

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;
using System.Net;
using System.Net.Sockets;
namespace RemoteControl
{
    public partial class Server : Form
    {
        int recv;
        int port
        byte[] data = new byte[1024];
        string stringData;
        IPEndPoint ipep;
        Socket newsock;
        EndPoint Remote;
```

```

public Server()
{
    InitializeComponent();
}

void Connect()
{
    list_box.Items.Clear();

    if (newsock != null)
        newsock.Close();

    ipep = new IPEndPoint(IPAddress.Any, port);

    newsock = new Socket(AddressFamily.InterNetwork,
        SocketType.Dgram, ProtocolType.Udp);

    newsock.Bind(ipep);

    list_box.Items.Add("Waiting for a client...");

    IPEndPoint temp = new IPEndPoint(IPAddress.Any, 0);

    Remote = (EndPoint)(temp);

    recv = newsock.ReceiveFrom(data, ref Remote);
    list_box.Items.Add("Message received from: " + Remote.ToString());
    list_box.Items.Add(Encoding.ASCII.GetString(data, 0, recv));

    string welcome = "Welcome client!!!";

    data = Encoding.ASCII.GetBytes(welcome);
    newsock.SendTo(data, data.Length, SocketFlags.None, Remote);
}

```

```

private void Send_Click(object sender, EventArgs e)
{

    string input = txt_Nhap.Text;
newsock.SendTo(Encoding.ASCII.GetBytes(input), Remote);

    if (input == "exit" || input == "EXIT"        {

        list_box.Items.Add("Stopping server

        newsock.Close();

        Close();

        return;

    }

    try

    {

        newsock.SetSocketOption(SocketOptionLevel.Socket,
SocketOptionName.ReceiveTimeout, 3000);

        data = new byte[1024];

        recv = newsock.ReceiveFrom(data, ref Remote);
stringData = Encoding.ASCII.GetString(data, 0, recv);
list_box.Items.Add(stringData);

    }

    catch (SocketException ex)

    {

```

```

        if (ex.ErrorCode == 10054)
list_box.Items.Add("Timeout ");

    }

    txt_Nhap.Text = "";

}

private void btnCheck_Click(object sender, EventArgs e)
{
    if(check_port())    {

        MessageBox.Show("Port number OK", "Thông Báo",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        Connect();

    }

    else

        MessageBox.Show("Port number error", "Thông Báo",
MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

bool check_port()

{

    try

    {

        if (string.IsNullOrEmpty(txt_Port.Text))
MessageBox.Show("Input port number", "Thông Báo", MessageBoxButtons.OK,
MessageBoxIcon.Warning);

```

```
        else
        {
            port = Int32.Parse(txt_Port.Text);
            return true;
        }
        return false;

    }
    catch(Exception ex)
    {
        return false;
    }
}

}
```

Server

Port:

Check Listen

list\_box

Send