



RealEvo-Qt 产品手册

图形界面中间件

SPC003001 V1.00 Date: 2018/05/16

产品使用手册

类别	内容
关键词	Qt QtCreator 调试
摘 要	在 Windows 环境下使用 QtCreator 集成开发环境开发并调试 SpaceChain OS Qt 应用程序

修订历史

版本	日期	原因
V1.00	2018/05/16	创建文档

目 录

第 1 章 简介	2
1.1 Qt	2
1.2 Qt Creator	2
1.3 Qwt	2
1.4 RealEvo-QtSylixOS	2
第 2 章 安装 RealEvo-QtSylixOS	3
2.1 安装 Qt Creator	3
2.2 安装 RealEvo-QtSylixOS	3
第 3 章 配置 RealEvo-QtSylixOS	5
3.1 打开 RealEvo-QtSylixOS	5
3.2 配置 RealEvo-QtSylixOS	5
3.3 配置 SylixOS 设备	5
3.4 配置构建套件 (Kit)	8
3.5 构建套件 (Kit) 的选择	9
第 4 章 部署 Qt 共享库	11
4.1 部署 Qt 共享库	11
4.2 屏幕校准	12
第 5 章 开发 Qt Widgets 应用程序	14
5.1 新建 Qt Widgets 应用程序	14
5.2 修改 Qt Widgets 应用程序	16
5.3 构建 Qt Widgets 应用程序	19
5.4 运行 Qt Widgets 应用程序	20
5.5 调试 Qt Widgets 应用程序	23
5.6 Attach 方式调试 Qt Widgets 应用程序	26
5.7 Qt Widgets 应用程序加入 Qwt	28
第 6 章 开发 Qt Quick 应用程序	30
6.1 新建 Qt Quick 应用程序	30
6.2 修改 Qt Quick 应用程序	34
6.3 分析 Qt Quick 应用程序	34
第 7 章 开发其它类型的项目	37
7.1 开发 Qt 控制台应用程序	37
7.2 开发 SylixOS 纯 C 语言项目	37
7.3 开发 SylixOS 纯 C++语言项目	38
7.4 开发 SylixOS 共享库项目	39
7.5 开发 SylixOS 静态库项目	40
第 8 章 FAQ	41
8.1 如何使用 QtSerialPort 类库	41
8.2 如何使用 VNC	41
8.3 如何旋转屏幕	41

第1章 简介

1.1 Qt

Qt 是一个 1991 年由奇趣科技公司开发的跨平台 C++ 图形用户界面应用程序开发框架。距 1995 年发布的第一个版本，Qt 已经有超过 20 年的历史。它既可以用于开发 GUI（Graphical User Interface，即图形用户界面）程序，也可以用于开发诸如控制台工具和服务器等的非 GUI 程序。

2008 年，奇趣科技公司被诺基亚公司收购，Qt 也因此成为诺基亚公司旗下的编程语言工具。2009 年，诺基亚公司将 Qt 的授权协议增加了对商业友好的 LGPL v2.1 协议。2012 年，Qt 被 Digia 公司收购，同年推出了 Qt 5.0 版本。

Qt 已经被广泛地应用在各种行业领域的桌面和嵌入式及移动计算系统；据了解，全球有超过 7000 家公司和 80 万的开发者在使用 Qt。

Qt 的官方网站如下：<http://www.qt.io/>

1.2 Qt Creator

Qt Creator 是 2009 年 Qt 被诺基亚公司收购后推出的一款轻量级的跨平台的集成开发环境。Qt Creator 主要是为了帮助 Qt 新用户更快速地入门并运行项目，还可以提高有经验的 Qt 开发者的工作效率。

Qt Creator 包括项目生成向导、高级的 C++ 代码编辑器、浏览文件及类的工具、图形化的 GDB 调试前端等功能，集成了 Qt Designer（界面设计师）、Qt Assistant（帮助文件系统）、Qt Linguist（翻译工具）、qmake 构建工具等。

1.3 Qwt

Qwt，全称是 Qt Widgets for Technical Applications，是一个基于 LGPL 协议的开源项目，可生成各种统计图和仪表图。

Qwt 的官方网站如下：https://osdn.jp/projects/sfnet_qwt/

1.4 RealEvo-QtSylixOS

2013 年 3 月，Qt 被移植到 SylixOS 嵌入式实时操作系统，SylixOS 系统成为全球极少数几个支持 Qt 的嵌入式实时操作系统。

RealEvo-QtSylixOS 是北京翼辉信息技术有限公司专门为开发 SylixOS 系统上的 Qt 应用程序而推出的软件。

RealEvo-QtSylixOS 集成了 Qt Creator 的 SylixOS 插件、各种类型处理器上的基于 LGPL 协议的 Qt 共享库、支持 Python 的 gdb 调试工具、Qwt 等。

SylixOS 插件使得 Qt Creator 几乎零配置就可以开发 SylixOS 上的 Qt 应用程序，一键部署 Qt 共享库到 SylixOS 设备使得 Qt 应用程序环境的配置变得异常简单，一键部署、运行、调试、分析 Qt 应用程序极大地提升了开发者的效率。

同时，RealEvo-QtSylixOS 集成了各种类型处理器上基于 LGPL 协议的 Qt 共享库，既使得开发者免于编译 Qt，将精力集中于开发 Qt 应用程序上。

注：LGPL 允许商业软件通过类库引用（link）方式使用 LGPL 类库而不需要开源商业软件的代码。这使得采用 LGPL 协议的开源代码可以被商业软件作为类库引用并发布和销售。

第2章 安装 RealEvo-QtSylixOS

2.1 安装 Qt Creator

打开 SylixOS IDE 安装程序如图 2.1 所示，选择“Install QtCreator”安装 Qt Creator 开发环境。



图 2.1 SylixOS IDE 安装程序

注：需要首先安装 Qt Creator，再安装 RealEvo-QtSylixOS，并且不能安装其它版本的 Qt Creator。

2.2 安装 RealEvo-QtSylixOS

点击 SylixOS IDE 安装程序中的“Install RealEvo-QtSylixOS”安装选项开始安装 QtSylixOS，指定安装到 QtCreator 的安装目录，根据需要选择对应 CPU 体系架构的 Qt SDK，默认全选（建议用户安装全部 Qt SDK），如图 2.2 所示。

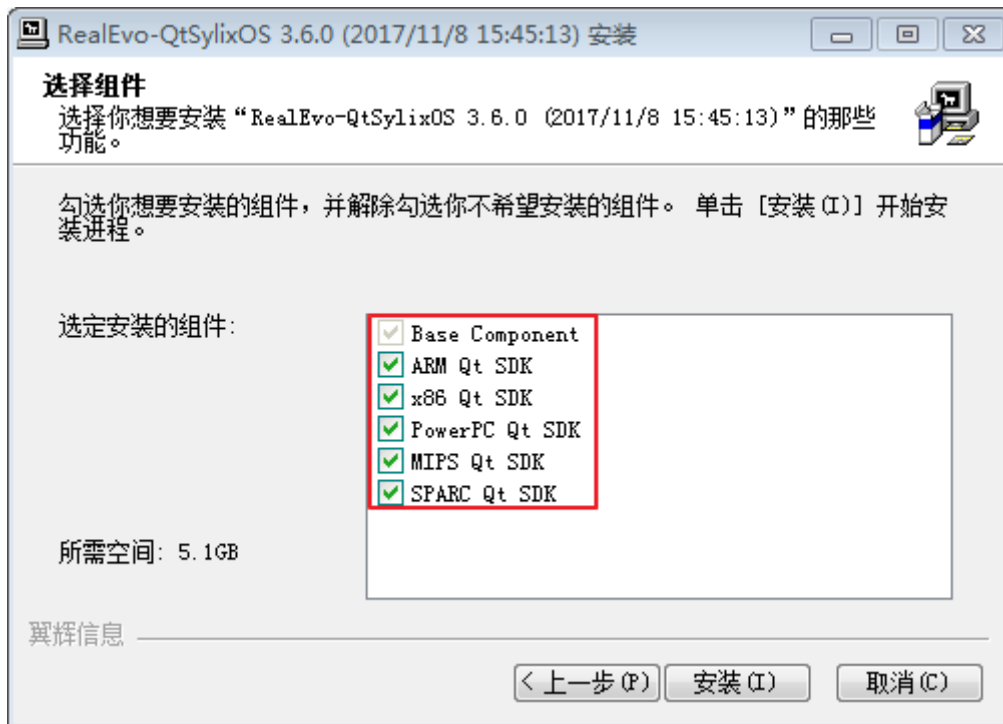


图 2.2 RealEvo-QtSylixOS CPU 体系模块

注：安装 RealEvo-QtSylixOS 前需要退出 Qt Creator。

第3章 配置 RealEvo-QtSylixOS

3.1 打开 RealEvo-QtSylixOS

双击 RealEvo-QtSylixOS 的桌面图标 ，即可打开 RealEvo-QtSylixOS。

3.2 配置 RealEvo-QtSylixOS

点击菜单“工具 → 选项...”，弹出选项对话框，点击左边栏的“SylixOS”，将切换到“SylixOS”子页，如图 3.1 所示。

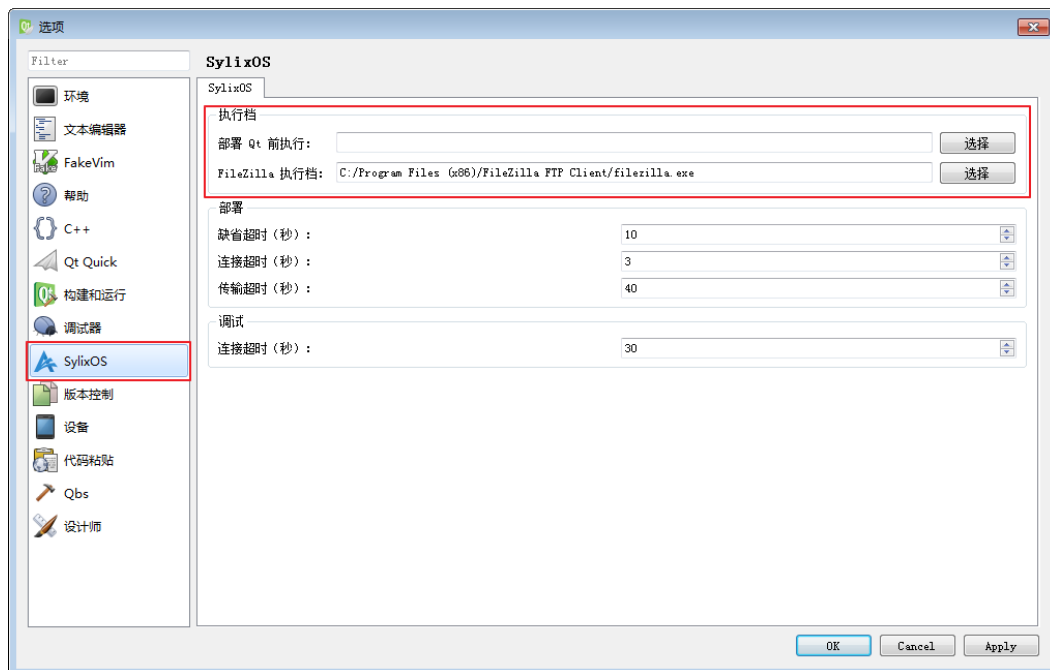


图 3.1 选项对话框

“执行档”组内，“部署 Qt 前执行”输入框可以指定一个 Windows 的可执行程序，该可执行程序在每次部署 Qt 共享库到 SylixOS 设备前都会被执行，该可执行程序一般实现对被部署的文件系统进行修改，以达到对不同的 SylixOS 设备进行定制化的目的。

“执行档”组内，“FileZilla 执行档”输入框可以指定一个 FileZilla 的可执行程序 filezilla.exe，如果用户将 FileZilla 安装到默认的目录，那么将会自动识别。

“部署”和“调试”组内的时间参数不建议用户修改，保持默认即可。

3.3 配置 SylixOS 设备

点击图 3.1 中左边栏的“设备”，将切换到“设备”子页，点击“添加...”按钮，将弹出设备设置向导选择对话框，如图 3.2 所示。

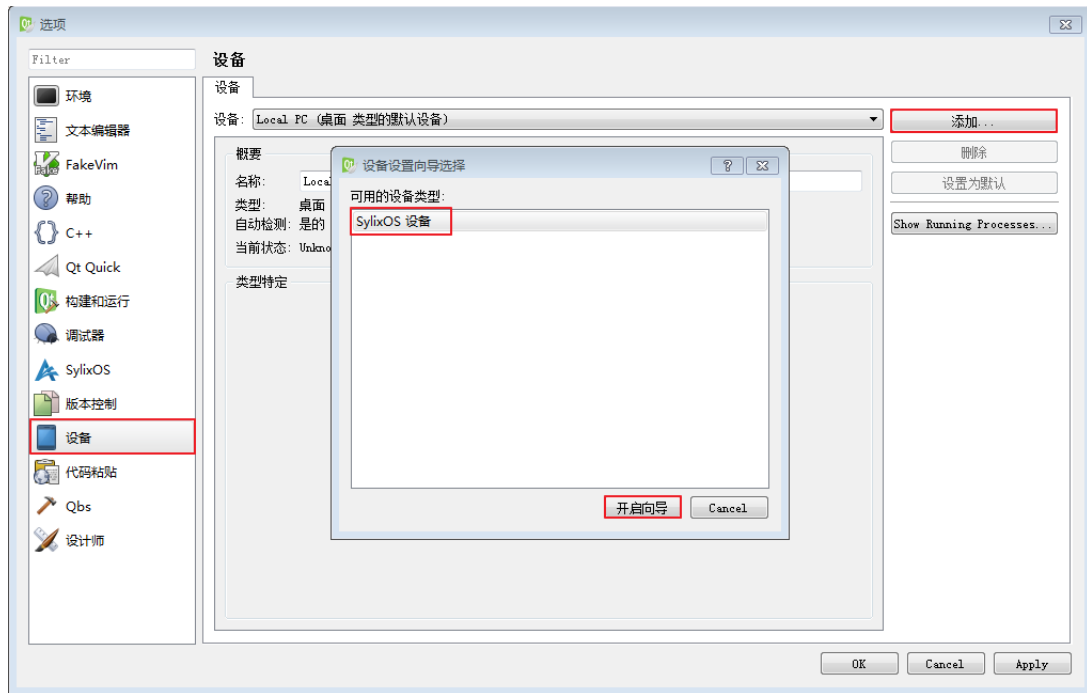


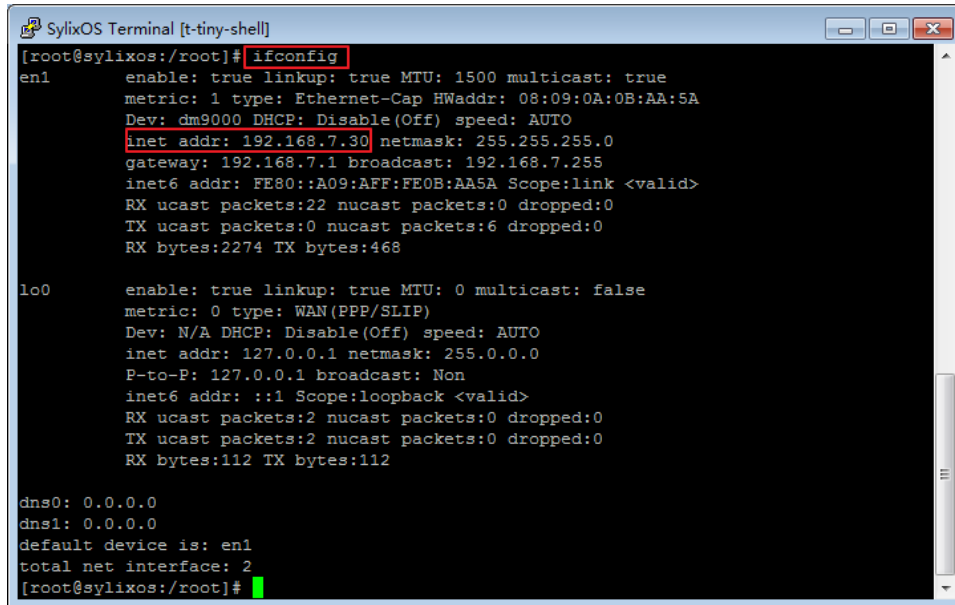
图 3.2 设备设置向导选择对话框

选择“SylixOS 设备”，然后点击“开启向导”按钮开始 SylixOS 设备配置设置向导，如图 3.3 所示。



图 3.3 SylixOS 设备配置设置向导

在“名字”输入框输入 SylixOS 设备的名字，如“SylixOS mini2440 设备”；在“设备 IP”输入框输入 SylixOS 设备的 IP 地址，如“192.168.7.30”；SylixOS 设备的 IP 地址可以在 SylixOS 设备的终端上输入“ifconfig”命令查看，如图 3.4 所示，“inet addr”的值便是 SylixOS 设备的 IP 地址。



```

SylixOS Terminal [t-tiny-shell]
[root@sylixos:/root]# ifconfig
en1      enable: true linkup: true MTU: 1500 multicast: true
        metric: 1 type: Ethernet-Cap HWaddr: 08:09:0A:0B:AA:5A
        Dev: dm9000 DHCP: Disable(Off) speed: AUTO
        inet addr: 192.168.7.30 netmask: 255.255.255.0
        gateway: 192.168.7.1 broadcast: 192.168.7.255
        inet6 addr: FE80::A09:AFF:FE0B:AA5A Scope:link <valid>
        RX ucast packets:22 nucast packets:0 dropped:0
        TX ucast packets:0 nucast packets:6 dropped:0
        RX bytes:2274 TX bytes:468

lo0      enable: true linkup: true MTU: 0 multicast: false
        metric: 0 type: WAN(PPP/SLIP)
        Dev: N/A DHCP: Disable(Off) speed: AUTO
        inet addr: 127.0.0.1 netmask: 255.0.0.0
        P-to-P: 127.0.0.1 broadcast: Non
        inet6 addr: ::1 Scope:loopback <valid>
        RX ucast packets:2 nucast packets:0 dropped:0
        TX ucast packets:2 nucast packets:0 dropped:0
        RX bytes:112 TX bytes:112

dns0: 0.0.0.0
dns1: 0.0.0.0
default device is: en1
total net interface: 2
[root@sylixos:/root]#

```

图 3.4 ifconfig 命令

在“GDB 端口”输入框输入设备开启 Qt 应用程序调试时监听的端口号，如“1234”；在“用户名”输入框输入用户名“root”；在“密码”输入框输入密码“root”。

在构建套件下拉框选择设备适配的构建套件（Kit）（构建套件（Kit）在配置设备的时候只能选择这里适配的设备）。

点击“下一步”按钮进入设备总结对话框，在设备总结对话框点击“完成”按钮进入设备测试对话框（需要确保 SylixOS 设备可以通过网络连接 PC 机），如图 3.5 所示。



```

Device Test

尝试连接到 SylixOS 设备: 192.168.7.30
正在连接...
连接成功.
正在登录...
登录成功.
启动 SylixOS 设备测试...

[root@sylixos:/root]#
cat /proc/cpuinfo
CPU       : SAMSUNG S3C2440A (ARM920T 405/101MHz NonFPU)
WORDLEN   : 32
NCFPU     : 1
ACTIVE    : 1
PWRLevel  : Top level
CACHE     : 32KBytes L1-Cache (D-16K/I-16K)
PACKET    : Mini2440 Packet

[root@sylixos:/root]#
cat /proc/version
SylixOS kernel version: 1.4.0 (3) Code name: LongYuan BSP version 5.1.2 for
LongYuan
(compile time: Nov 12 2016 18:48:28)
GCC:4.9.3

[root@sylixos:/root]#
Device test finished successfully.

Close

```

图 3.5 设备测试对话框

设备测试成功后点击“Close”按钮完成设置，返回“设备”子页后点击“Apply”按钮应用设置，如图 3.6 所示。

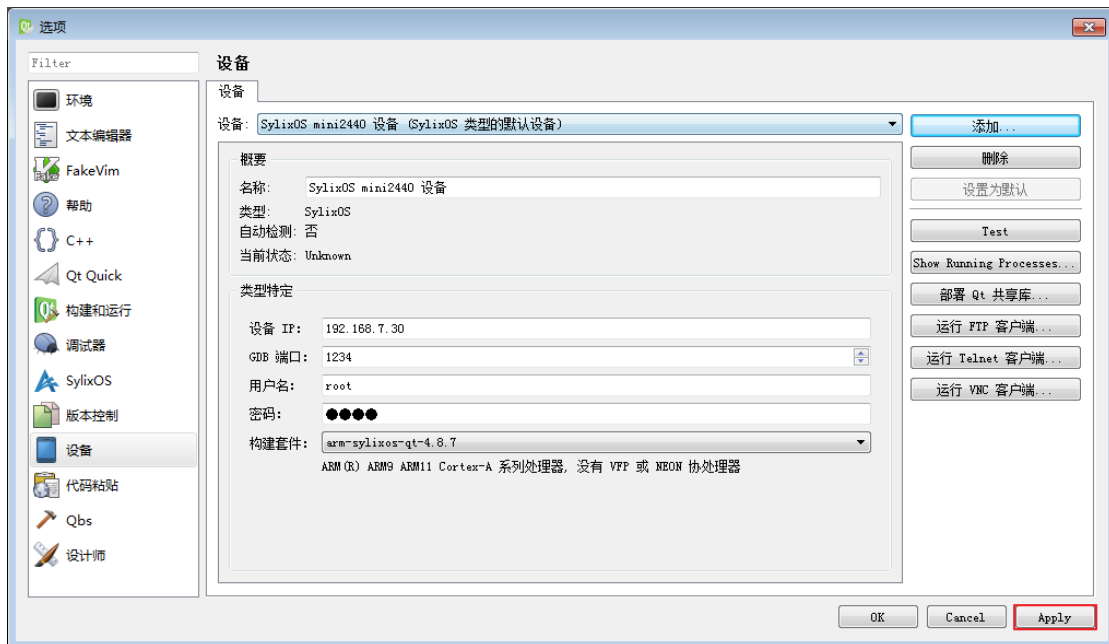


图 3.6 “设备”子页

在图 3.6 中，分别点击“运行 FTP 客户端...”、“运行 Telnet 客户端...”、“运行 VNC 客户端...”三个按钮可快速地运行 FTP 客户端、Telnet 客户端、VNC 客户端并连接到 SylixOS 设备，VNC 的详细信息见 8.2 节。

3.4 配置构建套件 (Kit)

点击选项对话框左边栏的“构建和运行”，并切换到“构建套件 (Kit)”选项卡，如图 3.7 所示。

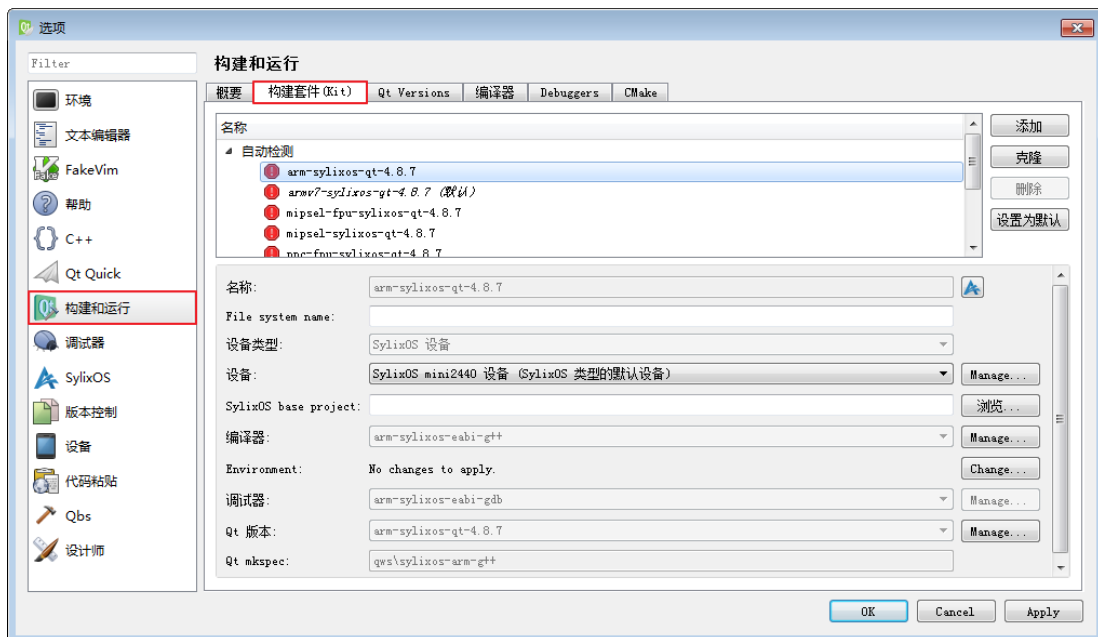


图 3.7 “构建套件 (Kit)” 选项卡

选中“自动检测”下的“arm-sylixos-qt-4.8.7”构建套件 (Kit)，设备选择“SylixOS mini2440 设备 (SylixOS 类型的默认设备)”；“SylixOS base project”通过点击右侧的“浏览...”按钮选择“CPU type”为“arm920t”的已经编译完成的 SylixOS base 工程的目录，如图 3.8 所示。

注：“设备”选项将自动匹配与 SylixOS base 工程编译器类型相同的设备（已经创建成功的设备）。

注：SylixOS base 工程的创建方法可参考《RealEvo-IDE 使用手册》。

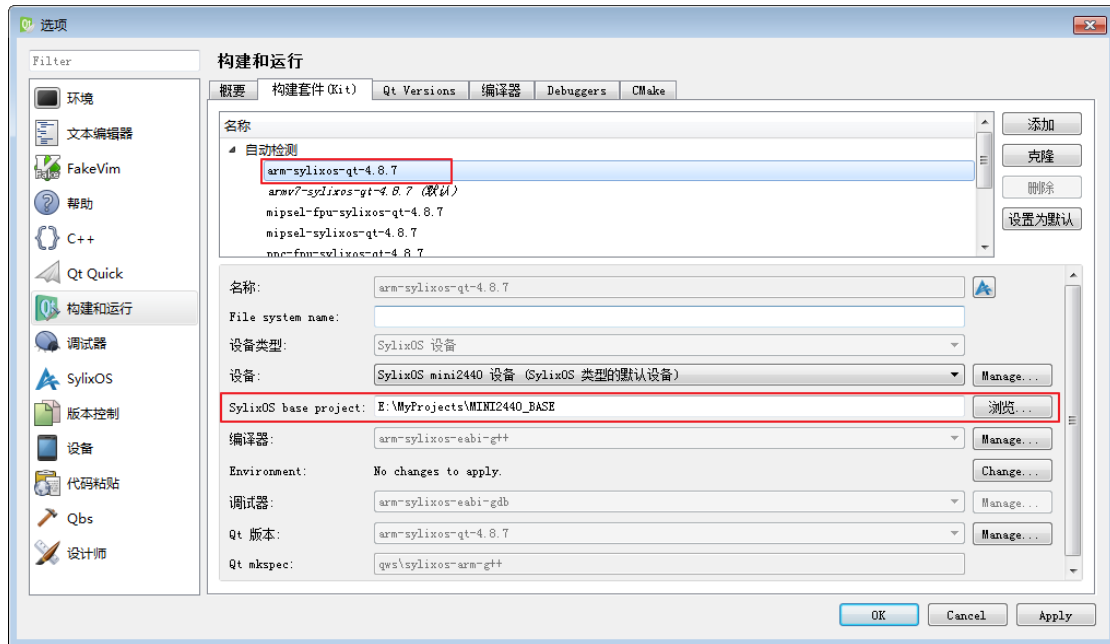


图 3.8 SylixOS base project 设置和设备选择

选中“自动检测”下的“armv7-sylixos-qt-4.8.7”构建套件 (Kit)，设备选择“SylixOS smart210 设备 (SylixOS 类型的默认设备)”（如果有），“SylixOS base project”通过点击右侧的“浏览...”按钮选择您的“CPU type”为“cortex-a8”的已经编译完成的 SylixOS base 工程的目录。

最后点击“OK”按钮完成设置（如果构建套件 (Kit) 配置成功，对应的构建套件 (Kit) 前面的红色叹号会消失。如图 3.8 所示所有的构建套件 (Kit) 都已经配置成功）。

3.5 构建套件 (Kit) 的选择

“arm-sylixos-qt-4.8.7”构建套件 (Kit) 使用“-march=armv4”软浮点参数编译，适用于所有的 ARM 处理器（如 ARM9、ARM11、Cortex-A 系列等）。

“armv7-sylixos-qt-4.8.7”构建套件 (Kit) 使用“-march=armv7-a -mfp=neon”参数编译，适用于所有带 NEON 协处理器的 ARMv7-A 架构的处理器（如 Cortex-A8、Cortex-A9 等）。该构建套件 (Kit) 使用了 ARM 的 VFP-v3 及 NEON 的 SIMD 指令集，性能较“arm-sylixos-qt-4.8.7”构建套件 (Kit) 的要好。

“mipsel-sylixos-qt-4.8.7”构建套件 (Kit) 使用“-mips32 -msoft-float”参数编译，适用于所有不带硬浮点协处理器的 mips32、mips64 处理器（如龙芯 loongson1b 等）。

“mipsel-fpu-sylixos-qt-4.8.7”构建套件（Kit）使用“-mips32 -mhard-float”参数编译，适用于所有带硬浮点协处理器的 mips32、mips64 处理器（如龙芯 loongson1a、loongson2x、loongson3x、君正 jz4780、24kf 等）。

“ppc-sylixos-qt-4.8.7”构建套件（Kit）使用“-msoft-float”参数编译，适用于不带硬浮点协处理器的 PowerPC 处理器。

“ppc-fpu-sylixos-qt-4.8.7”构建套件（Kit）使用“-mhard-float”参数编译，适用于带硬浮点协处理器的 PowerPC 处理器（如 MPC750、MPC83xx 等）。

“x86-sylixos-qt-4.8.7”构建套件（Kit）适用于所有的 x86 体系结构的 32 位处理器。

“x64-sylixos-qt-4.8.7”构建套件（Kit）适用于所有的 x86 体系结构的 64 位处理器。

第4章 部署 Qt 共享库

4.1 部署 Qt 共享库

注：如果 SylixOS 设备已经部署了 Qt 共享库，可以跳过本章节，直接进入第 5 章章节。

点击菜单“工具 → 选项...”，弹出选项对话框，点击左边栏的“设备”，将切换到“设备”子页，如图 4.1 所示。

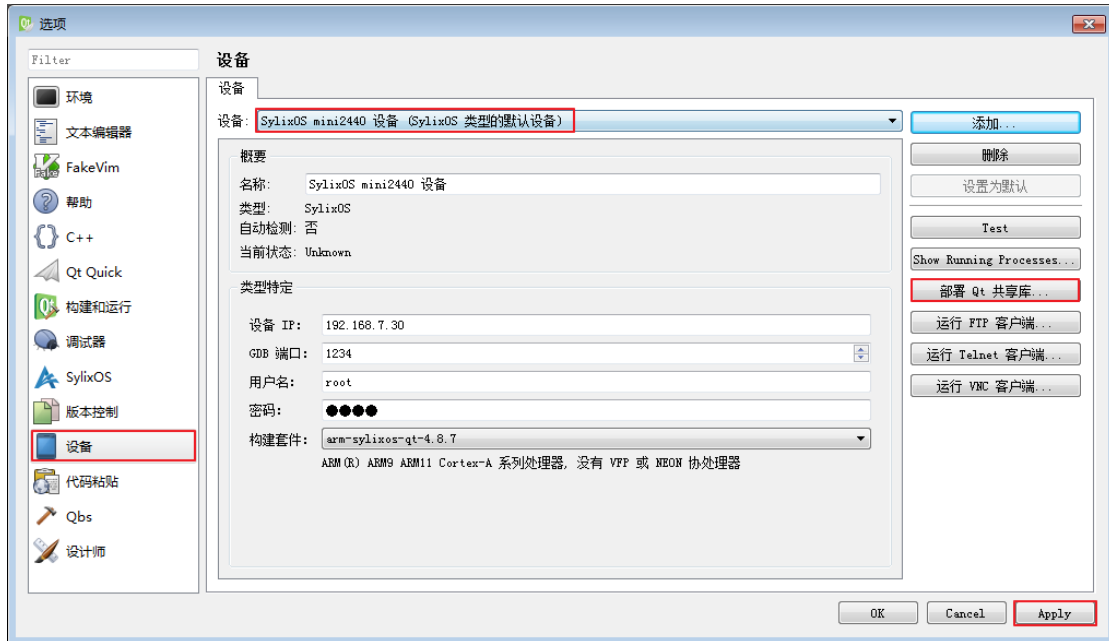


图 4.1 “设备”子页

设备下拉框选择“SylixOS mini2440 设备（SylixOS 类型的默认设备）”，然后点击“部署 Qt 共享库...”按钮，将弹出“部署 Qt 到 SylixOS 设备”对话框，如图 4.2 所示。

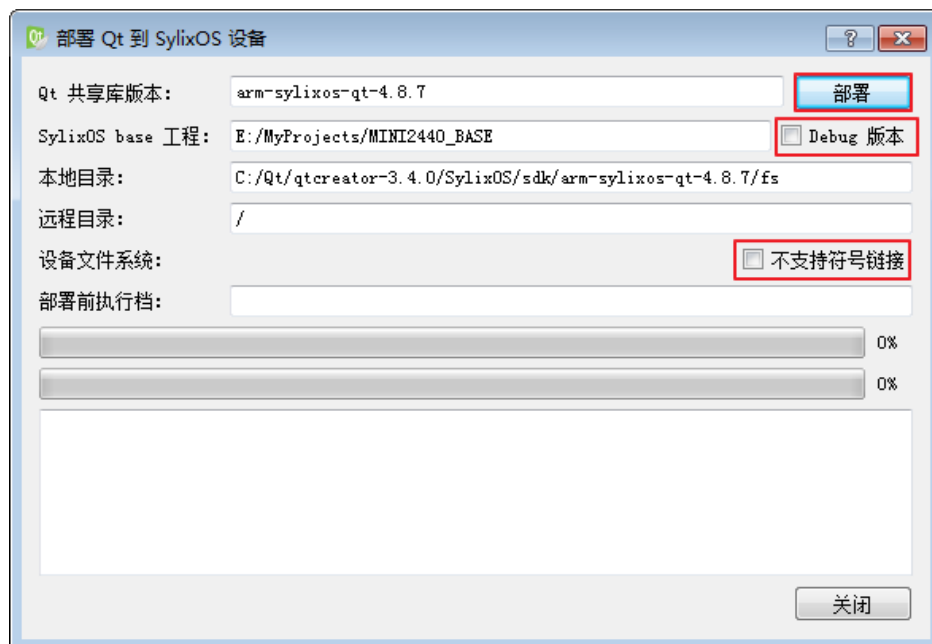


图 4.2 “部署 Qt 到 SylixOS 设备”对话框

如果您想部署 SylixOS base 工程里的 Debug 版本的共享库和内核模块，请勾选“Debug 版本”复选框，否则将部署“Release”版本的共享库和内核模块。

注：如果 SylixOS 设备使用不支持符号链接的文件系统（如 FAT32）作为存储介质的文件系统，需要勾选“不支持符号链接”复选框。

最后点击“部署”按钮开始 Qt 共享库的部署，如图 4.3 所示。

如果用户在“SylixOS”选项子页的“部署 Qt 前执行”输入框中指定了一个 Windows 的可执行程序，在部署前，该 Windows 可执行程序将被运行。

注：需要注意在部署使用硬盘作为存储介质的平台时，如果硬盘的分区大小可能会导致部署失败的情况。

注：部署 Qt 共享库时，需要依赖 SylixOS base 工程的：libsylixos、libcextern、libsalsa、openssl 四个组件，因此，创建 SylixOS base 工程时需要同时编译以上四个组件。

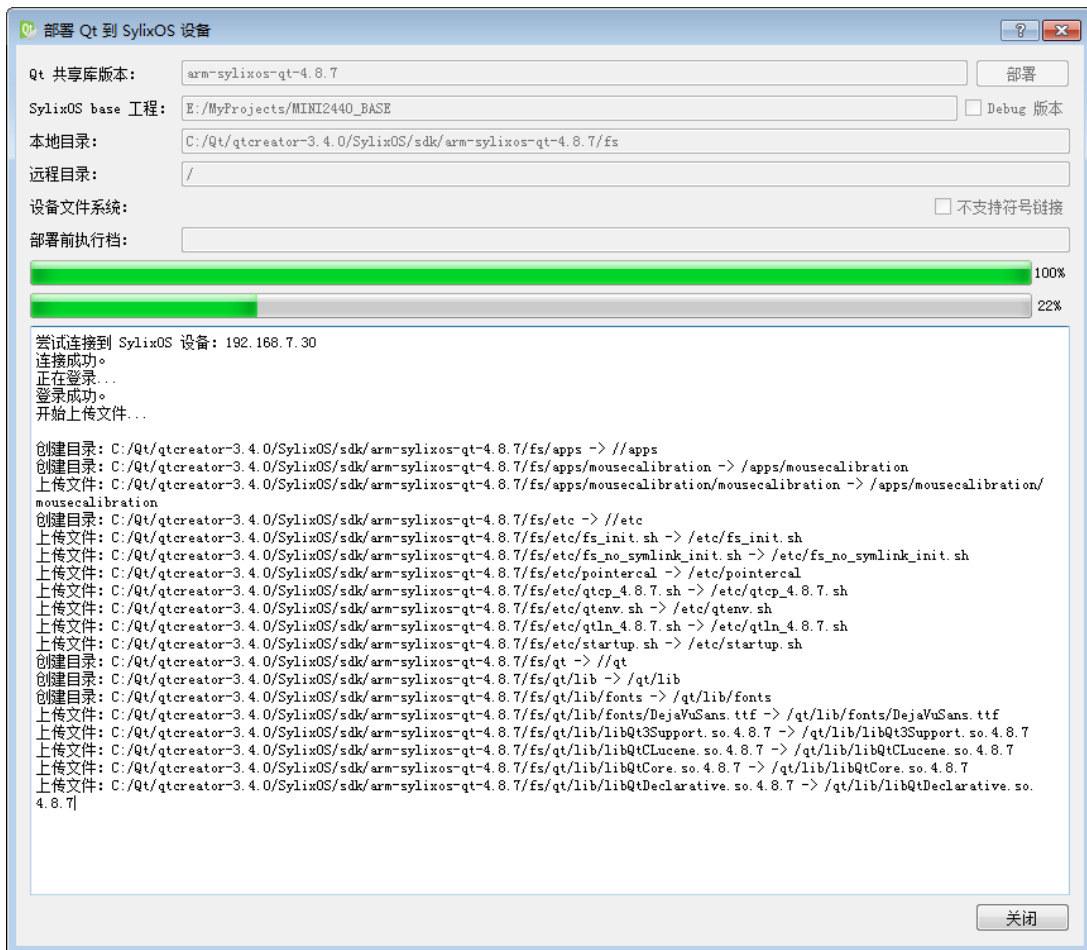


图 4.3 开始部署 Qt 共享库

4.2 屏幕校准

部署和初始化文件系统完毕后，将弹出屏幕校准询问对话框，如图 4.4 所示。

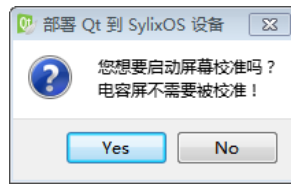


图 4.4 屏幕校准询问对话框

如果您的 SylixOS 设备使用的是电阻屏，请点击“**Yes**”按钮开始屏幕校准；如果您的 SylixOS 设备使用的是电容屏，请点击“**No**”按钮跳过屏幕校准。

按照 LCD 屏幕的提示，点击 LCD 屏幕的“+”符号完成屏幕校准，如图 4.5 所示。

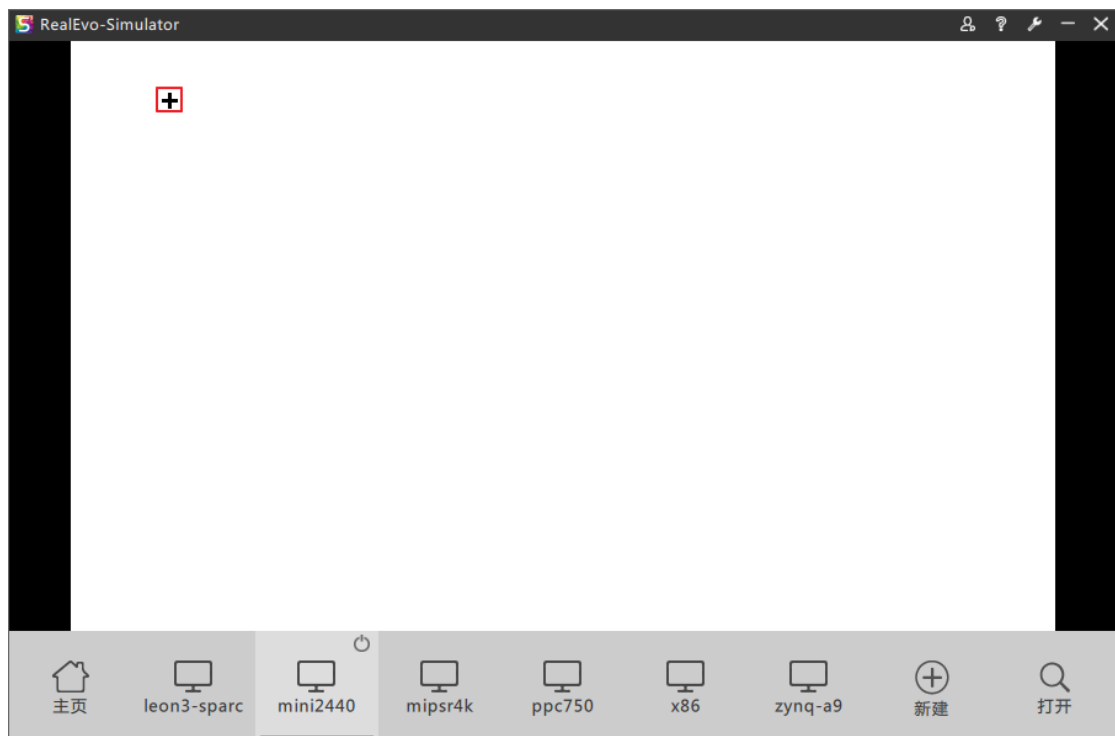


图 4.5 屏幕校准

注：模拟器平台中“leon3-sparc”、“mipsr4k”和“ppc750”不支持显示功能，“x86”不支持鼠标功能。

第5章 开发 Qt Widgets 应用程序

5.1 新建 Qt Widgets 应用程序

点击菜单“文件→新建文件或项目..”，将弹出新建对话框，如图 5.1 所示。

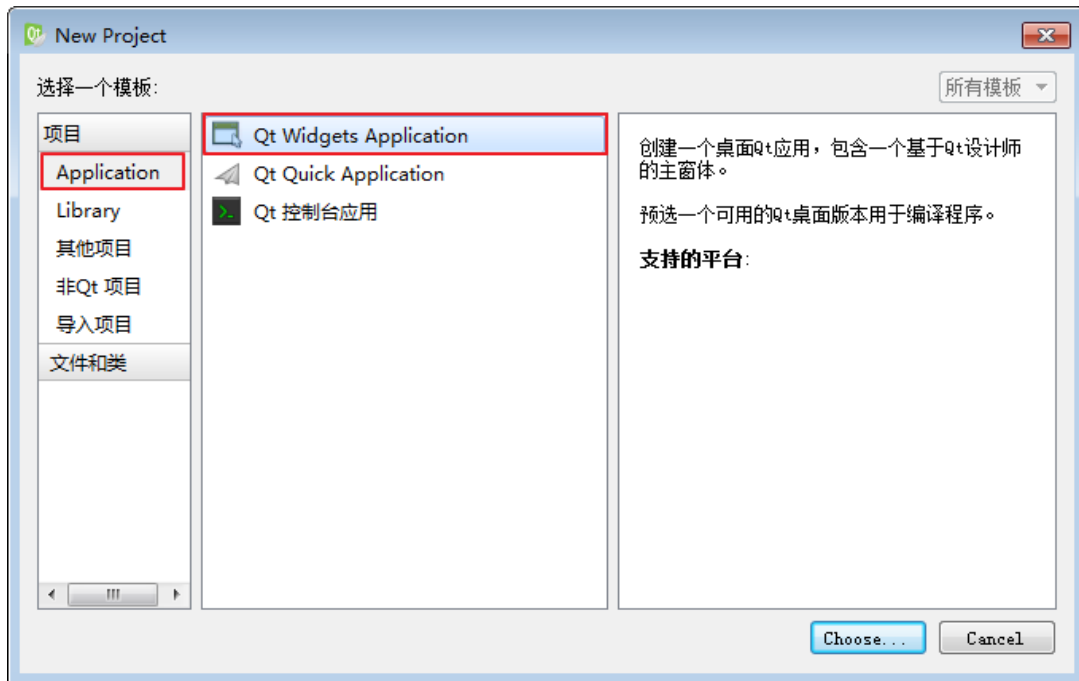


图 5.1 新建对话框

选择项目类型“Application”下的“Qt Widgets Application”模板，然后点击“Choose..”按钮进入“Qt Widgets Application”向导，如图 5.2 所示。

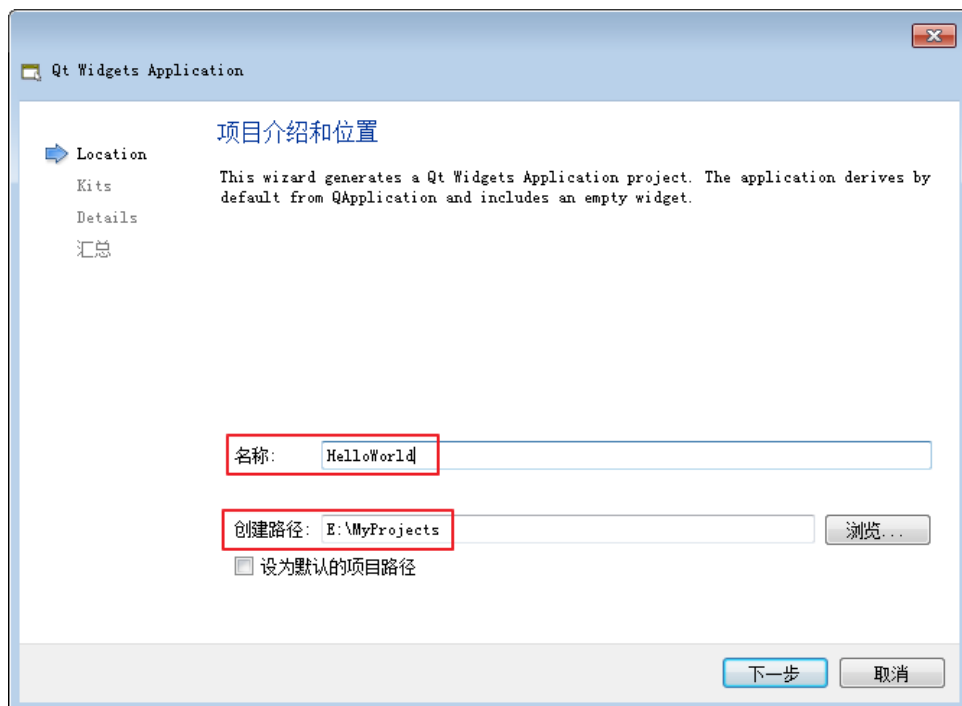


图 5.2 “Qt Widgets Application” 向导

输入项目名称为“HelloWorld”，创建路径通过点击“浏览...”按钮选择一个合适的项目存放文件夹，如“E:\MyProjects”文件夹，然后点击“下一步”按钮进入下一步，如图 5.3 所示。

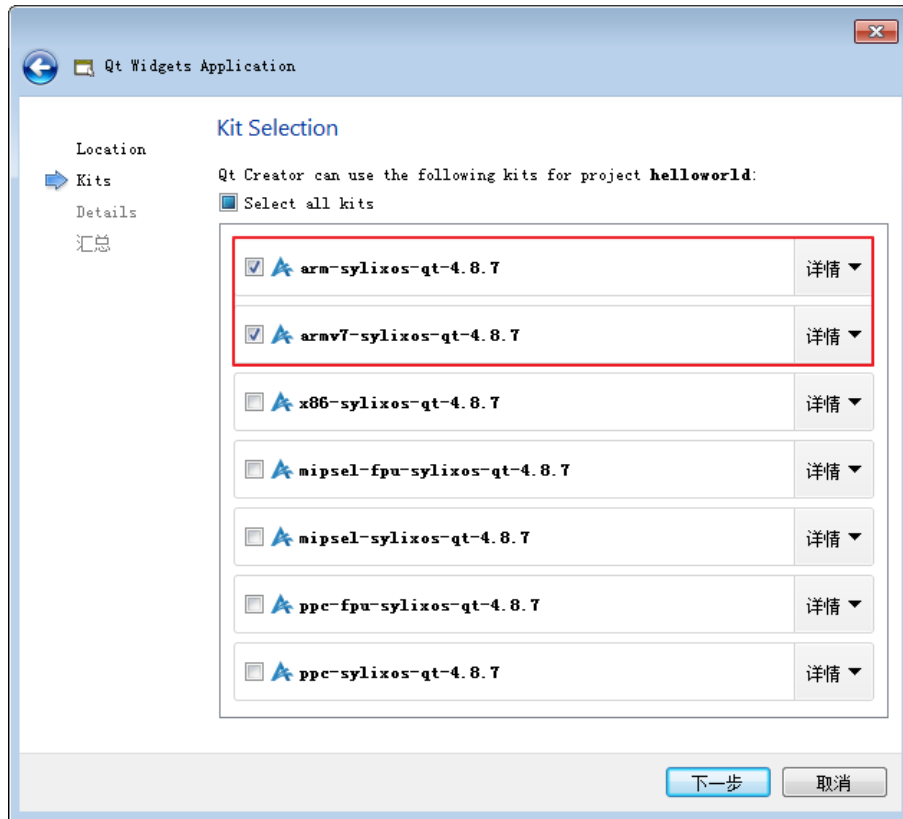


图 5.3 构建套件 (Kit) 选择

构建套件 (Kit) 可以选择“arm-sylixos-qt-4.8.7”和“armv7-sylixos-qt-4.8.7”，或者全选，然后点击“下一步”按钮按默认流程完成项目创建，创建完毕后，HelloWorld 项目如图 5.4 所示。

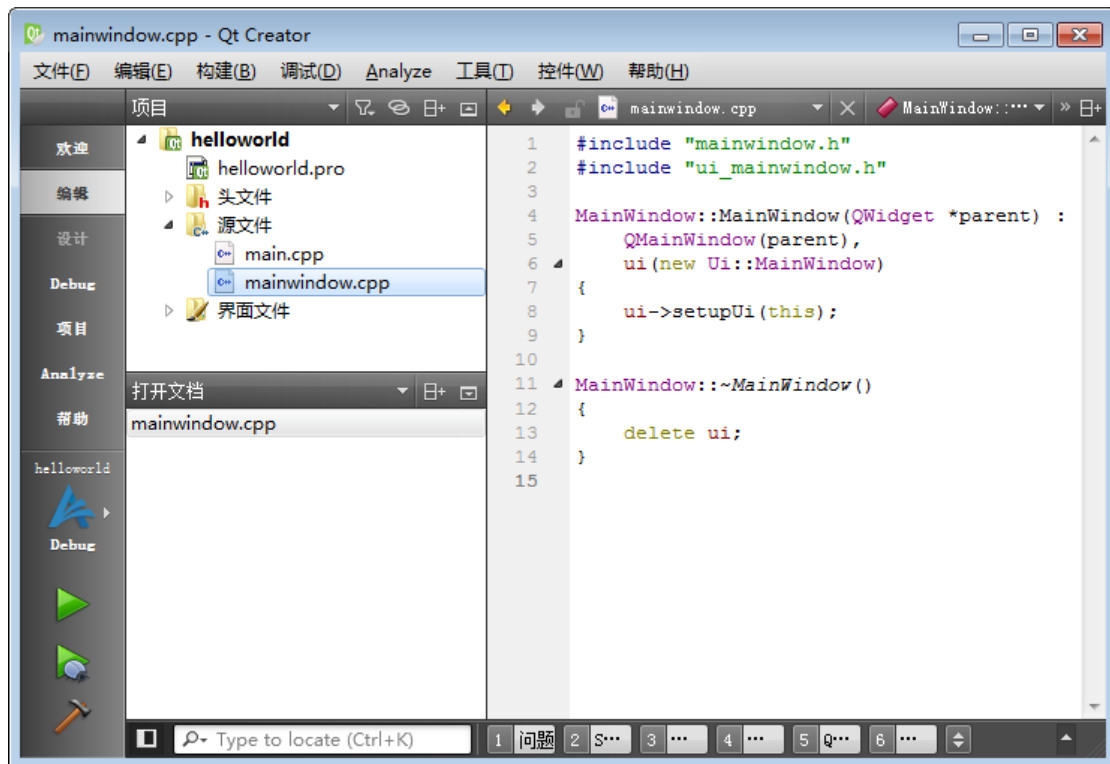


图 5.4 HelloWorld 项目

5.2 修改 Qt Widgets 应用程序

双击项目文件浏览器界面文件目录下的 `mainwindow.ui` 文件（如图 5.5 所示）即可打开界面设计师，如图 5.6 所示。

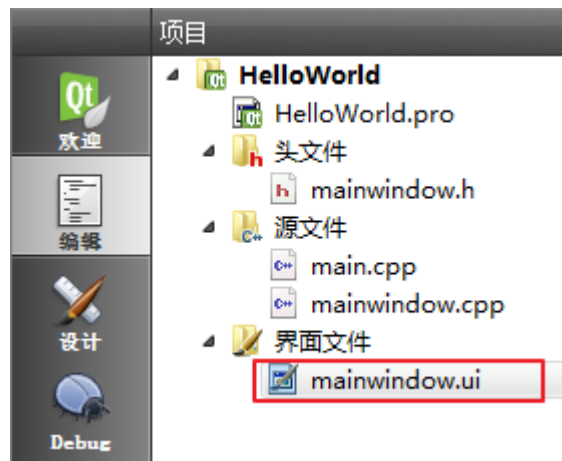


图 5.5 mainwindow.ui 文件

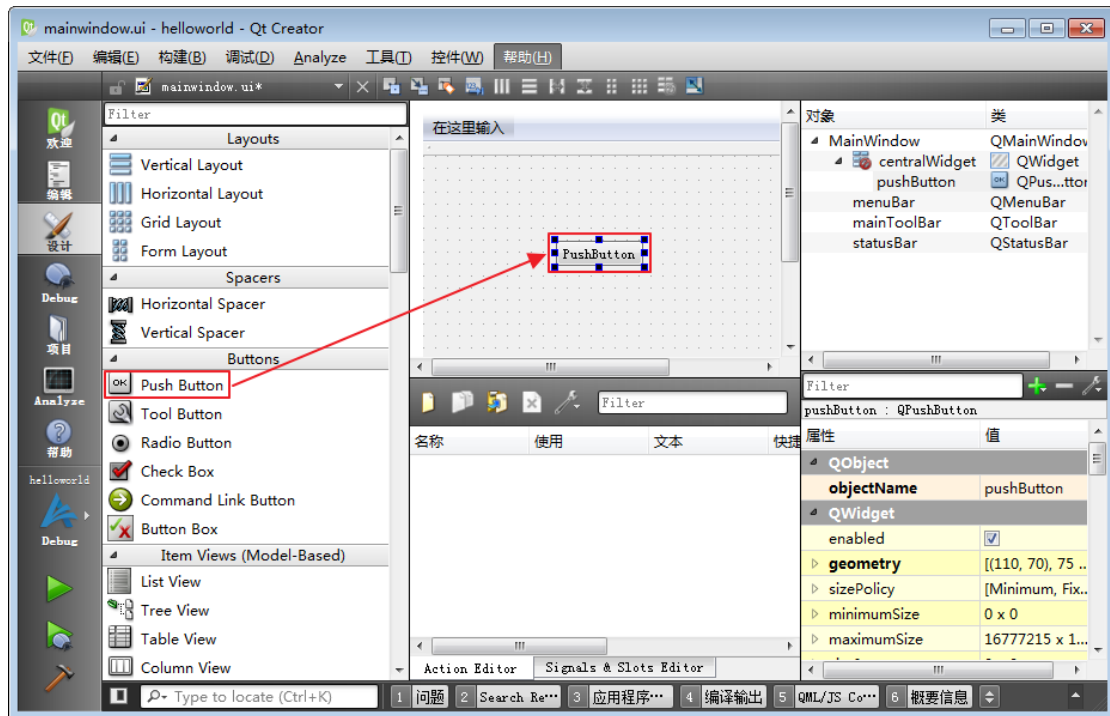


图 5.6 界面设计师

拖动一个“Push Button”类型的按钮到窗口的中间位置，然后保存文件（Ctrl+S），点



击编辑按钮

修改 mainwindow.h 文件如程序清单 5.1 所示（红色加粗字体为修改部分，下同）：

程序清单 5.1 mainwindow.h 文件

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
```

```
~MainWindow();

private slots:
    void btnClick();

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H
```

修改 mainwindow.cpp 文件如程序清单 5.2 所示。

程序清单 5.2 mainwindow.cpp 文件

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    connect(ui->pushButton, SIGNAL(clicked()), this, SLOT(btnClick()));
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::btnClick()
{
    printf("hello Qt\n");
}
```

代码的功能比较简单，界面只有一个按钮，当按钮按下时，在控制台打印“hello Qt”。

5.3 构建 Qt Widgets 应用程序



点击左下角的构建按钮，即可完成项目构建。当前使用的构建套件（Kit）是



“arm-sylixos-qt-4.8.7”，构建的是 Debug 版本，点击按钮可以完成构建套件（Kit）及版本切换，如图 5.7 所示。

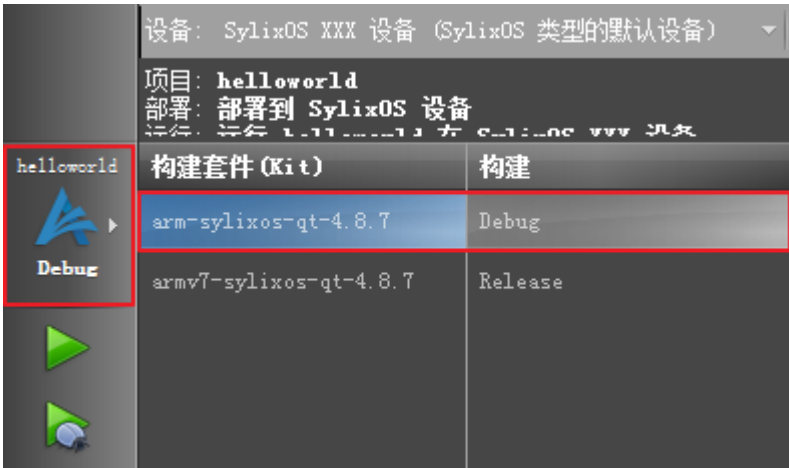


图 5.7 构建套件（Kit）及版本

构建的结果存放在项目存放文件夹下的“build-HelloWorld-arm_sylixos_qt_4_8_7-Debug”子文件夹，子文件夹名字取决于你当前选择的构建套件（Kit）和版本，文件夹内容如图 5.8 所示。

计算机 > 文档 (E:) > MyProjects > build-HelloWorld-arm_sylixos_qt_4_8_7-Debug					
包含到库中 共享 刻录 新建文件夹					
名称	修改日期	类型	大小		
HelloWorld	2016/11/9 10:19	文件	340 KB		
HelloWorld.strip	2016/11/9 10:19	STRIP 文件	23 KB		
main.o	2016/11/9 10:19	O 文件	136 KB		

图 5.8 构建结果目录

注：如图 5.8 所示，HelloWorld.strip 文件是去除掉不必要信息后的 HelloWorld 可执行文件，该文件比 HelloWorld 文件要小很多，适合部署到设备文件系统中。

5.4 运行 Qt Widgets 应用程序

我们需要确认项目的部署配置是“部署到 SylixOS 设备”，运行配置是“运行


HelloWorld 在 SylixOS mini2440 设备”。点击左侧的项目按钮，进入项目设置页，如图 5.9 所示。



图 5.9 项目设置页

选中构建套件 (Kit) “arm-sylixos-qt-4.8.7” 的“运行”按钮，确认项目的部署配置是“部署到 SylixOS 设备”，运行配置是“运行 HelloWorld 在 SylixOS mini2440 设备”；如果不是，点击“添加”按钮添加即可，如图 5.10 和图 5.11 所示。

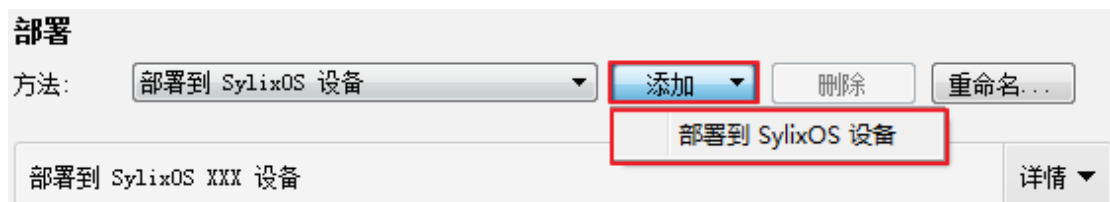


图 5.10 添加部署方法

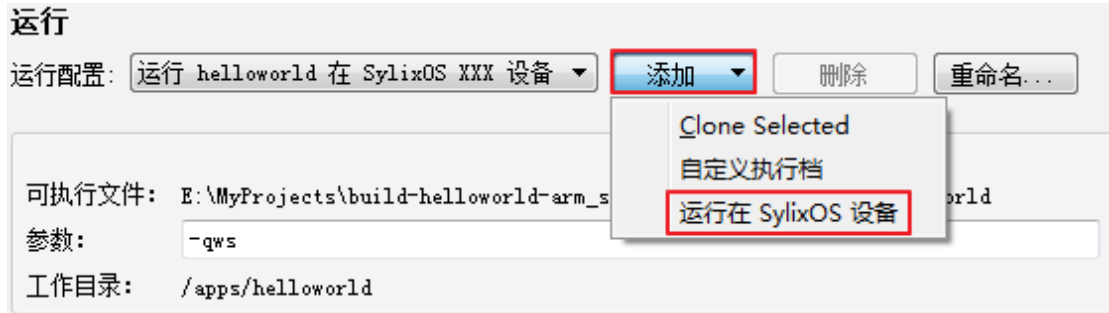


图 5.11 添加运行配置

可以看到，Qt 应用程序的工作目录是“/apps/HelloWorld”目录，参数为“-qws”，如果以后您的 Qt 应用程序有其它参数，可以在“参数”输入框中追加到“-qws”后。

需要注意的是，部署步骤除了会部署 Qt 应用程序的可执行文件外，还会部署项目文件（如 HelloWorld 项目的 HelloWorld.pro 文件）的“INSTALLS”变量指定的文件或目录，如果您的 Qt 应用程序有其它文件或目录需要部署，请增加或修改“INSTALLS”变量。“INSTALLS”变量的语法如程序清单 5.3 所示。

程序清单 5.3 “INSTALLS”变量语法

```
sylixos {
XXX.files    = 本地文件或目录名
XXX.path     = SylixOS 设备文件系统路径
INSTALLS    += XXX
}
```

假设 HelloWorld 项目有一个图片目录 imagedir 和一个配置文件 config.xml（这个图片目录和配置文件在 HelloWorld 项目的文件夹内）需要部署到 SylixOS 设备的

“/apps/HelloWorld”目录；我们需要在 HelloWorld.pro 文件中添加如程序清单 5.4 所示的内容。

程序清单 5.4 “INSTALLS”变量示例

```
sylixos {
imagedir.files    = imagedir
imagedir.path     = /apps/HelloWorld
configfile.files  = config.xml
configfile.path   = /apps/HelloWorld
INSTALLS          += imagedir configfile
}
```

注：开始行的“sylixos {”和结束行的“}”并非必须，但这里面的内容（特别是 SylixOS 设备文件系统路径）只与 SylixOS 系统平台相关，为了使 Qt 应用程序代码更好地跨平台，建议在开始行加上“sylixos {”和在结束行加上“}”。

如果需要查看有哪些文件和目录会部署到 SylixOS 设备上，可以点击“详情▼”按钮，如图 5.12 所示。

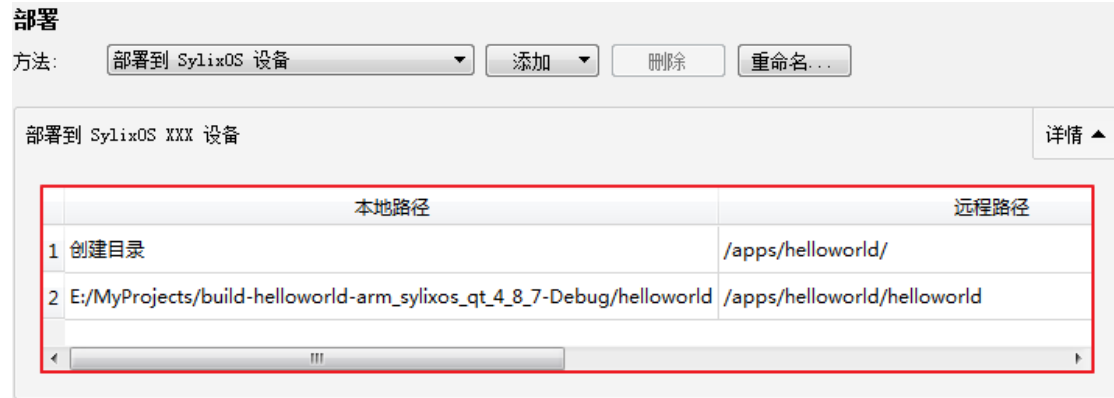
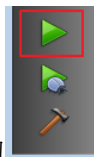


图 5.12 部署文件列表



点击运行按钮，即可开始部署及运行应用程序，如图 5.13 所示。

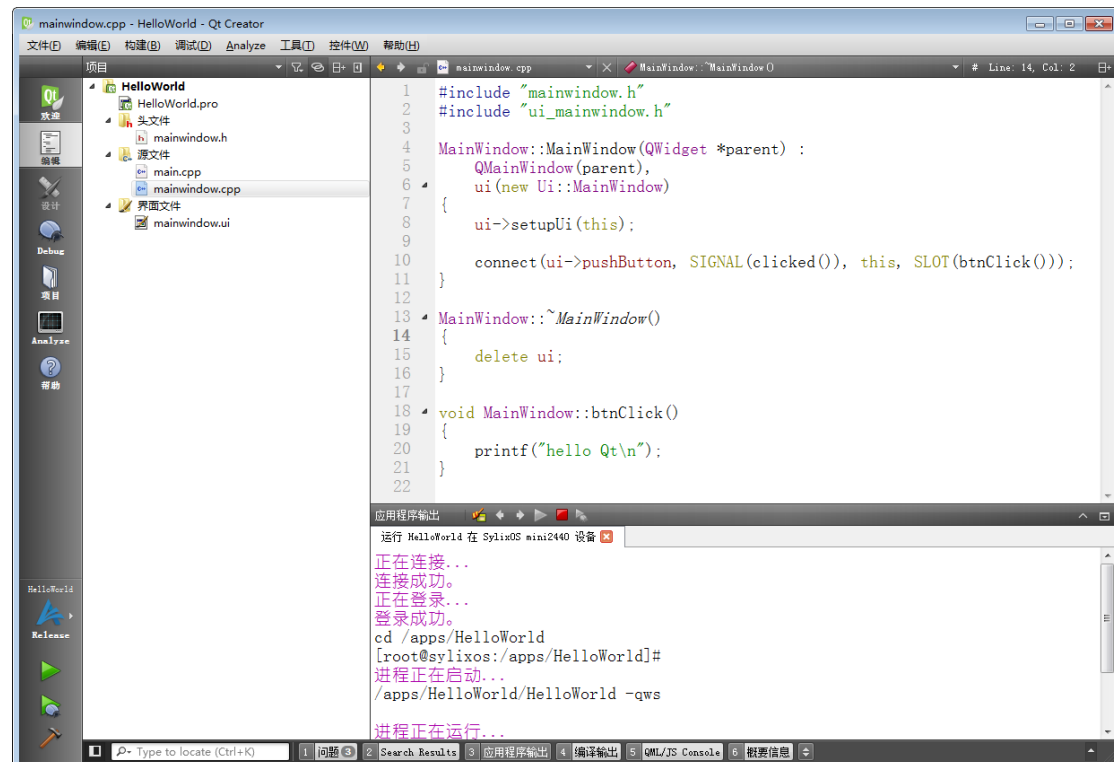


图 5.13 部署应用程序

SylixOS 设备的 LCD 屏幕将显示 HelloWorld 应用程序的主窗口，如图 5.14 所示。

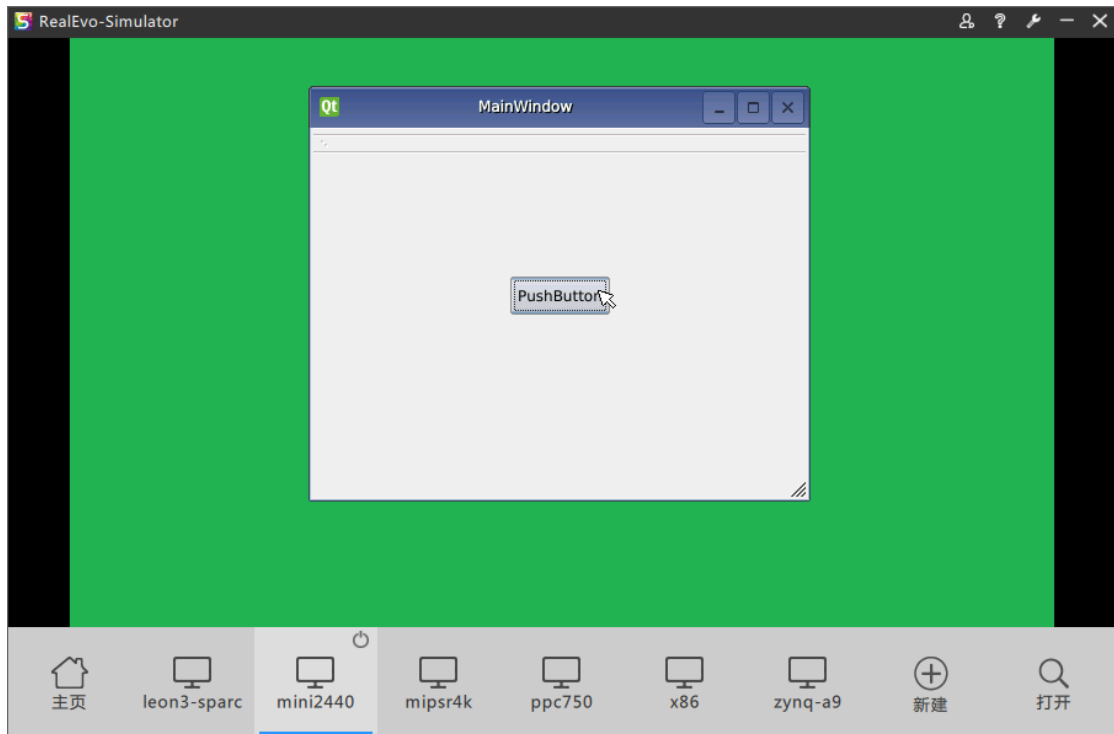


图 5.14 HelloWorld 应用程序

当 HelloWorld 应用程序里的按钮被按下时，将在 Qt Creator 的 HelloWorld 应用程序控制台输出“hello Qt”，如图 5.15 所示。



图 5.15 HelloWorld 应用程序控制台

点击 HelloWorld 应用程序控制台窗口红色的“终止”按钮，将关闭 HelloWorld 应用程序，如图 5.15 所示。

5.5 调试 Qt Widgets 应用程序


在 mainWindow.cpp 文件的 MainWindow::btnClick 函数添加一个断点（单击图 5.16 中小红球的位置即可完成断点添加），如图 5.16 所示。

```

18 void MainWindow::btnClick()
19 {
20     printf("hello Qt\n");
21 }
22

```

图 5.16 添加断点


 点击调试按钮 (键盘 F5)，即可开始部署及调试应用程序，如图 5.17 所示。

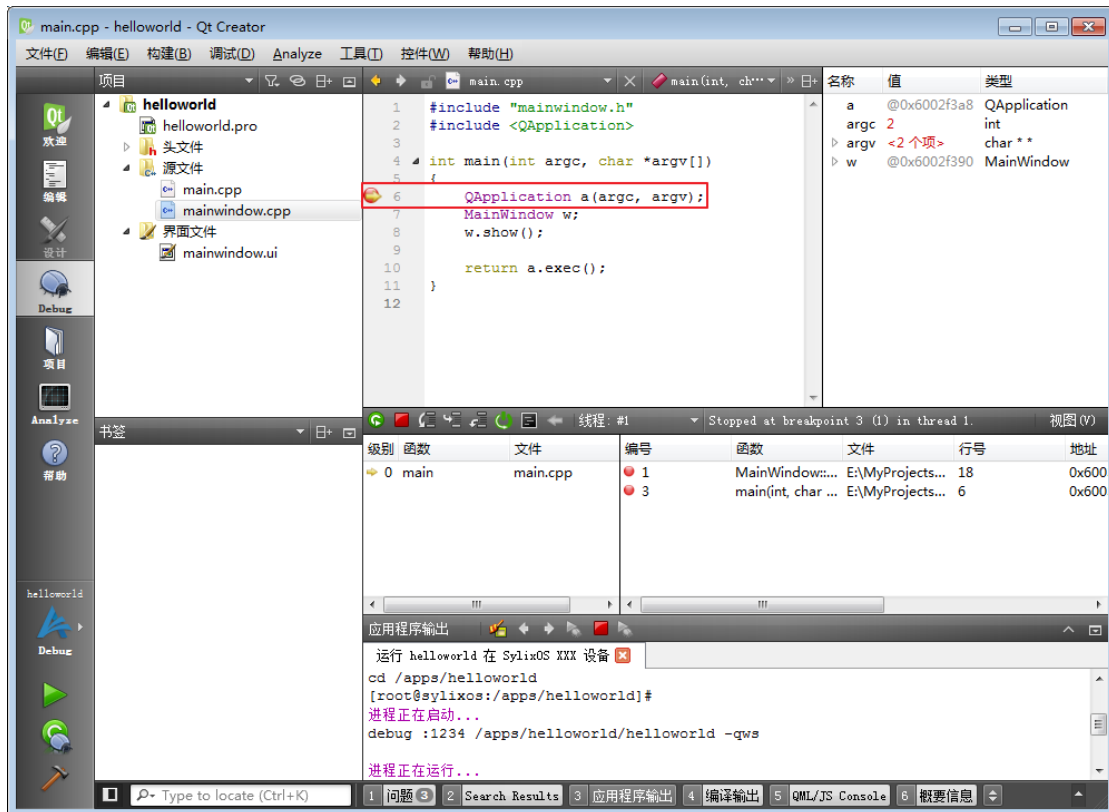


图 5.17 调试界面

程序在 main 函数入口处停了下来，这时我们可以在右侧的变量窗口查看局部变量值，在底部的调用栈窗口查看当前线程的调用栈。

键盘 F10 是单步跳过，F11 是单步进入，“Shift + F11”是单步跳出，“Ctrl+F10”是运行到光标所在行，F5 是继续全速运行。也可以点击调用栈窗口上方的调试控制工具栏


 控制调试过程。

按下键盘 F5 全速运行，SylixOS 设备的 LCD 屏幕将显示 HelloWorld 应用程序的主窗口，如图 5.14 所示。

由于我们在 MainWindow::btnClick 函数添加了一个断点，当 HelloWorld 应用程序里的按钮被按下时，程序将停在该断点，如图 5.18 所示。

```

18 void MainWindow::btnClick()
19 {
20     printf("hello Qt\n");
21 }
22

```

图 5.18 断点到达

按下键盘 F10 单步跳过，将在 Qt Creator 的 HelloWorld 应用程序控制台输出“hello Qt”，如图 5.19 所示。

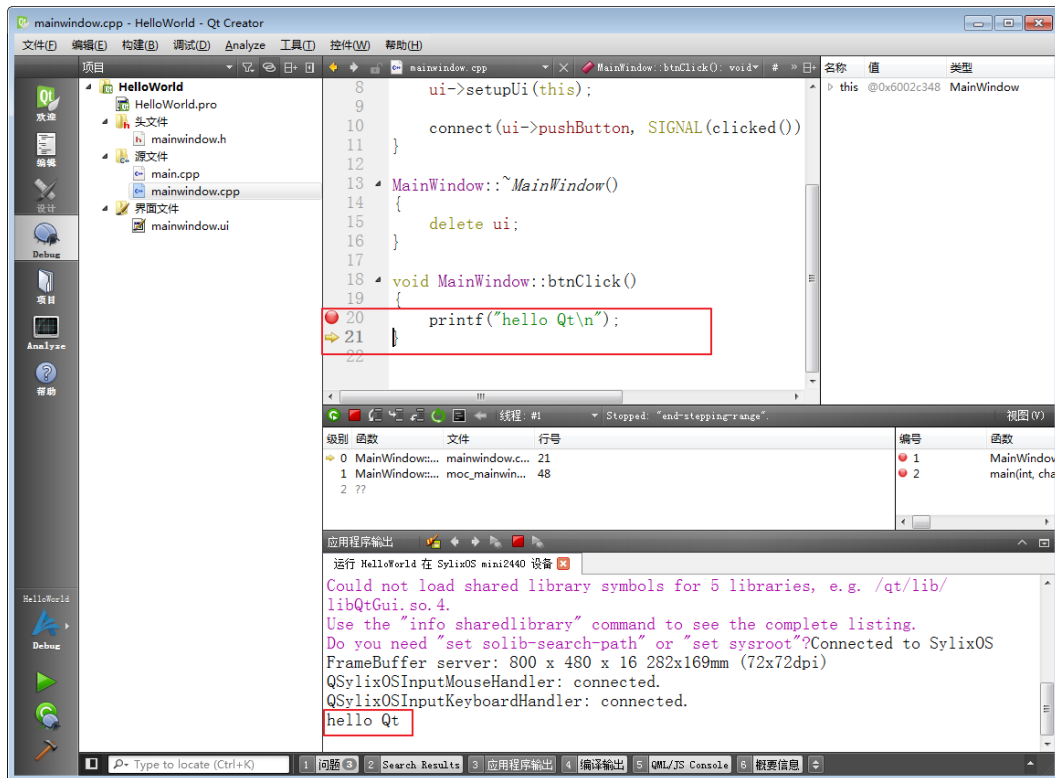


图 5.19 单步运行

点击菜单“调试 → 终止调试”将终止调试，也将关闭 HelloWorld 应用程序，如图 5.20 所示。

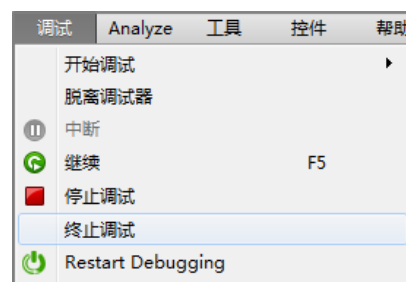


图 5.20 终止调试

注：当前的 Qt Creator 在使用“停止调试”再重新“开始调试”时会有问题，请始终使用“终止调试”的方式结束一次调试过程。

5.6 Attach 方式调试 Qt Widgets 应用程序

在开发过程中，我们可能会遇上这样一种情况：一个 Qt Widgets 应用程序已经在 SylixOS 设备上运行了一段时间，但由于该应用程序的代码编写得有些问题，导致该应用程序的运行逻辑出现了问题，我们希望能够连接该应用程序进行调试从而查找应用程序的代码中的缺陷。

RealEvo-QtSylixOS 支持以 Attach 方式调试一个正在运行的 Qt Widgets 应用程序。

为了说明 Attach 调试 Qt Widgets 应用程序的使用方式，我们假设以下几点：

- SylixOS 设备已经运行了 HelloWorld 项目的 Debug 版本的 HelloWorld 应用程序；
- HelloWorld 项目当前设置为 Debug 版本；
- HelloWorld 项目所使用的构建套件（Kit）的 SylixOS 设备就是需要被调试的 SylixOS 设备；
- RealEvo-QtSylixOS 也已经打开了 HelloWorld 项目，并将 HelloWorld 项目设置为当前的活动项目，如图 5.21 所示。

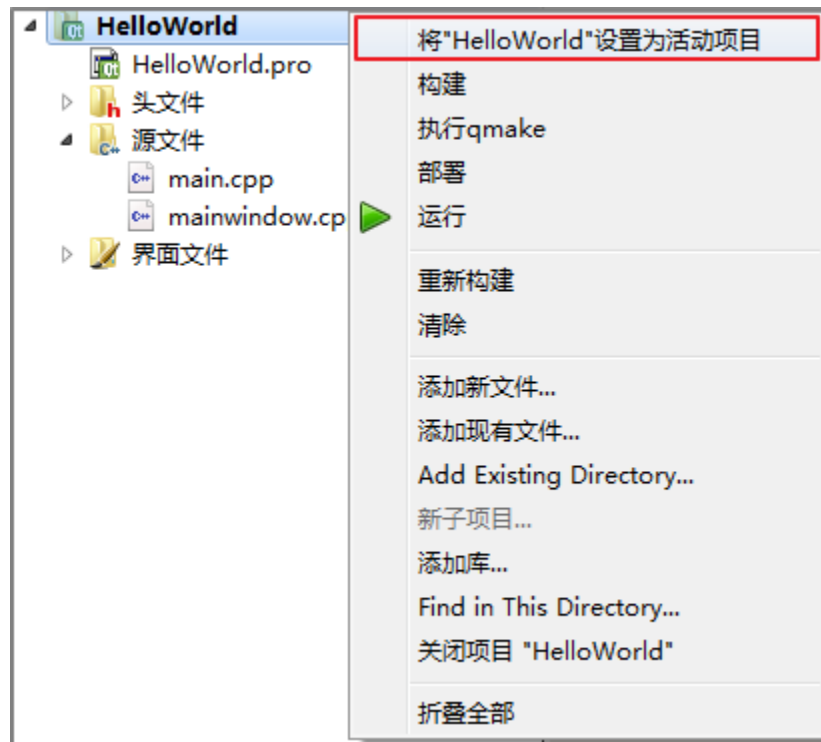


图 5.21 设置活动项目

点击菜单“调试→开始调试→Attach 到远程的 SylixOS 应用程序...”菜单，如图 5.22 所示。

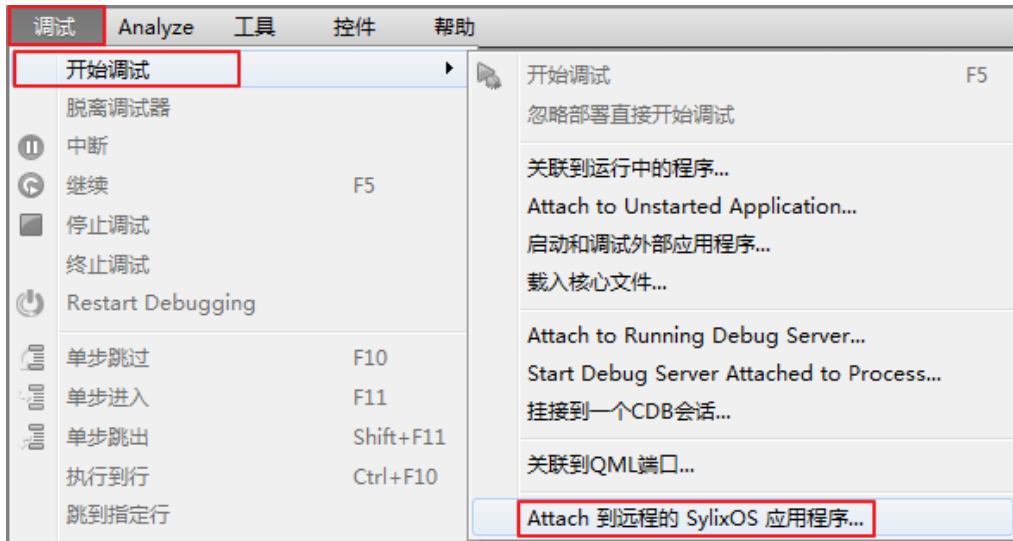


图 5.22 Attach 应用程序

将弹出进程列表对话框，如图 5.23 所示。

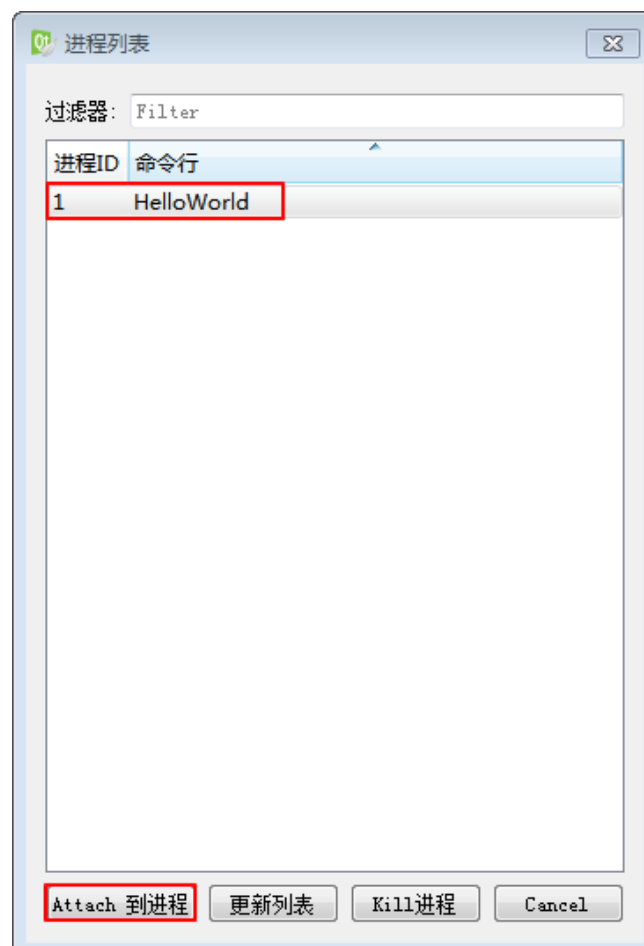


图 5.23 进程列表对话框

当前在 SylixOS 设备上运行的进程会显示在进程列表中，选中目标进程“HelloWorld”，然后点击“Attach 到进程”按钮，将开始调试，如图 5.24 所示。

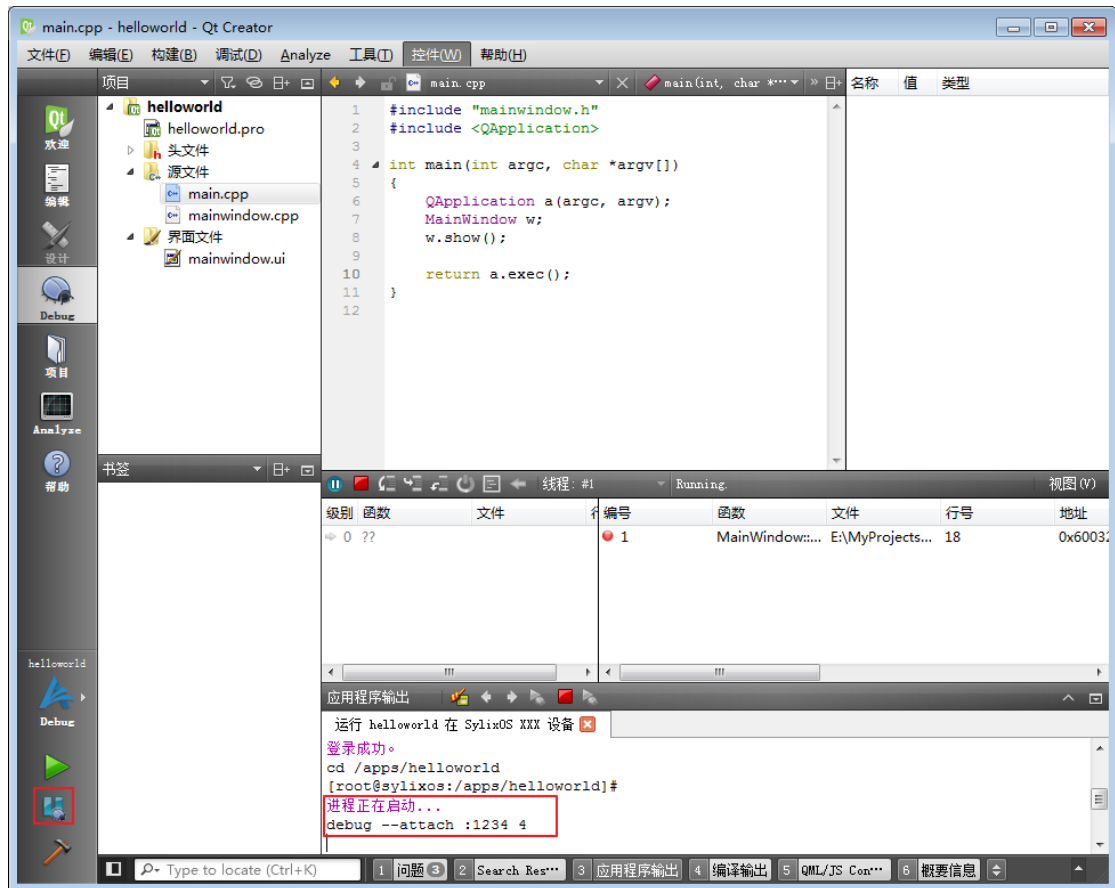


图 5.24 开始调试

调试的方法与 5.5 章节一致，这里不再详述，不同的是：点击菜单“调试→终止调试”将终止调试，但并不关闭 HelloWorld 应用程序，HelloWorld 应用程序将继续在 SylixOS 设备上运行。

5.7 Qt Widgets 应用程序加入 Qwt

如果在 Qt Widgets 应用程序的界面文件（如 HelloWorld 项目的 mainwindow.ui 文件）上使用了如图 5.25 所示的 Qwt 窗口部件 widgets。

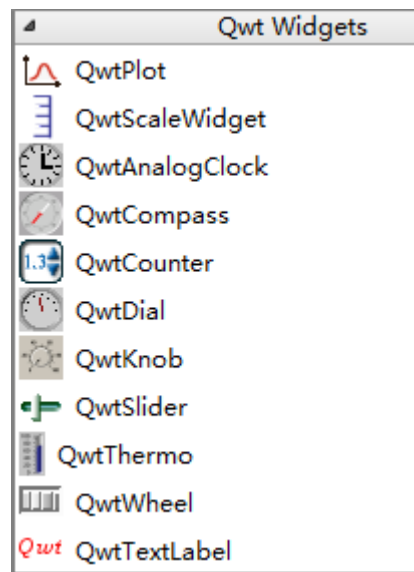


图 5.25 Qwt 窗口部件

则需要修改项目文件（如 HelloWorld 项目的 HelloWorld.pro 文件），加入如程序清单 5.5 所示的内容。

程序清单 5.5 链接参数

```
sylixos {  
LIBS      += -lqwt  
INCLUDEPATH += $(QTDIR)/../qwt  
}
```

其中，“LIBS”变量指定了 Qt 应用程序的链接参数：-lxxx 表示链接 libxxx.so 共享库，-Lyyy 表示在 yyy 目录里查找共享库。“INCLUDEPATH”变量指定了头文件搜索路径。

注：如果您的 Qt 应用程序还需要链接其它的共享库或静态库，可根据上面规则的说明进行修改。

第6章 开发 Qt Quick 应用程序

6.1 新建 Qt Quick 应用程序

点击菜单“文件 → 新建文件或项目..”，将弹出新建对话框，如图 6.1 所示。

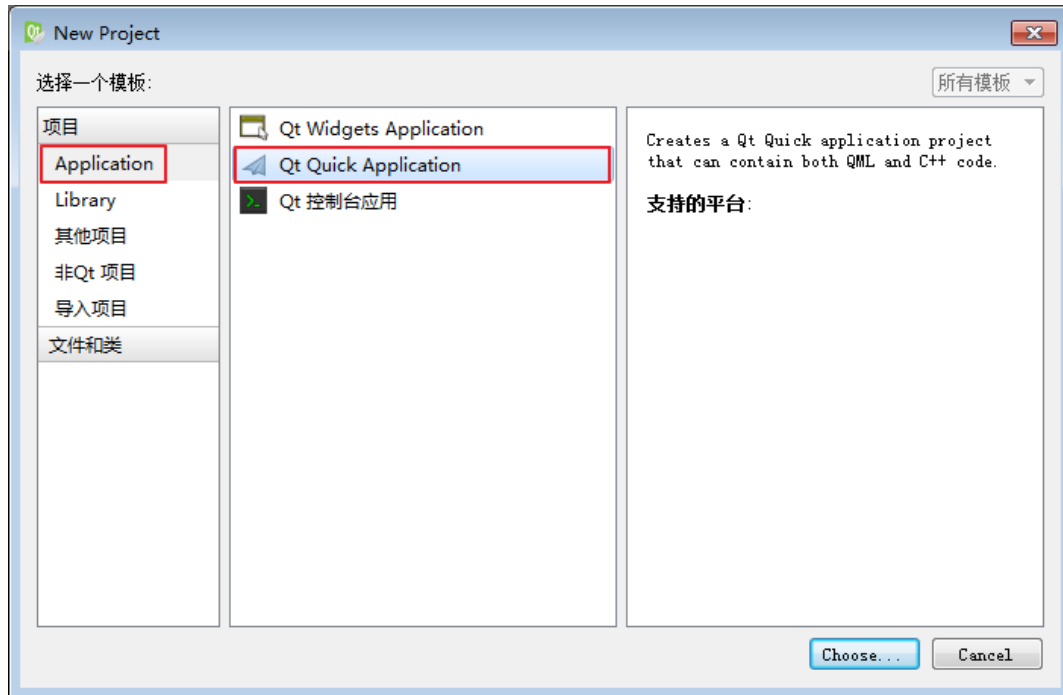


图 6.1 新建对话框

选择项目类型“Application”下的“Qt Quick Application”模板，然后点击“Choose..”按钮进入“Qt Quick Application”向导，如图 6.2 所示。

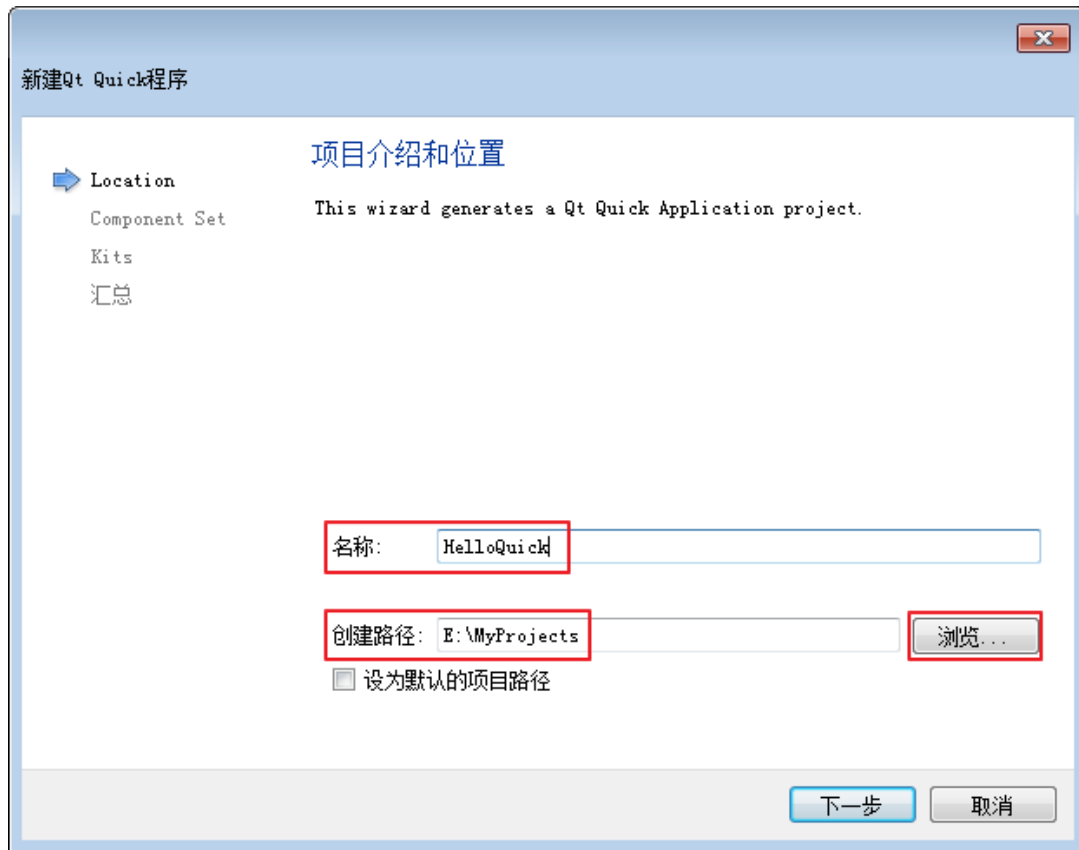


图 6.2 “Qt Quick Application”向导

输入项目名称为“HelloQuick”，创建路径通过点击“浏览...”按钮选择一个合适的项目存放位置，如“E:\MyProjects”目录，然后点击“下一步”按钮进入下一步，如图 6.3 所示。

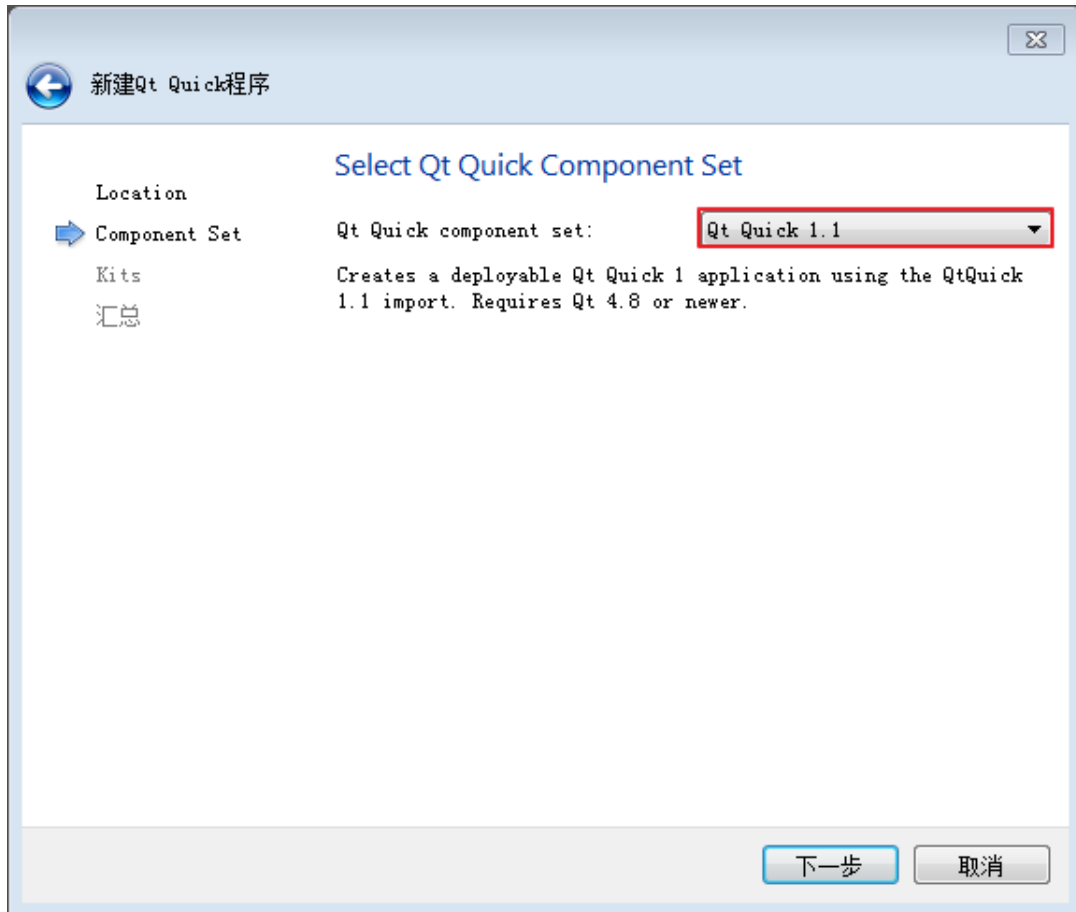


图 6.3 Qt Quick 组件选择

选择“Qt Quick component set”为“Qt Quick 1.1”，然后点击“下一步”按钮进入下一步，如图 6.4 所示。

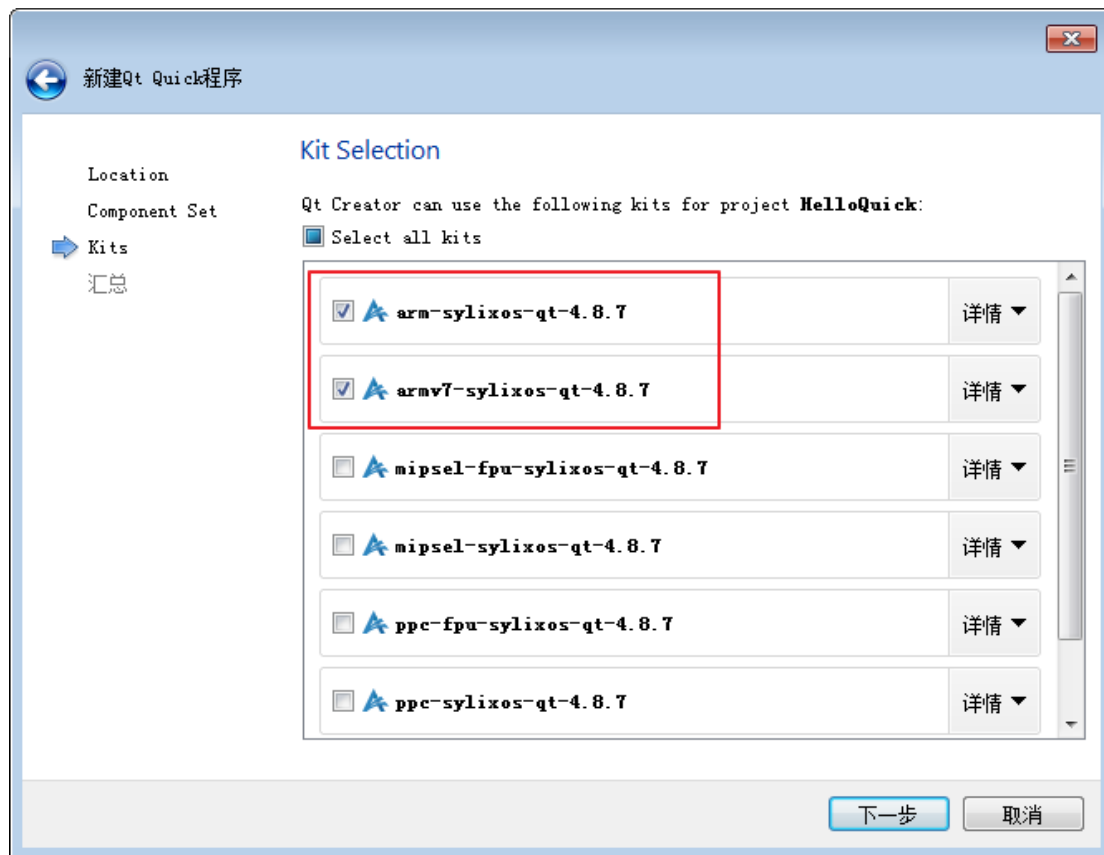


图 6.4 构建套件 (Kit) 选择

构建套件 (Kit) 可以选择 “arm-sylixos-qt-4.8.7” 或 “armv7-sylixos-qt-4.8.7”，或者全选，然后点击 “下一步” 按钮按默认流程完成项目创建，创建完毕后，HelloQuick 项目如图 6.5 所示。

注：如果看不到任何 SylixOS 的构建套件 (Kit)，一般是因为上面的 “Qt Quick component set” 没有选择为 “Qt Quick 1.1”。

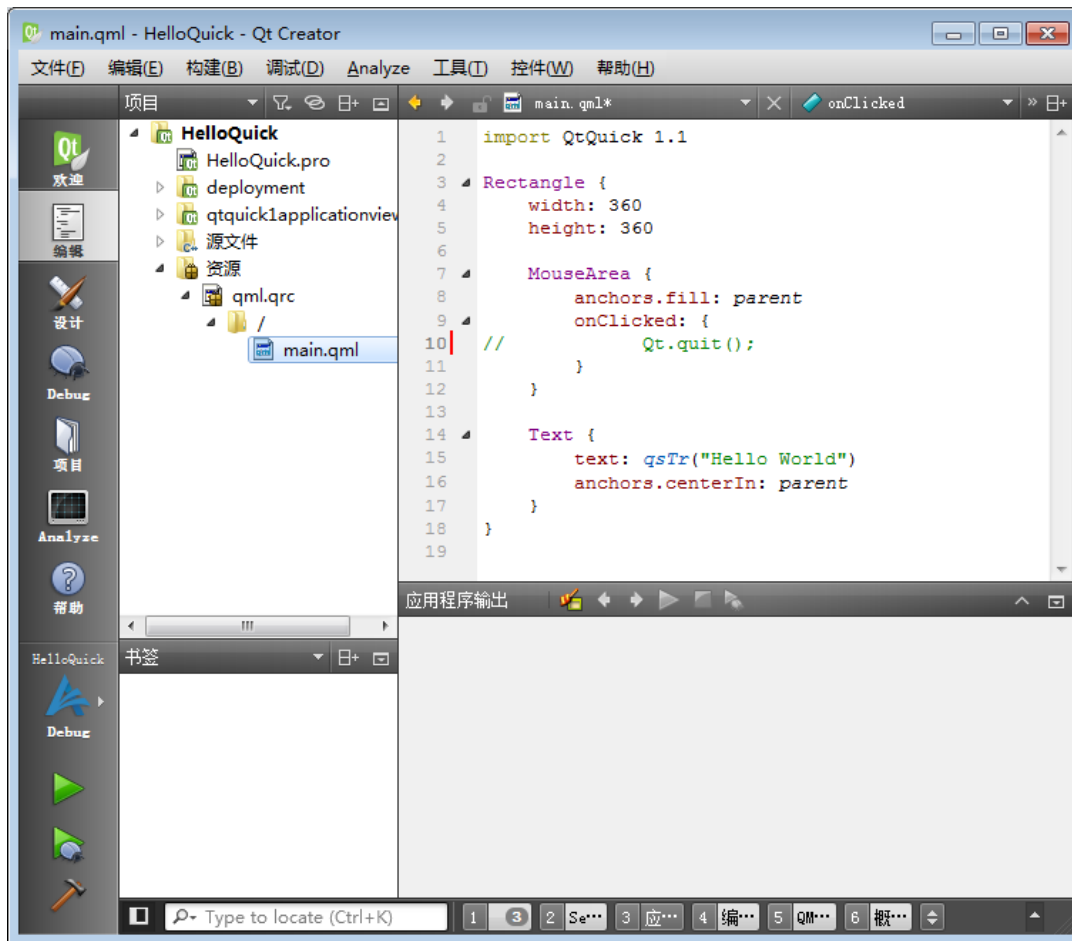


图 6.5 HelloQuick 项目

6.2 修改 Qt Quick 应用程序

为了避免点击 HelloQuick 应用程序主界面时 HelloQuick 应用程序主动退出的问题，需要修改资源下的 main.qml 文件，注释掉“onClicked”处理内的“Qt.quit();”语句，如图 6.5 所示。

Qt Quick 应用程序的编译、运行、调试方法与 Qt Widgets 应用程序开发的一致，您可以分别参考 5.3 章节、5.4 章节、5.5 章节，这里不再详述，下面讲解 Qt Quick 应用程序的分析方法。

6.3 分析 Qt Quick 应用程序

点击菜单“分析 → QML Profiler”，如图 6.6 所示，即可开始部署及分析应用程序，如图 6.7 所示。



图 6.6 QML Profiler 菜单

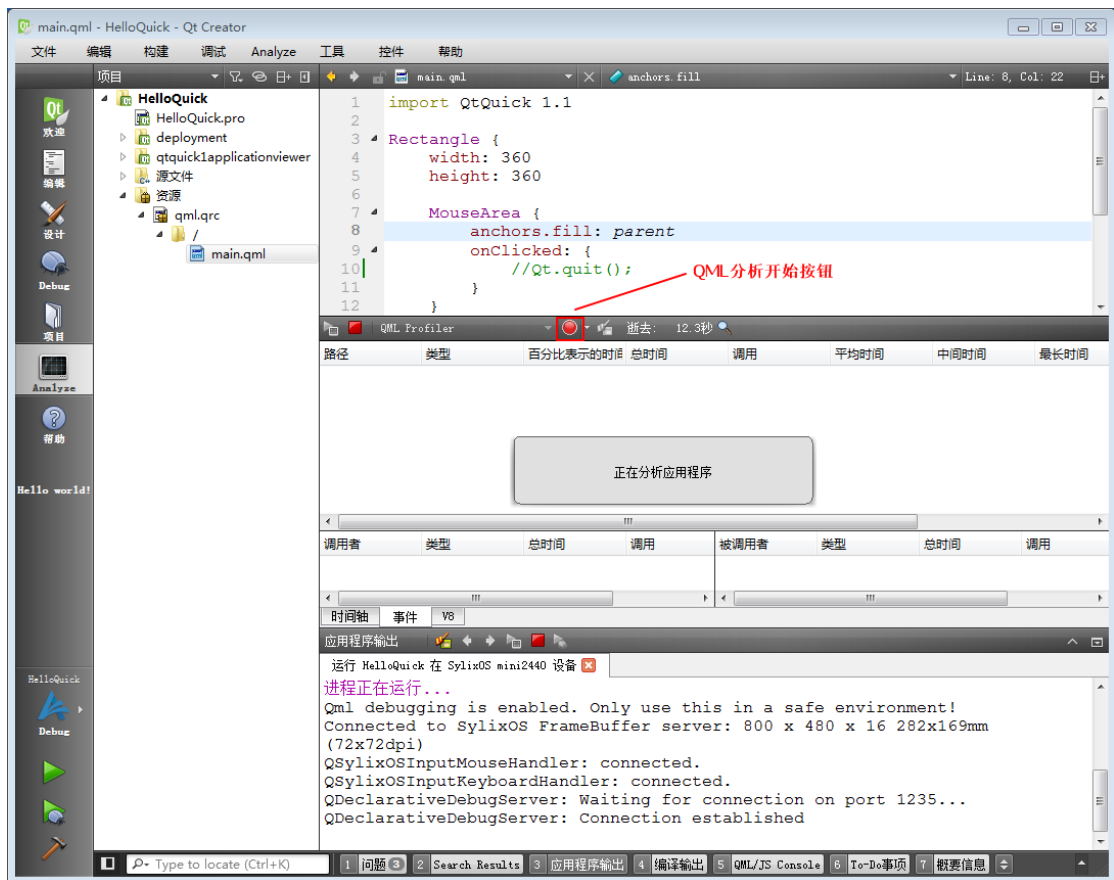


图 6.7 QML 分析界面

应用程序运行起来后，如果 QML 分析器未开始工作，点击 QML 分析器的“启用”按钮，QML 分析器开始工作。

在 SylixOS 设备的 LCD 屏幕上操作 HelloQuick 应用程序，经过一段时间后，如 10 秒后，再次点击 QML 分析器的“启用”按钮停止分析，QML 分析器便会将过去 10 秒内 HelloQuick 应用程序的耗时数据和发生的事件以图形化的形式显示出来，如图 6.8 所示。

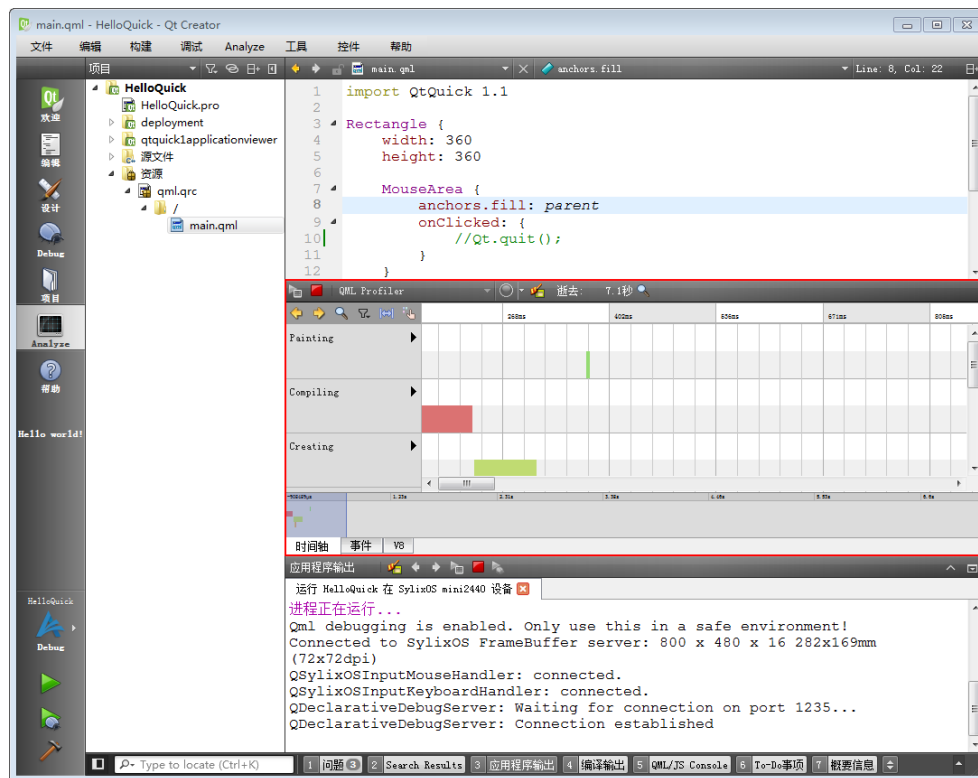


图 6.8 QML 分析结果

点击 HelloQuick 应用程序控制台窗口红色的“终止”按钮，将关闭 HelloQuick 应用程序，如图 6.9 所示。

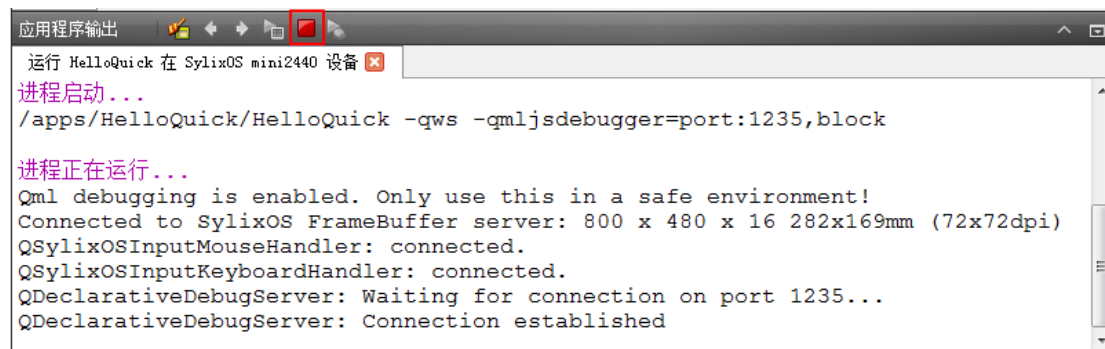


图 6.9 关闭按钮

第7章 开发其它类型的项目

7.1 开发 Qt 控制台应用程序

RealEvo-QtSylixOS 支持开发 Qt 控制台应用程序，Qt 控制台应用程序与 Qt Widgets 应用程序的区别在于：Qt 控制台应用程序不能使用 QtGui 组件（如：Qt 窗口部件 QWidget、Qt 对话框 QDialog、Qt 主窗口 QMainWindow），但依然可以使用 Qt 核心 QtCore、Qt 网络 QtNetwork、Qt 数据库 QSql 等组件。

Qt 控制台应用程序的新建流程如下：

- 点击菜单“文件 → 新建文件或项目..”，将弹出新建对话框，选择项目类型“Application”下的“Qt 控制台应用”模板，如图 7.1 所示。
- 在后续的“Kit Selection”对话框中选择 SylixOS 的构建套件（Kit）（如“arm-sylixos-qt-4.8.7”和“armv7-sylixos-qt-4.8.7”）。

Qt 控制台应用程序的编译、运行、调试方法与 Qt Widgets 应用程序开发的一致，详见 5.3 章节、5.4 章节、5.5 章节。

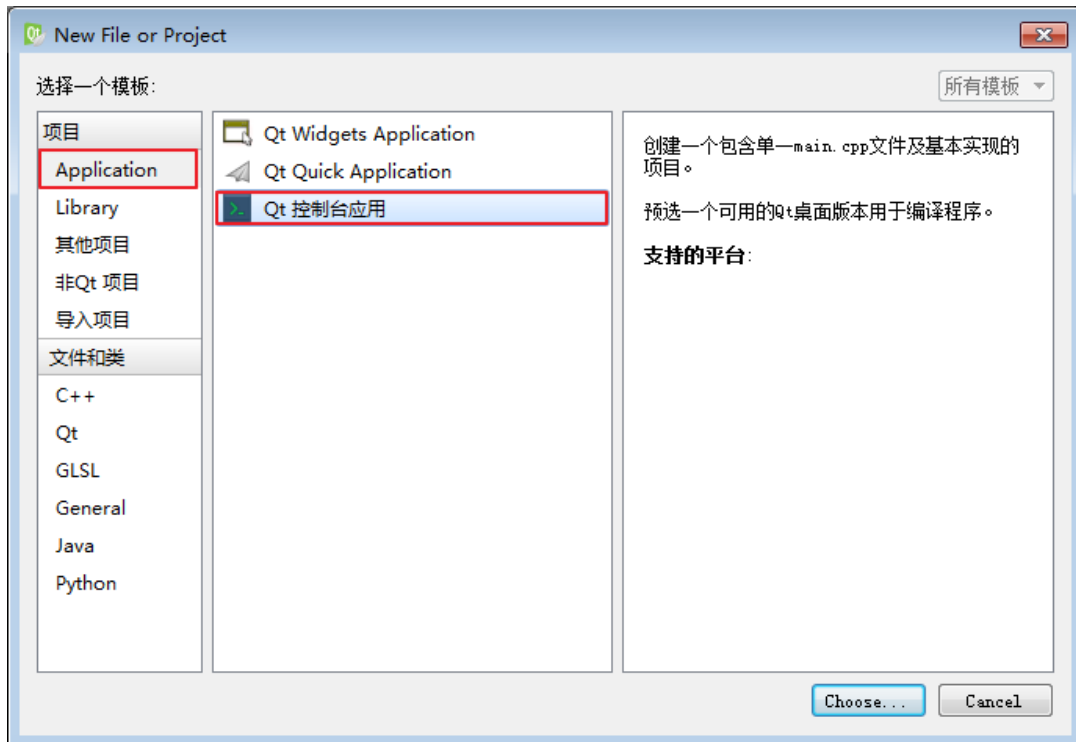


图 7.1 Qt 控制台应用

7.2 开发 SylixOS 纯 C 语言项目

RealEvo-QtSylixOS 支持开发 SylixOS 纯 C 语言项目，SylixOS 纯 C 语言项目使用 qmake 生成 Makefile，所以免于编写 Makefile，SylixOS 纯 C 语言项目不能使用任何的 Qt 组件。

SylixOS 纯 C 语言项目的新建流程如下：

- 点击菜单“文件 → 新建文件或项目..”，将弹出新建对话框，选择项目类型“非 Qt 项目”下的“纯 C 语言项目”模板，如图 7.2 所示。

- 在后续的“Kit Selection”对话框中选择 SylixOS 的构建套件（Kit）（如“arm-sylixos-qt-4.8.7”和“armv7-sylixos-qt-4.8.7”）。

SylixOS 纯 C 语言项目的编译、运行、调试方法与 Qt Widgets 应用程序开发的一致，详细见 5.3 章节、5.4 章节、5.5 章节。

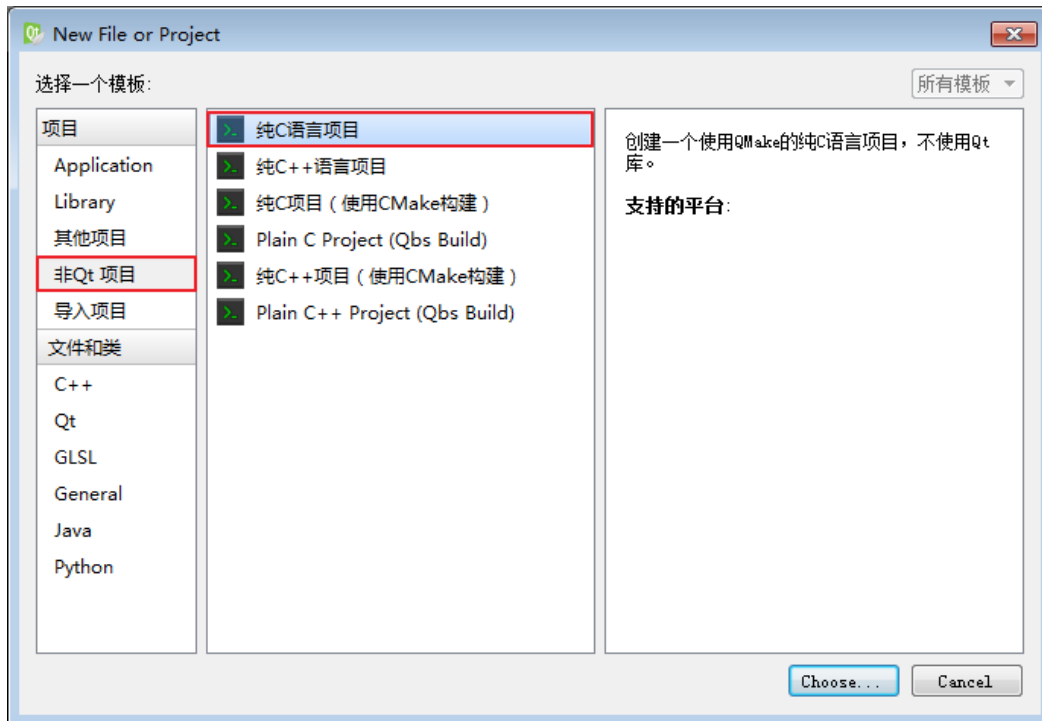


图 7.2 纯 C 语言项目

7.3 开发 SylixOS 纯 C++语言项目

RealEvo-QtSylixOS 支持开发 SylixOS 纯 C++语言项目，SylixOS 纯 C++语言项目使用 qmake 生成 Makefile，所以免于编写 Makefile，SylixOS 纯 C++语言项目不能使用任何的 Qt 组件。

SylixOS 纯 C++语言项目的新建流程如下：

- 点击菜单“文件 → 新建文件或项目..”，将弹出新建对话框，选择项目类型“非 Qt 项目”下的“纯 C++语言项目”模板，如图 7.3 所示。
- 在后续的“Kit Selection”对话框中选择 SylixOS 的构建套件（Kit）（如“arm-sylixos-qt-4.8.7”和“armv7-sylixos-qt-4.8.7”）。

SylixOS 纯 C++语言项目的编译、运行、调试方法与 Qt Widgets 应用程序开发的一致，详细见 5.3 章节、5.4 章节、5.5 章节。

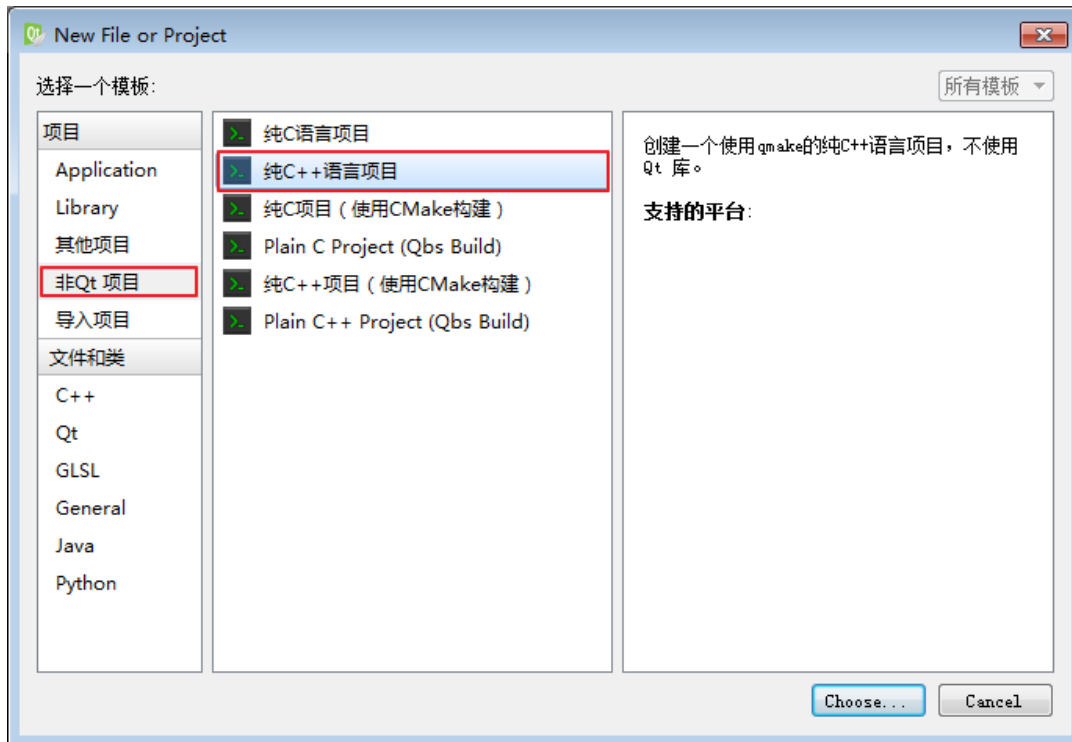


图 7.3 纯 C++语言项目

7.4 开发 SylixOS 共享库项目

RealEvo-QtSylixOS 支持开发 SylixOS 共享库项目，SylixOS 共享库项目使用 qmake 生成 Makefile，所以免于编写 Makefile，SylixOS 共享库项目可以使用任何的 Qt 组件。

SylixOS 共享库项目的新建流程如下：

- 点击菜单“文件 → 新建文件或项目..”，将弹出新建对话框，选择项目类型“Library”下的“C++库”模板，如图 7.4 所示。
- 在后续的项目介绍和位置对话框中的类型组合框中选择“共享库”。
- 在后续的“Kit Selection”对话框中选择 SylixOS 的构建套件（Kit）（如“arm-sylixos-qt-4.8.7”和“armv7-sylixos-qt-4.8.7”）。

SylixOS 共享库项目的编译方法与 Qt Widgets 应用程序开发的一致，详细见 5.3 章节。SylixOS 共享库项目的部署目录为“/usr/lib”，点击“运行”按钮将部署 SylixOS 共享库到 SylixOS 设备的“/usr/lib”目录。

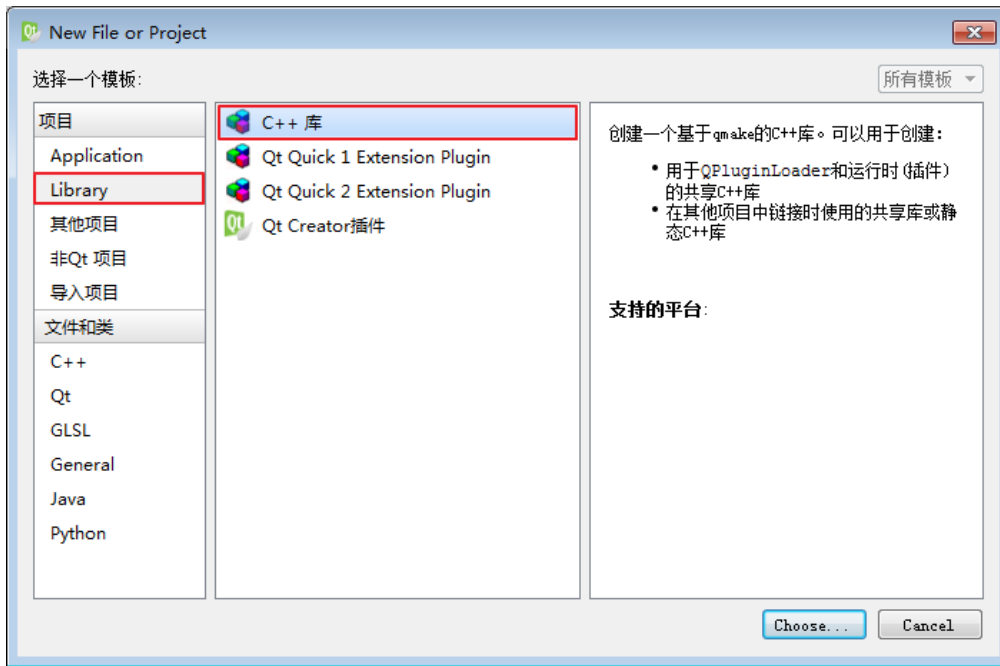


图 7.4 C++库

7.5 开发 SylixOS 静态库项目

RealEvo-QtSylixOS 支持开发 SylixOS 静态库项目，SylixOS 静态库项目使用 qmake 生成 Makefile，所以免于编写 Makefile，SylixOS 静态库项目可以使用任何的 Qt 组件。

SylixOS 静态库项目的新建流程如下：

- 点击菜单“文件 → 新建文件或项目..”，将弹出新建对话框，选择项目类型“Library”下的“C++库”模板，如图 7.4 所示。
- 在后续的项目介绍和位置对话框中的类型组合框中选择“静态库”。
- 在后续的“Kit Selection”对话框中选择 SylixOS 的构建套件（Kit）（如“arm-sylixos-qt-4.8.7”和“armv7-sylixos-qt-4.8.7”）。

SylixOS 静态库项目的编译方法与 Qt Widgets 应用程序开发的一致，详细见 5.3 章节。SylixOS 静态库项目不支持部署功能。

第8章 FAQ

8.1 如何使用 QtSerialPort 类库

需要修改项目文件（如 HelloWorld 项目的 HelloWorld.pro 文件），加入如程序清单 8.1 所示的内容。

程序清单 8.1 QtSerialPort 类库使用

```
sylixos {  
LIBS      += -lQtSerialPort  
INCLUDEPATH += $(QTDIR)/../QtSerialPort  
}
```

注：关于 QtSerialPort 类的进一步使用，请参考 Qt 官方的 wiki: <http://wiki.qt.io/QtSerialPort>

8.2 如何使用 VNC

如果需要使用 VNC，请在 SylixOS 的 Shell 中输入如程序清单 8.2 所示的命令以设置并保存 “QWS_DISPLAY” 环境变量。

程序清单 8.2 使用 VNC

```
QWS_DISPLAY=VNC:sylixosfb:$DISPLAY  
varsave
```

然后重新启动 Qt 应用程序，最后在如图 4.1 所示的“设备选项页面”中点击“运行 VNC 客户端...”按钮。

8.3 如何旋转屏幕

如果需要旋转屏幕 90 度，请在 SylixOS 的 Shell 中输入如程序清单 8.3 所示的命令以设置并保存 “QWS_DISPLAY” 环境变量。

程序清单 8.3 旋转屏幕

```
QWS_DISPLAY=Transformed:Rot90  
varsave
```

然后重新启动 Qt 应用程序。

如果需要同时旋转屏幕 90 度和使用 VNC，请在 SylixOS 的 Shell 中输入如程序清单 8.4 所示的命令以设置并保存 “QWS_DISPLAY” 环境变量。

程序清单 8.4 旋转屏幕和使用 VNC

```
QWS_DISPLAY=Transformed:Rot90:VNC:sylixosfb:$DISPLAY  
varsave
```

然后重新启动 Qt 应用程序。

8.4 出现部署失败的情况

如果在使用 Qt 进行部署时，出现部署失败的情况，需要进行以下几步的确认：

1. 确认图 3.2 中是否正确设置了 SylixOS 设备（对应正确的编译工具链）；

2. 点击图 3.6 中的“Test”按钮确认网络是否连通（如图 3.5 所示）；
3. 确认图 3.8 中是否已经成功设置了对于设备的 SylixOS BASE 路径；
4. 确认图 4.2 中是否正确选择了“Debug 版本”选项（需要根据实际编译的 SylixOS BASE 进行正确选择）；
5. 确认图 4.2 中是否正确选择了“不支持符号链接”选项（在 SylixOS 系统中 TpsFs 文件系统、Yaffs 文件系统支持符号链接、Fat32 文件系统不支持符号链接）。