Comunicacións industriais

Práctica 02: Comunicación serie cun S7 200 en modo Freeport

Descrición da práctica:

Neste caso faremos unha comunicación serie entre o PC e un S7 200 de Siemens, configurando o PLC en modo Freeport.

Coñecementos previos:

- Configuración e funcionamento do porto serie. Normas RS 232 e RS 485. Hai multitude de documentación sobre estes dous protocolos, polo que será convinte antes de abordar esta práctica, traballar a teoría sobre os mesmos.
- Manexo básico dun S7 200. Comunicación desde Microwin co mesmo.
- Programación en CSharp.

Material necesario:

- PC con Visual CSharp 2017, Office e Microwin instalados.
- Autómata S7 200.

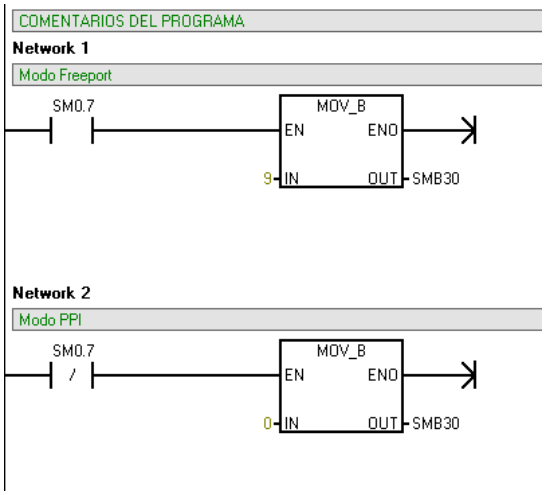
Especificacións:

- Desde o PLC enviarémoslle catro bytes ao PC que representarán, por exemplo, os valores de catro temperaturas comprendidas entre 0 e 255. Non se traballará con números en coma flotante, polo que as temperaturas serán datos numéricos enteiros (non nos preocupa de momento ter precisión nas lecturas).
- Na aplicación existirán dous formularios, un principal e un de configuración dos parámetros da comunicación.
- Disporemos dun sistema de ficheiro **.xml** para gardar a última configuración válida da comunicación e, deste xeito non ter que estar a modificala cada vez que se estableza unha nova.
- Enviaremos a un documento de Excel os datos dos catro termómetros.
- No S7 200 faremos un programa básico para enviar os catro valores a visualizar.

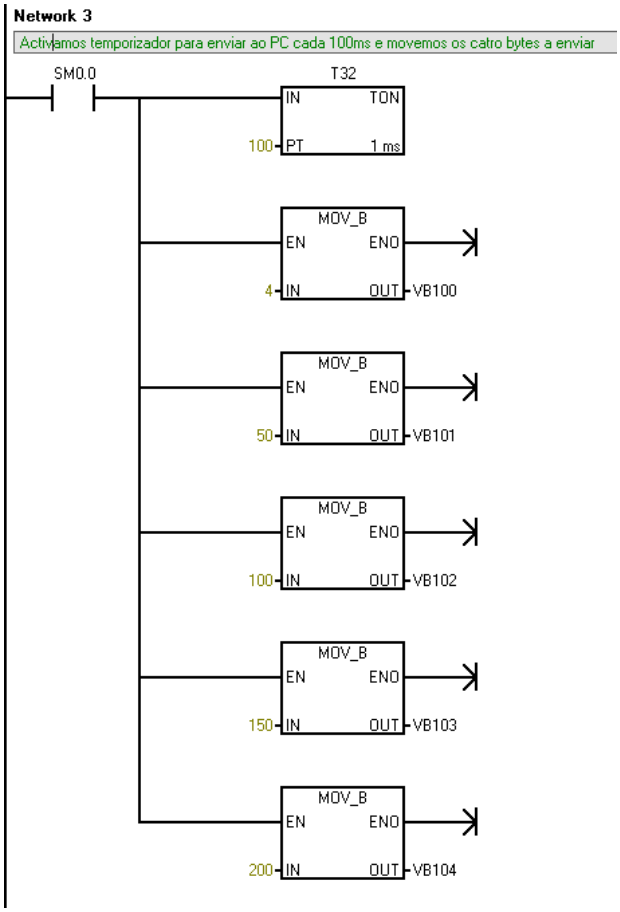
Solución:

O programa do autómata é simple.

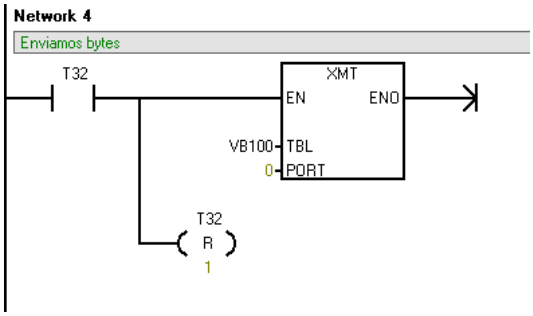
Utilizamos o selector de modos do PLC para poñer o autómata en modo Freeport ou en modo PPI.



Activamos o temporizador e colocamos os catro bytes a enviar.

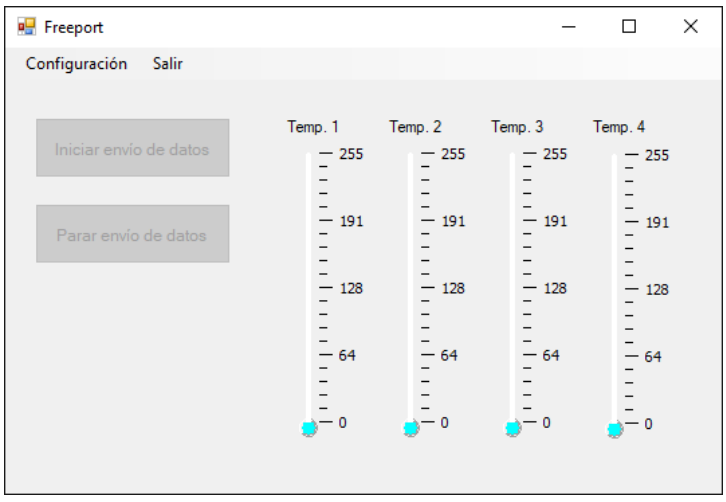


E enviamos.

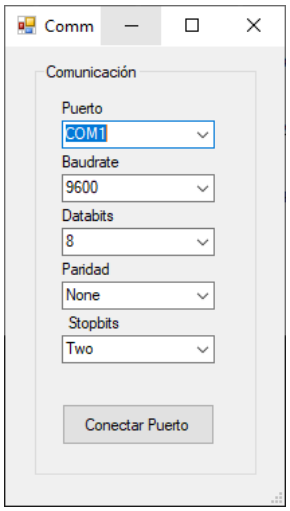


Agora imos con CSharp. O aspecto dos formularios será algo parecido ao seguinte:

Formulario principal:



Formulario de configuración:



Desde o programa principal (que arrancará cos botóns Iniciar envío de datos e Parar envío de datos desactivados, mentres non se conecte co autómata) chamaremos ao de configuración:

```
private void Btn_Config_Click(object sender, EventArgs e)
{
    frm_principal = this;//frm_principal = o formulario actual
    //Creamos en memoria o segundo formulario
    Frm_Secundario frm_secundario = new Frm_Secundario();
    frm_secundario.ShowDialog();//Amosa o segundo formulario, bloqueando o primeiro
}
```

As variables globais que temos neste caso son:

```
//Crear variable de tipo Frm_Principal
public static Frm_Principal frm_principal = null;

MyFunc funciones = new MyFunc();
AxiThermometerX[] temp = new AxiThermometerX[4];//Declara a matriz de termómetros

//Declara a matriz de bytes empregados na recepción con Read()
byte[] BufferRecepcion = new byte[4];

//Para traballar con Excel
string ruta = Application.StartupPath;
string archivo;

Microsoft.Office.Interop.Excel.Application xlApp;
Microsoft.Office.Interop.Excel.Workbook xlWorkBook;
Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;

object misValue = System.Reflection.Missing.Value;

int posicion = 2;//Para marcar posición nas filas do Excel
```

Ao principio do programa engadimos a seguinte liña para poder recoller datos polo porto serie.

```
public Frm_Principal()
{
    InitializeComponent();
    //A seguinte liña imprescindible para recoller caracteres polo porto serie
    Puerto.DataReceived += new
System.IO.Ports.SerialDataReceivedEventHandler(Recepcion);
}
```

No formulario principal, ao arrancar enchemos a matriz de termómetros cos catro que temos.

```
        private void Frm_Principal_Load(object sender, EventArgs e)
        {
            for (int i = 0; i < 4; i++)
            {
                    //Inicializa a matriz de termómetros
                    temp[i] = (AxiThermometerX)this.Controls["Temp" + Convert.ToString(i)];
              }
        }
```

Creamos o evento **Recepcion** para recoller a información que entra polo porto serie e chamamos ao evento **Actualizar** para que os procese.

```
        private void Recepcion(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            //Leer tantos bytes como lonxitude teña o buffer de recepción
            Puerto.Read(BufferRecepcion, 0, (int)(BufferRecepcion.Length));
            this.Invoke(new EventHandler(Actualizar));
        }



        private void Actualizar(object s, EventArgs e)
        {
            for(int i=0; i < 4; i++)
            {
                //Coloca o valor de cada byte no correspondente termómetro
                temp[i].Position = BufferRecepcion[i];
            }
        }
```

No botón de Inicio de envío creamos o documento de Excel e inicializámolo. Fixémonos que, neste caso créase un documento de cada vez cun nome que ven dado polo ano, o mes, o día, a hora, o minuto e o segundo no que empeza a toma de datos.

```
        private void Btn_InicioEnvio_Click(object sender, EventArgs e)
        {

            //Proceso para enviar un dato a la hoja de cálculo

            xlApp = new Microsoft.Office.Interop.Excel.Application();
            xlWorkBook = xlApp.Workbooks.Add(misValue);

            xlWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
            xlWorkSheet.Cells[1, 1] = "Fecha:";
            xlWorkSheet.Cells[1, 2] = "Hora:";
            xlWorkSheet.Cells[1, 3] = "Temp 1:";
            xlWorkSheet.Cells[1, 4] = "Temp 2:";
            xlWorkSheet.Cells[1, 5] = "Temp 3:";
            xlWorkSheet.Cells[1, 6] = "Temp 4:";

            archivo = "\\Excel\\" + DateTime.Now.ToString("yyyy_MM_dd_HH_mm_ss") + ".xls";

            //En C# poñemos dúas veces \\ para que non interprete como
            //unha secuencia de escape
            //A seguinte liña emprégase a primeira vez que temos que crear
            //o documento de Excel
            xlWorkBook.SaveAs(ruta + archivo,
Microsoft.Office.Interop.Excel.XlFileFormat.xlWorkbookNormal,
                misValue, misValue, misValue, misValue,
                Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive,
                misValue, misValue, misValue, misValue, misValue);
```

```
            //Para ter visible o documento de Excel mentres se executa
            xlApp.Application.Visible = true;
            Temp_1.Enabled = true;//Activa o temporizador
        }
```

No botón de Parar, paramos o envío a Excel.

```
        private void Btn_PararEnvio_Click(object sender, EventArgs e)
        {
            //Proceso para liberar o documento de Excel
            xlWorkBook.Close(true, misValue, misValue);
            xlApp.Quit();
            releaseObject(xlWorkSheet);
            releaseObject(xlWorkBook);
            releaseObject(xlApp);

            Temp_1.Enabled = false;//Desactiva o temporizador
            posicion = 2;
        }

        private void releaseObject(object obj)//Método para liberar o documento de Excel
        {
            try
            {
                System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);
                obj = null;
            }
            catch (Exception ex)
            {
                obj = null;
                MessageBox.Show("Ocorreu unha excepción ao liberar o documento " +
ex.ToString());
            }
            finally
            {
                GC.Collect();
            }
        }
```

So nos queda programar o temporizador para que estea a enviar os datos ao documento de Excel. Fixémonos como collemos o nome do arquivo coas variables **ruta+archivo**.

```
        private void Temp_1_Tick(object sender, EventArgs e)
        {
            object misValue = System.Reflection.Missing.Value;

            xlWorkBook = xlApp.Workbooks.Open(ruta + archivo, 0, false, 5, null, null,
false, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows, null, true, false, 0, true,
false, false);
xlWorkSheet = (Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
//Poñemos a data
xlWorkSheet.Cells[posicion, 1] = DateTime.Now.ToString("dd/MM/yyyy");
//Poñemos a hora
xlWorkSheet.Cells[posicion, 2] = DateTime.Now.ToString("HH:mm:ss")
xlWorkSheet.Cells[posicion, 3] = temp[0].Position; //Poñer o dato do termómetro_1
xlWorkSheet.Cells[posicion, 4] = temp[1].Position; //Poñer o dato do termómetro_2
xlWorkSheet.Cells[posicion, 5] = temp[2].Position; //Poñer o dato do termómetro_3
xlWorkSheet.Cells[posicion, 6] = temp[3].Position; //Poñer o dato do termómetro_4
xlWorkBook.Save();

            posicion++;
        }
```

Agora programamos o formulario de comunicación.

Ao arrancar recollemos datos do ficheiro .xml de configuración.

```
        private void Frm_Secundario_Load(object sender, EventArgs e)
        {
            //Declaramos unha variable matriz de cadeas para conter os nomes dos portos que
se atopen no ordenador
            string[] portos = SerialPort.GetPortNames();
            //Colocamos no combo a matriz de portos atopada
            foreach (string port in portos)
            {
                Cmb_Puerto.Items.Add(port);
            }
            Cmb_Puerto.SelectedIndex = 0;

            Cmb_Paridad.DataSource = Enum.GetValues(typeof(System.IO.Ports.Parity));
            Cmb_Stopbits.DataSource = Enum.GetValues(typeof(System.IO.Ports.StopBits));


            if(File.Exists(Application.StartupPath + "\\config.xml"))//Control de error por
si no existe config.xml
            {
                XDocument xdoc = XDocument.Load(Application.StartupPath + "\\config.xml");
                var elementos = from dato in xdoc.Descendants() select dato;
                foreach (var dato in xdoc.Descendants("Configuracion"))
                {
                    Cmb_Puerto.SelectedIndex = Convert.ToByte(dato.Element("Puerto").Value);
                    Cmb_Baudrate.SelectedIndex =
Convert.ToByte(dato.Element("Baudrate").Value);
                    Cmb_Databits.SelectedIndex =
Convert.ToByte(dato.Element("Databits").Value);
                    Cmb_Paridad.SelectedIndex =
Convert.ToByte(dato.Element("Paridad").Value);
                    Cmb_Stopbits.SelectedIndex =
Convert.ToByte(dato.Element("Stopbits").Value);
                }
            }
        }
```

E no botón de conectar, asignamos os parámetros desexados ao porto e gardamos os novos datos no documento .xml.

```
private void Btn_Conectar_Click(object sender, EventArgs e)
        {
            this.Close();

            Frm_Principal.frm_principal.Btn_InicioEnvio.Enabled = true;
            Frm_Principal.frm_principal.Btn_PararEnvio.Enabled = true;

            Frm_Principal.frm_principal.Puerto.Close();
            Frm_Principal.frm_principal.Puerto.PortName = Cmb_Puerto.Text;
            Frm_Principal.frm_principal.Puerto.BaudRate =
Convert.ToInt16(Cmb_Baudrate.Text);
            Frm_Principal.frm_principal.Puerto.DataBits =
Convert.ToInt16(Cmb_Databits.Text);
            Frm_Principal.frm_principal.Puerto.Parity =
(System.IO.Ports.Parity)Cmb_Paridad.SelectedItem;
            Frm_Principal.frm_principal.Puerto.StopBits =
(System.IO.Ports.StopBits)Cmb_Stopbits.SelectedItem;
            Frm_Principal.frm_principal.Puerto.Open();

            XmlDocument doc = new XmlDocument();
            XmlElement raiz = doc.CreateElement("Configuracion");
            doc.AppendChild(raiz);
            XmlElement puerto = doc.CreateElement("Puerto");

puerto.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Puerto.SelectedIndex)));
```

```
            raiz.AppendChild(puerto);
            XmlElement baudrate = doc.CreateElement("Baudrate");

baudrate.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Baudrate.SelectedIndex)));
            raiz.AppendChild(baudrate);
            XmlElement databits = doc.CreateElement("Databits");

databits.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Databits.SelectedIndex)));
            raiz.AppendChild(databits);
            XmlElement paridad = doc.CreateElement("Paridad");

paridad.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Paridad.SelectedIndex)));
            raiz.AppendChild(paridad);
            XmlElement stopbits = doc.CreateElement("Stopbits");

stopbits.AppendChild(doc.CreateTextNode(Convert.ToString(Cmb_Stopbits.SelectedIndex)));
            raiz.AppendChild(stopbits);
            doc.Save(Application.StartupPath + "\\config.xml");
        }
```

Para que funcione todo isto, temos que engadir as librerías seguintes no formulario de configuración.

```
using System.IO.Ports;
using System.IO;
using System.Xml;
using System.Xml.Linq;
using System.Linq;
```

As dúas primeiras para o porto serie e as tres últimas para o ficheiro **.xml**