

TUTORIAL

MOTOR DE PASSO

Autor: Tiago Lone
Nível: Básico
Criação: 14/12/2005
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

Esse tutorial irá abordar a programação para controle dos motores de passo do *Módulo de Motores e Displays*. Esse tipo de motor permite um controle preciso do movimento e da posição de seu eixo, sendo ideal para trabalhos onde a precisão é um fator importante. Vamos criar um projeto que permite o ajuste de todos os parâmetros do movimento do eixo desse motor e com isso explicar todos os detalhes de programação necessários para o controle do motor de passo do Kit Didático de Robótica.

2 – Material

Para esse tutorial é necessário o *Módulo Principal* e o *Módulo de Motores e Displays* com apenas o motor de passos conectado à placa principal desse módulo. Para a criação do programa será necessário o Borland Delphi 6. A seguir a imagem da montagem do Kit necessária para esse tutorial.

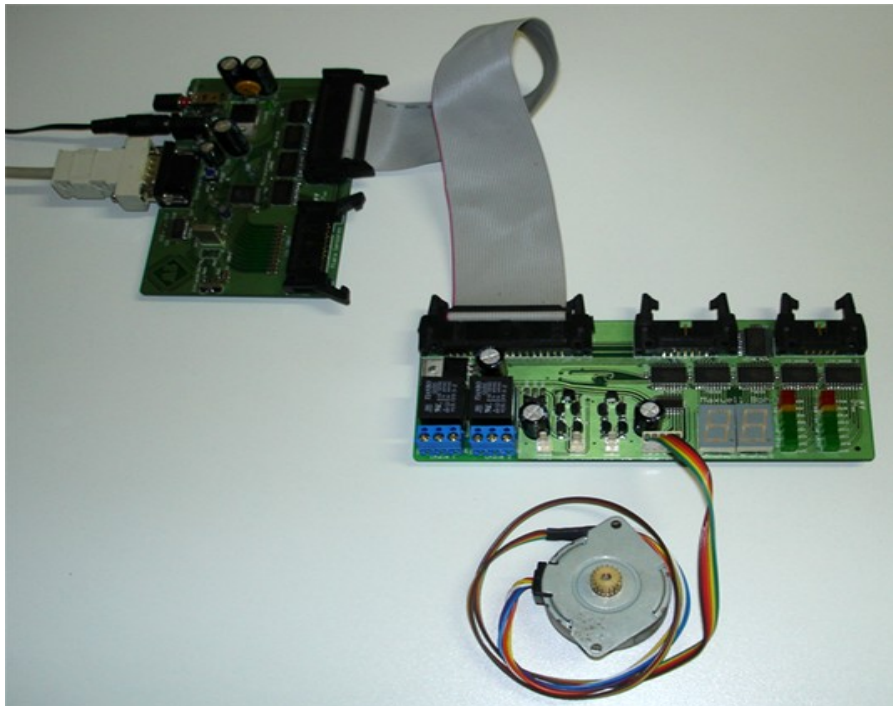


Figura 1: Montagem do Kit utilizada nesse tutorial.

3 – Projeto

O projeto que será feito nesse tutorial é um programa que permite o ajuste de todos os parâmetros do movimento do eixo do motor de passos. Será possível selecionar o sentido da rotação do eixo, a velocidade e o número de passos que o movimento deverá ter. A aparência desse programa será a seguinte.



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

Devemos então criar essa 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.

Essa interface possui cinco tipos diferentes de componentes gráficos. Ela possui três Labels, utilizados para indicar os parâmetro de sentido, velocidade e número de passos. Possui também um ComboBox para seleção do sentido da rotação, dois ScrollBar para ajuste da velocidade e do número de passos, um CheckBox para selecionar se o giro vai ser contínuo e dois Buttons, um para enviar o comando para que o motor gire e outro para enviar um comando para ele parar o movimento. Todos esses componentes encontram-se na aba “Standard” da barra de componentes.



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

Vamos adicionar um Label e um ComboBox para a seleção de sentido da rotação. O componente Label possui o seguinte ícone.



Figura 4: Ícone do componente Label.

E o componente ComboBox possui o seguinte ícone.



Figura 5: Ícone do componente ComboBox.

Adicionamos um Label e modificamos as seguintes propriedades.

Name = LabelSentido
Caption = Sentido:
Font/Style/fsBold = true

Em seguida adicionamos um componente ComboBox e modificamos as propriedades a seguir.

Name = ComboBoxSentido
Style = csDropDownList
Items.Strings = Anti-Horário, Horário
ItemIndex = 0

Com isso nosso Form irá se parecer com o seguinte.



Figura 6: Form após adição dos componentes para seleção de sentido da rotação.

Agora vamos inserir os componentes para ajuste da velocidade de rotação. Aqui vamos utilizar um componente Label e um ScrollBar. O componente ScrollBar, assim como o Label, encontra-se na aba “Standard” da barra de componentes e possui o seguinte ícone.



Figura 7: Ícone do componente ScrollBar.

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

Name = LabelVelocidade
Caption = Velocidade:
Font/Style/fsBold = true

E as seguintes propriedades do ScrollBar.

Name = ScrollBarVelocidade
Max = 255
Position = 127

Assim teremos a seguinte aparência no nosso Form principal.



Figura 8: Form com controles para ajuste de sentido e velocidade da rotação.

O próximo grupo de componentes que vamos adicionar é muito parecido com o anterior, sendo um Label um ScrollBar e um CheckBox. Assim como os componentes anteriores o

CheckBox pode ser encontrado na aba “Standard” da barra de componentes e possui o seguinte ícone.



Figura 9: Ícone do componente CheckBox.

Adicionamos esses três componentes e modificamos as seguintes propriedades do Label.

Name	=	LabelPassos
Caption	=	Passos:
Font/Style/fsBold	=	true

Agora modificamos essas propriedades do ScrollBar.

Name	=	ScrollBarPassos
Max	=	254
Position	=	127

E por fim, alteramos as seguintes propriedades do CheckBox.

Name	=	CheckBoxContinua
Caption	=	Rotação Contínua:

Com isso teremos a seguinte interface.



Figura 10: Interface com todos os controles para ajuste do sentido, velocidade e número de passos da rotação.

Nesse ponto só nos falta adicionar os dois botões para iniciar e para parar o giro do motor de passos. Para isso adicionamos dois componentes Button que podem ser encontrados na aba “Standard” da barra de componentes. Esse componente possui o seguinte ícone.



Figura 11: Ícone do componente Button.

Temos que modificar as seguintes propriedades dos botões. Primeiro do botão para início da rotação.

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

Em seguida as propriedades do botão para parar a rotação.

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

Um último detalhe que vamos modificar é a propriedade Caption do Form principal. Como copiamos o projeto do tutorial Base, essa propriedade possui o valor “Projeto Base”. Vamos

modificar essa propriedade para “Motor de Passo”. Com isso finalizamos a construção de nossa interface gráfica. A seguir a imagem dessa interface finalizada.



Figura 12: Interface finalizada.

O próximo passo é implementar o código para controle do motor de passos. Para isso precisamos criar um manipulador para o evento `OnClick` do botão de início e um para o botão de parada da rotação do motor.

Vamos criar primeiro o manipulador do botão para início da rotação. 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 preparar os parâmetros da rotação e enviar o comando para o Kit. O método que vamos utilizar para acionar o motor de passos é o `StepMotorOn`. Esse método possui a seguinte declaração.

```
Procedure StepMotorOn(dir, speed, steps : Integer);
```

Esse método possui três parâmetros, um indicando o sentido da rotação, um a velocidade e por fim o número de passos do movimento. Para o parâmetro de direção passamos um valor igual a “0” para indicar um movimento anti-horário ou um valor “1” para indicar sentido horário. O parâmetro de velocidade pode receber valores na faixa de 0 a 255, sendo que quanto maior o valor maior será a velocidade da rotação. Apenas uma observação, se esse valor for muito

próximo de 255 o motor pode não responder por limitações mecânicas do motor que limitam sua velocidade. Por fim temos o parâmetro de número de passos. Esse parâmetro pode receber um valor de 0 a 255 indicando o número de passos que o motor deve dar. Existe dois valores especiais nessa faixa, um é o valor 0 que indica que o motor deve ser desligado e outro é o valor 255 que indica que o motor deve girar indefinidamente até receber um outro comando StepMotorOn.

Vamos então implementar o código para acionar o motor de passos. O primeiro passo é preparar os parâmetros. A seguir o início do código desse manipulador.

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

var
    // Armazena direção
    direcao : Integer;

    // Armazena velocidade
    velocidade : Integer;

    // Armazena número de passos
    passos : Integer;

begin
    // Define direção
    if ComboBoxSentido.ItemIndex = 0 then
        direcao := 0
    else
        direcao := 1;

    ...
end;
```

Nesse trecho de código criamos três variáveis do tipo Integer para armazenar os parâmetros de direção, velocidade e número de passos. Em seguida verificamos, através da propriedade ItemIndex do ComboBox, qual é a direção que foi selecionada e setamos a variável “direcao” com o valor correspondente ao sentido. Em seguida vamos verificar o valor para os parâmetros de velocidade e de número de passos.

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

```

var
    // Armazena direção
    direcao : Integer;

    // Armazena velocidade
    velocidade : Integer;

    // Armazena número de passos
    passos : Integer;

begin
    // Define direção
    if ComboBoxSentido.ItemIndex = 0 then
        direcao := 0
    else
        direcao := 1;

    // Define velocidade
    velocidade := ScrollBarVelocidade.Position;

    // Define número de passo
    if CheckBoxContinua.Checked then
        passos := 255
    else
        passos := ScrollBarPassos.Position;

    ...
end;

```

O parâmetro de velocidade será o valor da propriedade Position do ScrollBar. Ele sempre estará na faixa de 0 a 255 porque definimos que a propriedade Max, que define o valor máximo de retorno da posição desse componente, fosse 255 e por padrão o valor mínimo já é 0. Com esse código teremos nosso parâmetro de velocidade armazenado na variável de nome “velocidade”.

Após definir o parâmetro de velocidade vamos definir o número de passos. Para isso temos que verificar se a opção de rotação contínua está selecionada. Se essa opção estiver selecionada vamos definir que o parâmetro de número de passos será igual a 255, que como vimos

anteriormente, indica que o giro será contínuo. Caso essa opção não esteja marcada, vamos definir que o número de passos é igual à propriedade Position do ScrollBar correspondente ao parâmetro de número de passos.

Com isso temos armazenado nas variáveis “direcao”, “velocidade” e “passos” todos os parâmetros que necessitamos, faltando apenas chamar o método que controla o motor de passo. É isso que fazemos em seguida.

```
Procedure TFormMain.ButtonEnviarClick(Sender:
                                                    Tobject);

var
    // Armazena direção
    direcao : Integer;

    // Armazena velocidade
    velocidade : Integer;

    // Armazena número de passos
    passos : Integer;

begin
    // Define direção
    if ComboBoxSentido.ItemIndex = 0 then
        direcao := 0
    else
        direcao := 1;

    // Define velocidade
    velocidade := ScrollBarVelocidade.Position;

    // Define número de passo
    if CheckBoxContinua.Checked then
        passos := 255
    else
        passos := ScrollBarPassos.Position;

    // Envia comando para acionar o motor de passo
```

```
Kit.StepMotorOn(direcao, velocidade, passos);  
end;
```

Essa última linha que adicionamos apenas chama o método StepMotorOn com os parâmetros que armazenamos anteriormente. Com isso já podemos fazer um teste de controle do motor de passo do *Módulo de Motores e Displays*. 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, modifique alguns parâmetros e pressione o botão enviar, o motor de passo deverá se movimentar. Teste algumas vezes com parâmetros diferentes.

Vamos implementar agora o código do manipulador de eventos do botão “Parar”. Como vimos anteriormente, para parar um motor temos que passar como parâmetro de número de passos o valor “0”. Então vamos 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 motor  
    Kit.StepMotorOn(0, 0, 0);  
end;
```

Pronto, temos um programa que controla o motor de passo e ajusta todos os parâmetros do movimento. Sua aparência final ficou assim.



Figura 13: Programa finalizado.

Podemos selecionar nessa interface o sentido da rotação, a velocidade e o número de

passos. Em seguida é preciso apenas pressionar o botão enviar e o motor irá se movimentar. Caso seja necessário interromper o movimento do motor antes do término de todos os seus passos então podemos pressionar o botão “Parar” e o movimento será interrompido.

4 – Dicas

Quando selecionamos o CheckBox da opção de rotação contínua, o valor do ScrollBar de número de passos é ignorado no nosso código. Então seria interessante desabilitar esse componente pois ajustar seu valor não irá interferir no movimento do motor.

Para fazer isso temos que implementar uma verificação no evento OnClick do CheckBox, que é executado toda vez que o CheckBox é “clicado” e conseqüentemente tem seu estado modificado. Essa verificação testa o estado do CheckBox, se este estiver selecionado então desabilitamos o controle ScrollBar de número de passos, caso contrário esse componente é habilitado.

Para criar o manipulador do evento OnClick do CheckBox podemos apenas dar um duplo clique sobre o CheckBox e o seguinte código será criado.

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

Adicionando o código responsável pela verificação do estado do CheckBox e por habilitar e desabilitar o ScrollBar temos o seguinte.

```
Procedure TFormMain.CheckBoxContinuaClick(Sender:
                                                    TObject);
begin
    // Verifica se o CheckBox está selecionado
    if CheckBoxContinua.Checked then
        // Desabilita componente ScrollBar de passos
        ScrollBarPassos.Enabled := false
    else
        // Habilita componente ScrollBar de passos
        ScrollBarPassos.Enabled := true;
end;
```

Rode o programa e observe que quando a opção de rotação contínua for selecionada o ScrollBar relativo ao ajuste do número de passos será desabilitado. A seguir uma imagem com o ScrollBar desabilitado e uma com ele habilitado.



Figura 14: Quando o CheckBox de rotação contínua for selecionado o ScrollBar de número de passos é desabilitado.



Figura 15: Quando o CheckBox de rotação contínua não estiver selecionado o ScrollBar de número de passos fica habilitado.

Outra dica para incrementar ainda mais esse programa seria que a velocidade do motor de passos fosse atualizada imediatamente no motor sem a necessidade de se pressionar o botão "Iniciar". Isso poderia ser feito capturando o evento OnChange do ScrollBar que ajusta a velocidade da rotação, e dentro desse evento reenviar o comando para acionamento do motor com a nova velocidade.

5 – Conclusão

Nesse tutorial vimos como controlar o motor de passo do *Módulo de Motores e Displays*. Com o projeto que criamos foi possível entender o funcionamento do método `StepMotorOn` e o significado de todos os seus parâmetros. Com isso já é possível explorar ao máximo tudo que o motor de passo do Kit Didático de Robótica pode oferecer.