作业7 QSerialPort串口开发及应用（串口发送）

布置时间：5月12日

收集时间：5月18日24：00

提交方式：姓名\_学号\_作业7，如：平毅\_2009060707\_作业7.doc

内容：在Deepin操作系统（可在Windows版本上先进行开发）Qt Creator开发环境中设计一个QMainWindow类型的窗体。设置窗口大小为400X200分辨率，主窗口（标注应用名称为：“网络Socket通信”，如图2所示），放置信息发送框（LineEdit），网口信息发送按钮（PushButton）以及串口信息接收框（TextEdit），按照参考样例布局。编写程序实现点击发送按钮，应用可发送发送框内所含字符串信息（尝试发送中文字符），同时底部的文本框能够接收到来自服务器的数据流（尝试接收中文字符），服务器端通过网络调试工具（Hercules Setup Utility ）及串口助手查看(网络IP地址和端口号需统一，设置为LocalHost及8080)，可参考图2应用示例。



图1 界面布局图（完整版）



图2 网络socket收发数据应用演示（Qt为Client， HW为Server）

解答：

1. 新建Qt工程文件

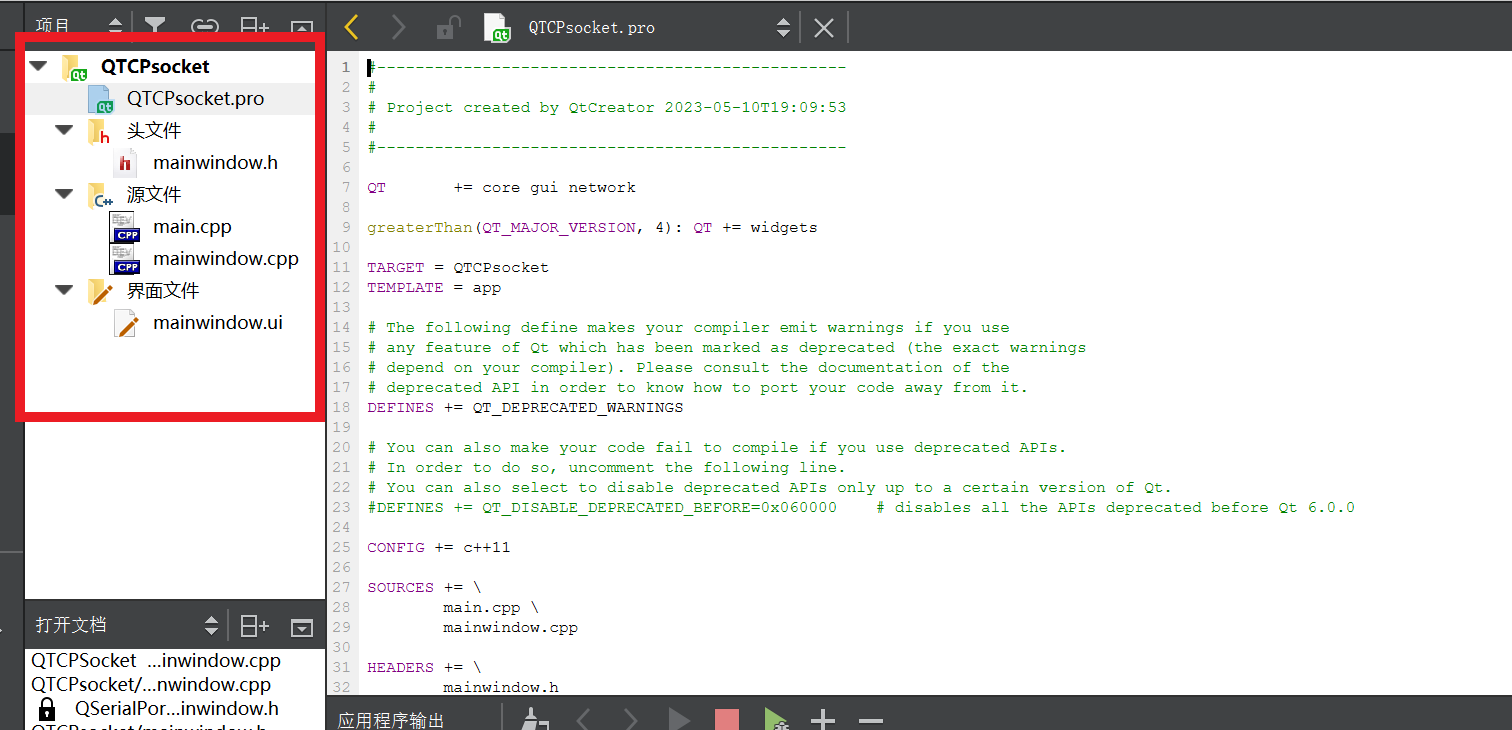


图1 Qt工程文件框架图

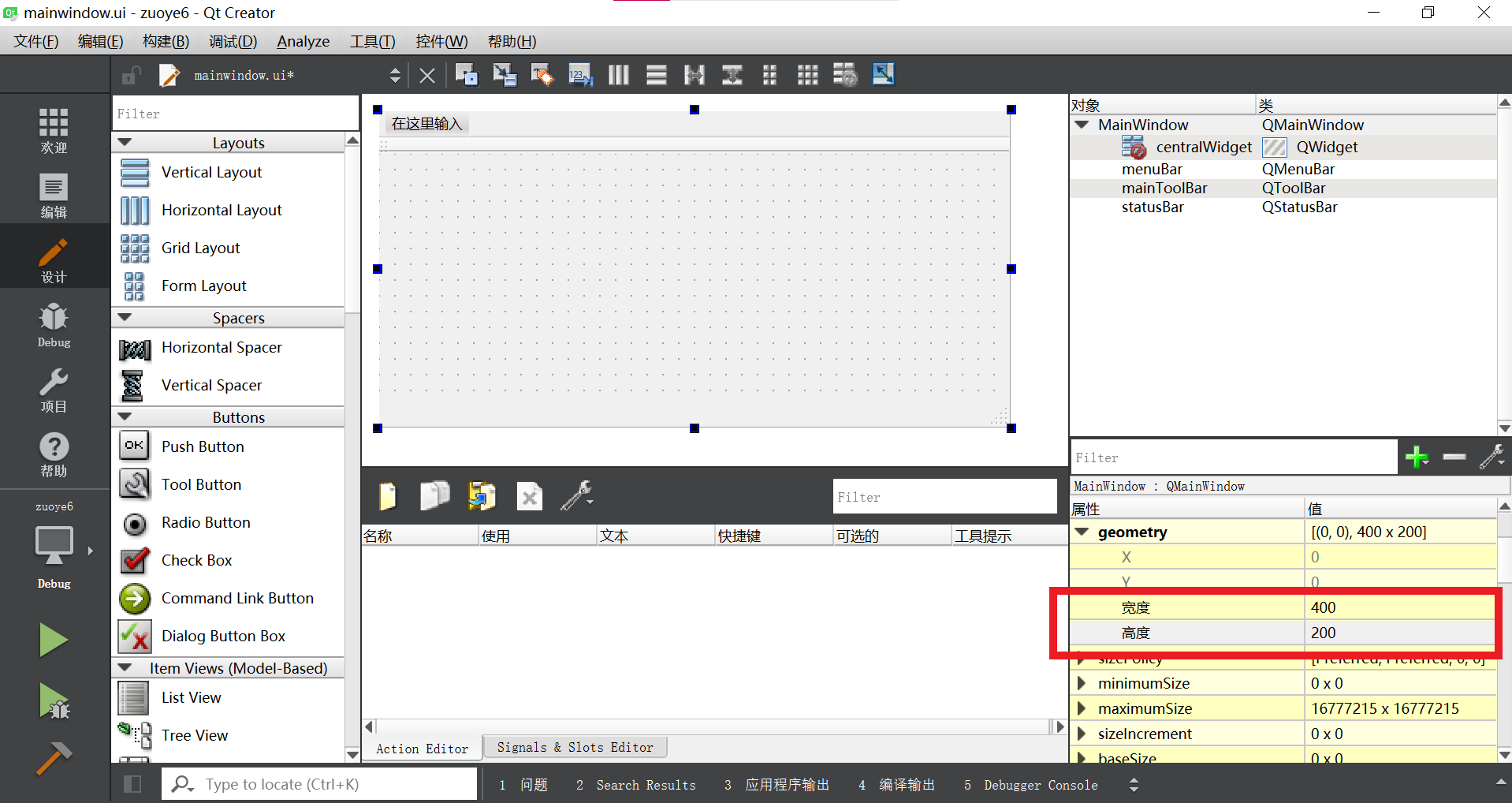
1. mainwindow.ui文件中的QMainWindow对象进行分辨率（geometry参数）的设置

图2 QMainWindow几何参数设置

1. 在mainwindow界面中添加TextEdit、LineEdit和PushButton控件，

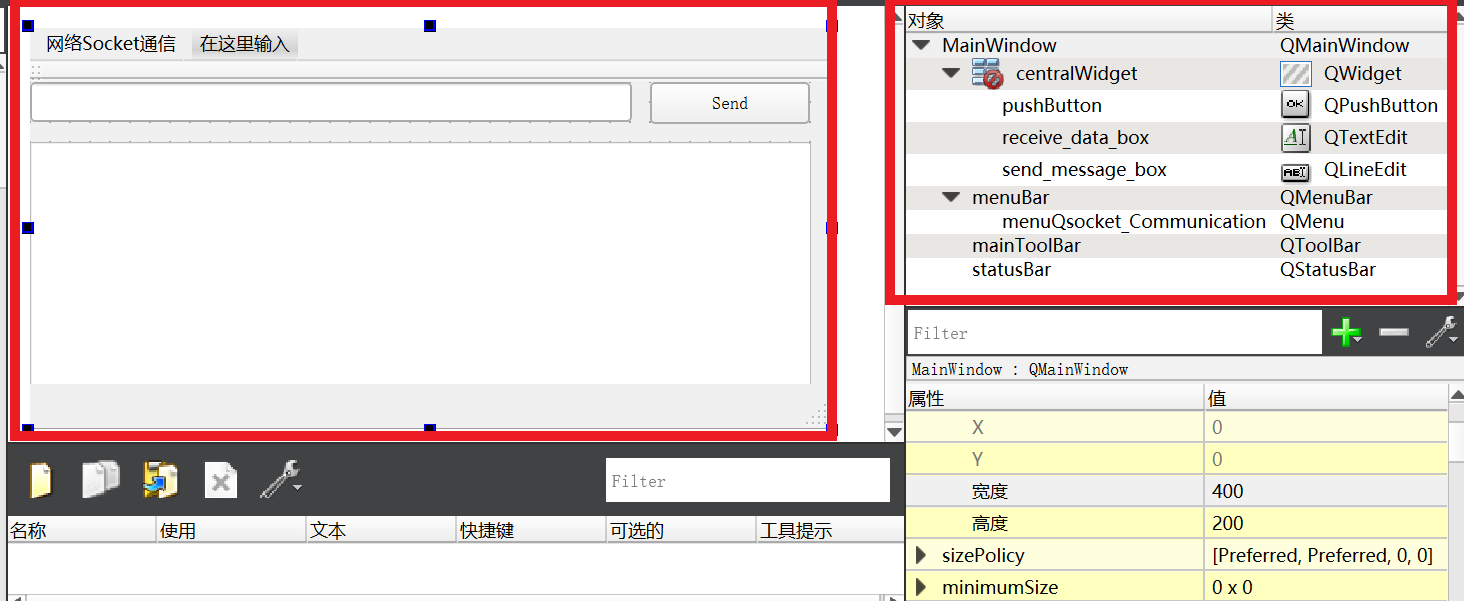


图3 添加控件并修改文字

1. 插入头文件声明（在 mainwindow.h文件中插入private slots：）

private slots:

void on\_pushButton\_clicked();

void read\_data();

1. 添加源文件函数。

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

TCPSocket = new QTcpSocket();

TCPSocket->*connectToHost*("127.0.0.1",8080);

//TCPSocket->connectToHost(QHostAddress::LocalHost,8080);

TCPSocket->*open*(QIODevice::ReadWrite);

if(TCPSocket->isOpen())

{

QMessageBox::information(this,"Qt With LSQ","Connected To The Server.");

}

else

{

QMessageBox::information(this,"Qt With LSQ","Not Connected To The Server.");

}

connect(TCPSocket,SIGNAL(readyRead()),this,SLOT(read\_data()));

}

MainWindow::~*MainWindow*()

{

delete ui;

}

void MainWindow::read\_data(){

if(TCPSocket)

{

if(TCPSocket->isOpen())

{

QByteArray a=TCPSocket->readAll();

QString Message=QString::fromLocal8Bit(a);

ui->receive\_data\_box->append(Message);

}

}

}

void MainWindow::on\_pushButton\_clicked()

{

if(TCPSocket)

{

if(TCPSocket->isOpen())

{

QString WriteData = ui->send\_message\_box->text().append(char(10));

TCPSocket->write(WriteData.toStdString().c\_str());

}

else

{

QMessageBox::information(this,"Qt With LSQ","Error:"+TCPSocket->errorString());

}

}

else

{

QMessageBox::information(this,"Qt With LSQ","Error:"+TCPSocket->errorString());

}

}

1. 项目测试

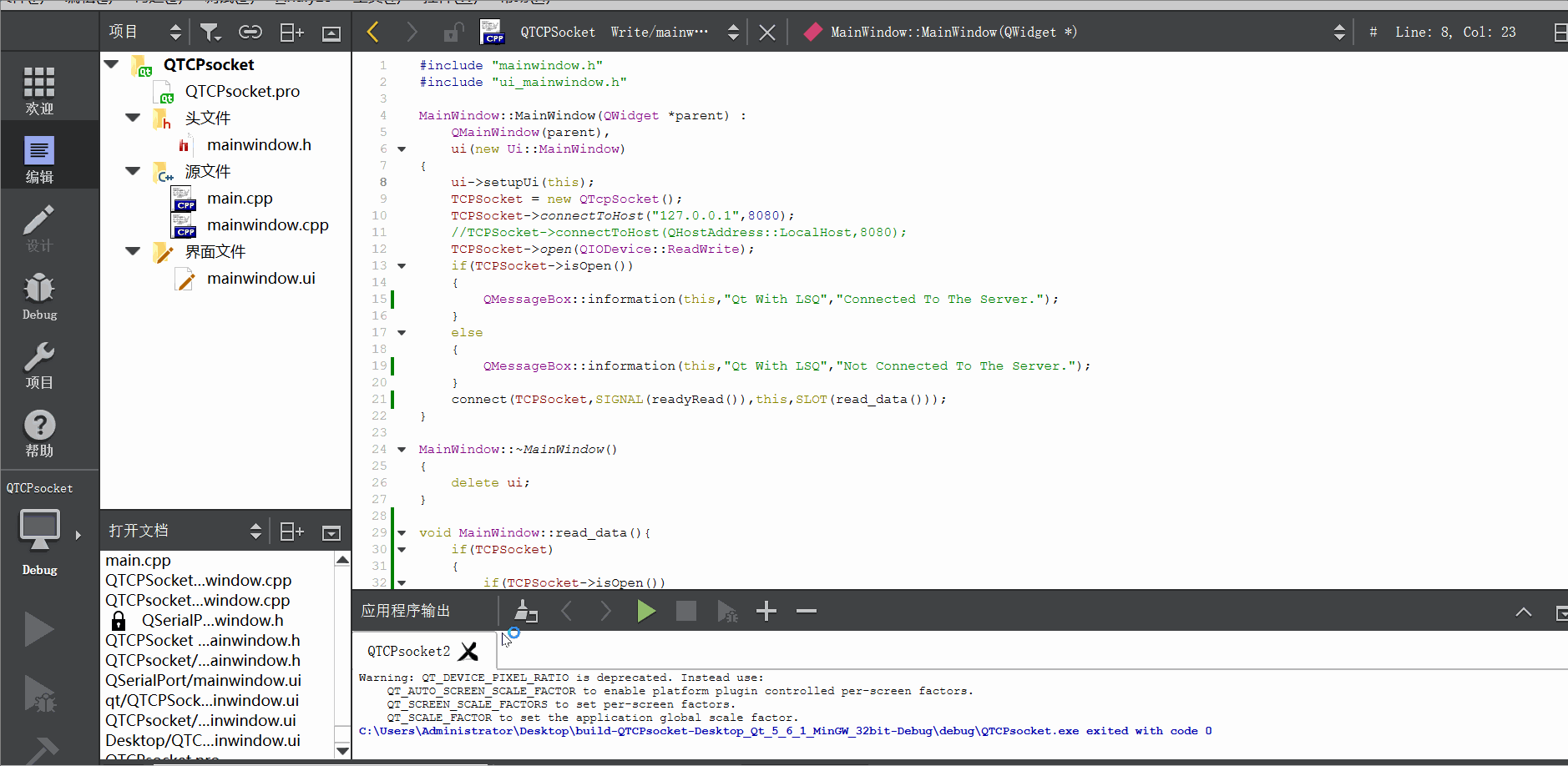
通过构建（Build）检查是否存在error或者warning，在通过构建后点击运行按钮进行测试，经验证实现了点击按钮关闭窗口的功能。

图4 Qt Creator验证测试