

Introdução a lógica de relés

Instruções da Lógica de Relés

Este texto foi concebido para familiariza-lo com a operação do software SCPWS1 e conduzi-lo passo a passo no processo de criação, edição e teste de programas para PLC utilizando as instruções LADDER. Para iniciar um novo programa, vá até o menu "Arquivo" >> "Configurações de Hardware" e escolha o modelo de controlador desejado e então a opção "Aceitar" para aceitar esta opção. Caso não exista uma porta de comunicação no sistema pode aparecer uma mensagem de erro que deve ser ignorada. Caso apareça a mensagem de erro, basta cancela-la e clicar no botão sair ao invés de aceitar.

A programação utiliza três menus para a inserção dos elementos no programa. São eles:

- a- Menu de bobinas e funções de final de linha
- b- Menu de contatos e edição de programa
- c- Funções matemáticas e de comparação entre números inteiros

Menu bobinas e funções de final de linha

O menu de bobinas e funções de final de linha é apresentado na figura 1.



Fig. 1 – Menu de bobinas e funções de final de linha

As funções de bobina são compostas pela bobina simples, bobina set e bobina reset e são descritas a seguir em maiores detalhes.



Bobina simples, deve ser inserida no final da linha torna o operador verdadeiro enquanto o resultado da lógica da linha for verdadeira. Pode trabalhar com operador do tipo Q(saída digital) R(rele auxiliar) CR(reset de contador).



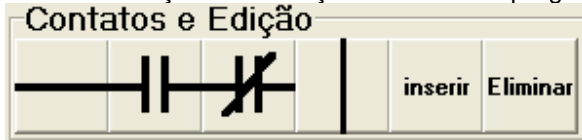
Bobina Set, deve ser inserida no final da linha torna o operador verdadeiro quando a lógica da linha sofrer uma transição de falso para verdadeiro. Pode trabalhar com operador do tipo Q(saída digital) R(rele auxiliar).



Bobina Reset, deve ser inserida no final da linha torna o operador falso quando a lógica da linha sofrer uma transição de falso para verdadeiro. Pode trabalhar com operador do tipo Q(saída digital) R(rele auxiliar).

Menu contatos e edição

Este menu é utilizado para inserção de conexão horizontal vertical contatos abertos e fechados, além de inserção e eliminação de linhas no programa.



Preenche conexão horizontal.



Preenche conexão vertical.



Insere linha em branco acima.



Elimina linha atual do programa.



Insere contato normalmente aberto que pode estar associado a operador tipo I(entrada digital) Q(saída digital), R(rele auxiliar), T(temporizador), C(contador).



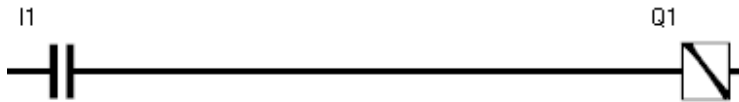
Insere contato normalmente fechado que pode estar associado a operador tipo I(entrada digital) Q(saída digital), R(rele auxiliar), T(temporizador), C(contador).


Simulador de entradas e saídas do CLP

Para que seja possível testar os programas desenvolvidos pode-se utilizar o simulador do CLP, disponível no menu "simulações" >> Simulador de I/O CPWS1.



Criação de Programas com o SCPWS1

Introduza o seguinte programa de apenas uma linha (rung), o qual consiste de uma só instrução de entrada contato NA e uma instrução de saída do tipo bobina Simples. Existe mais de uma forma de fazer isto, mas por enquanto mostraremos a forma que consideramos a mais utilizada.



Primeiro de um clique com o mouse a esquerda da área de programação, deixando o primeiro ponto de inserção marcado com um retângulo, em seguida clique  presente no menu contatos e edição. Depois de inserir o elemento, ele deve ser endereçado, no nosso caso como uma entrada digital com operador tipo I, endereço 1.

Se você acidentalmente inserir uma instrução e deseja remove-la, de um clique com o botão esquerdo do mouse na instrução que deve ser removida e logo em seguida pressione a tecla “DEL” no teclado.

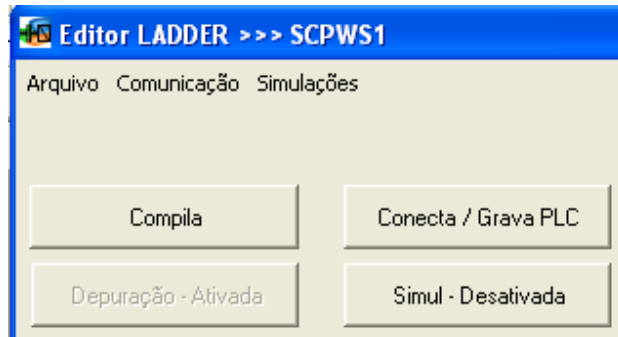
Preencha a linha até o final com a conexão horizontal  e de um clique com o botão esquerdo do mouse na instrução de bobina simples  e esta será inserida no final da linha a direita, assim que o elemento é inserido deve-se digitar seu endereço, no nosso programa exemplo deve ser Q1.

Compilação do programa

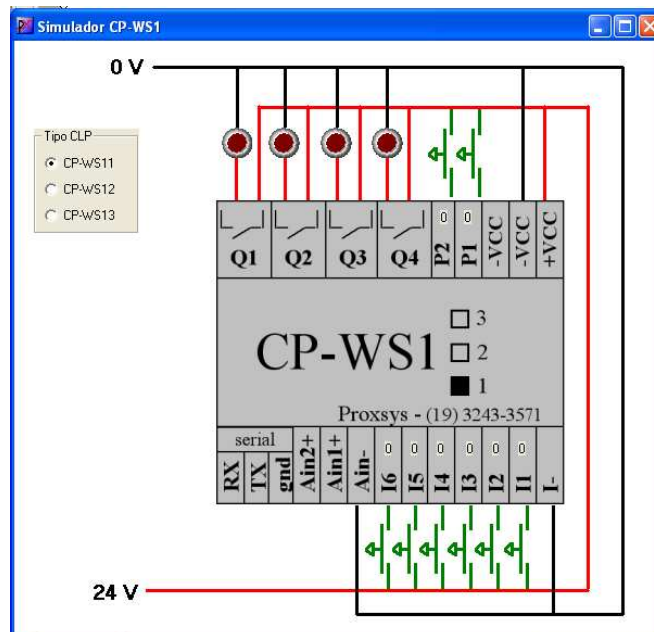
O programa desenvolvido deve ser compilado antes de ser simulado ou enviado para a memória do controlador. Para fazer isso, clique no botão Compila, aparecerá uma janela com o progresso da compilação e também a quantidade total de memória utilizada para o modelo de controlador escolhido.



Quando a compilação for concluída, feche a janela de STATUS da compilação e clique no botão de simulação que deve estar com a inscrição “Simul-Desativada”.



Em seguida, escolha o menu Simulações >> Simulador I/O CPWS1, a janela abaixo deve abrir:

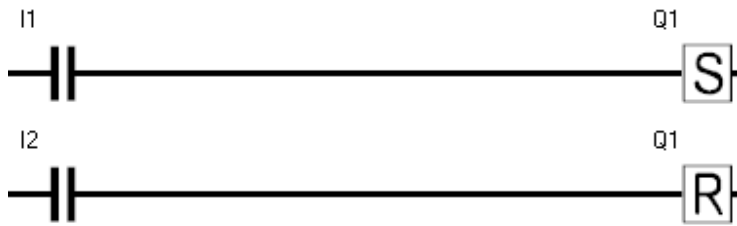


O simulador permitirá testar o funcionamento do programa, no nosso caso ao pressionarmos o botão presente na entrada I1, o contato da saída Q1 se fecha e a lâmpada deverá ligar.

Assim com o programa no modo de simulação, um acionamento na chave conectada na entrada I1, fará a saída Q1 ser acionada.

Para/Partida (Stop/Start) usando SET e RESET

Crie as linhas de programa mostradas abaixo. Uma vez introduzidas as linhas de forma correta, compile o programa e faça a simulação.



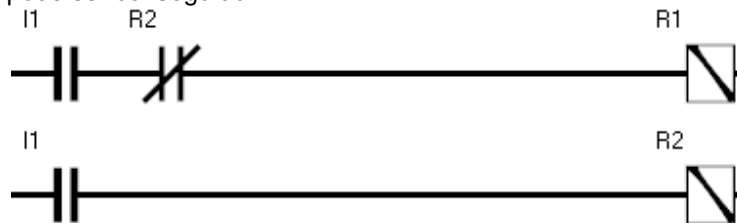
Ative as chaves Start(I1) e Stop(I:2) e assegure que as instruções de saída SET e RESET respondam como estipulado em seu texto. Uma vez que tenha a saída ligada é possível desligá-la quando perdemos a alimentação na chave STOP ? Qual a solução para este problema?

Circuito para ligar e desligar uma saída com uma única botoeira

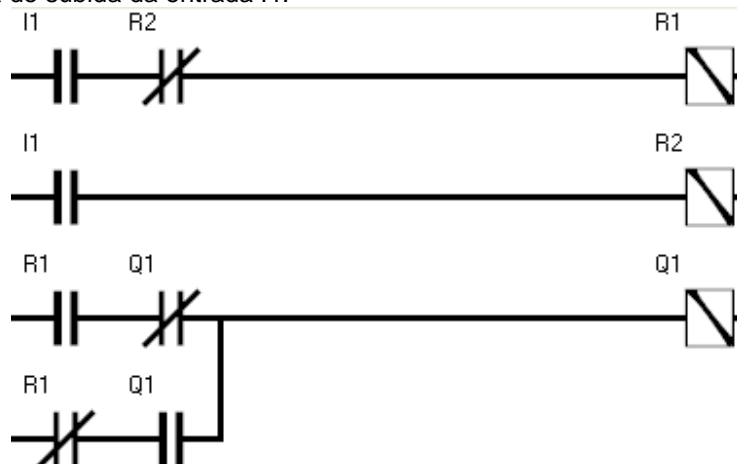
Existem situações no controle discreto em que o estado de uma variável não é suficiente como informação. Mais do que isto, o momento em que ocorre a comutação é relevante. O fato de detectarmos o instante de subida do sinal de desligado para ligado chama-se detecção de borda.

No programa anterior, utilizamos os operadores “I” e “Q” que estão relacionados com entradas e saídas do controlador. Em alguns casos, é necessário memorizar o estado de variáveis internas na memória do controlador. No caso de um sinal discreto, podemos utilizar o operador “R”.

Para avaliarmos esta condição no PLC, utilizamos o conceito do ciclo de varredura, pois a informação ficará disponível apenas durante o tempo de um ciclo. O trecho de programa abaixo ilustra como isso pode ser conseguido.



Analisando o programa, percebe-se que quando I1 é levado a nível lógico alto, R1 também sobe, pois o contato fechado série de R2 assim o permite. Porém a partir do segundo ciclo de varredura R2 torna-se ativo, ocasionando a queda de R1. Como conseqüência R1 ficou ligado ou em nível alto durante um ciclo de varredura, a partir do instante em que I1 foi ativado. Diz-se então que R1 sinalizou a borda de subida da entrada I1.



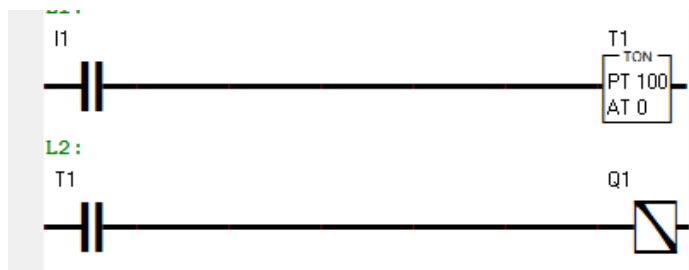
Uma aplicação prática e muito útil é a necessidade de ligar e desligar uma saída do controlador utilizando apenas uma entrada. Este tipo de aplicação teria uma grande complexidade de implementação caso fossem utilizados apenas botoeiras e reles eletromecânicos, porém o uso do CLP facilita bastante esta implementação. Monte o circuito e teste seu funcionamento.

Temporizadores - TON

TON

Temporizador ativo na energização. O TON inicia a temporização após sua entrada ser verdadeira e ativa o contato associado ao operador após decorrido o tempo de preset (PT). Pode trabalhar com operador tipo T (timer) deve ter o parâmetro de tempo de preset informado durante a inserção do bloco ou após bloco inserido com duplo-click sobre o parâmetro PT permitindo sua edição.

Uma outra função muito importante na programação em lógica de reles é o temporizador. No editor ladder SCPws1, temos disponível um temporizador do tipo TON – Timer ON, ou seja, a ação de temporização é executada com retardo na ativação.



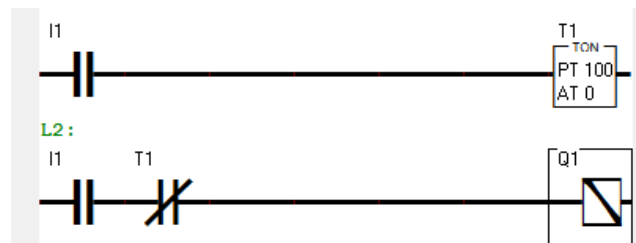
Monte o circuito da figura acima, ele exemplifica o funcionamento do temporizador. A lógica de entrada do bloco controla seu funcionamento e neste caso quando a entrada I1 é ativada o timer T1 inicia a contagem de tempo e após 1 segundo a saída Q1 é ativada. Observe que a base de tempo do temporizador é 10 mS, com isso para atingir o tempo de 1 segundo é necessário ajustar o preset do timer para 100. Se quiséssemos tempo de 10 segundos o ajuste seria 1000, 50 segundos 5000 e assim por diante. O maior valor de ajuste possível para o bloco temporizador é 32000, ou seja, 320 segundos.

Para tempos maiores que 320 segundos é necessário cascatear timer e contador. Isso será mostrado mais adiante. No exemplo mostrado a entrada foi acionada houve a contagem do tempo e a saída foi acionada.

Configuração de Temporizadores – TON como TOF

Em alguns casos pode ser necessária uma operação em que a ação de temporização é executada com retardo na desativação, ou seja, queremos acionar uma entrada e fazer a saída ficar ativa por um determinado tempo e então desativar. Operação contrária ao TON e em alguns casos chamada TOF.

Isso será possível utilizando a solução mostrada na figura a seguir. Neste exemplo ao acionar a entrada I1 o timer T1 começa a contar o tempo ao mesmo tempo que a saída Q1 é ativada na linha seguinte pois I1 dá condição e como T1 ainda não chegou ao final da contagem de tempo o contato fechado de T1 permite o acionamento da saída Q1.



Quando o tempo de 1 segundo for atingido, o contato normalmente fechado de T1 irá abrir, fazendo a saída Q1 desligar gerando a ação de retardo no desligamento de Q1. Variações deste circuito podem ser utilizadas sempre utilizando a mesma lógica de acionamento na entrada do bloco TON e um contato normalmente fechado do timer no circuito que queremos a operação contrária.

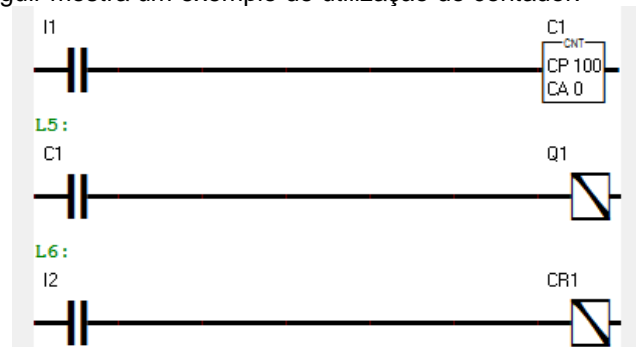
Contador – função CNT

CNT

Contador. O CNT conta a transição da lógica de entrada quando está varia de falso para verdadeiro até ser atingido o valor de preset (CP), quando então congela a contagem e ativa contato tipo C (contador) associado ao operador. Pode trabalhar com operador tipo C (contador). Deve ter o parâmetro de preset de contagem informado durante a inserção do bloco ou após o bloco inserido com duplo-click sobre o parâmetro CP permitindo sua edição.

Para zerar a contagem deve ser utilizada uma bobina simples com o operador CR (reset do contador) e lógica apropriada para ativação desta conforme a necessidade.

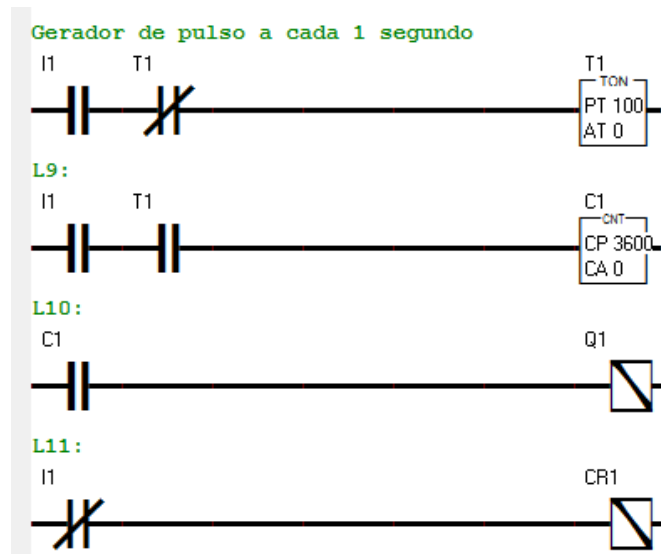
O contador também é uma função bastante utilizada em lógica de relés e está disponível no SCPws1. A figura a seguir mostra um exemplo de utilização do contador.



Neste exemplo cada transição de desligado para ligado de I1 gera um pulso que é contado até que o limite de preset “CP”, que é 100, seja atingido. Quando a contagem chegar a 100, o contato C1 associado ao contador fecha acionando a saída Q1. Para resetar ou zerar o contador é necessário acionar a bobina CR1. No exemplo quando a entrada I2 do controlador é acionada o valor da contagem voltará a zero e nenhuma contagem é realizada até o desligamento de I2.

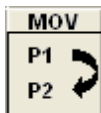
Cascadeamento de Timer e Contador

Com o objetivo de atingir tempos maiores que 320 segundos é necessário cascatear temporizadores e contadores, isso quer dizer, o temporizador irá gerar pulsos a intervalos de tempo pré-definidos e o contador irá contar estes pulsos. A figura a seguir mostra como isso é possível.

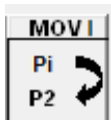


No circuito da figura acima quando a entrada I1 é acionada são gerados pulsos de 1 segundo, que são contados pelo contador C1. Como o preset de C1 é 3600 ele contará 3600 segundos = 1 hora e acionará a saída Q1. Quando a entrada I1 for desligada o contador é zerado e o gerador de pulsos também é paralisado.

Movimentação de variáveis



Movimentação de variáveis. Permite movimentar valores para variáveis inteiras. O parâmetro P1 representa uma memória inteira do tipo M, L, T, C ou um valor numérico constante inteiro precedido pela letra K que é a origem do dado. O parâmetro P2 representa uma memória inteira tipo M ou long L que é o destino de P1. Para inserir uma memória nos parâmetros P1 ou P2, basta digitar o número da memória desejada durante a implantação do bloco. No caso de necessidade de inserção de uma constante inteira tipo K, basta digitar a letra K e em seguida o valor numérico constante inteiro. O bloco de movimentação de variáveis é único na linha e posicionado no final da mesma e deve ser precedido por um contato de controle, mesmo que a movimentação seja necessária constantemente. Neste caso um contato normalmente fechado de um rele auxiliar tipo R, sempre fechado, deve ser utilizado.



Movimentação indexada de variáveis. Permite movimentar valores de uma variável indexada para variáveis inteiras tipo M ou long tipo L. O parâmetro Pi representa uma memória do tipo M cujo valor representa o número da memória M que será a origem do dado a ser transferido para a memória M ou L endereçada no parâmetro P2 que é o destino.

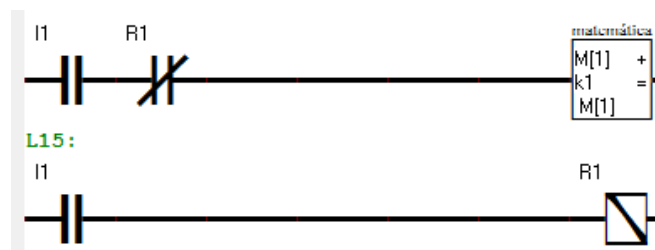
Operação matemática de final de linha - MAT



Operação matemática de final de linha ativada pela lógica de entrada do bloco.

Neste bloco P1 representa o primeiro operando, OP representa a operação matemática, P2 o segundo operando e R o resultado. Os blocos de operação matemática de final de linha devem ser inseridos no final da linha e são executados quando a lógica de entrada é verdadeira.

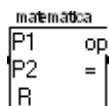
Os operandos podem ser posição de memória inteiras tipo M, long tipo L ou valores de constantes inteiras precedidos pela letra K (constante inteira). O resultado deve ser obrigatoriamente uma posição de memória inteira M ou long L.



Na figura acima cada acionamento de I1 faz com que a memória inteira M1 seja incrementada em 1, até que um novo ciclo de liga/desliga de I1 seja realizado.

Menu Matemáticas e comparação

Este menu contém funções para operação com números inteiros de 16 bits ou long de 32 bits. É possível realizar soma, multiplicação, subtração e divisão com números inteiros variáveis de 16 bits, operadores M, na faixa de 0 a +32767 e com números do tipo long de 32 bits operador do tipo L, na faixa de 0 a +2147483647.



Neste bloco P1 representa o primeiro operando, OP representa a operação matemática, P2 o segundo operando e R o resultado. Os blocos de operação matemática devem ser inseridos em linhas exclusivas em um total máximo de 6 blocos por linha.

Os operandos podem ser posição de memória inteiras tipo M, long tipo L ou valores de constantes inteiras precedidos pela letra K (constante inteira). O resultado deve ser obrigatoriamente uma posição de memória inteira M ou long L.



O bloco comparador compara os operadores P1 e P3, de acordo com o operador P2, (P1>P3), (P1<P3) ou (P1=P3). O resultado da comparação faz a saída do bloco ser verdadeira ou falsa e deve ser associado a uma bobina simples, set ou reset. Só é possível inserir um bloco comparador por linha de programa. Os valores de P1, P2, e P3 devem ser inseridos durante a implantação do bloco no programa P1 e P3 devem ser posições de memória tipo M, L, valor atual de temporizador (entrar como operador T), valor atual de contador (entrar como operador C), constantes inteiras precedidas pelo letra K (constante inteira).