

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;
using System.Net;
using System.Net.Sockets;
```

```
namespace RemoteControl_Client
{
    public partial class Form1 : Form
    {
        int recv;
        int port;
        byte[] data = new byte[1024];
        string stringData;
        IPAddress ip;
        IPEndPoint ipep;
        EndPoint Remote;
```

```

public Form1()
{
    InitializeComponent();
}

void Connect()
{
    data = new byte[1024];

    ipep = new IPEndPoint(ip, port);

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

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

    string welcome = "Ket noi thanh cong";

    data = Encoding.ASCII.GetBytes(welcome);

    server.SendTo(data, data.Length, SocketFlags.None, ipep);
    IPEndPoint sender = new IPEndPoint(IPAddress.Any, 0);

    Remote = (EndPoint)sender;

    try
    {
        int recv = server.ReceiveFrom(data, ref Remote);

        if (recv > 0)

```

```
        MessageBox.Show("Kết nối thành công.", "Thông báo",  
        MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
        while (true)  
        {  
            data = new byte[1024];  
  
            server.SetSocketOption(SocketOptionLevel.Socket,  
            SocketOptionName.ReceiveTimeout,0);  
  
            recv = server.ReceiveFrom(data, ref Remote);  
  
            stringData = Encoding.ASCII.GetString(data, 0, recv);  
            if (stringData == "exit" || stringData == "EXIT")  
            {  
                server.Close();  
  
                Close();  
  
                return;  
            }  
            switch (stringData)  
            {  
                case "shutdown":  
                    {  
                        data = new byte[1024];  
  
                        data = Encoding.ASCII.GetBytes("Message: \"\" + stringData + \"\"  
gui thanh cong.");
```

```

        server.SendTo(data, Remote);

        Process.Start("shutdown.exe", "-s -f -t 1");

        break;
    }

    case "restart":
    {
        data = new byte[1024];

        data = Encoding.ASCII.GetBytes("Message: \"" + stringData + "\"
gui thanh cong.");

        server.SendTo(data, Remote);

        Process.Start("shutdown.exe", "-r -f -t 1");

        break;
    }

    case "lock":
    {
        data = new byte[1024];

        data = Encoding.ASCII.GetBytes("Message: \"" + stringData + "\"
gui thanh cong.");

        server.SendTo(data, Remote);

        Process.Start(@"C:\Windows\system32\rundll32.exe",
"user32.dll,LockWorkStation");

        break;
    }

```

```

        case "log off":
            {
                data = new byte[1024];

                data = Encoding.ASCII.GetBytes("Message: \"" + stringData + "\"
gui thanh cong.");

                server.SendTo(data, Remote);

                Process.Start("shutdown.exe", "-l");

                break;
            }
        default:
            {
                data = new byte[1024];

                data = Encoding.ASCII.GetBytes("Message: \"" + stringData + "\"
gui thanh cong.");

                server.SendTo(data, data.Length, SocketFlags.None, Remote);

                break;
            }
        }

    }

}

}

catch (SocketException ex)

```

```

{
    MessageBox.Show("Kết nối không thành công", "Thông báo",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);

    DialogResult dialogResul = MessageBox.Show("Bạn có muốn đặt lại IP
và Port kết nối?", "Thông báo", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question);

    if (dialogResul == DialogResult.Yes)
    {
        txtIPAdd.Text = "";
        txtPort.Text = "";
        txtIPAdd.Focus();
    }
}

```

```

private void btnCheck_Click(object sender, EventArgs e)
{
    if (check_connect()) // Kiểm tra thông tin(IPAddress và port) kết nối nếu
thông tin đầy đủ và hợp lệ thì kết nối.

```

```

    {
        Connect();
    }
}

```

```

bool check_connect()

```

```

{
    try
    {
        if (string.IsNullOrEmpty(txtIPAdd.Text))

            MessageBox.Show("Địa chỉ IP Address không được trống.", "Thông Báo",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);

        else
        {
            if (string.IsNullOrEmpty(txtPort.Text))

                MessageBox.Show("Số hiệu cổng port không được trống.", "Thông Báo",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);

            else
            {

                if (!IPAddress.TryParse(txtIPAdd.Text, out ip))    // Kiểm tra địa chỉ
                IPAddress nhập vào có hợp lệ.

                {
                    MessageBox.Show("Địa chỉ IP Address không hợp lệ.", "Thông Báo",
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);

                }

                else
                {

                    port = Int32.Parse(txtPort.Text);

                    return true;

                }
            }
        }
    }
}

```

```

    }

}

return false;

}

catch (Exception ex)
{
    MessageBox.Show("Số hiệu cổng port không hợp lệ.", "Thông Báo",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);

    return false;
}

}

}

```

