



天启智能科技有限公司

T-CHIP INTELLIGENT TECHNOLOGY CO., LTD

Firefly 技术案例

Qt 环境部署

Ubuntu 篇

2025/8/20



一、什么是 Qt?

Qt 是一个跨平台 C++ 图形用户界面应用程序开发框架, 可构建高性能的桌面、移动及 Web 应用程序, 也可用于开发非 GUI 程序, 比如控制台工具和服务器。Qt 是面向对象的框架, 使用特殊的代码生成扩展以及一些宏, Qt 很容易扩展, 并且允许真正的组件编程。Qt 因其模块化程度高, 可重用性好, 可以快速上手等特点, 成为了一款非常受欢迎的开发框架。

二、Ubuntu18/20 部署

Firefly 针对 Firefly 产品上运行的 Ubuntu18/20 系统提供了对应的 Qt 环境包, 可以快速部署、开发。两个系统的环境包不通用, 请根据你的 Firefly 设备所运行的系统版本, 选择对应的环境包。

环境包[请点击此处下载](#), 提取码: FFQT

环境包采用交叉编译的模式, 在一个平台上生成另一个平台上的可执行代码。我们需要在 x86_64 架构的 Linux 电脑上进行开发和编译, 并将编译生成的可执行文件放到 Arm64 架构的目标设备上运行, 所以环境包需要往主机端和设备端都部署一遍。

注意: 不同版本的环境包有些许差别, 后面的部署和环境准备部分只能取一种进行讲解, 具体使用时请查看环境包内自带的文档。

1、主机端部署

关于主机端的选择: 请使用和目标设备有相同系统的主机进行开发。

例如: 需要给运行 Ubuntu20 的 Firefly 设备开发 Qt 程序, 那么主机也应该选择系统为 Ubuntu20 的电脑。

1.1 开始部署

解压环境包到电脑后, cd 到解压出来的目录中, 开始进行部署。主要分为 3 步:

```
# 将文件复制到系统 /opt/ 目录下
cp -rfd firefly_qt5.15_arm64_20.04 /opt/

# 将 sysroot 也复制到 /opt/ 目录下
cp -rfd sysroot /opt/

# 复制环境变量文件到 /etc/profile.d/ 下
cp host/host_qtEnv.sh /etc/profile.d/
```



1.2 环境准备

安装交叉编译器。

```
sudo apt update
sudo apt install -y crossbuild-essential-arm64
```

然后重新登录或者重启电脑，以确保之前复制的环境变量能生效。

1.3 验证

环境压缩包中还附带一个 example 文件夹，里面包含几个简单的 Qtdemo，可以用于验证环境是否搭建成功。一切正常则说明环境部署成功。

```
# 例如使用 analogclock 来验证
cd example/analogclock

# 执行 qmake 自动生成编译系统
qmake

# 执行 make 开始编译
make
```

2、设备端部署

2.1 开始部署

解压环境包到设备后，cd 到解压出来的目录中，开始进行部署。主要分为 2 步：

```
# 将 ext 部分复制到系统的 /opt/ 目录下，注意此处顺便是将 ext 重命名成了 firefly_qt5.15
cp -rfd firefly_qt5.15_arm64_20.04/ext /opt/firefly_qt5.15

# 复制环境变量文件到 /etc/profile.d/ 下
cp target/target_qtEnv.sh /etc/profile.d/
```

2.2 环境准备

设备上还需要安装一些必要的库：

```
apt install -y libmtdev1 libinput10 libxkbcommon0 \
    libdouble-conversion3 libicu66 libharfbuzz0b \
    libwebpdemux2 libwebpmux3
```

打开/etc/profile.d/target_qtEnv.sh 可以看到其中有三部分注释掉的环境变量：

```
#XCB
#export QT_QPA_PLATFORM=XCB
#export QT_QPA_EGLFS_INTEGRATION=XCB_EGL

#LinuxFB
#export QT_QPA_PLATFORM=LINUXFB
#export QT_QPA_FB_DRM=1
```



```
#EGLFS
#export QT_QPA_PLATFORM=EGLFS
#export QT_QPA_EGLFS_INTEGRATION=EGLFS_KMS #in minimal ubuntu, use eglfs_kms.
#export QT_QPA_EGLFS_KMS_ATOMIC=1 #use drm atomic api or legacy api(two different)
#export QT_QPA_EGLFS_HIDECURSOR=1 #hide cursor
```

根据你想使用的 QPA_PLATFORM 来取消对应部分的注释。比如 Ubuntu 环境一般使用 XCB，取消这部分的注释，然后保存并退出编辑。

```
#XCB
export QT_QPA_PLATFORM=XCB
export QT_QPA_EGLFS_INTEGRATION=XCB_EGL
```

如果使用 XCB 的话，还需要安装一些额外的库：

```
apt install -y libxcb-icccm4 libxcb-image0 libxcb-shm0 \
    libxcb-keysyms1 libxcb-render0 libxcb-render-util0 \
    libxcb-shape0 libxcb-sync1 libxcb-xfixes0 \
    libxcb-xinerama0 libxcb-xkb1 libxcb-randr0 \
    libsm6 libice6 libxkbcommon-x11-0
```

最后执行 reboot 重启设备。

2.3 验证

将之前在主机端编译好的可执行文件，转移到设备端运行；如果运行正常，则说明设备端环境搭建成功。

```
LD_LIBRARY_PATH=/opt/firefly_qt5.15/lib ./analogclock
```



三、Ubuntu22 部署

如果目标 Firefly 设备运行的是 Ubuntu22 系统，则不需要交叉编译开发，安装 Qt 更加简单方便。

Firefly 针对 Ubuntu22 将 Qt 环境做成了 deb 包，在目标设备上使用 apt 即可安装：

```
# 安装基础环境
apt update
apt install -y qtcreator qtbase5-dev
# 根据开发需要，安装额外的 Qt 组件与开发环境，例如
apt install -y libqt5multimedia5 qtmultimedia5-dev libqt5quick5 qtdeclarative5-dev
```

四、QtCreator 的配置

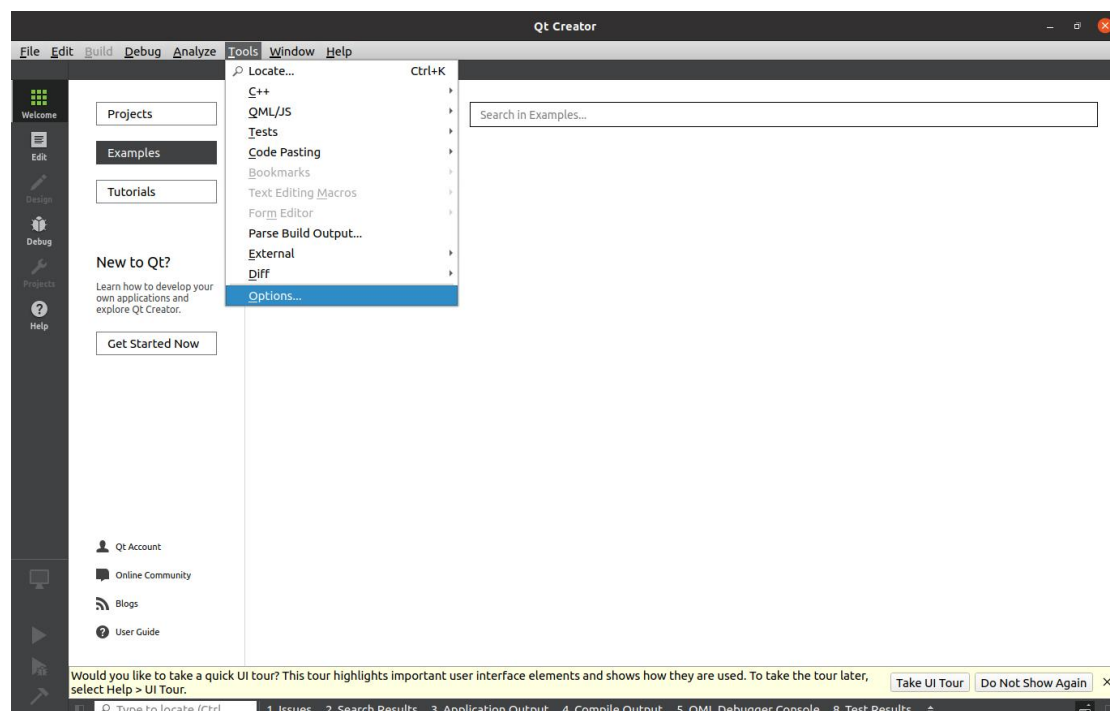
Qt 开发常使用 QtCreator 作为 IDE，前文中我们搭建了 Qt 开发环境，那么接下来将介绍如何确保 QtCreator 能使用我们建好的环境，而不是 IDE 自带的或者其他位置的环境。

1、Ubuntu18/20

在主机端可以使用如下命令快速安装 QtCreator：

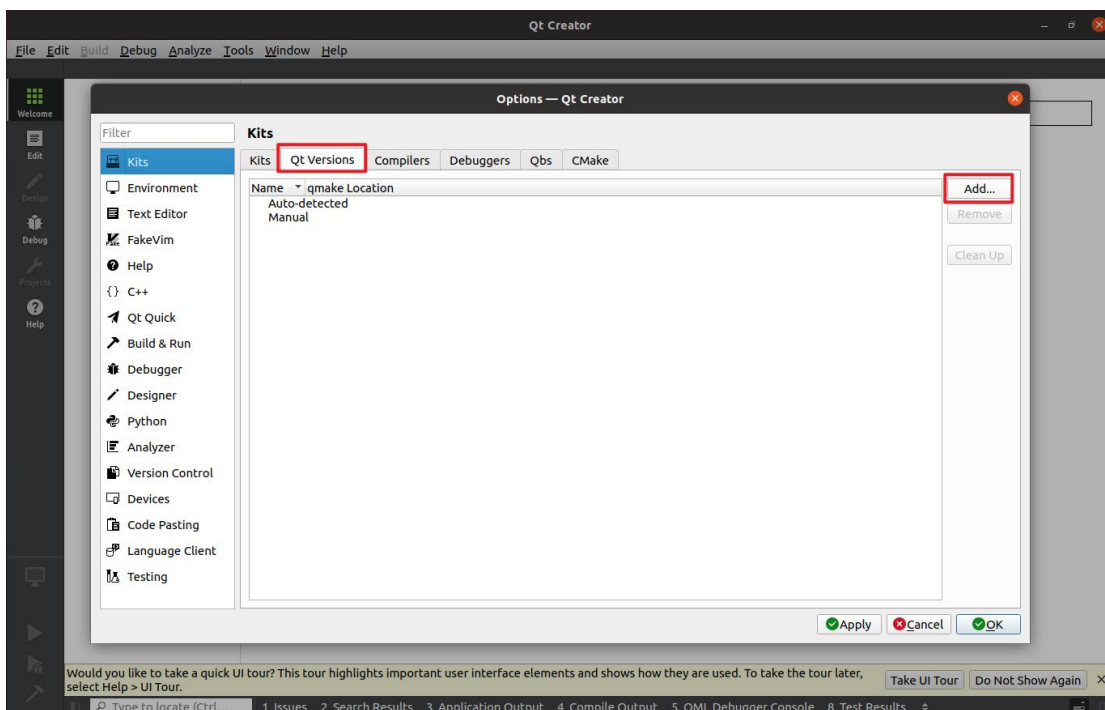
```
sudo apt update
sudo apt install qtcreator
```

启动 QtCreator 后，首先点击 tools->options 打开设置界面：

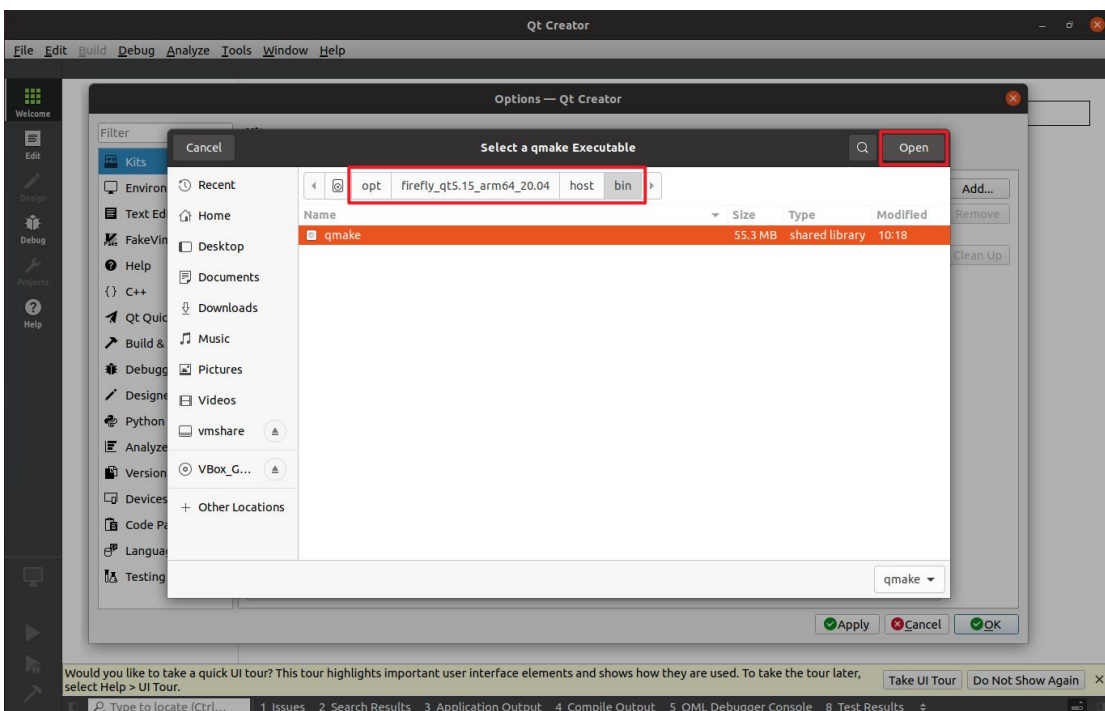




然后选择 QtVersions，点击 add 添加 Qt 版本：



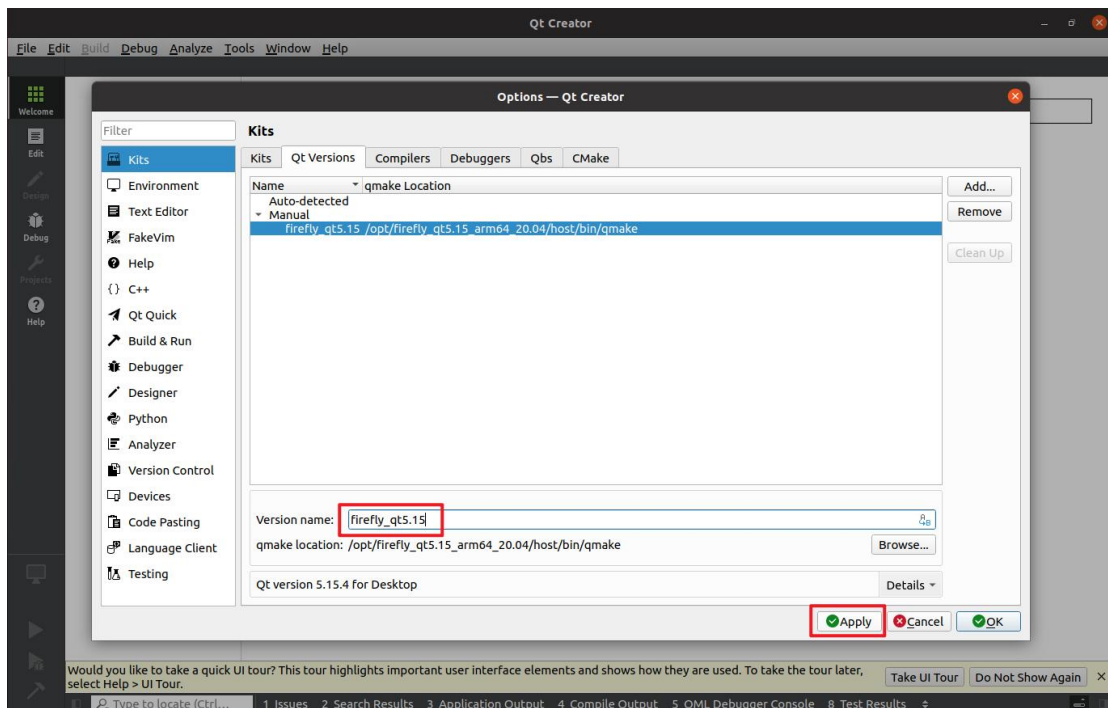
在弹出的窗口中选择我们之前复制到/opt 中的 Qt 环境：



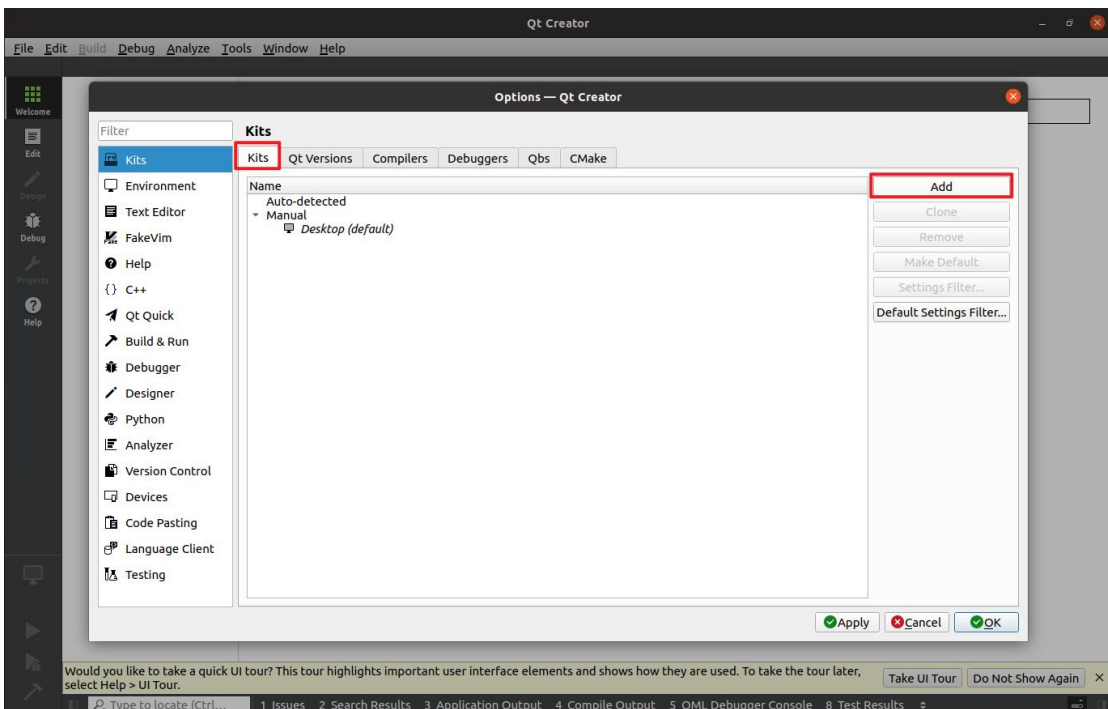
注意: 此处的路径, 根据环境包的版本不同, 会有一些差异, 但最终都是要选择/opt 下 qmake 的位置



可以给新添加的 QtVersion 起个名字，然后点击 apply 使添加生效：

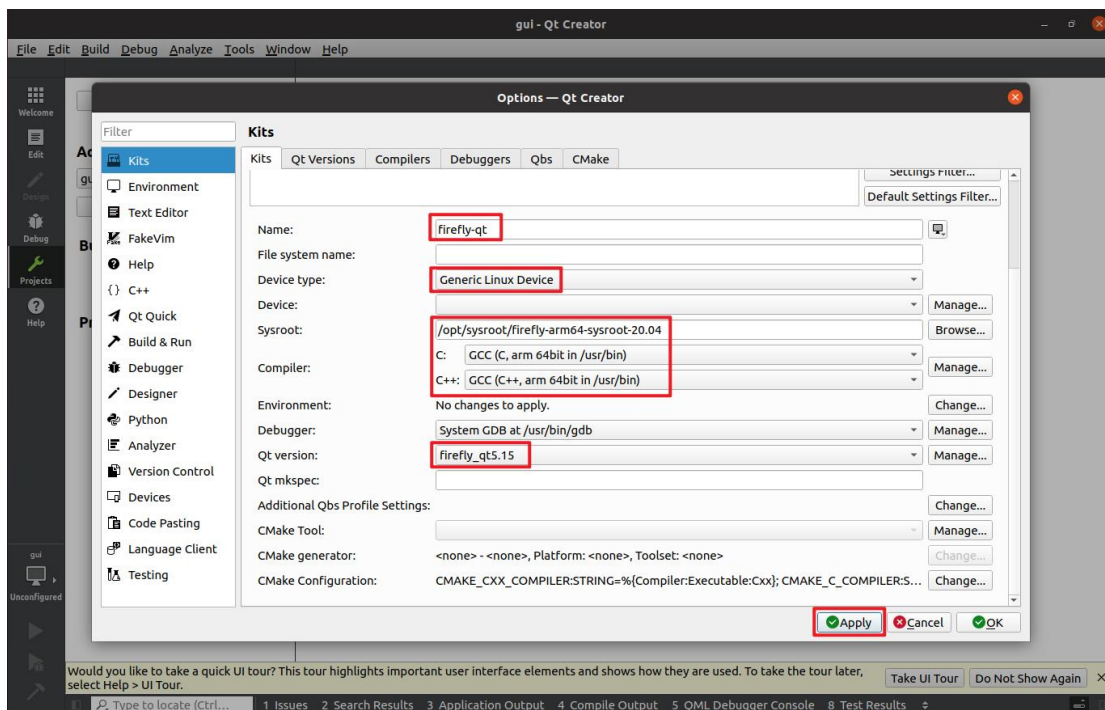


之后选择 Kits 页面，添加新的 Kit：



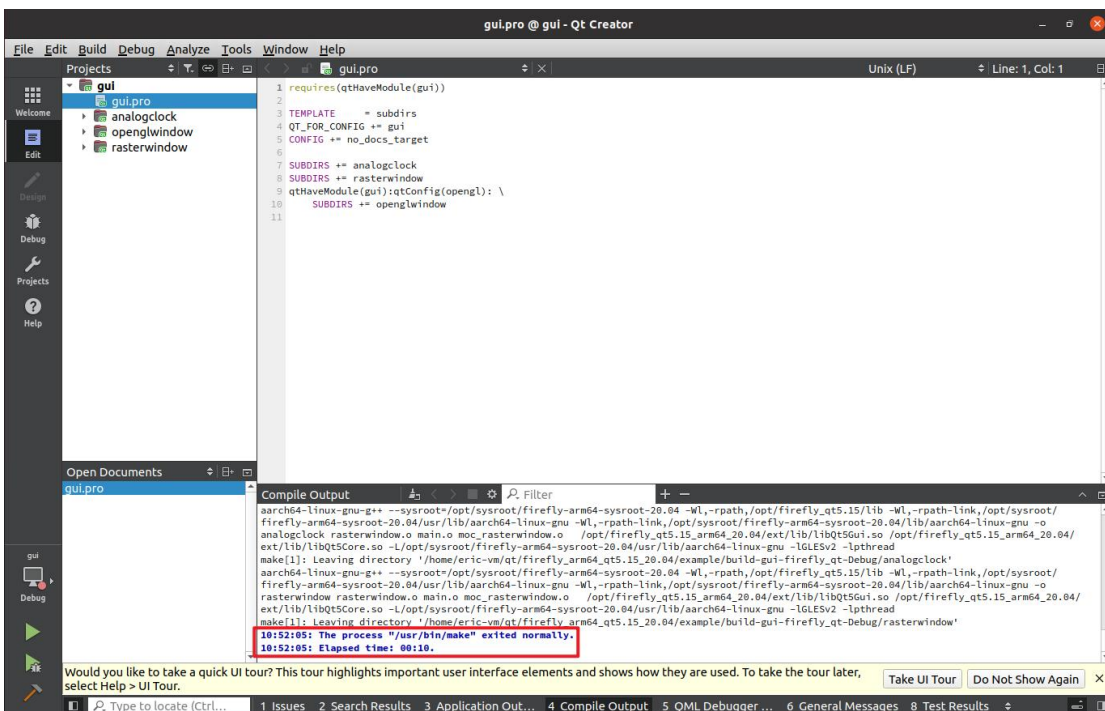


配置新的 Kit 如下：



配置的意义：首先给 Kit 起个名字，方便区分，DeviceType 需要选择 GenericLinuxDevice。重点是后面三个配置，Sysroot 需要选择我们之前复制到/opt 下的 sysroot；编译器要选择 arm64bit 的 GCC；QtVersion 选择我们上一步添加的 QtVersion，然后点击 apply 应用更改。

最后可以打开一个项目，选择我们新增的 Kit 进行编译，编译成功则说明我们配置正确：

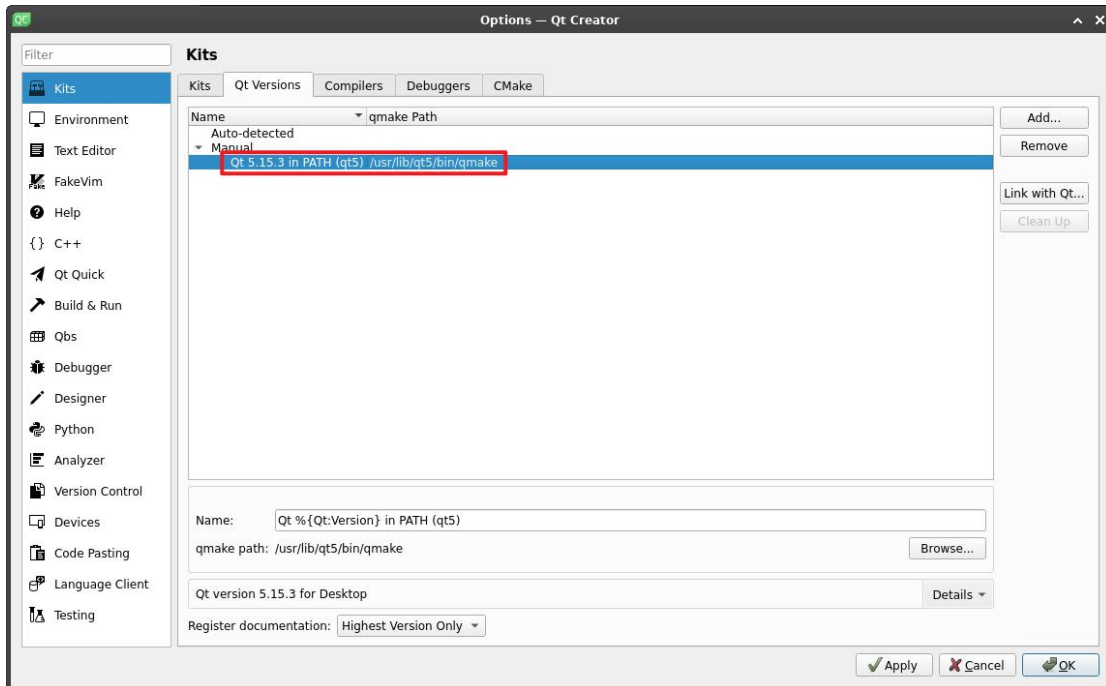




2、Ubuntu22

目标设备是 Ubuntu22 系统，不需要交叉编译，所以直接在设备端安装 QtCreator。

因为是通过 apt 安装的 Qt 环境,所以打开 QtCreator 它应该可以自动探测到 qmake 位置：



QtCreator 自动探测到的 qmake 路径为 /usr/lib/qt5/bin/qmake，无需再手动添加 QtVersion。

Kit 的配置中，因为不需要交叉编译，所以不设置 Sysroot，另外 QtVersion 选择自动探测到的 QtVersion，剩余其他配置和上一章的保持一致即可。



五、常见问题

1、解压失败

如果解压环境包的时候遇到这种错误：

```
tar: xxxxx: Directory renamed before its status could be extracted
tar: Exiting with failure status due to previous errors
```

可以给 tar 加上-P 参数重新解压，比如：

```
tar -Pzxf firefly_arm64_qt5.15_20.04_20230415.tgz
```

2、找不到 Qt 库

运行 Qt 程序时如果出现找不到 Qt 库的情况：

```
./analogclock: error while loading shared libraries: libQt5Gui.so.5: cannot open shared object file: No such file or directory
```

请检查环境变量文件`/etc/profile.d/target\_qtEnv.sh`是否存在？存在的话是否生效？Qt 库路径是否有按照文档要求放到指定位置？

3、多版本 Qt 冲突

运行 Qt 程序时如果出现报错说存在未定义的符号：

```
./analogclock: symbol lookup error: ./analogclock: undefined symbol: _ZN9QGradientD1Ev, version Qt_5
```

说明 Qt 程序找到了和自身版本不匹配的 Qt 库。这种问题常出现在有多个 Qt 版本环境的系统中，比如本来我们需要`/opt/firefly\_qt5.15`下的环境，但是 Qt 程序却尝试使用`/lib/aarch64-linux-gnu/libQt5Gui.so.5`，导致出现问题。

使用 ldd 命令可以查看程序尝试使用的库：

```
root@firefly:~/analogclock# ldd analogclock | grep -i qt
libQt5Gui.so.5 => /opt/firefly_qt5.15/lib/libQt5Gui.so.5 (0x0000007f8e0b2000)
libQt5Core.so.5 => /opt/firefly_qt5.15/lib/libQt5Core.so.5 (0x0000007f8daf6000)
```

解决方法一：在编译前就指定好程序优先查找库的路径，在项目的 pro 文件中添加：

```
QMAKE_RPATHDIR += /opt/firefly_qt5.15/lib
```

解决方法二：运行程序时指定 LD_LIBRARY_PATH：

```
LD_LIBRARY_PATH=/opt/firefly_qt5.15/lib ./analogclock
```



4、找不到显示

运行 Qt 程序时如果出现无法连接到显示:

```
qt.qpa.xcb: could not connect to display
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.

This application failed to start because no Qt platform plugin could be initialized.
Reinstalling the application may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc,
wayland-egl, wayland, wayland-xcomposite-egl, webgl, xcb.

Aborted
```

大概率是环境变量问题, 需要指定显示, 通常为:

```
# x11 环境使用:
export DISPLAY=:0
# wayland 环境使用:
export WAYLAND_DISPLAY=wayland-0
```

商务合作: sales@t-firefly.com
技术支持: service@t-firefly.com
Firefly 官网: www.t-firefly.com
开源社区: dev.t-firefly.com



欢迎扫码关注Firefly开源团队



公众号



视频号