

B&B ELECTRONICS **TECHNICAL NOTES**



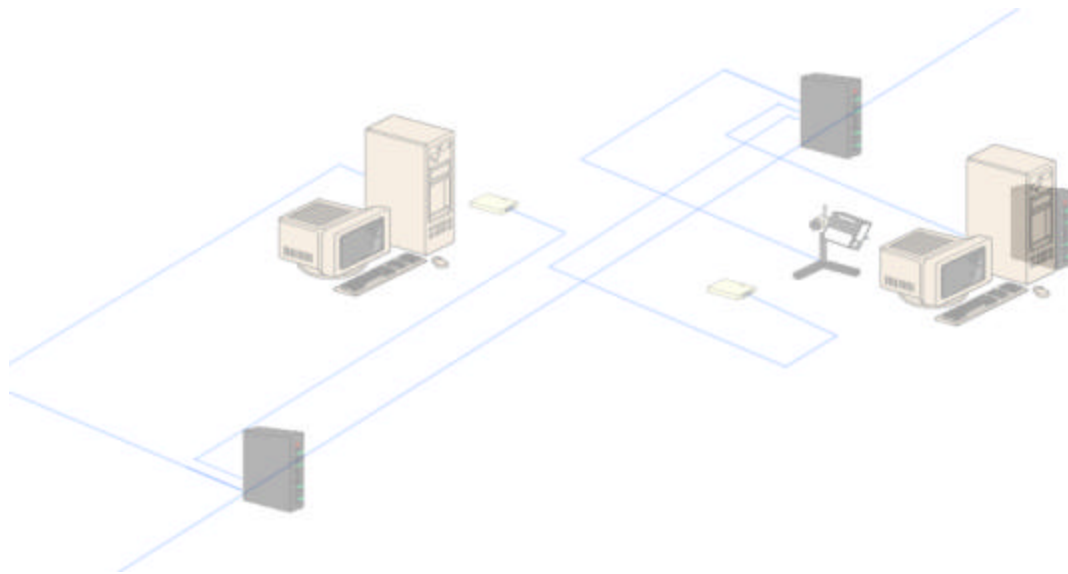
Vlinx™ Ethernet to Serial Servers

Scalable Designs Using Vlinx Ethernet Serial Servers

Design 1: Client Software Connections

Overview

For some of the people using the Vlinx Ethernet Serial Server in their solutions scalability is a key concern. They may be using only a dozen servers at this point but by the time the implementation is complete they could be using several hundred. It is these cases that we'll be shining the spotlight on in this article.



International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

BB-WP10

Getting Started

Scalability is important in any design. Nothing is more discouraging than having to re-engineer a system when it comes time to make small scale expansions. A well-designed solution would allow for new devices to be added without breaking the existing structure or taking too much time.

In this article we're going to use a fabricated company to demonstrate from start to finish one method of creating your scalable Ethernet serial server solution.

Maximum Industries

Hardwired serial communications throughout Maximum Industries facility would not be practical due to its large size. Since their monitoring equipment consisted of serial devices it was decided that using Industrial Ethernet would be the best method of moving data around the facility. For this task the Vlinx Ethernet Serial Servers was selected and the design work began. In the beginning the plans call for five cooling vats which each include a temperature monitor that outputs 232 data. Although a pretty simple setup, scalability was a huge factor as the plant is expected to grow to the point where they would be utilizing hundreds of these cooling vats.

The source code for this example will be provided as a reference. Please note that the sample program has been produced in Microsoft Visual Basic.

Client PC Connection

Five Vlinx Ethernet Serial Servers were used, each connected to one of the cooling vat temperature sensors. In addition to being connected serially to the sensors the Vlinx servers will also be connected to local LAN. The main goal is to allow the data from all of the sensors to be gathered, analyzed and logged by a single data PC located somewhere off the plant floor. The easiest way to connect all of these servers to the Data PC might be to setup a virtual COM port on the Data PC for each remote Vlinx server. Although it might seem to be the obvious choice at first this method has limitations when scalability is considered. The Microsoft Windows operating system limits the number of COM ports in the system to 256. In addition, many controls designed for serial communications have limits as low as 10 or 16.

So how do you get around this? The answer is a direct Winsock connection. A well designed system not only allows more than 256 clients but allows them to be added painlessly without jeopardizing the rest of the system.

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

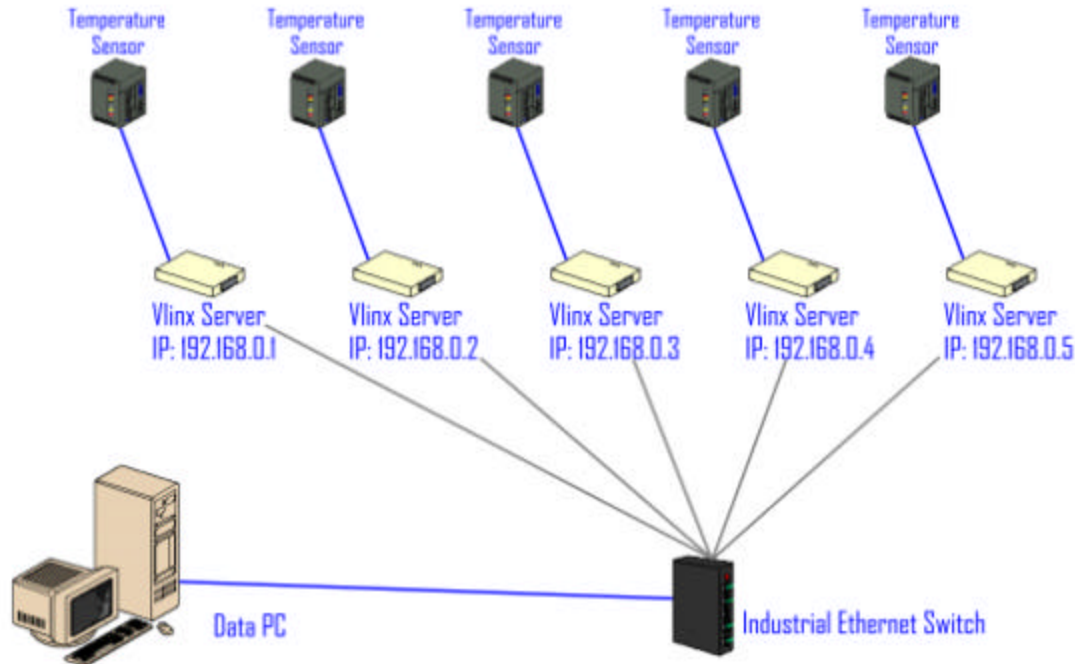


Diagram 1.1 – Basic Network Topology

From Diagram 1.1 you can see the basic network topology that Maximum Industries used to connect all of their temperature sensors to the Data PC. Note the linear IP addressing scheme that they have setup for the Vlinux Servers. It's pretty easy to see the next Vlinux Server added to the network will be given the address 192.168.0.6. This will be a very important part of the design, which we will get into later on in this example.

So how will all of these devices be configured? Right out of the box they are actually very close to where they need to be with a few exceptions.

- **IP Address** – needs to be set to correspond to it's location in the network layout
- **Baud Rate, Data/Parity/Stop, Flow Control** – without the virtual COM port connection setting are changed through hardware settings

The Client Software

Let's take a look at the client software. Visual Basic was chosen for this example because it allowed us the easiest means of demonstrating the theory without a lot of overhead code. The project consists of two forms and a module file. The entire project is downloadable from www.bb-elec.com.

- **frm_Main.frm** – This is the starting point of the application. It's also an MDI (Multiple Document Interface) parent. On startup this form loads itself before entering a loop, which is responsible for creating a set amount of frm_Connection.frm objects. This number is set within the mod_Globals.bas file by changing the NUM_VLINUX_SERVERS constant. The value is defaulted to 5 for this example.
- **frm_Connection.frm** – This is the form which is responsible for all of the socket communications. Most of the application code resides in this form. Please note that an instance of this form is created for each connection (NUM_VLINUX_SERVERS).

In addition to these two form files mod_Globals.bas is an important part of the project. This file contains the constants that define how the application will run. These values include how many servers to connect to and how long to wait before timing out on a connection.

Code Summary

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

BB-WP10

Let's go ahead and take a look at the source code behind the whole project. Most of it resides as event handlers under frm_Connection.frm but there is some small but very important work that goes on inside frm_Main.frm. Here's a look at the source code for frm_Main.frm.

```
Option Explicit
Private ConnectionForms(NUM_VLINX_SERVERS) As frm_Connection
*****
'Function: MDIForm_Load()
'Starting point of the sample
*****
Private Sub MDIForm_Load()
Dim counter As Integer

'Show all client connection windows
For counter = 1 To NUM_VLINX_SERVERS
'Create Connection Form Object
Set ConnectionForms(counter) = New frm_Connection

'Setup connection (IP, Local Port #)
Call ConnectionForms(counter).SetupConnection(IP_RANGE & Trim(Str(IP_START_OCTET + counter)), SOCKET_PORT_LOCAL + counter)

'Show Connection Object Form
ConnectionForms(counter).Show
Next counter
End Sub
```

Although there isn't much to this code it is very important when it comes to the overall scalability of the design. Note that the code creates a new instance of the connection form object for each Vlinx Ethernet Serial Server that it is to connect to.

```
Set ConnectionForms(counter) = New frm_Connection
```

These objects are child forms; each of which is responsible for the communications with its personal Vlinx Server. The number of new objects forms that the application creates is dictated by the following declaration, which can be found in mod_Globals.bas:

```
Public Const NUM_VLINX_SERVERS = 5
```

Looking at this code it now also becomes apparent why the linear pattern of IP addresses for the Vlinx servers was so important.

```
Call ConnectionForms(counter).SetupConnection(IP_RANGE & Trim(Str(IP_START_OCTET + counter)), SOCKET_PORT_LOCAL + counter)
```

At first glance this may look intimidating but broken down it should make perfect sense. It's a call to the SetupConnection() function inside of frm_Connection.frm. Since it's called from inside the loop it increments the IP address and the local port number for each connection that it makes to a Vlinx Server. The local port number is the Winsock port on the local system that is used to make the connection to the Vlinx Server. Only one connection at a time can be made through a socket port so this number needs to be incremented for every Vlinx Server. In our example these lines of code are called 5 times to open a connection to each of our five example Vlinx Ethernet Serial Servers.

Now that we have created our connection form objects its time to take a look at them and see what it is that they do. Initially when a new form object is created nothing actually happens, other than a new form (window) is added to the screen. At this point no communications take place. The connection isn't made until the user selects the "Connect" button on the form. Before we look at what happens when the connection is made we do need to look at what happens when the SetupConnection() function is called. You will remember that frm_Main.frm calls this function immediately after the new connection form object is created.

```
*****
'Function: SetupConnection()
'Called by frm_Main to initialize settings for this connection
*****
Public Function SetupConnection(szDestinationAddress As String, nLocalPort As Integer)
l_DestinationAddress = szDestinationAddress
l_LocalPort = nLocalPort
```

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

```
Me.Caption = I_DestinationAddress
```

```
'Setup Winsock Control Settings
```

```
WinsockConnection.RemoteHost = I_DestinationAddress
```

```
WinsockConnection.RemotePort = SOCKET_PORT_REMOTE
```

```
WinsockConnection.LocalPort = I_LocalPort
```

```
UpdateStatus (STATUS_DISCONNECTED)
End Function
```

Note that I_DestinationAddress and I_LocalPort are private variables to the form object. This function is another very simple one. Its job is to handle the initialization of the values needed to make a connection to its assigned Vlinx Ethernet Serial Server.

Now to find out what happens when you click on that “Connect” button. This code is slightly more involved than the rest of the functions and may require a little more time to go through.

```
*****
'Event Handler: bConnect_Click()
'Fired when the user clicks on the connect button...connects to the remote server
*****
Private Sub bConnect_Click()
'Check and make sure that the connection is disconnected before we try to connect
If Not (WinsockConnection.State = sckClosed) Then
    WinsockConnection.Close
End If

'Request connection
WinsockConnection.Connect

'Start stopwatch
StopwatchControl.StartTimer

'Wait for connection to complete
UpdateStatus (STATUS_CONNECTING)
Do While (WinsockConnection.State = sckConnecting)
    DoEvents

'Check for error condition
If (WinsockConnection.State = sckError) Then
    UpdateStatus (STATUS_ERROR)
    Exit Do
End If

'Check to see if we have been ordered to disconnect
If (WinsockConnection.State = sckClosed) Then
    Exit Do
End If

'Check to make sure that we haven't timed out yet
If (StopwatchControl.GetClock >= TIMEOUT_CONNECTION) Then
    UpdateStatus (STATUS_TIMEOUT)
    Exit Do
End If
Loop

If (WinsockConnection.State = sckConnected) Then
    UpdateStatus (STATUS_CONNECTED)
End If
End Sub
```

Just glancing at the code you can see that a whole lot of what’s going on in this function is checking the “State” of the Winsock connection. What this means is we are asking the Winsock ActiveX control that it is doing. This is a very important factor in deciding what we can do. Table 1.1 is a listing of the possible states which a Winsock connection can be in.

Constant	Value	Description
----------	-------	-------------

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

sckClosed	0	Default. Closed
sckOpen	1	Open
sckListening	2	Listening
sckConnectionPending	3	Connection pending
sckResolvingHost	4	Resolving host
sckHostResolved	5	Host resolved
sckConnecting	6	Connecting
sckConnected	7	Connected
sckClosing	8	Peer is closing the connection
sckError	9	Error

Table 1.1 – Winsock States

Note that after we make the actual connection request (`WinsockConnection.Connect`) we move into a loop and wait for the connection state to become “Connected”. We don’t want to leave the function while the state of the connection is **sckConnectionPending** as this misleads the application to think that we have connected successfully. There are a number of ways to get out of the loop, most of them bad. If you exit the function with a status of **sckConnected** your communications will move smoothly.

Once you have run the application you will see that you have a form (window) for each serial server in your design. Each has its own instance of that connect button, which will act the same way per its own Vlinx Server. So the mystery of how to design a scalable application/network solution is answered. Using this design of a linear addressing scheme coupled with an application providing dynamic connections to any number of serial servers you can allow your solution to grow without costly design changes. Currently in this design we have 5 serial servers. To add another one the following simple steps would need completed:

- Add the new serial server to the network
- Make sure the new server has an address falling into the networks linear addressing scheme. The next device should be setup to 192.168.0.6
- Change the number of servers constant declaration in `mod_Globals.bas` from `Public Const NUM_VLINX_SERVERS = 5` to `Public Const NUM_VLINX_SERVERS = 6`

Once this is complete the software should automatically know that we are now dealing with 6 devices with addresses ranging from **192.168.0.1** to **192.168.0.6**. This same process can be followed to add new servers.

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

Summary

With a well-designed network of Vlinx Ethernet Serial Servers you can have a network, which is not only easy to setup but is also very scalable. Questions welcome.

Jeremy Severson

B&B Electronics - 707 Dayton Road - Ottawa IL 61350

Contact me: JSeverson@bb-elec.com

International Headquarters:

B&B Electronics Mfg. Co. 707 Dayton Road P.O. Box 1040 Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

B&B Electronics Ltd Westlink Commercial Park Oranmore Co. Galway Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com

