



# 第30篇 (网络) UDP

代码地址：<https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Network03>

代码地址：<https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Network04>

一、发送端（客户端）

二、接收端（服务器端）

## 一、发送端（客户端）

1. 新建Qt Gui应用。项目名为 `udpSender`，基类选择 `QWidget`，类名为 `Widget`。完成后在 `udpSender.pro` 文件中添加一行代码：`QT += network`，并保存该文件。

2. 在 `widget.ui` 文件中，往界面上添加一个 `Push Button`，更改其显示文本为“开始广播”，然后进入其单击事件槽函数。

3. 我们在 `widget.h` 文件中更改。

添加头文件：`#include <QtNetwork>`

添加private私有对象：`QUdpSocket *sender;`

4. 我们在 `widget.cpp` 中进行更改。

在构造函数中添加：`sender = new QUdpSocket(this);`

更改“开始广播”按钮的单击事件槽函数：

```
void Widget::on_pushButton_clicked() // 开始广播
{
    QByteArray datagram = "hello world!";
    sender->writeDatagram(datagram.data(), datagram.size(),
                        QHostAddress::Broadcast, 45454);
}
```

这里定义了一个 `QByteArray` 类型的数据报 `datagram`，其内容为“hello world!”。然后我们使用 `QUdpSocket` 类的 `writeDatagram()` 函数来发送数据报，这个函数有四个参数，分别是数据报的内容，数据报的大小，主机地址和端口号。对于数据报的大小，它根据平台的不同而不同，但是这里建议不要超过512字节。这里使用了广播地址

`QHostAddress::Broadcast`，这样就可以同时给网络中所有的主机发送数据报了。对于端

口号，它是可以随意指定的，但是一般1024以下的端口号通常属于保留端口号，所以我们最好使用大于1024的端口，最大为65535。我们这里使用了45454这个端口号，一定要注意，在下面要讲的服务器程序中，也要使用相同的端口号。

5. 发送端就这么简单，下面可以先运行程序。

## 二、接收端（服务器端）

### 1. 新建Qt Gui 应用

工程名为 `udpReceiver`，基类选择 `QWidget`，类名为 `Widget`。完成后在 `udpSender.pro` 文件中添加一行代码：`QT += network`，并保存该文件。

此时工程文件列表中应包含两个项目，如下图。



2. 我们在 `udpReceiver` 项目中的 `widget.ui` 文件中，向界面上添加一个 `Label` 部件，更改其显示文本为“等待接收数据！”，效果如下。

3. 我们在 `udpReceiver` 工程中的 `widget.h` 文件中更改。

添加头文件：`#include <QtNetwork>`

添加private私有对象：`QUdpSocket *receiver;`

添加私有槽函数：

```
private slots:
void processPendingDatagram();
```

4. 我们在 `udpReceiver` 工程中的 `widget.cpp` 文件中更改。

在构造函数中：

```
receiver = new QUdpSocket(this);
receiver->bind(45454, QUdpSocket::ShareAddress);
connect(receiver, SIGNAL(readyRead()),
this, SLOT(processPendingDatagram()));
```

我们在构造函数中将 `receiver` 绑定到45454端口，这个端口就是上面发送端设置的端口，二者必须一样才能保证接收到数据报。这里使用了绑定模式

`QUdpSocket::ShareAddress`，它表明其他服务也可以绑定到这个端口上。因为当 `receiver` 发现有数据报到达时就会发出 `readyRead()` 信号，所以将其和数据报处理函数相关联。

数据报处理槽函数实现如下：

```
void Widget::processPendingDatagram() //处理等待的数据报
{
    while(receiver->hasPendingDatagrams()) //拥有等待的数据报
    {
        QByteArray datagram; //用于存放接收的数据报
        //让datagram的大小为等待处理的数据报的大小，这样才能接收到完整的数据
        datagram.resize(receiver->pendingDatagramSize());
        //接收数据报，将其存放到datagram中
        receiver->readDatagram(datagram.data(), datagram.size());
        //将数据报内容显示出来
        ui->label->setText(datagram);
    }
}
```

5. 我们在项目列表中 `udpReceiver` 项目上点击鼠标右键，在弹出的菜单上选择run菜单来运行该工程。如下图所示。



6. 第一次运行该程序时，系统可能会提示警告，我们选择“解除阻止”。注意，如果是在linux下，你可能还需要关闭防火墙。

7. 我们同时再运行 `udpSender` 程序。然后点击其上的“发送广播”按钮，这时会在 `udpReceiver` 上显示数据报的内容。效果如下。

