MYD-YT113X_Qt 应用笔记

Qt 环境安装与 MEasy-HMI 编译

文件状态： [√] 草稿 [ ] 正式发布	**文件标识：**	MYIR-MYD-YT113X-SW-AN-ZH-L5.4.61
	当前版本：	V1.0
	作　　者：	Nico
	创建日期：	2023-06-08
	最近更新：	2023-08-08

版 本 历 史

版本	作者	参与者	日期	备注
V1.0	Nico	--	20230608	创建该文档

目 录

1.概述

QT 是一个跨平台的图形应用开发框架，被应用在不同尺寸设备和平台上，QT 同时提供不同版权版本供用户选择。在 Qt 应用开发中，推荐使用 QtCreator 集成开发环境，可以在 Linux PC 下开发 Qt 应用，自动化地交叉编译为开发板的 ARM 架构。

本章使用 MYIR 构建的 SDK 工具作为交叉编译工具，配合 QtCreator 快速开发图形类应用程序。

2.硬件资源

无

3.软件资源

- Ubuntu18.04 desktop 或 Ubuntu20.04 desktop（推荐）
- Qtcreator 4.12
- gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi.tar.xz
- MYIR-MEasy-HMI 2.X

4.环境准备

需要安装 ubuntu 桌面系统，由于本人的原始开发环境为 ubuntu18.04desktop, 后续均在 ubuntu18.04 desktop 版本下操作。桌面系统请您自行安装。

需要安装米尔电子提供的编译交叉工具链，路径：03-Tools\Complie Toolchain\gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi.tar.xz。

在米尔提供的开发 SDK 包里获取 mxapp2.X 源代码，路径 04-Sources/mxapp2.tar.gz 或者从 platform/framework/auto/qt_demo/MXAPP 获得。

制作 QT 编译链

在制作 QT 编译链之前，一定要完成《MYD-YT113X_Linux 软件开发指南 V1.1》文档中 3.4 章节编译 QT 步骤。

首先将 03-Tools\Complie Toolchain\gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi.tar.xz 编译链解压，本文在下面路径解压，用户请根据自己的实际情况解压。

```
sur@myir:~/opt$ tar -xf gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi.tar.xz
```

在编译完成 QT 之后，在下面目录会生成 Qt_5.12.5 目录

```
sur@myir:~/auto-t113x-linux/platform/framework/qt/qt-everywhere-src-5.12.5/Qt_5.12.5
```

将 Qt_5.12.5 这个目录复制到上面解压的编译链目录中

```
sur@myir:~/auto-t113x-linux/platform/framework/qt/qt-everywhere-src-5.12.5$ cp -r Qt_5.12.5/ ~/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/
```

然后在 gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/ Qt_5.12.5/目录下创建 qt.conf 文件，并添加下面的内容

```
sur@myir:~/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/Qt_5.12.5$ vim qt.conf

[Paths]
Documentation=../../Docs/Qt-5.12.2
Examples=../../Examples/Qt-5.12.2
Sysroot=../arm-buildroot-linux-gnueabi/sysroot
Prefix=..
```

最后一步

由于只有 T113-S3 full 镜像和 T113-I full 镜像可以安装 QT 环境

如果是 T113-S3 full 镜像，将下面路径的目录复制到前面解压的编译链目录中

```
sur@myir:~/auto-t113x-linux/out/t113/myir-image-yt113s3-emmc-full/longan/buildroot/host$
cp -r arm-buildroot-linux-gnueabi/ ~/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/Qt_5.12.5/
```

如果是 T113-I full 镜像，将下面路径的目录复制到前面解压的编译链目录中

```
sur@myir:~/auto-t113x-linux/out/t113_i/myir-image-yt113i-full/longan/buildroot/host$
cp -r arm-buildroot-linux-gnueabi/ ~/opt/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/Qt_5.12.5/
```

至此编译链制作完成，按照下面的步骤安装环境即可。

5.操作步骤

5.1.安装 Qt Creator

QT Creator 官网下载（以 4.12-rc1 版本为例）

官方下载地址：http://download.qt.io/development_releases/qtcreator/

QtCreator 安装包是一个 ubuntu 系统下的二进制程序，在 ubuntu 的终端下直接执行就可以完成安装 ./qt-creator-opensource-linux-x86_64-4.12.0-rc1.run

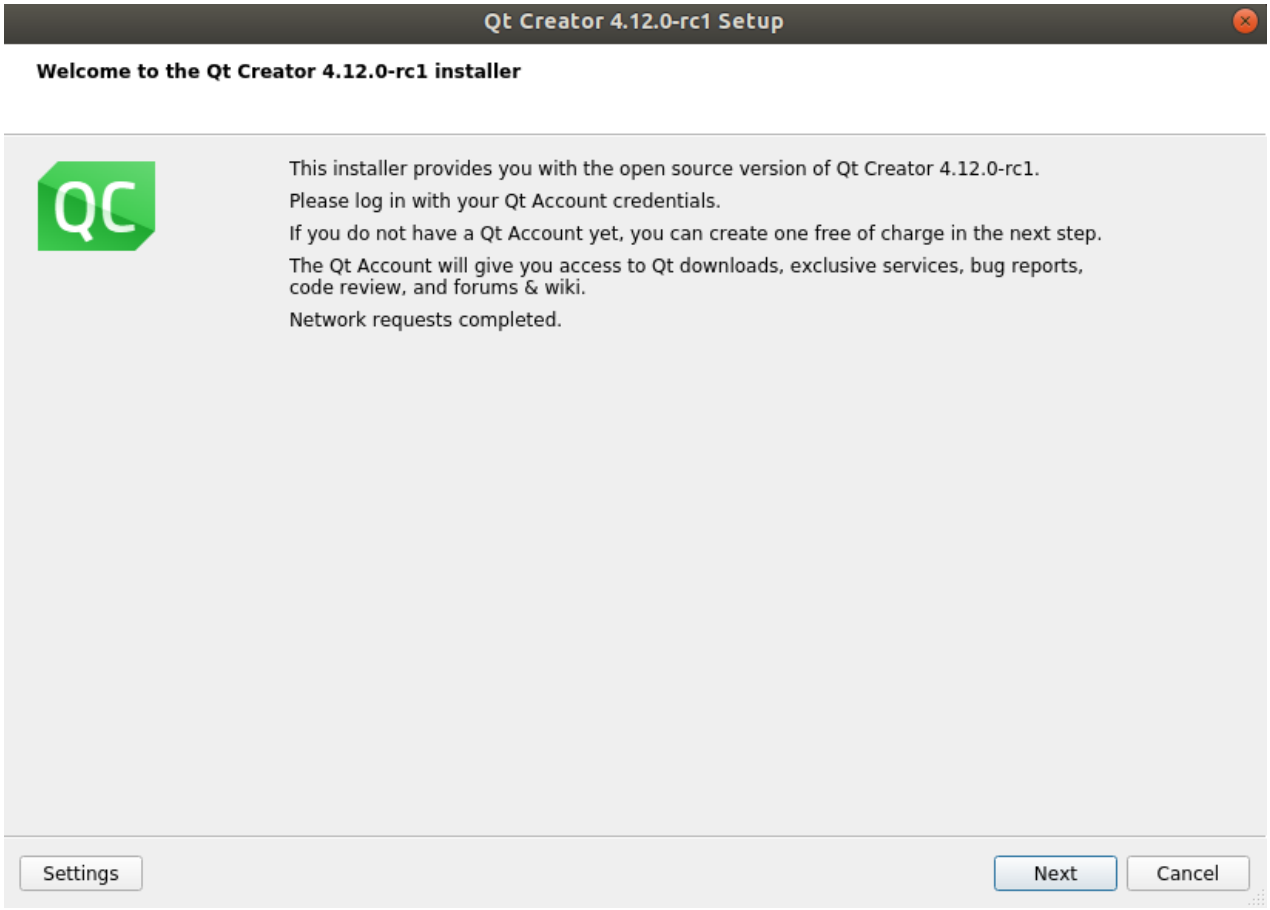

图 5-1. QTCreator 安装

点击 next, 输入账号密码，账号需要自行在官网注册 https://login.qt.io/register

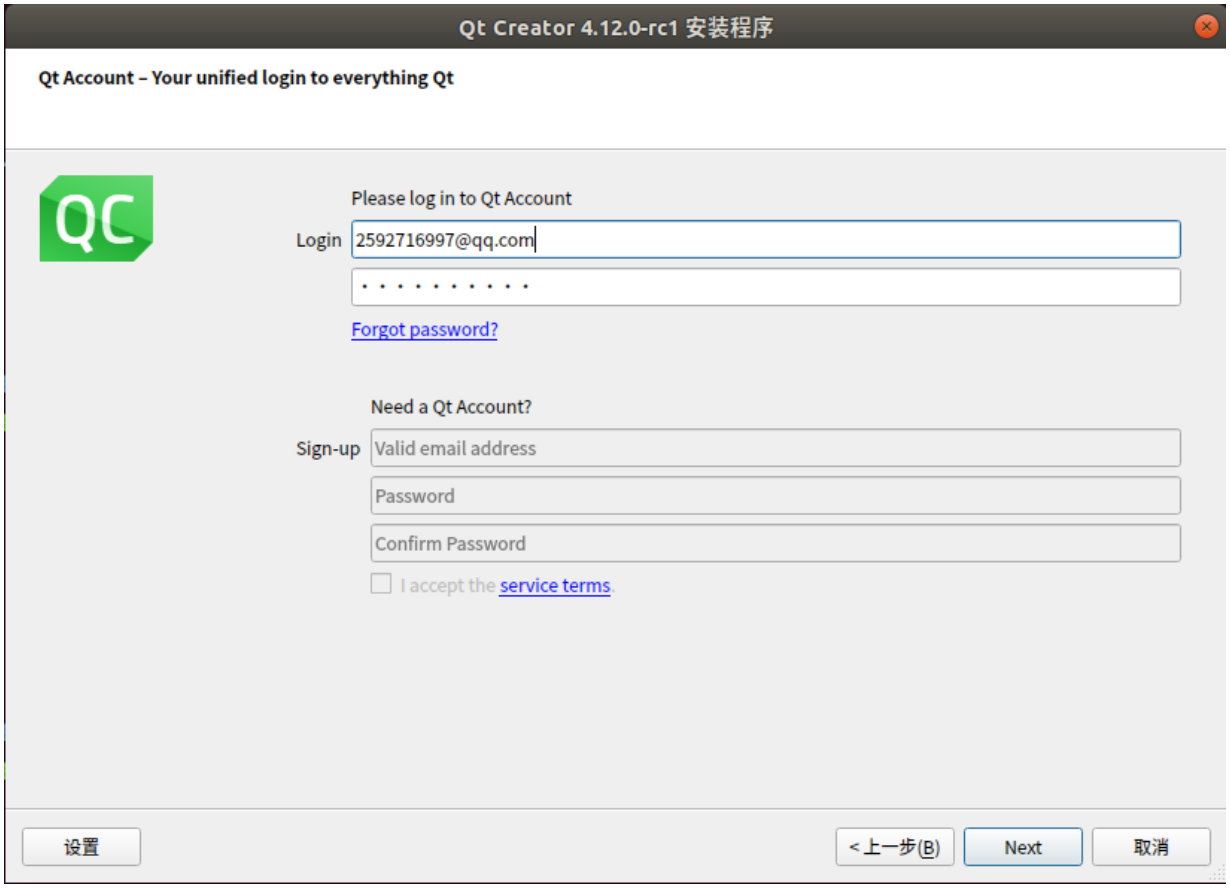

图 5-2. 登录账号密码

其他配置项请自行选择，配置安装路径

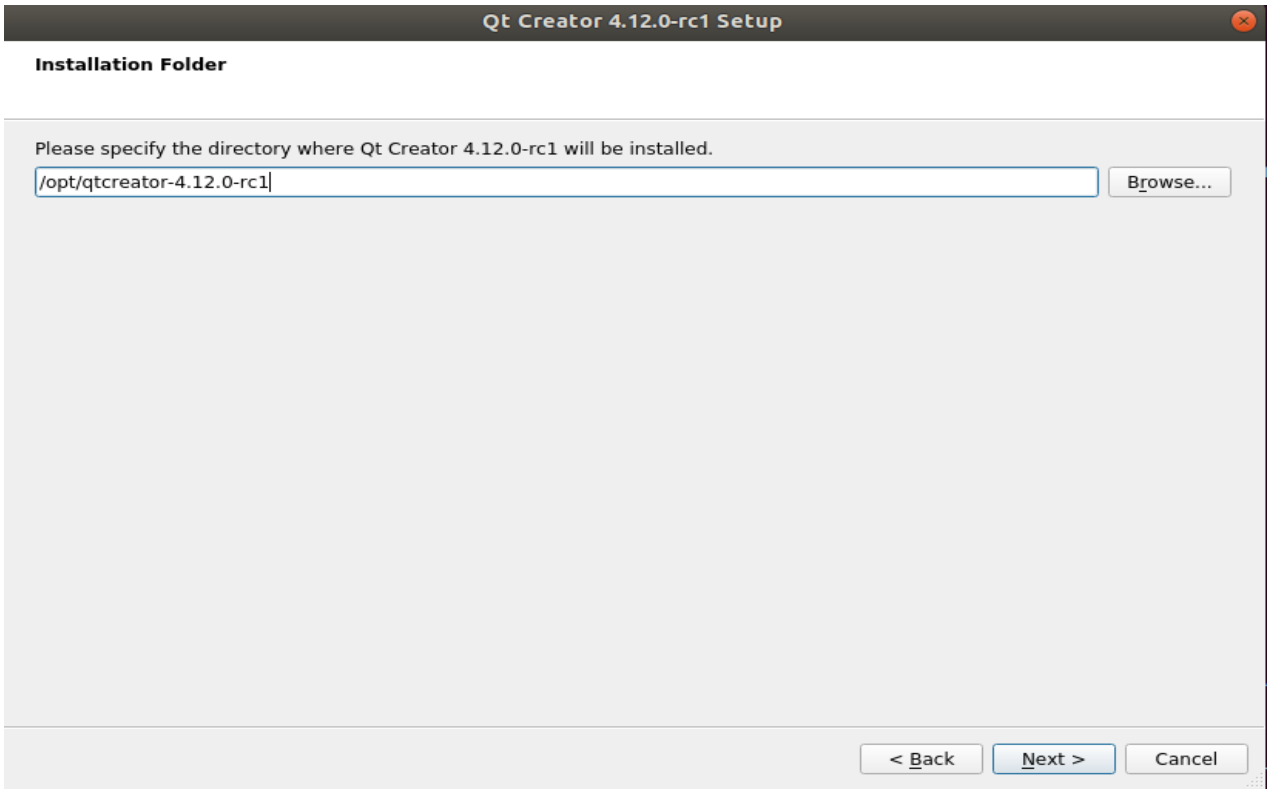

图 5-3. 选择安装路径

后续操作按提示安装完成即可

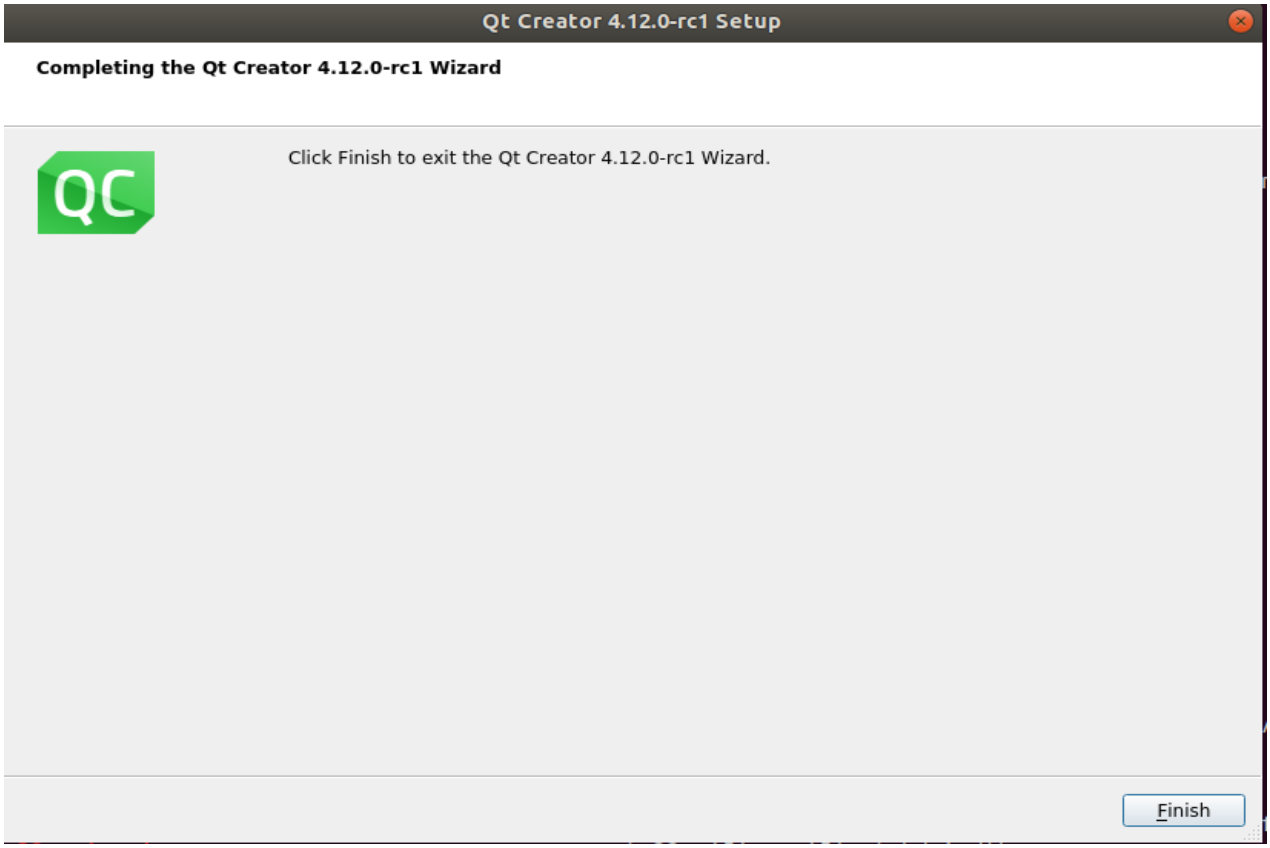

图 5-4. 安装完成

Qtcreator 安装完成，可进行开发环境的配置

5.2.配置交叉编译环境

1) 打开 QTCreator，请从终端执行"qtcreator.sh"来启动 QtCreator，参考如下：

/opt/qtcreator-4.12.0-rc1/bin/qtcreator.sh &

2) 运行 QtCreator 后,依次点击 Tools -> Options ,出现选项对话框,在左边点击 Kits ,右边选择 Compilers 标签.

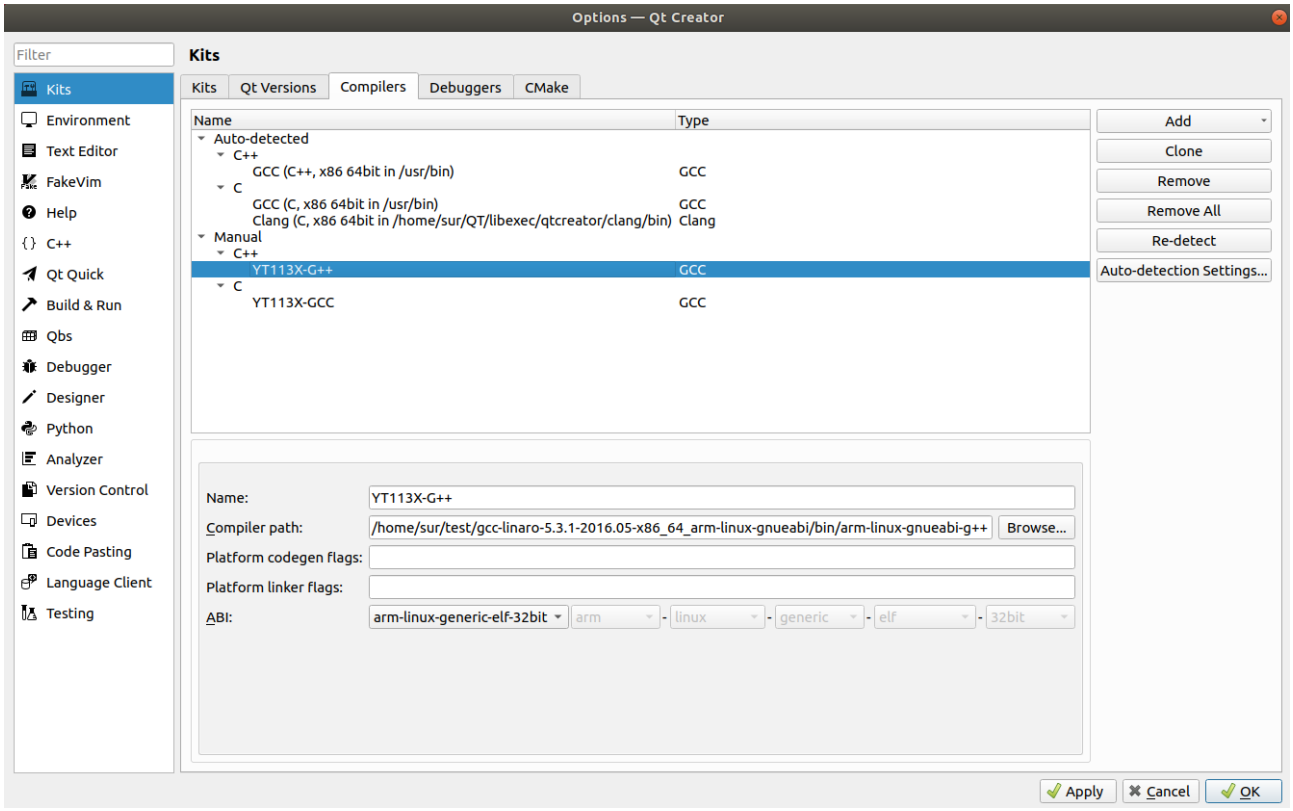

图 5-5. 选择编译器

3) 点击右侧 Add ,弹出下拉列表后,选择 G++ ,在下面填写"Name"为"YT113X-G++"，"Compiler path"点击旁边的"Browse.."按钮选 择到 arm-linux-gnueabi-g++的路径，例子中的路径是"/home/sur/test/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/bin"。

同样的选择 GCC，自行按照 G++的方式选择 arm-linux-gnueabi-gcc 编译器填写完成后，点击"Apply"。

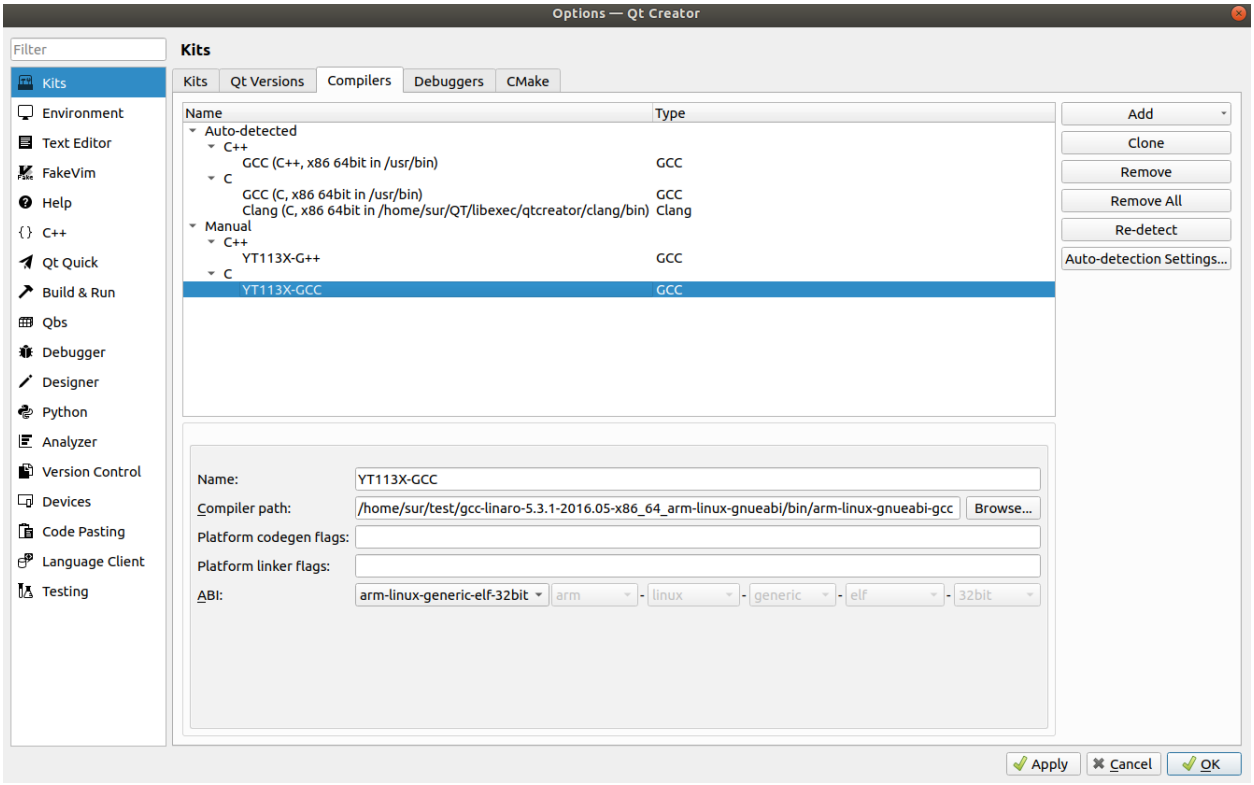

图 5-6. 选择编译器路径

4) 选择"Qt Version"标签，在右侧点击"Add..."，会弹出 qmake 路径选择对话框，这里以"/home/sur/test/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/Qt_5.12.5/bin/qmake"为例子。选择"qmake"文件后，点击"Open"按钮。

"Version name"改为 Qt %{Qt:Version} (MYIR-YT113X-System)。然后点击"Apply"按钮。

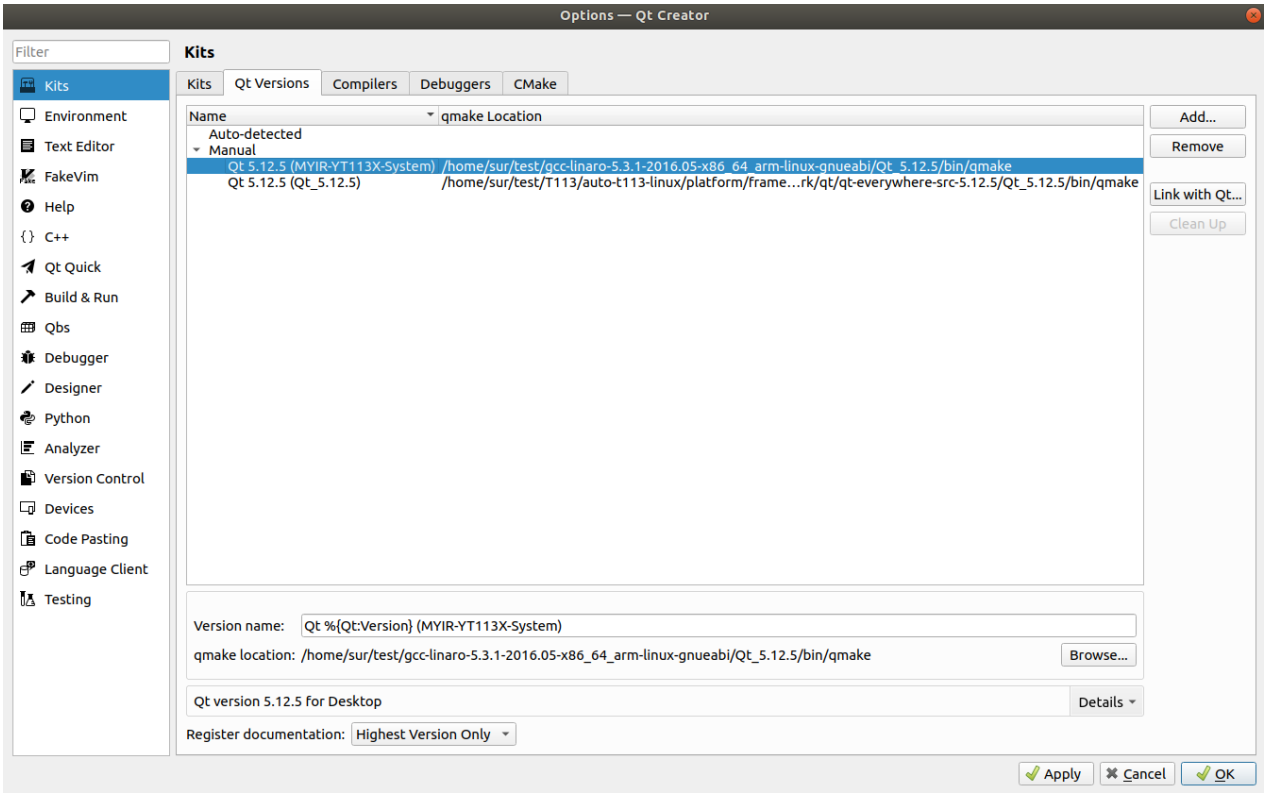

图 5-7. 配置版本信息

注意：如出现以下错误，原因是 GLIBC 版本需要大于 2.28，请更新 glibc 版本或者直接使用 Ubuntu 20 以上版本开发。

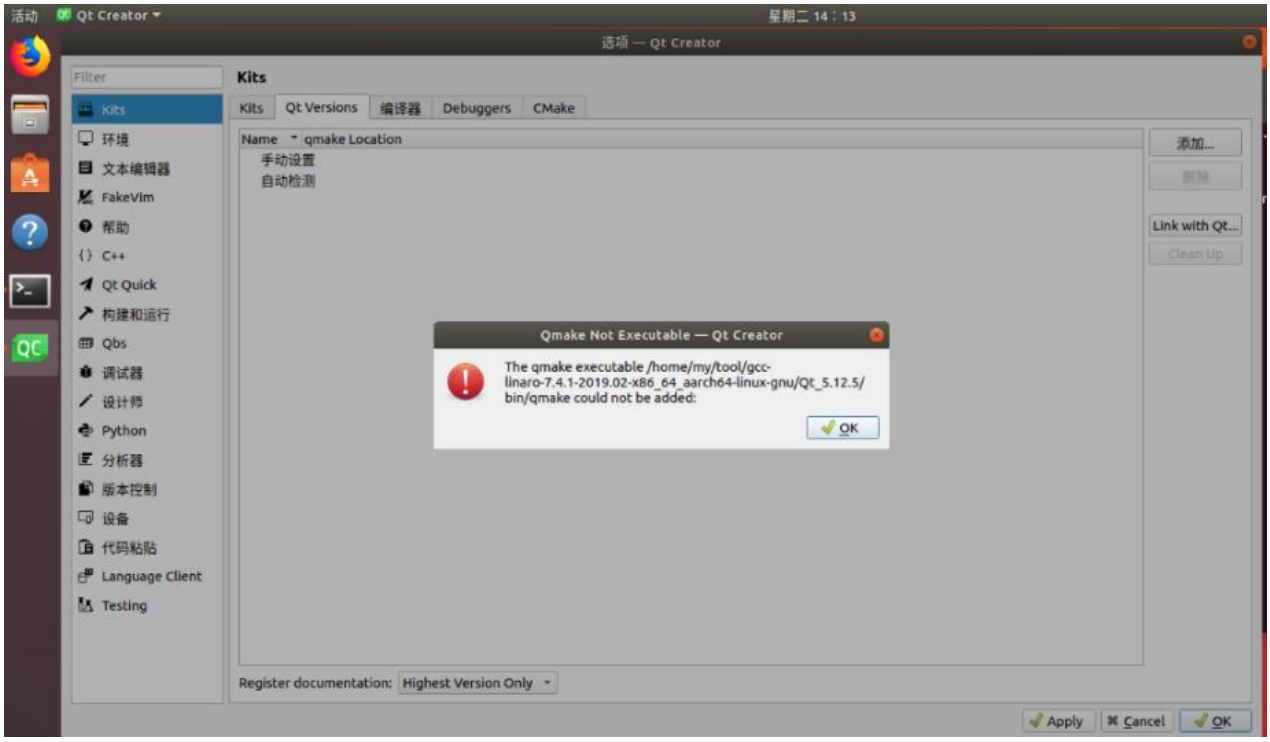

图 5-8. Qmake 配置错误

5) 选择左侧"Device"，点击右边的"Add..."按钮，填写内容"Name"为"MYD-YT113X-Board"，"Host name"为开发板的 IP 地址(可以暂时填写任意一个址)，"Username"为"root"，然后击"Next"进行下一步配置。(此项为可选项)

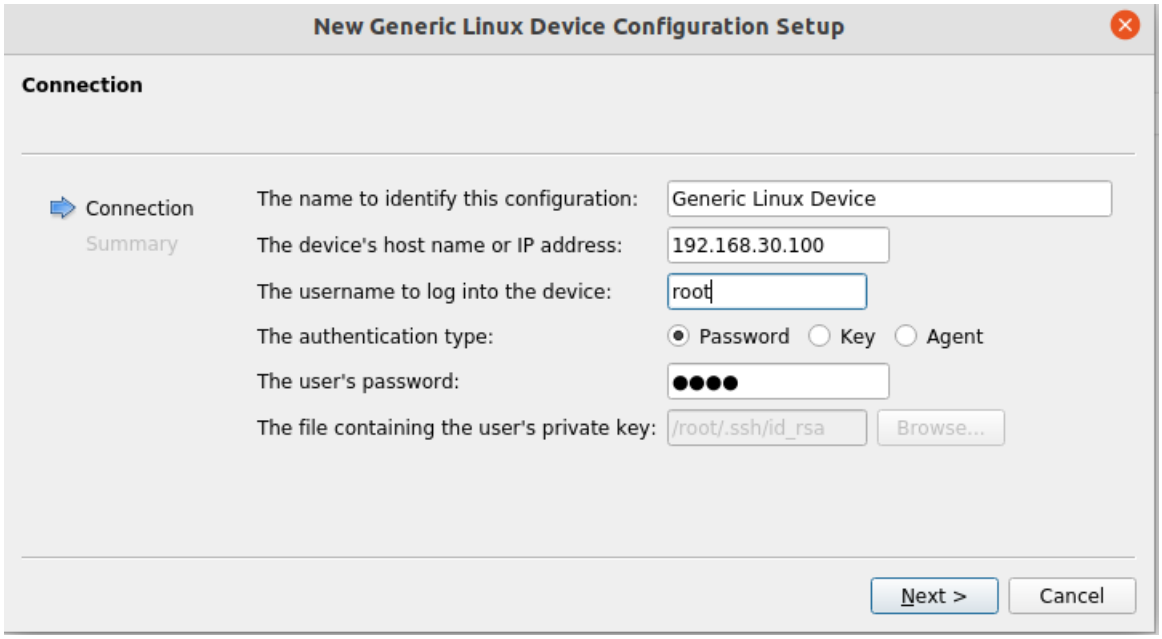

图 5-9. 配置设备信息

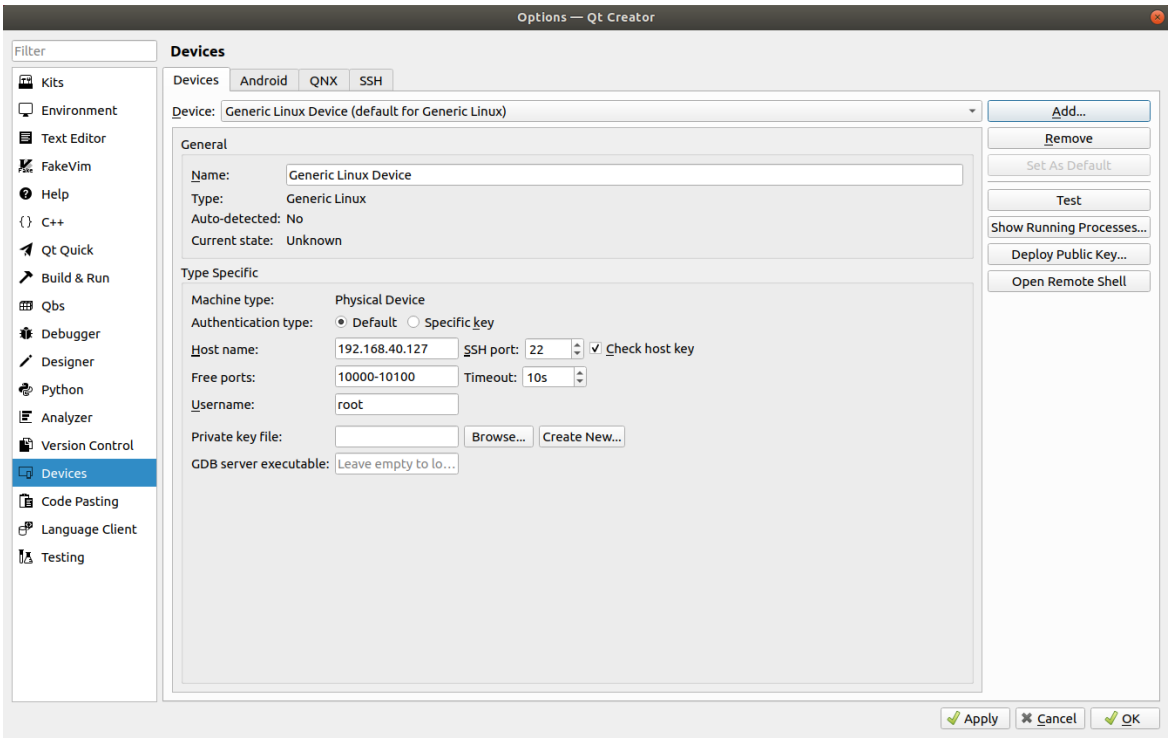

图 5-10. 完整的配置设备信息

6) 点击左侧"Build & Run"回到"Kits"标签下，"Name"设置为"YT113X-dev-kit"，"Device"选择"Generic Linux Device"选项了。"Sysroot"选择目标设备的系统目录，这里以"/home/sur/test/gcc-linaro-5.3.1-2016.05-x86_64_arm-linux-gnueabi/Qt_5.12.5/arm-buildroot-linux-gnueabi/sysroot"为例。"Compiler"选择之前配置的名称"YT113X-G++"，"Qt version"选择之前配置的名称"Qt 5.12.5 (MYIR-YT113X-System)"，"Qt mkspec"填写为"linux-arm-gnueabi-g++"（选填）。其它默认即可，最后点击"Apply"和"OK"按钮。

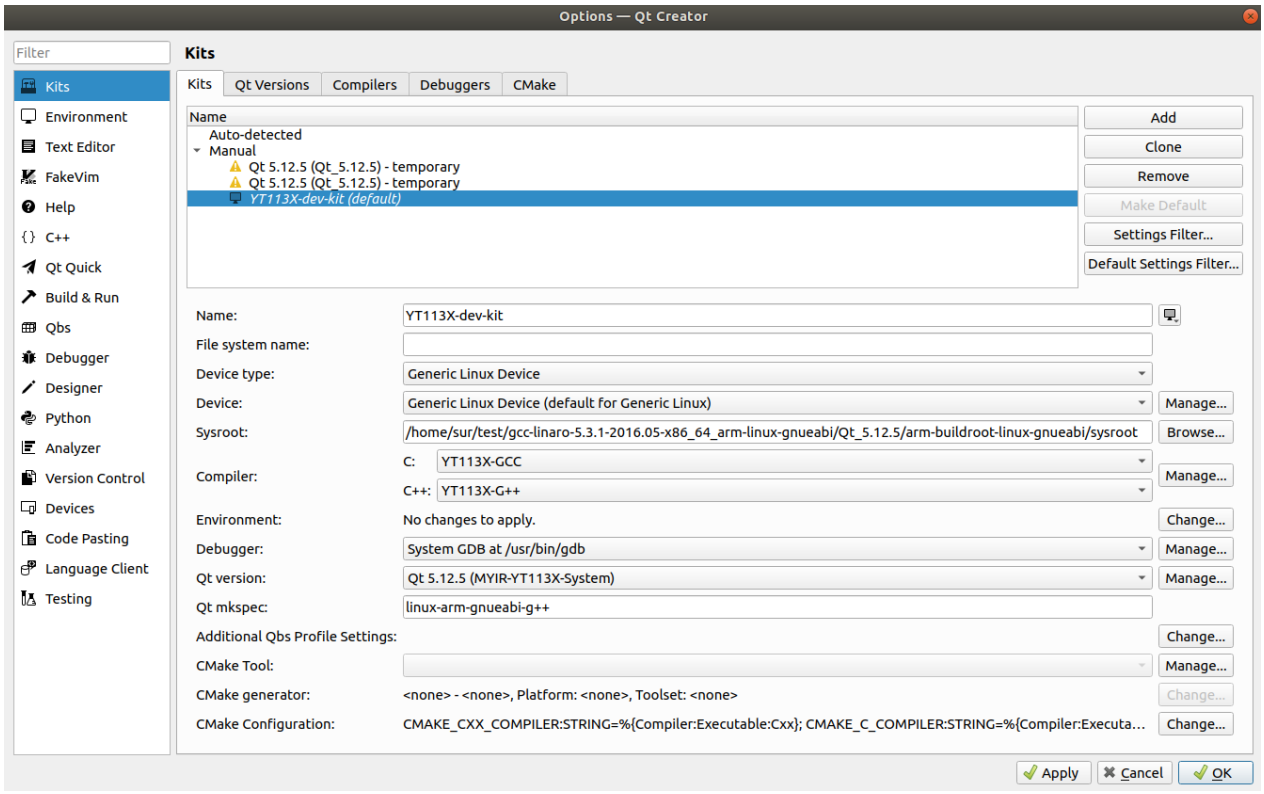

图 5-11. 配置开发板信息

至此 QT 的开发环境搭建完成，后续就可以进行 QT 应用程序的开发。

5.3.Measy HMI2.X 编译

将 mxapp2.tar.gz 拷贝到 Ubuntu 下的一个工作目录中并解压出源代码。配置为相应的编译工具套件，就可以编译此例程。

在菜单栏选择"File"->"Open File or Project"，在打开的对话框中，浏览到"MXAPP"例程的目录下，选择"mxapp2.pro"文件，点击"Open"按钮。

项目打开后，在左侧菜单列中，选择"Projects"图标，右侧界面切换为 manage kits 管理界面，在"Build & Run"标签下，选择"YT113X-dev-kit"选项的 kit，这样项目就会使用"YT113X -dev-kit"的相关配置 kit 构建应用。

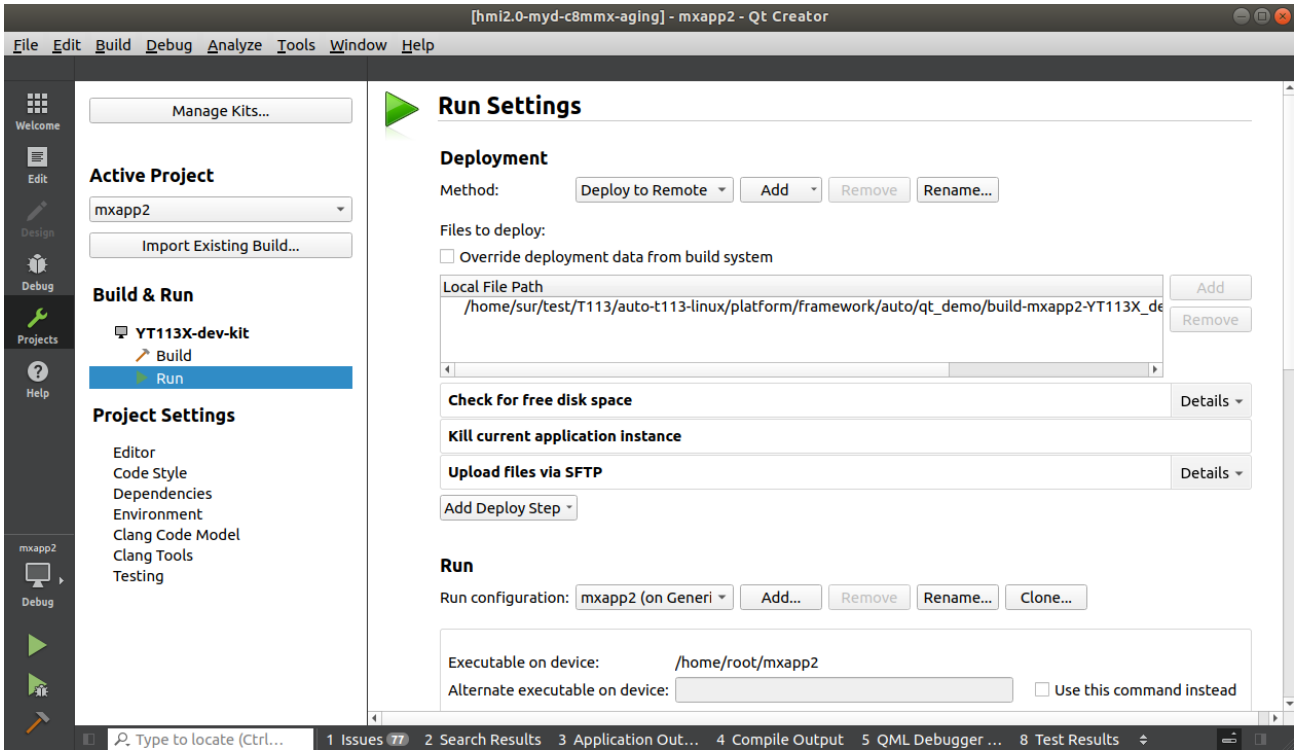

图 5-12.打开项目

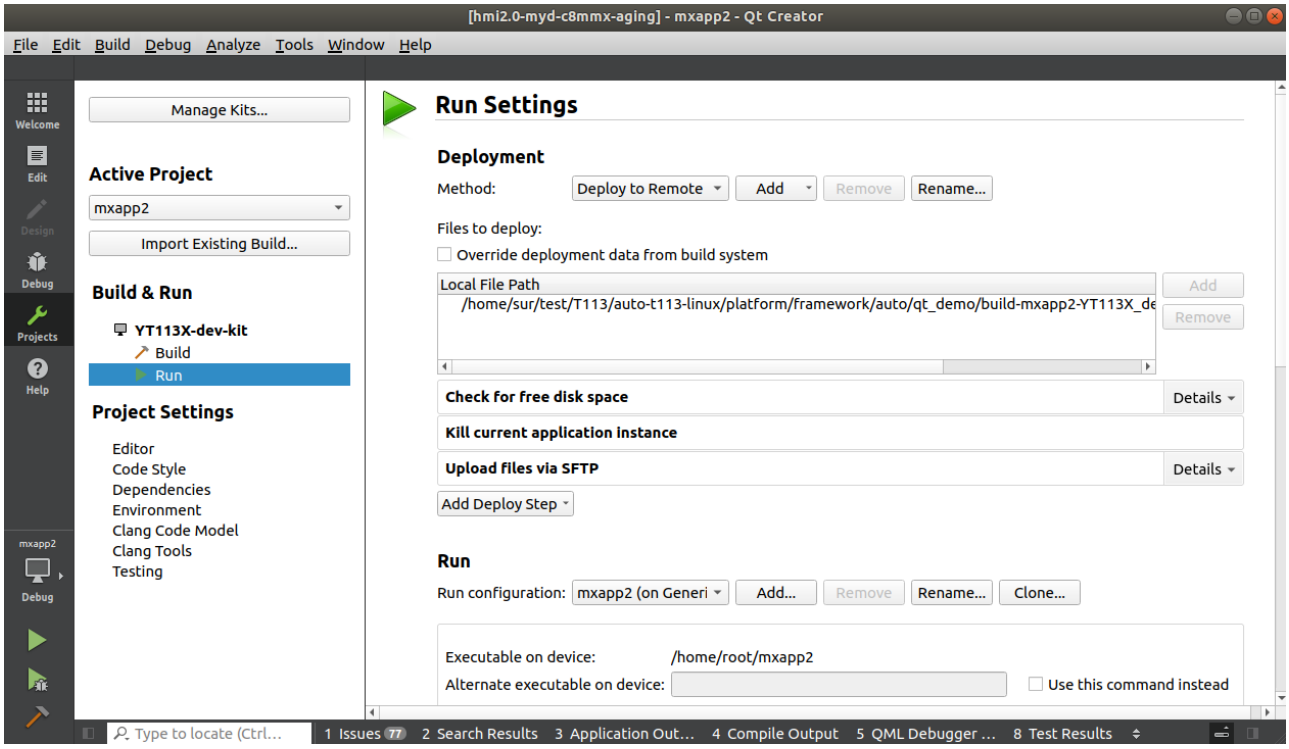

图 5-13.选择配置的 Kit

点击菜单栏"Build"->"Build"按钮，即可完成项目的编译，同时下侧会有编译过程输出。

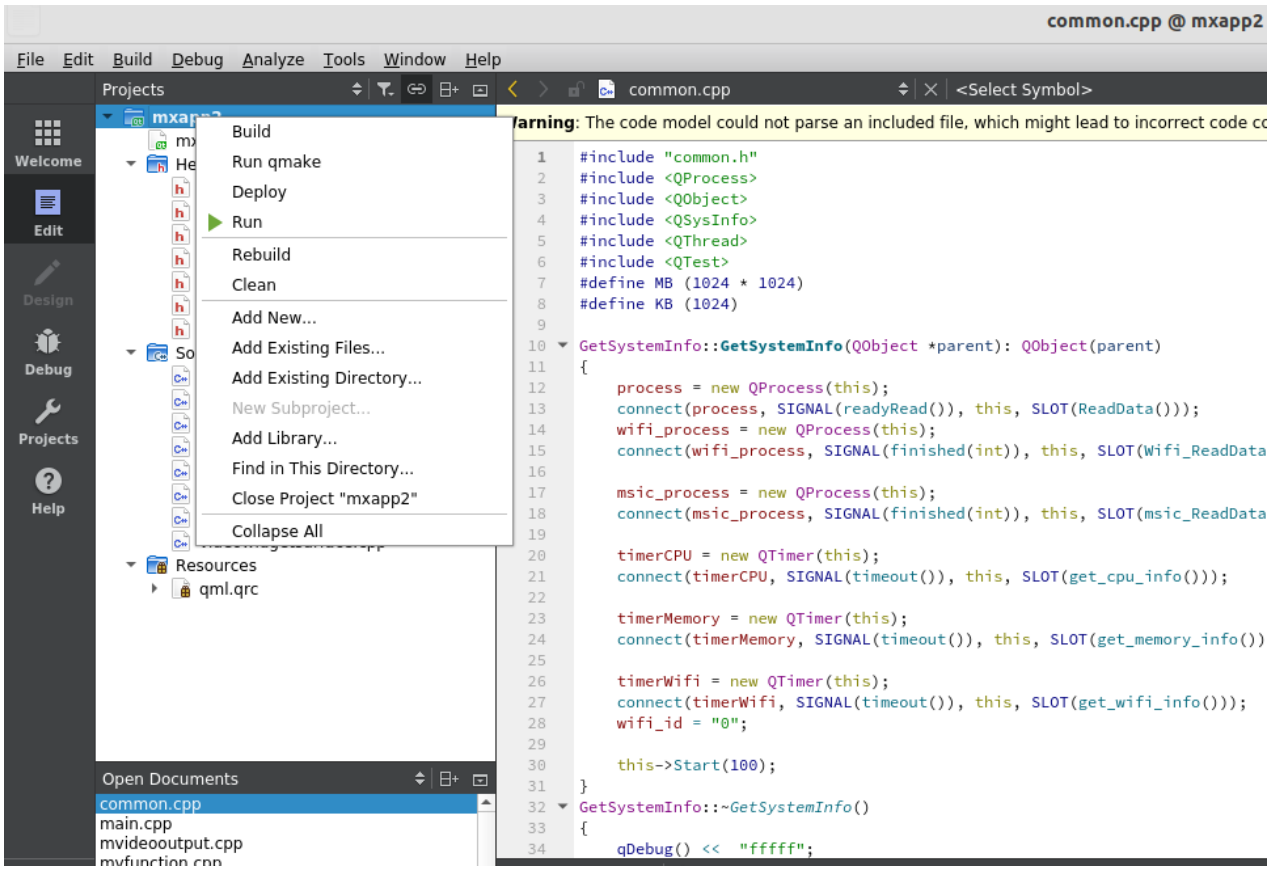

图 5-14.编译项目

图 5-15. 编译完成

QtCreator 构建 mxapp2 项目后，编译好的二进制文件存放在指定的目录下。然后将 mxapp2 文件拷贝到开发板下运行即可。

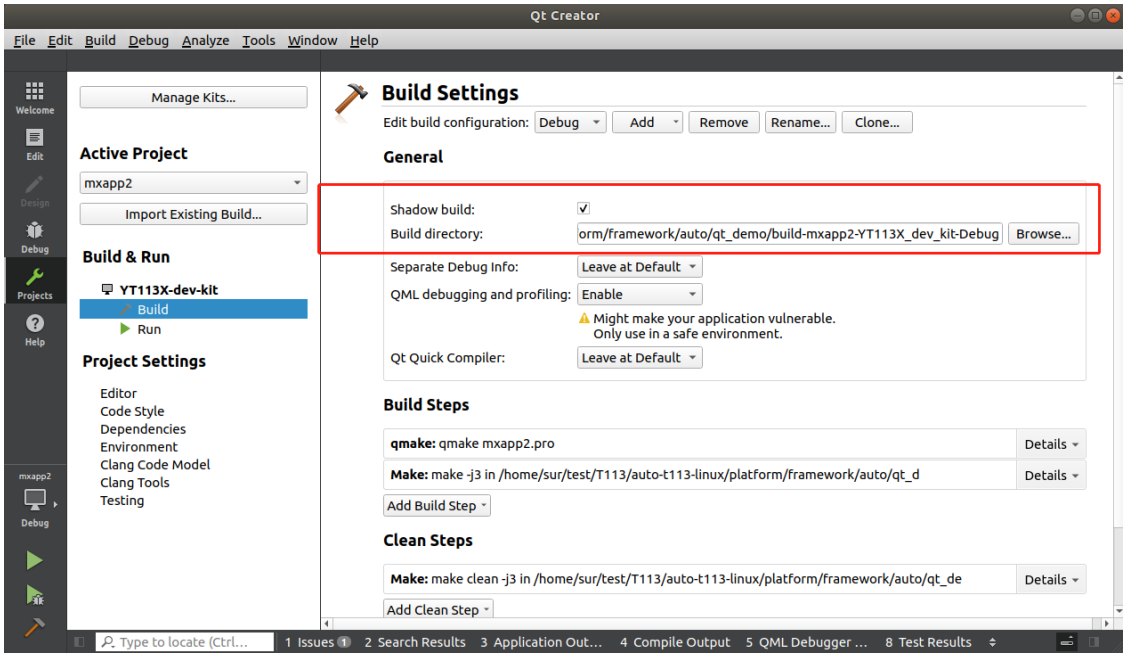

图 5-16.输出目录

5.4. SDK 集成应用程序

当应用程序开发完成后，需要进行与系统的集成；开机自启动等操作。

首先将 QT 应用程序源代码拷贝到 platform/framework/auto/qt_demo 目录下

并将该目录下 build.sh 增加如下配置代码

```
if [ -d ./MXAPP ];then
        cd ./MXAPP
        make distclean
        ./makeMXAPP
        echo "=====build MXAPP success!!!======"
        cd ../
fi
```

clean.sh 目录下增加如下代码

```
if [ -d ./MXAPP ];then
        cd ./MXAPP
        make distclean
        cd ../
fi
```

在 MXAPP 目录下增加一个编译控制脚本文件 makeMXAPP(命名同 build.sh 中添加)，代码如下：

```
#!/bin/sh
PATH=$LICHEE_BR_OUT/host/bin/:$PATH
$QT_INSTALL_DIR/bin/qmake  -o Makefile mxapp2.pro
make -j32
```

增加到开机自启动中。开机脚本路径为：platform/framework/auto/rootfs/etc/qtenv.sh

```
export  QTDIR=/usr/local/Qt_5.12.5
if [ -d $QTDIR ];then

        export  QT_ROOT=$QTDIR
        export  PATH=$QTDIR/bin:$PATH
```

```
export  LD_LIBRARY_PATH=$QTDIR/lib:/usr/lib/cedarx/:$LD_LIBRARY_PATH

export QT_QPA_PLATFORM_PLUGIN_PATH=$QT_ROOT/plugins
export QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
export QT_QPA_FONTDIR=$QT_ROOT/fonts

export QML_IMPORT_PATH=$QTDIR/qml
export QML2_IMPORT_PATH=$QTDIR/qml

TouchDevice="generic ft5x06 (79)"
for InputDevices in /sys/class/input/input*
do
        DeviceName=`cat $InputDevices/name`
        if [[ $DeviceName == $TouchDevice ]];then
          TouchDeviceNum=${InputDevices##*input}
          export QT_QPA_EVDEV_TOUCHSCREEN_PARAMETERS=/dev/input/eve
nt$TouchDeviceNum
          echo "add "/dev/input/event$TouchDeviceNum "to Qt Application."
          break
        fi
done
if [ ! -n "$TouchDeviceNum" ]; then
 echo "Error:Input device $TouchDevice can not be found,plz check it!"
fi

export QT_QPA_PLATFORM=linuxfb
export set TSLIB_TSDEVICE=/dev/input/dev/input/event$TouchDeviceNum
export set TSLIB_CONFFILE=/etc/ts.conf
export set TSLIB_PLUGINDIR=/usr/lib/ts
export set TSLIB_CALIBFILE=/etc/pointercal
export set TSLIB_CONSOLEDEVICE=none
export set TSLIB_FBDEVICE=/dev/fb0
```

```
    export QT_QPA_GENERIC_PLUGINS=evdevtouch,evdevmouse:/dev/input/eve
nt6
    #export QT_QPA_EGLFS_INTEGRATION=eglfs_mali
    #export QT_QPA_FB_HIDECURSOR=1
    #export QT_QPA_EGLFS_HIDECURSOR=1
    #export QT_QPA_EGLFS_ROTATION=90

    #export QWS_MOUSE_PROTO=Intellimouse:/dev/input/event6

    #export DBUS_SESSION_BUS_ADDRESS=`cat /tmp/dbusaddr`
    mkdir -p /dev/shm
    ulimit -c unlimited
    mxapp2 &
    echo "find qt5 installed done"
fi
```

- /dev/fb0：表示第一个主显示的接口
- generic ft5x06 (79)：触摸屏接口名字
- mxapp2 &：应用程序启动名

至此 QT 应用程序开发与集成就完成，在 SDK 目录下执行如下命令即可打包生产镜像

```
PC$ ./build.sh
PC$ ./build.sh qt
PC$ ./build.sh
PC$ ./build.sh pack
```

具体方式可参考《MYD-YT113X Linux 系统开发指南》相关章节

6.参考文献

https://ubuntu.com/download/desktop

https://www.qt.io/

附录一 联系我们

深圳总部

地址：深圳市龙岗区坂田街道发达路云里智能园 2 栋 6 楼 04 室

负责区域：广东、广西、海南、重庆、云南、贵州、四川、西藏、香港、澳门

传真：0755-25532724　电话：0755-25622735

武汉研发中心

地址：武汉东湖新技术开发区关南园一路 20 号当代科技园 4 号楼 1601 号

电话：027-59621648

华东地区

地址：上海市浦东新区金吉路 778 号浦发江程广场 1 号楼 805 室

负责区域：上海、福建、浙江、江苏、安徽、山东

传真：021-62087085　电话：021-62087019

华北地区

地址：北京市大兴区荣华中路 8 号院力宝广场 10 号楼 901 室

负责区域：辽宁、吉林、黑龙江、北京、天津、河北、山西、内蒙古、

湖北、湖南、江西、河南、陕西、甘肃、宁夏、青海、新疆

传真：010-64125474　电话：010-84675491

销售联系方式

网址：www.myir-tech.com

邮箱：sales.cn@myirtech.com

技术支持联系方式

电话：0755-22316235（深圳）027-59621647/027-59621648(武汉)

邮箱：support.cn@myirtech.com

如果您通过邮件获取帮助时，请使用以下格式书写邮件标题：

[公司名称/个人--开发板型号] 问题概述

这样可以使我们更快速跟进您的问题，以便相应开发组可以处理您的问题。

附录二 售后服务与技术支持

凡是通过米尔电子直接购买或经米尔电子授权的正规代理商处购买的米尔电子全系列产品，均可享受以下权益：

1、6 个月免费保修服务周期
2、终身免费技术支持服务
3、终身维修服务
4、免费享有所购买产品配套的软件升级服务
5、免费享有所购买产品配套的软件源代码，以及米尔电子开发的部分软件源代码
6、可直接从米尔电子购买主要芯片样品，简单、方便、快速；免去从代理商处购买时，漫长的等待周期
7、自购买之日起，即成为米尔电子永久客户，享有再次购买米尔电子任何一款软硬件产品的优惠政策
8、OEM/ODM 服务

如有以下情况之一，则不享有免费保修服务：
1、超过免费保修服务周期
2、无产品序列号或无产品有效购买单据
3、进液、受潮、发霉或腐蚀
4、受撞击、挤压、摔落、刮伤等非产品本身质量问题引起的故障和损坏
5、擅自改造硬件、错误上电、错误操作造成的故障和损坏
6、由不可抗拒自然因素引起的故障和损坏

产品返修：

用户在使用过程中由于产品故障、损坏或其他异常现象，在寄回维修之前，请先致电米尔电子客服部，与工程师进行沟通以确认问题，避免故障判断错误造成不必要的运费损失及周期的耽误。

维修周期：

收到返修产品后，我们将即日安排工程师进行检测，我们将在最短的时间内维修或更换并寄回。一般的故障维修周期为 3 个工作日（自我司收到物品之日起，不计运输过程时间），由于特殊故障导致无法短期内维修的产品，我们会与用户另行沟通并确认维修周期。

维修费用：

在免费保修期内的产品，由于产品质量问题引起的故障，不收任何维修费用；不属于免费保修范围内的故障或损坏，在检测确认问题后，我们将与客户沟通并确认维修费用，我们仅收取元器件材料费，不收取维修服务费；超过保修期限的产品，根据实际损坏的程度来确定收取的元器件材料费和维修服务费。

运输费用：

产品正常保修时，用户寄回的运费由用户承担，维修后寄回给用户的费用由我司承担。非正常保修产品来回运费均由用户承担。