

@[toc]

一、UDP介绍：

用户数据报协议，属于传输层的协议，无连接，不保证传输的可靠性。对于来自应用层的数据包，直接加上UDP报头然后传送给IP。UDP头部中有一个校验和字段，可用于差错的检测，但是UDP是不提供差错纠正的。此外IPV4不强制这个校验和字段必须使用，但IPV6是强制要求使用的。

二、C#实现hello world

用C#、Java或python编写一个命令行/控制台的简单hello world程序，实现如下功能：在屏幕上连续输出50行“hello cqjtu! 重交物联2019级”；同时打开一个网络UDP 套接字，向另一台室友电脑发送这50行消息。

1、创建项目

- 打开vs2022，选择创建控制台应用



2、代码

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
```



```

        Console.WriteLine("按下任意按键开始发送...");
        Console.ReadKey();

        int m;

        //做好链接准备
        UdpClient client = new UdpClient(); //实例一个端口
        IPAddress remoteIP = IPAddress.Parse("127.0.0.1"); //假设发送给这个IP
        int remotePort = 11000; //设置端口号
        IPEndPoint remotePoint = new IPEndPoint(remoteIP, remotePort); //实例化一个远程端点

        for(int i = 0; i < 50; i++)
        {
            //要发送的数据: 第n行: hello cqjtu! 重交物联2018级
            string sendString = null;
            sendString += "第";
            m = i+1;
            sendString += m.ToString();
            sendString += "行: hello cqjtu! cj物联2019级";

            //定义发送的字节数组
            //将字符串转化为字节并存储到字节数组中
            byte[] sendData = null;
            sendData = Encoding.Default.GetBytes(sendString);

            client.Send(sendData, sendData.Length, remotePoint); //将数据发送到远程端点
        }
        client.Close(); //关闭连接

        //提示信息
        Console.WriteLine("");
        Console.WriteLine("数据发送成功, 按任意键退出...");
        System.Console.ReadKey();
    }
}

```

- 服务端代码:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;

namespace Server
{
    class Program
    {
        static void Main(string[] args)
        {
            int result;

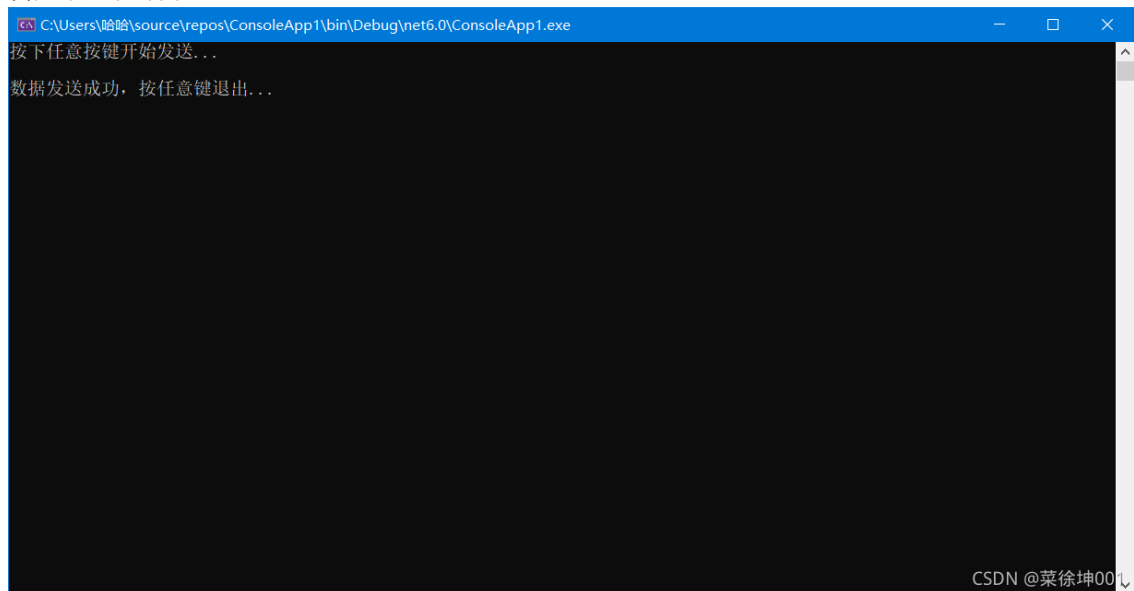
```

```

string str = "第50行: hello cqjtu! cj物联2019级";
UdpClient client = new UdpClient(11000);
string receiveString = null;
byte[] receiveData = null;
//实例化一个远程端点, IP和端口可以随意指定, 等调用client.Receive(ref
remotePoint)时会将该端点改成真正发送端点
IPEndPoint remotePoint = new IPEndPoint(IPAddress.Any, 0);
Console.WriteLine("正在准备接收数据...");
while (true)
{
    receiveData = client.Receive(ref remotePoint); //接收数据
    receiveString = Encoding.Default.GetString(receiveData);
    Console.WriteLine(receiveString);
    result = String.Compare(receiveString, str);
    if (result == 0)
    {
        break;
    }
}
client.Close(); //关闭连接
Console.WriteLine("");
Console.WriteLine("数据接收完毕, 按任意键退出...");
System.Console.ReadKey();
}
}
}

```

- 客户端运行结果:



- 服务端运行结果:

```
第22行: hello cqjtu! cjq物联2019级
第23行: hello cqjtu! cjq物联2019级
第24行: hello cqjtu! cjq物联2019级
第25行: hello cqjtu! cjq物联2019级
第26行: hello cqjtu! cjq物联2019级
第27行: hello cqjtu! cjq物联2019级
第28行: hello cqjtu! cjq物联2019级
第29行: hello cqjtu! cjq物联2019级
第30行: hello cqjtu! cjq物联2019级
第31行: hello cqjtu! cjq物联2019级
第32行: hello cqjtu! cjq物联2019级
第33行: hello cqjtu! cjq物联2019级
第34行: hello cqjtu! cjq物联2019级
第35行: hello cqjtu! cjq物联2019级
第36行: hello cqjtu! cjq物联2019级
第37行: hello cqjtu! cjq物联2019级
第38行: hello cqjtu! cjq物联2019级
第39行: hello cqjtu! cjq物联2019级
第40行: hello cqjtu! cjq物联2019级
第41行: hello cqjtu! cjq物联2019级
第42行: hello cqjtu! cjq物联2019级
第43行: hello cqjtu! cjq物联2019级
第44行: hello cqjtu! cjq物联2019级
第45行: hello cqjtu! cjq物联2019级
第46行: hello cqjtu! cjq物联2019级
第47行: hello cqjtu! cjq物联2019级
第48行: hello cqjtu! cjq物联2019级
第49行: hello cqjtu! cjq物联2019级
```

CSDN @菜徐坤001

4. 使用Wireshark进行抓包

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
2	0.000067	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
3	0.000082	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
4	0.000095	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
5	0.000108	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
6	0.000121	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
7	0.000133	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
8	0.000145	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
9	0.000157	127.0.0.1	127.0.0.1	UDP	66	52674 → 11000 Len=34
10	0.000175	127.0.0.1	127.0.0.1	UDP	67	52674 → 11000 Len=35
11	0.000189	127.0.0.1	127.0.0.1	UDP	67	52674 → 11000 Len=35
12	0.000204	127.0.0.1	127.0.0.1	UDP	67	52674 → 11000 Len=35
13	0.000221	127.0.0.1	127.0.0.1	UDP	67	52674 → 11000 Len=35
14	0.000235	127.0.0.1	127.0.0.1	UDP	67	52674 → 11000 Len=35

> Frame 8: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{...} id 0
> Null/loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> User Datagram Protocol, Src Port: 52674, Dst Port: 11000
> Data (34 bytes)

```
0000 02 00 00 00 45 00 00 3e e6 7d 00 00 80 11 00 00 ....E...> .....
0010 7f 00 00 01 7f 00 00 01 cd c2 2a f8 00 2a ce 15 .....
0020 b5 da 38 d0 d0 a3 ba 68 65 6c 6c 6f 20 63 71 6a ..8....h ello cqj
0030 74 75 a3 a1 d6 d8 db bb ce ef c1 aa 32 30 31 39 tu.....2019
```

CSDN @菜徐坤001

三、Form窗口程序使用 UDP 通信

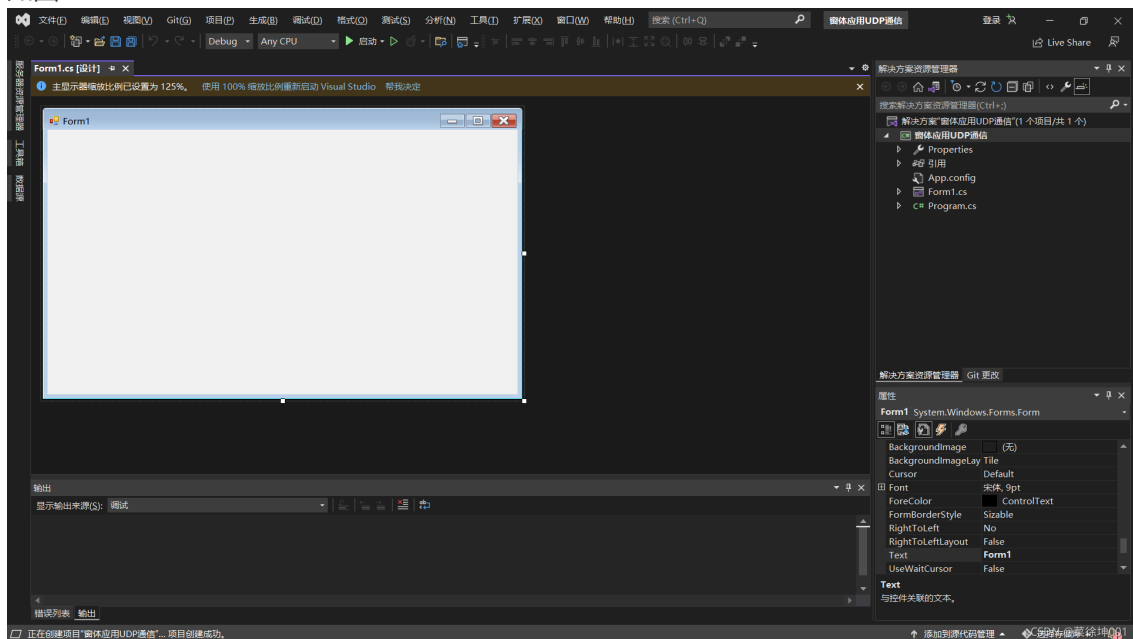
用VS2017/2019 的C#编写一个简单的Form窗口程序，有一个文本框 textEdit和一个发送按钮 button，运行程序后，可以在文本框里输入文字，如“hello cqjtu! 重交物联2019级”，点击 button，将这些文字发送给室友电脑，采用UDP套接字；

1、新建项目

- 选择Windows窗口应用

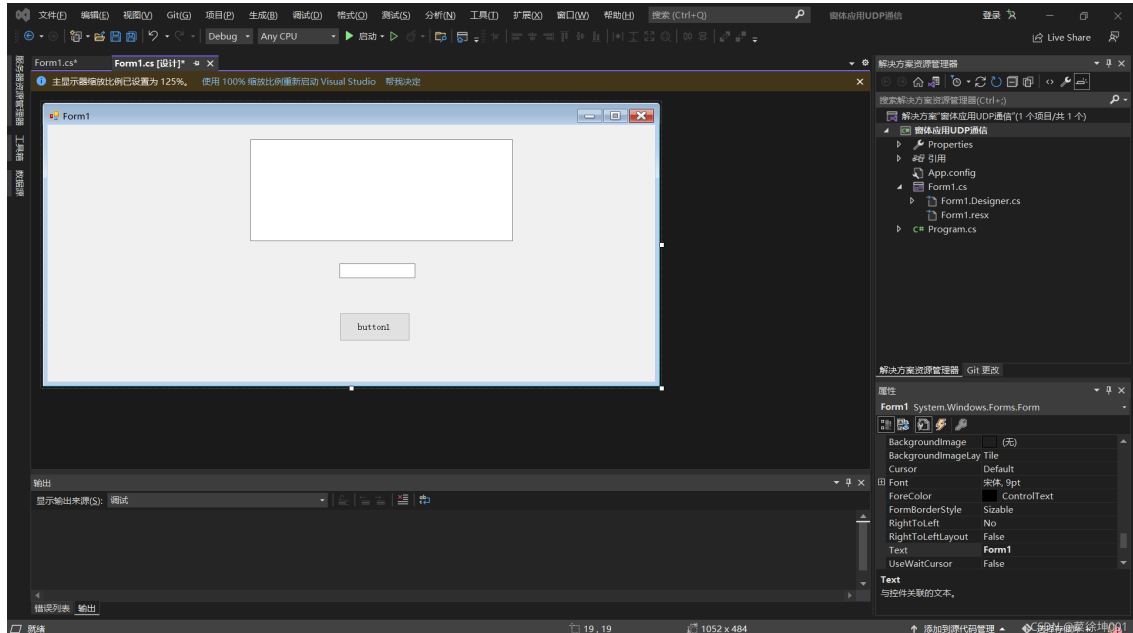


- 如图:

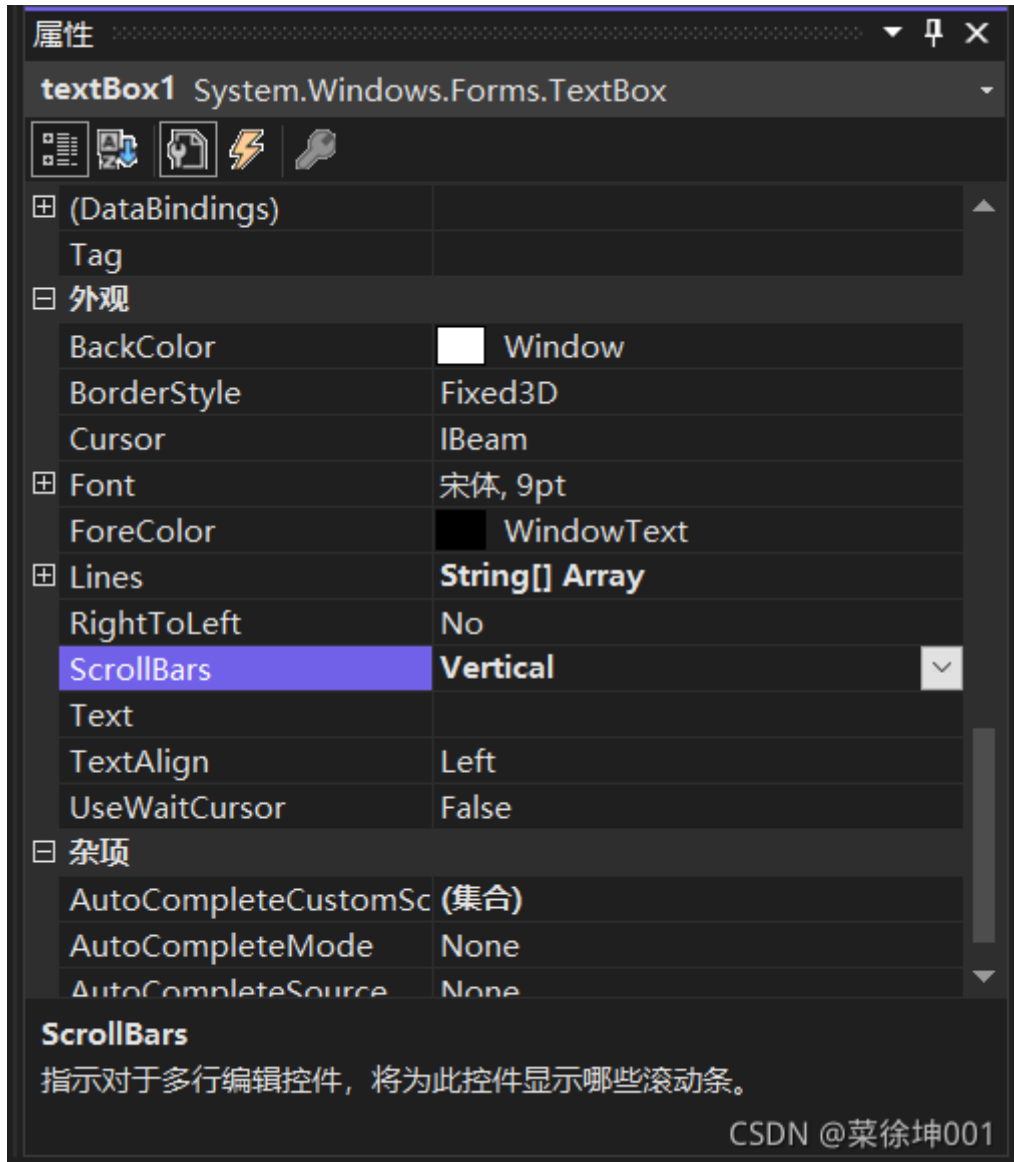


2、界面设计

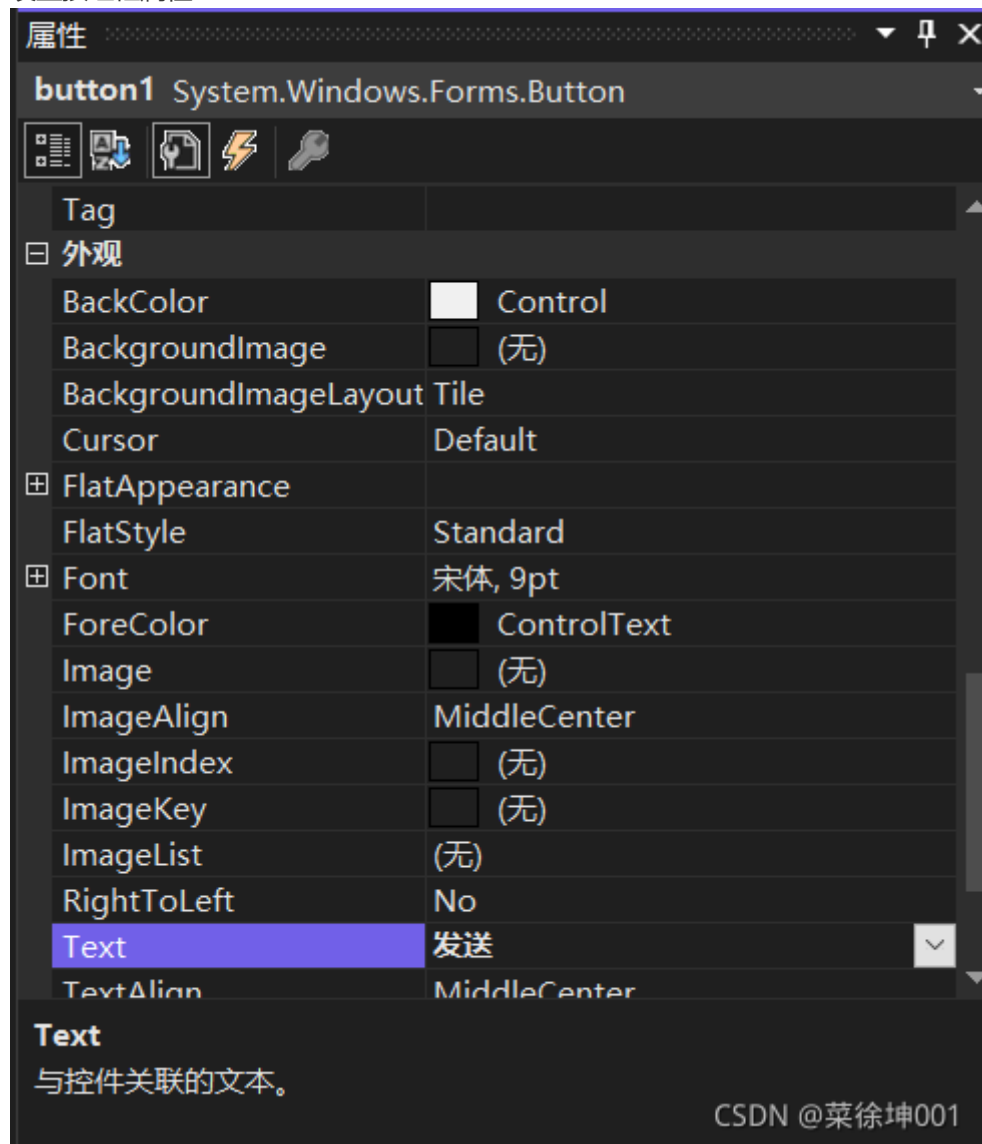
- 从工具箱内拖 2 个TextBox 和 1 个Button 控件



- 设置消息显示框属性



设置按钮框属性:



- 运行结果:

客户端:



服务端：

```
Perform operations 1 : -----
1. Wait for connect...
2. Get a connect
3. Server Get Message:Hello
4. Completed...
-----
```

```
Perform operations 2 : -----
1. Wait for connect...
2. Get a connect
3. Server Get Message:你好
4. Completed...
-----
```

CSDN @菜徐坤001

4. 使用Wireshark进行抓包

Wireshark packet capture analysis showing a TCP connection and data exchange between 127.0.0.1. The packet list shows a SYN, ACK, PSH sequence. The packet details pane shows the data field containing a hex string. The packet bytes pane shows the raw data with a red box highlighting the ASCII text "hello cq".

CSDN @菜徐坤001

四、总结

本次实验总体来说内容还是比较多的，包含用C#写一个控制台程序，通过UDP发送给另一台电脑相关信息，以及通过可视化窗口的形式再次实现上述要求，并编写端口扫描程序，再进行抓包，总体来说还是有一定难度，在网上找了不少的参考资料，收获较多。