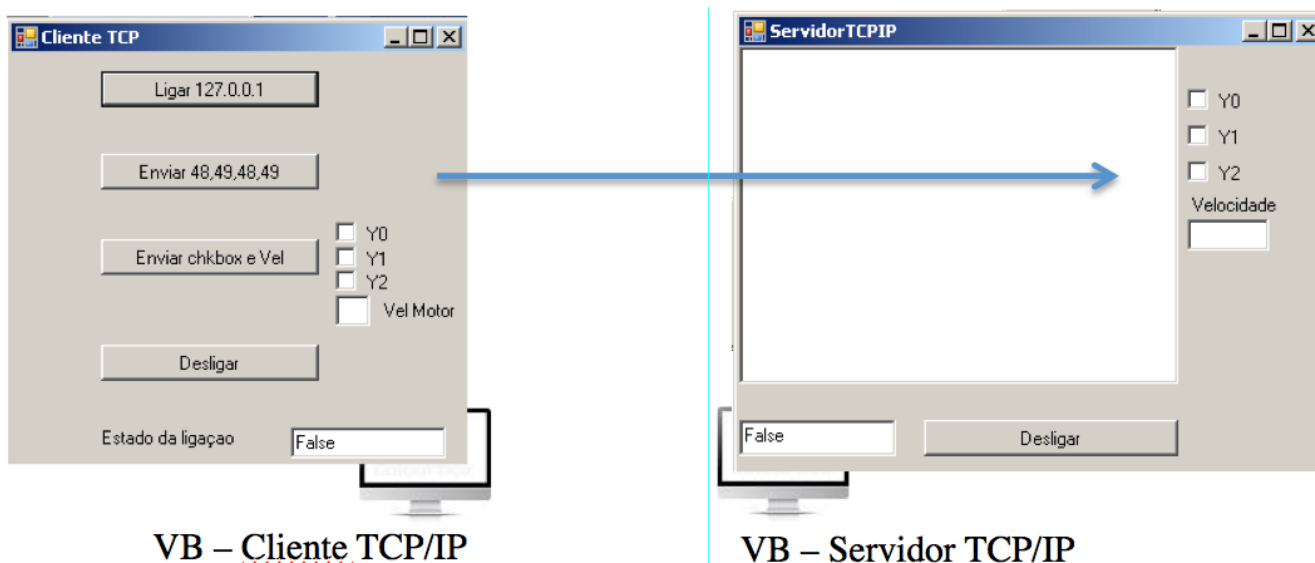




1. Introdução

Neste trabalho pretende-se enviar mensagens de texto entre dois programas Windows, desenvolvidos em VBasic. Um dos programas atua como cliente TCPIP e o outro como servidor TCPIP (ver Figura). Depois do programa servidor estar ativo, o programa cliente pode estabelecer uma ligação TCPIP entre ambos. Enquanto a ligação estiver estabelecida ambos os programas podem trocar dados entre si.



Comunicação entre dois programas, via TCPIP

Pretende-se apresentar objetos do tipo TcpClient, TcpListener e NetworkStream. Estes objetos são necessários para desenvolver em VBasic programas Windows que atuem como clientes e servidores TCP/IP, capazes de transmitir dados através da Internet.

2. Descrição do trabalho

A partir do programa cliente (lado esquerdo da figura) pretende-se selecionar as checkbox Y0, Y1, Y2 e escrever a velocidade na caixa de texto "VelMotor", quando o utilizador premir o botão "Enviar ChkBox e Vel" esses dados devem ser enviados para o programa servidor, visualizados nas checkbox e na caixa de texto "Velocidade" do servidor. Analise e implemente os exemplos apresentados no final deste documento.

Na aplicação servidora, crie 4 checkbox (X0,X1,X2,X3) e uma caixa de texto (VelMedida) .

Enquanto a ligação TCPIP estiver estabelecida (pela aplicação Cliente), o servidor deve enviar, de segundo a segundo, o estado das checkbox X0,X1,X2,X3, e a VelMedida para a aplicação cliente.

A aplicação Cliente deve mostrar ao utilizador o estado das checkbox, e a velocidade definida na aplicação servidora.

Client TCP

Ligar a 127.0.0.1

Desligar 127.0.0.1

Envio

☐ Y0

☐ Y1

☐ Y2

Velocidade do motor

Enviar

Recepção

☐ X0

☐ X1

☐ X2

☐ X3

Velocidade do motor

Connection status

Cliente TCP

Servidor TCP

Recepção

☐ Y0

☐ Y1

☐ Y2

Velocidade do motor

Envio

☐ X0

☐ X1

☐ X2

☐ X3

Velocidade do motor

Desligar

Connection status

Servidor TCP

3. Servidor TCPIP

```
1 Imports System.Net
2 Imports System.Net.Sockets
3
4 1 referência
5 Public Class Form1
6     'The ip address of the server
7     Dim local_address As IPAddress = IPAddress.Parse("127.0.0.1")
8
9     'A server object will accept connection requests
10    Dim server As New TcpListener(local_address, 81)
11
12    'A connection object will receive data
13    Dim connection As New TcpClient
14
15 0 referências
16 Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
17     'Start the TCP server. Will wait for clients to connect
18     server.Start()
19 End Sub
20
21 0 referências
22 Private Sub TimerReceive_Tick(sender As Object, e As EventArgs) Handles TimerReceive.Tick
23
24     If server.Pending() = True Then
25         connection = server.AcceptTcpClient()
26     End If
27
28     If connection.Connected Then
29         Dim message_size As Integer = connection.Available()
30
31         If message_size > 0 Then
32             Dim message_in_stream As NetworkStream = connection.GetStream()
33             Dim buffer(message_size) As Byte
34             message_in_stream.Read(buffer, 0, message_size)
35
36             Dim message_in As String = ""
37             Dim i As Integer
38             For i = 0 To message_size - 1
39                 message_in = message_in + Chr(buffer(i))
40             Next i
41
42             TextBoxReceber.Text = message_in + vbCrLf + TextBoxReceber.Text
43
44             CheckBoxY0.Checked = Mid(message_in, 1, 1)
45             CheckBoxY1.Checked = Mid(message_in, 2, 1)
46             CheckBoxY2.Checked = Mid(message_in, 3, 1)
47             TextBoxVelocidadeMotor.Text = Asc(Mid(message_in, 4, 1))
48         End If
49     End If
50 End Sub
51
52 0 referências
53 Private Sub TimerVerifyConnection_Tick(sender As Object, e As EventArgs) Handles TimerVerifyConnection.Tick
54     ToolStripStatusLabelConnectionStatus.Text = If(connection.Connected, "Connection established", "Not connected")
55 End Sub
56
57 0 referências
58 Private Sub TimerSend_Tick(sender As Object, e As EventArgs) Handles TimerSend.Tick
59     If connection.Connected = True Then
60         'Declare a byte array and set it according to the state of X0 to X3
61         Dim buffer(5) As Byte
62         buffer(0) = CheckBoxX0.CheckState + 48
63         buffer(1) = CheckBoxX1.CheckState + 48
64         buffer(2) = CheckBoxX2.CheckState + 48
65         buffer(3) = CheckBoxX3.CheckState + 48
66         buffer(4) = CInt(TextBoxVelocidadeMotorLida.Text)
67
68         'Now send the array
69         Dim message_out_stream As NetworkStream
70         message_out_stream = connection.GetStream()
71         message_out_stream.Write(buffer, 0, 5)
72     End If
73 End Sub
74 End Class
```

4. Cliente TCPIP

```
1 Imports System.Net
2 Imports System.Net.Sockets
3
4 1 referência
5 Public Class Form1
6     'The ip address of the server
7     Dim ip_address As IPAddress = IPAddress.Parse("127.0.0.1")
8
9     'A client object will request connection req
10    Dim client As New Sockets.TcpClient
11
12 0 referências
13 Private Sub ButtonLigar_Click(sender As Object, e As EventArgs) Handles ButtonLigar.Click
14     client.Connect(ip_address, 81)
15 End Sub
16
17 0 referências
18 Private Sub TimerReceive_Tick(sender As Object, e As EventArgs) Handles TimerReceive.Tick
19     'Update connection state
20     ToolStripStatusLabel.Text = If(client.Connected, "Connection established", "Not connected")
21
22     If client.Connected Then
23         Dim message_size As Integer = client.Available
24
25         'Read message if size is > 0
26         If message_size > 0 Then
27             'Copy stream to byte array buffer
28             Dim message_in_stream As NetworkStream = client.GetStream()
29             Dim buffer(5000) As Byte 'a buffer to copy the received data
30             message_in_stream.Read(buffer, 0, message_size)
31
32             'Convert byte array buffer to string message_in
33             Dim message_in As String = ""
34             Dim i As Integer
35             For i = 0 To message_size - 1 'copy from buffer to message_in
36                 message_in = message_in + Chr(buffer(i))
37             Next i
38
39             'Process message to get values for X0 to X3
40             CheckBoxX0.Checked = Mid(message_in, 1, 1)
41             CheckBoxX1.Checked = Mid(message_in, 2, 1)
42             CheckBoxX2.Checked = Mid(message_in, 3, 1)
43             CheckBoxX3.Checked = Mid(message_in, 4, 1)
44             TextBoxVelocidadeMotorLida.Text = Asc(Mid(message_in, 5, 1))
45         End If
46     End If
47 End Sub
48
49 0 referências
50 Private Sub ButtonDesligar_Click(sender As Object, e As EventArgs) Handles ButtonDesligar.Click
51     client.Close()
52     client = New Sockets.TcpClient
53 End Sub
54
55 0 referências
56 Private Sub ButtonEnviarCheckBox_Click(sender As Object, e As EventArgs) Handles ButtonEnviarCheckBox.Click
57     If client.Connected = True Then
58         'Declare a byte array and set it according to the state of X0 to X3
59         Dim buffer(5) As Byte
60         buffer(0) = CheckBoxY0.CheckState + 48
61         buffer(1) = CheckBoxY1.CheckState + 48
62         buffer(2) = CheckBoxY2.CheckState + 48
63         buffer(3) = CInt(TextBoxVelocidadeMotor.Text)
64
65         'Now send the array
66         Dim message_out_stream As NetworkStream
67         message_out_stream = client.GetStream()
68         message_out_stream.Write(buffer, 0, 4)
69     End If
70 End Sub
71 End Class
```