



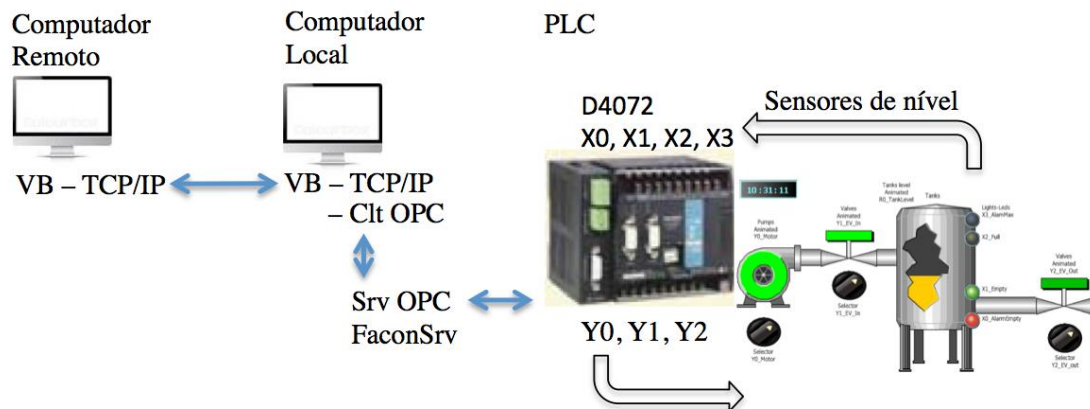
Informática Industrial 2021/2022

Trabalho prático Nº 8

Supervisão e controlo remoto de um PLC (TCP/IP e OPC)

Cliente TCP/IP para controlo de um PLC a partir de um Computador Remoto.

Neste trabalho (uma semana), pretende-se que um utilizador em qualquer ponto do mundo possa comunicar com um PLC e, dessa forma, controlar e monitorizar remotamente um processo (ex. reservatório de água).



Nos trabalhos anteriores, o controlo e a supervisão remota do PLC não podia exceder algumas centenas de metros, quando era utilizada uma ligação Rs232 entre o PLC e o computador local a partir do qual se efetuava a supervisão e o controlo.

Neste trabalho são utilizados dois computadores e um PLC (figura):

- Um dos computadores, o **computador local**, encontra-se **próximo do PLC e comunica com o PLC através de uma ligação Rs232**. A comunicação Rs232 utiliza localmente o controlo “SerialPort” ou como no último trabalho, utilizando o servidor OPC (FACONSrv).
- O segundo computador, **computador remoto**, utiliza uma **ligação TCP/IP** para comunicar com o computador local, e possui a interface de supervisão e controlo remoto do PLC (Reservatório de água).

Devem ser desenvolvidas duas aplicações em Visual Basic:

- A **aplicação local**, desenvolvida em Visual Basic, utiliza um objeto **TCPLListener (Servidor)** para receber ligações TCP/IP e um objeto do tipo cliente “**FaconSrv**” para comunicar com o PLC.
- A **aplicação remota** utiliza um objeto do tipo **TCPCClient** para comunicar com a aplicação local por **TCP/IP**, permitindo a supervisão e o controlo remoto das variáveis do processo. A **aplicação remota** deve ter uma interface que permita **visualizar o estado dos sensores** de nível do reservatório da figura, e o **estado das saídas** o estado do motor e das electroválvulas de entrada e de saída (figura seguinte).

Tal como nos trabalhos anteriores, o motor “**Y0_Motor**” e a electroválvula de entrada “**Y1_EV_In**” permitem encher o reservatório.

No reservatório existem **4 sensores de nível**:

- “X0_AlarmEmpty”,
- ”X1_Empty”,
- ”X2_Full”
- ”X3_AlarmMax”

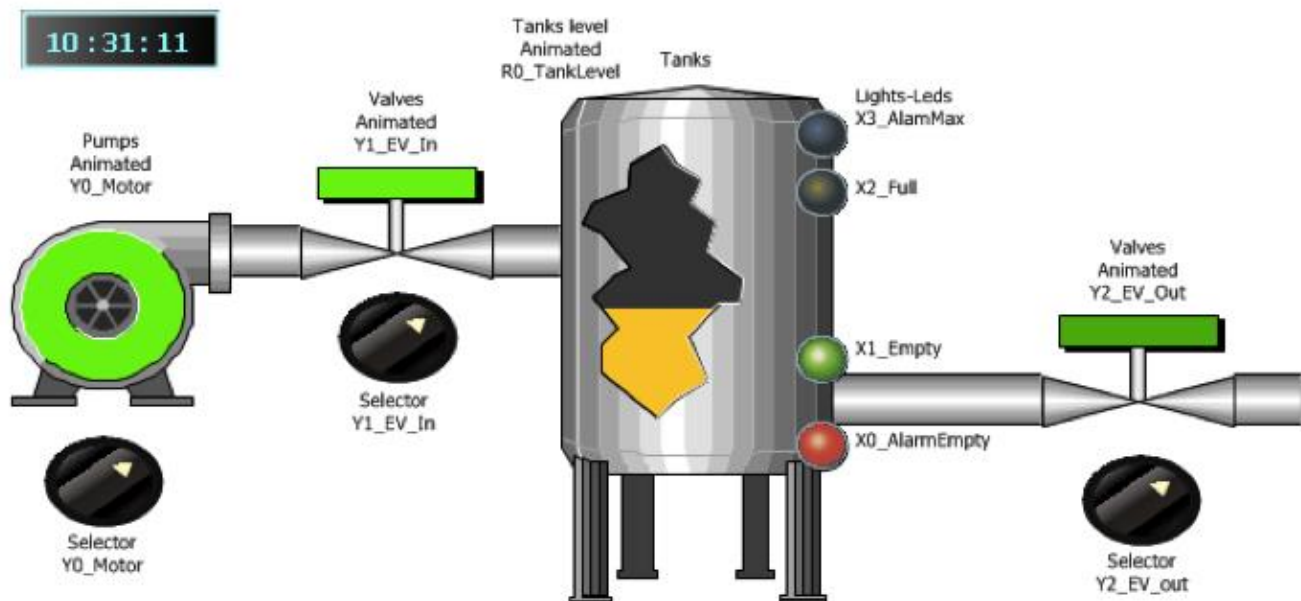
Assume-se que os sensores ficam ativos quando detetam água, a entrada digital respetiva recebe 24 V.

A electroválvula de saída “Y2_EV_Out” que permite a saída da água do reservatório, e ativada quando o nível de água no reservatório é superior ao nível “X1_Empty”.

O motor e a electroválvula de entrada são acionados quando o nível de água é inferior ao nível “X1_Empty” e desligadas quando o nível de água igualar ou exceder o nível “X2_Full”.

Gerar alarmes no écran do computador quando o nível de água atingir ou exceder o nível “X3_AlarmMax” ou passar a baixo do nível “X0_AlarmEmpty”.

As saídas digitais Y0, Y1 e Y2 do PLC permitem, respetivamente: ligar o motor de entrada, a electroválvula de entrada e a electroválvula de saída.



Conhecimentos a adquirir

Introdução às comunicações TCP/IP e a realização de Servidores/Clientes.

Familiarização com os objetos de comunicação TCP/IP:

- TcpClient
- TcpListener
- NetworkStream

Importante:

- Neste trabalho o PLC não tem programa, todo o controlo e supervisão é efetuado a partir do computador/VBasic.
- O trabalho será avaliado por questionário individual, numa semana seguinte à entrega do mesmo.

Bibliografia

- Slides e apontamentos sobre comunicações TCP/IP, com exemplos em VisualBasic
- [Steve 2004] **Practical Industrial Data Networks, Design, Installation and Troubleshooting**, Mackay Steve et.al , Elsevier, 2004.
- [Halsall] **Data Communications, Computer Networks and OSI**, Fred Halsall, Addison-Wesley. Cota:681.324G.25
- [Rosen 2003] **Web Application Architecture, Principles, Protocols and Practices**, Leon Shklar, Richard Rosen, John Wiley & Sons, 2003

Normas TCP / IP

IP protocol specification <http://www.rfc-editor.org/rfc/rfc791.txt>

TCP protocol specification <https://www.rfc-editor.org/rfc/rfc793.txt>

TCP/IP tutorial <https://tools.ietf.org/html/rfc1180>

01 - Understanding Local Area Networking (60 min)

LAN: Data, nodes, Client, Server, Peer, Net adapter, hub, switch, router, meio, Transport protocol, Largura de banda, wireless access point, serial, data rate transfer, IP address, Virtual LAN, Topologia, Ethernet, Frames, Tipos de Servidores

<https://www.youtube.com/watch?v=t9TmvFvYfWw>

04 - Understanding Internet Protocol (50 min)

IP address: classes, loopback, Binary to decimal, public and private address, DNS, Gateway, NAT, IPv4, IPv6, subnetting

https://www.youtube.com/watch?v=EkP4Ap_QQHc

OSI Model Explained (animated)

https://www.youtube.com/watch?v=O_rsqVtaloI

Hub, Switch or Router? Network Devices Explained (7 min)

https://www.youtube.com/watch?v=Ofjsh_E4HfY

How Network Address Translation Works (10 min)

<https://www.youtube.com/watch?v=QBqPzHEDzvo>

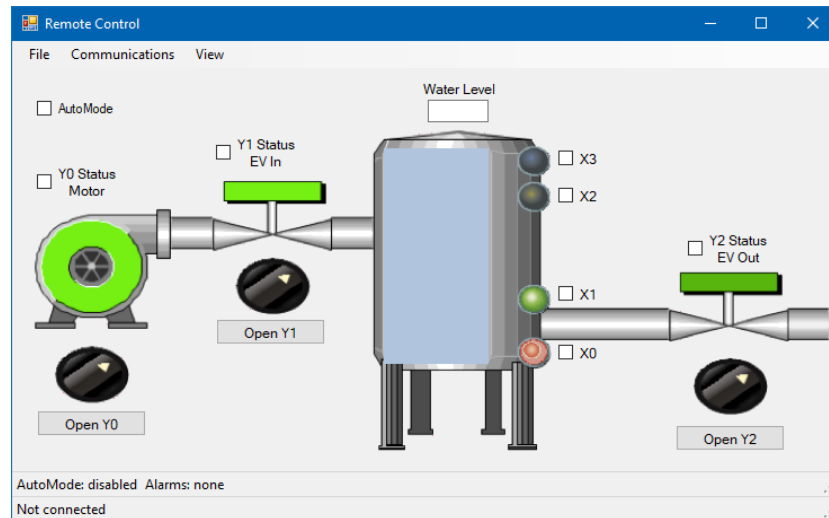
Automatic IP Address Assignment: How DHCP Works

<https://www.youtube.com/watch?v=RUZohsAxPxQ>

Inside the Domain Name System (13 min)

<https://www.youtube.com/watch?v=GlZC4Jwf3xQ>

Anexo – Programa CLIENTE



```

Private Sub TimerSend_Tick(sender As Object, e As EventArgs) Handles TimerSend.Tick
    Dim bb() As Byte = {48, 49, 48, 49, 48}

    bb(0) = Convert.ToInt32(Command.Y0) + 48
    bb(1) = Convert.ToInt32(Command.Y1) + 48
    bb(2) = Convert.ToInt32(Command.Y2) + 48

    If client.Connected = True Then
        Dim stream As NetworkStream = client.GetStream()
        stream.Write(bb, 0, 3)
    End If
End Sub

```

```

Private Sub TimerReceive_Tick(sender As Object, e As EventArgs) Handles TimerReceive.Tick
    If client.Connected Then
        Dim availableBytesToRead = client.Available

        'ler availableBytesToRead
        If availableBytesToRead > 0 Then
            Dim stream As NetworkStream = client.GetStream()

            Dim buffer(availableBytesToRead) As Byte
            stream.Read(buffer, 0, availableBytesToRead)

            'Converter os caracteres para string e colocar na variável texto_recebido
            Dim i As Integer
            Dim texto_recebido = ""
            For i = 0 To availableBytesToRead - 1
                texto_recebido = texto_recebido + Chr(buffer(i))
            Next i

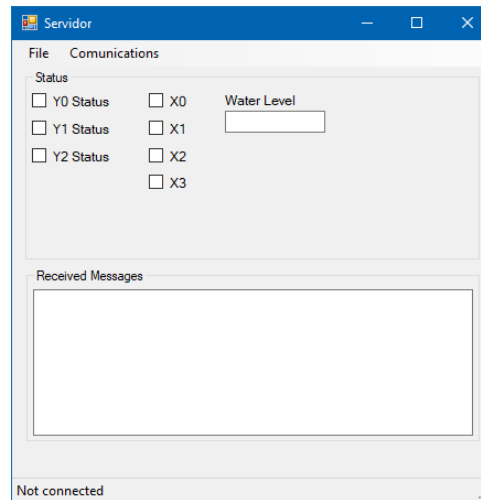
            Status.X0 = Mid(texto_recebido, 1, 1)
            Status.X1 = Mid(texto_recebido, 2, 1)
            Status.X2 = Mid(texto_recebido, 3, 1)
            Status.X3 = Mid(texto_recebido, 4, 1)
            Status.Y0 = Mid(texto_recebido, 5, 1)
            Status.Y1 = Mid(texto_recebido, 6, 1)
            Status.Y2 = Mid(texto_recebido, 7, 1)

            CheckBoxStatusY0.Checked = Status.Y0
            CheckBoxStatusY1.Checked = Status.Y1
            CheckBoxStatusY2.Checked = Status.Y2
            CheckBoxStatusX0.Checked = Status.X0
            CheckBoxStatusX1.Checked = Status.X1
            CheckBoxStatusX2.Checked = Status.X2
            CheckBoxStatusX3.Checked = Status.X3

        End If
    End If
End Sub

```

Anexo – Programa Servidor



```

Private Sub Timer_Tick(sender As Object, e As EventArgs) Handles Timer.Tick
    'Accept a connection if there is a pending request
    If server.Pending() = True Then
        connection = server.AcceptTcpClient()
    End If

    'Receive command from the client
    If connection.Connected Then
        Dim message_size As Integer = connection.Available
        Dim stream As NetworkStream = connection.GetStream()

        'Read message if size is > 0
        If message_size > 0 Then
            'Copy stream to byte array buffer

            Dim buffer(5000) As Byte 'a buffer to copy the received data
            stream.Read(buffer, 0, message_size)

            'Convert byte array buffer to string message_in
            Dim message_in As String = ""
            Dim i As Integer
            For i = 0 To message_size - 1 'copy from buffer to message_in
                message_in = message_in + Chr(buffer(i))
            Next i

            ReceivedMessagesTextBox.Text = message_in + vbCrLf + ReceivedMessagesTextBox.Text

            'Auxiliar variables
            Dim Y0, Y1, Y2 As String

            'Process message to get values for Y0, Y1, Y2
            Y0 = Mid(message_in, 1, 1)
            Y1 = Mid(message_in, 2, 1)
            Y2 = Mid(message_in, 3, 1)

            'Send received message to PLC
            fs.SetItem(CSG, "Y0", Y0)
            fs.SetItem(CSG, "Y1", Y1)
            fs.SetItem(CSG, "Y2", Y2)

        End If

    End If

    'Read the status from the PLC and send to the client
    CheckBoxStatusX0.CheckState = fs.GetItem(CSG, "X0")
    CheckBoxStatusX1.CheckState = fs.GetItem(CSG, "X1")
    CheckBoxStatusX2.CheckState = fs.GetItem(CSG, "X2")
    CheckBoxStatusX3.CheckState = fs.GetItem(CSG, "X3")
    CheckBoxStatusY0.CheckState = fs.GetItem(CSG, "Y0")
    CheckBoxStatusY1.CheckState = fs.GetItem(CSG, "Y1")
    CheckBoxStatusY2.CheckState = fs.GetItem(CSG, "Y2")

    If connection.Connected Then
        Dim status(7) As Byte

        status(0) = CheckBoxStatusX0.CheckState + 48
        status(1) = CheckBoxStatusX1.CheckState + 48
        status(2) = CheckBoxStatusX2.CheckState + 48
        status(3) = CheckBoxStatusX3.CheckState + 48
        status(4) = CheckBoxStatusY0.CheckState + 48
        status(5) = CheckBoxStatusY1.CheckState + 48
        status(6) = CheckBoxStatusY2.CheckState + 48

        Dim stream As NetworkStream = connection.GetStream()
        stream.Write(status, 0, status.Length)
    End If
End Sub

```