

2022년 IoT기반 스마트 솔루션 개발자 양성과정



Embedded Application

13-TCP Communication in C#

담당 교수 : 윤 종 이

010-9577-1696

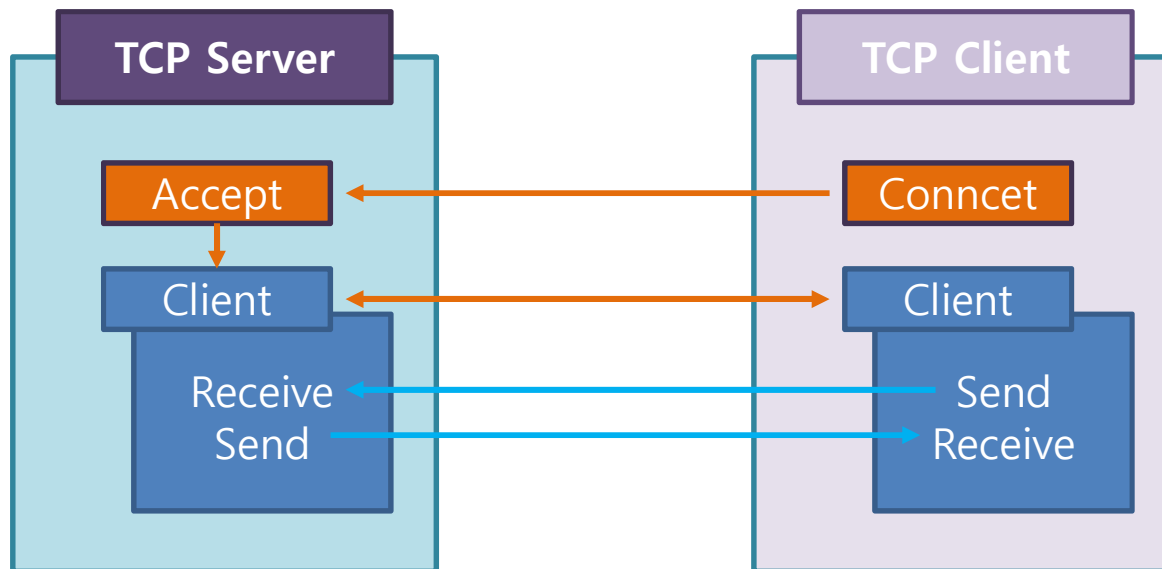
ojo1696@naver.com

<https://cafe.naver.com/yoons2022>



충북대학교 공동훈련센터

TCP Blockdiagram



새 프로젝트

새 프로젝트 구성

Windows Forms 앱(.NET Framework) C# Windows 데스크톱

프로젝트 이름(N)
TCP_Server

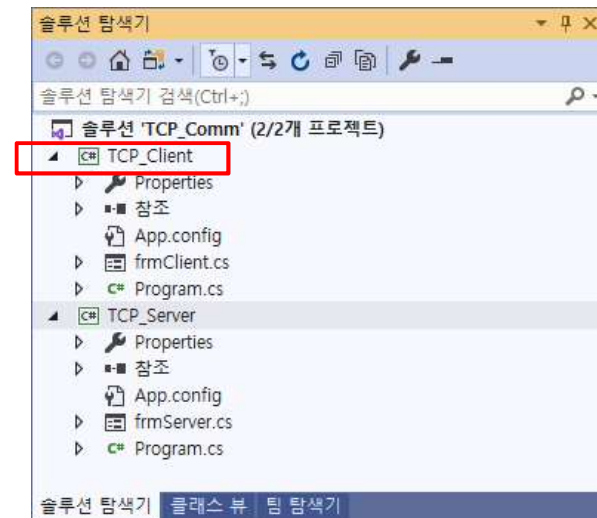
위치(L)
C:\Users\User\source\repos

솔루션 이름(M) ⓘ
TCP_Comm

☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

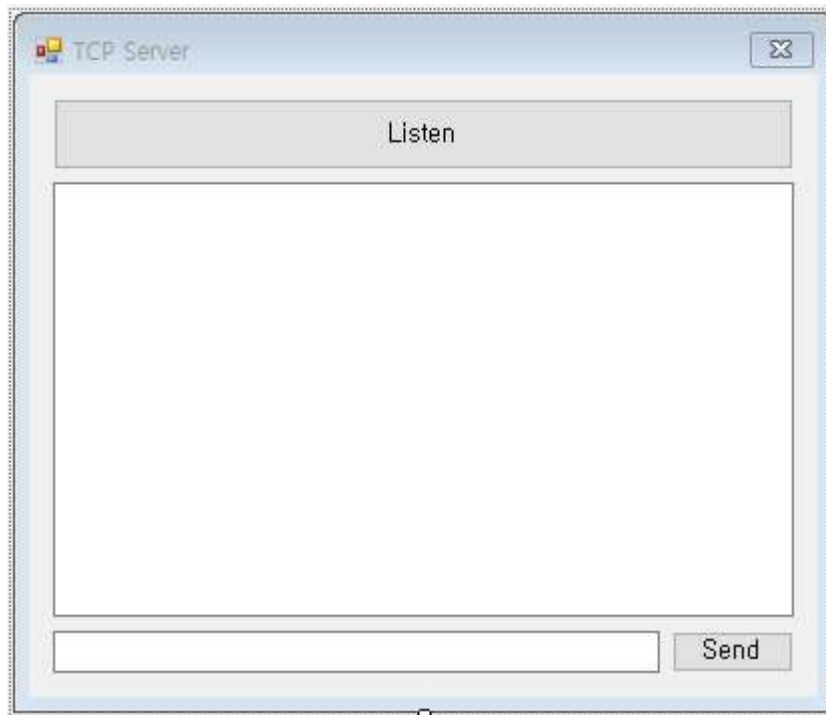
프레임워크(F)
.NET Framework 4.8

뒤로(B) 만들기(C)



충북대학교 공동훈련센터

TCP Server Design



Component	Property
btnListen	Listen
txtRxMsg	
txtSendMsg	
btnSend	Send



Using and Define

```
using System;  
using System.Windows.Forms;  
using System.Threading;  
using System.Net;  
using System.Net.Sockets;  
using System.IO;
```

```
private int LocalPort = 13000;  
private IPAddress LocalAddress = IPAddress.Parse("127.0.0.1");  
  
private TcpListener Listener = null;  
private TcpClient Client = null;  
  
private Thread ListenThread;  
private bool Listening = false;  
  
private StreamReader Reader;  
private StreamWriter Writer;  
private delegate void SetTextDelegate(string getString);
```



TCPmsgReceive()

```
private void TCPmsgReceive(string msg)
{
    txtRxMsg.AppendText(msg + "WrWn");
}
```



```

private void Listen() {
    try {
        Listener = new TcpListener(LocalAddress, LocalPort);
        Listener.Start();
        while (Listening) {
            Client = Listener.AcceptTcpClient();
            string msg = "Connected to Client!";
            this.BeginInvoke(new SetTextDelegate(TCPmsgReceive), new object[] { msg });
            NetworkStream stream = Client.GetStream();
            Reader = new StreamReader(stream);
            Writer = new StreamWriter(stream);
            while (Client.Connected) {
                Thread.Sleep(10);
                if (stream.CanRead) {
                    string strReceived = "Client : " + Reader.ReadLine();
                    this.BeginInvoke(new SetTextDelegate(TCPmsgReceive), new object[] { strReceived });
                }
            }
            Client.Close();
        }
    } catch (SocketException ex) {
        MessageBox.Show(ex.Message.ToString(), "TCP Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

btnListen_Click()

```
private void btnListen_Click(object sender, EventArgs e) {  
    if (btnListen.Text == "Listen") {  
        Listening = true;  
        ListenThread = new Thread(new ThreadStart(Listen));  
        ListenThread.Start( );  
        btnListen.Text = "Close";  
        txtRxMsg.AppendText("Listener Start!WrWn");  
    } else {  
        btnListen.Text = "Listen";  
        Listening = false;  
        Listener.Stop( );  
        ListenThread.Abort( );  
        txtRxMsg.AppendText("Listener Close!WrWn");  
    }  
}
```



btnSend_Click()

```
private void btnSend_Click(object sender, EventArgs e)
{
    if ((Listening) && (txtSendMsg.Text != "")){
        txtRxMsg.AppendText("Send : " + txtSendMsg.Text + "WrWn");
        Writer.WriteLine("Server," + txtSendMsg.Text);
        Writer.Flush( );
        txtSendMsg.Clear( );
    }
}
```



FormClosing()

```
private void frmServer_FormClosing(object sender, FormClosingEventArgs e)
{
    if (ListenThread != null) ListenThread.Abort( );
    if (Listener != null) Listener.Stop( );
    if (Client != null) Client.Close( );
    if (Reader != null) Reader.Close( );
    if (Writer != null) Writer.Close( );
}
```



TCP Client Design

TCP Client

User

IP Address

Connect

Send

Component	Property
txtUser	
txtIPAddress	127.0.0.1
txtRxMsg	
txtSendMsg	
btnConnect	Connect
btnSend	Send



Using and Define

```
using System;  
using System.Windows.Forms;  
using System.Threading;  
using System.Net;  
using System.Net.Sockets;  
using System.IO;
```

```
private int LocalPort = 13000;  
private IPAddress LocalAddress = IPAddress.Parse("127.0.0.1");  
  
private TcpClient Client = null;  
private Thread ReceiveThread;  
private bool Connected = false;  
  
private NetworkStream stream;  
private StreamReader Reader;  
private StreamWriter Writer;  
  
private delegate void SetTextDelegate(string getString);
```



TCPmsgReceive()

```
private void TCPmsgReceive(string msg)
{
    txtRxMsg.AppendText(msg + "WrWn");
}
```



Receive()

```
private void Receive( ) {  
    while (Connected) {  
        Thread.Sleep(1);  
  
        if (stream.CanRead) {  
            string strReceived = Reader.ReadLine( );  
  
            if (strReceived.Length > 0) {  
                strReceived = "Receive : " + strReceived;  
                this.BeginInvoke(new SetTextDelegate(TCPmsgReceive), new object[ ] { strReceived });  
            }  
        }  
    }  
}
```



btnConnect_Click()

```
private void btnConnect_Click(object sender, EventArgs e) {  
    try {  
        IPAddress = IPAddress.Parse(txtIPAddress.Text);  
        Client = new TcpClient( );  
        Client.Connect(IPAddress, LocalPort);  
        Connected = true;  
        stream = Client.GetStream( );  
        Reader = new StreamReader(stream);  
        Writer = new StreamWriter(stream);  
  
        txtRxMsg.AppendText( "Connected to Server!" + "WrWn");  
  
        ReceiveThread = new Thread(new ThreadStart(Receive));  
        ReceiveThread.Start( );  
    } catch (SocketException ex) {  
        MessageBox.Show(ex.Message.ToString( ), "TCP Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  
    }  
}
```



btnSend_Click()

```
private void btnSend_Click(object sender, EventArgs e) {  
    txtRxMsg.AppendText("Send : " + txtSendMsg.Text + "WrWn");  
  
    Writer.WriteLine(txtUser.Text + "," + txtSendMsg.Text);  
    Writer.Flush( );  
    txtSendMsg.Clear( );  
}
```

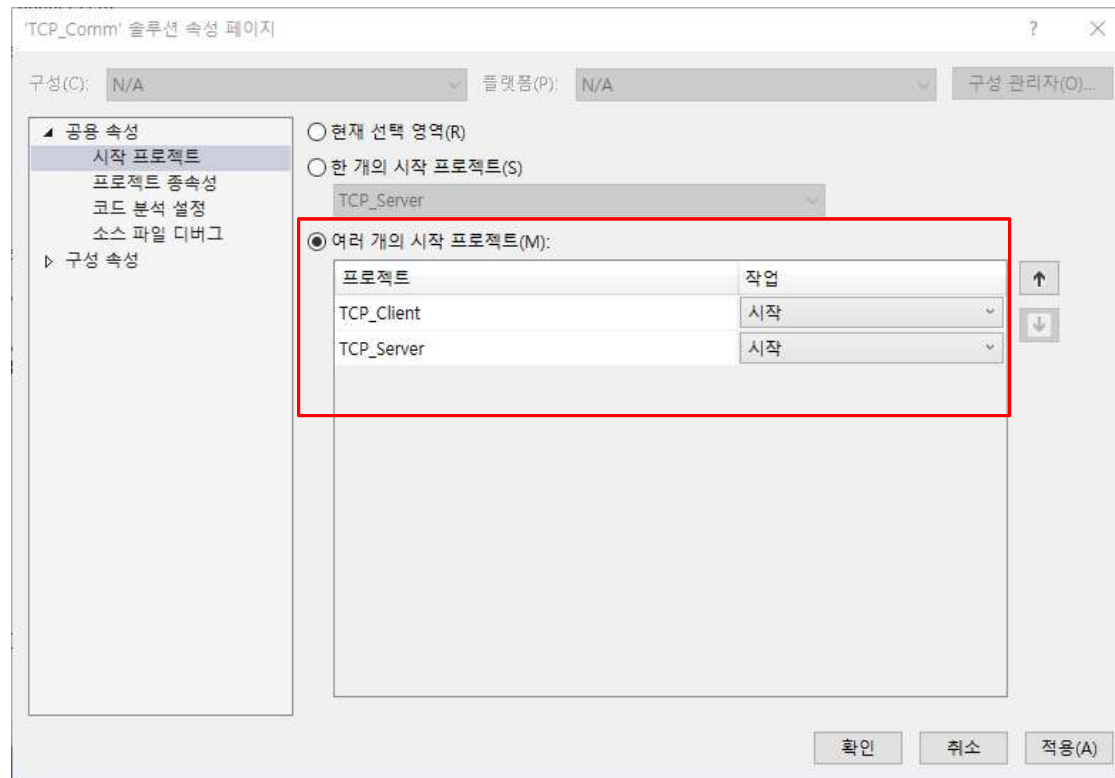


FormClosing()

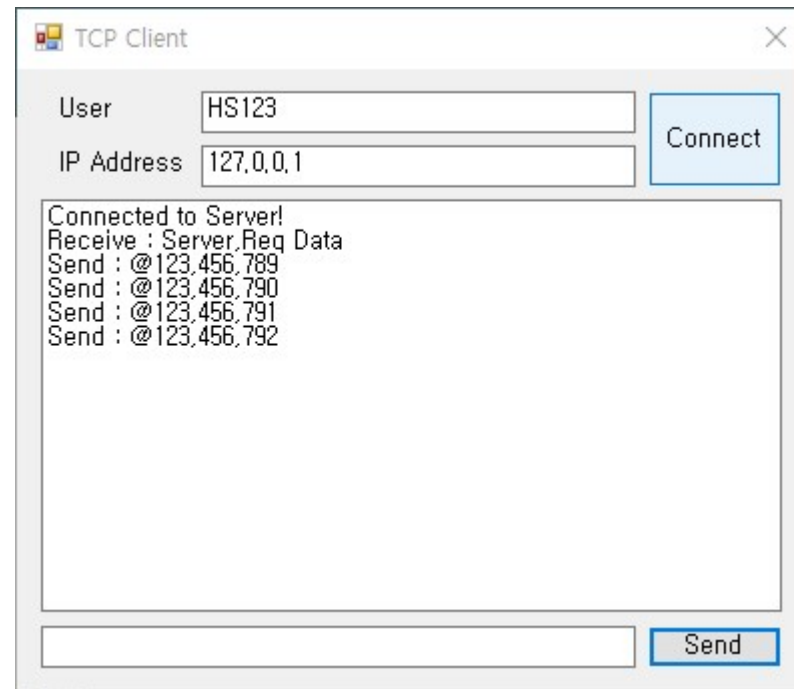
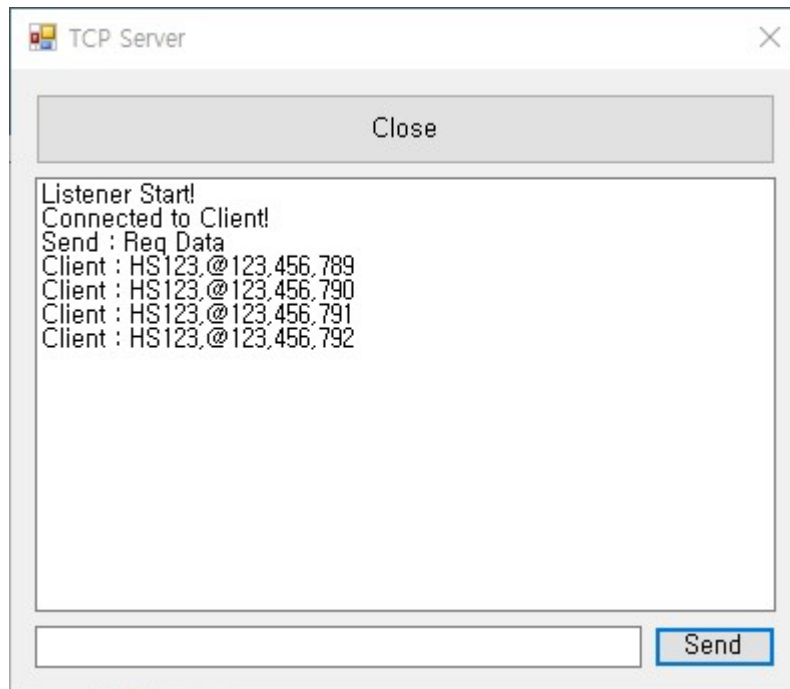
```
private void frmClient_FormClosing(object sender, FormClosingEventArgs e) {  
    Connected = false;  
    if (ReceiveThread != null) ReceiveThread.Abort( );  
    if (Reader != null) Reader.Close( );  
    if (Writer != null) Writer.Close( );  
    if (Client != null) Client.Close( );  
}
```



시작 프로젝트



Run



Thread

