97 TCP客户机功能实现

第三步，实现TcpClient的通信功能。

97.1 实现功能

C:\Users\Minwei\Projects\Qt\TcpClient\tcpclientwindow.h：

```
#ifndef TCPCLIENTWINDOW_H
#define TCPCLIENTWINDOW_H

#include <QMainWindow>
#include <QLabel>
#include <QTcpSocket>

QT_BEGIN_NAMESPACE
namespace Ui { class TcpClientWindow; }
QT_END_NAMESPACE

class TcpClientWindow : public QMainWindow
{
    Q_OBJECT

public:
    TcpClientWindow(QWidget *parent = nullptr);
    ~TcpClientWindow();

private slots:
    void on_m_actConnect_triggered();
    void on_m_actDisconnect_triggered();
    void on_m_actClear_triggered();

    void on_m_btnSend_clicked();

    void on_m_socket_connected();
    void on_m_socket_readyRead();
    void on_m_socket_disconnected();

    void on_m_socket_stateChanged(
        QAbstractSocket::SocketState socketState);
    void on_m_socket_error(
        QAbstractSocket::SocketError socketError);

private:
    Ui::TcpClientWindow *ui;
    QLabel* m_labConnectionState;
    QLabel* m_labSocketState;
    QLabel* m_labSocketError;
    QTcpSocket* m_socket;
};

#endif // TCPCLIENTWINDOW_H
```

C:\Users\Minwei\Projects\Qt\TcpClient\tcpclientwindow.cpp：

```cpp
#include <QHostInfo>
#include <QMetaEnum>

#include "tcpclientwindow.h"
#include "ui_tcpclientwindow.h"

TcpClientWindow::TcpClientWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::TcpClientWindow)
    , m_labConnectionState(new QLabel("连接状态：断开"))
    , m_labSocketState(new QLabel(" 套接字状态："))
    , m_labSocketError(new QLabel(" 套接字错误："))
    , m_socket(new QTcpSocket(this))
{
    ui->setupUi(this);

    m_labConnectionState->setMinimumWidth(196);
    ui->m_statusBar->addWidget(m_labConnectionState);
    m_labSocketState->setMinimumWidth(197);
    ui->m_statusBar->addWidget(m_labSocketState);
    m_labSocketError->setMinimumWidth(197);
    ui->m_statusBar->addWidget(m_labSocketError);

    connect(m_socket, SIGNAL(connected()),
        this, SLOT(on_m_socket_connected()));
    connect(m_socket, SIGNAL(readyRead()),
        this, SLOT(on_m_socket_readyRead()));
    connect(m_socket, SIGNAL(disconnected()),
        this, SLOT(on_m_socket_disconnected()));

    connect(m_socket, SIGNAL(stateChanged(QAbstractSocket::SocketState)),
        this,
SLOT(on_m_socket_stateChanged(QAbstractSocket::SocketState)));
    connect(m_socket, SIGNAL(error(QAbstractSocket::SocketError)),
        this, SLOT(on_m_socket_error(QAbstractSocket::SocketError)));
}

TcpClientWindow::~TcpClientWindow()
{
    delete ui;
}

void TcpClientWindow::on_m_actConnect_triggered()
{
    m_socket->connectToHost(ui->m_editRemoteAddr->text(),
        ui->m_editRemotePort->text().toUShort());
}

void TcpClientWindow::on_m_actDisconnect_triggered()
{
    m_socket->disconnectFromHost();
}

void TcpClientWindow::on_m_actClear_triggered()
{
```

```
    ui->m_editOutput->clear();
}

void TcpClientWindow::on_m_btnSend_clicked()
{
    QByteArray data((ui->m_editSend->text() + "\n").toUtf8());

    qint64 nbytes = m_socket->write(data);

    if (nbytes == -1)
        ui->m_editOutput->appendPlainText("发送失败");
    else
        ui->m_editOutput->appendPlainText(QString("向%1:%2发送%3字节：%4").
            arg(m_socket->peerAddress().toString()).arg(m_socket->peerPort()).
            arg(nbytes).arg(QString(data).trimmed()));
}

void TcpClientWindow::on_m_socket_connected()
{
    ui->m_editOutput->appendPlainText(QString("建立%1:%2连接").
        arg(m_socket->peerAddress().toString()).arg(m_socket->peerPort()));
    m_labConnectionState->setText(QString("连接状态：%1:%2").
        arg(m_socket->peerAddress().toString()).arg(m_socket->peerPort()));

    ui->m_btnSend->setEnabled(true);
}

void TcpClientWindow::on_m_socket_readyRead()
{
    while (m_socket->canReadLine())
    {
        QByteArray data = m_socket->readLine();

        ui->m_editOutput->appendPlainText(QString("从%1:%2接收%3字节：%4").
            arg(m_socket->peerAddress().toString()).arg(m_socket->peerPort()).
            arg(data.size()).arg(QString(data).trimmed()));
    }
}

void TcpClientWindow::on_m_socket_disconnected()
{
    ui->m_btnSend->setEnabled(false);

    ui->m_editOutput->appendPlainText(QString("断开%1:%2连接").
        arg(m_socket->peerAddress().toString()).arg(m_socket->peerPort()));
    m_labConnectionState->setText("连接状态：断开");
}

void TcpClientWindow::on_m_socket_stateChanged(
    QAbstractSocket::SocketState socketState)
{
    m_labSocketState->setText(QString(" 套接字状态：%1").arg(
        QMetaEnum::fromType<QAbstractSocket::SocketState>().
        valueToKey(socketState)));
```

```
109 }
110
111 void TcpClientWindow::on_m_socket_error(
112     QAbstractSocket::SocketError socketError)
113 {
114     m_labSocketError->setText(QString(" 套接字错误：%1").arg(
115         QMetaEnum::fromType<QAbstractSocket::SocketError>().
116         valueToKey(socketError)));
117 }
```

97.2 测试验证

运行效果如图所示：