

TUTORIAL SERVO-MOTOR

Autor: Luís Fernando Patsko e Tiago Lone
Nível: Intermediário
Criação: 26/12/2005
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 Servo-Motores do MEC1000 ou do *Módulo de Entradas, Saídas e Servo-Motores* do KDR5000. Esse tipo de motor fornece força suficiente para movimentar estruturas mecânicas consideráveis e além disso possui um controle da posição do eixo. Vamos criar um projeto que permite ligar e desligar os servo-motores e ajustar a posição de seu eixo. Também veremos como um servo-motor pode ser conectado ao Kit, e assim desfrutar de todas as suas funcionalidades.

2 – Material

Para esse tutorial é necessário o *Módulo Principal* e o *Módulo de Entradas, Saídas e Servo-Motores* com um ou mais servo-motores conectados à ele. Para maiores informações de como ligar os servo-motores ao módulo ver o tópico sobre servo-motores deste tutorial.

Para a criação do programa será necessário o Borland Delphi 6. A seguir a imagem da montagem do Kit necessária para esse tutorial.

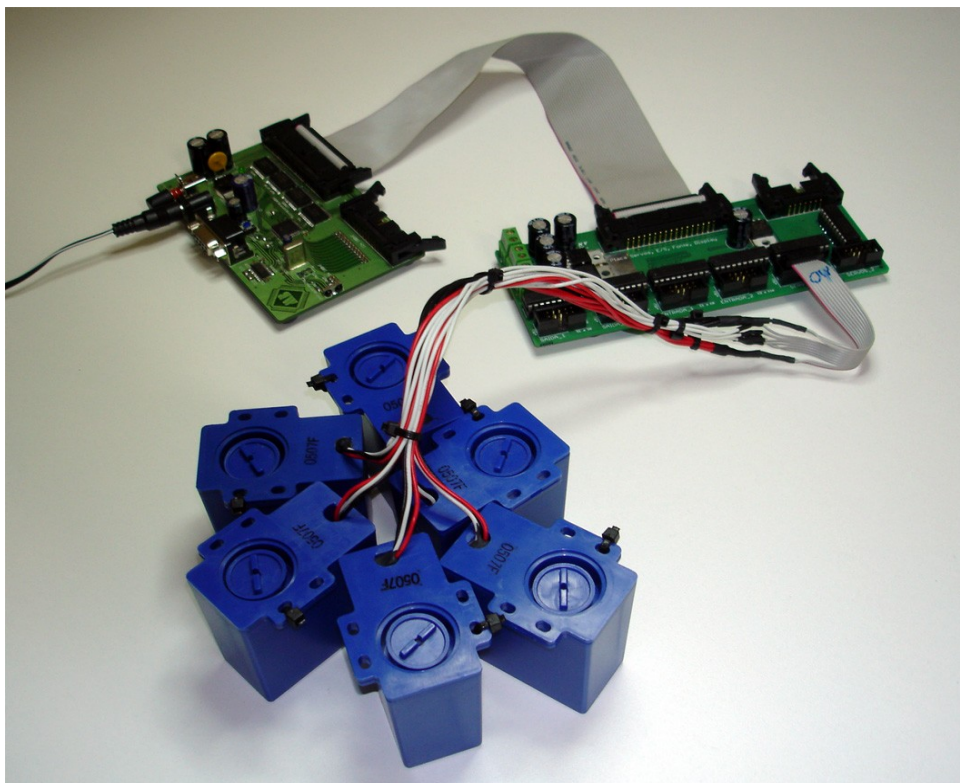


Figura 1: Montagem utilizada nesse tutorial.

3 – Projeto

Nesse tutorial vamos desenvolver um programa para controlar os servo-motores do

Módulo de Entradas, Saídas e Servo-Motores e do MEC1000. Será possível ligar e desligar os servo-motores além de ajustar a posição de seu eixo. A aparência desse programa será a seguinte.



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

Nosso primeiro passo é 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 dois componentes do tipo Label que são utilizados para identificação das configurações. Outro componente é um ComboBox que é utilizado para selecionar o servo-motor que queremos controlar. Existe também um ScrollBar para ajustar a posição do eixo do motor e por fim dois botões utilizados para ligar e desligar o motor selecionado. 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 do motor que queremos controlar. 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 o Label e modificamos as seguintes propriedades.

Name = LabelServoMotor
Caption = Servo-Motor:
Font/Style/fsBold = true

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

Name = ComboBoxServoMotor
Style = csDropDownList
Items.Strings = Servo-Motor 0, Servo-Motor 1, Servo-Motor 2,
Servo-Motor 3, Servo-Motor 4, Servo-Motor 5,
Servo-Motor 6, Servo-Motor 7
ItemIndex = 0

Com isso nosso Form irá se parecer com o seguinte.



Figura 6: Form após adição dos componentes para seleção do servo-motor que será controlado.

Agora vamos inserir os componentes para ajuste da posição do eixo do motor. 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 = LabelPosicao
Caption = Posição:
Font/Style/fsBold = true

E as seguintes propriedades do ScrollBar.

Name = ScrollBarPosicao
Max = 255
Position = 127

Assim teremos a seguinte aparência no nosso Form principal.



Figura 8: Form com os controles para ajuste do motor que será controlado e da posição do eixo desse.

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



Figura 9: Ícone do componente Button.

Temos que modificar as seguintes propriedades dos botões. Primeiro do botão para ligar o motor.

Name = ButtonLigar
Caption = Ligar
Font/Style/fsBold = true

Em seguida as propriedades do botão para desligar o motor.

Name = ButtonDesligar
Caption = Desligar
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 “Servo-Motores”. Com isso finalizamos a construção de nossa interface gráfica. A seguir a imagem dessa interface finalizada.



Figura 10: Interface finalizada.

O próximo passo é implementar o código para controle dos servo-motores. Para isso precisamos criar um manipulador para o evento OnClick do botão “Ligar” e um para o botão “Desligar”.

Vamos criar primeiro o manipulador do botão que liga os motores. Para fazer isso podemos selecionar o componente Button que possui o texto “Ligar”, 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.ButtonLigarClick(Sender:
                                                                    TObject);
begin
end;
```

Dentro desse manipulador vamos implementar o código que liga o motor. O método que vamos utilizar para ligar os servo-motores é o ServoMotorOn. Esse método possui a seguinte declaração.

```
Procedure ServoMotorOn(motor : Integer;  
                        pos : Byte);
```

Esse método possui dois parâmetros, um indicando o servo-motor que será ligado e outro indicando a posição que o eixo do motor deverá se manter. O *Módulo de Entradas, Saídas e Servo-Motores* suporta até 8 servo motores, logo o primeiro parâmetro pode variar de 0 a 7. O parâmetro de posição do eixo pode variar de 0 a 255.

Normalmente os servo-motores permitem uma movimentação de seu eixo de apenas 180 graus. O valor do parâmetro de posição ajusta em que posição, desse curso de 180 graus, o eixo do motor deverá permanecer. Os quatro primeiros servo-motores do *Módulo de Entradas, Saídas e Servo-Motores* possuem um controle preciso de sua posição. Já os quatro últimos possuem um controle menos preciso permitindo posicionar o eixo apenas no início, meio ou fim do curso de 180 graus.

Em muitas aplicações um curso de 180 graus de movimento não é suficiente. Para resolver esse problema podemos fazer uma modificação nos servo motores para que eles possam ser utilizados em aplicações que necessitem de giro contínuo. No entanto, com essa modificação perdemos o controle da posição do eixo do motor e teremos apenas controle do sentido da rotação. Assim, se o parâmetro de posição for igual a “0” o servo-motor modificado irá girar para um lado e se esse for “255” o motor irá girar para o lado oposto. Há nesse curso um tutorial que ensina a fazer a modificação para giro contínuo nos servo-motores.

Agora vamos implementar o código para acionar os servo-motores. A seguir o código do manipulador do botão “Ligar”.

```
Procedure TFormMain.ButtonLigarClick(Sender :  
                                     TObject);  
  
begin  
    // Liga e posiciona eixo do servo motor  
    kit.ServoMotorOn( ComboBoxServoMotor.ItemIndex,  
                    ScrollBarPosicao.Position );  
  
end;
```

Como podemos observar, é muito simples ligar um servo-motor. Foi necessário apenas chamar o método ServoMotorOn com os parâmetros que obtemos dos componentes ComboBox e ScrollBar.

Os itens do ComboBox foram adicionados na ordem de modo que o índice deles dentro desse componente equivale ao parâmetro que deve ser passado ao método ServoMotorOn para

indicar o servo-motor que eles correspondem. Dessa forma é necessário apenas saber o índice do servo-motor selecionado no ComboBox e passar esse como parâmetro.

O parâmetro de posição 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 isso, se os servo-motores já estiverem conectados, podemos fazer um teste de controle dos servo-motores do *Módulo de Entradas, Saídas e Servo-Motores*. Se os servo-motores ainda não foram conectados é interessante ler a seção sobre servo-motores no final desse tutorial ou deixar para fazer o teste mais para frente.

Para testar o 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, selecione um motor que está conectado e pressione o botão ligar. O motor deverá se posicionar na posição selecionada no ScrollBar de ajuste de posição. Teste algumas vezes com posições diferentes.

Vamos implementar agora o código do manipulador de eventos do botão “Desligar”. Para isso vamos utilizar o método ServoMotorOff. Esse método possui a seguinte declaração.

```
Procedure ServoMotorOff(motor : Integer);
```

O único parametro desse método é o servo-motor que queremos desligar. Então vamos criar o manipulador do evento OnClick do botão “Desligar” dando um duplo clique sobre ele e depois inserir o seguinte código.

```
Procedure TFormMain.ButtonDesligarClick(Sender :  
TObject);  
  
begin  
    // Desliga servo motor  
    kit.ServoMotorOff(ComboBoxServoMotor.ItemIndex);  
end;
```

Pronto, temos um programa que controla os servo-motores e ajusta a posição de seu eixo. Sua aparência final ficou assim.



Figura 11: Programa finalizado.

Podemos selecionar nessa interface o servo que vamos controlar e definir a posição de seu eixo. Em seguida é preciso apenas pressionar o botão “Ligar” e o motor será ligado. Para desligar o motor basta pressionar o botão “Desligar”. A seguir veremos um pouco sobre como trabalhar com os servo motores.

4 – Ligação e Adaptação de Servo-Motores

As saídas para conexão com servo-motores foram desenvolvidas tendo como objetivo possibilitar seu uso nas mais variadas aplicações. A utilização de servo-motores, em conjunto com as funcionalidades do Kit, possibilita a elaboração de sistemas mecânicos automatizados, como robôs.

Para criar e controlar esses esquemas, é necessário conectar adequadamente os servo-motores. Torna-se indispensável conhecer a pinagem das saídas de servo-motores, baseando-se na imagem a seguir.

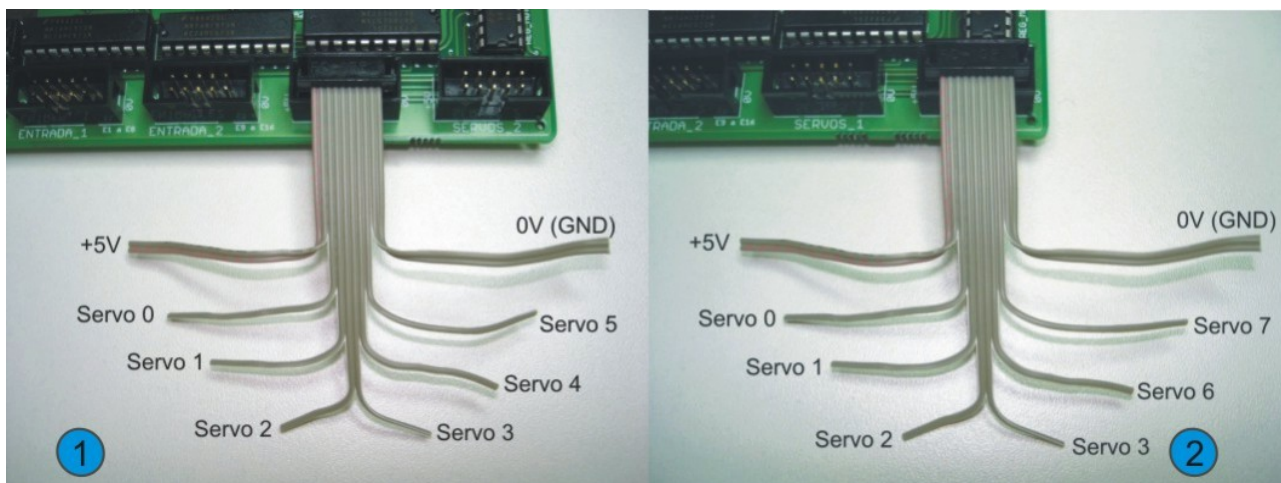


Figura 12: Pinagem dos conectores de controle de servo-motores. 1-Conector 0. 2-Conector 1.

Analisando a imagem acima, podemos perceber algumas características desses

conectores. Cada um deles possui quatro pinos de alimentação, sendo dois de 5V e dois de 0V. Também é possível perceber que cada conector possui quatro saídas de controle compartilhados e duas saídas de controle únicas. Cada sinal de controle compartilhado pode ser usado em ambos os conectores simultaneamente, sabendo que eletricamente são o mesmo sinal. Por exemplo, um servo-motor conectado ao pino “Servo 2” do Conector 1 terá comportamento idêntico a um servo ligado ao pino “Servo 2” do Conector 0. As saídas exclusivas de cada Conector são totalmente independentes, ou seja, um comando enviado a um servo-motor no Conector 0 não afetará o estado de nenhum servo-motor ligado ao Conector 1.

A diferença entre os conectores de servo-motores do KDR5000 e do MEC1000 está na sua quantidade. O MEC1000 possui apenas um conector, cuja pinagem é a mesma do Conector 0 do KDR5000, restringindo a 6 o número máximo de servo-motores conectados.

Fazendo uso do programa criado com esse tutorial, das funcionalidades do Kit e conhecendo o esquema dos conectores, o usuário é então capaz de criar um sistema de automação, estabelecendo relações entre os componentes do Kit, de modo que eles interajam de acordo com a aplicação a qual sejam destinados.

Os servo-motores originalmente possuem um sistema controle de seu eixo, possibilitando um posicionamento preciso. Quando um sinal é enviado ao servo-motor, o controle verifica a posição atual do eixo e compara com a posição desejada, fazendo então o ajuste necessário. Entretanto, esse controle acaba gerando uma limitação no movimento, que de acordo com o modelo, pode variar entre 90 a 270 graus. Dependendo da aplicação, essa limitação pode ser um problema principalmente quando é necessário um motor que possua mais torque que os motores convencionais. Uma solução para esse problema é a conversão do motor para a capacidade de exercer giro contínuo. Desse modo, embora não seja mais possível controlar a posição do seu eixo, o motor pode girar livremente. A conversão para giro contínuo é um processo simples, mas trabalhoso, que será explicado em mais detalhes num tutorial a parte.

A utilização dos servo-motores encontra uma limitação na fonte de alimentação do KDR5000 e do MEC1000. É necessário o uso de alimentação externa quando houver o desejo de ligar mais de um servo-motor simultaneamente, pois o seu consumo não é suportado pela fonte de alimentação, que é capaz de fornecer apenas algumas centenas de miliamperes. Como consequência, a fonte poderá superaquecer e danos podem ser causados ao equipamento.

5 – Dicas

Seria interessante que quando modificamos a posição do eixo do motor isso se refletisse imediatamente sem a necessidade de pressionar o botão “Ligar”. Para fazer isso precisamos interceptar o evento OnChange do ScrollBar de ajuste de posição. Temos então que criar um manipulardo para esse evento. Fazemos isso dando um duplo clique sobre o componente ScrollBar. O seguinte código será criado.

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

Dentro desse manipulador vamos inserir o código para reposicionar o eixo do motor. Ele ficará assim.

```
Procedure TFormMain.ScrollBarPosicaoChange(  
Sender: TObject);  
  
begin  
    // Liga e posiciona eixo do servo-motor  
    kit.ServoMotorOn( ComboBoxServoMotor.ItemIndex,  
                    ScrollBarPosicao.Position );  
  
end;
```

Pronto, com isso ao modificarmos o ajuste de posição essa será refletida imediatamente pelo eixo do motor.

6 – Conclusão

Nesse tutorial vimos como controlar os servo-motores tanto do *Módulo de Entradas, Saídas e Servo-Motores* quanto do MEC1000. Com o projeto que criamos foi possível entender o funcionamento do método `ServoMotorOn` e `ServoMotorOff` além do significado de seus parâmetros. Vimos também o que deve ser feito para ligar um servo às portas de controle do Kit. Com isso já é possível explorar ao máximo tudo que o servo-motor pode oferecer.