

Manual de Utilização

■ Automação

Software de Programação TwidoSuite

Sumário

Configuração do Hardware	3
Programação.....	7
Descrição das Funções das Instruções Básicas	12
Blocos de Funções.....	17
Debug.....	34
Tabela de animação.....	35
Cabo de Programação e Conversores.....	36

Software

Configuração do Hardware

Antes de iniciarmos a programação, necessitamos informar a CPU quais os cartões iremos utilizar em nossa aplicação. Isto deve ser configurado no software e depois descarregado no PLC. Necessitamos fazer isto para que o PLC reconheça corretamente o endereçamento que iremos atribuir para as entradas e saídas.

1. Escolher a língua Inglês ou Francês;
2. Modo de Programação.

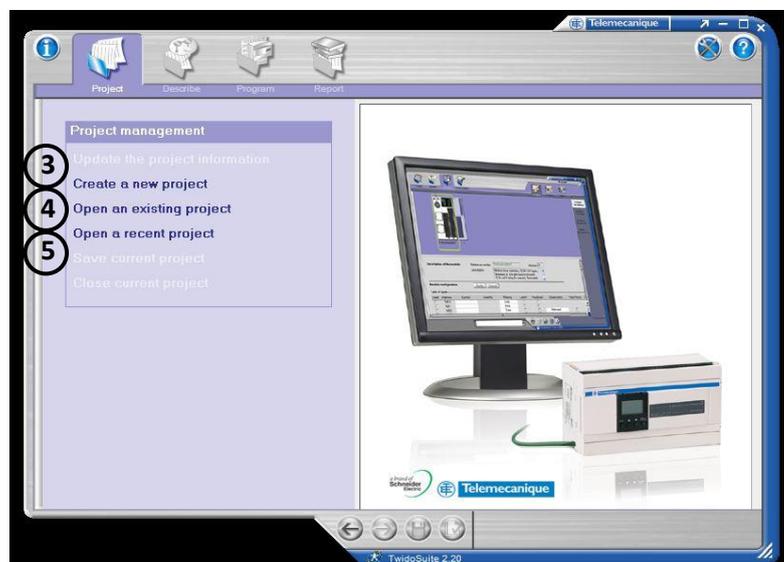


Project



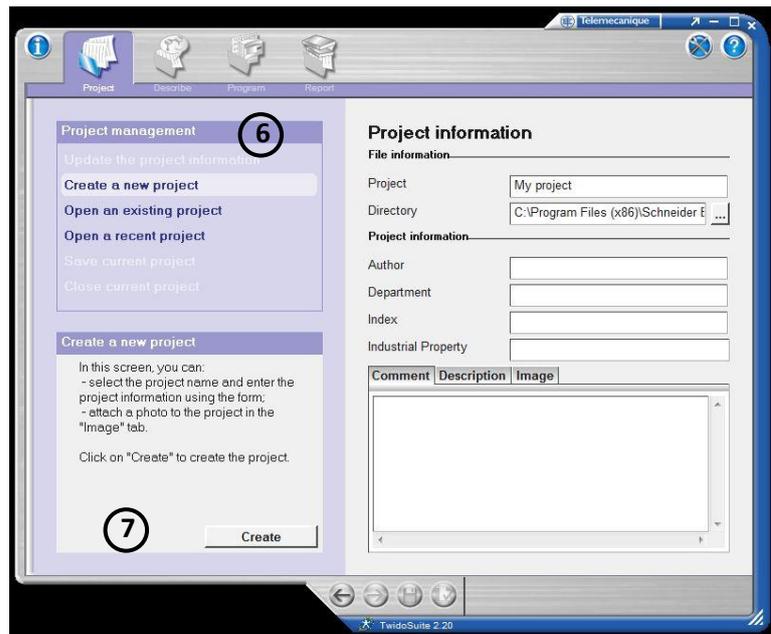
Primeiro passo é criar ou abrir uma aplicação

3. Criar um novo programa;
4. Abrir um programa existente;
5. Abrir um programa recente.



Segundo passo, inserir informações sobre o projeto:

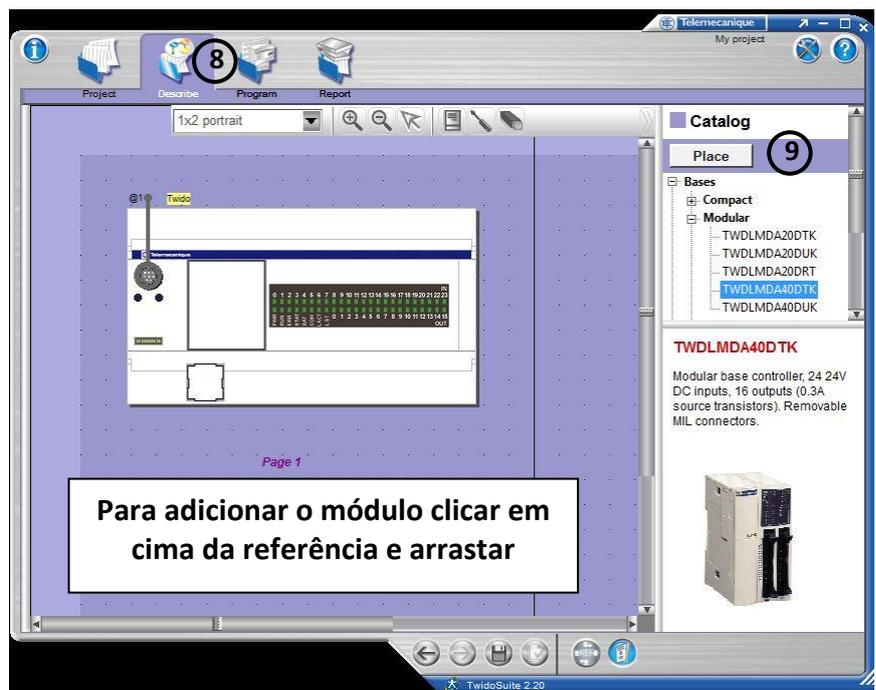
6. Informações sobre o Projeto;
7. Criar o projeto.



Describe 

Segundo passo é configurar o Hardware:

8. Para configurar o Hardware clicar em describe;
9. Adicionar Base, Modulo Expansão, I/O digital e analógico, 2ª porta de comunicação, Relógio, Memória e Módulos de Comunicação.



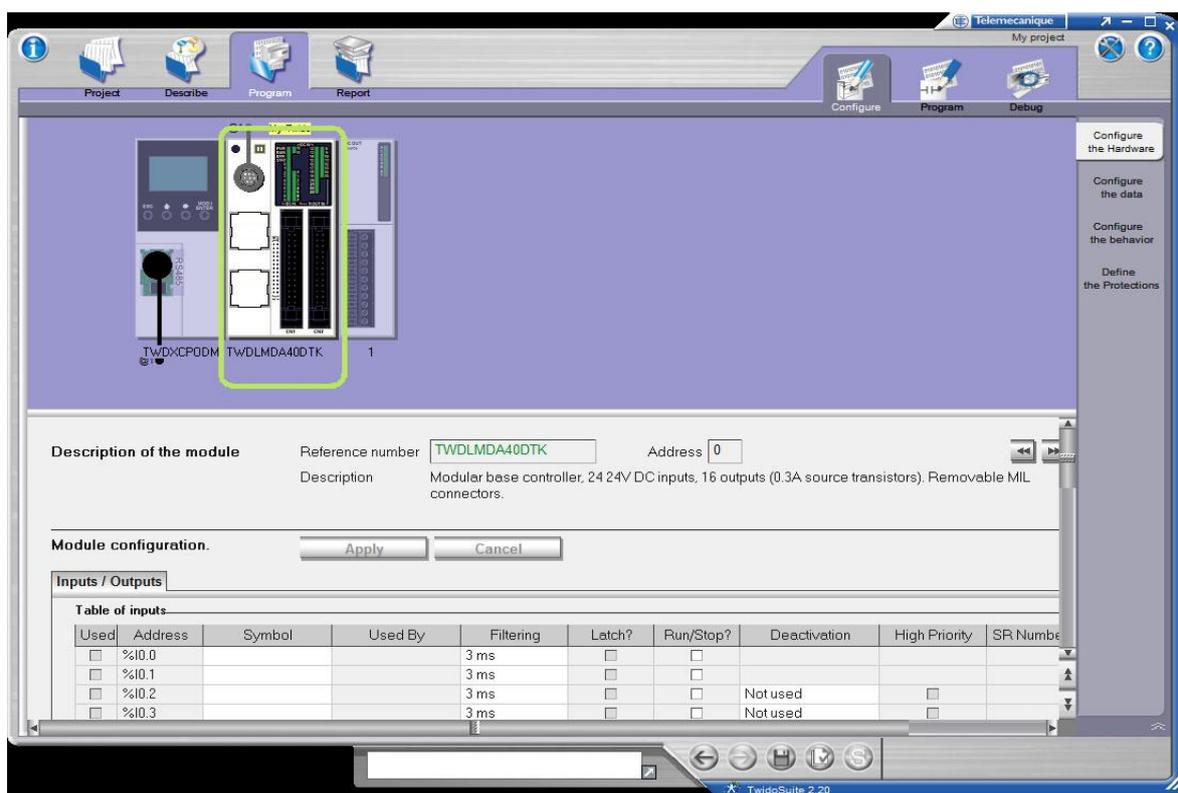
Program

Dentro da pasta Program temos acesso a configuração dos módulos de expansão, blocos de funções, acesso a programação, transferência do programa ao PLC.

Isto tudo está dividido em subpastas, observar no canto superior direito da pagina e na lateral direita.

Configure

Na pasta Configure temos acesso a configuração de entradas digitais, analógicas, bits de memória, bloco de funções e Aplicação.



Configure the Hardware - Configurar o Hardware I/Os

Configure the Data - Configurar variáveis internas (bits, words, ponto flutuante), Bloco de Funções, Blocos de I/Os (Contadores rápidos e geradores de pulsos) e Blocos de funções avançadas (PID)

Configures Behavior - Configuração da Aplicação (Watchdog, Start in Run...)

Define the Protections - Colocar senha na aplicação ou programa

Configuração das Entradas

Symbol - Dar um símbolo/nome para entrada

Used By- Se a entrada já está sendo utilizada no programa por uma instrução ou um bloco de funções e indica estado ou valor da variável;

Filtering - Filtro de entrada;

Deactivation - Habilita a entrada para chamar um evento ou sub-rotina.

Escolha: (falling edge (borda de descida), rising edge (borda de subida) ou both edge (ambos borda de subida e descida)

High Priority- Prioridade da entrada em relação a outro evento ou sub-rotina

SR Number - O numero da sub-rotina que deverá ser acionada pela entrada

Run/Stop - O PLC entra em RUN somente com a entrada no estado 1 (ligado)

1. Configuração das Saídas

Controller Status - indica o estado do controlador. Se está em RUN saída estado 1, se está STOP ou ERRO saída estado 0

2. Configuração de Entradas e Saídas Analógicas

Type - Tipo de entrada analógica depende o cartão pode ser configurada em tensão, corrente ou temperatura.

Scope - Configurar a resolução da entrada analogia

Normal - 0 a 4095

Customizado - alterar o valor da resolução que pode ser -32768 a 32767 ou o valor desejado exemplo: 0 a 100

Celsius -1000 a 5000 x 0,1 °C

Fahrenheit -1480 a 9320x0,1 °F

Observação: não esquecer de configurar as entradas/saídas analógicas, pois as mesmas estão desabilitadas.

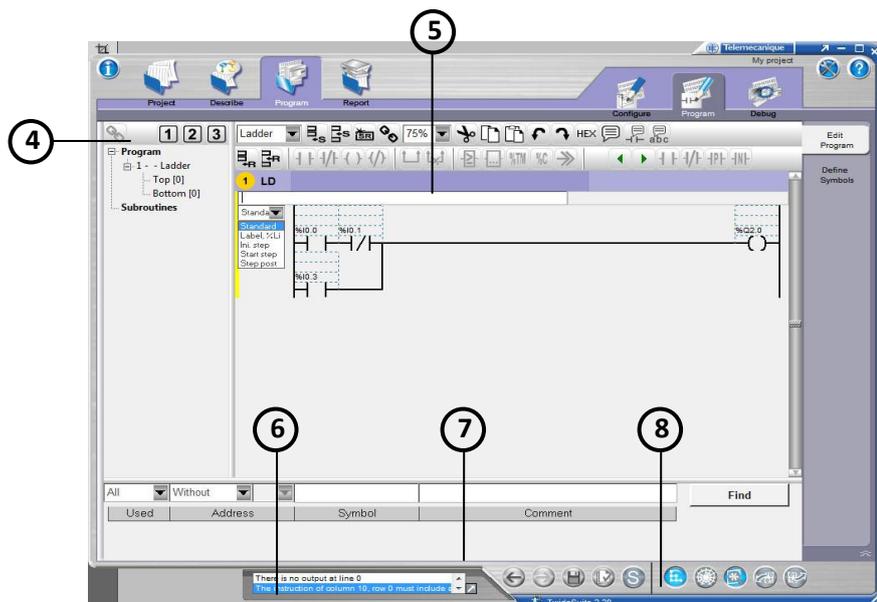
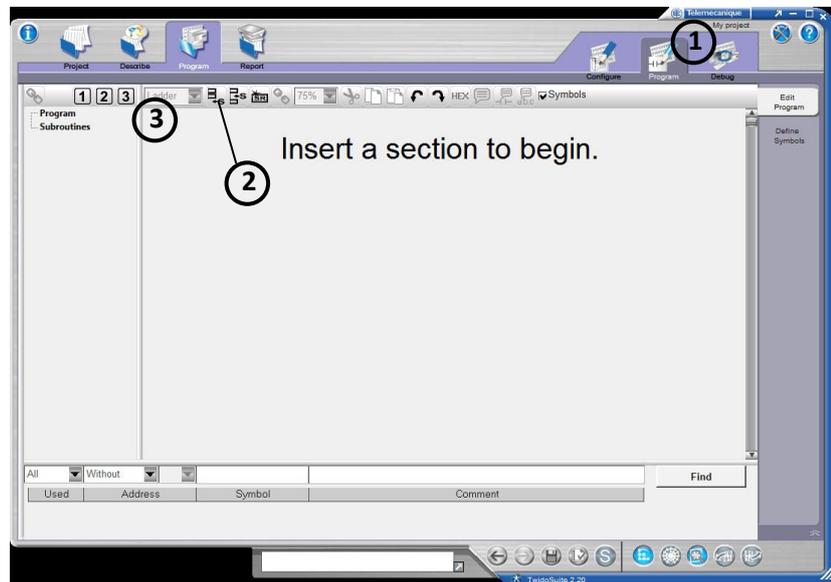
Programação

Dentro da pasta Programa vamos ter acesso à programação com opções de inserir seções, rungs e sub-rotinas onde teremos possibilidade de importar e exportar as mesmas.

Na subpasta (Define Symbols) canto direito, temos acesso a definir símbolos as variáveis

Segue abaixo passo a passo para programação

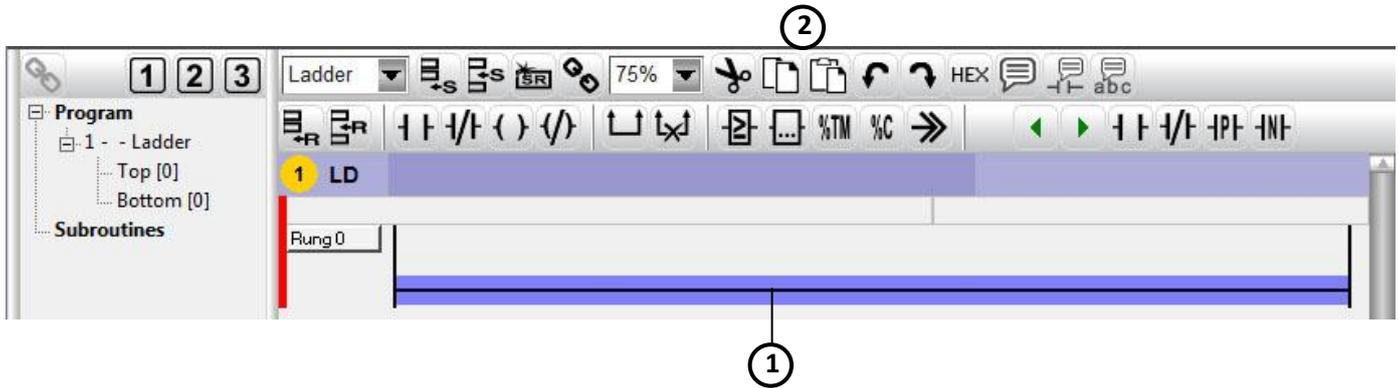
1. Acesso ao Programa
2. Adicionar uma seção
3. Definir a linguagem de programação (Ladder)



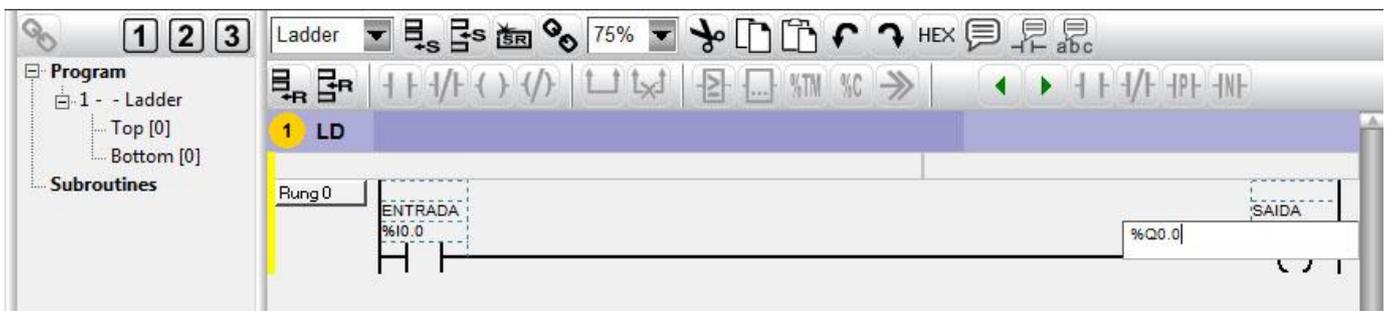
4. Informações do Programa
5. Editar o programa
6. Listas de erros
7. Informações das variáveis contadoras, temporizador...
8. Barra de funções: importar e exportar programa, aparecer/desaparecer, informações sobre programa e variáveis.

Construindo uma linha do programa

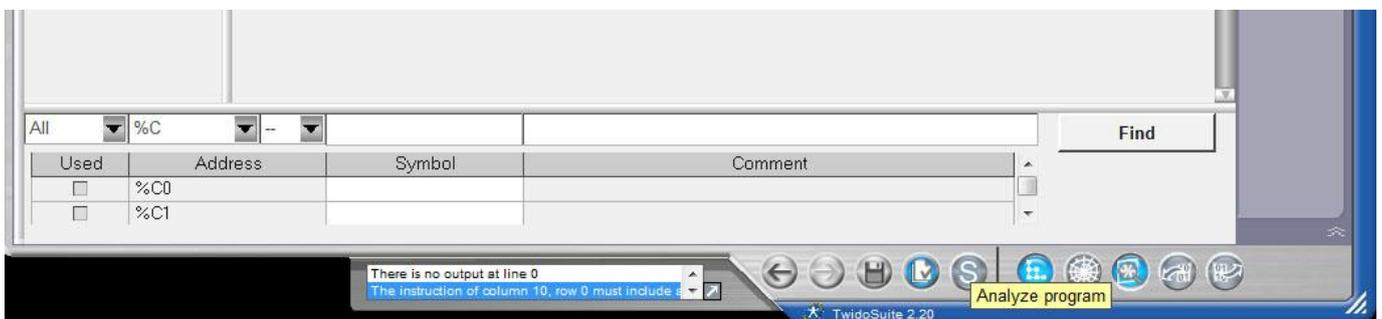
Ao inserir uma seção automaticamente aparece uma rung. Para editar clicar na linha (1) da rung onde será inserido o contato ou bloco de função e fica disponível os botões na barra de instruções Ladder(2),



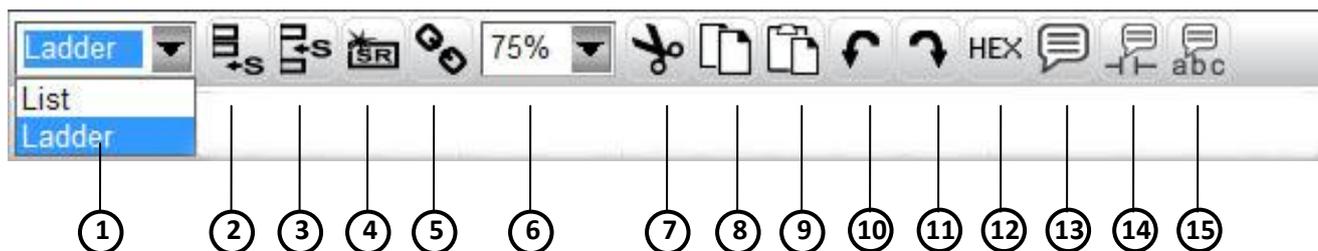
Clicar no botão da instrução desejada, automaticamente aparece à instrução na linha, para atribuir o endereço ao contato duplo click no campo acima do contato e para configurar um bloco de função duplo click no bloco de função.



Na barra de funções, clicar no ícone Analyze  para checar o programa e verificar possíveis erros.

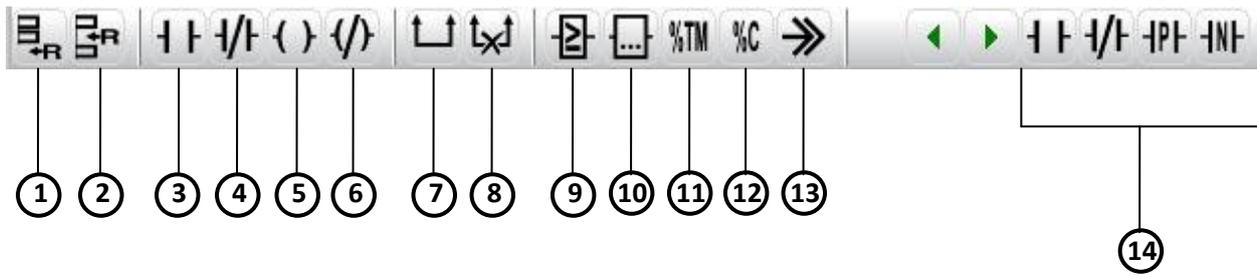


Barra do Programa



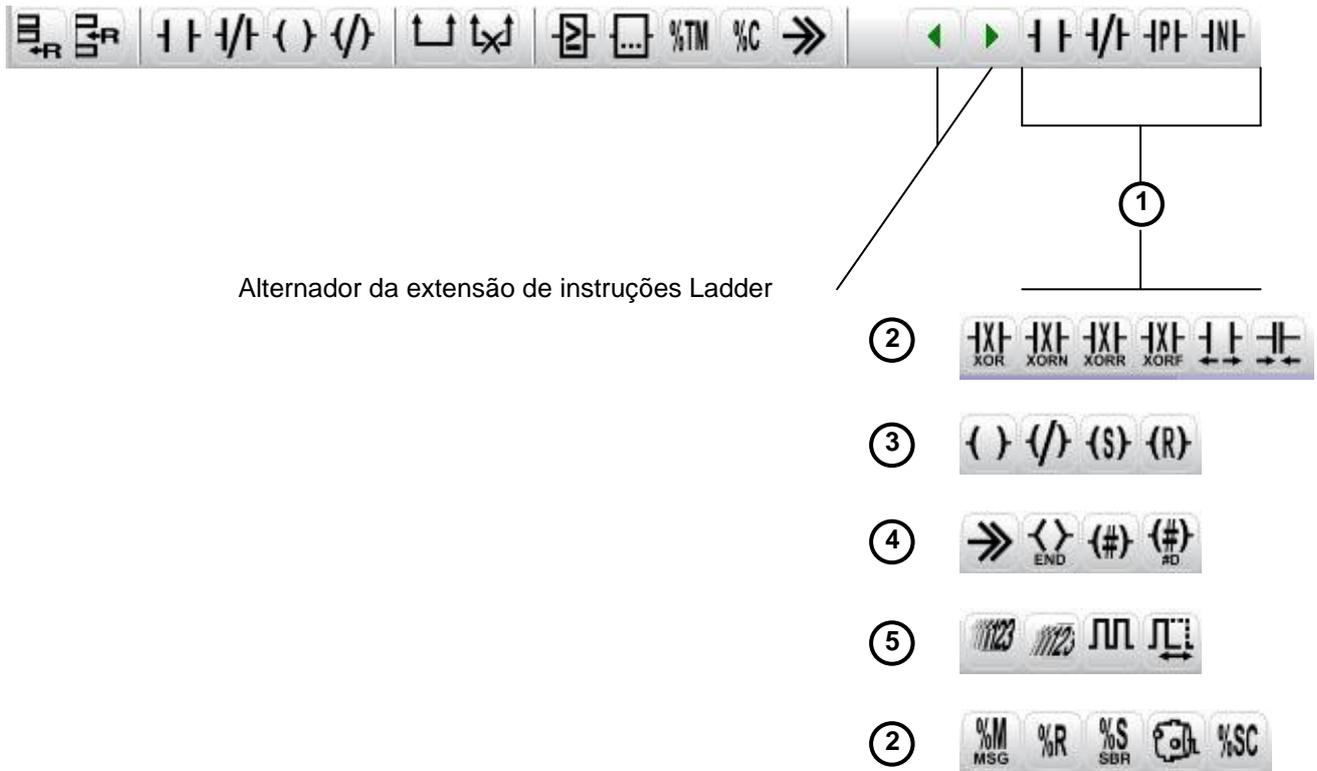
- 1- Escolha Ladder ou Lista
- 2- Adicionar uma seção
- 3- Inserir uma seção entre seções
- 4- Adicionar uma sub-rotina
- 5- Recorta uma seção possibilidade de inserir em outra seção
- 6- Zoom
- 7- Recorta
- 8- Copiar
- 9- Colar
- 10- Retorno
- 11- Avanço
- 12- Display em hexadecimal ou decimal
- 13- Habilita/desabilita comentário nas rungs
- 14- Habilita/desabilita comentários instrução ladder
- 15- Habilita/desabilita símbolos

Barra de Instruções Ladder



1. Adicionar uma rung
2. Inserir uma rung
3. Contato normalmente aberto
4. Contato normalmente fechado
5. Bobina
6. Bobina inversa
7. Link
8. Remover link
9. Bloco de Comparação
10. Bloco Operate
11. Bloco Temporizador
12. Bloco Contador
13. Adicionar um salto ou chamar uma sub rotina
14. Extensão de instruções Ladder

Extensão de instruções Ladder

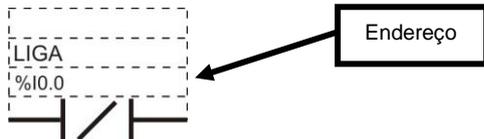


1. Contatos: normal aberto, normal fechado, transição positivo, transição negativo.
2. Contatos especiais: XOR, XORN, XORR, XORF, OPN e SHORT.
3. Bobinas: normal, inversa, set e reset
4. Bobinas especiais: JMP/SR, RET, END, Grafset
5. Blocos de funções básicas; %FC, %VFC, %PLS, %PWM
6. Blocos de funções avançados: %MSG, %R, %SBR, %DR, %SC

Descrição das Funções das Instruções Básicas



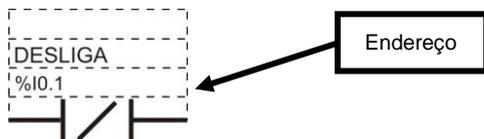
Contato Normal Aberto: Quando o Bit associado (endereçado) à instrução é acionado, a instrução passa de nível lógico = 0 para, nível lógico = 1, habilitando a linha de programação, ou linha lógica. Exemplo:



Quando o dispositivo de entrada enviar um sinal para o PLC, a instrução assume nível lógico = 1.



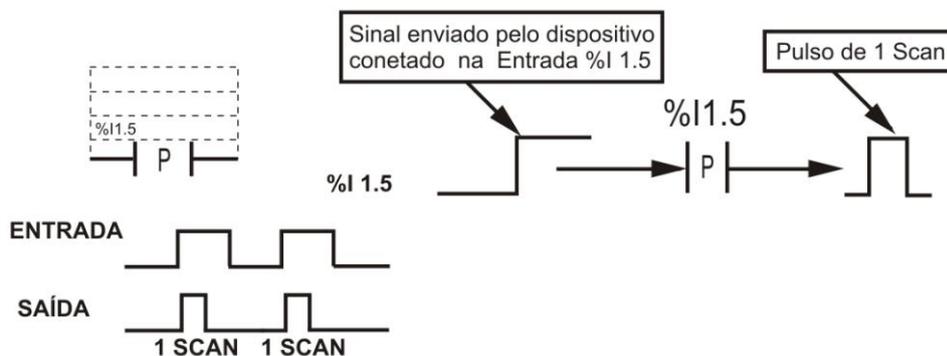
Contato Normal Fechado: Quando o Bit associado (endereçado) à instrução é acionado, a instrução passa de nível lógico = 1, para nível lógico = 0, desabilitando a linha de programação, ou linha lógica. Exemplo:



Quando o dispositivo de entrada enviar um sinal para o PLC, a instrução assume nível lógico = 0.

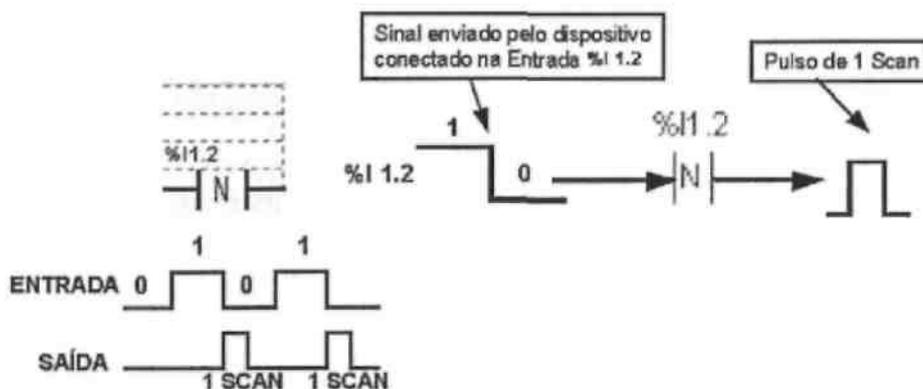


Contato Transicional Tipo P : Esta instrução gera um pulso de 1 scan, quando o seu bit associado passa de nível lógico = 0 para nível lógico=1 (borda de subida). Exemplo:

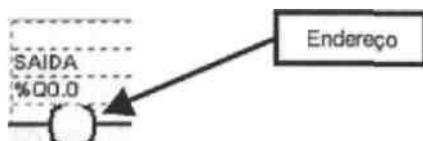




Contato Transicional Tipo N : Esta instrução gera um pulso de 1 scan, quando o seu bit associado passa de nível lógico=1 para nível lógico = 0 (borda de descida) Exemplo:



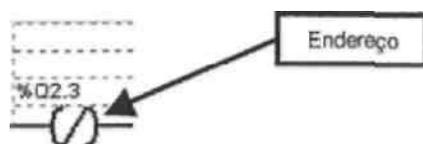
Bobina: Quando todas as condições (instruções) da linha lógica onde está ligada a Bobina estiverem acionadas, o bit relacionado a mesma assume nível lógico=1. E todas as instruções de contato aberto, fechado ou transitórios relacionados a este Bit (com o mesmo endereço) serão habilitados. Exemplo:



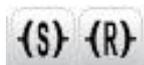
Quando a bobina passar para nível lógico = 1, o dispositivo da saída %Q2.0 será acionado.



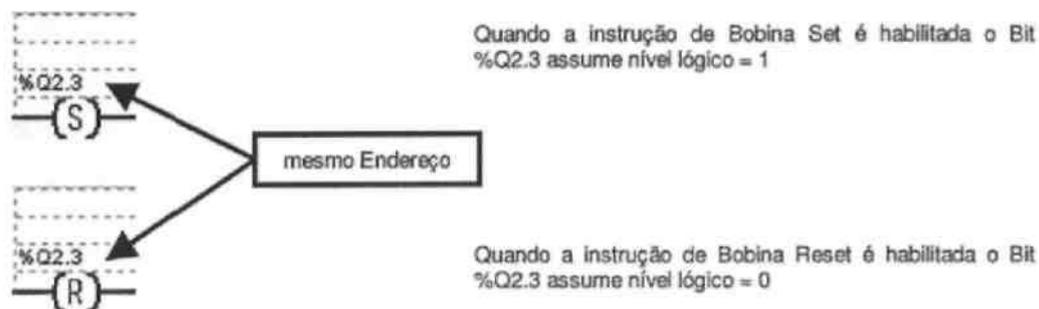
Bobina Inversa: Quando todas as condições (instruções) da linha lógica onde está ligada a bobina estiverem acionadas, o bit relacionado a mesma assume nível lógico = 0. E todas as instruções de contato aberto, fechado ou transitórios relacionados a este Bit (com o mesmo endereço) serão desabilitados. Exemplo:



Quando a bobina passar para nível lógico = 0, o dispositivo da saída %Q2.3 será acionado.

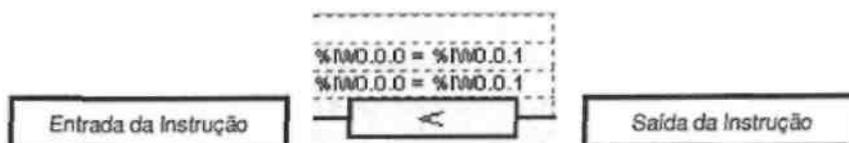


Bobina SET/RESET: Estas bobinas são utilizadas em conjunto, ambas relacionadas a um mesmo Bit (endereço). Quando a linha lógica habilita uma instrução de Bobina Set seu bit associado assume nível lógico = 1, e todas as instruções de contato aberto, fechados ou transitórios relacionados a este Bit (com o mesmo endereço) serão habilitados. Este só passará para nível lógico = 0 quando a linha lógica da Bobina Reset for habilitada, sendo que neste instante a bobina set deve estar desabilitada. Exemplo:



Bloco Comparador

Esta instrução é programada em forma de uma expressão, onde digitamos os valores a serem comparados, e a comparação desejada < (1 menor que 2), > (1 maior que 2), = (1 igual 2), ou <> (1 diferente de 2). Exemplo:





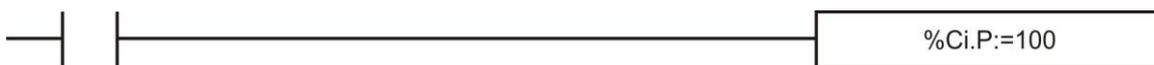
Instrução Operate

A instrução Operate permite ao programador realizar as mais complexas funções em sua lógica de programação, com ele podemos programar desde uma simples operação matemática até uma transferência de dados. Exemplos:

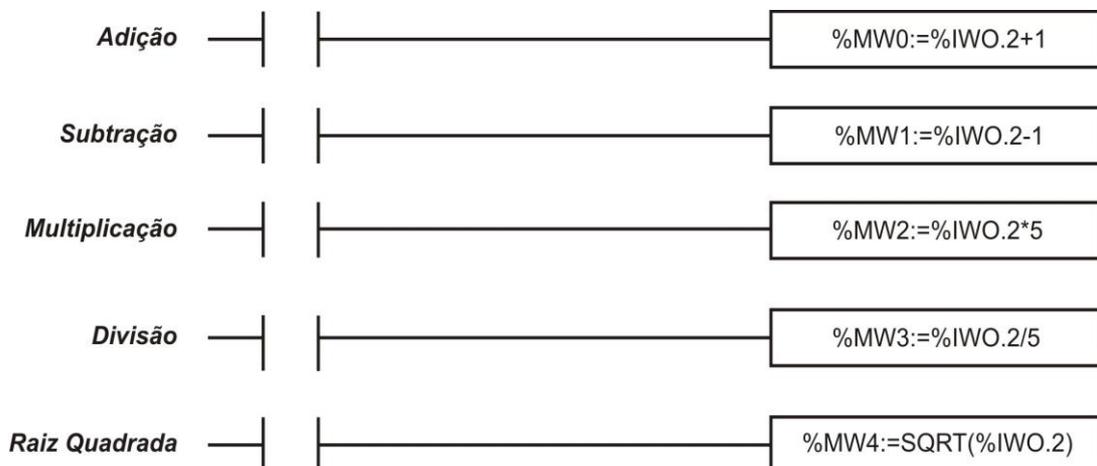
Podemos usar o Operate para alterarmos o valor de contagem do temporizador, este fica armazenado na palavra %T*Mi*.P (i = nome do temporizador).



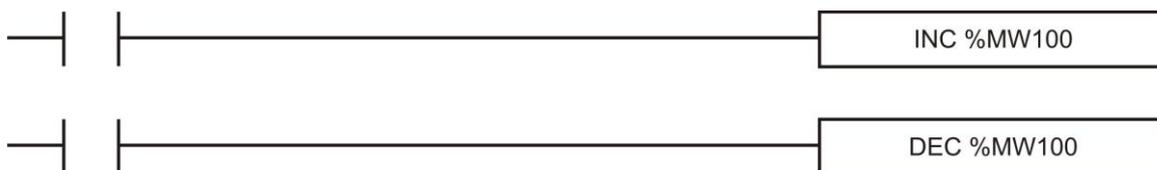
Podemos usar o Operate para alterarmos o valor de contagem do contador, este fica armazenado na palavra %C*i*.P (i = nome do contador).



Podemos usar o Operate para realizarmos operações matemáticas.

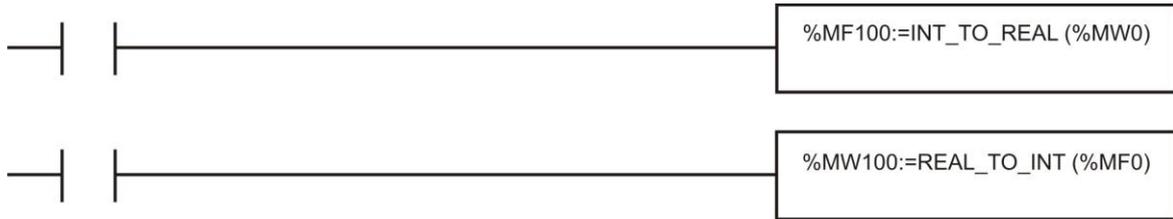


Podemos usar o Operate para efetuar incrementos e decrementos.



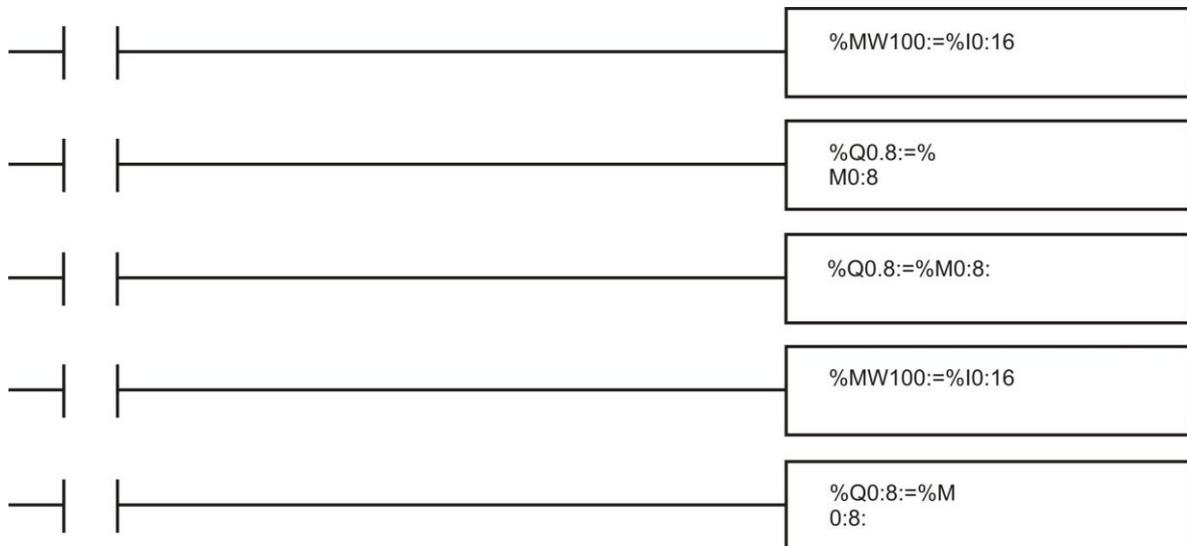
As instruções INC (incremento) e DEC (Decremento), executam suas funções de acordo com o tempo de Scan, por tanto se utilizarmos a função INC em nosso programa como está no exemplo acima, o incremento ocorrerá muito rápido, isto devido a função INC realizar um incremento a cada ciclo de Scan, ou seja, de poucos ms em poucos ms (tempo de duração do Scan).

Podemos usar o Operate para efetuar conversões de valores. (Ex. Inteiro para Real, etc.)



Para realizarmos as conversões, devemos nos atentar quanto aos tipos de variáveis que estamos convertendo e os tipos de variáveis onde iremos salvar a conversão.

Podemos utilizar o Bloco Operate para Indexar um endereço. (Ex. os valores das entradas I0 a I16 serão armazenadas em cada bit da MW100, exemplo o Bit0 da MW100 é igual a entrada I0 (MW100:X0=I0))



Blocos de Funções

Timers (Temporizadores)

O Bloco temporizador é composto pelas seguintes entradas, saídas e variáveis:

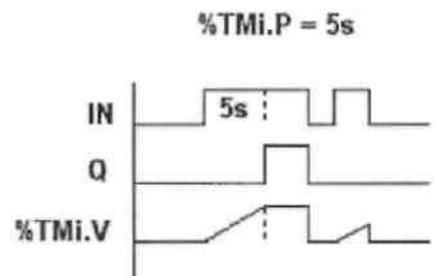
- **IN** - Habilitação do bloco;
- **Q** - Saída que é acionada quando o bloco estiver no valor de preset;
- **%TMI*.V** - Variável onde fica armazenado o valor em processo de temporização (Acumulado);
- **%TMI.P** - Variável onde fica armazenado o valor de preset (Tempo de duração da temporização);
- **%TMI*.Q** - Endereço para deslocamento da saída Q do bloco.

* Substituir o *i* pelo número do Bloco Temporizador utilizado.

Podemos configurar o Bloco Temporizador para trabalhar em 3 Modos:

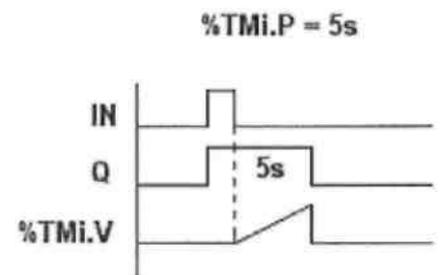
TON

Quando a entrada IN é habilitada o TON conta um tempo, definido pelo programador na configuração do bloco. Após este tempo a saída Q é habilitada e permanece enquanto IN estiver habilitada. Quando desabilitamos a entrada IN o temporizador é ressetado independente do valor de contagem.



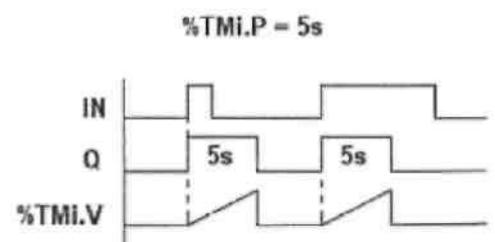
TOF

Quando a entrada IN é habilitada, automaticamente a saída Q também é. Quando IN for desabilitada o TOF conta um tempo definido pelo programador na configuração do bloco e, após este tempo, a saída Q é desabilitada.



TP

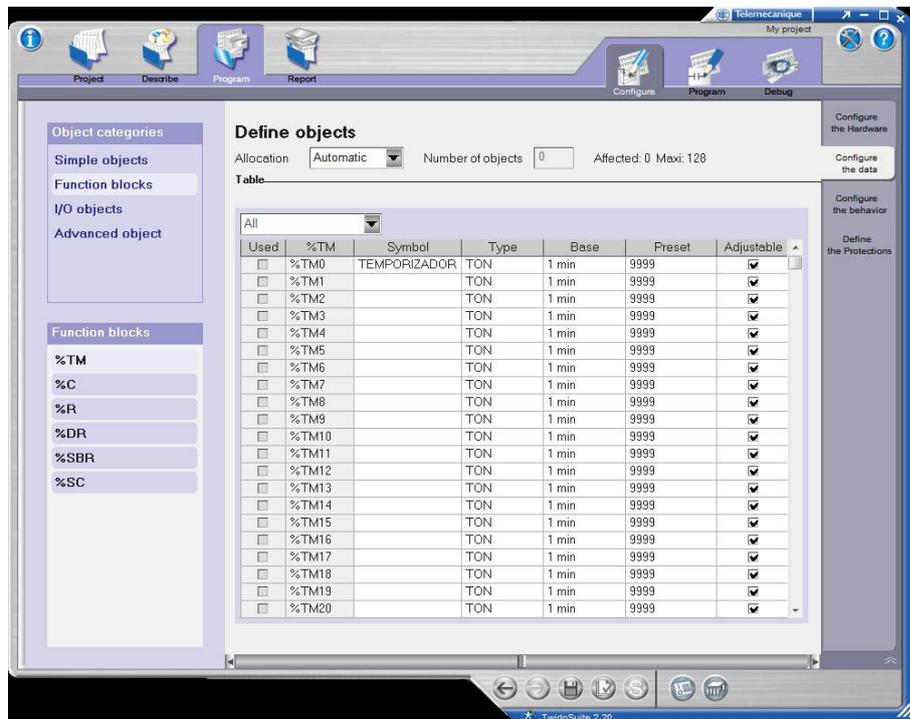
Quando a entrada IN é habilitada, a saída Q gera um pulso de duração configurada pelo programador.



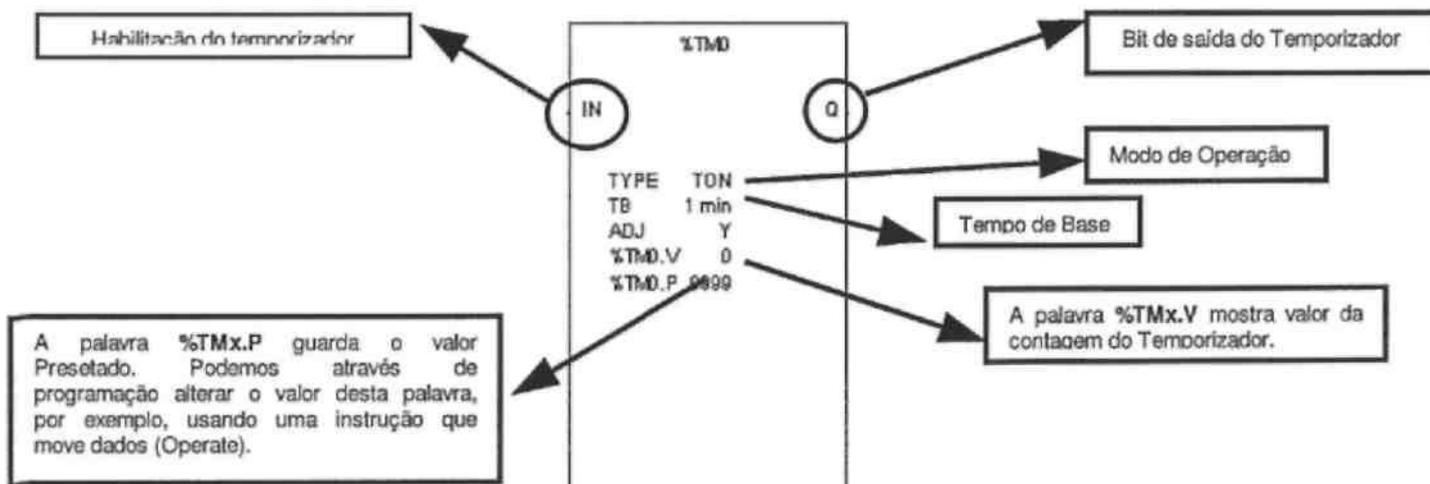
Parametrização do Timer

Para parametrizar o bloco temporizador dar duplo click no bloco, visualiza a janela de configuração onde é possível parametrizar qualquer bloco de função.

1. **Type:** seleccionar o modo de operação
2. **Base:** seleccionar a base de tempo
3. **Preset:** definir o valor do tempo, este valor será multiplicado pela base de tempo,
Ex: 1min (base) x 5 (preset) = 5 minutos
4. **Adjustable:** habilita a alterar o valor do preset.



Veja abaixo o Timer (Temporizador):



Counters (Contadores) %C

O Bloco Contador é composto pelas seguintes entradas, saídas e variáveis:

R - Entrada **Reset**, para zerar o bloco;

S - Entrada **Set**, para setar o valor do preset, ou seja, saltar de onde ele estiver para o valor de preset;

CU - Entrada **Counter up**, é a entrada que faz a contagem crescente;

CD - Entrada **Counter Down**, é a entrada que faz a contagem decrescente;

F - Saída **Full**, é a saída que será acionada quando o bloco estiver estourado, passado de 9999 para 0;

D - Saída **Done**, é a saída que será acionada quando o bloco tiver chegado no valor presetado;

E - Saída **Empty**, é a saída que será acionada quando o bloco estiver estourado, passado de 0 para 9999;

%Ci.V - Variável onde fica armazenado o valor acumulado do bloco;

%Ci.P - Variável onde fica armazenado o valor de preset (Valor a ser contado);

%CLF - Endereço para deslocamento da saída F do bloco;

%Ci.D - Endereço para deslocamento da saída D do bloco;

%Ci.E - Endereço para deslocamento da saída E do bloco.

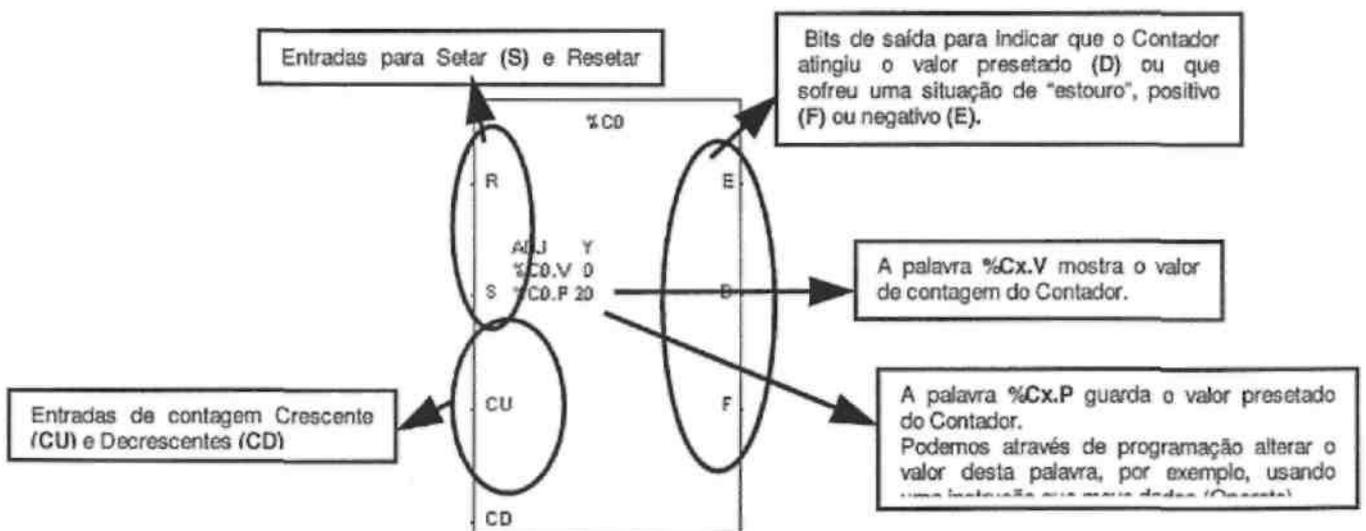
Este bloco é usado para realizar contagens crescentes e decrescentes.

Temos 2 entradas de contagem, uma que executa contagem crescente (**CU**) e outra decrescente (**CD**). Ao habilitarmos tais entradas estas vão incrementar ou decrementar o Valor de Contagem, e quando este for exatamente igual ao valor Presetado na configuração do bloco que atua a saída **D (Done)**.

Quando o valor de contagem ultrapassar 9999, este volta a 0 e aciona a saída **F (Full)**,

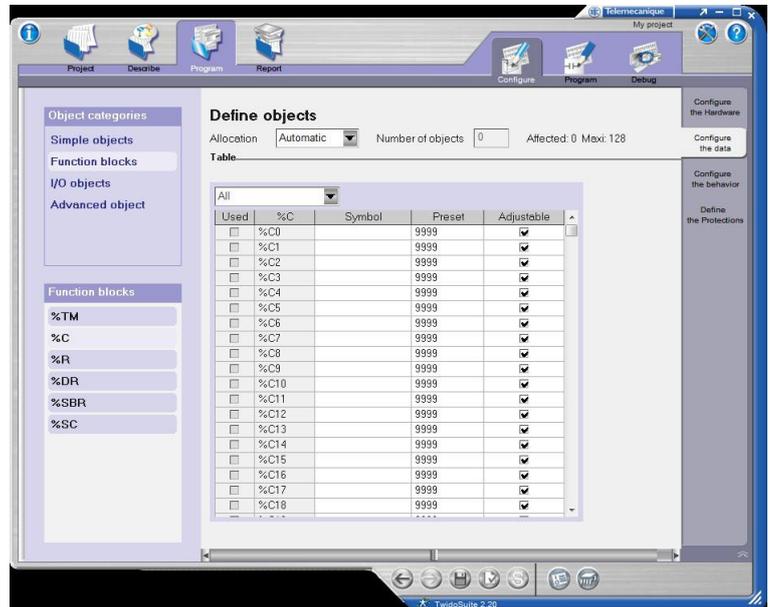
quando o valor de contagem estiver em 0 e sofrer um decremento, este passa para 9999 acionando a saída **E (Empty)**.

Para ressetar o valor de Contagem do contador temos de habilitar a entrada **R (Reset)**, por outro lado se quisermos que o valor de contagem seja igual ao valor Presetado, habilitamos a entrada **P (Preset)**.



Parametrização do Contador

1. **Preset:** inserir o valor a contar
2. **Adjustable:** habilita a alterar o valor do preset

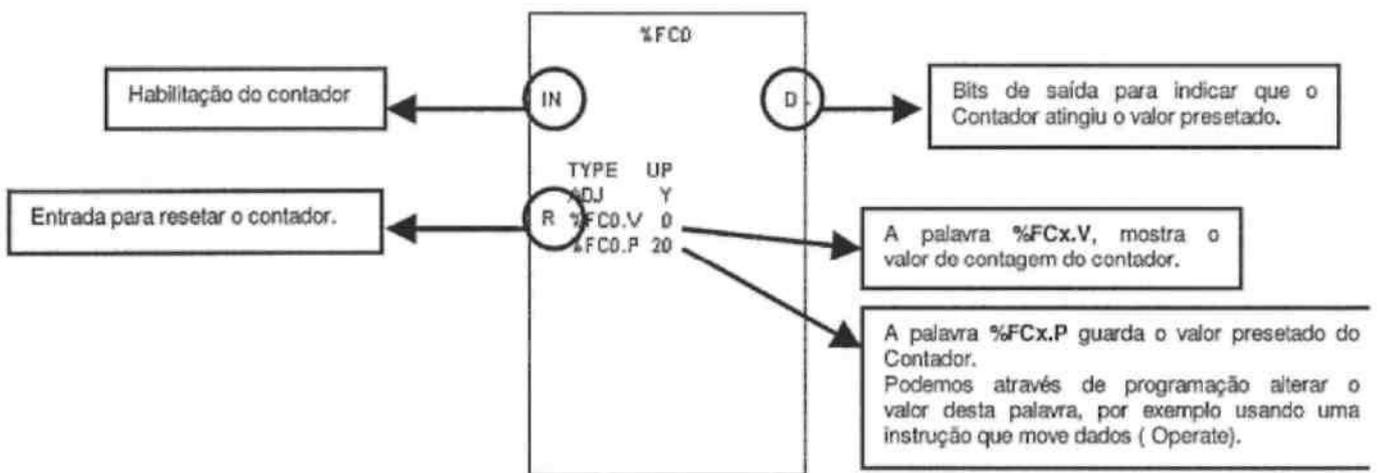


Fast Counter



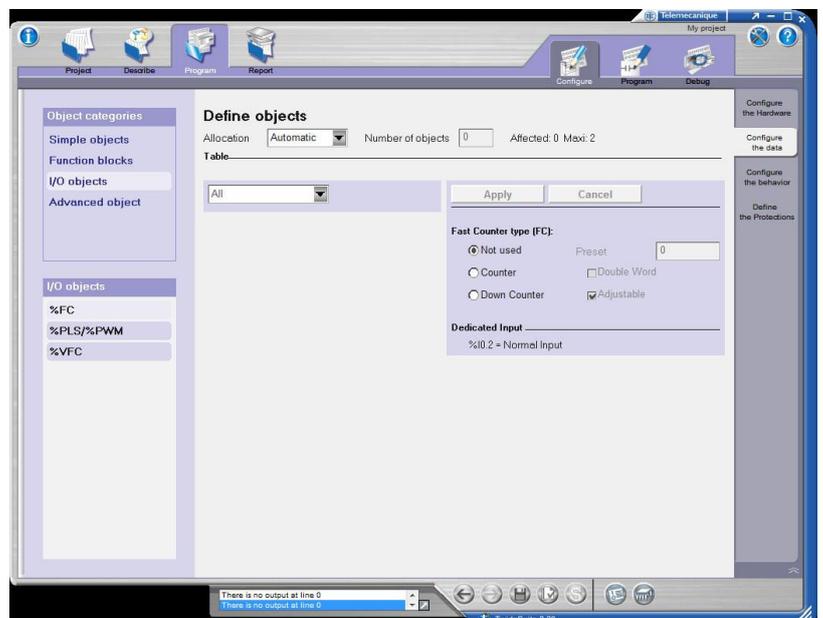
Este bloco é utilizado para realizar contagens rápidas (5 kHz) crescentes ou decrescentes. Para cada Bloco Fast Counter temos uma entrada dedicada onde, ao habilitarmos o mesmo, este é que fará a contagem. Esta entrada dedicada já vem declarada ao inserirmos o contador rápido, portanto não necessitamos endereçar a mesma, mas necessitaremos verificar a entrada declarada para o bloco que irei utilizar, para que possamos fazer a conexão física do dispositivo de campo. Para que o contador inicie a contagem, necessitamos habilitá-lo através da entrada IN, e para resetar temos que habilitar a entrada **R (Reset)**.

Este contador rápido possui apenas a saída **D (Done)**, que é habilitada sempre que contador chegar ao valor presetado.



Parametrização do Fast Counter

1. **Fast Counter Type:** selecionar o tipo de contador rápido, crescente ou decrescente
2. **Dedicated Input:** endereço da entrada dedicada para este bloco
3. **Preset:** inserir o valor de contagem



Very Fast Counter



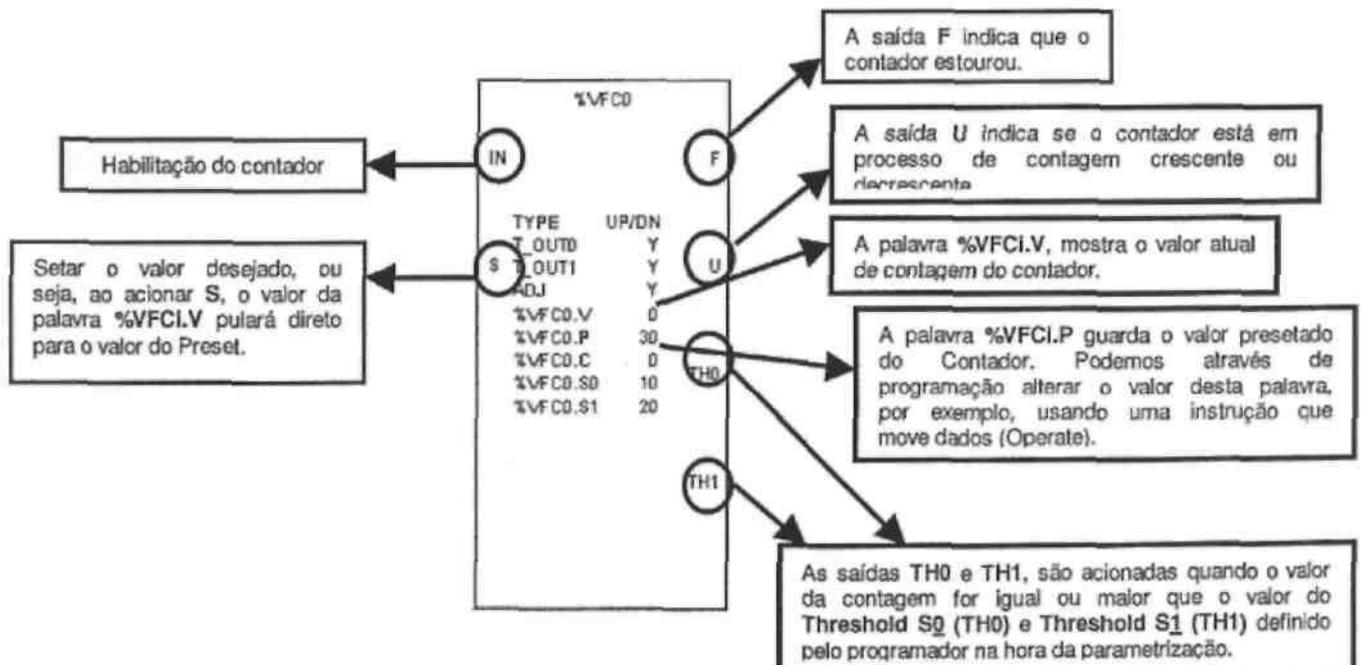
Este bloco é utilizado para realizar contagens muito rápidas (até 20 kHz) nos modos **crescente e decrescente, crescente, decrescente e também como medidor de frequência.**

Para cada Bloco Very Fast Counter configurado como contador, temos **entradas dedicadas**, onde ao selecionarmos o tipo de contagem que o mesmo irá realizar, ele automaticamente nos informará quais entradas ele utilizará e para que servirá estas entradas. Os Blocos Very Fast Counter disponibilizam alguns recursos mais além dos tradicionais:

- **Threshold S0 e Threshold S1**, estes dois itens funcionam como preset auxiliares, pois no Very Fast Counter você possui duas saídas (**TH0 e TH1**) que são acionadas de acordo com o valor estipulado para cada **Threshold** pelo programador na hora da parametrização do bloco.
- **Reflex Outputs**, este item funciona da seguinte forma, na hora da parametrização no item Reflex Outputs, podemos programar duas saídas para que elas sejam acionadas de acordo com o valor dos **Thresholds**.

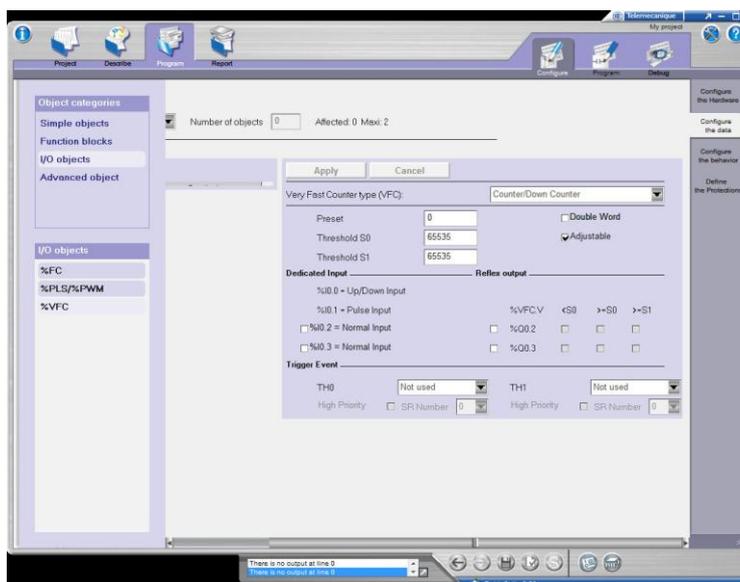
Importante: para que as saídas reflex sejam acionadas é preciso habilitá-las:

- %Q0.2 - %VFC0.R tem que estar em nível lógico 1 para que essa saída seja acionada conforme configuração.
- %Q0.3 - %VFC0.S tem que estar em nível lógico 1 para que essa saída seja acionada conforme configuração.



1. Parametrização do Very Fast Counter (%VFC0) como contador crescente e decrescente (Counter/Down Counter)

1. **Very fast counter type (VFC):** selecionar o tipo de contagem
2. **Threshold:** adicionar o valor desejado para os Threshold SO e S1
3. **Dedicated Inputs:** entradas dedicadas e suas funções
4. **Reflex Outputs:** habilita as saídas reflex de acordo com os thresholds
5. **Trigger event:** habilita as interrupções a serem executadas caso o valor dos thresholds sejam alcançados



Entradas Dedicadas (Dedicated Inputs):

%I0.0 - Up/Down Input = Entrada que indica a direção da contagem. Em 1 temos contagem crescente e em 0 contagem decrescente.

%I0.1 - Pulse Input = Entrada de contagem dos pulsos.

%I0.2 - Preset Input ■ Entrada opcional. Para contagem crescente, quando ativada, carrega 0 no valor atual de contagem %VFC0.V. Para contagem decrescente, carrega o valor do preset %VFC0.P no valor atual de contagem %VFC0.V.

%I0.3 - Catch Input = Entrada opcional que quando ativada carrega o valor atual de contagem contido em %VFC0.V na palavra %VFC0.C.

Saídas Reflexas

%Q0.2 - Reflex Output 0 = Saída opcional que pode ser configurada conforme o valor dos Thresholds SO e S1. Além da configuração no bloco, o bit %VFC0.R tem que estar em nível lógico 1 para que essa saída seja acionada conforme configuração.

%Q0.3 - Reflex Output 1 ■ Saída opcional que pode ser configurada conforme o valor dos Thresholds SO e SI. Além da configuração no bloco, o bit %VFC0.S tem que estar em nível lógico 1 para que essa saída seja acionada conforme configuração.

Essas saídas podem ser configuradas para as seguintes condições:

<S0 = Aciona se o valor atual da contagem %VFC0.V for menor que o valor do Threshold SO. >S0 = Aciona se o valor atual da contagem %VFC0.V for maior que o valor do Threshold SO. >S1 = Aciona se o valor atual da contagem %VFC0.V for maior que o valor do Threshold SI.

Obs: as três condições podem ser selecionadas e as saídas só serão acionadas se os bits %VFC0.R e %VFC0.S estiverem setados (nível lógico 1)

Trigger Event

Essa função habilita eventos que são iniciados quando os valores dos Threshold SO (TH0) e Threshold SI (TH1) são alcançados. Ao alcançar esses valores, uma chamada de subrotina é habilitada.

Configuração do Evento:

É possível selecionar como o evento vai ser acionado dependendo da troca de estado das saídas TH0 e TH1 do bloco.

Falling edge = Borda de descida

Rising edge = Borda de subida

Both edges = Ambas as condições

Após configurar como o evento será acionado, configuramos o número da subrotina a ser iniciada e se ela será prioritária.

2. Parametrização do Very Fast Counter (%VFC0) como contador crescente e decrescente utilizando as vias A e B do Encoder (Counter/Down Counter bi-phase)

Entradas Dedicadas (Dedicated Inputs):

%I0.0 - Pulse Phase B = Entrada de contagem da Via B do Encoder.

%I0.1 - Pulse Phase A = Entrada de contagem da Via A do Encoder.

%I0.2 - Preset Input = Entrada opcional. Para contagem crescente, quando ativada, carrega 0 no valor atual de contagem %VFC0.V. Para contagem decrescente, carrega o valor do preset %VFC0.P no valor atual de contagem %VFC0.V.

%I0.3 - Catch Input = Entrada opcional que quando ativada carrega o valor atual de contagem contido em %VFC0.V na palavra %VFC0.C.

Obs: As configurações de Saídas Reflexas e Trigger Event são iguais às apresentadas Up/Down Counter.

Parametrização do Very Fast Counter (%VFC0) como contador crescente (Single Up Counter)

Entradas Dedicadas (Dedicated Inputs):

%I0.0 - Normal Input = Entrada não utilizada

%I0.1 - Pulse Input = Entrada de contagem.

%I0.2 - Preset Input = Entrada opcional. Quando ativada, carrega 0 no valor atual de contagem %VFC0.V.

%I0.3 - Catch Input = Entrada opcional que quando ativada carrega o valor atual de contagem contido em %VFC0.V na palavra %VFC0.C.

Obs: As configurações de Saídas Reflexas e Trigger Event são iguais às apresentadas Up/Down Counter.

3. Parametrização do Very Fast Counter (%VFC0) como contador decrescente (Single Down Counter)

Entradas Dedicadas (Dedicated Inputs):

%I0.0 - Normal Input = Entrada não utilizada

%I0.1 - Pulse Input = Entrada de contagem.

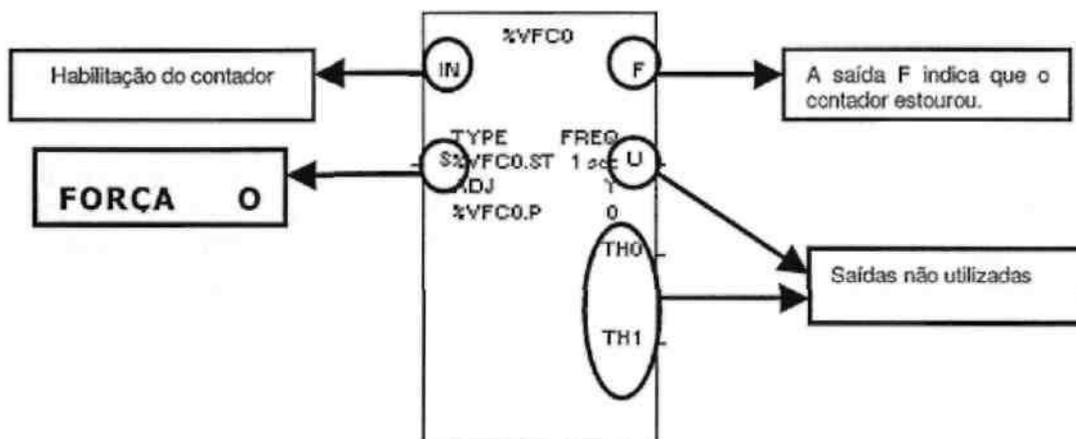
%I0.2 - Preset Input = Entrada opcional. Quando ativada, carrega o valor do preset %VFC0.P no valor atual de contagem %VFC0.V.

%I0.3 - Catch Input = Entrada opcional que quando ativada carrega o valor atual de contagem contido em %VFC0.V na palavra %VFC0.C.

Obs: As configurações de Saídas Reflexas e Trigger Event são iguais às apresentadas Up/Down Counter.

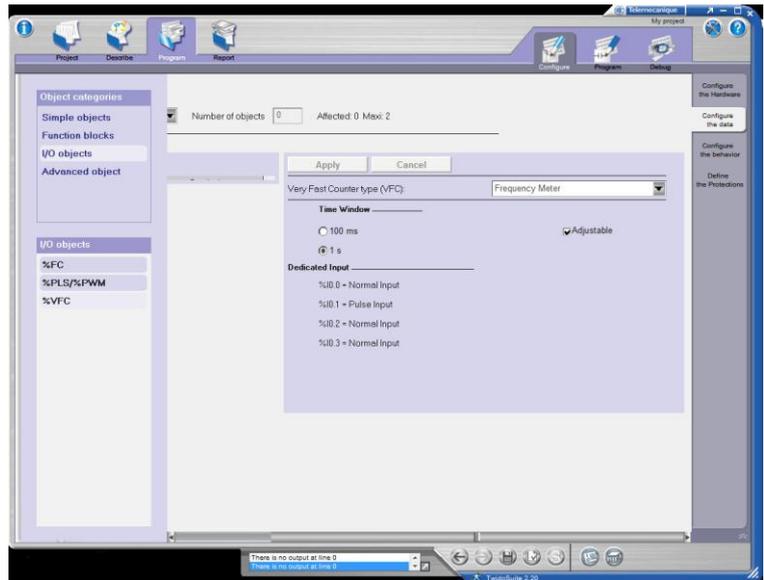
4. Very Fast Counter como Medidor de Frequência (Frequency Meter)

Utilizado para medições de frequências de 10Hz a 20KHz. Duas bases de tempo podem ser selecionadas na parametrização ou alteradas durante o programa (via Bloco Operate) modificando a palavra %VFCi.T. As bases de tempo são 100ms para medidas de 100Hz a 20kHz ou 1s para medidas de 10Hz a 20kHz.



Parametrização do Very Fast Counter (%VFC0) como Medidor de Frequência (Frequency Meter)

1. **Time Window:** configurar a base de tempo
2. **Dedicated Inputs:** entradas dedicadas e suas funções



Entradas Dedicadas (Dedicated Inputs):

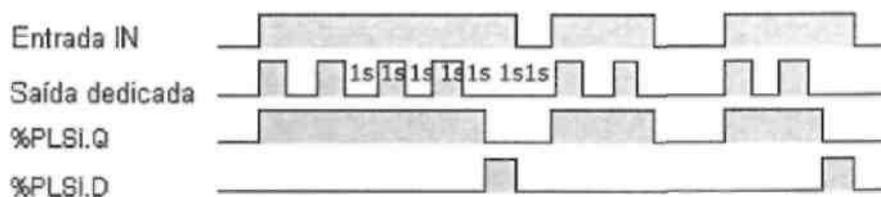
- %I0.0 - Normal Input = Entrada não utilizada.
- %I0.1 - Pulse Input = Entrada de contagem.
- %I0.2 - Normal Input = Entrada não utilizada.
- %I0.3 - Normal Input = Entrada não utilizada.

PLS (Gerador de Pulso)

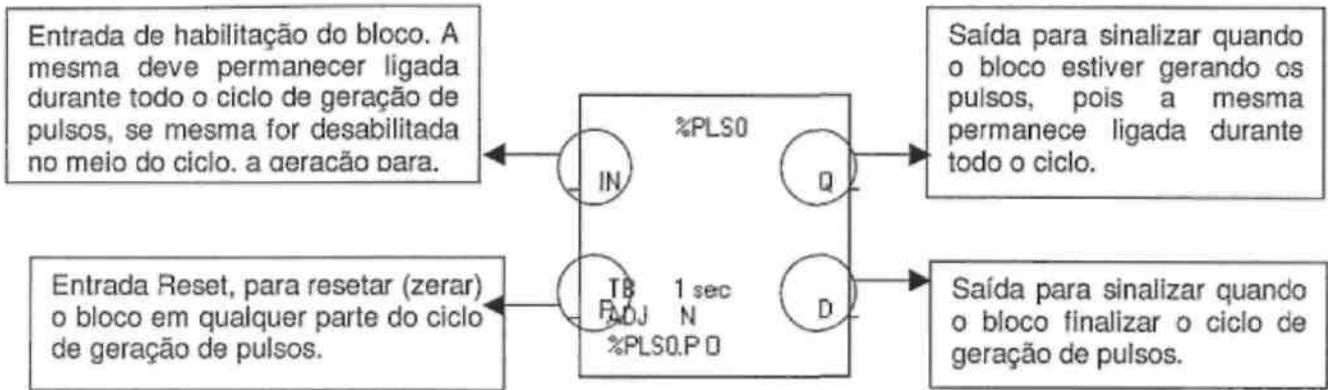
Este bloco é utilizado para gerar pulsos nas saídas dedicadas de acordo com a parametrização realizada.

Ao alimentar a entrada IN, automaticamente a saída dedicada é acionada durante 50% do tempo parametrizado do pulso (Preset x Base de tempo), e após ter passado os 50% do tempo, a saída é desabilitada e permanece assim durante os outros 50% do tempo parametrizado. O quantidade de pulsos é parametrizado através da variável %PLSi.N

Ex. Se parametrizarmos o bloco para gerar 4 pulsos, tempo do pulso de 2 seg. o bloco irá gerar os pulsos 1s ligado e 1s desligado, da seguinte forma.

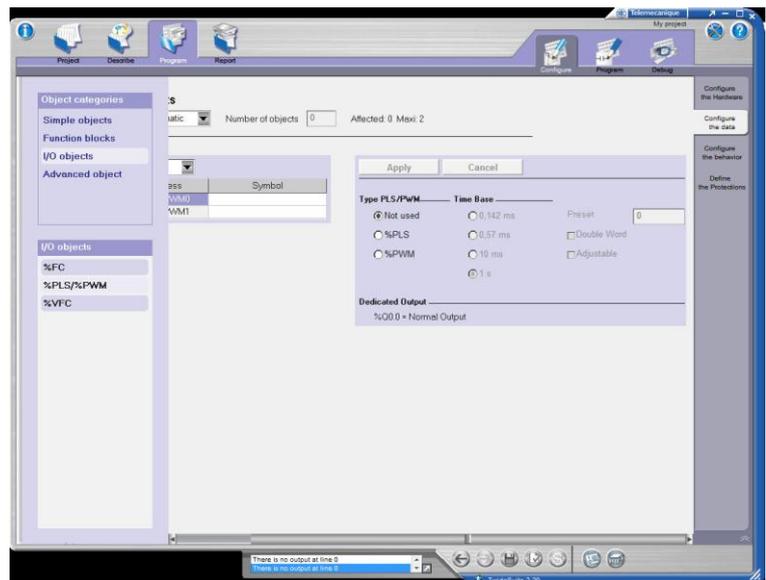


PLSi.Q - Variável que informa o estado do gerador, %PLSi.Q=1 está gerando pulsos
 PLSi.D - Variável que indica fim do ciclo, %PLSi.D = 1 atingiu o numero de pulsos conforme a variável %PLSi.N



Parametrização do bloco PLS

1. **Type PLS/PWM:** seleção do bloco que você irá utilizar
2. **Time base:** base de tempo
3. **Preset:** inserir o tempo de duração de cada pulso
Ex: 10 (preset) x 1s (time base) = 10 segundos
4. **Dedicated Output:** saída dedicada que serão gerados os pulsos



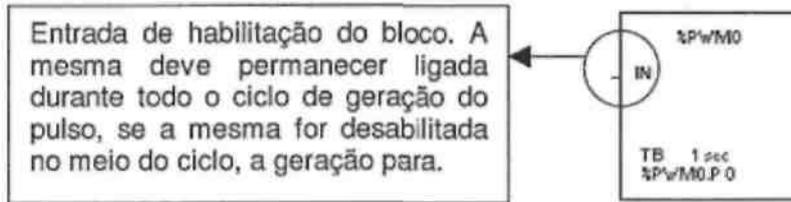
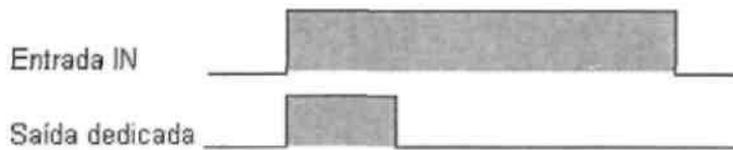
Obs.: Para parametrizar a quantidade de pulsos, deve-se inserir a quantidade de pulsos desejável na variável %PLSi.N

PWM (Gerador de Pulso)



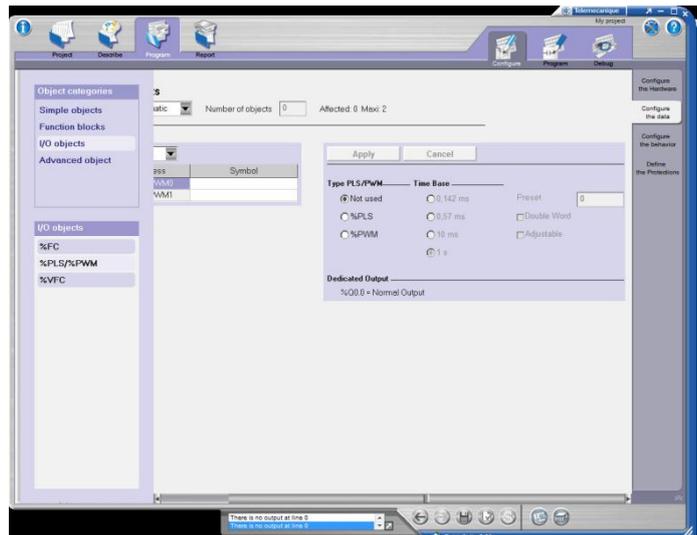
Este bloco é utilizado para gerar um pulso na saída dedicada de acordo com a parametrização realizada.

Ao alimentar a entrada IN, automaticamente a saída dedicada é acionada durante a porcentagem do tempo parametrizado (Preset) que deve ser definida na variável %PwMi.R, e após ter passado a porcentagem do tempo, a saída é desabilitada e permanece assim durante o restante do tempo parametrizado (Preset). Ex. Se parametrizarmos o bloco para gerar o pulso com o tempo de 10 seg., porém a porcentagem do tempo definido é de 20% o bloco irá gerar o pulso da seguinte forma.



Parametrização do PWM

1. **Type PLS/PWM:** seleção do bloco que você irá utilizar
2. **Time base:** base de tempo
3. **Preset:** inserir o tempo de duração de cada pulso
Ex: 10 (preset) x 1s (time base) = 10 segundos
4. **Dedicated Output:** saída dedicada que serão gerados os pulsos



Bloco Drum



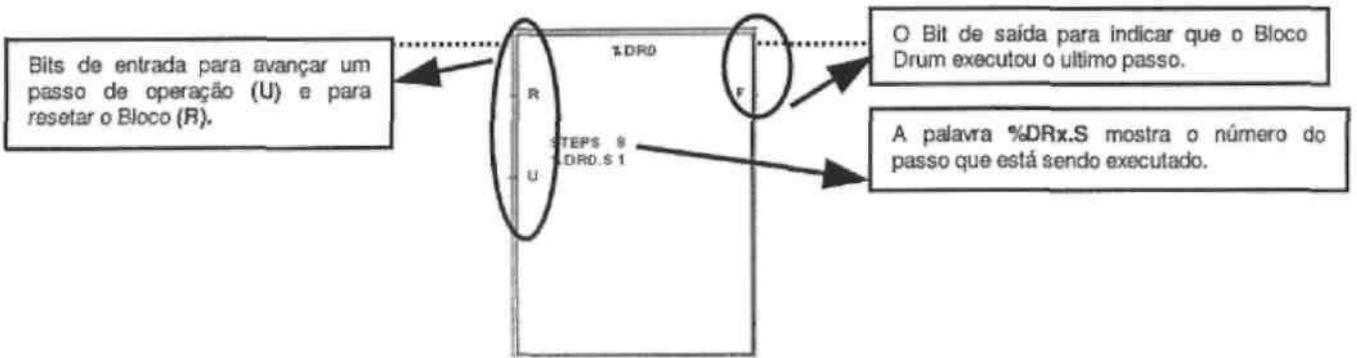
Este bloco executa ações passo-a-passo (Seqüencial), pode trabalhar com 08 passos de operação controlando até 16 dispositivos de saída. Este bloco é, na verdade, o equivalente em programação de um carne mecânico. O Bloco Drum possui as seguintes entradas e saídas:

- R - Entrada **Reset**, para zerar o bloco;
- U - Entrada **Up**, para passar de um passo para o outro;
- F - Saída **Full**, é acionada quando o bloco chega no ultimo passo;

- %DRi.P - Variável onde fica armazenado o valor de preset;
- %DRi.S - Variável onde fica armazenado o passo em que o bloco se encontra;
- %DRi.F - Endereço para deslocamento da saída F do bloco.

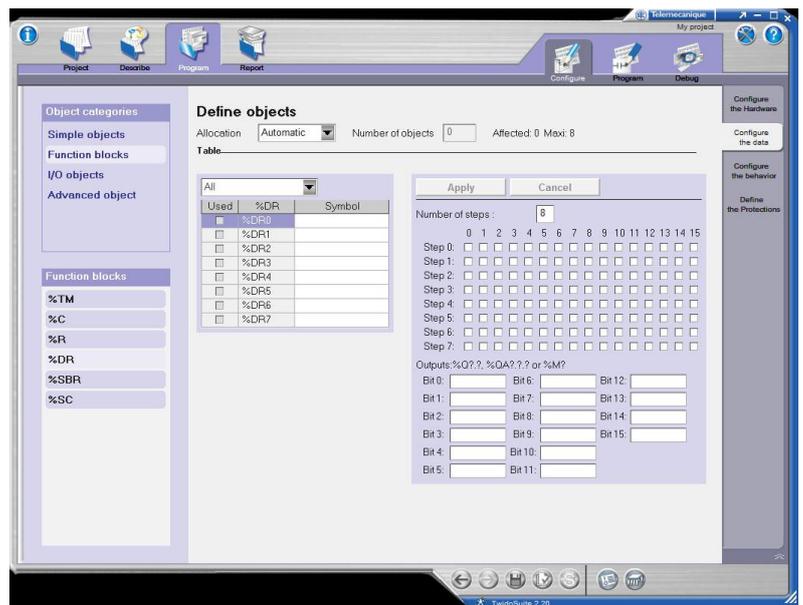
A cada vez que a entrada U (Up) é habilitada, o bloco executa um passo de operação (máximo 08 passos), dentro deste passo os bits de saídas configurados serão habilitados (máximo 16).

A palavra **% DRx.S** exibe o número do passo que está sendo executado. Quando todos os passos configurados forem executados, a saída **F (Full)** será habilitada.



Parametrização do Bloco Drum:

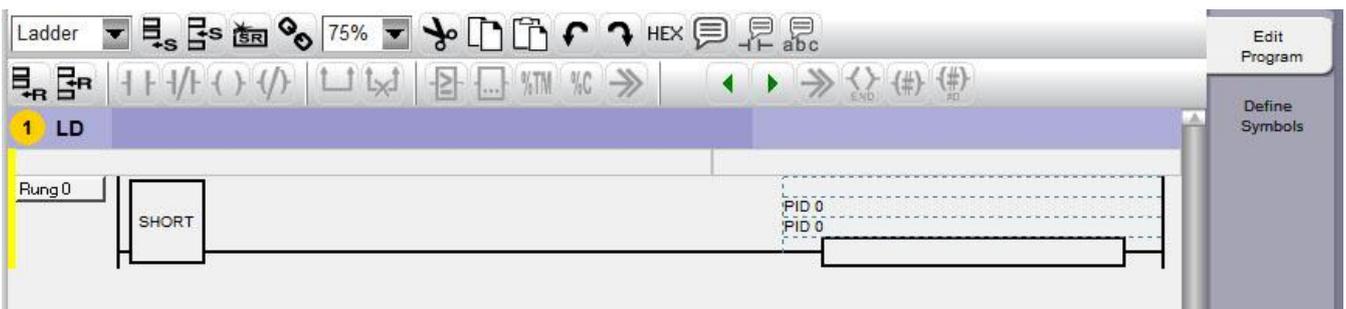
- Number of steps:** numero de passos que será realizado
Selecionar quais os bits que serão acionados no passo correspondente
- Outputs:** declarar os Bits de saída que serão atuados



PID

O regulador PID é uma função auxiliar de ajustes aplicado aos processos seqüenciais e processos simples.

Para utilizar a função PID, apenas endereçar no bloco operate (Ladder)



Configuração bloco PID

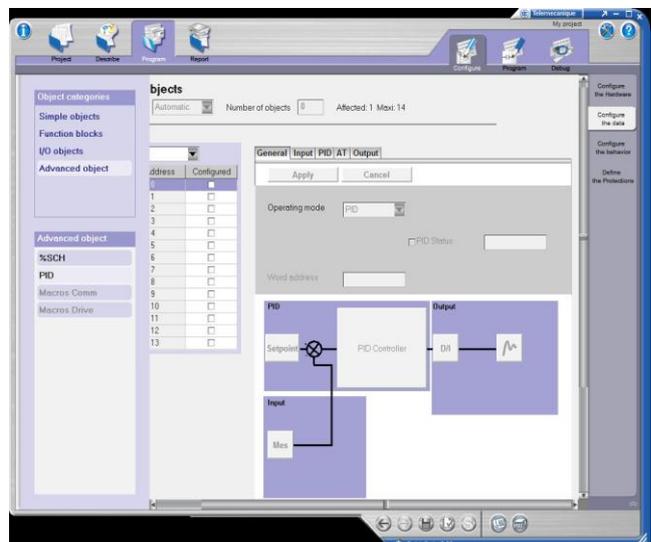
1^o passo habilitar a configuração do PID selecionando **Configured**

Used	Address	Configured
<input checked="" type="checkbox"/>	PID 0	<input type="checkbox"/>
<input type="checkbox"/>	PID 1	<input type="checkbox"/>
<input type="checkbox"/>	PID 2	<input type="checkbox"/>
<input type="checkbox"/>	PID 3	<input type="checkbox"/>
<input type="checkbox"/>	PID 4	<input type="checkbox"/>
<input type="checkbox"/>	PID 5	<input type="checkbox"/>
<input type="checkbox"/>	PID 6	<input type="checkbox"/>
<input type="checkbox"/>	PID 7	<input type="checkbox"/>
<input type="checkbox"/>	PID 8	<input type="checkbox"/>
<input type="checkbox"/>	PID 9	<input type="checkbox"/>
<input type="checkbox"/>	PID 10	<input type="checkbox"/>
<input type="checkbox"/>	PID 11	<input type="checkbox"/>
<input type="checkbox"/>	PID 12	<input type="checkbox"/>
<input type="checkbox"/>	PID 13	<input type="checkbox"/>

2^o passo escolher o modo de operação/controle através do **Operating mode**

Configuração General

1. **Operating mode:** selecionar o modo de operação
2. **PID Status:** status do PID podendo ser armazenado em uma memória
3. **Setpoint:** valor que se deseja obter
4. **Mes:** entrada onde se obtém o feedback do sistema



Configuração Input

1. **Measure:** variável de entrada analógica, onde se obtém o feedback do sistema
2. **Conversion:** autorizado, converte o valor de entrada measure, limitando entre mínimo e máximo
3. **Alarms:** alarmes, set um bit nas saídas para os valores que atinge nível alto e baixo

General	Input	PID	AT	Output
<input type="button" value="Apply"/> <input type="button" value="Cancel"/>				
Measure	Conversion	Alarms		
<input type="text"/>	<input type="checkbox"/> Authorize	<input type="checkbox"/> Authorize	Low	Output
	Min		<input type="text"/>	<input type="text"/>
	Max		High	Output
	<input type="text"/>		<input type="text"/>	<input type="text"/>

Configuração PID

1. **Setpoint:** valor que se deseja obter
2. **Parameters:** valores de Kp (constante proporcional), Ti (tempo da integral) e Td (tempo da derivada)
3. **Sampling Period:** intervalo de tempo entre as atualizações do sinal de saída

General | Input | PID | AT | Output

Apply Cancel

Setpoint	Corrector type	Parameters	Sampling Period
<input type="text"/>	PID	Kp (x 0,01) <input type="text"/>	(10ms)
		Ti (x 0,1s) <input type="text"/>	<input type="text"/>
		Td (x 0,1s) <input type="text"/>	

Configuração Output

1. **Action:** comportamento do sinal de saída reverso, direto ou definir reverso/direto através de um bit
2. **Limits:** limita a saída para o sistema, proíbe, autoriza ou proíbe/autoriza através de um bit
3. **Manual mode** valor a ser assumido pelo processo em uma saída analógica em modo manual proíbe, autoriza ou proíbe/autoriza através de um bit
4. **Output analog:** saída analógica para o sistema a ser controlado
5. **Output PWM:** habilita a saída pulsada

General | Input | PID | AT | Output

Apply Cancel

Action	Limits	Manual mode	Output analog	Output PWM
Reverse	Inhibit	Inhibit	<input type="text"/>	<input type="checkbox"/> Authorize Period (0,1s)
Bit <input type="text"/>	Min <input type="text"/> Max <input type="text"/>	Bit <input type="text"/> Output <input type="text"/>	<input type="text"/>	Output <input type="text"/>

Configuração AT

1. **AT mode:** habilita o modo auto tuning
2. **Measurement limit:** limite da entrada do processo
3. **Output setpoint:** variável de saída analógica quando o sistema estiver em auto tuning

General | Input | PID | AT | Output

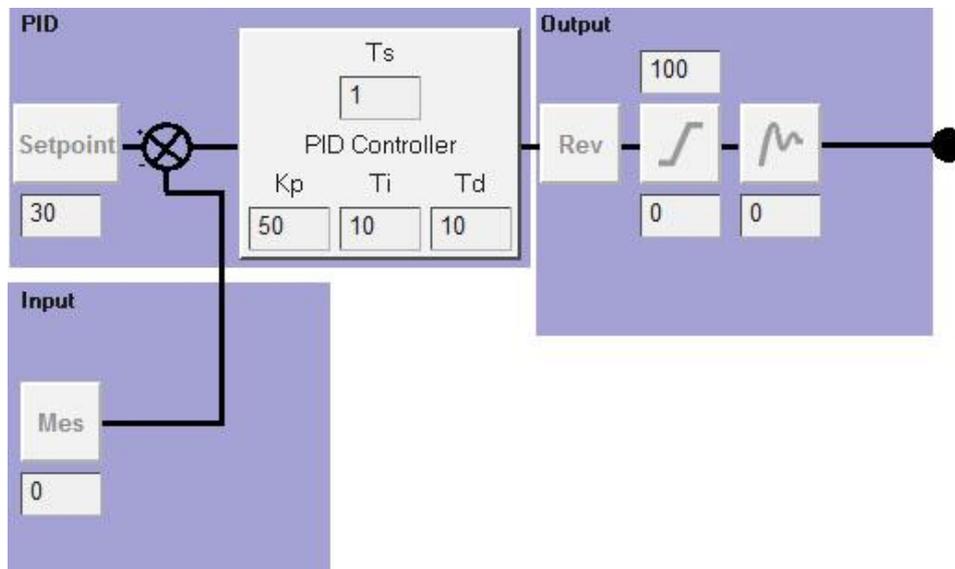
Apply Cancel

AT mode	Measurement limit	Output setpoint
<input type="checkbox"/> Authorize	<input type="text"/>	<input type="text"/>

Monitorar a aplicação PID

É possível visualizar os valores das variáveis e o comportamento do PID através do bloco PID e do gráfico

Acessar o Bloco PID na pasta Program / Configure - e selecionar o PID configurado na ultima janela irá aparecer a opção do gráfico



Bits e Palavras de Sistema

Bits de Sistema

%S0	: a 1, retorno a frio
%S1	: a 1, retorno a quente
%S4 a S7	: base de tempo 10 ms, 100 ms, 1 s, 1 mn.
%S8	: TSX37 não configurado, S8=1 saídas a 0, S8=0 saídas ajustáveis
%S9	: a 1, força as saídas repli
%S10	: a 0, defeito I/O
%S11	: a 1, tempo de WATCHDOG
%S13	: a 1, primeiro ciclo(scan) após RUN
%S15	: a 1, error cadeia de caracteres
%S16	: a 0, error entradas/saídas tarefa
%S17	: a 1 .ultrapassagem de palavra 16 bit
%S18	: a 1, ultrapassagem 15 bit + sinal ou error aritmético
%S19	: a 1, ultrapassagem tempo de tarefa periódica
%S20	: a 1, ultrapassagem de index
%S21	: a 1, inicialização grafcet
%S22	: a 1, reset das etapas
%S23	: a 1, congelamento do grafcet
%S26	: a 1, ultrapassagem de possibilidade de ativação grafcet
%S30	: a 0, desativa a tarefa mestre
%S31	: a 0, desativa a tarefa rápida
%S38	: a 0, desativa as tarefas de eventos
%S39	: a 1, saturação dos tratamentos de tarefas de eventos
%S40 a 47	: a 0, error entradas/saídas do rack correspondente 0 a 7
%S49	: a 1, rearmamento automático a cada 10 s., sadias estáticas
%S50	: a 1, escritura SW50 a 53, colocação em hora do relógio
%S51	: a 1, perda da hora do relógio
%S59	: a 1, ajuste da data atual, por incremento na SW59
%S66	: a 1, display 7 segmento (não disponível)
%S67	: a 0, pilha cartucho(PCMCIA) memória funcionando
%S68	: a 0, pilha memória RAM funcionando
%S69	: a 1, visualização de 16 words maxi em display frontal do controlador

Software

%S70	: a 1 pelo sistema refresco de words de intercâmbio em rede TSX 07
%S90	: a 1 Atualização de words comuns
%S98	: a 1 botão visualização mod. ASi SAZ10 substituído por uma entrada
%S99	: a 1 botão visualização frontal TSX37 substituído por uma entrada
%S100	: protocolo tomada terminal (0=Uni-TE , 1=ASCII)

Palavras de Sistema

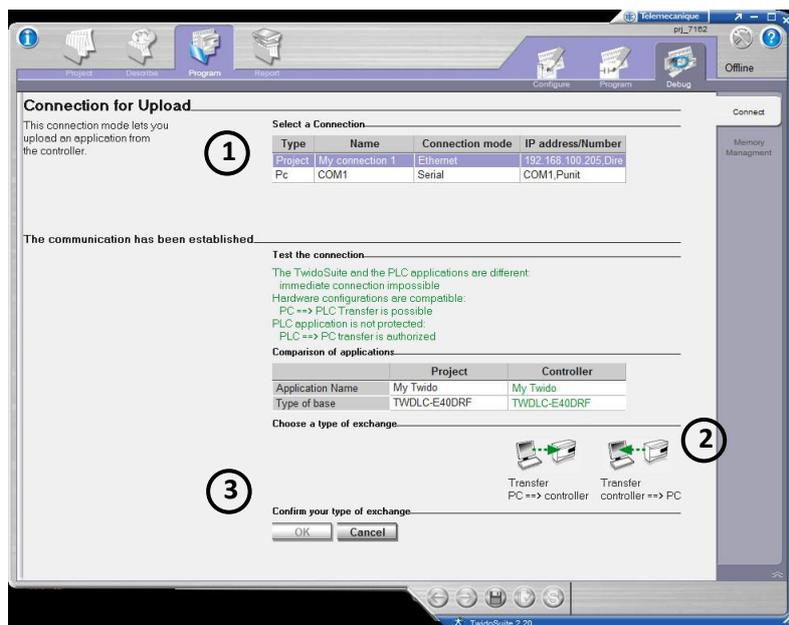
%SW0	: valor do período da tarefa mestre, periódica
%SW1	: valor do período tarefa <u>rápida</u>
%SW8	: controle da aquisição das entradas em cada tarefa. (SW8:0=T.mestre)
%SW9	: controle da atualização das saídas em cada tarefa. (SW8:0=T.mestre)
%SW10	: primeiro ciclo após retorno a frio
%SW11	: valor do WATCHDOG
%SW12	: endereço Unitelway tomada terminal
%SW13	: endereço principal estação
%SW17	: status de defeito c/ palavras flutuante
%SW18	: contador de tempo absoluto, incrementado a cada 100 ms.
%SW20	: numero etapas ativas, a ativar ou a desativar
%SW21	: numero transições validas, a validar ou a invalidar
%SW30	: Tempo do ultimo ciclo tarefa mestre
%SW31	: Tempo do ciclo máximo tarefa mestre
%SW32	: Tempo do ciclo mínimo tarefa mestre
%SW33	: Tempo do ultimo ciclo tarefa rápida
%SW34	: Tempo do ciclo máximo tarefa rápida
%SW35	: Tempo do ciclo mínimo tarefa rápida
%SW48	: Numero de evento tratados
%SW49	: Função relógio calendário Corrente: dia da semana;
%SW50	: Função relógio calendário Corrente : segundos
%SW51	: Função relógio calendário Corrente : horas e minutos
%SW52	: Função relógio calendário Corrente : mês e dia
%SW53	: Função relógio calendário Corrente : século e ano
%SW54	: Função relógio calendário Ultima parada defeito ou stop: segundos e cod.defeito
%SW55	: Função relógio calendário Ultima parada defeito ou stop: horas e minutos
%SW56	: Função relógio calendário Ultima parada defeito ou stop: mês e dia
%SW57	: Função relógio calendário Ultima parada defeito ou stop : século e ano
%SW58	: código ultima parada e dia da semana
%SW59	: Ajuste incrementai da data e hora para o display frontal TSX37
%SW67	: comando e estado (default %MW, hexadecimal,incremento)
%SW68	: índice máximo e índice corrente (default palavras .0 a 15)
%SW69	: endereço da primeira palavra (default %MW0)
%SW80 a 86	: Gestão mensagens e telegrama
%SW96	: Controle/Diag. de funções de save/retrieve
%SW97	: Numero de %MW a ser salvaguardadas
%SW98	: Endereço entrada digital p/ B. DIAG. (modulo SAZ10)
%SW99	: Endereço entrada digital p/ B. DIAG. (bloco display)
%SW108	: números de bit forçados
%SW109	: números de vias analógicas forçados
%SW124	: tipo de ultimo defeito UC encontrado
%SW125	: tipo de defeito bloqueante
%SW126 a 127	: endereço de instrução de defeito bloqueante
%SW128 a 155	: FIPIO

Debug



Transferência do programa para o controlador PLC

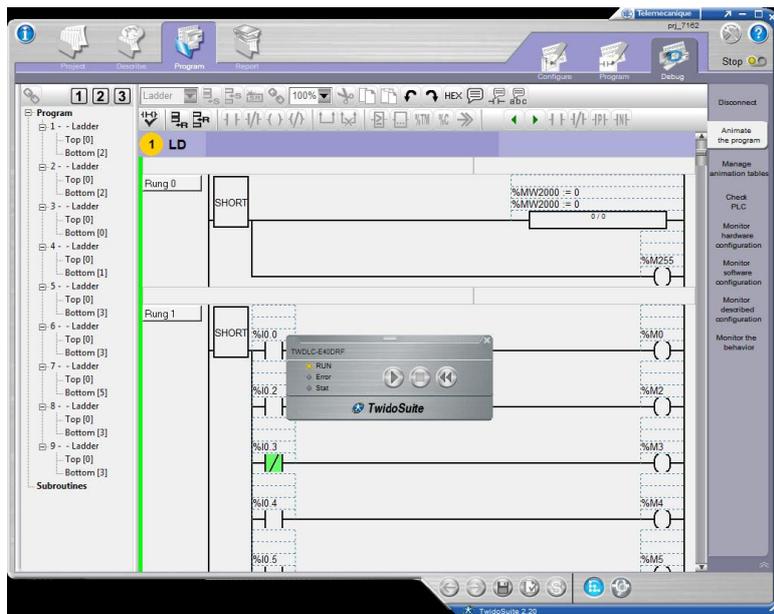
1. Selecionar a conexão verificando qual a porta de comunicação está ligado o PLC;
2. Selecionar a troca de informações, se o programa será transferido do PC para PLC ou PLC para o PC;
3. Confirmar o tipo de troca de informações;



4. Na janela que aparecer após, clicar em "Yes" para confirmar a transferência;



5. Realizada a transferência do programa, aparecerá um pequeno console para controle do PLC. Dê um clique no ícone  para colocar o PLC em modo **RUN**.



6. Clique em **OK** para confirmar o procedimento.

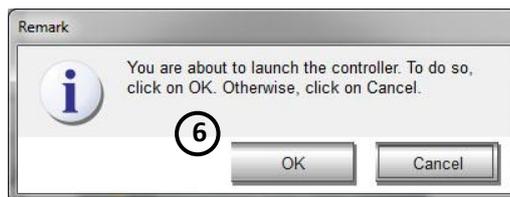
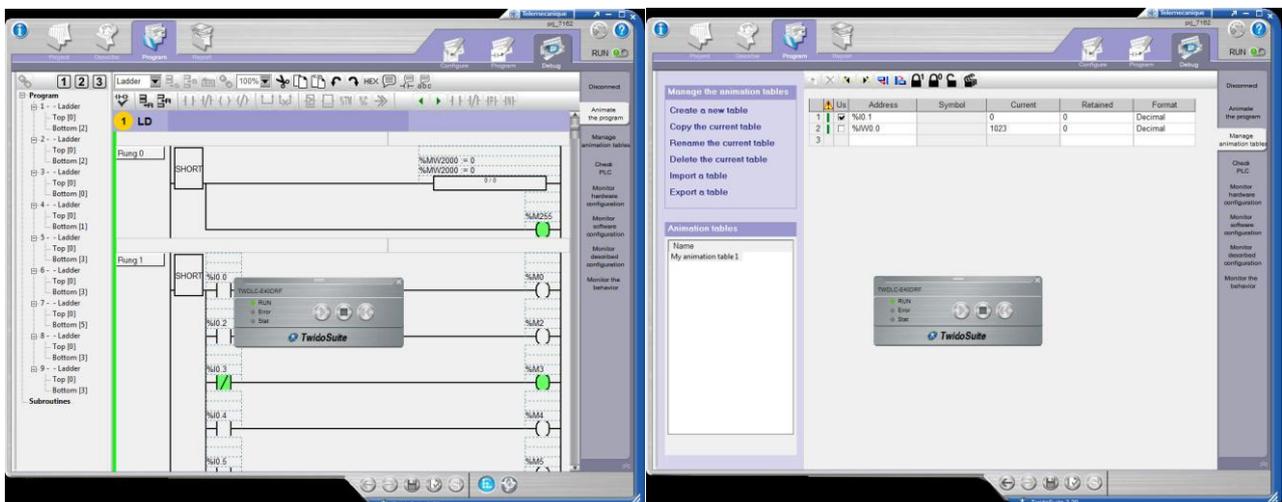


Tabela de animação



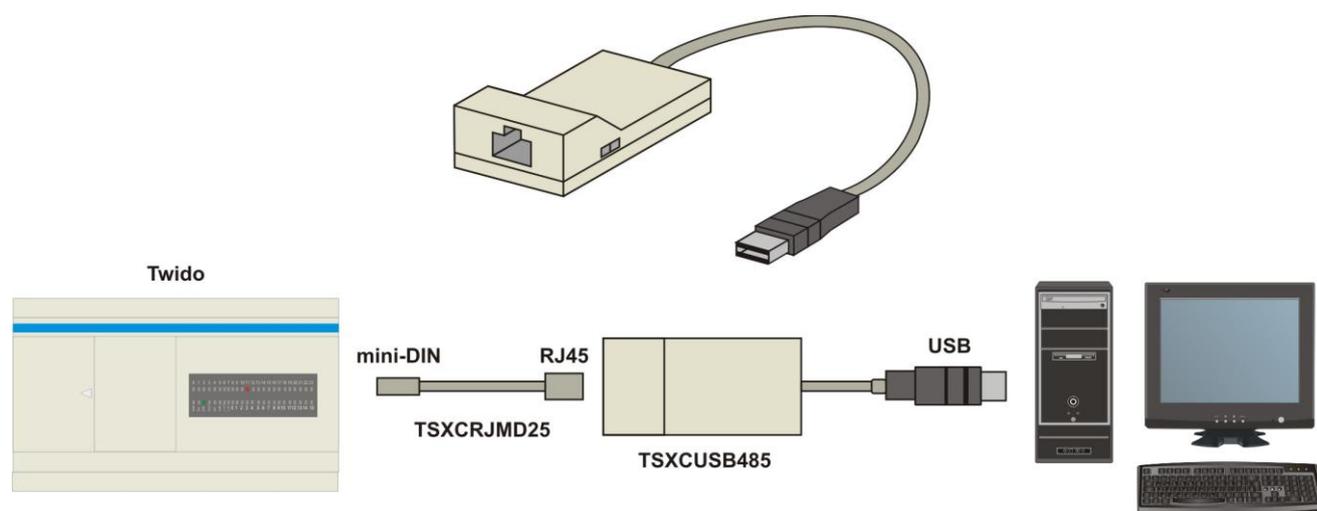
Monitorea a aplicação e possibilite de alterar e monitorea valor de variáveis através **manage animation tables** do lado esquerdo da pagina.

Cabo de Programação e Conversores

Conversor: USB - RS485: TSXCUSB485

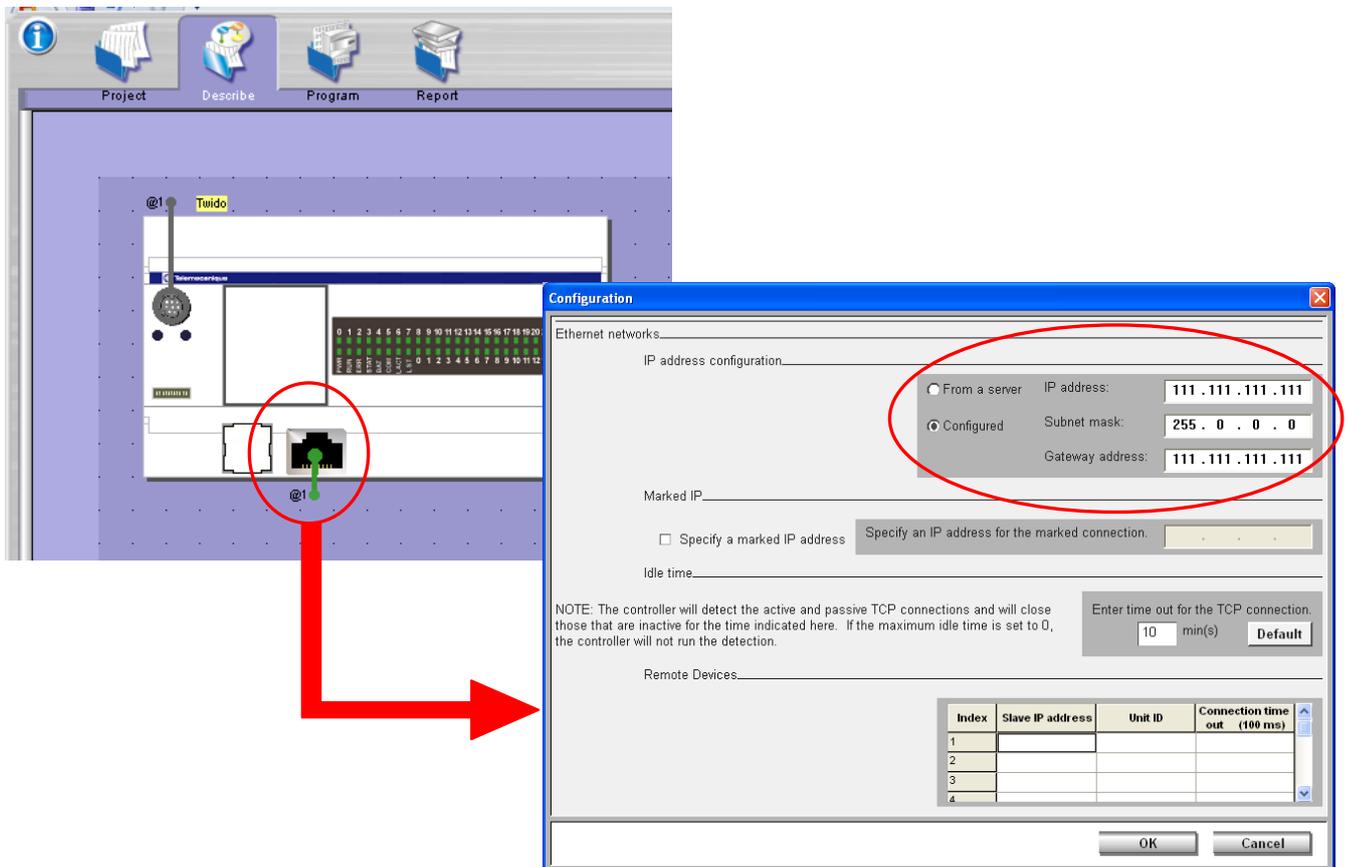
O conversor TSXCUSB485 é um dispositivo que permite um PC se conectar pela sua porta USB a dispositivos remotos usando interface serial RS 485. Este dispositivo é totalmente compatível com os protocolos Modbus e Unitelway mas requer a instalação dos drives padrões Schneider.

Para fazer realizar a conexão do PC com o PLC Twido para programação, utiliza-se o conversor TSXCUSB485 juntamente com o cabo TSX CRJMD 25, como mostrado abaixo:



Para controladores Twido que possuam porta de comunicação ethernet, existe a possibilidade de programação através de um cabo ethernet crossover ou normal dependendo da situação (conexão PC – CLP ou PC – SWITCH – CLP).

Após a criação do projeto no Twido Suite, configurar um IP address válido (Menu DESCRIBE >> duplo clique sobre o desenho da porta ethernet):



Obs.: a subnet mask e o gateway address são preenchidos automaticamente assim que selecionados estes campos.

O controlador que não possui IP address configurado pelo usuário, permanece com o endereço padrão definido pelo MAC address. O MAC address está localizado ao lado da porta de comunicação principal:



O endereço IP padrão, expresso em notação decimal é definido como segue:

085.016.xxx.yyy, onde:

- 085.016. é um cabeçalho compartilhado por todos os endereços IP derivados do endereço MAC;
- xxx e yyy são os dois últimos números do endereço MAC.

Por exemplo, o endereço IP derivado do endereço MAC 00.80.F4.81.01.11 é 085.016.001.17.

Para realizar o download da aplicação em um controlador via porta ethernet, proceder da seguinte forma:

1. Criar a conexão via ethernet através do caminho: Menu PROGRAM >> DEBUG >> CONNECT >> MANAGE CONNECTION;
2. Adicionar nova conexão;
3. Seleciona modo de conexão Ethernet e digitar o IP address do CLP a ser programado;
4. Aplicar as alterações;
5. Seguir procedimentos de download descritos na seção de transferência de programa;

Se a conexão estabelecida entre o PC e o CLP é ponto a ponto, o PC deve ser configurado com outro endereço IP e de mesmo cabeçalho que o IP address do CLP.

