

TUTORIAL TECLADO

Autor: Tiago Lone
Nível: Básico
Criação: 20/12/2005
Última versão: 18/12/2006



<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 utilizar o teclado do *Módulo de Motores e Displays* do KDR5000. Criaremos um projeto que permita ler e apagar teclas do buffer de teclado que o Kit possui. Além disso, esse programa será capaz de identificar se existe alguma tecla pressionada no teclado.

2 – Material

Para esse tutorial é necessário o *Módulo Principal* e o *Módulo de Motores e Displays* que é onde o teclado é conectado. Para a criação do programa será necessário o Borland Delphi 6. A seguir a imagem do *Módulo de Motores e Displays* com o teclado conectado.

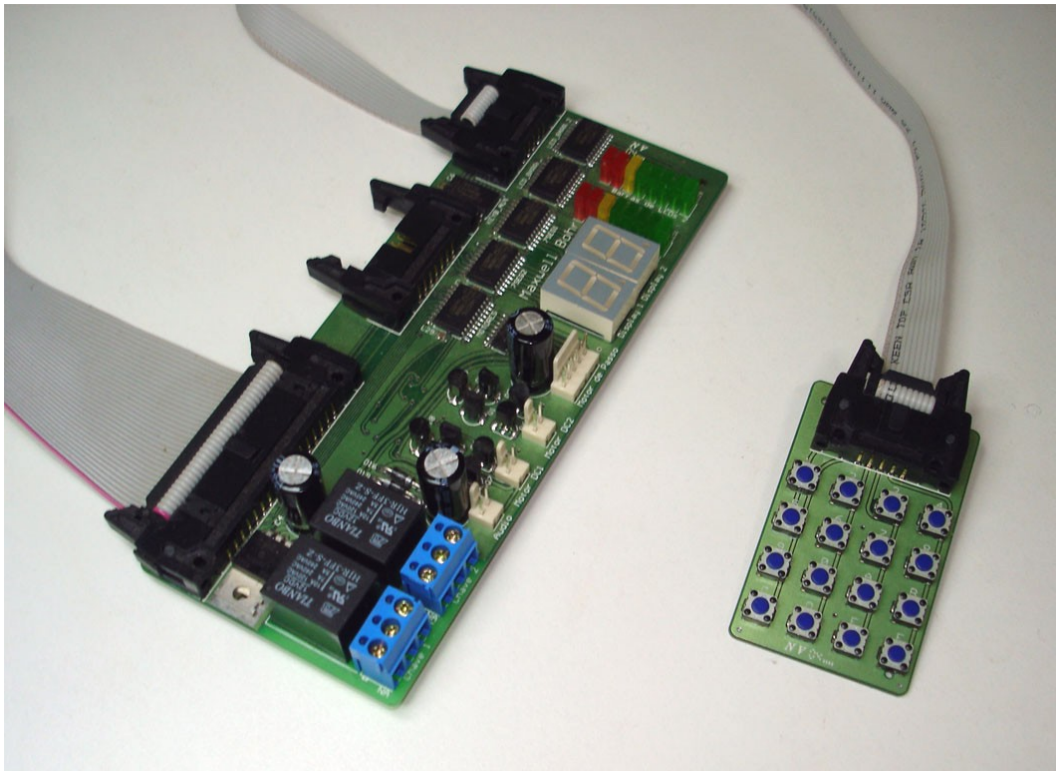


Figura 1: Módulo de Motores e Displays com o teclado conectado.

3 – Projeto

O projeto que iremos criar nesse tutorial permitirá que verifiquemos se há alguma tecla do teclado do Kit pressionada no momento. Ele também irá permitir que sejam lidas as teclas armazenadas no buffer de teclado do Kit. Esse programa será bem simples e irá se parecer com o

seguinte.



Figura 2: Programa que será criado nesse tutorial.

O primeiro passo para desenvolver esse programa é criar a interface gráfica. 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.

A interface do programa é composta por um componente Label e três Buttons. O Label irá apresentar a tecla lida do buffer ou a tecla que estiver pressionada no momento, dependendo do botão pressionado. Os botões oferecem acesso a três funcionalidades diferentes. O primeiro lê uma tecla instantaneamente. Ela serve para verificar se existe alguma tecla pressionada. Existe também um botão para ler uma tecla do buffer do teclado. Por fim temos uma tecla para limpar o buffer do teclado.

Tanto o componente Label quanto o Button podem ser encontrados na aba de componentes “Standard”. A seguir uma imagem dessa aba.

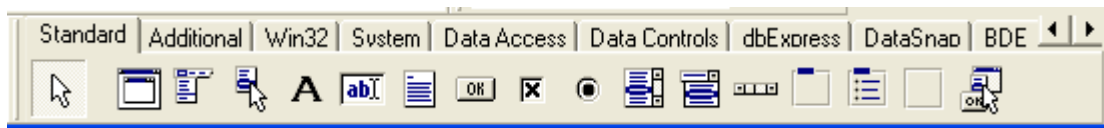


Figura 3: Aba "Standard" da Barra de componente.

O componente Label possui o seguinte ícone:



Figura 4: Ícone do componente Label.

O componente Button possui o seguinte ícone:



Figura 5: Ícone do componente Button.

Primeiro adicionamos um componente Label. Temos que modificar as seguintes propriedades desse componente.

Name	=	LabelTecla
Caption	=	-
Font/Style/fsBold	=	true
Font/Color	=	clNavy
Align	=	alTop
Alignment	=	taCenter
AutoSize	=	False

Nosso Form terá a seguinte aparência.

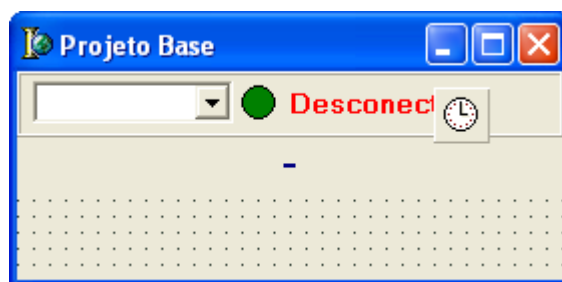


Figura 6: Form após a adição do Label.

Vamos adicionar agora os três botões. Eles vão possuir propriedades semelhantes e a seguir listamos as propriedades que devem ser modificadas em cada um dos botões. As propriedades foram listadas do botão mais à esquerda, o “Ler Agora”, para o mais à direita, o “Limpar Buffer”.

Name	=	ButtonLerAgora
Caption	=	Ler Agora
Font/Style/fsBold	=	true

Name	=	ButtonLerBuffer
Caption	=	Ler Buffer
Font/Style/fsBold	=	true

Name	=	ButtonLimparBuffer
Caption	=	Limpar Buffer
Font/Style/fsBold	=	true

Após a adição desses três botões 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

“Teclado”.



Figura 7: Interface gráfica completa do programa.

Com a interface gráfica completa vamos para a implementação do código. A primeira funcionalidade que vamos implementar é a verificação de botão pressionado no teclado. Essa funcionalidade poderá ser acessada a partir do botão “Ler Agora”. Por isso precisamos criar um manipulador para o evento `OnClick` desse botão. 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.ButtonLerAgoraClick(Sender:
                                                    TObject);
begin
end;
```

Dentro desse manipulador temos que adicionar nosso código para verificação de tecla pressionada. Para isso utilizaremos um método da biblioteca de controle do Kit. Esse método é o `KeyboardReadNow` e possui a seguinte declaração.

```
Function KeyboardReadNow() : Byte;
```

Como podemos ver, esse método não possui nenhuma parâmetro. Ele retorna um valor do tipo `Byte` que indica qual tecla está pressionada no teclado no momento da execução desse comando. O teclado do Kit possui 16 teclas e esse método irá retornar um valor de 0 a 15 para representar essas teclas. Se não houver nenhuma tecla pressionada o valor retornado será 255.

Utilizaremos esse método no manipulador de evento do clique do botão “Ler Agora”. A seguir como ficará esse código.

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

```

var
    tecla : Byte;

begin
    // Lê tecla pressionada
    tecla := kit.KeyboardReadNow;

    ...

end;

```

Nesse início de código declaramos uma variável denominada “tecla” que irá conter o valor da tecla pressionada no momento. Para ler essa tecla chamamos o método KeyboardReadNow e armazenamos seu retorno na variável “tecla”.

Temos agora que verificar se havia alguma tecla pressionada. Fazemos isso com uma estrutura “if” e dependendo do resultado modificamos o texto do Label para avisar que não havia nenhuma tecla pressionada ou para apresentar a tecla que estava pressionada. O código para fazer isso é o seguinte.

```

Procedure TFormMain.ButtonLerAgoraClick(Sender:
                                                    TObject);

var
    tecla : Byte;

begin
    // Lê tecla pressionada
    tecla := kit.KeyboardReadNow;

    // Verifica se havia tecla pressionada
    if tecla = 255 then
        LabelTecla.Caption := 'Nenhuma Tecla
                                Pressionada'
    else
        LabelTecla.Caption := IntToStr(tecla);
end;

```

Se houver alguma tecla pressionada vamos utilizar o método IntToStr para converter o

valor numérico da tecla em uma String de modo que esse valor possa ser apresentado no Label.

Com isso já podemos fazer um teste do nosso programa. 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, mantenha algum botão do teclado do Kit pressionado e clique no botão “Ler Agora”. O texto do Label deverá mostra o botão que estava pressionado no momento do clique. Se o botão for acionado sem que nenhuma tecla do teclado do Kit esteja pressionada, então uma mensagem dizendo que não havia uma tecla pressionada será apresentada no Label.



Figura 8: Aparência do programa após ler uma tecla que estava pressionada.



Figura 9: Resposta do programa quando se tenta ler uma tecla pressionando o botão "Ler Agora" e não há nenhuma tecla pressionada.

Nosso próximo passo é implementar o código para a leitura de teclas no buffer de teclado do Kit. Esse buffer possui 10 posições e armazena as teclas pressionadas. Dessa forma não precisamos ficar verificando continuamente se existe alguma tecla pressionada. O método utilizado para ler esse buffer é o `KeyboardReadBuffer`, que é muito semelhante ao método `KeyboardReadNow`. Esse método retorna um valor do tipo `Byte` com a tecla mais antiga que estava armazenada no buffer. Se não houver nenhuma tecla armazenada o retorno é igual a 255. A seguir a declaração desse método.

```
Function KeyboardReadBuffer() : Byte;
```

Temos que criar um manipulador para o botão “Ler Buffer”. A forma mais fácil de fazer isso é dar um duplo clique sobre esse componente. Com isso será gerado o seguinte código.

```
Procedure TFormMain.ButtonLerBufferClick(Sender :
```

```
TObject);
```

```
begin
```

```
end;
```

E agora vamos adicionar o código para a leitura. Esse código é muito semelhante ao do botão anterior. Veja ele a seguir.

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

```
var
```

```
tecla : Byte;
```

```
begin
```

```
// Lê tecla armazenada no buffer
```

```
tecla := kit.KeyboardReadBuffer;
```

```
// Verifica se o buffer estava vazio
```

```
if tecla = 255 then
```

```
LabelTecla.Caption := 'Nenhuma Tecla no Buffer'
```

```
else
```

```
LabelTecla.Caption := IntToStr(tecla);
```

```
end;
```

Nesse momento podemos testar o programa novamente. 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, pressione alguns botões no teclado e em seguida vá pressionando o botão “Ler Buffer”. Observe que o Label irá apresentar os botões que foram pressionados anteriormente e que tinham sido armazenados no buffer.



Figura 10: Aparência do programa após ler uma tecla do buffer.



Figura 11: Tentativa de leitura de teclas do buffer quando esse está vazio.

A última coisa que falta para ser implementada é a funcionalidade do botão “Limpar Buffer”. Primeiro temos que criar um manipulador para o evento `OnClick` desse botão. Fazemos isso dando um duplo clique sobre esse componente. O seguinte código será criado.

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

Dentro desse manipulador vamos adicionar uma linha que chama o método `KeyboardClearBuffer`. Esse método limpa o buffer do teclado do Kit. Ele possui a seguinte declaração.

```
Procedure KeyboardClearBuffer();
```

O código do manipulador do evento `OnClick` do botão “Limpar Buffer” ficará da seguinte maneira.

```
Procedure TFormMain.ButtonLimparBufferClick(  
Sender: TObject);  
  
begin  
// Limpa buffer do teclado  
kit.KeyboardClearBuffer;  
end;
```

Com isso finalizamos a implementação do código do nosso programa e já podemos testar todas as suas funcionalidades.

4 – Conclusão

Nesse tutorial aprendemos como utilizar o teclado do KDR5000. Criamos um programa que utiliza os três métodos para interação com o teclado permitindo que possamos verificar se uma tecla está pressionada, ler uma tecla do buffer e apagar o buffer do teclado. Os métodos que utilizamos foram o `KeyboardReadNow`, `KeyboardReadBuffer` e `KeyboardClearBuffer`. Com isso podemos utilizar o teclado em qualquer projeto que necessite de uma entrada de dados através de um teclado.