**TCP/IP协议**

**在.pro中加入QT+=network**

**客户端：**

1. **声明客户端套接字,构造函数**

QTcpSocket\*tcpSocket; //声明客户端套接字

tcpSocket=new QtcpSocket(this); //实例化套接字对象

connect(tcpSocket,SIGNAL(connected()),this,SLOT(dataWrite())); //连接信号与槽，一旦连接到服务器，则发送数据

connect(tcpSocket,SIGNAL(error(QAbstractSocket::SocketError)),this,SLOT(displayError(QAbstractSocket::SocketError)));//连接信号与槽，如果发现运行中套接字出现错误，则输出错误信息

1. **连接服务器void newTcpConnect();**

tcpSocket->abort();//终止过去的连接

tcpSocket->connectToHost(QHostAddress(“127.0.0.1”),8080);//连接到服务器

1. **传输数据 void dataWrite();**

tcpSocket->write(“this is a test program.”); //写入数据

1. **捕获错误 void displayError(QAbstractSocket::SocketError);**

qDebug()<<tcpSocket->errorString(); //输出错误提示

tcpSocket->close(); //关闭套接字

**服务端：**

1. **声明客户端套接字和服务端套接字**

QTcpSocket\*tcpSocket; //声明客户端套接字

QTcpServer\*tcpServer; //声明服务端套接字

tcpSocket=new QtcpSocket(this); //实例化套接字对象

tcpServer=new QTcpServer(this);//实例化套接字对象

connect(tcpServer,SIGNAL(newConnection()),this,SLOT(acceptConnection()));//连接信号与槽，发现有客户端连接到服务器，则接受连接

startListen(); //开始监听

1. **监听 void startListen();**

If(!tcpServer->listen(QHostAddress::Any,8080))//如果监听出现错误

{

close(); //则关闭

return; //返回

}

1. **接受连接 void acceptConnection();**

tcpSocket=tcpServer->nextPendingConnection(); //获取一个客户端连接

connect(tcpSocket,SIGNAL(readyRead()),this,SLOT(dataRead()));//连接信号与槽，做好了接收数据的准备

1. **读取数据 void dataRead();**

QString strData=tcpSocket->readAll(); //读取来自客户端的所有数据

**UDP协议**

**在.pro中加入QT+=network**