Acessando a Porta Paralela com o Delphi

Vitor Amadeu Souza

vitor@cerne-tec.com.br

Introdução

A Porta Paralela pode ser usada de forma a permitir a interface do computador com o mundo externo para acionamento de lâmpadas, motores assim como a leitura de sensores. Neste artigo, iremos ver uma aplicação prática de acesso a porta paralela pelo Delphi utilizando para isso o componte IO Port.

O Hardware

No exemplo apresentado, poderemos ligar e desligar um led usando para isso a porta paralela do computador utilizando o Delphi. A porta paralela padrão DB25 que é composta de 25 pinos possui os pinos de dados presentes dos pinos 2 ao pino 9 (Acesso aos pinos D0, D1, D2, D3, D4, D5, D6 e D7). Neste caso, o led ficará conectado ao pino D0 da porta de forma a poder ser acionado pela mesma. Vejamos na figura abaixo o esquema que irá permitir o teste do nosso programa:



Observe que o hardware é muito simples, bastando ligar um resistor de 220R em série com o LED. Caso você deseje ligar uma carga de maior potência, como por exemplo uma lâmpada, recomendo o seguinte hardware:



Atenção: Muito cuidado ao realizar estas interfaces pois qualquer engano pode danificar parcialmente ou integralmente o seu computador.

O Software

De posse do hardware pronto e conectado a porta paralela do seu PC, já podemos iniciar a parte referente a programação do Delphi. Porém para máquinas que tenho o Windows superior ao 2000, será necessário instalar um driver que permita que a porta paralela seja acessada pelo nosso programa. No link <u>WWW.cerne-tec.com.br/tutoriais.htm</u> você encontrará um tutorial que irá lhe mostrar como fazer isso.

Para realizar o acesso a porta paralela, iremos utilizar o componente IO PORT que pode ser baixado gratuitamente no seguinte link: <u>http://www.winsoft.sk/IOPort.htm</u> . A versão free deste componente apenas apresentará uma mensagem informando isso toda vez que for executado porém toda a parte funcional do mesmo é mantida. Após realiar o download do mesmo faça a instalação deste componente no seu PC, já que o mesmo já vem com o instalador embutido.

Agora inicialize o Delphi 5, a seguinte tela será apresentada:



O componente IOPORT estará disponível na paleta System, como apresentado abaixo:

🕼 Delphi 5 - Project1 💦 🔲 🗖 🔀									
File Edit Se	arch View Proie	ct Run Component Database Tools Help							
L 🛃 🖬 🖼 🛃 🍯 Standard Additional Wind? System Data Access Data Londrids ADU InterBase Midas LLL									
		r5DdeClientConv T							
Object Inspecto	or 🛛 🗙	💵 Form1 📃 🗖 🔀							
Form1: TForm1	•								
Properties Eve	ents								
Action	·								
ActiveControl									
Align	alNone								
⊞Anchors	[akLeft,akTop]								
AutoScroll	True								
AutoSize	False								
BiDiMode	bdLeftToRight								
	[biSystemMenu =								
BorderStyle	bsSizeable								
BorderWidth	0								
Caption	Form1								
ClientHeight	480								
ClientWidth	614								
Color	CIBtnFace 🗧								
⊞ Constraints	(TSizeConstrain								
Cursor	crDefault								
DefaultMonitor	dmActiveForm								
DockSite	False								
DragKind	dkDrag								
DragMode	dmManual								
Enabled	True								
⊞ Font	(TFont)								
FormStyle	fsNormal								
Height	514								
HelpContext	0								
HelpFile									
2 hidden									

Clique duas vezes neste componente para o mesmo ser adicionado no Formulário.

Agora adicione dois botões de comando ao nosso formulário, o mesmo está disponível na paleta Standard, conforme apresentado abaixo:

🥼 Delphi 5 - Project1	
	t Run Component Database Tools Help
🗋 D 🚅 • 🔛 🕼 🗳 💕	🛃 🛷 🛛 Standard Additional Win32 Sustem Data Access Data Controls ADO InterBase Midas 💶 🕨
Object Inspector 🛛 🗙	🖉 Form1 📃 🗖 🔀
IOPort1: TIOPort	
Properties Events	
About Version 3.3, Copyri	
DirectAccess False	
Name IOPort1	
Tag 0	••••••••••••••••••••••••••••••••••••••
	100 III III III III III III III III III
	₽.₩.
All shown	

O resultado será o apresentado abaixo:

Pelphi 5 - Project 1									
Ele Edit Search View Project Run Component Database Tools Help									
📄 🖻 🗲 🖌 🔲 🧖 😩 😰 🦛 🔰 Standard 🛛 Additional 🛛 Win32 🛛 System 🗋 Data Access 🗋 Data Controls 🖉 ADD 🖉 InterBase 🛛 Midas 🕅 🗠									
		@ # TC □ ▶ • II `b `c`							
Object Inspecto	or 🛛 🛛	🖉 Form1 📃 🗖 🔁							
Form1: TForm1	•								
Properties Eve	ents								
Action	^								
ActiveControl									
Align	alNone	· · · · · · · · · · · · · · · · · · ·							
⊞Anchors	[akLeft,akTop]								
AutoScroll	True								
AutoSize	False	170							
BiDiMode	bdLeftToRight								
⊞BorderIcons	[biSystemMenu								
BorderStyle	bsSizeable								
BorderWidth	0								
Caption	Form1								
ClientHeight	480	· · · · · · · · · · · · · · · · · · ·							
ClientWidth	614	Button1 Button2							
Color	CIBtnFace								
⊞ Constraints	(TSizeConstrain								
Cursor	crDefault								
DefaultMonitor	dmActiveForm								
DockSite	False								
DragKind	dkDrag								
DragMode	dmManual								
Enabled	True								
⊞ Font	(TFont)								
FormStyle	fsNormal								
Height	514								
HelpContext	0								
HelpFile	×								
2 hidden									

Agora clique no Botão Button1. Procure no Object Inspector do Delphi a opção Caption e altere a mesma para Ligar e para o Button2 coloque desligar. A propriedade caption permite alterar o texto apresentado ao usuário que está manipulando o software. A propriedade caption está localizada conforme apresentado abaixo, juntamente com o resultado dos botões depois da alteração informada:



Muito bem, vamos agora a parte referente a programação de forma que possamos ligar e desligar o led. Clique duas vezes sobre o botão Ligar. Neste momento irá abrir o Code Editor do Delphi, janela onde poderemos programar as funcionalidades do mesmo. Neste ponto, digite o seguinte comando:

ioport1.Port[888]:=1;

Vamos entender melhor esta sintaxa. O *ioport1* está se referindo ao componente que faz acesso a porta paralela e está no nosso formulário, que é o ioport. O *port[888]* faz referência ao endereço da porta paralela que no PC é o decimal 888 ou em hexa 0x378. Ao atribuirmos a ioport1.Port[888] o valor 1, na verdade estamos fazendo que a linha D0 dos 8 bits da porta paralela fique em nível alto, deixando desta forma o led que está conectado no PC ligado. Após a digitação deste comando, teremos a seguinte situação:

		\mathbf{X}
Unit	(<u>ب</u>
<pre>procedure TForm1.Button1Click(Sender: TObject);</pre>		~
begin		
 ioport1.Port[888]:=1; 		
• end;		
• end.		~
	3	
	Unit procedure TForm1.Button1Click(Sender: TObject); begin ioport1.Port[888]:=1; end; end.	Unit procedure TForm1.Button1Click(Sender: TObject); begin ioport1.Port[888]:=1; end; end.

Agora clique duas vezes sobre o botão desligar no formulário principal e escreva o seguinte comando:

ioport1.Port[888]:=0;

O que está sendo feito agora é atribuir a porta paralela o valor 0, o que faz que todos os bits vão a nível 0, deixando assim o led desligado. O resultado na tela de programação será o seguinte:

🖹 Unit1.pas		
TForm1	Unit1	
uriables/Constants 	<pre>* procedure TForm1.Button2Click(Sender: TObject); begin</pre>	
	end.	~
36: 29 Modified	Insert	

Agora execute o programa pressionando a tecla F9 e veja que conforme você pressiona o botão Liga o led irá ligar e conforme pressiona o desliga o mesmo se apagará.

Conclusão

De acordo com a aplicação, podemos perfeitamente usar a porta paralela para fazer acesso ao mundo externo e permitir desta forma que o PC possa atuar também neste meio como em indústrias, automação residencial e etc.