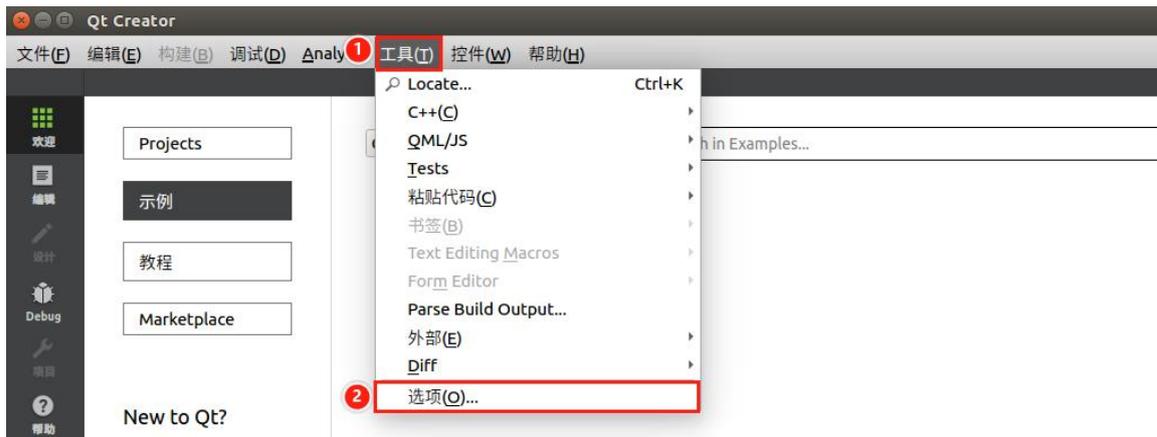


图 2.5.1 3 打开 Qt creator 的工具选项

如图所示添加 Generic Linux Device 设备。

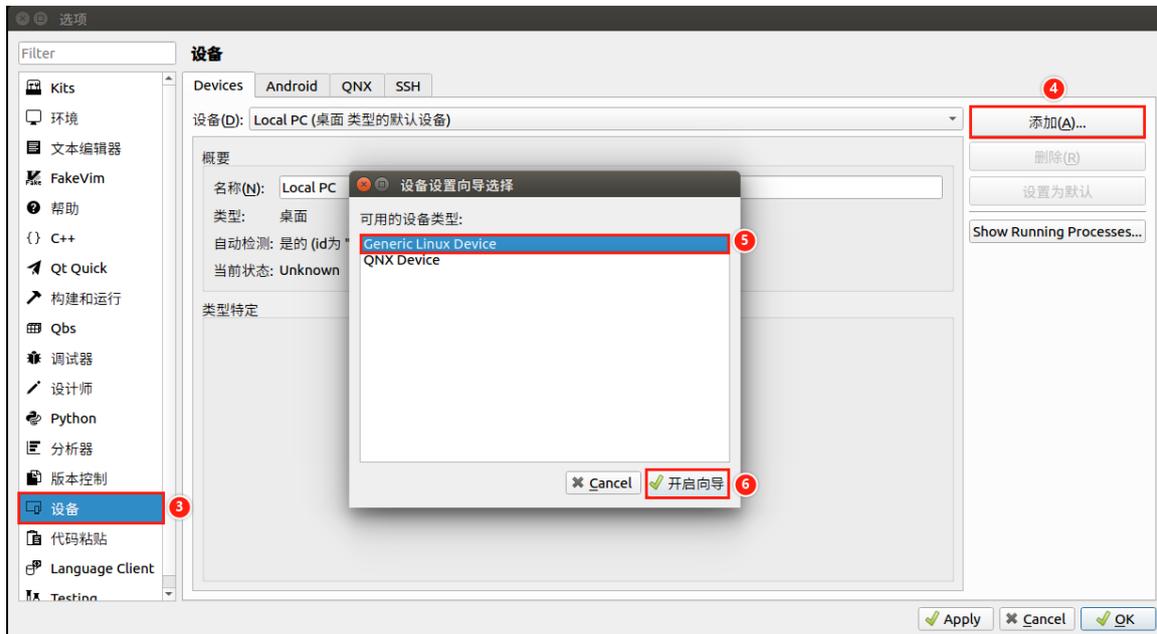


图 2.5.14 添加 Generic Linux Device 设备

依次填写配置名称、开发板 IP 地址和用户名。这里我设置名称为 IMX6U-rsync, 开发板接网线直连电脑再执行 ifconfig 指令得到开发板 IP 地址为 192.168.1.219, 出厂系统用户名为 root。开发板 IP 地址请根据自己实际情况填写。

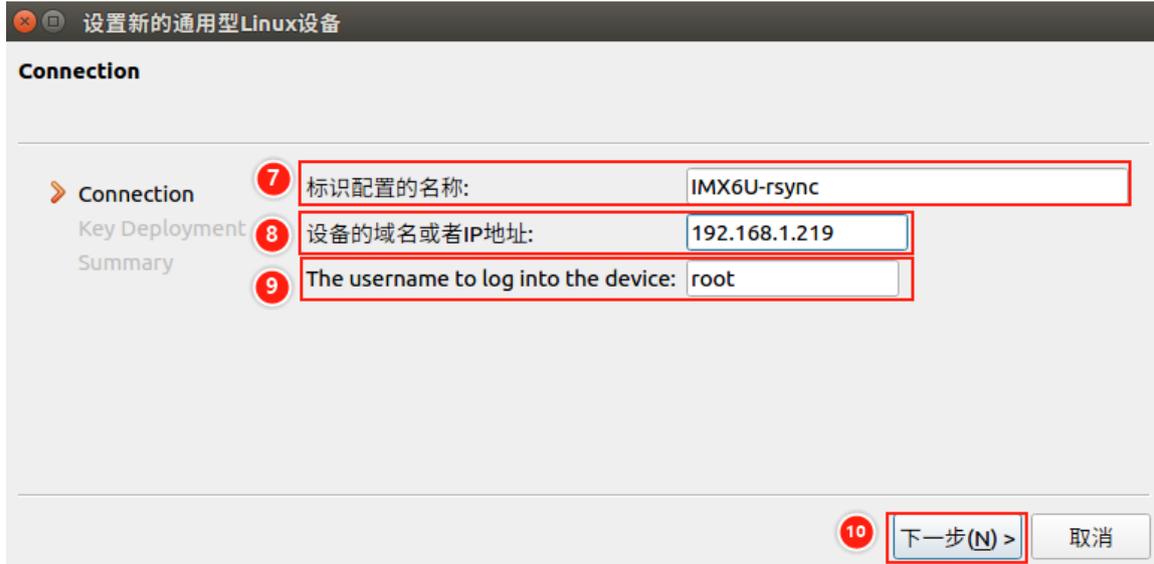


图 2.5.1 5 添加 Generic Linux Device 设备相关信息

点击下一步。

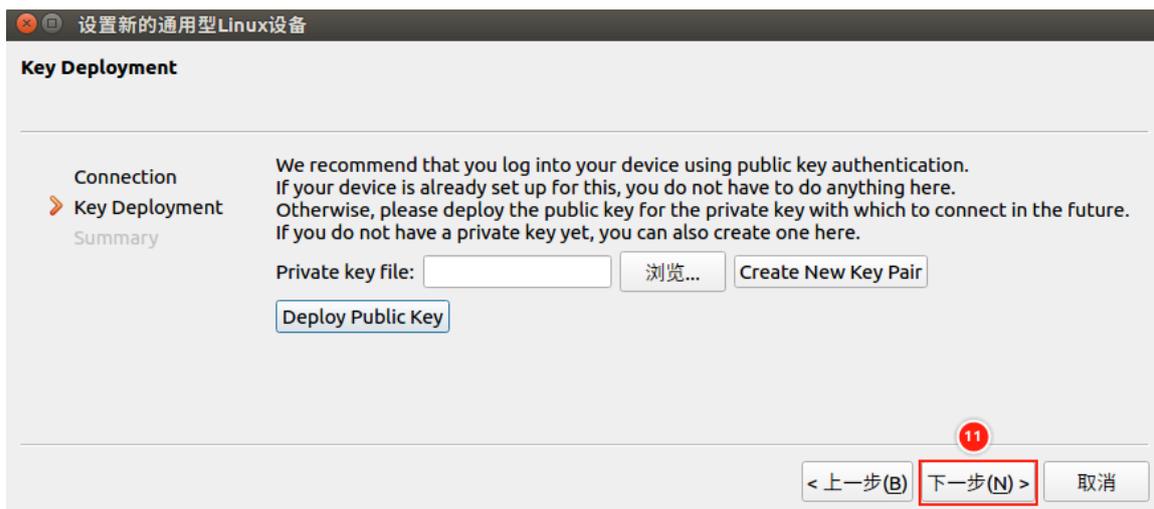


图 2.5.1 6 点击下一步

点击完成。



图 2.5.1 7 点击完成

这里会测试连接, 如果失败请检查下 Ubuntu 是否开启了 FTP 服务并关闭防火墙, 确保 Ubuntu 和开发板出厂系统能 ping 通。



图 2.5.1 8 测试设备连接

设置验证类型为 Default 并应用保存。

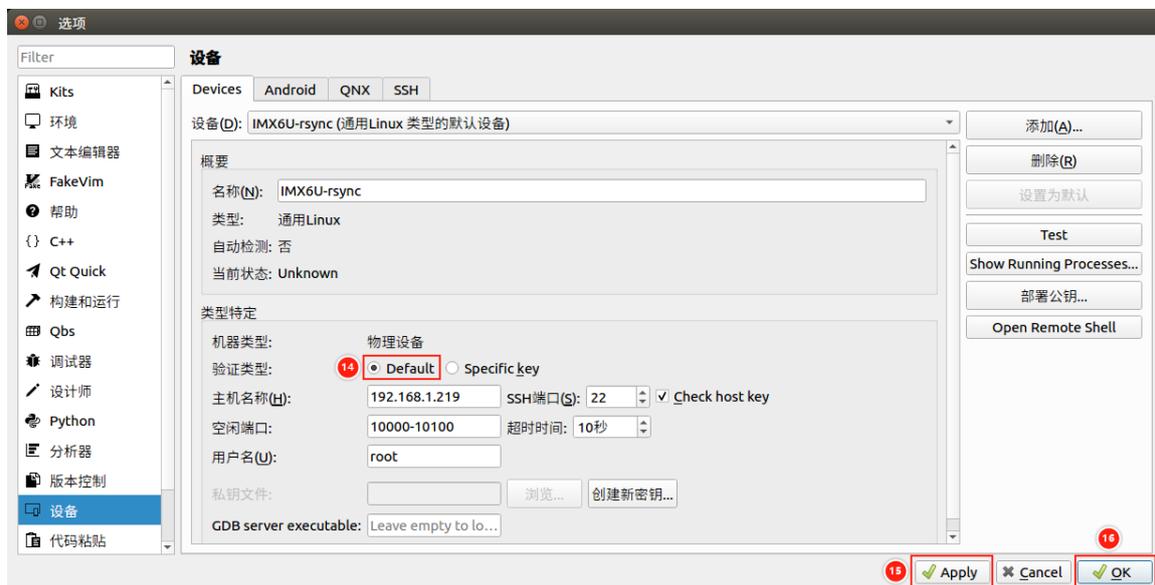


图 2.5.1 9 设置验证类型为 Default

至此, 我们的设备信息及验证方式都设置好了, 接下来设置 Kits。依次打开工具 -> 选项, 设置 Kits 配置。

其中,名称我们修改成 IMX6U-rsync, C++选择 2.3 小节设置的 ATK-IMX6U-GCC, Qt version 选择 2.3 小节里配置好的 ATK-IMX6U Qt 5.12.9, Qt mkspec 填写 linux-oe-g++. 其他具体配置参考下图。

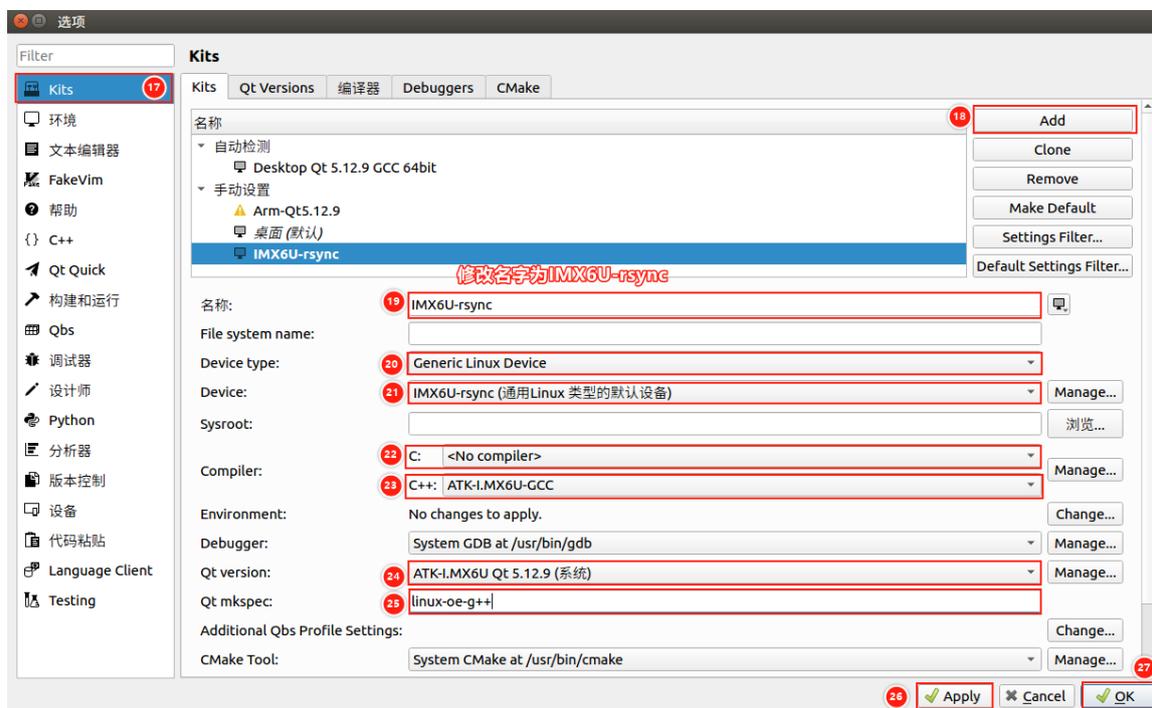


图 2.5.1 10 设置 Kits 配置

接下来就是验证 Qt 远程调试,参考 2.4 小节的方法创建一个 test 项目,选择使用刚刚创建的 IMX6U-rsync 的 Kits 套件。

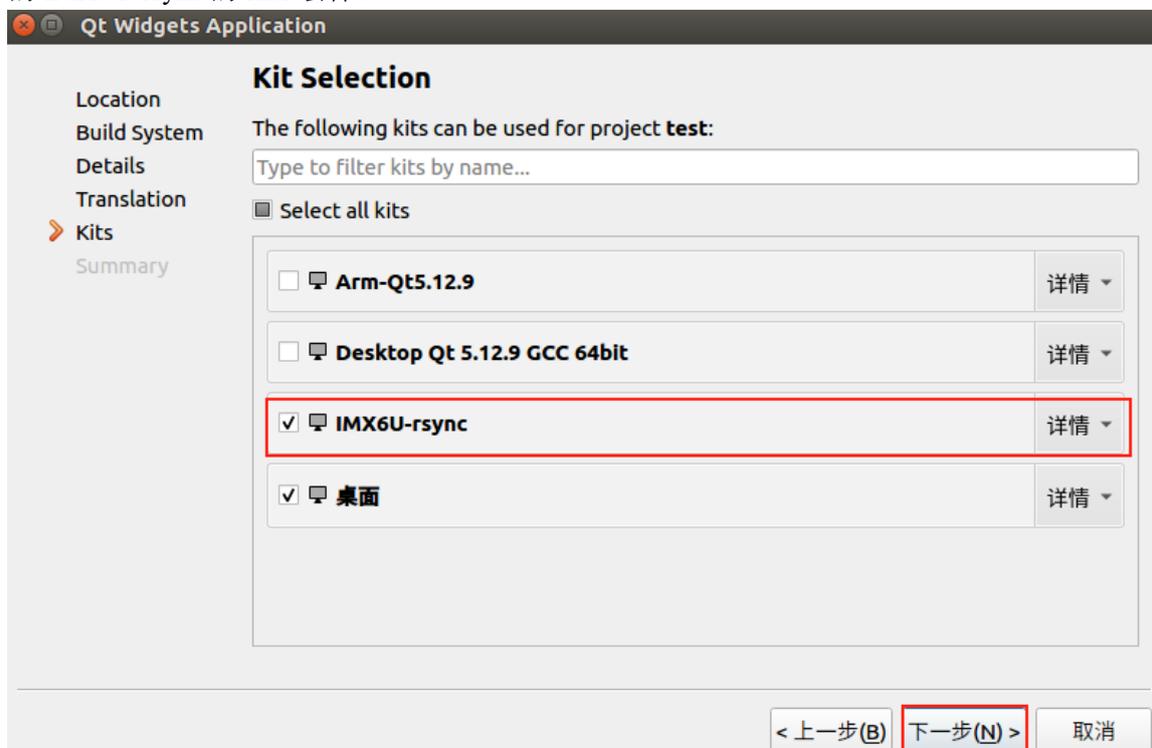


图 2.5.1 11 创建 test 项目并选择 Kits

选择使用 IMX6U-rsync 来调试。

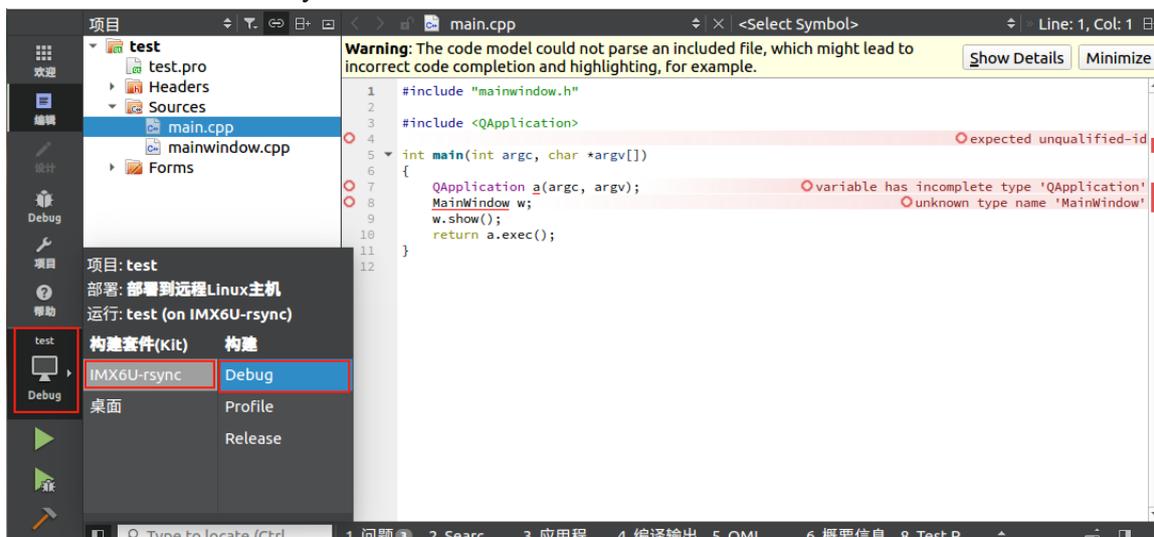


图 2.5.1 12 选择调试方式

对项目右键，在菜单栏中选择构建，再运行程序。

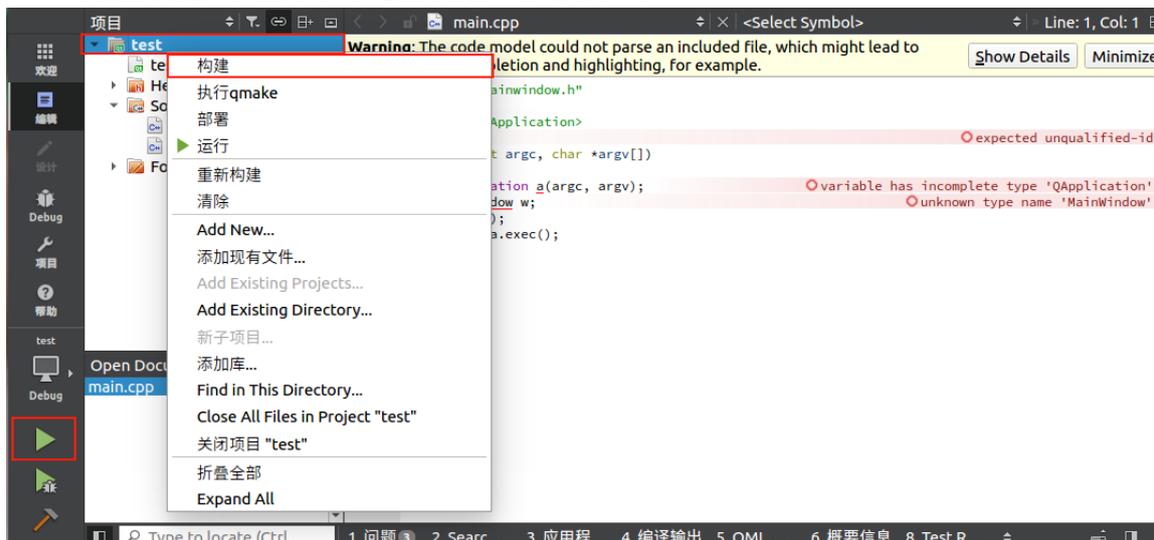


图 2.5.1 13 构建并运行项目

在应用程序输出栏可以看到此程序已运行，在开发板 LCD 屏幕上可以看到一片空白，这是因为我们运行的 test 是空白项目。至此，Qt 远程调试验证结束。

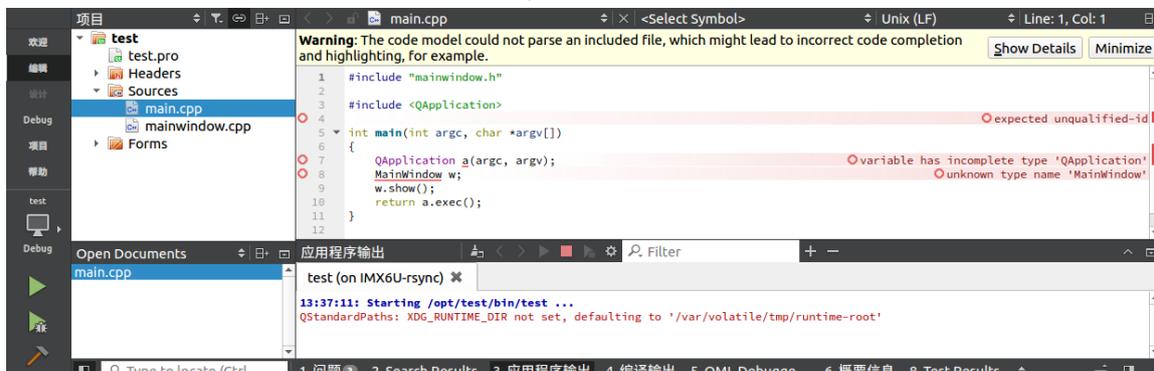


图 2.5.1 14 程序运行，开发板屏幕出现空白窗口

## 2.5.2 无 rsync 方式 Qt 远程调试

由于正点原子 I.MX6U 出厂 Qt 文件系统（文件系统 V1.8 之前的）里没有 sftp 或 rsync，所以我们需要换另外一种方法。原理很简单，就是使用 ssh 和 scp 把程序发送到板卡上再执行。

查看文件系统版本号的方法可以看 2.5.1 小节，建议更新最新系统，使用 rsync 方式进行 Qt 远程调试，会比较方便。

打开菜单栏的工具，打开选项。

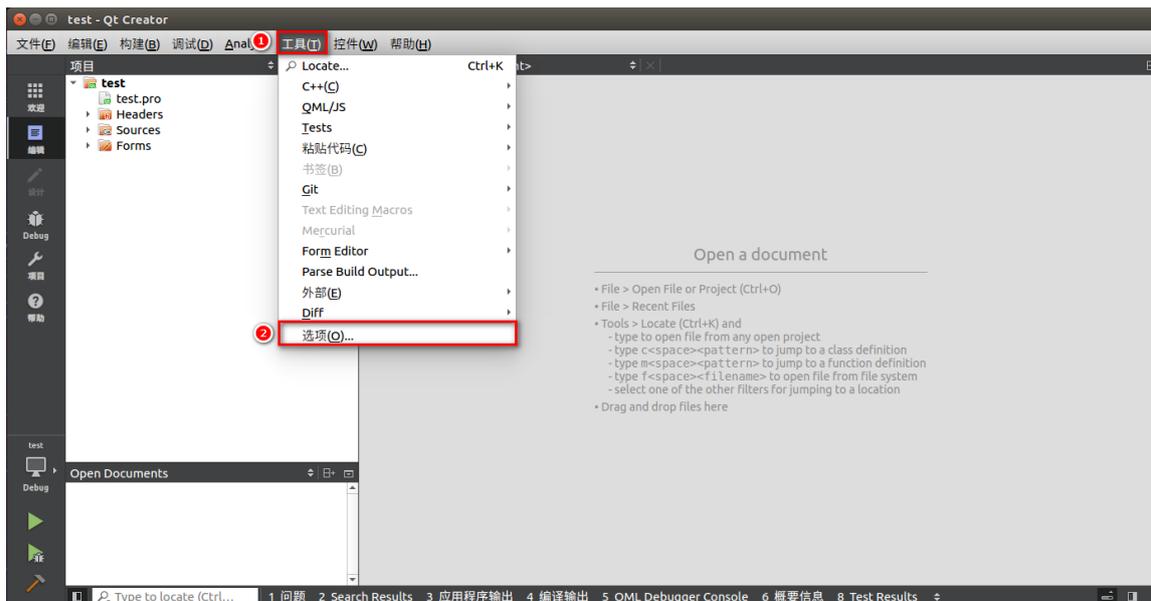


图 2.5.2 1 打开选项

点击设备处，把名称命名为“I.MX6U”。在第 6 步填写在开发板的 ip 地址。如果 ip 地址不是固定的，可以看快速体验手册第五章第 5.5 小节设置静态 ip。默认端口为 22。



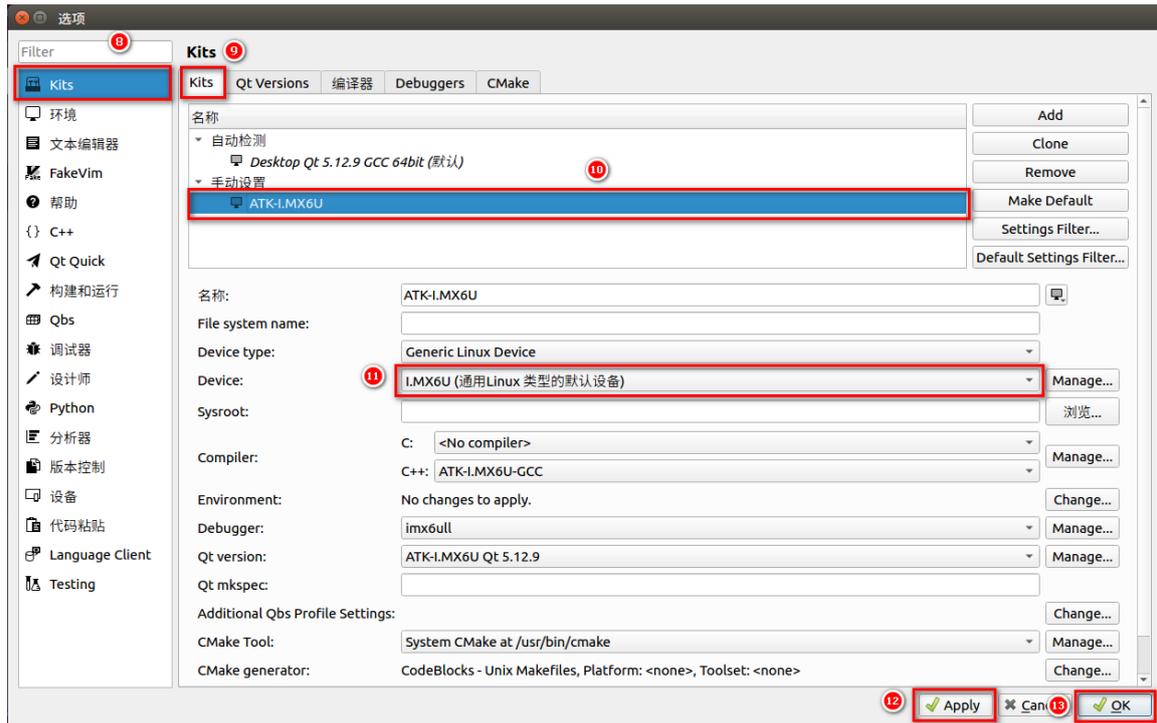


图 2.5.2 3 编辑套件

确保目前是选择 ATK-I.MX6U 套件 Debug 模式构建。

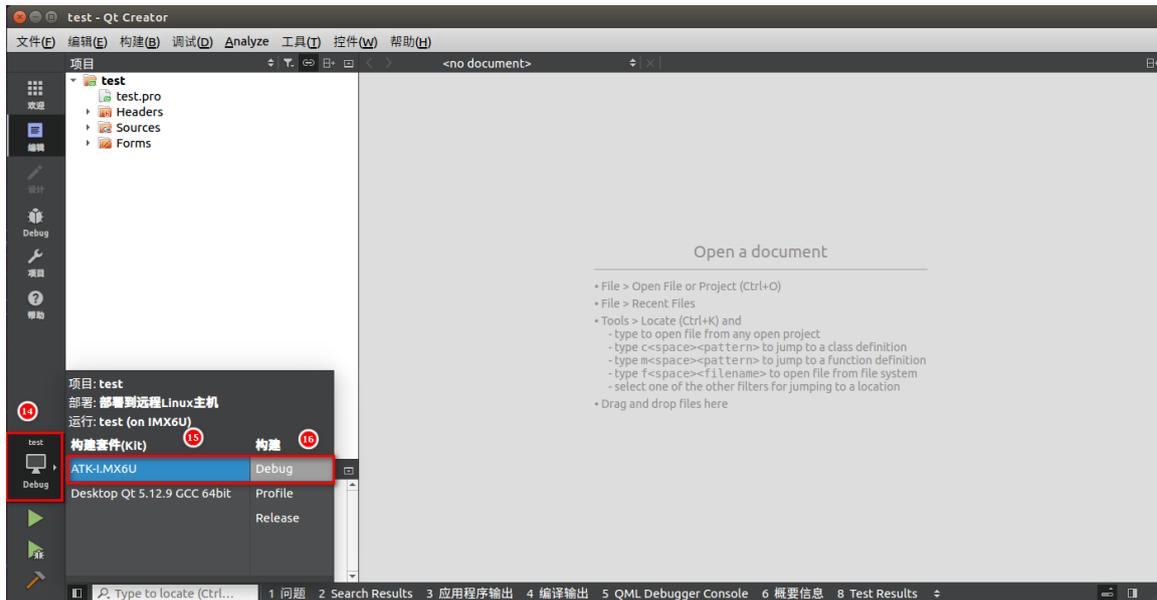


图 2.5.2 4 确保是 ATK-I.MX6U 套件

点击项目处, 选择 ATK-I.MX6U 设备, 选择 Run。删除通过 SFTP 上传文件, 因为我们出厂文件系统没有 sftp, 不能通过 sftp 上传。

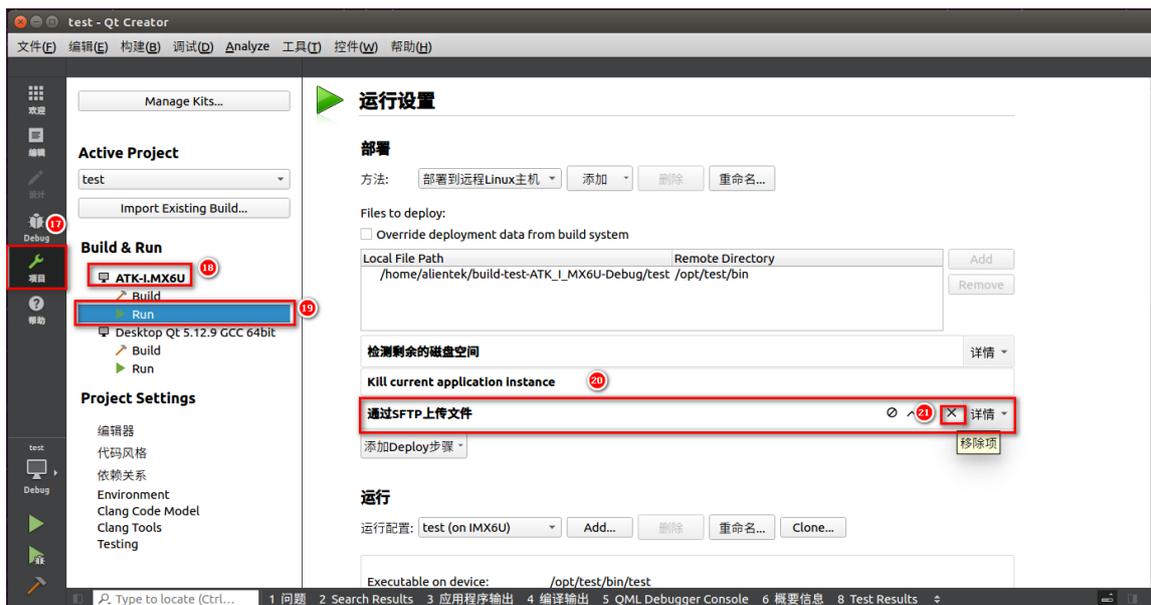


图 2.5.2 5 删除 SFTP 上传文件

添加 Deploy 步骤，选择“Custom Process Step”。

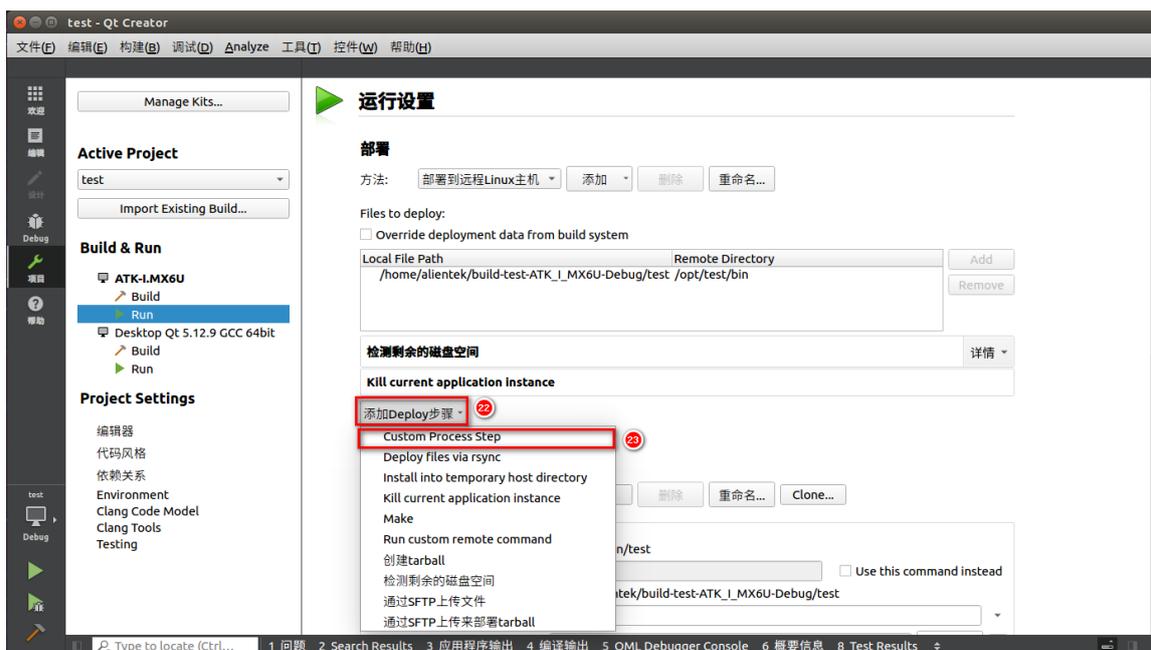


图 2.5.2 6 添加“Custom Process Step”

添加 Command 为 ssh, Arguments: 为 -p % {Device:SshPort} % {Device:UserName} @ % {Device:HostAddress} 'mkdir -p % {CurrentRun:Executable:Path}'。Working directory: 默认为 % {buildDir}。变量解释这里就不写了, 看名字也看的出来。

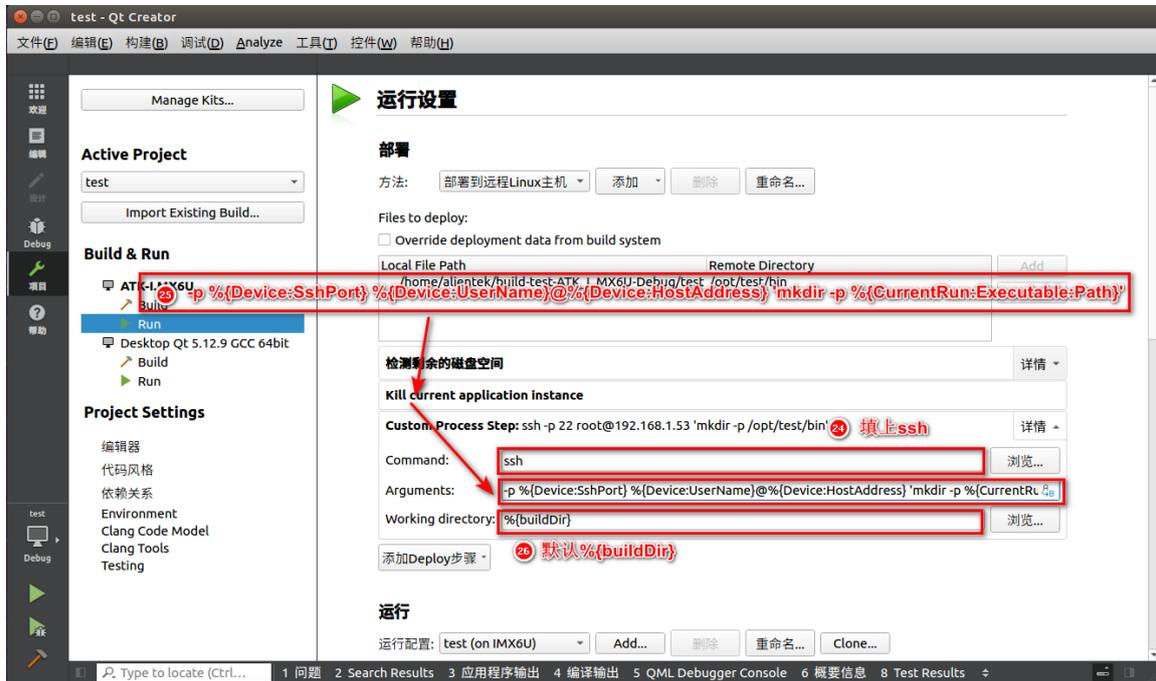


图 2.5.2 7 添加指令与变量

再添加一个 Custom Process Step

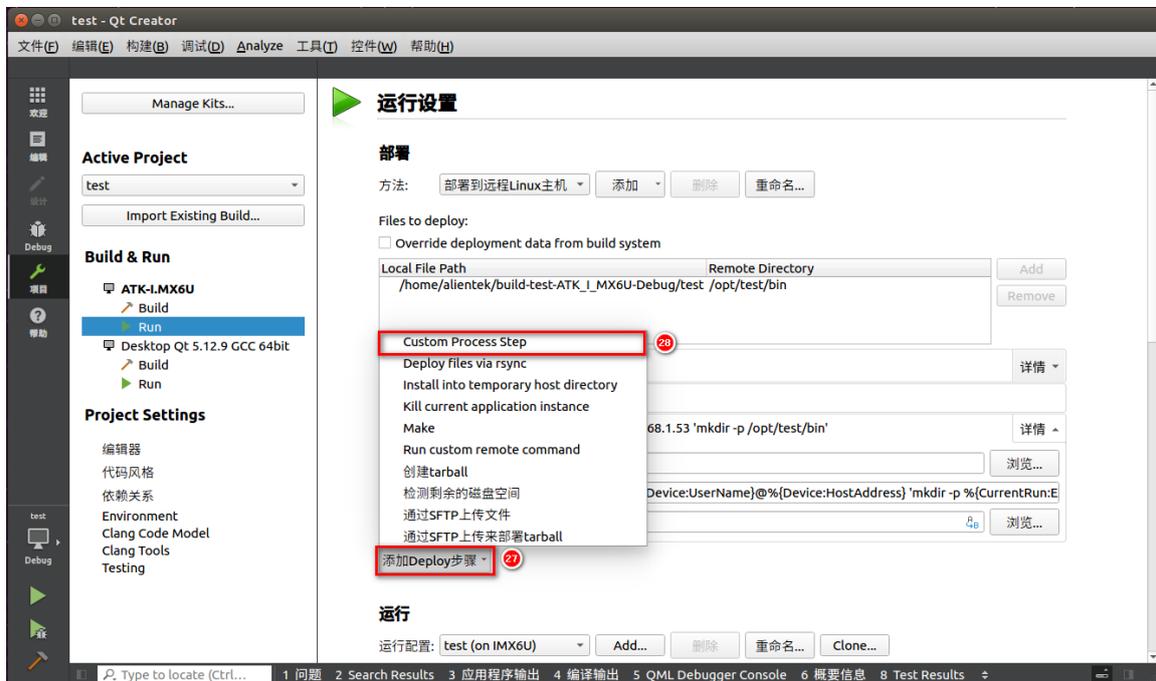


图 2.5.2 8 再添加一个“Custom Process Step”

添加 Command 为 scp, Arguments: 为 -P % {Device:SshPort} % {CurrentRun:Executable:FileName} % {Device:UserName} @ % {Device:HostAddress} : % {CurrentRun:Executable:FilePath}。 Working directory: 默认为 % {buildDir}。

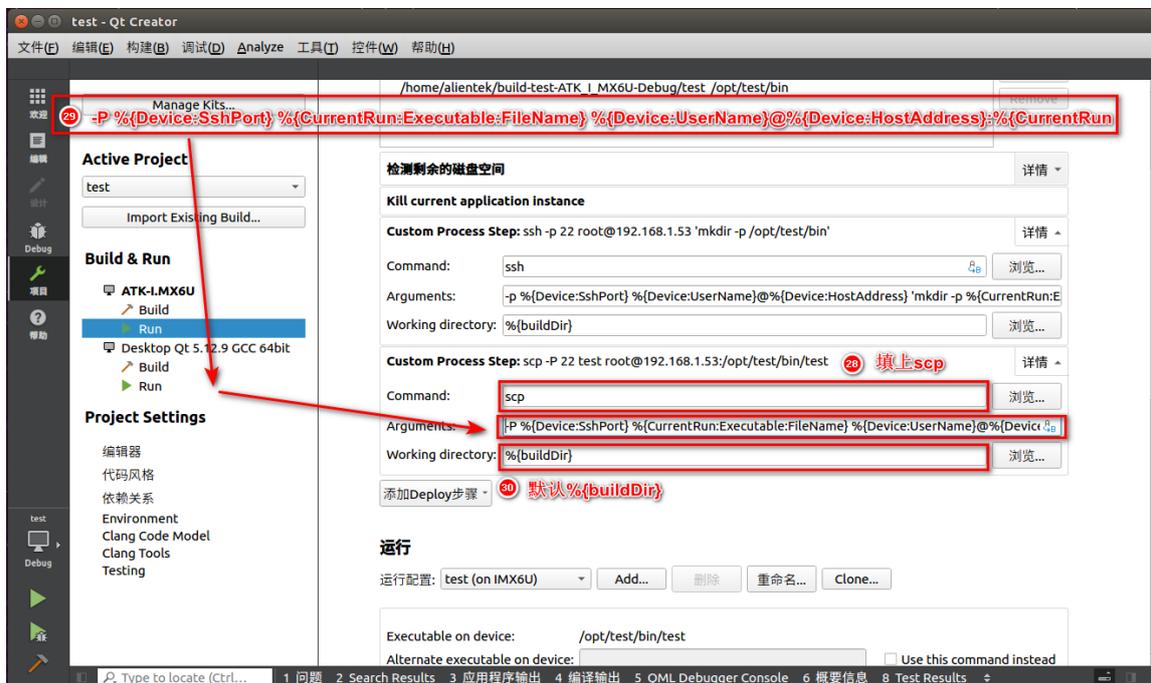


图 2.5.2 9 添加指令与变量

返回到编辑, 点击运行或按 F5 进入调试模式。下图为点击 32 步中的运行。可以看到已经把编译的程序发送到开发板上执行。开发板上的 LCD 屏也显示了一个空白的窗口。

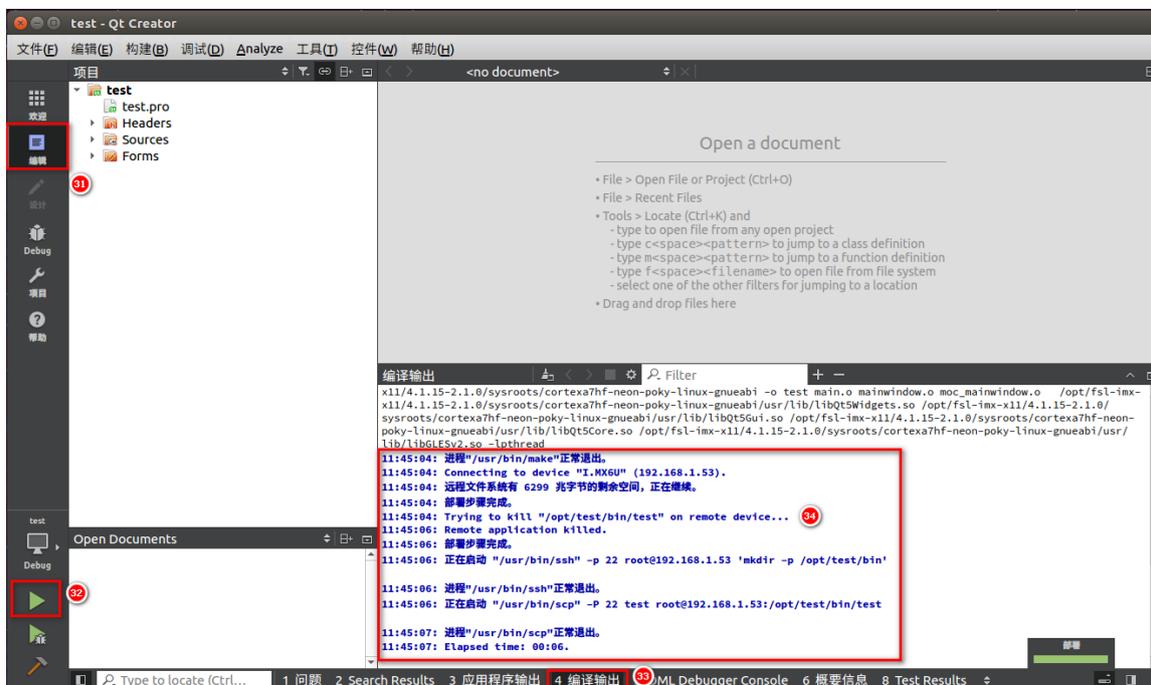


图 2.5.2 10 点击运行, 开发板上的 LCD 出现一个空白的窗口

## 第三章 Ubuntu 本机 Kits 编译 Qt 应用程序

### 3.1 Ubuntu 编译 Qt 应用程序

在 2.4 小节里我们在 test 工程里已经选择使用 ATK-I.MX6U Kit。那么别外一个就是 Qt Creator 安装时自带的 Kit，名字叫 Desktop Qt5.12.9 GCC 64bit。我们要在 Ubuntu 上运行，那么我们就选择此项，编译出的程序就可以在 Ubuntu 上运行啦！

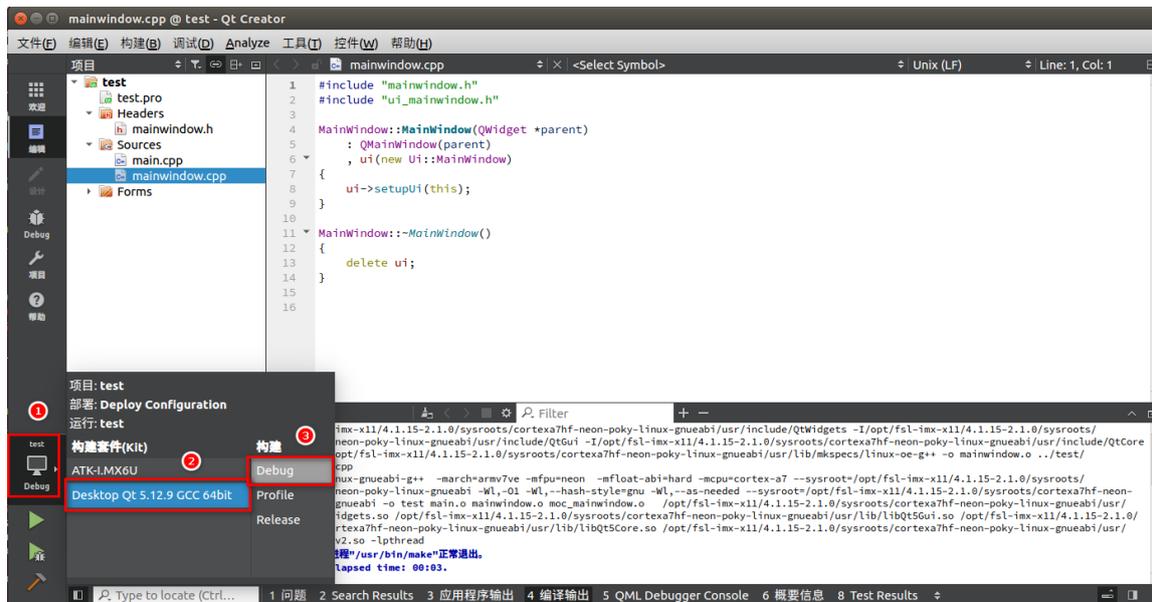


图 3.1 1 选择自带的桌面 Kit 编译

点击运行，可以看到程序执行时是显示一个空白的窗口。

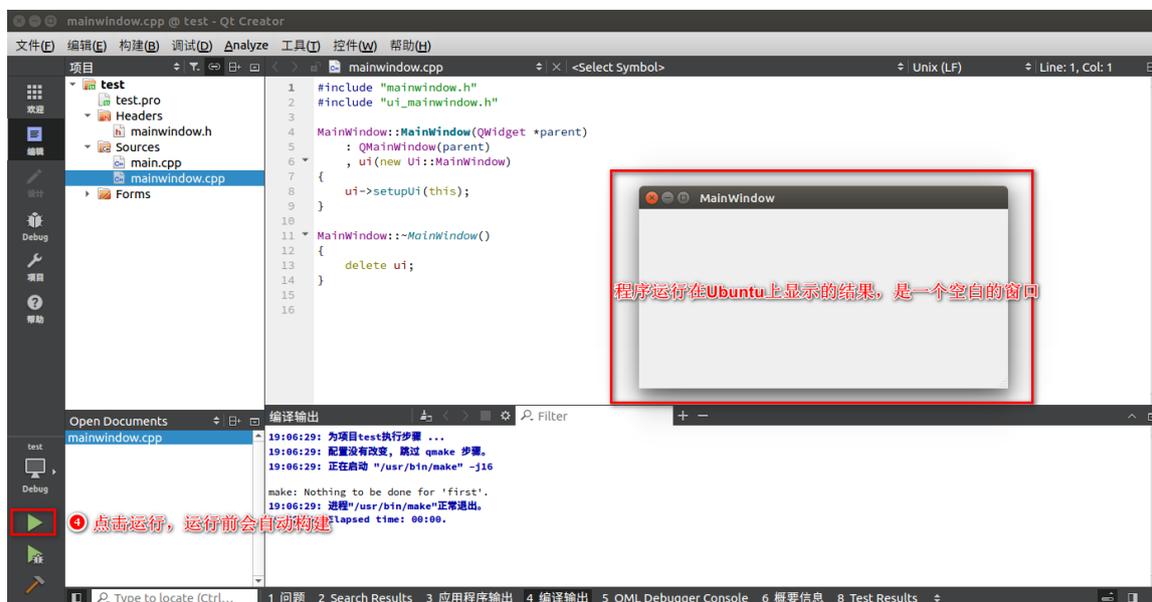


图 3.1 2 Ubuntu 上运行编译的 Qt 应用程序

至此正点原子 I.MX6U 出厂系统 Qt 的交叉编译环境，已经搭建完成，希望对大家有所帮助！

## 附录-A