

TUTORIAL LEITURA DE SENSORES

Autor: Tiago Lone
Nível: Básico
Criação: 07/03/2006
Última versão: 18/12/2006



Maxwell Bohr
Instrumentação Eletrônica

<http://www.maxwellbohr.com.br>
contato@maxwellbohr.com.br

PdP

Pesquisa e Desenvolvimento de Produtos

<http://www.automato.com.br>
atendimento@automato.com.br

1 – Introdução

Nesse tutorial vamos aprender como podemos ler, através de um programa no computador, parâmetros do ambiente que são capturados por sensores conectados ao *Módulo de Sensores* ou ao *Módulo de Sensores Genérico* do KDR5000 ou às entradas de sensores do MEC1000. Criaremos dois projetos, um em que fazemos a leitura de um sensor de cada vez e outro em que fazemos a leitura de vários sensores em uma única operação e apresentamos esses valores em um gráfico.

2 – Material

Nesse tutorial vamos utilizar o *Módulo de Sensores* e o *Módulo de Sensores Genérico* do KDR5000 ou o MEC1000, com alguns sensores conectados a eles. Para a criação do programa será necessário o Borland Delphi 6. A seguir a imagem do *Módulo de Sensores* com vários sensores conectados a ele.

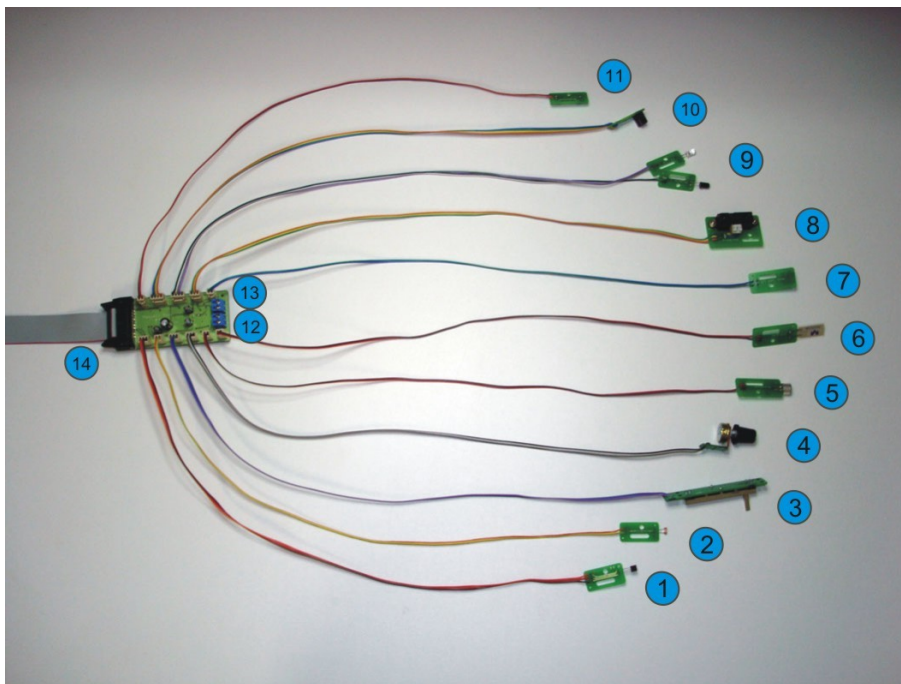


Figura 1: *Módulo de Sensores.* 1-Sensor de Temperatura, 2-Sensor de Luminosidade, 3-Potenciômetro Linear, 4-Potenciômetro Angular, 5-Microfones, 6-Sensor de Vibração, 7-Sensor de Peso, 8-Sensor de Distância, 9-Par Óptico I, 10-Par Óptico II, 11-Chave Magnética, 12-Entrada auxiliar I, 13-Entrada auxiliar II, 14-Conector flat de 20 vias para conexão com o Módulo Principal.

3 – Projeto

Vamos criar dois projetos nesse tutorial. O primeiro será um programa que apenas lê o valor da leitura de um sensor e o apresenta em forma numérica. Será possível ler apenas um sensor de cada vez. A interface desse programa será a seguinte.



Figura 2: Primeiro programa que será criado nesse tutorial.

O segundo programa irá ler todos os sensores simultaneamente e apresentará essas leituras em uma forma de gráfico. Sua interface será a seguinte.

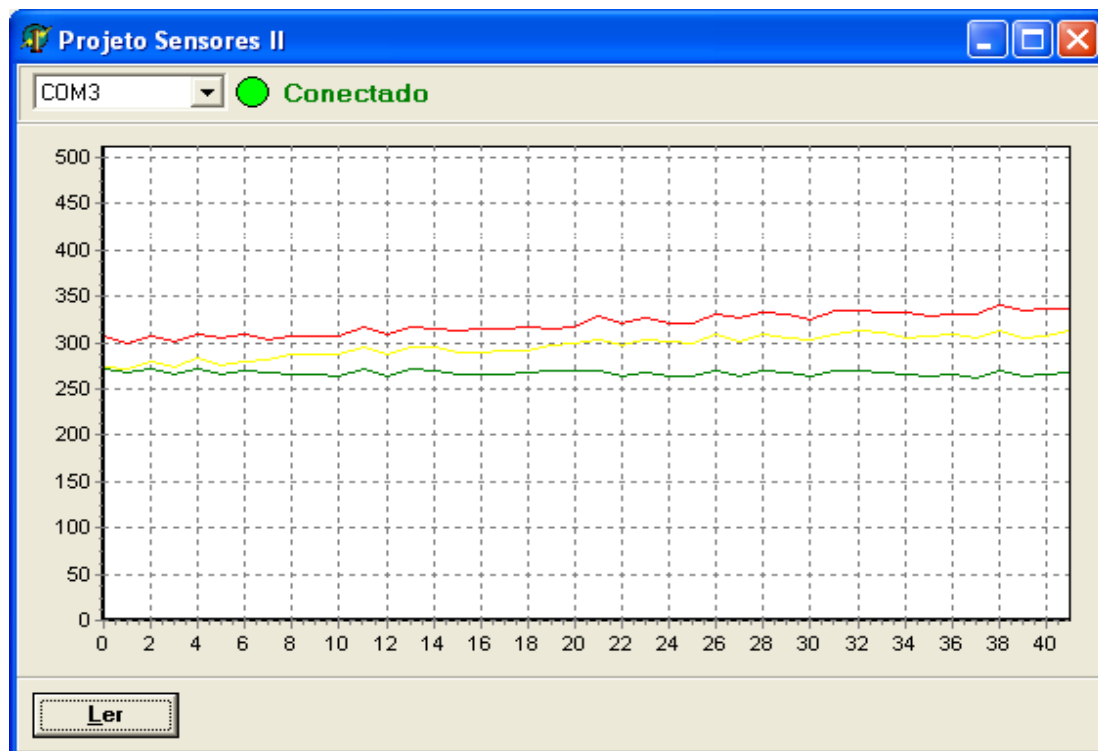


Figura 3: Segundo programa que será criado nesse tutorial.

Vamos iniciar o desenvolvimento criando a interface gráfica do primeiro programa.

ItemIndex = 0

Dessa forma teremos nosso Form principal com a seguinte aparência.

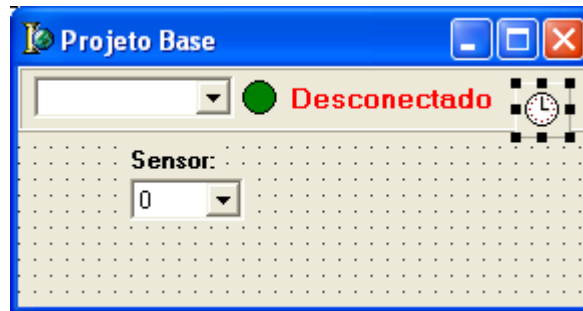


Figura 7: Aparência do Form após adição dos componentes para seleção do sensor que será lido.

Em seguida adicionamos mais dois Labels e modificamos as seguintes propriedades do primeiro.

Name = LabelLeitura
Caption = Leitura:
Font/Style/fsBold = true

E as seguintes propriedades do outro Label, que irá apresentar o valor da leitura do sensor.

Name = LabelValorLeitura
Caption = 0
Font/Style/fsBold = true
Font/Style/Size = 14

Assim teremos a seguinte interface gráfica.

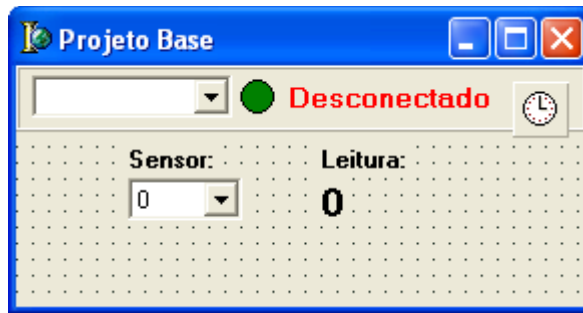


Figura 8: Interface gráfica após a adição dos componentes para apresentação do valor da leitura do sensor.

O último passo para finalizar nossa interface é a adição de um botão para realizar a leitura. Utilizaremos um componente Button que pode ser encontrado na aba “Standard” e possui o seguinte ícone.



Figura 9: Ícone do componente Button.

Modificamos as seguintes propriedades.

Name	=	ButtonLer
Caption	=	Ler
Font/Style/fsBold	=	true

Assim a interface gráfica está completa. Vamos apenas modificar um detalhe, a propriedade Caption do Form principal, que ainda possui a mensagem atribuída a ela no projeto Base. Vamos modificar essa propriedade de “Projeto Base” para “Sensores I”. A aparência final do Form principal será a seguinte.

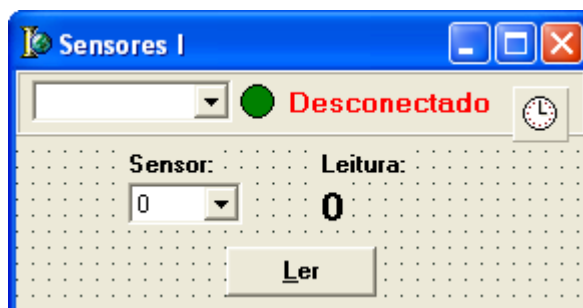


Figura 10: Interface gráfica completa do programa.

Com a interface gráfica completa, vamos para a implementação do código. O primeiro

passo para isso é criar um manipulador para o evento `OnClick` do botão “Ler”. Para fazer isso podemos selecionar o componente `Button` correspondente, ir no **Object Inspector**, selecionar a aba `Events` e dar um duplo clique sobre a linha que está escrito `OnClick`. Uma forma mais fácil de fazer isso é apenas dar um duplo clique sobre o botão no `Form` e com isso o `Delphi` irá criar automaticamente um manipulador para o evento `OnClick`. O seguinte código será criado.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

begin
end;
```

Dentro desse manipulador adicionamos o código que irá fazer a leitura do sensor selecionado. Para isso, utilizaremos o método `SensorReadNow` da biblioteca de controle do `Kit`. Esse método possui a seguinte declaração.

```
Function SensorReadNow(sensor : Integer) :Integer;
```

Como podemos ver, esse método possui um parâmetro do tipo `Integer`. Esse parâmetro indica qual canal de sensor deve ser lido. Tanto o *Módulo de Sensores* quanto o *Módulo de Sensores Genérico* possuem 16 canais de sensores, logo esse parâmetro pode variar de 0 à 15 quando estamos utilizando esses módulos. Esse método retorna um valor do tipo `Integer`. Esse retorno representa a leitura do sensor. Para sensores analógicos esse valor varia de 0 à 1023 e para sensores digitais esse valor é 0 ou 1.

Utilizaremos esse método no manipulador de evento do clique do botão “Ler”. A seguir, o início do código desse manipulador.

```
procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

var
    // Variável para armazenar valor da leitura
    valor : integer;

begin
    // Faz a leitura e armazena na variável "valor"
    valor:=
        kit.SensorReadNow(ComboBoxSensor.ItemIndex);

    ...
```

end;

Nesse início de código declaramos uma variável denominada “valor” que irá conter o valor da leitura do sensor. Para fazer essa leitura, chamamos o método `SensorReadNow` e passamos como parâmetro o valor da posição do item selecionado no `ComboBox` de seleção de sensores. A propriedade `ItemIndex` de um `ComboBox` indica a posição do item selecionado no momento.

Os itens do `ComboBox` de seleção de sensor foram adicionados na ordem correta para que a posição deles dentro do `ComboBox` fosse equivalente ao valor que deve ser passado como parâmetro para representá-los. Seguindo a tabela abaixo, podemos localizar a resposta de qual sensor será lida para determinado canal selecionado através do `ComboBox`.

Canal	Módulo de Sensores	Módulo de Sensores Genérico
00	Potenciômetro II	Conector 1: Primeira entrada analógica
01	Microfone	Conector 1: Segunda entrada analógica
02	Vibração	Conector 1: Terceira entrada analógica
03	Chave Magnética	Conector 1: Entrada digital
04	Temperatura	Conector 2: Primeira entrada analógica
05	Luminosidade	Conector 2: Segunda entrada analógica
06	Potenciômetro I	Conector 2: Terceira entrada analógica
07	Não utilizada	Conector 2: Entrada digital
08	Auxiliar I	Conector 3: Primeira entrada analógica
09	5V	Conector 3: Segunda entrada analógica
10	12V	Conector 3: Terceira entrada analógica
11	Par Óptico I	Conector 3: Entrada digital
12	Peso	Conector 4: Primeira entrada analógica
13	Distância	Conector 4: Segunda entrada analógica
14	Auxiliar II	Conector 4: Terceira entrada analógica
15	Par Óptico II	Conector 4: Entrada digital

O retorno do método `SensorReadNow` é armazenado na variável denominada “valor”.

Com o valor armazenado, falta apenas apresentar esse valor. É o que fazemos em seguida.

```
procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

var
    // Variável para armazenar valor da leitura
    valor : integer;

begin
    // Faz a leitura e armazena na variável "valor"
    valor:=
        kit.SensorReadNow(ComboBoxSensor.ItemIndex);

    // Converte a leitura para String e atribui
    // à propriedade Caption do "LabelValorLeitura"
    LabelValorLeitura.Caption := IntToStr(valor);
end;
```

Essa nova linha de código apenas converte o valor armazenado na variável denominada “valor” para uma String, utilizando o método IntToStr, e atribui essa String à propriedade Caption do Label denominado “LabelValorLeitura”.

Pronto, com isso finalizamos nosso programa e podemos fazer um teste de seu funcionamento. Para isso, vamos no menu **Run – Run** ou pressionamos F9. Se não houver nenhum erro o programa será compilado e executado. Com um Kit conectado em alguma porta serial podemos testar se o programa está funcionando. Selecione a porta serial correta, selecione algum canal de sensor e pressione o botão “Ler”.

O texto do Label denominado “LabelValorLeitura” deverá apresentar o valor da leitura realizada. A aparência do programa deverá ser a seguinte.



Figura 11: Aparência final do programa.

Agora que finalizamos esse primeiro programa vamos iniciar o desenvolvimento do próximo. Como já foi dito, esse segundo programa irá ler todos os sensores simultaneamente, apresentar essas informações de forma gráfica e terá a seguinte aparência.

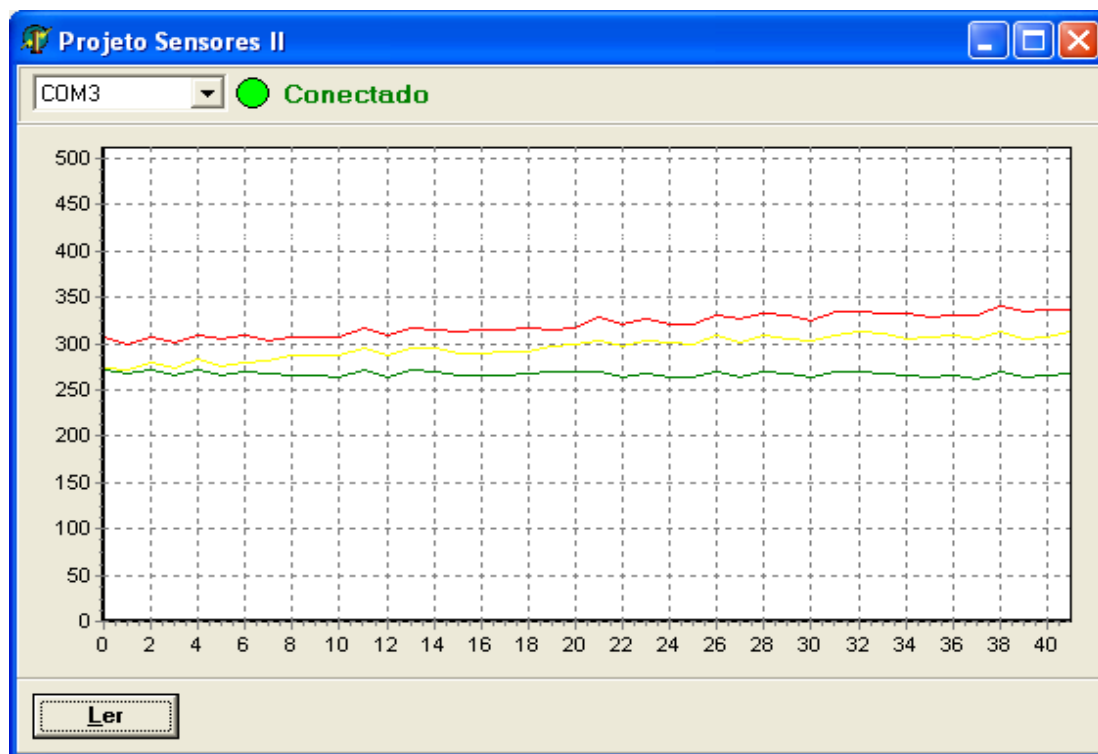


Figura 12: Aparência do segundo programa que será criado nesse tutorial.

Como no projeto anterior, nosso primeiro passo será o desenvolvimento da interface gráfica do programa. Novamente, vamos utilizar o projeto criado no tutorial Base e em cima dele adicionar componentes e código. Para isso copiamos o projeto daquele tutorial e vamos adicionar alguns componentes gráficos extras. Como fizemos no projeto anterior, vamos modificar a propriedade “Caption” do Form principal de “Projeto Base” para “Projeto Sensores II”.

A interface desse programa é composta por um componente Chart, que irá apresentar as informações dos sensores em forma de gráficos, um Button, utilizado para realizar uma leitura e um Panel utilizado apenas para criar uma barra inferior onde será adicionado o botão de leitura.

Os componentes Button e Panel podem ser encontrados na aba “Standard” da barra de componentes. Já o Chart é encontrado na aba “Additional”. Já vimos o ícone do componente Button anteriormente, a seguir o ícone do componente Panel.



Figura 13: Ícone do componente Panel.

Vamos adicionar um componente Panel ao Form principal e modificar as seguintes propriedades.

Name = PanelInferior
Align = alBottom
Caption = (deixar em branco)
Height = 41

Dentro desse Panel adicionamos um componente Button e modificamos as seguintes propriedades.

Name = ButtonLer
Caption = Ler
Font/Style/fsBold = true

Dessa forma teremos a seguinte interface até o momento.

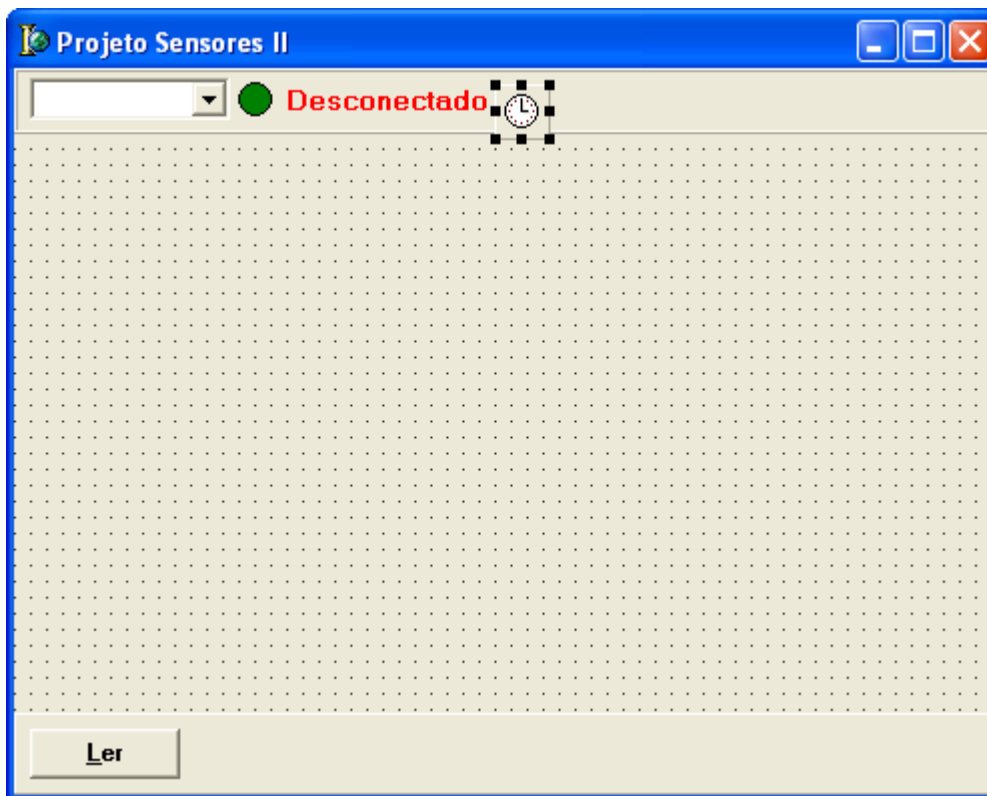


Figura 14: Interface do programa após a adição de um Panel e um Button.

Agora temos que adicionar o componente Chart que pode ser encontrado na aba “Additional” da barra de componentes.



Figura 15: Aba "Additional" da barra de componentes.

Esse componente possui o seguinte ícone:



Figura 16: Ícone do componente Chart.

Adicionamos esse componente e modificamos as seguintes propriedades.

Name	=	ChartSensores
BackColor	=	clWhite
View3D	=	false
Align	=	alClient

Após modificar essas propriedades clicamos com o botão direito do mouse sobre o componente Chart, onde um menu será apresentado e selecionamos a opção “EditChart” nesse menu. Com isso, será apresentada uma janela onde podemos configurar vários parâmetros do componente Chart. Essa janela tem a seguinte aparência.

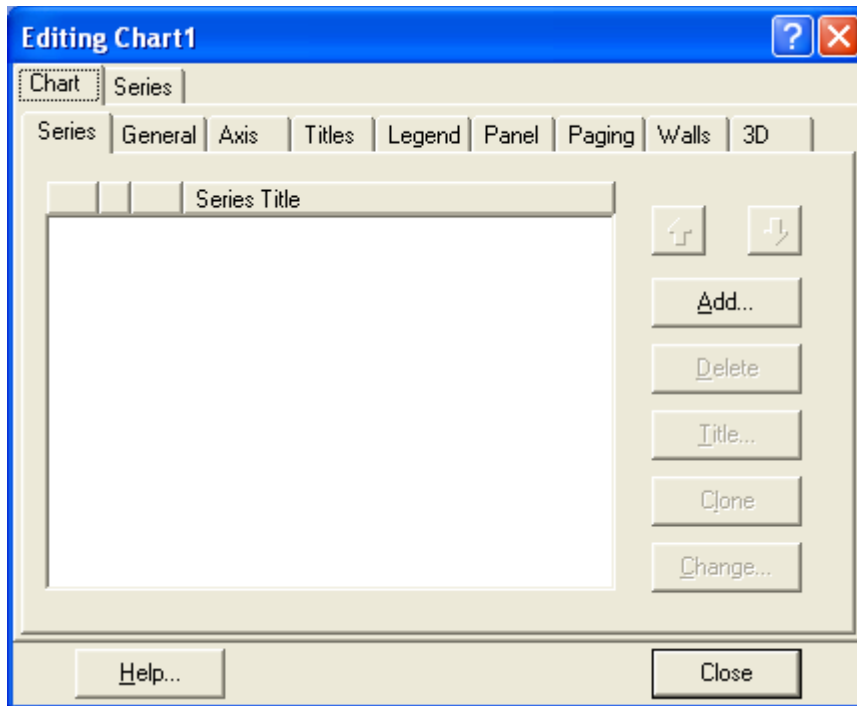


Figura 17: Janela de configuração de um componente Chart.

Temos que adicionar gráficos ao componente Chart. Esses gráficos são denominados “Series” na nomenclatura desse componente. Cada canal de sensor deverá possuir uma “Serie” correspondente a ele. Como os módulos que estamos utilizando suportam até 16 canais de sensores, então vamos adicionar 16 “Series” a esse Chart. Podemos fazer isso indo na aba “Chart / Series”, da janela de configuração que nos foi apresentada, e nessa aba pressionar o botão “Add..”. Será apresentada uma tela onde podemos selecionar o tipo do gráfico que queremos adicionar. Todas as 16 “Series” que iremos adicionar serão do tipo “Fast Line”.

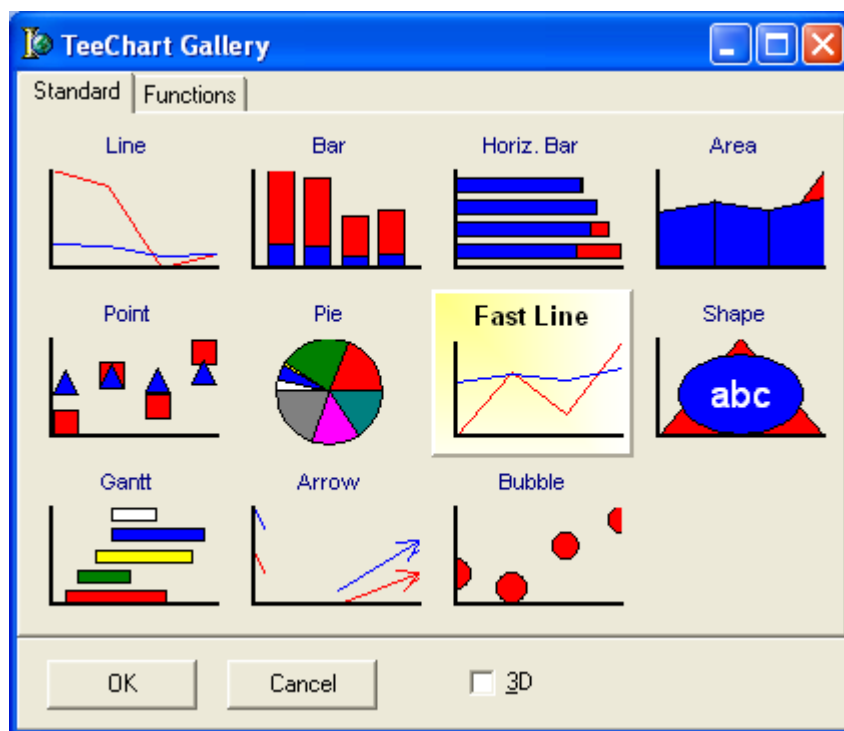


Figura 18: Tela para a seleção do tipo de "Serie".

Após a adição de todos os gráficos, a janela de configuração terá a seguinte aparência.

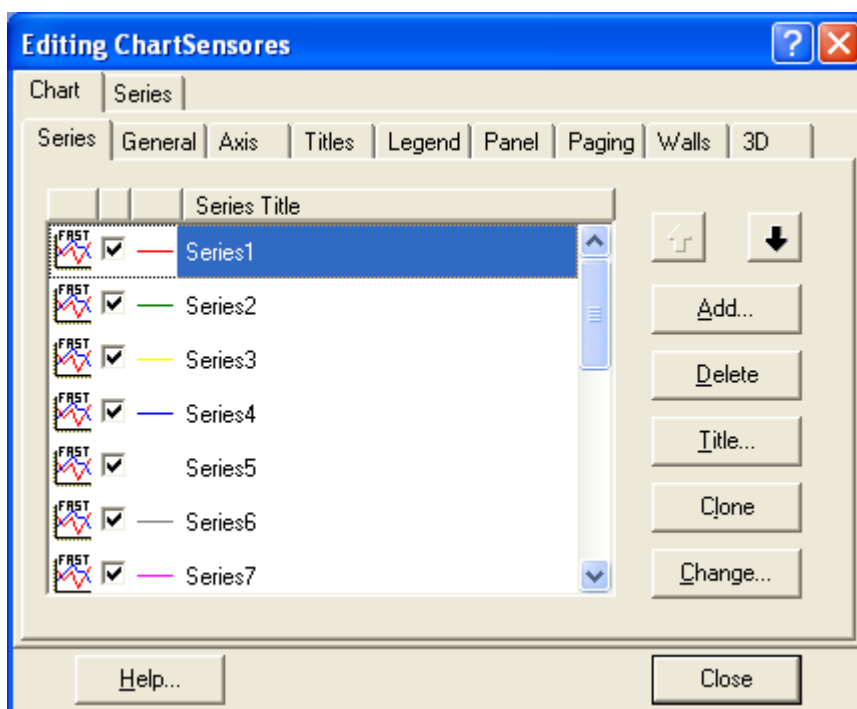


Figura 19: Janela de configuração do Chart após adição das "Series".

Para uma melhor visualização dos gráficos, vamos remover o título e a legenda do Chart. Para isso temos que desmarcar a opção "Visible" na aba "Chart / Titles" e "Chart / Legend".

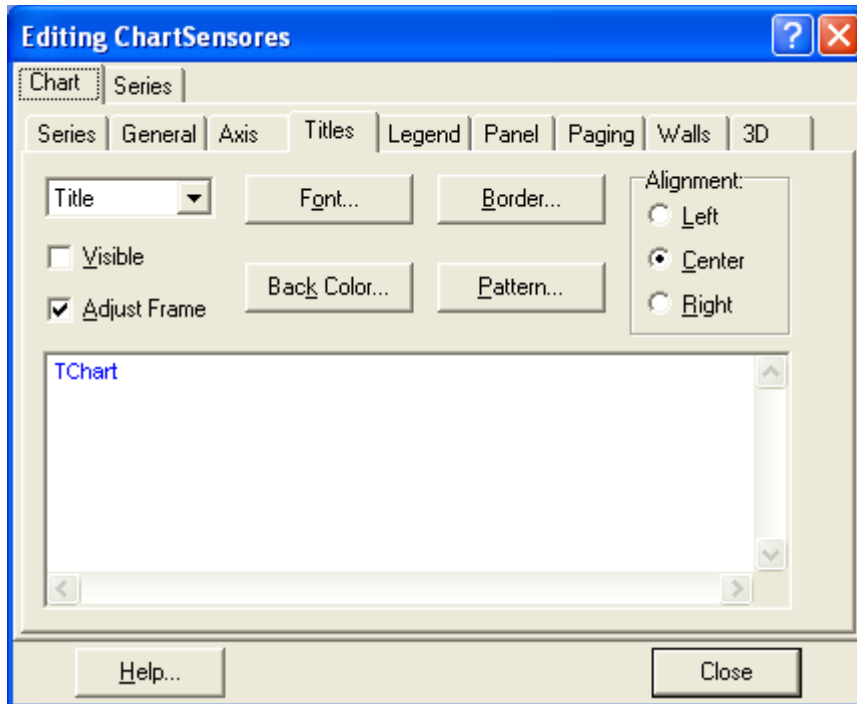


Figura 20: Tela de configuração para remoção do título do gráfico. Observe que a opção "Visible" foi desmarcada.

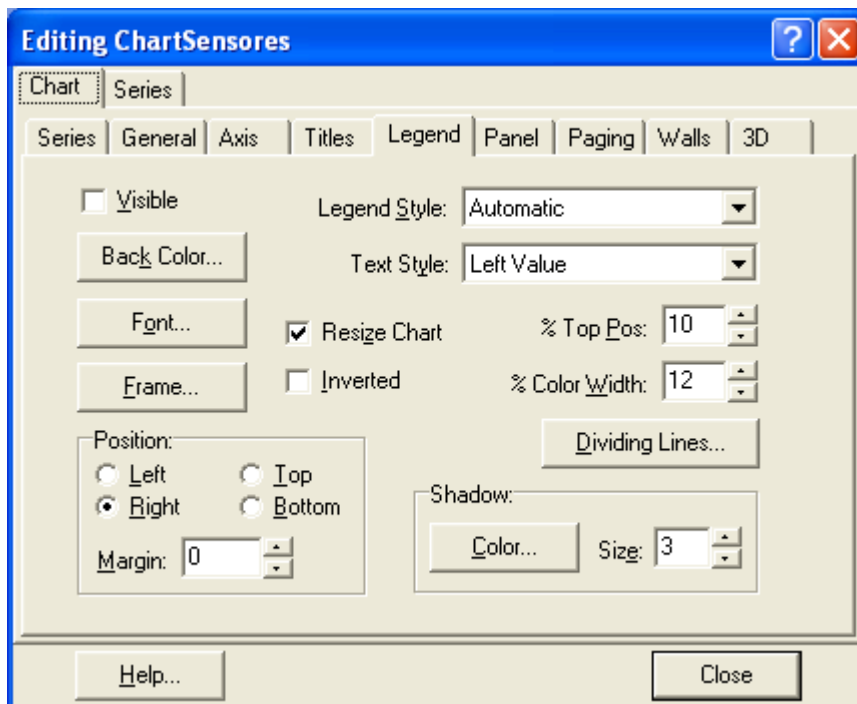


Figura 21: Tela de configuração para remoção da legenda do gráfico. Observe que a opção "Visible" foi desmarcada.

Outra configuração que é interessante que modifiquemos é o valor mínimo e máximo apresentados no eixo Y do gráfico. Pela configuração padrão esses valores são definidos

automaticamente conforme vamos adicionando valores, no entanto, como o valor da leitura dos sensores varia de 0 à 1023, então podemos definir que o eixo Y irá apresentar valores apenas nessa faixa. Isso facilitará a interpretação do gráfico pois o valor mínimo e máximo de apresentação será sempre o mesmo.

Podemos modificar esses valores na aba “Chart/Axis”. Nessa aba selecionamos o eixo “Left”, que é o eixo Y, em seguida desmarcamos a opção “Automatic” desse eixo. As duas opções abaixo dessa opção serão habilitadas permitindo que selecionemos o valor máximo e mínimo do eixo.

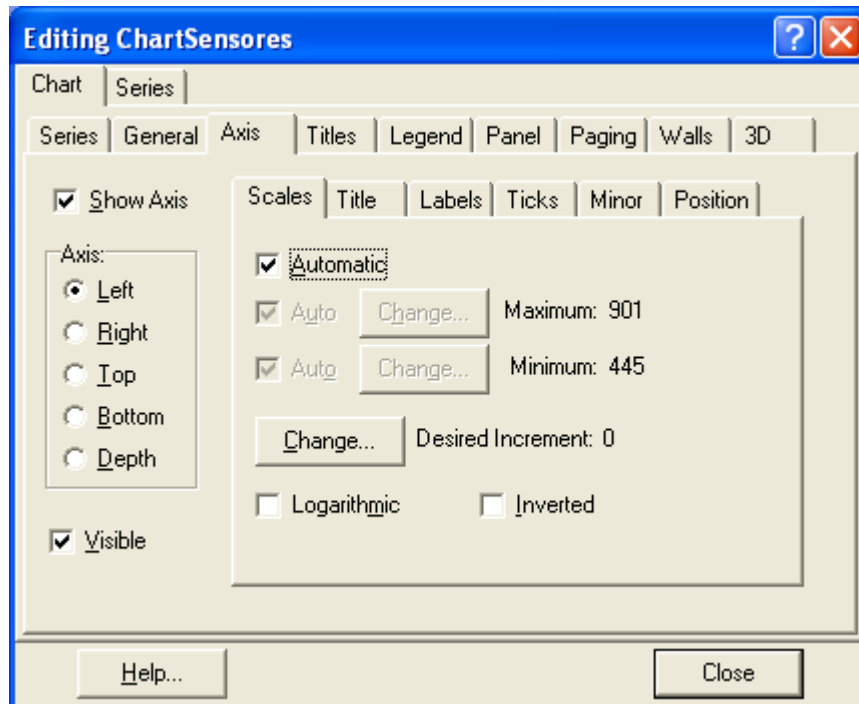


Figura 22: Aba "Chart/Axis" com o eixo "Left" selecionado e a opção "Automatic" ainda marcada.

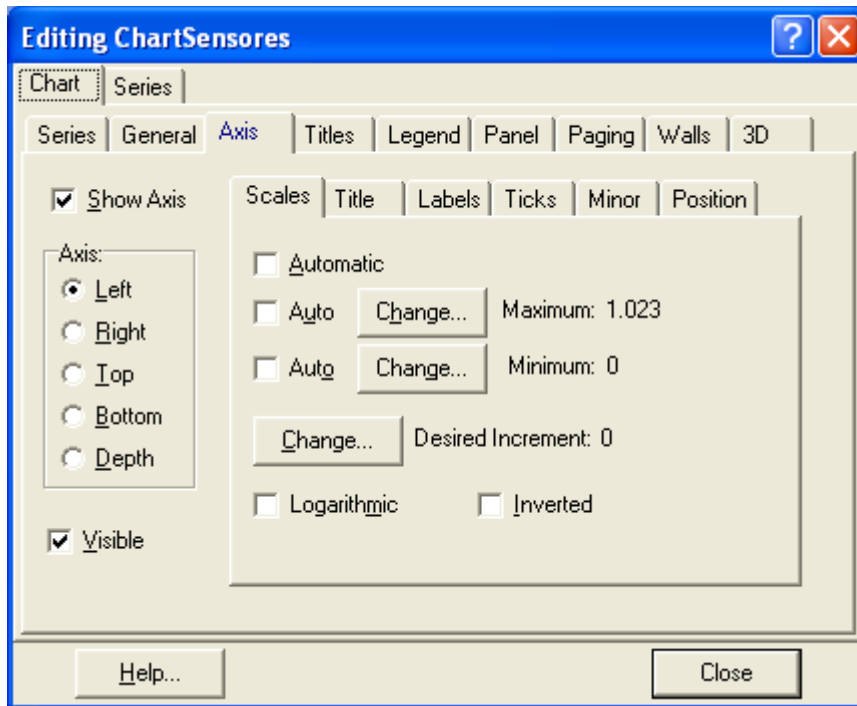


Figura 23: Aba "Chart/Axis" com o eixo "Left" selecionado e a opção "Automatic" já desmarcada. Duas novas opções de seleção de valor máximo e mínimo foram habilitadas.

A aparência do Form principal deve ser a seguinte nesse momento.

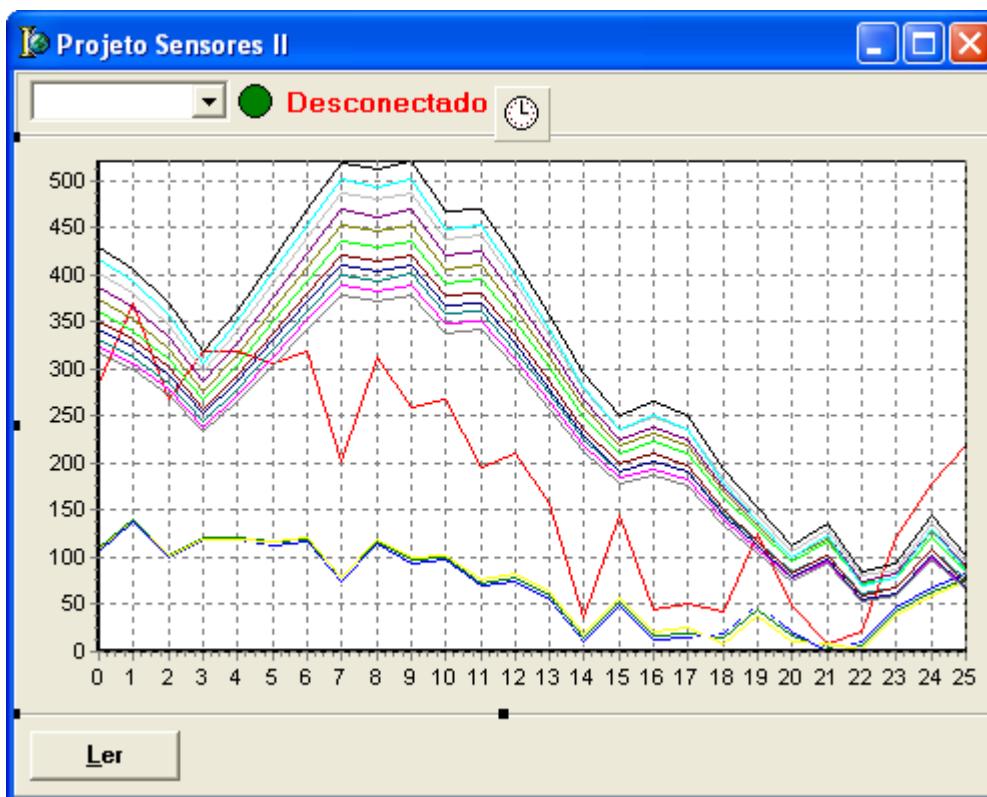


Figura 24: Aparência final do Form principal.

Assim finalizamos a interface do programa. Podemos iniciar o desenvolvimento do código. O primeiro passo para isso é criar um manipulador para o evento `OnClick` do botão “Ler”. Para fazer isso podemos selecionar o componente `Button` correspondente, ir no **Object Inspector**, selecionar a aba `Events` e dar um duplo clique sobre a linha que está escrito `OnClick`. Uma forma mais fácil de fazer isso é apenas dar um duplo clique sobre o botão no `Form` e com isso o Delphi irá criar automaticamente um manipulador para o evento `OnClick`. O seguinte código será criado.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

begin
end;
```

Dentro desse manipulador adicionamos o código que irá fazer a leitura dos sensores e apresentar em forma de gráfico no `Chart` a resposta. Para isso, utilizaremos o método `SensorReadAll` da biblioteca de controle do Kit. Esse método possui a seguinte declaração.

```
Procedure SensorReadAll(var data :
                            DynIntegerArray);
```

Como podemos ver, temos um parâmetro por referência que é um vetor dinâmico de valores do tipo `Integer`. É nesse vetor que será retornado a leitura de todos os canais de sensores. O código que faz essa leitura dentro do manipulador de eventos do botão “Ler” é o seguinte.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

var
    // Vetos de Inteiros
    valores : DynIntegerArray;

    // Variável índice para o "for"
    i : integer;

begin
    // Lê todos os canais de sensores e armazena
    // na variável denominada "valores"
    kit.SensorReadAll(valores);

    ...

end;
```

Nesse trecho de código são declaradas duas variáveis, um vetor de inteiros denominado “valores” que irá armazenar o valor da leitura dos sensores e um inteiro denominado “i” que servirá de índice para uma estrutura “for” que veremos a seguir. Após a declaração das variáveis, é invocado o método SensorReadAll que irá ler todos os canais de sensores e armazenar os valores lidos no vetor de inteiros passado como parâmetro, no caso o vetor denominado “valores”. Com isso temos os valores das leituras. E o código a seguir mostra como fazemos para inserir esses valores no gráfico.

```
Procedure TFormMain.ButtonLerClick(Sender:
                                                    TObject);

var
    // Vetor de Inteiros
    valores : DynIntegerArray;

    // Variável índice para o "for"
    i : integer;
begin
    // Lê todos os canais de sensores e armazena
    // na variável denominada "valores"
    kit.SensorReadAll(valores);

    // Insere os valores lidos no gráfico
    for i:=0 to Length(valores)-1 do
        ChartSensores.Series[i].Add(valores[i]);
end;
```

Com isso finalizamos a implementação do código do nosso programa e já podemos testar todas as suas funcionalidades. Para isso, vamos no menu **Run – Run** ou pressionamos F9. Se não houver nenhum erro o programa será compilado e executado. Com um Kit conectado em alguma porta serial podemos testar se o programa está funcionando. Selecione a porta serial correta e pressione o botão “Ler” varias vezes. A aparência final do programa será a seguinte.

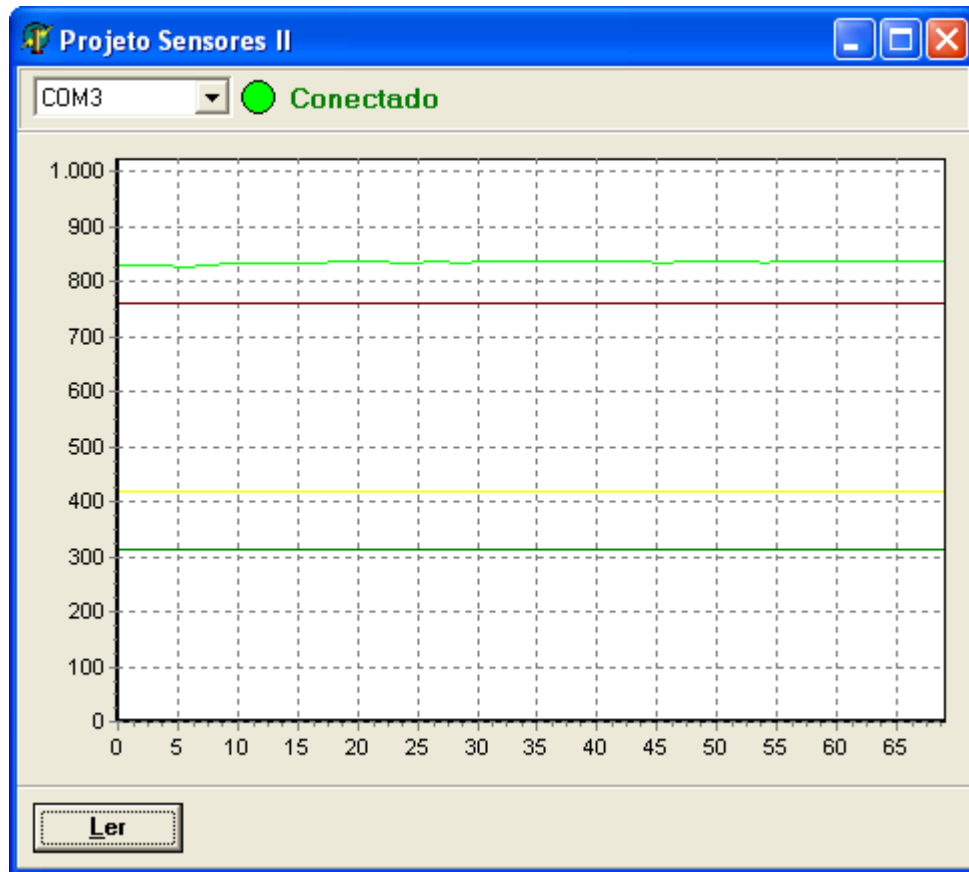


Figura 25: Programa finalizado.

4 – Dicas

Uma funcionalidade interessante que podemos implementar nos programas criados nesse tutorial é uma atualização automática da leitura dos sensores. Para isso temos apenas que adicionar uma linha de código dentro do manipulador do evento OnTimer do Timer que verifica se existe um Kit conectado à porta serial selecionada. O código desse manipulador ficaria assim nos dois programas que foram criados.

```

Procedure TFormMain.TimerCheckTimer(Sender:
                                                    TObject);
begin
    // Verifica se há um Kit respondendo
    if (kit.IsConnected) then
    begin
        // "Acende" LED. Muda cor de fundo do Shape
        ShapeLED.Brush.Color := clLime;
    end
end

```

```

// Muda texto para indicar conexão
LabelConnected.Caption := 'Conectado';

// Muda cor do texto para verde
LabelConnected.Font.Color := clGreen;

// Atualiza leitura de sensores
ButtonLerClick(ButtonLer);
end
else
begin
// “Apaga” LED. Muda cor de fundo do Shape
ShapeLED.Brush.Color := clGreen;

// Muda texto para indicar erro de conexão
LabelConnected.Caption := 'Desconectado';

// Muda cor do texto para vermelho
LabelConnected.Font.Color := clRed;
end;
end;

```

Observe que foi adicionada apenas uma linha a esse manipulador. Essa linha simplesmente chama o manipulador do evento de clique do botão “Ler”. Podemos testar os programas e verificar que agora a leitura dos sensores é feita automaticamente sem a necessidade de pressionar o botão “Ler”.

5 – Conclusão

Nesse tutorial, aprendemos como fazer leitura de sensores a partir de um programa de computador. Criamos dois programas utilizando métodos diferentes para a leitura dos sensores. No primeiro fazemos a leitura de um canal por vez e para isso utilizamos o método da biblioteca de controle SensorReadNow. No segundo a leitura de todos os canais de sensores é feita simultaneamente e para isso utilizamos o método SensorReadAll. Com os conhecimentos adquiridos aqui, podemos utilizar em nossos projetos, de forma muito simples, a leitura de parâmetros do ambiente.