

SENSOR DE TEMPERATURA Y HUMEDAD

VISTA POR PC

De igual manera que en el ejemplo del LM35, usamos la comunicación serial del Arduino para ingresar información a la PC.



LISTADO DEL PROGRAMA

```
//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma link "CPort"
#pragma link "CPortCtl"
#pragma link "ILSegmentIndicator"
#pragma link "ILSegmentText"
#pragma link "LPComponent"
#pragma link "LPDrawLayers"
#pragma link "LPTransparentControl"
#pragma link "SLComponentCollection"
#pragma link "ULBasicControl"
#pragma link "ILLed"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{

//ComPort1->Port = "COM1"; //me falla en Builder 6
ComPort1->BaudRate = br9600;
ComPort1->Parity->Bits = prNone;
ComPort1->DataBits = dbEight;
```

```

    ComPort1->StopBits = sbOneStopBit;
//ComPort1->ShowSetupDialog(); // probar con esto
    ComPort1->Open();

    // ComTerminal1->Connected=true;

if (ComPort1->Connected == false) {
ShowMessage("Problemas abriendo el puerto serie");
exit(-1);
}

//ComPort1->Port NO FUNCIONA USAR SHOWSETUPDIALOG Y PIDE EL COM
//ComPort1->Connected=True;
ComPort1->Open();
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{

}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    ComPort1->ShowSetupDialog();
}
//-----

void __fastcall TForm1::Button5Click(TObject *Sender)
{

    ComPort1->BaudRate = br9600;
    ComPort1->Parity->Bits = prNone;
    ComPort1->DataBits = dbEight;
    ComPort1->StopBits = sbOneStopBit;
    ComPort1->Open();

Timer1->Enabled=true;

}
//-----

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    String a,b,C,P;
    float T1,T2,H1,H2;
T1 =StrToFloat(Edit4->Text);
T2 =StrToFloat(Edit5->Text);
H1 =StrToFloat(Edit6->Text);
H2 =StrToFloat(Edit7->Text);

ComPort1->ReadStr(a,13);
Memo1->Lines->Add(a);
Edit3->Text=a; //muestra todo
//-----

b=a;

//*****

```

```

C= b.SubString(2,5);

    Edit1->Text=C;

P= b.SubString(9,13);

    Edit2->Text=P;

ILSegmentText1->Text= C;
ILSegmentText2->Text= P;

//*****

if((C<T1)|| (C>T2))

    {

        //instrucciones se ejecutan si condición se cumple

        ILLed1->Value=true;// ENCIENDE LED

    }

else

    {

        //instrucciones se ejecutan si condición NO se cumple
        ILLed1->Value=false;// APAGA LED
    }
//.....

if((P<H1)|| (P>H2))

    {

        //instrucciones se ejecutan si condición se cumple

        ILLed2->Value=true;// ENCIENDE LED

    }

else

    {

        //instrucciones se ejecutan si condición NO se cumple
        ILLed2->Value=false;// APAGA LED
    }

//-----
ComPort1->Close();

Timer2->Enabled=true;
Timer1->Enabled=false;
}

//-----

void __fastcall TForm1::Timer2Timer(TObject *Sender)
{

```

```

    Button5->Click();

    Timer2->Enabled=false;
}
//-----

void __fastcall TForm1::Label8Click(TObject *Sender)
{
    if(Memo1->Visible==false)
    {
        //instrucciones se ejecutan si condición se cumple

        Memo1->Visible=true;
        Edit3->Visible=true;
        Edit1->Visible=true;
        Edit2->Visible=true;
    }
    else
    {
        //instrucciones se ejecutan si condición NO se cumple

        Memo1->Visible=false;
        Edit3->Visible=false;
        Edit1->Visible=false;
        Edit2->Visible=false;
    }
}
//-----

```

