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）

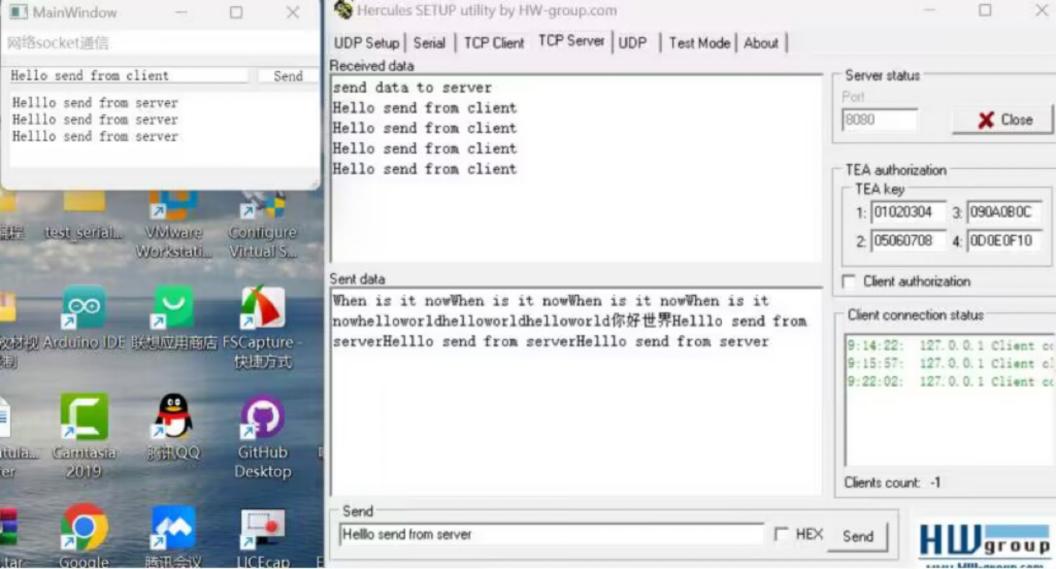


图2 网络socket收发数据应用演示（Qt为Client， HW为Server）

1.新建项目工程文件

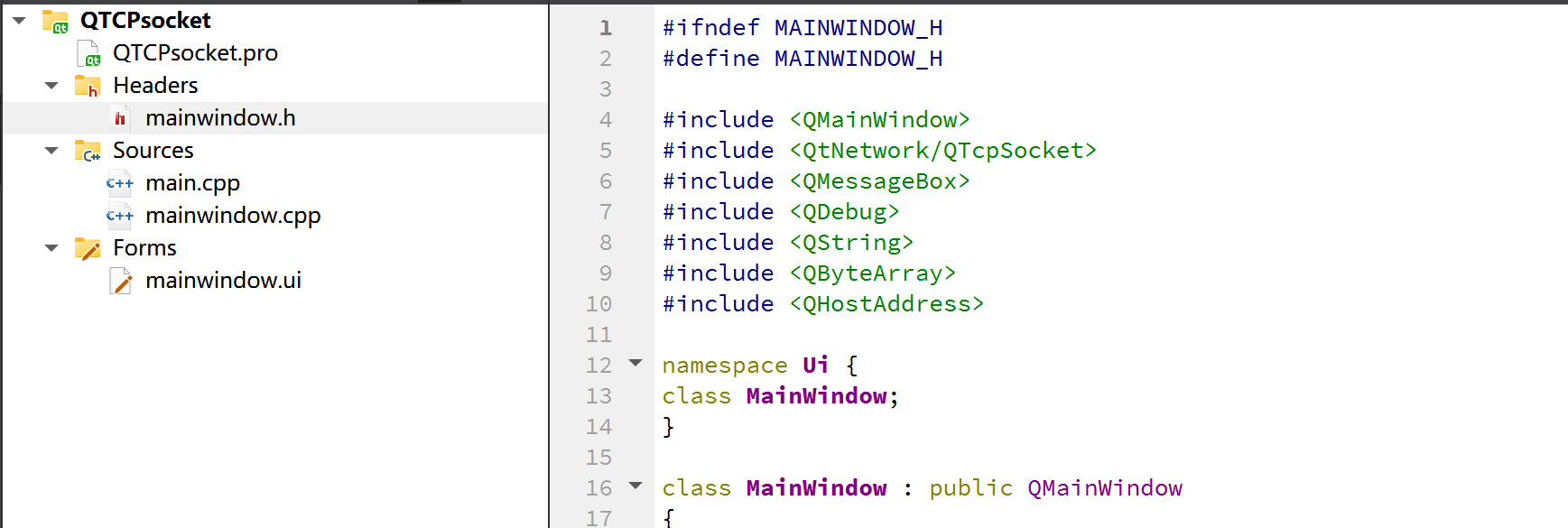


图1.项目工程文件

1. 进行ui界面布局

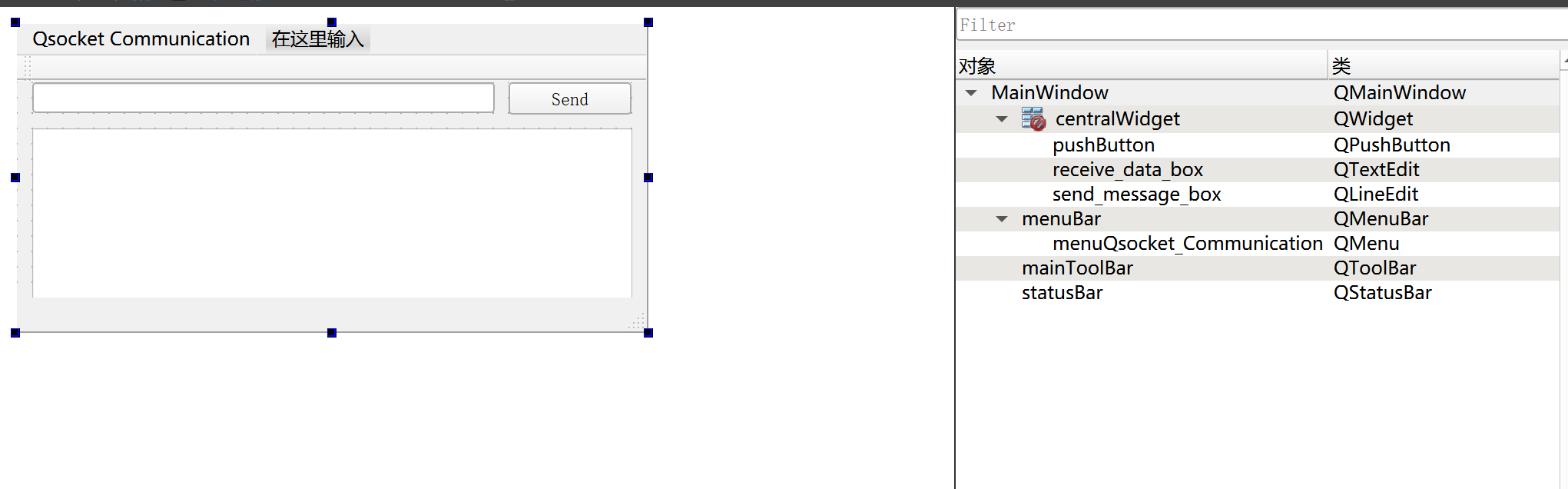


图2 ui界面布局

1. 代码编写

#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 Mr Ping","Connected To The Server.");

}

else

{

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

}

}

MainWindow::~***MainWindow***()

{

delete ui;

}

1. 槽函数代码

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 Mr Ping","Error:"+TCPSocket->errorString());

}

}

else

{

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

}

}

5.Assignm7 test

