

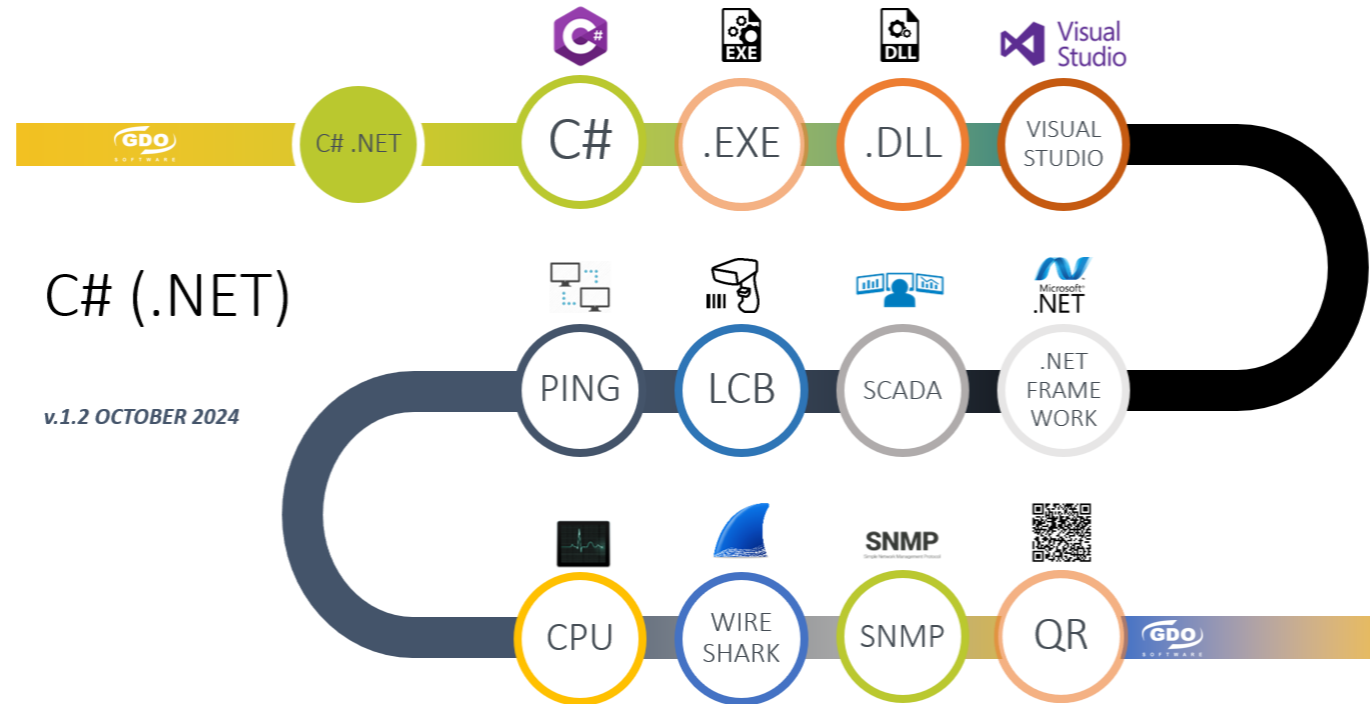
CHAPTER 3. C# (.NET)

v.1.2 OCTOBER 2024



Ricardo Moraleda Gareta

[Director of the software development at GDO Software]





C# .NET



C#



.EXE



.DLL



VISUAL STUDIO

C# (.NET)

v.1.2 OCTOBER 2024



PING



LCB



SCADA



.NET FRAMEWORK



CPU



WIRE SHARK

SNMP
Simple Network Management Protocol

SNMP



QR





C# (.NET)



C# (C sharp)

It is an object-oriented programming language developed and standardized by Microsoft as part of its .NET platform.

Its basic syntax is derived from C/C++ and uses the .NET platform object model, similar to that of Java, although it includes improvements derived from other languages.

<https://docs.microsoft.com/es-es/dotnet/csharp/>

In this tutorial I am not going to teach you how to program C#, just examples oriented to industrial control or typical functionalities in this field both .EXE and .DLL (classes library or user control)

Industrial Control



1. SCADA control of temperature probe and cold values of wine tanks with concurrent tasks and through the Modbus TCP protocol.

2. Barcode reading driver using Datalogic USB reader.



3. Ping – ConsoleApplication



4. Monitor PC CPU and log to CSV



5. SNMP protocol reading driver



For this I will use the Visual Studio 2015 IDE and the .NET 4.5.2 framework

<https://visualstudio.microsoft.com/es/free-developer-offers/>
(Community version is free)



SCADA (Modbus TCP)



Control

Comunicación OK
Habilitado

ID: 18
Descripción: 253
Zona: FILA2
IP: 192.168.10.110

Manual (0) / Auto (1)

Controlador Temperatura

PV: 18.1
Set Point: 17.5

Limite Superior SP (°C): 1
Calib. sonda (°C): -1.7

D. Inicial: 1081 g/L D. Actual: 1075 g/L
CHARD ECO CP
CHCP190819

DEN_MAX	DEN_MIN	SP
1083	1059	17.5
1058	1016	14.5
1015	1005	16.5
1004	992	17.5

MENSAJE	DESCENSO_DENSIDAD
OXIGEN 5 mg	20
NUTRICIO + ...	22
CLARIFICACIÓ	30

FECHA DEN DIF TEMP

21/08/2019 9:17	1081		
22/08/2019 8:30	1079		14.0
23/08/2019 8:25	1077	2	13.8
24/08/2019 9:42	1075	2	15.6

Alarmas = 2

18/09/2019 23:46:12

Comunicaciones

250	17,9 °C	SP: 15 °C	D: 1087 g/L	FRESC 1086 (12)	XAR GRAU
251	17,9 °C	SP: 15 °C	D: 1092 g/L	FRESC 1086 (12)	XAR GRAU
252	17,9 °C	SP: 15 °C	D: 1029 g/L	FRESC 1086 (12)	XAR GRAU
253	16,9 °C	SP: 14 °C	D: 1045 g/L	FERM 1073	MCB 1
254	15,5 °C	SP: 14 °C	D: 1024 g/L	FERM 1073	XAR 1
255	15,8 °C	SP: 17,5 °C	D: 1086 g/L	FRESC 1086 (12)	XAR GRAU
256	18,8 °C	SP: 17,5 °C	D: 1083 g/L	FRESC 1086 (12)	XAR GRAU
257	16,8 °C	SP: 17,5 °C	D: 1082 g/L	FRESC 1073 (10)	MCB 1
258	16,8 °C	SP: 14 °C	D: 1045 g/L	FERM 1073	XAR 1
259	15,9 °C	SP: 14 °C	D: 1024 g/L	FERM 1079	MAC 1
260	22 °C	SP: 20 °C	D: 8 g/L	DESACTIVADO	DESACTIVADO
261	19,5 °C	SP: 18 °C	D: 1001 g/L	FERM 1066 (B)	XAR MAC
262	16,4 °C	SP: 14 °C	D: 999 g/L	FERM 1079	XAR MAC
263	15,7 °C	SP: 14 °C	D: 1044 g/L	FERM 1086 (12)	MAC GRAU
264	14,8 °C	SP: 14 °C	D: 1017 g/L	FERM 1079	MAC GRAU
265	14,6 °C	SP: 14 °C	D: 1040 g/L	FERM 1079	MCB 1

FECHA_INICIO DEPOSITO TEMPERATURA ESTADO_ALARMA MENSAJE

18/09/2019 23:45:41	267	20.4	UNACK_ALM	TEMPERATURA FUERA DE MARGEN LO/HI
18/09/2019 18:14:39	248		UNACK_ALM	TEMPERATURA FUERA DE MARGEN LO/HI



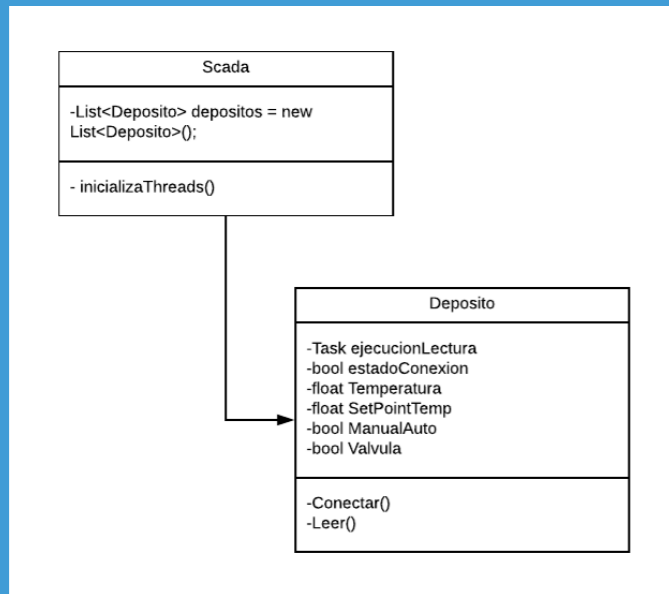
C# (.NET)



SCADA control wine tanks

Application with WindowsForm (.EXE) and SQL database. In this part we will focus on C# in a summarized and schematic way.

The Scada class is static and contains a List of type Deposit previously initialized from the DB..



Calling the initializeThreads() method generates as many Tasks as there are buckets and initializes them.

The task refers to the Read() method of the Deposit class.

```

public static void inicializaThreads()
{
    try
    {
        foreach (Deposito deposito in Scada.depositos)
        {
            if (deposito.habilitado)
            {
                deposito.ejecucionLectura = new Task(() => deposito.Leer());
                deposito.ejecucionLectura.Start();
                Thread.Sleep(1009);
            }
        }
    }
    catch (Exception e)
    {
        Scada.Log.Error("Error al inicializar threads. " + e.ToString());
    }
}
  
```



C# (.NET)



The Leer() method of Deposit contains an infinite loop or while(true) where it calls a data reading method **readMultipleWords()** by ModBus TCP passing the connection or socket, the position to read and the amount of data.

This function reads 12 memory locations from the Modbus TCP server in one go for each configured scan cycle – Sleep(scan).

```

public void Leer()
{
    try
    {
        while (true)
        {
            int[] datosLeidos = new int[12]; //temperatura(2), setpoint temp (2), manualauto (1), abrircerrar valvula (1), cpu (1), ram (1), tempup (2), tempdown (2)
            int[] PV = new int[2];
            int[] SP = new int[2];
            int[] SPUPLEVEL = new int[2];
            int[] SPDOWLEVEL = new int[2];

            if (estadoConexion)
            {
                //lee datos del iot por modbus tcp
                datosLeidos = Scada.factoriaModbus.readMultipleWords(this.serverSocket, 1, Convert.ToInt32(this.wordTempPV) - 1, datosLeidos.Length);
            }
            else
            {
                Thread.Sleep(Convert.ToInt32(this.scan) * 2);
                Conectar();
            }

            Thread.Sleep(Convert.ToInt32(this.scan));
        }
    }
    catch (Exception e)
    {
        Scada.Log.Error("[ID: " + this.descripcion + "] " + e.ToString());
    }
}

```

In the first instance, if it is offline, call Connect() which will set connectionState to true if it is.

This is now infinite until a failure captured by the catch statement occurs, in which if it is due to a communication failure, a reconnection, etc., would be implemented



C# (.NET)



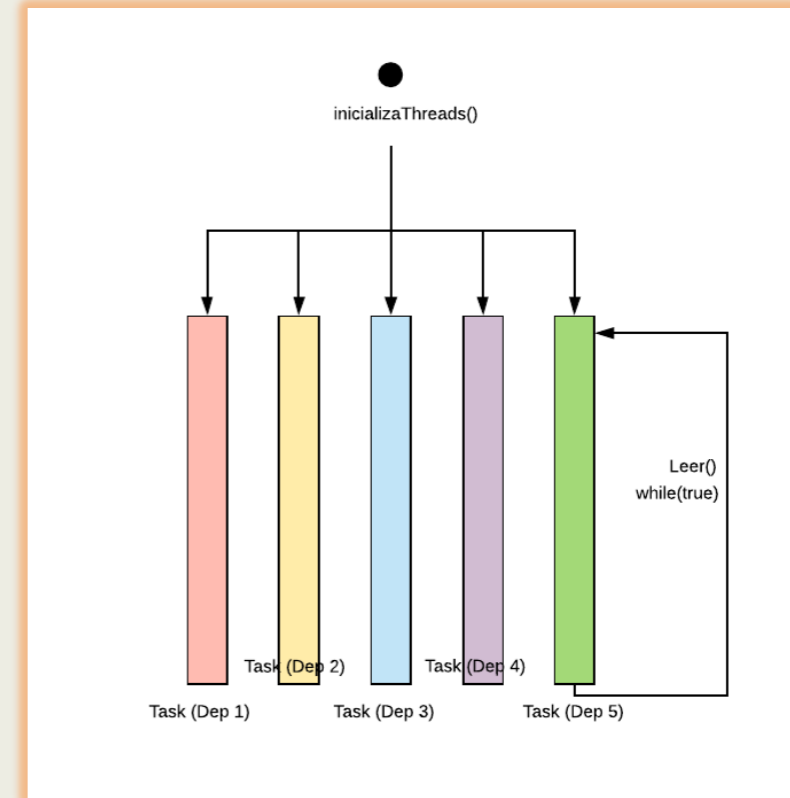
The Conectar() method performs a Connect() of a Socket passing the IP of the modbus server and the port, which is 502.

```

public bool Conectar()
{
    try
    {
        //IP y puerto destino
        IPEndPoint ip = new IPEndPoint(IPAddress.Parse(this.ip), this.puertoTCP);
        this.serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        this.serverSocket.SendTimeout = 5000;
        this.serverSocket.Connect(ip);
        this.estadoConexion = true;
        Scada.Log.Info("[ID: " + this.descripcion + "] Conectado satisfactoriamente a " + this.ip);
    }
    catch (Exception e)
    {
        this.estadoConexion = false;
        Scada.Log.Error("[ID: "+this.descripcion+"] " + e.ToString());
    }
    Scada.Log.Info("[ID: " + this.descripcion + "] Estado de la conexión con el PLC: " + this.estadoConexion);
    return this.estadoConexion;
}

```

In summary, if we had 5 repositories, 5 tasks (threads) would be generated in parallel, each executing its Leer() method infinitely.





C# (.NET)



The Modbus TCP **readMultipleWords()** method is implemented as follows.

```

public int[] readMultipleWords(Socket socket, int unidad, int referencia, int cantidad)
{
    // Construir la trama Modbus/TCP
    buffer = new byte[12];
    byte[] buffer2 = new byte[cantidad * 2];
    int i = 0;

    for (i = 0; i < 5; i++)
    {
        buffer[i] = (byte)0;
    }

    buffer[5] = (byte)6;
    buffer[6] = (byte)unidad;
    buffer[7] = (byte)3;
    buffer[8] = (byte)(referencia >> 8);
    buffer[9] = (byte)(referencia & 0xFF);
    buffer[10] = (byte)0;
    buffer[11] = (byte)cantidad;

    int sendedDataLength = socket.Send(buffer); //cabecera

    buffer = new byte[137];
    int receivedDataLength = socket.Receive(buffer);

    for (i = 0; i < buffer2.Length; i++)
    {
        buffer2[i] = buffer[i + 9];
    }

    return getWords(buffer2);
}

```

Byte 0	Transaction identifier. Copied server, usually 0.
Byte 1	Byte 1 Transaction identifier. Copied by the server, usually 0.
Byte 2	Byte 2 Protocol identifier = 0.
Byte 3	Byte 3 Protocol identifier = 0.
Byte 4	Byte 4 Field length (byte high) = 0. Since the messages are less than 256
Byte 5	Byte 5 Length field (low byte). Number of next bytes.
Byte 6	Byte 6 Unit identifier (previously slave address).
Byte 7	Byte 7 Modbus function code.
Byte 8 & more	Byte 8 & more The necessary data.

According to the protocol, the following request frame is sent and the response is obtained with the 12 requested records from the indicated position.

Capturing (response) network data with Wireshark

The image shows a Wireshark capture of a Modbus TCP request and response. The request is highlighted in red and shows a 'Function Code: Read Holding Registers (3)'. The response is also highlighted in red and shows a list of 12 registers, with the first two (100 and 101) having non-zero values (16772 and 16752 respectively).

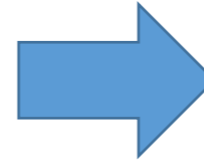
```

> Frame 337: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0
> Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 502, Dst Port: 52178, Seq: 34, Ack: 25, Len: 33
  Modbus/TCP
    Transaction Identifier: 0
    Protocol Identifier: 0
    Length: 27
    Unit Identifier: 1
  Modbus
    .000 0011 = Function Code: Read Holding Registers (3)
    [Request Frame: 335]
    [Time from request: 0.000271000 seconds]
    Byte Count: 24
    > Register 99 (UINT16): 0
    > Register 100 (UINT16): 16772
      Register Number: 100
      Register Value (UINT16): 16772
    > Register 101 (UINT16): 0
    > Register 102 (UINT16): 16752
    > Register 103 (UINT16): 0
    > Register 104 (UINT16): 0
    > Register 105 (UINT16): 0
    > Register 106 (UINT16): 0
    > Register 107 (UINT16): 0
    > Register 108 (UINT16): 0
    > Register 109 (UINT16): 0
    > Register 110 (UINT16): 0

```




Barcode Generation



BAR-CODE



YouTube Based on this link: https://www.youtube.com/watch?v=UJm_1BYn9hs

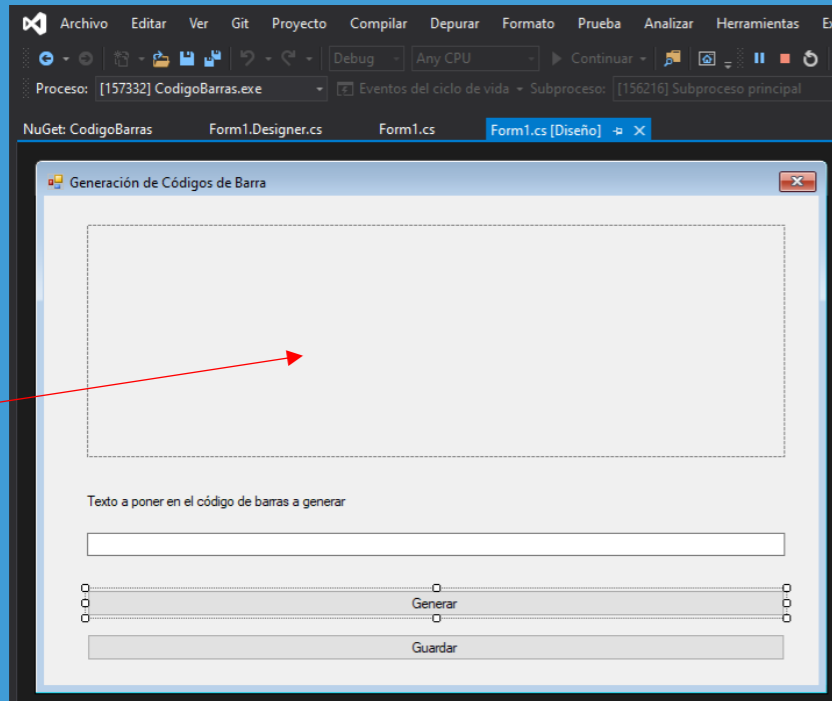
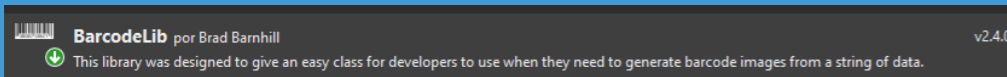


C# (.NET)



Barcode Generation

Brad Barnhill's BarcodeLib library is used. v2.4



Dashboard
600x200

- 1° You must write the text you want.
- 2° Click the Generate button to view the generated barcode.
- 3° Click the Save button if you want to save to disk in PNG format.

```

1 referencia
private void Generar_Click(object sender, EventArgs e)
{
    BarcodeCodigo = new Barcode();
    Codigo.IncludeLabel = true;
    if (!"".Equals(textBox1.Text))
    {
        panel1.BackgroundImage = Codigo.Encode(BarcodeLib.TYPE.CODE128, textBox1.Text, Color.Black, Color.White, panel1.Width, panel1.Height);
        Guardar.Enabled = true;
    }
}

1 referencia
private void Guardar_Click(object sender, EventArgs e)
{
    Image imgFinal = (Image)panel1.BackgroundImage.Clone();
    SaveFileDialog cajaDialogo = new SaveFileDialog();
    cajaDialogo.AddExtension = true;
    cajaDialogo.Filter = "Image PNG (*.png)|*.png";
    cajaDialogo.ShowDialog();
    if (!string.IsNullOrEmpty(cajaDialogo.FileName))
    {
        imgFinal.Save(cajaDialogo.FileName, ImageFormat.Png);
    }
    imgFinal.Dispose();
}

```

A barcode of CODE 128 is generated:

https://es.wikipedia.org/wiki/Code_128



C# (.NET)



Code 128

Code 128 is a high-density barcode, widely used for logistics and parcels. You can encode alphanumeric characters or just numeric characters. With this code it is possible to represent all the characters in the ASCII table, including the control characters.

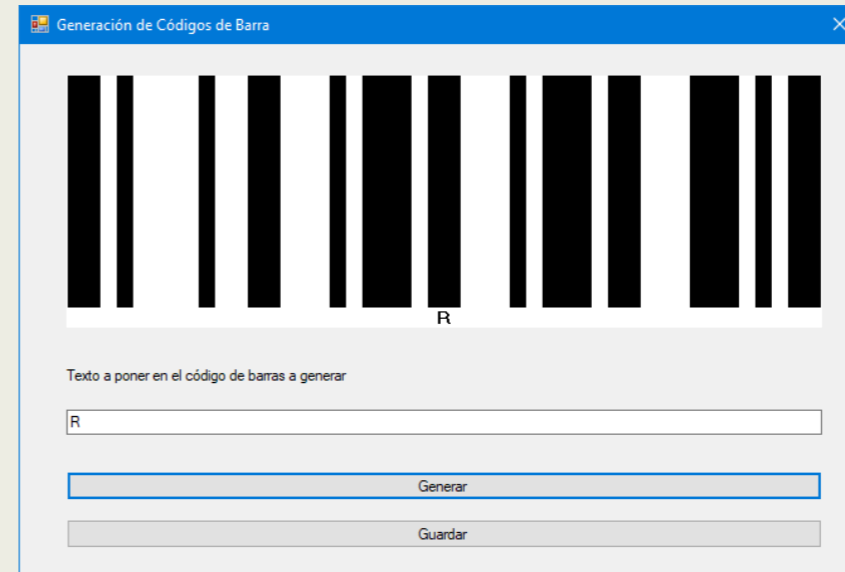
To understand how this code is encoded, we must keep in mind that each ASCII is encoded by 1st.
For example, the ASCII character <space> is made up of

- Two black bars
- A white bar
- Two black bars
- Two white bars
- Two black bars
- Two white bars
- TOTAL= 11 Bars.

The code actually includes six zones.

- On the left, a blank area that should be the length of two characters.
- The starting character.
- A variable number of ASCII characters and is the most useful of this code.
- One digit to check the integrity of the data.
- An end character or "Stop character"
- On the right, a blank area equivalent to two characters.

Example with the character "R" ASCII 82



Try scanning this code with your mobile:





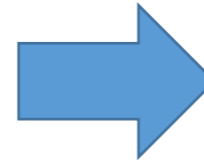
QR Generation



Generación de códigos QR



Texto a poner en el código QR a generar



QR CODE



YouTube Based on this link: https://www.youtube.com/watch?v=R_EAZ19AkHU

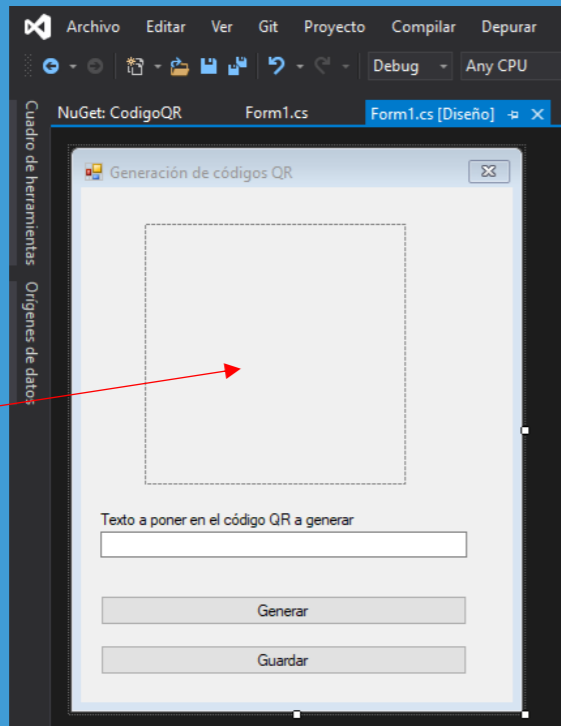
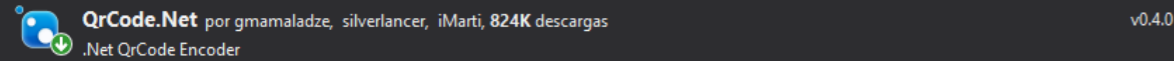


C# (.NET)

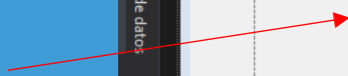


QR code generation

The QrCode.Net v0.4.0 library is used



PictureBox
200x200



- 1° You must write the text you want.
- 2° Click the Generate button to view the generated QR code.
- 3° Click the Save button if you want to save to disk in PNG format.

```

1 referencia
private void Generar_Click(object sender, EventArgs e)
{
    if (!"".Equals(textBox1.Text))
    {
        QrEncoder qrEncoder = new QrEncoder(ErrorCorrectionLevel.H);
        QrCode qrCode = new QrCode();
        qrEncoder.TryEncode(textBox1.Text.Trim(), out qrCode);

        GraphicsRenderer renderer = new GraphicsRenderer(new FixedCodeSize(400, QuietZoneModules.Zero), Brushes.Black, Brushes.White);
        MemoryStream ms = new MemoryStream();
        renderer.WriteToStream(qrCode.Matrix, ImageFormat.Png, ms);
        var imageTemporal = new Bitmap(ms);
        var imagen = new Bitmap(imageTemporal, new Size(new Point(imgQr.Width, imgQr.Height)));
        imgQr.BackgroundImage = imagen;
        Guardar.Enabled = true;
    }
}

1 referencia
private void Guardar_Click(object sender, EventArgs e)
{
    Image imgFinal = (Image)imgQr.BackgroundImage.Clone();
    SaveFileDialog cajaDialogo = new SaveFileDialog();
    cajaDialogo.AddExtension = true;
    cajaDialogo.Filter = "Image PNG (*.png)|*.png";
    cajaDialogo.ShowDialog();
    if (!string.IsNullOrEmpty(cajaDialogo.FileName))
    {
        imgFinal.Save(cajaDialogo.FileName, ImageFormat.Png);
    }
    imgFinal.Dispose();
}

```



C# (.NET)



Code QR (Quick Response)

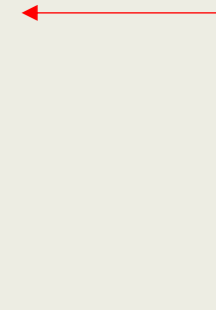
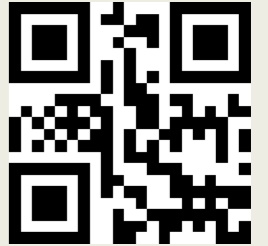
https://es.wikipedia.org/wiki/C%C3%B3digo_QR

QR is the evolution of the barcode. It is a module for storing information in a dot matrix or in a **two-dimensional** barcode. The matrix is read on the mobile device by a specific reader (QR reader) and immediately takes us to an Internet application, a location map, an email, a web page or a profile on a social network.

It has three squares in the corners that allow the reader to detect the position of the code.

The creators' goal was for the code to allow its content to be read at high speed. QR codes are very common in Japan, where they are the most popular two-dimensional code.

Example with the character "R" ASCII82




Try scanning this code with your mobile:





Barcode Driver



 To receive the barcode through the USB port and interpret it. To be used it must be imported into an EXE project.



C# (.NET)



Barcode reader driver

DLL driver importable by any application to perform the function of receiving data via USB (COM).

Calling Conectar() opens the serial port and waits to receive data.

```

public bool Conectar(string portName, int baudRate, Parity parity, int dataBits, StopBits stopBits)
{
    try
    {
        _portName = portName;
        _baudRate = baudRate;
        _parity = parity;
        _dataBits = dataBits;
        _stopBits = stopBits;

        _serialPort = new SerialPort(portName, baudRate, parity, dataBits, stopBits);
        _serialPort.Handshake = Handshake.None;
        _serialPort.DataReceived += new SerialDataReceivedEventHandler(sp_DataReceived);
        _serialPort.ReadTimeout = 500;
        _serialPort.WriteTimeout = 500;

        if (!_serialPort.IsOpen)
        {
            _serialPort.Open();
            return true;
        }

        return false;
    }
    catch (Exception)
    {
        return false;
    }
}

```

The data received event is done through the **sp_DataReceived()** delegate through the SerialDataReceivedEventHandler() event, when the serial port's **ReadExisting()** method is called.

```

void sp_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
    try
    {
        Thread.Sleep(500);
        string data = _serialPort.ReadExisting();
    }
    catch (Exception)
    { }
}

```

```

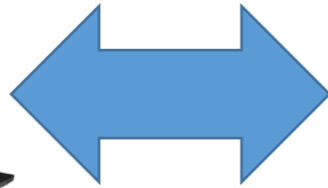
public bool Ack(string comando)
{
    try
    {
        _serialPort.WriteLine(comando);
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

```

After reading, an Ack() programmed with a command that the reader understands can be sent. It can be OK, beep OK, green led or KO, beep KO, red led.



PING



```

C:\Users\r.moraleda>ping www.google.es

Haciendo ping a www.google.es [172.217.17.3] con 32 bytes de datos:
Respuesta desde 172.217.17.3: bytes=32 tiempo=36ms TTL=54
Respuesta desde 172.217.17.3: bytes=32 tiempo=11ms TTL=54
Respuesta desde 172.217.17.3: bytes=32 tiempo=12ms TTL=54
Respuesta desde 172.217.17.3: bytes=32 tiempo=11ms TTL=54

Estadísticas de ping para 172.217.17.3:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 11ms, Máximo = 36ms, Media = 17ms
  
```

*Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
10	2.470686	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=254/65024, ttl=128 (reply in 11)
11	2.482196	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=254/65024, ttl=54 (request in 10)
12	4.485067	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=255/65280, ttl=128 (reply in 13)
13	4.496989	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=255/65280, ttl=54 (request in 12)
16	6.499628	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=256/1, ttl=128 (reply in 17)
17	6.514041	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=256/1, ttl=54 (request in 16)
20	8.516586	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=257/257, ttl=128 (reply in 21)
21	8.529157	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=257/257, ttl=54 (request in 20)
22	10.532151	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=258/513, ttl=128 (reply in 23)
23	10.544698	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=258/513, ttl=54 (request in 22)
33	12.546115	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=259/769, ttl=128 (reply in 34)
34	12.557921	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=259/769, ttl=54 (request in 33)
53	14.560237	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=260/1025, ttl=128 (reply in 54)
54	14.573897	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=260/1025, ttl=54 (request in 53)
63	16.575784	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=261/1281, ttl=128 (reply in 64)
64	16.588277	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=261/1281, ttl=54 (request in 63)
65	18.590663	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=262/1537, ttl=128 (reply in 66)
66	18.603187	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=262/1537, ttl=54 (request in 65)
67	20.605603	192.168.1.151	172.217.17.3	ICMP	74	Echo (ping) request id=0x0001, seq=263/1793, ttl=128 (reply in 68)
68	20.618117	172.217.17.3	192.168.1.151	ICMP	74	Echo (ping) reply id=0x0001, seq=263/1793, ttl=54 (request in 67)

> Frame 67: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0

> Ethernet II, Src: IntelCor_89:cf:79 (c0:b6:f9:89:cf:79), Dst: Sagemcom_23:f7:61 (70:0b:01:23:f7:61)

> Internet Protocol Version 4, Src: 192.168.1.151, Dst: 172.217.17.3

> Internet Control Message Protocol

- Type: 8 (Echo (ping) request)
- Code: 0
- Checksum: 0x4c54 [correct]
- [Checksum Status: Good]
- Identifier (BE): 1 (0x0001)
- Identifier (LE): 256 (0x0100)
- Sequence number (BE): 263 (0x0107)
- Sequence number (LE): 1793 (0x0701)
- [Response frame: 68]

> Data (32 bytes)



Sending PING commands



C# (.NET)



Ping

Executable that sends 10 pings to www.google.es and prints the result.

```

Program.cs
ConsoleApplication1
1 using System;
2 using System.Net.NetworkInformation;
3 using System.Threading;
4
5 namespace ConsoleApplication1
6 {
7     class Program
8     {
9         static void Main(string[] args)
10        {
11            bool resultado;
12
13            for (int i=1; i<=10;i++)
14            {
15                resultado = ping("www.google.es", 2000);
16                Console.WriteLine(i.ToString() + " - " + resultado.ToString());
17                Thread.Sleep(2000);
18            }
19        }
20
21        public static bool ping(string host, int timeout)
22        {
23            Ping pingSender = new Ping();
24
25            PingReply reply = pingSender.Send(host,timeout);
26
27            if (reply.Status == IPStatus.Success)
28            {
29                return true;
30            }
31            return false;
32        }
33    }
34 }
35

```

The **System.Net.NetworkInformation** library is used by creating a **Ping** object and calling **Send()**.


The host is passed, in this case www.google.es and the timeout, in this case 2 seconds.

With **Console.WriteLine()** the result of the **ping()** method executed every 2 seconds is printed to the console.

```

C:\WINDOWS\system32\cmd.exe
1- True
2- True
3- True
4- True
5- True
6- True
7- True
8- True
9- True
10- True
Presione una tecla para continuar . . .

```

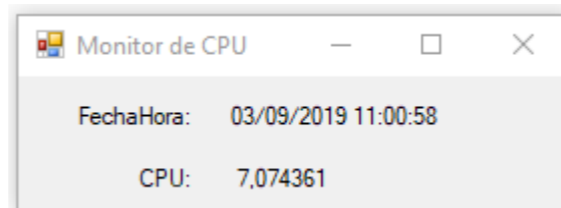
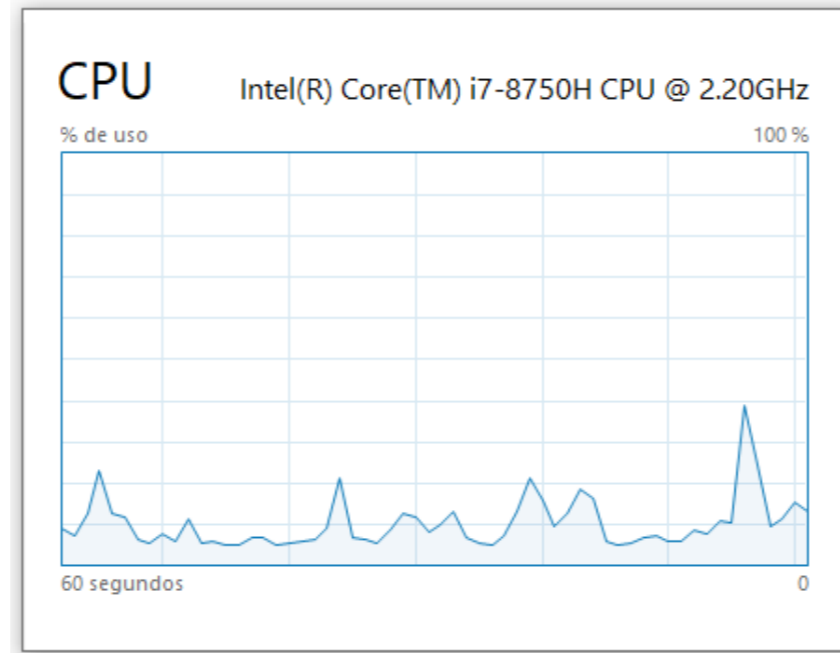
The previous page shows a Wireshark  capture of ICMP packets when running the application. There are 10 requests and 10 replies.



CPU Monitor



PC CPU Monitor



CSV



	A	B
1	03/09/2019 11:00:47	3,734244
2	03/09/2019 11:00:48	3,806889
3	03/09/2019 11:00:49	4,319988
4	03/09/2019 11:00:50	3,662477
5	03/09/2019 11:00:51	6,046159
6	03/09/2019 11:00:52	4,609344
7	03/09/2019 11:00:53	10,97619
8	03/09/2019 11:00:54	20,00967
9	03/09/2019 11:00:55	3,267673
10	03/09/2019 11:00:56	5,195621
11	03/09/2019 11:00:57	8,973086
12	03/09/2019 11:00:58	7,074361
13	03/09/2019 11:00:59	11,46766
14	03/09/2019 11:01:00	7,468812
15	03/09/2019 11:01:01	6,705626
16	03/09/2019 11:01:02	6,129755
17	03/09/2019 11:01:03	4,225739
18	03/09/2019 11:01:04	5,770069



C# (.NET)



Monitor CPU write to csv

Executable that reads the % CPU and logs it into a CSV

```

1  using System;
2  using System.Windows.Forms;
3  using System.Diagnostics;
4  using System.IO;
5
6  namespace MonitorCPU
7  {
8      3 references
9      public partial class Form1 : Form
10     {
11         private PerformanceCounter CPUCounter = new PerformanceCounter("Processor", "% Processor Time", "_Total");
12
13         1 reference
14         public Form1()
15         {
16             InitializeComponent();
17             this.timer1.Enabled = true;
18             this.timer1.Interval = 1000;
19         }
20
21         1 reference
22         private void timer1_Tick(object sender, EventArgs e)
23         {
24             this.label4.Text = DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss");
25             this.label1.Text = this.CPUCounter.NextValue().ToString();
26             this.WriteFile();
27         }
28
29         1 reference
30         private void WriteFile()
31         {
32             using (StreamWriter stream = new FileInfo("cpuLog.csv").AppendText())
33             {
34                 stream.WriteLine(this.label4.Text + ";" + this.label1.Text);
35             }
36         }
37     }
38 }

```

The `System.Diagnostics.PerformanceCounter()` library is used, passing the desired counter in this case % Processor Time – Total.

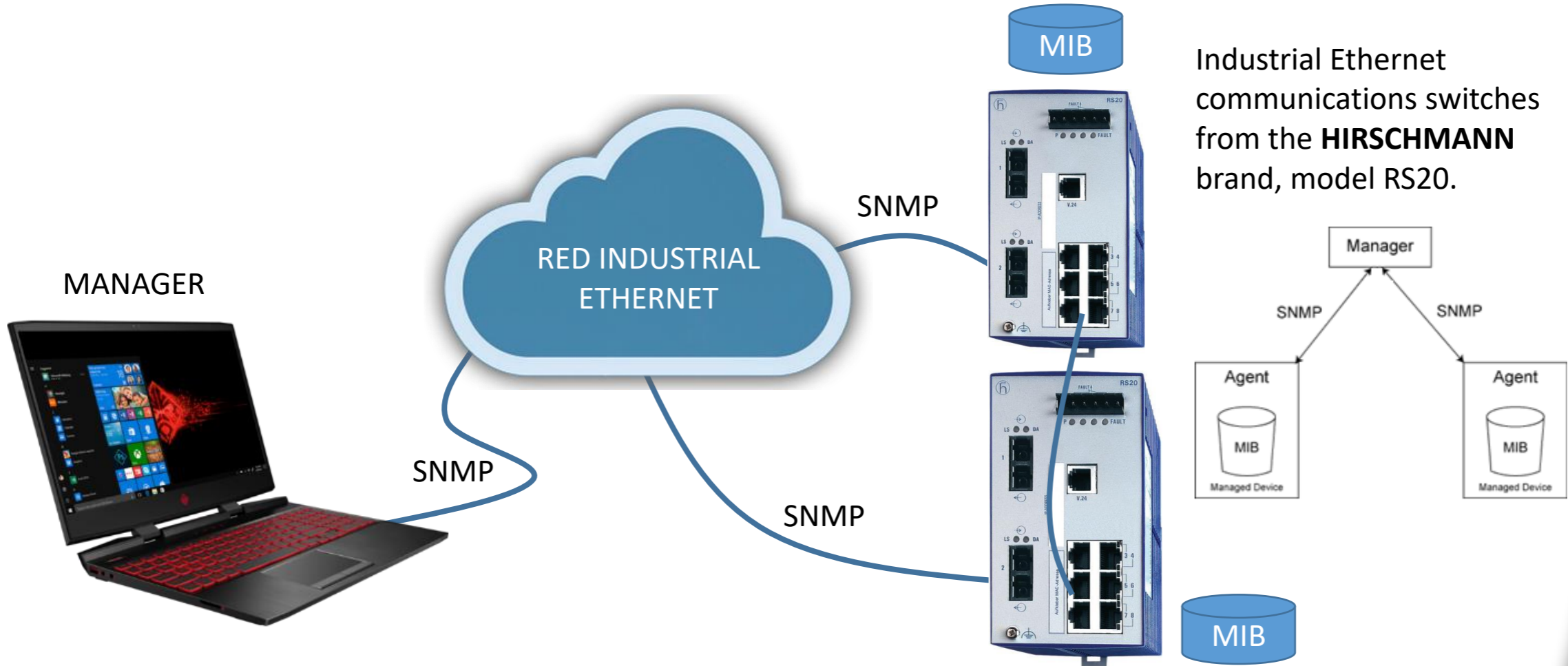
A thread is created with an interval of 1 second where the value is collected with `CPUCounter.NextValue()` and later saved in the csv file through the `writeFile()` method..




Driver SNMP

SNMP

Simple Network Management Protocol



Industrial Ethernet communications switches from the **HIRSCHMANN** brand, model RS20.

 To receive data via SNMP from the MIB of the device to be connected.

MIB: Management Information Base

Reference record for OID 1.3.6.1.4.1.248

1 iso > 3 identified-organization, org, iso-identified-organization > 6 dod > 1 internet > 4 private > 1 enterprise, enterprises > 248 hirschmann



C# (.NET)



Driver SNMP

Driver DLL importable by any application to perform the function of receiving data over Ethernet through the SNMP (UDP) protocol.

```

public SNMPv1CommunicationInterface(int version, string hostAddress, string community, int receiveTimeout, int sendTimeout)
{
    try
    {
        this.version = version;
        this.hostAddress = hostAddress;
        this.community = community;
        this.receiveTimeout = receiveTimeout;
        this.sendTimeout = sendTimeout;
        dSocket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);
        dSocket.SendTimeout = this.sendTimeout;
    }
    catch (SocketException)
    {
        throw;
    }
}

```

It is valid for any Ethernet device that speaks SNMP. Currently the majority; such as printers, switches, routers, servers, etc.

When calling getMIBEntry() passing an OID it connects to the device, returns an object with its numerical or alphanumeric value and disconnects.

```

public MibEntry getMIBEntry (string oid)
{
    SNMPv1CommunicationInterface snmp = null;
    MibEntry entrada = new MibEntry();

    try
    {
        snmp = new SNMPv1CommunicationInterface(version, host, community, receiveTimeout, sendTimeout);
        snmp.setSocketTimeout(timeout);
        SNMPVarBindList newVars = snmp.getMIBEntry(oid);
        SNMPSequence pair = (SNMPSequence)(newVars.getSNMPObjectAt(0));
        SNMPObjectIdentifier snmpOID = (SNMPObjectIdentifier)pair.getSNMPObjectAt(0);
        SNMPObject snmpValue = pair.getSNMPObjectAt(1);
        snmp.closeConnection();

        if (snmpValue is SNMPInteger)
        {
            BigInteger binteger = (BigInteger)snmpValue.getValue();
            entrada.resultado = MibEntry.MIBENTRY_OK;
            entrada.oid = snmpOID.getDigits();
            entrada.valorAlfanumerico = "" + binteger.valor;
            entrada.valorNumerico = binteger.valor;
            return entrada;
        }
        else if (snmpValue is SNMPOctetString)
        {
            entrada.resultado = MibEntry.MIBENTRY_OK;
            entrada.oid = snmpOID.getDigits();
            entrada.valorAlfanumerico = snmpValue.toString();
            entrada.valorNumerico = 0;
            return entrada;
        }
    }
    catch (SNMPException e)
    {
        entrada.resultado = MibEntry.MIBENTRY_ERROR;
        entrada.codigoExcepcion = e.errorStatus;
        entrada.mensajeExcepcion = e.message;
    }

    try
    {
        snmp.closeConnection();
    }
    catch (Exception)
    {
    }
    return entrada;
}

```

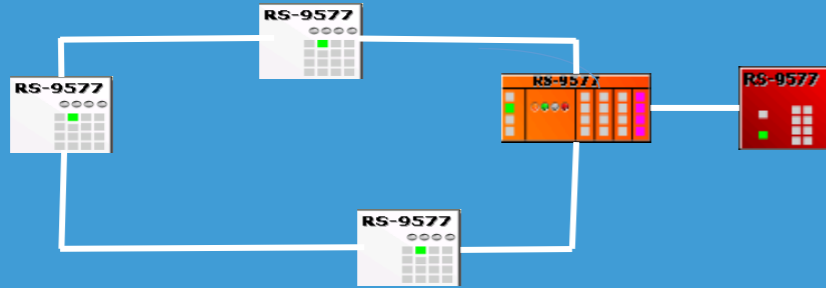


C# (.NET)



SNMP Driver Utilization

I have used this driver to import it into a SCADA (Wonderware System Platform / Intouch) to obtain data from the network of switches to be monitored.



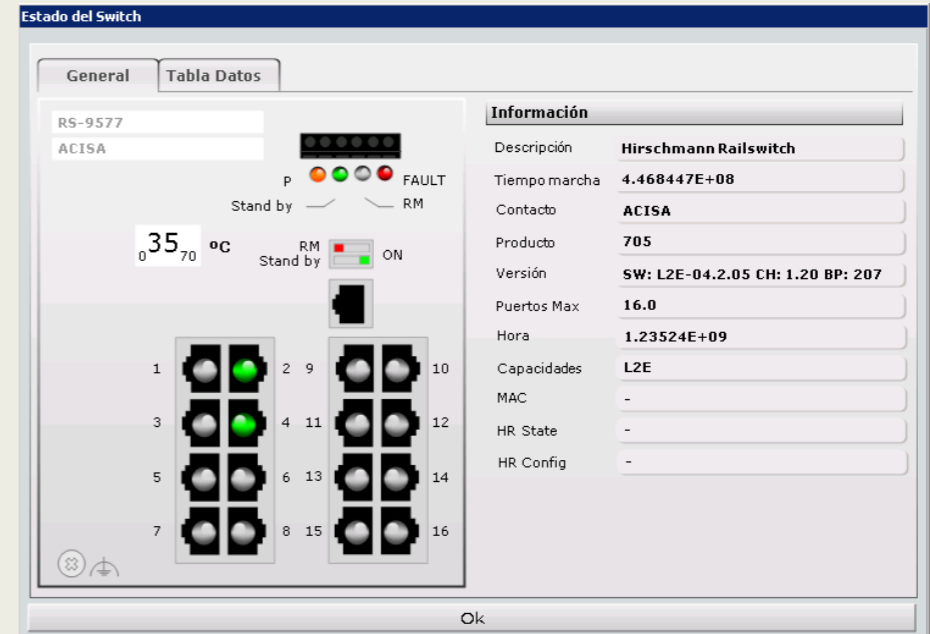
Utilization. Instantiate the DLL and call getMIBEntry() passing the OID.

```
12 Me.ifNumber = libreriaSNMP.getMIBEntry("1.3.6.1.2.1.2.1.0").valorNumerico - 1;
13 Me.sysDescr = libreriaSNMP.getMIBEntry("1.3.6.1.2.1.1.1.0").valorAlfanumerico;
14 Me.sysUpTime = libreriaSNMP.getMIBEntry("1.3.6.1.2.1.1.3.0").valorNumerico;
15 Me.sysContact = libreriaSNMP.getMIBEntry("1.3.6.1.2.1.1.4.0").valorAlfanumerico;
16 Me.sysName = libreriaSNMP.getMIBEntry("1.3.6.1.2.1.1.5.0").valorAlfanumerico;
17 Me.sysLocation = libreriaSNMP.getMIBEntry("1.3.6.1.2.1.1.6.0").valorAlfanumerico;
18 Me.hmSysProduct = libreriaSNMP.getMIBEntry("1.3.6.1.4.1.248.14.1.1.1.0").valorAlfanumerico;
19 Me.hmSysVersion = libreriaSNMP.getMIBEntry("1.3.6.1.4.1.248.14.1.1.2.0").valorAlfanumerico;
20 Me.hmSysModulePortCapacity = libreriaSNMP.getMIBEntry("1.3.6.1.4.1.248.14.1.1.8.0").valorAlfanumerico;
21 Me.hmSystemTime = libreriaSNMP.getMIBEntry("1.3.6.1.4.1.248.14.1.1.30.0").valorNumerico;
22 Me.hmSysSoftwareCapability = libreriaSNMP.getMIBEntry("1.3.6.1.4.1.248.14.1.1.34.0").valorAlfanumerico;

60 mibEntries = libreriaSNMP.getMIBEntriesNumber("1.3.6.1.2.1.2.2.1.8", Me.ifNumber);
61 for i = 1 to Me.ifNumber step 1
62     Me.ifOperStatus[i] = mibEntries.entradas[i].valorNumerico;
63 next;
```

Clicking on the element opens a detail window with the information. Each parameter has an OID associated by the manufacturer.

- Temperature, connected ports, configured switch information, alarms, states, TX, RX bytes, errors, etc.



C# (.NET)

v.1.2 OCTOBER 2024



<https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099>

<https://www.linkedin.com/company/gdo-electric1996/>

<https://www.linkedin.com/in/melvin-francis/>

RICARDO MORALEDA GARETA