

TUTORIAL

BUZZER

Autor: Tiago Lone

Nível: Básico

Criação: 15/12/2005

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 veremos como trabalhar com o Buzzer do *Módulo de Motores e Displays*. Esse dispositivo é utilizado para emitir avisos sonoros. Para auxiliar nas explicações de como controlar esse ítem através de um programa, vamos criar um programa que permite acionar o Buzzer e ajustar os parâmetros da emissão do áudio, mais especificamente o tempo de emissão e a frequência do som.

2 – Material

O programa desenvolvido nesse tutorial vai utilizar o *Módulo Principal* e o *Módulo de Motores e Displays* com apenas o Buzzer conectado à placa principal desse módulo. Para a criação do programa será necessário o Borland Delphi 6. A seguir a imagem da montagem do KDR5000 necessária para esse tutorial.

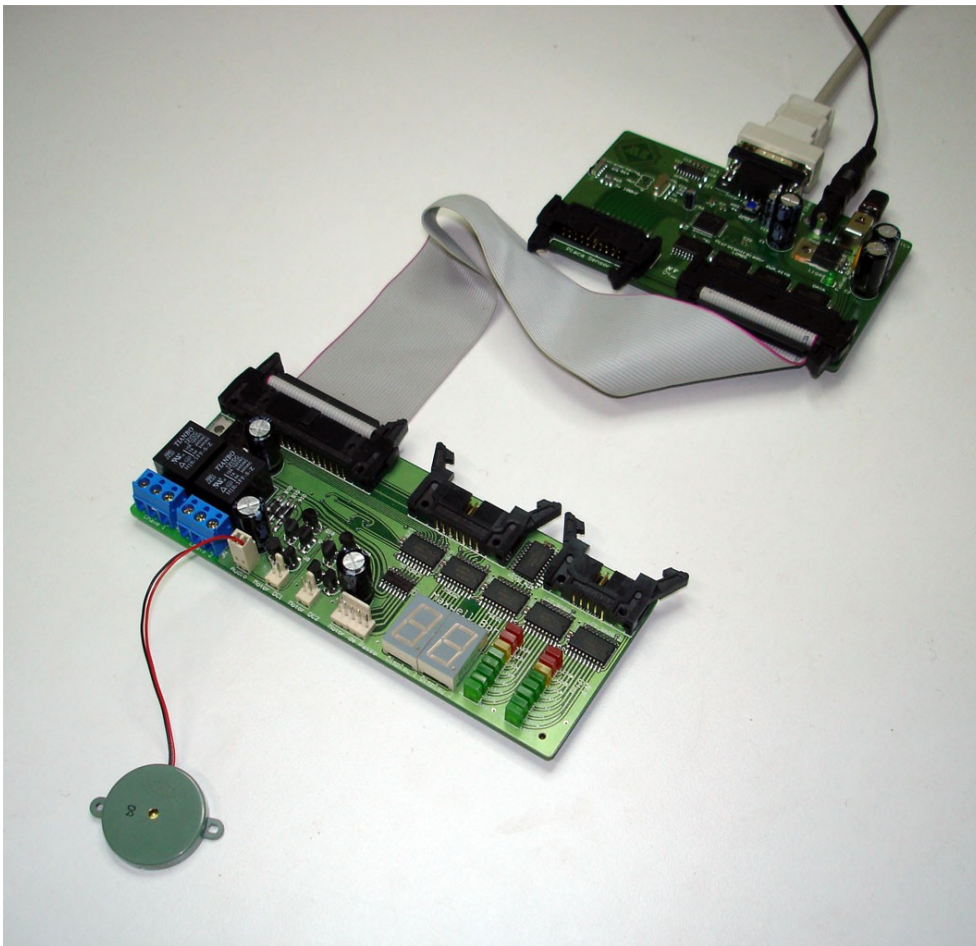


Figura 1: Montagem do KDR5000 utilizada nesse tutorial.

3 – Projeto

Nesse tutorial será desenvolvido um programa que permite o ajuste dos parâmetros da emissão de som pelo Buzzer. Vamos poder ajustar o tempo de emissão e a frequência do som. A interface gráfica do programa será a seguinte.

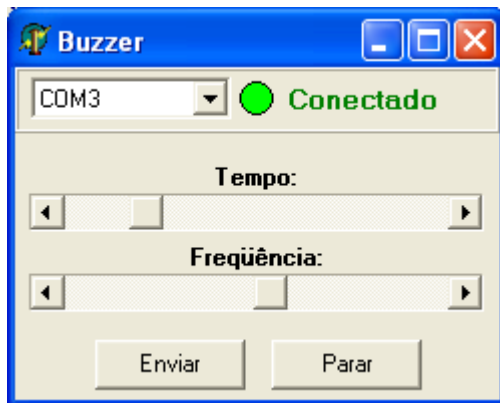


Figura 2: Interface gráfica do programa que será criado nesse tutorial.

Para criar esse programa vamos primeiro desenvolver sua interface gráfica. Vamos utilizar o projeto criado no tutorial Base que já nos fornece algumas funcionalidades interessantes. Para isso copiamos o projeto daquele tutorial e em cima dele vamos adicionar alguns componentes gráficos extras.

A primeira coisa que modificaremos no projeto é a propriedade Caption do Form principal, que possui o texto “Projeto Base”. Vamos modificar para “Buzzer”. Com isso já podemos começar a adicionar os componentes gráficos ao Form.

Será necessário adicionar apenas três tipos de componentes gráficos ao Form. Esses componentes serão Labels, ScrollBars e Buttons. Os Labels apenas contém um texto indicando a propriedade que será modificada pelo ScrollBar logo abaixo dele. Os ScrollBars modificam algum parâmetro da emissão do áudio, mais especificamente o tempo de emissão e a frequência do som. Por fim, os botões são utilizados para enviar o comando e assim iniciar a emissão do áudio ou para parar essa emissão. Todos esses componentes podem ser encontrados na aba “Standard” da barra de componentes.



Figura 3: Aba "Standard" da Barra de componente.

Vamos adicionar um Label e um ScrollBar para o ajuste do tempo de emissão do áudio. O componente Label possui o seguinte ícone.



Figura 4: Ícone do componente Label.

Em seguida adicionamos um componente ScrollBar que encontra-se na aba “Standard” da barra de componentes e possui o seguinte ícone.



Figura 5: Ícone do componente ScrollBar.

Adicionamos os dois componentes e modificamos as seguintes propriedades do Label.

Name = LabelTempo
Caption = Tempo:
Font/Style/fsBold = true

E as seguintes propriedades do ScrollBar.

Name = ScrollBarTempo
Max = 65535
Position = 1000

Com isso o Form terá a seguinte aparência.



Figura 6: Form após a adição de componentes para o ajuste do tempo de emissão do áudio.

Agora vamos inserir os componentes para ajuste da frequência do som emitido. Assim como para o ajuste do tempo de emissão, vamos adicionar aqui um Label e um ScrollBar. No Label modificaremos as seguintes propriedades.

Name = LabelFrequencia
Caption = Frequência:
Font/Style/fsBold = true

E no ScrollBar as seguintes.

Name = ScrollBarFrequencia
Max = 65535
Position = 5000

Com isso teremos o Form da seguinte maneira.



Figura 7: Form com componentes para ajuste de tempo e frequência do som.

Para finalizar a interface desse programa vamos adicionar dois botões, um para enviar o comando de início da emissão do áudio e outro para parar a emissão. Dessa forma temos que adicionar dois componentes Button, que podem ser encontrados na aba “Standard” da barra de componentes. Esse componente possui o seguinte ícone.



Figura 8: Ícone do componente Button.

Temos que modificar as seguintes propriedades dos botões. Vamos modificar primeiro as propriedades do botão para início da emissão do áudio.

Name = ButtonIniciar
Caption = Iniciar
Font/Style/fsBold = true

Em seguida as propriedades do botão para parar a emissão.

Name	=	ButtonParar
Caption	=	Parar
Font/Style/fsBold	=	true

Com isso finalizamos a construção de nossa interface gráfica. A seguir a imagem dessa interface finalizada.



Figura 9: Interface final.

Agora vamos implementar o código para controle do Buzzer. Para isso precisamos criar um manipulador para o evento OnClick do botão para início da emissão e um para o botão de parada. Vamos criar primeiro o manipulador do botão para início. Para fazer isso podemos selecionar o componente Button, que possui o texto “Iniciar”, 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.ButtonEnviarClick(Sender:
                                                    TObject);
begin
end;
```

Dentro desse manipulador vamos implementar o código para comandar o início da emissão do som com os parâmetros que foram ajustados através da interface gráfica. O método que vamos utilizar para acionar o Buzzer é o AudioOn. Esse método possui a seguinte declaração.

```
Procedure AudioOn(time, freq : Integer);
```

Esse método possui dois parâmetros, um que define o tempo que o som permanecerá ligado em milisegundos e outro que define a frequência dele. Esses dois parâmetros podem variar em uma faixa de 0 à 65535 pois são valores de 16 bits. Se qualquer um desses parâmetros for igual a “0” então o áudio será desligado imediatamente.

Já tendo definido, através da propriedade Max e Min, que os ScrollBars retornam um valor entre 0 e 65535 em sua propriedade Position, então o código dentro do manipulador do botão “Iniciar” será muito simples. Temos apenas que chamar o método AudioOn e passar como parâmetros o valor das propriedades Position dos ScrollBar respectivos a cada propriedade. O código será o seguinte.

```
Procedure TFormMain.ButtonEnviarClick(Sender:
                                                    TObject);
begin
    // Envia o comando para acionar o audio
    Kit.AudioOn(ScrollBarTempo.Position,
                ScrollBarFrequencia.Position );
end;
```

Pronto, com isso já podemos testar nosso programa. 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, modifique os parâmetros e pressione o botão “Enviar”. O buzzer deverá emitir um “Beep”. Teste algumas vezes com parâmetros diferentes.

Vamos implementar agora o código do manipulador de eventos do botão “Parar”. Temos que criar o manipulador do evento OnClick do botão “Parar” dando um duplo clique sobre ele e depois inserir o seguinte código.

```
Procedure TFormMain.ButtonPararClick(Sender:
                                                    TObject);
begin
    // Desliga o audio
    Kit.AudioOn(0, 0);
end;
```

Pronto, temos um programa que controla o buzzer e ajusta todos os parâmetros da emissão do áudio. A aparência final do programa ficou assim.

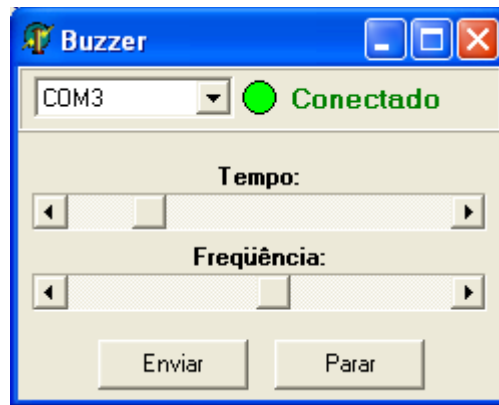


Figura 10: Aparência final do programa.

Podemos ajustar nessa interface o tempo e a frequência do som. Em seguida é preciso apenas pressionar o botão enviar e o som será emitido. Para interromper o som podemos pressionar o botão “Parar”.

4 – Dicas

Uma funcionalidade interessante que poderíamos implementar é que ao modificar a frequência do som, através do ScrollBar referente a esse parâmetro, essa fosse refletida automaticamente no áudio sem a necessidade de pressionar o botão “Iniciar”.

Para isso vamos utilizar o evento do ScrollBar denominado OnChange, que é executado toda vez que há uma mudança na posição do ScrollBar. Capturando esse evento podemos tomar alguma atitude perante qualquer mudança na no ajuste de frequência. Para criar o manipulador para esse evento dê um duplo clique sobre o componente ScrollBar responsável pelo ajuste da frequência. O seguinte código será criado.

```
Procedure TFormMain.ScrollBarFrequenciaChange
                                (Sender: TObject);
begin
end;
```

Para atualizar a frequência do som temos que utilizar o método AudioOn passando como parâmetro a nova frequência juntamente com o tempo antigo. Precisamos de apenas uma linha de código para fazer isso e é o que mostramos a seguir.

```
Procedure TFormMain.ScrollBarFrequenciaChange
                                (Sender: TObject);
begin
    // Envia o comando para acionar o audio
```

```
Kit.AudioOn(ScrollBarTempo.Position,  
            ScrollBarFrequencia.Position );  
end;
```

Faça um teste, modifique o ajuste de frequência e perceba que agora ao modificar a frequência o som responde a mudança imediatamente sem a necessidade de pressionar o botão “Enviar”. Para parar o som pressione o botão “Parar”.

Uma outra dica interessante é desligar o som toda vez que o programa for fechado. Para fazer isso podemos implementar um código para desligar o som no evento OnDestroy do Form principal, que é executado quando o programa fecha. Já criamos um manipulador para esse evento no tutorial Base. Como copiamos aquele projeto, o manipulador para esse evento já existe nesse código fonte e possui o seguinte código.

```
Procedure TFormMain.FormDestroy(Sender: TObject);  
begin  
    kit.CloseCommunication;  
end;
```

Vamos adicionar o código para desligar o som. Isso é feito com o seguinte código.

```
Procedure TFormMain.FormDestroy(Sender: TObject);  
begin  
    // Desliga o som  
    kit.AudioOn(0, 0);  
  
    // Fecha a comunicação  
    kit.CloseCommunication;  
end;
```

Agora, quando fechamos nosso programa, o som será interrompido imediatamente.

5 – Conclusão

Nesse tutorial vimos como controlar o buzzer do *Módulo de Motores e Displays*. Com o projeto que criamos foi possível entender o funcionamento do método AudioOn e o significado de todos os seus parâmetros. Assim já podemos utilizar avisos sonoros em nossos projetos.