1. 新建工程文件如下，并且在pro文件中添加network

QT += core gui network 如图1

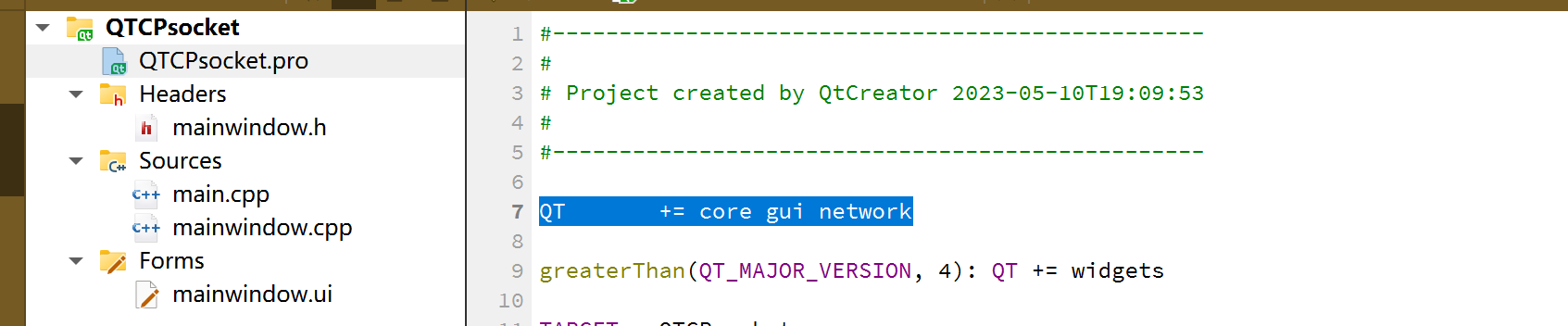


图1

1. 设计ui界面如下图

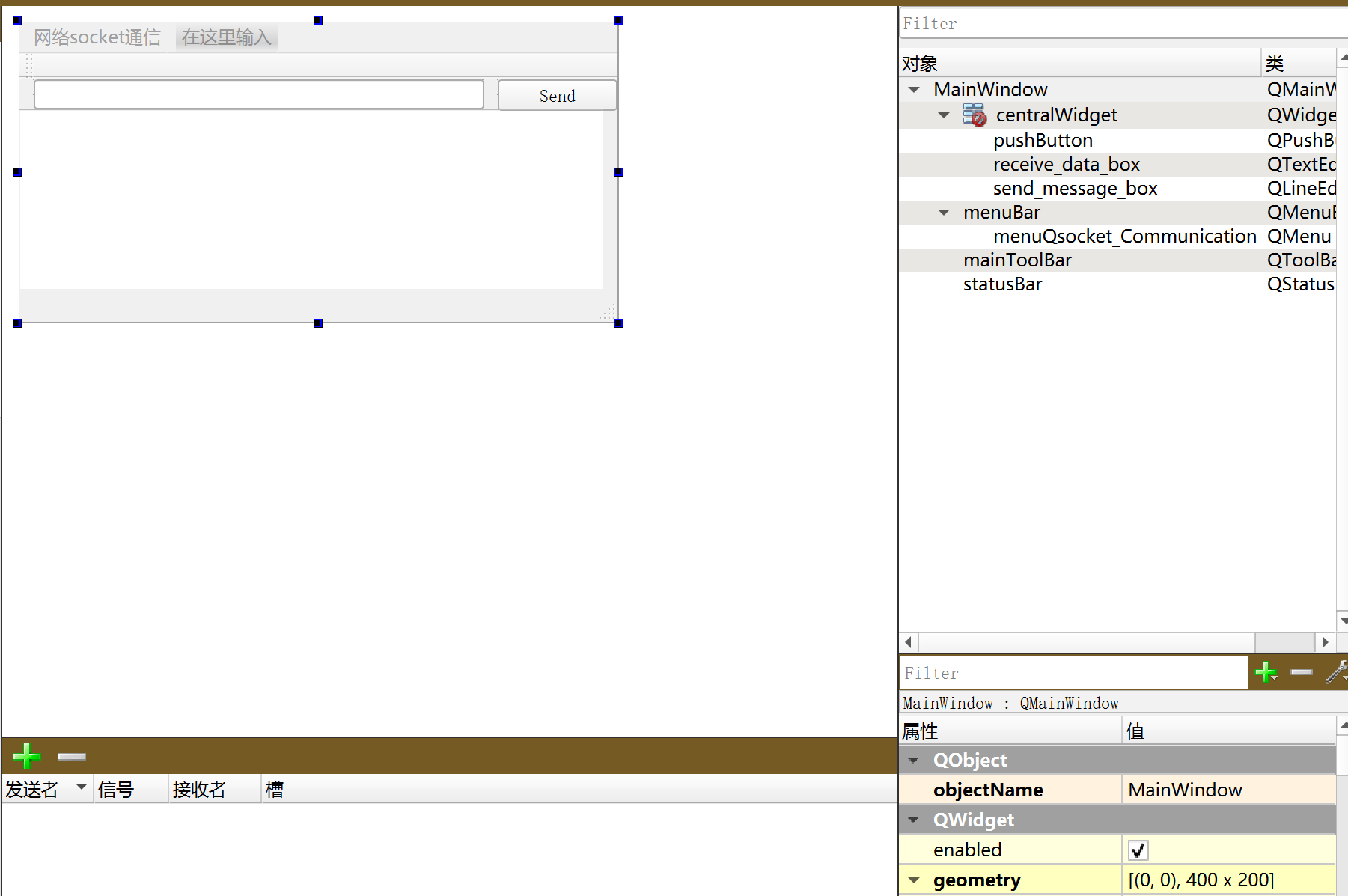


图2

3.在头文件中添加socket相应库文件如图3

#include <QMainWindow>

#include <QtNetwork/QTcpSocket>

#include <QMessageBox>

#include <QDebug>

#include <QString>

#include <QByteArray>

#include <QtNetwork/QHostAddress>

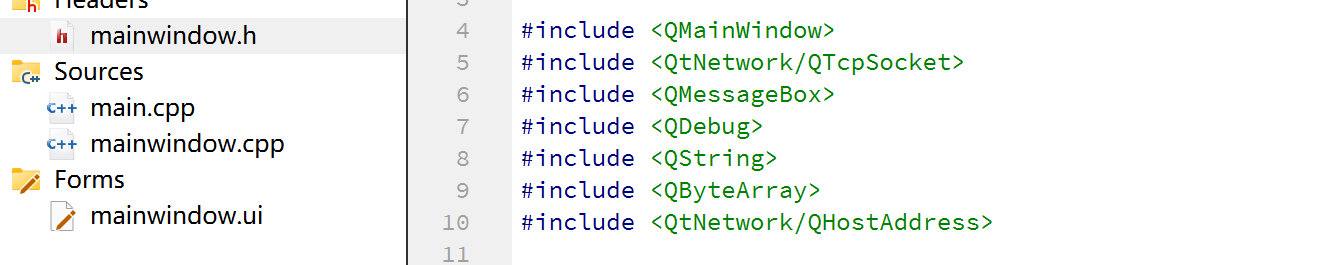


图3

声明端口和函数 如图4

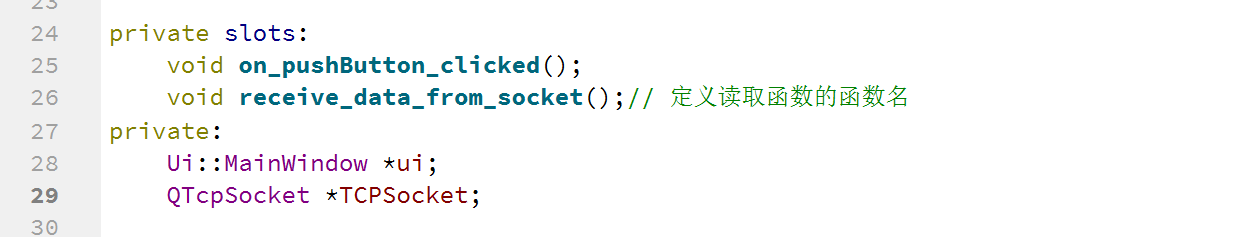


图4

4.在cpp文件中添加对端口的设置 如图5



图5

MainWindow::**MainWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

TCPSocket = new QTcpSocket();// 实例化 new

connect(TCPSocket,SIGNAL(readyRead()),this,SLOT(receive\_data\_from\_socket()));//；连接自定义槽函数

TCPSocket->*connectToHost*(QHostAddress::Any,8080); // 12，13行的作用是 实现某些功能 要强记

TCPSocket->*open*(QIODevice::ReadWrite);//赋予读写的功能

if(TCPSocket->isOpen())// is open 判断是否打开

{

QMessageBox::information(this,"chen fei fan","Connected To The Server.");

}

else

{

QMessageBox::information(this,"chen fei fan","Not Connected To The Server.");

}

}

5.将send按钮转为槽函数 如图6



图6

void MainWindow::**on\_pushButton\_clicked**()

{

if(TCPSocket)

{

if(TCPSocket->isOpen())

{

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

TCPSocket->write(WriteData.toLocal8Bit());//command.toLocal8Bit() toUtf8 toStdString().c\_str()

}

else

{

QMessageBox::information(this,"chen fei fan","Error:"+TCPSocket->errorString());

}

}

else

{

QMessageBox::information(this,"chen fei fan","Error:"+TCPSocket->errorString());

}

}

6.自定义函数读取函数**receive\_data\_from\_socket 如图7**

**（记得在头文件声明函数）**

****

**图7**

void MainWindow::**receive\_data\_from\_socket**()

{ if(TCPSocket)

{

if(TCPSocket->isOpen())

{

QByteArray nameArray = TCPSocket->readAll(); //从array数组的pos开始，取10个字节

QTextCodec \*codec = QTextCodec::codecForName("GBK");//指定QString的编码方式

QString name = codec->toUnicode(nameArray);//nameArray可以是char\*，可以是QByteArray

//QString MessageString = QString::fromStdString(nameArray.toStdString());

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

}

}

}

7.点击构建，并运行，打开网络监听助手选择 tcp server 一项开始测试

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）如图8

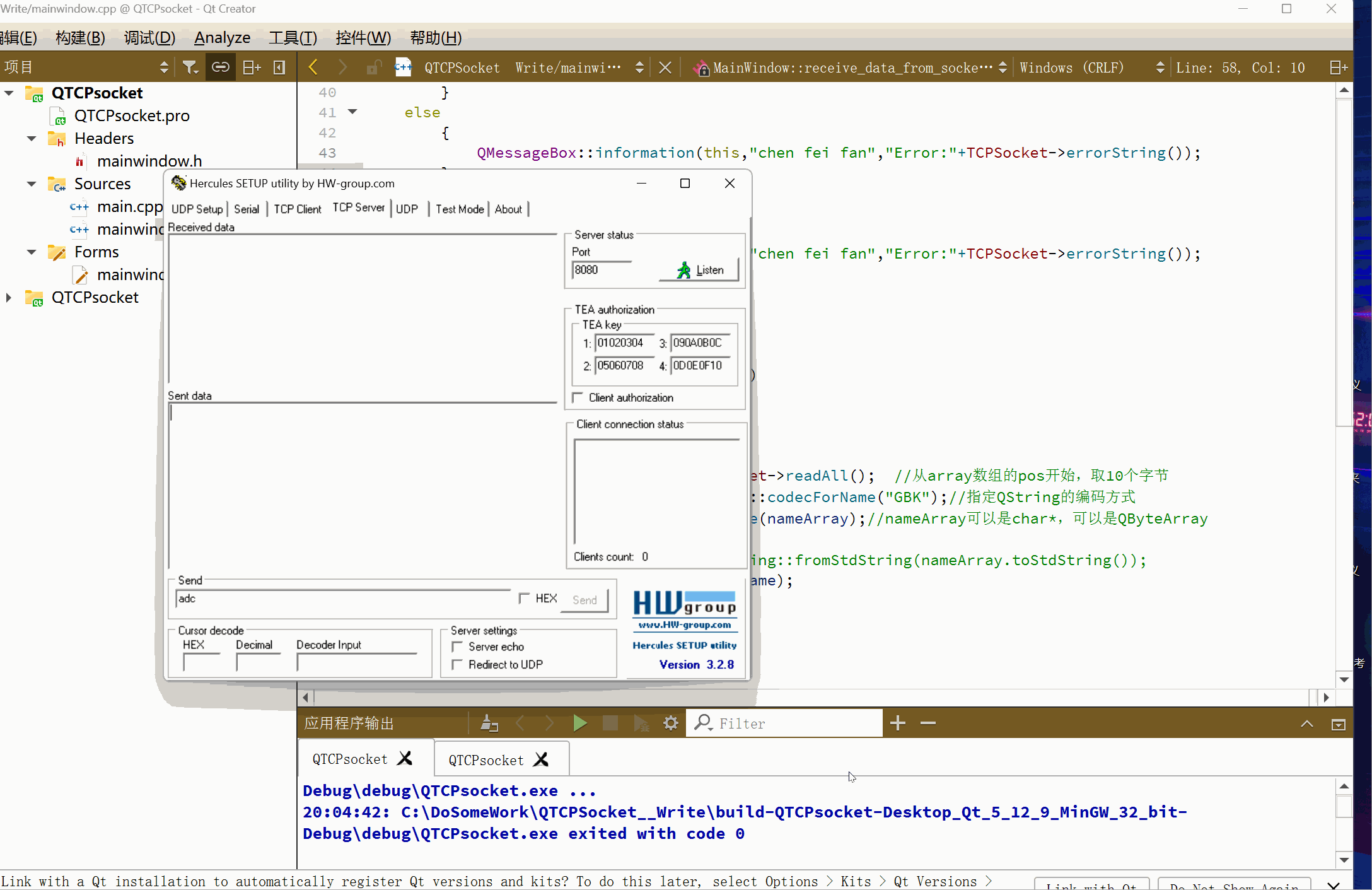


图8